

CA Business Service Insight

Guida all'implementazione 8.2.5



La presente documentazione, che include il sistema di guida in linea integrato e materiale distribuibile elettronicamente (d'ora in avanti indicata come "Documentazione"), viene fornita all'utente finale a scopo puramente informativo e può essere modificata o ritirata da CA in qualsiasi momento.

Questa Documentazione non può essere copiata, trasmessa, riprodotta, divulgata, modificata o duplicata per intero o in parte, senza la preventiva autorizzazione scritta di CA. Questa Documentazione è di proprietà di CA e non potrà essere divulgata o utilizzata se non per gli scopi previsti in (i) uno specifico contratto tra l'utente e CA in merito all'uso del software CA cui la Documentazione attiene o in (ii) un determinato accordo di confidenzialità tra l'utente e CA.

Fermo restando quanto enunciato sopra, se l'utente dispone di una licenza per l'utilizzo dei software a cui fa riferimento la Documentazione avrà diritto ad effettuare copie della suddetta Documentazione in un numero ragionevole per uso personale e dei propri impiegati, a condizione che su ogni copia riprodotta siano apposti tutti gli avvisi e le note sul copyright di CA.

Il diritto a stampare copie della presente Documentazione è limitato al periodo di validità della licenza per il prodotto. Qualora e per qualunque motivo la licenza dovesse cessare o giungere a scadenza, l'utente avrà la responsabilità di certificare a CA per iscritto che tutte le copie anche parziali del prodotto sono state restituite a CA o distrutte.

NEI LIMITI CONSENTITI DALLA LEGGE VIGENTE, LA DOCUMENTAZIONE VIENE FORNITA "COSÌ COM'È" SENZA GARANZIE DI ALCUN TIPO, INCLUSE, IN VIA ESEMPLIFICATIVA, LE GARANZIE IMPLICITE DI COMMERCIALIZZABILITÀ, IDONEITÀ A UN DETERMINATO SCOPO O DI NON VIOLAZIONE DEI DIRITTI ALTRI. IN NESSUN CASO CA SARÀ RITENUTA RESPONSABILE DA PARTE DELL'UTENTE FINALE O DA TERZE PARTI PER PERDITE O DANNI, DIRETTI O INDIRETTI, DERIVANTI DALL'UTILIZZO DELLA DOCUMENTAZIONE, INCLUSI, IN VIA ESEMPLIFICATIVA E NON ESAUSTIVA, PERDITE DI PROFITTI, INTERRUZIONI DELL'ATTIVITÀ, PERDITA DEL GOODWILL O DI DATI, ANCHE NEL CASO IN CUI CA VENGA ESPRESSAMENTE INFORMATO IN ANTICIPO DI TALI PERDITE O DANNI.

L'utilizzo di qualsiasi altro prodotto software citato nella Documentazione è soggetto ai termini di cui al contratto di licenza applicabile, il quale non viene in alcun modo modificato dalle previsioni del presente avviso.

Il produttore di questa Documentazione è CA.

Questa Documentazione è fornita con "Diritti limitati". L'uso, la duplicazione o la divulgazione da parte del governo degli Stati Uniti è soggetto alle restrizioni elencate nella normativa FAR, sezioni 12.212, 52.227-14 e 52.227-19(c)(1) - (2) e nella normativa DFARS, sezione 252.227-7014(b)(3), se applicabile, o successive.

Copyright © 2013 CA. Tutti i diritti riservati. Tutti i marchi, i nomi commerciali, i marchi di servizio e i loghi citati nel presente documento sono di proprietà delle rispettive aziende.

Contattare il servizio di Supporto tecnico

Per l'assistenza tecnica in linea e un elenco completo delle sedi, degli orari del servizio di assistenza e dei numeri di telefono, contattare il Supporto tecnico visitando il sito Web all'indirizzo <http://www.ca.com/worldwide>.

Sommario

Capitolo 1: Introduzione 9

Ruoli	9
Responsabile del catalogo dei servizi.....	10
Responsabile contratti	10
Esperto di business logic	11
Esperto delle sorgenti dati	11
Amministratore di sistema	12
Processo di soluzione	12
Pianificazione	13
Progettazione.....	14
Implementazione	16
Installazione e deployment	16

Capitolo 2: Pianificazione e progettazione 17

Sessione di raccolta dei requisiti (Tutti i ruoli)	18
Raccolta dei dati di esempio (esperto delle sorgenti dati)	20
Progettazione	21
Introduzione alla progettazione (responsabile contratti, esperto di business logic, esperto delle sorgenti dati)	22
Modellazione dei contratti (responsabile contratti)	23
Output della fase di modellazione dei contratti (responsabile contratti, esperto delle sorgenti dati)	42
Modellazione dei dati (Esperto delle sorgenti dati, Esperto di business logic)	43
Output della fase di modellazione dei dati (esperto delle sorgenti dati ed esperto di business logic)	58

Capitolo 3: Implementazione 59

Implementazione - Introduzione.....	59
Configurazione del framework (responsabile contratti)	62
Configurazione della libreria dei modelli (responsabile contratti)	62
Modalità di creazione dei contratti (responsabile contratti)	63
Creazione di contratti da un servizio.....	64
Creazione dei modelli del livello di servizio	66
Ciclo di vita di un contratto e il controllo delle versioni	66
Raccolta dei dati (esperto delle sorgenti dati)	68
Funzionalità dell'adapter.....	69
Ambiente dell'adapter	70
File principali	73

File di lavoro	73
Comunicazione dell'adapter	80
Impostazioni del registro di sistema dell'adapter	83
Modalità di esecuzione dell'adapter	85
Strumenti di configurazione e manutenzione	87
Configurazione di adapter e conversioni	87
Conversioni automatiche con script di conversione	125
Argomenti avanzati della funzionalità dell'adapter	126
Implementazione di business logic (esperto di business logic)	134
Flusso di lavoro di implementazione di business logic	136
Moduli di business logic	137
All'interno della business logic	138
Attivazione dei contratti (responsabile contratti)	177
Ricalcolo completo delle metriche del contratto	179
Creazione dei risultati finali (responsabile contratti)	180
Definizione delle impostazioni di protezione (amministratore)	180
Creazione di report	181
Configurazione delle pagine del dashboard	194
Aggiunta dei profili di notifica del livello di servizio	197

Capitolo 4: Installazione e deployment **201**

Introduzione	202
Preparazione	204
Installazione	206
Importazione o esportazione tra ambienti (esperto delle sorgenti dati)	207
Integrazione - Configurazione del server di posta elettronica (amministratore di sistema)	209
Impostazione delle preferenze di sistema (amministratore di sistema)	210
Impostazioni di protezione (amministratore di sistema)	210
Definizione delle impostazioni per la sincronizzazione SSA	211

Appendice A: Domini del servizio e categorie di dominio di esempio **213**

Appendice B: Esempi di case study **215**

Esempi di modellazione di contratto	215
Case study 1: disponibilità del sistema	216
Case study 2: disponibilità del sistema 2	217
Case study 3: tempi di risposta del sistema	219
Case study 4: prestazioni dell'assistenza tecnica	223
Case study 5: backup del sistema	225
Esempio di modellazione della metrica finanziaria	226

Case study 6: modellazione delle condizioni finanziarie di un contratto/servizio	226
Esempi di modellazione dei dati	234
Case study 7: prestazioni del server	234
Case study 8: prestazioni dell'assistenza tecnica	237
Esempio di utilizzo degli attributi personalizzati	245
Case study 9: più destinazioni dinamiche	245
Esempi di script di conversione	249
Case study 10: conversione automatica di base	249
Case study 11: aggiornamenti del modello di risorsa	252
Esempi di implementazione di business logic	255
Case study 12: utilizzo della logica di contatore per calcolare il numero di errori	256
Case study 13: gestione del gruppo di componenti dinamico	260
Case study 14: gestione clock di controllo del tempo.....	265
Esempi di scrittura di business logic efficace	271
Case study 15: metriche di gruppo	274
Case study 16: modelli di progettazione di business logic	278
Case study 17: funzionalità integrata.....	283
Case study 18: registrazione	285
Case study 19: procedura guidata di configurazione dell'adapter per origine dati basata su file	288
Creazione dell'adapter	290
Distribuzione degli adapter	302
Pianificazione dell'adapter	303
Case study 21: esempio di integrazione con LDAP	306
Case study 22: generazione di report con PERL	313

Appendice C: Specifiche di configurazione dell'adapter **317**

Struttura generale	317
Sezione General.....	318
Sezione Interface CA Business Service Insight	324
Sezione DataSourceInterface	327
Sezione SQL Interface.....	330
Sezione InputFormatCollection	334
Sezione TranslationTableCollection	339
Sezione TranslatorCollection.....	341

Appendice D: Definizione delle formule di business logic (esperto di business logic) **345**

Azioni da evitare durante la creazione di formule di business logic	345
Metriche di gruppo ed efficacia delle risorse.....	345

Capitolo 1: Introduzione

Questa guida descrive il processo di implementazione e i relativi ruoli, responsabilità e interazioni coinvolti nella pianificazione, progettazione, implementazione, installazione e distribuzione di CA Business Service Insight.

Questa sezione contiene i seguenti argomenti:

[Ruoli](#) (a pagina 9)

[Processo di soluzione](#) (a pagina 12)

Ruoli

Un ruolo è un utente che esegue un insieme comune di attività effettuate durante un'implementazione tipica. Il termine ruolo può fare riferimento anche alla stessa serie di attività. CA Business Service Insight ha un numero di ruoli predefiniti che possono essere modificati dall'amministratore di sistema. Inoltre, è possibile creare autorizzazioni specifiche e ruoli nuovi.

Per l'implementazione sono richiesti i ruoli seguenti:

- Responsabile del catalogo dei servizi
- Responsabile contratti
- Esperto di business logic
- Esperto delle sorgenti dati
- Amministratore di sistema

Le responsabilità e le qualifiche di ciascun ruolo vengono descritte di seguito.

Nota: lo stesso utente può eseguire diversi ruoli, come definito dall'amministratore di sistema.

Responsabile del catalogo dei servizi

Responsabilità:

- Gestione del catalogo dei servizi organizzativi
- Definizione del catalogo di offerta dei servizi dell'organizzazione
- Gestione dei requisiti per i report e le definizioni di contratto

Requisiti obbligatori:

- Dimestichezza con il processo di fornitura dei servizi E2E dell'organizzazione
- Dimestichezza con i concetti di CA Business Service Insight

Responsabile contratti

Responsabilità:

- Interazione con l'utente finale
- Realizzazione del catalogo dei servizi in base a requisiti definiti
- Creazione di contratti nuovi e mantenimento dei contratti esistenti
- (Facoltativo, consigliato) Responsabilità dell'implementazione di SLA per clienti specifici

Qualifiche:

- Comprensione dei principi e della logica delle metriche
- Utilizzo esperto delle funzionalità basate sulle GUI di CA Business Service Insight

Esperto di business logic

Responsabilità:

- Implementazione delle metriche
- Redazione di script di logica aziendale; mantenimento degli script di logica aziendale esistenti

Qualifiche:

- Nozioni di sviluppo di base e conoscenza approfondita del funzionamento dei linguaggi di scripting come VB-Script
- Buona conoscenza del flusso dati di CA Business Service Insight
- Esperto nell'implementazione di business logic di CA Business Service Insight
- Buona comprensione dell'architettura e dei componenti CA Business Service Insight
- Utilizzo esperto delle funzionalità basate sulle GUI di CA Business Service Insight

Esperto delle sorgenti dati

Responsabilità:

- Progettazione dell'infrastruttura CA Business Service Insight
- Creazione di nuovi adapter; mantenimento degli adapter esistenti, interazione con l'infrastruttura operativa per ottenere misure
- Sviluppo e mantenimento dell'infrastruttura di CA Business Service Insight

Qualifiche:

- Comprensione della struttura e dell'ambiente delle origini dati del cliente
- Buona conoscenza di lavoro di database, XML e linguaggi di scripting
- Comprensione dei processi di raccolta dei dati e di flusso dei dati di CA Business Service Insight
- Esperto in adapter CA Business Service Insight
- Comprensione dei componenti e dell'architettura di CA Business Service Insight
- Utilizzo esperto delle funzionalità basate sulle GUI di CA Business Service Insight
- Possesso di nozioni di sviluppo è un vantaggio

Amministratore di sistema

Responsabilità:

- Gestione di installazione e aggiornamenti
- Soddisfazione dei requisiti di sistema per hardware e software
- Gestione delle impostazioni di protezione: definizioni di ruolo e utente
- Controllo e gestione del sistema

Qualifiche:

- Comprensione delle reti e delle infrastrutture del cliente
- Comprensione dei componenti e dell'architettura di CA Business Service Insight
- Conoscenza del funzionamento di linguaggi di scripting, XML e database
- Comprensione del flusso di dati di CA Business Service Insight
- Utilizzo delle funzionalità basate sulle GUI di CA Business Service Insight

Processo di soluzione

Il processo di soluzione include tre fasi di base:

- Pianificazione e progettazione
- Implementazione
- Installazione e deployment

I ruoli descritti in precedenza partecipano alle varie fasi del processo di implementazione, ognuno in base alle proprie competenze. I ruoli interagiscono per completare l'intero processo dall'inizio alla fine, dalla definizione del contratto al report di output.

La struttura di questa guida è orientata al ruolo e al processo, in base al flusso generale delle fasi che costituiscono il processo di implementazione. La guida descrive il modo in cui ciascun ruolo è coinvolto e interagisce.

Il seguente paragrafo offre una panoramica delle fasi incluse nel processo di implementazione e nelle attività di ogni ruolo.

La parte restante della guida è strutturata per fasi con l'indicazione di quali ruoli partecipano alla fase. Ogni capitolo indica il responsabile di ruolo per ciascuna attività.

Di seguito è riportata una breve descrizione delle attività di base per ogni fase e le funzioni dei diversi ruoli. In ogni capitolo sono riportate ulteriori informazioni.

Pianificazione

Lo scopo della fase di pianificazione è identificare e quantificare tutti gli aspetti che devono essere considerati come parte dell'implementazione.

In questa fase, il responsabile contratti raccoglie i requisiti di business dal responsabile del catalogo dei servizi per comprendere le aspettative da CA Business Service Insight in seguito a un'implementazione riuscita. In questa fase è importante capire sia i requisiti attuali sia i piani futuri per assicurarsi che l'implementazione consenta una crescita e un'evoluzione facili e uniformi.

In base ai requisiti di implementazione definiti, è necessario che il responsabile contratti esamini il relativo processo esistente (se esiste) controllando input e output del processo esistente. In questa fase è necessario identificare tutte le origini di informazioni richieste e ottenere gli esempi, i contratti, i report di output e le origini di input utilizzate per calcolare gli output esistenti. È necessario specificare attentamente queste informazioni affinché il responsabile contratti identifichi tutti gli input necessari, in modo da derivare gli output previsti.

In questa fase, il responsabile contratti esamina i contratti per verificare che il catalogo dei servizi e i modelli forniti come parte dell'implementazione soddisfino le necessità esistenti e future, e che l'implementazione corrente copra tutte le aree del contratto.

Il responsabile contratti esamina i report e i relativi formati per verificare che tutte le misurazioni definite siano eseguite in modo da crearli con l'implementazione esistente.

L'esperto delle sorgenti dati quindi analizza i campioni dei dati di input per le misure e i calcoli necessari specificati dal responsabile contratti. Questo consente all'esperto delle sorgenti dati di identificare eventuali problemi di input che devono essere risolti, ad esempio formati personalizzati o lacune, e di determinare se sono necessarie origini dati aggiuntive.

Lo scopo di questo esame è verificare che le origini contengano le informazioni necessarie per calcolare le metriche richieste e le informazioni iniziali sul processo di recupero, ad esempio i metodi di comunicazione con le origini dati e la struttura dell'origine dati.

Progettazione

Durante la fase di progettazione, i requisiti raccolti di input e di output vengono convertiti nella struttura di modello CA Business Service Insight che comprende contratti, metriche e risorse. Questa operazione comporta la trasformazione e la concettualizzazione dei dati reali in modo che siano adatti per il framework CA Business Service Insight.

La progettazione di sistema comprende le seguenti operazioni:

Modellazione dei contratti

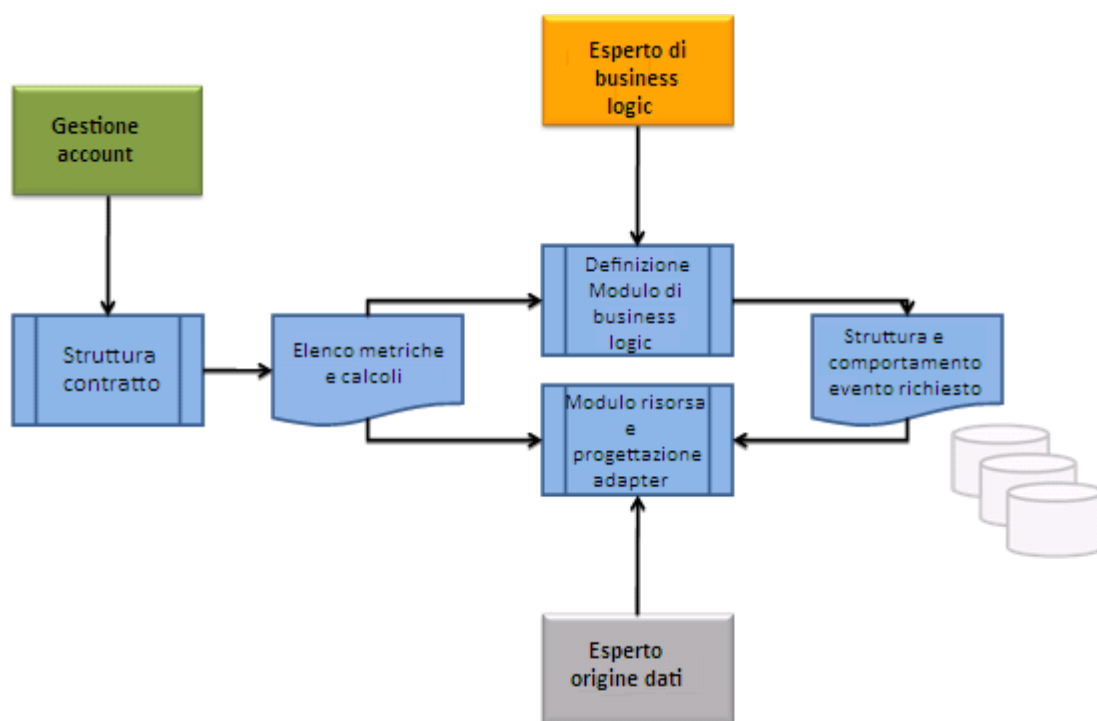
Gli accordi SLA del cliente vengono interpretati in contratti e vengono definite le entità del catalogo dei servizi CA Business Service Insight (ad esempio, le cartelle dei modelli). Questa operazione viene eseguita principalmente dal responsabile contratti.

Modellazione dei dati

I dati della risorsa vengono analizzati e modellati nel modello di risorsa CA Business Service Insight. Questa operazione viene eseguita dall'esperto delle sorgenti dati e dall'esperto di business logic.

Possono essere disponibili diversi metodi per la modellazione dei contratti o dei dati. Il più delle volte, tuttavia, esiste una procedura ottimale che non solo risolve il problema, ma offre anche una maggiore flessibilità o produttività, fornendo così un framework sicuro per un'ulteriore crescita.

Per agevolare l'esperto delle sorgenti dati nella selezione dei metodi più appropriati, vengono utilizzati case study con le soluzioni consigliate.



Come mostrato nel precedente diagramma del flusso di lavoro, come conseguenza del processo di modellazione dei contratti, il responsabile contratti fornisce come output l'elenco delle metriche che devono essere configurate nel sistema e la definizione del relativo schema di calcolo.

Esempio:

Cliente A, Tempo di risposta medio del sistema CNP.

L'elenco delle metriche viene fornito come input per l'esperto di business logic che definisce la struttura dell'evento richiesto e il comportamento dell'evento che consente la definizione dello script di calcolo necessario.

L'elenco delle metriche e i requisiti di struttura e comportamento dell'evento sono gli input per la progettazione del modello di risorsa e degli adapter dei dati da parte dell'esperto delle sorgenti dati.

Implementazione

In fase di implementazione, il sistema CA Business Service Insight viene configurato in base ai risultati della fase di progettazione. La fase di implementazione implica l'acquisizione dei risultati della fase di progettazione teorica e il loro utilizzo per creare i requisiti funzionali per CA Business Service Insight.

Alcuni di questi elementi da creare o trattare includono:

Responsabile contratti

- Configurazione dei servizi
- Creazione di contratti
- Configurazione di report e dashboard

Esperto delle sorgenti dati

- Configurazione dell'adapter
- Configurazione dell'infrastruttura

Esperto di business logic

- Moduli di business logic

Installazione e deployment

La fase di installazione e deployment è legata a installazione e integrazione del sistema di produzione, test e monitoraggio delle prestazioni e formazione degli utenti. In questa fase, la maggior parte delle attività viene eseguita dall'amministratore di sistema e dall'esperto delle sorgenti dati.

Capitolo 2: Pianificazione e progettazione

Questa sezione contiene i seguenti argomenti:

[Sessione di raccolta dei requisiti \(Tutti i ruoli\)](#) (a pagina 18)

[Raccolta dei dati di esempio \(esperto delle sorgenti dati\)](#) (a pagina 20)

[Progettazione](#) (a pagina 21)

Sessione di raccolta dei requisiti (Tutti i ruoli)

La sessione di raccolta dei requisiti costituisce il primo passaggio cruciale nell'implementazione di CA Business Service Insight e deve includere tutte le figure chiave coinvolte nell'implementazione.

Le informazioni necessarie per questa sessione devono essere fornite da utenti che hanno una notevole dimestichezza con gli SLA, compresi la business logic degli SLA, le modalità di gestione attuali degli SLA, i tipi di report generati e i requisiti di report completi previsti in futuro.

È richiesta la partecipazione dei ruoli seguenti:

- Responsabili del catalogo dei servizi
- Responsabili contratti
- Esperti delle sorgenti dati
- Esperti di business logic o di scripting
- Qualsiasi altro membro interessato del personale che abbia dimestichezza con gli SLA
- Responsabile di progetto (se nominato)

CA consiglia quanto segue:

- È importante assicurare la presenza di personale che possa prendere decisioni in merito a qualsiasi definizione poco chiara che potrebbe essere riscontrata durante la revisione degli SLA selezionati.
- È consigliabile che il team di implementazione riceva, prima di riunirsi, una copia degli SLA selezionati per il progetto. I partecipanti devono avere dimestichezza con tutte le origini dati pertinenti che riguardano gli SLA selezionati, essere in grado di fornire dump di dati relativi e spiegare le strutture interne, se necessario.

Gli obiettivi principali della sessione di pianificazione sono definire e specificare:

- Ambito di lavoro pianificato
- Criteri di successo
- Ruoli e responsabilità di tutte le parti coinvolte
- Processo di implementazione e pianificazione
- Requisiti hardware/software obbligatori

Tali obiettivi sono ottenuti mediante:

- Revisione degli SLA da usare nell'implementazione
- Revisione della business logic per ciascun obiettivo all'interno degli SLA selezionati
- Revisione di report, notifiche e requisiti dashboard relativi agli SLA selezionati

- Revisione delle origini dati pertinenti da utilizzare nell'implementazione
- Revisione della topologia di infrastruttura pertinente
- Raccolta dei dump di dati pertinenti dalle relative origini dati
- Identificazione delle figure chiave e rispettive responsabilità
- Definizione di canali di comunicazione per la durata dell'implementazione, riunioni di revisione, processo Q&A, e così via.
- Esame della pianificazione e del processo di implementazione
- Revisione dello stato corrente di affari, vale a dire, delle modalità di gestione degli SLA e degli elementi mancanti
- Definizione dei criteri per una corretta implementazione
- Definizione dei requisiti hardware/software obbligatori
- Definizione della sequenza temporale per l'implementazione

Raccolta dei dati di esempio (esperto delle sorgenti dati)

Per eseguire la configurazione iniziale degli adapter in posizione remota, ai fini della raccolta di dati è importante ottenere dati precedenti di esempio. Questi esempi devono avere la stessa struttura dei dati che vengono visualizzati in seguito, provenienti dal sistema effettivo da cui è necessario recuperare i dati per CA Business Service Insight. Oltre agli esempi, l'esperto delle sorgenti dati deve informarsi sull'origine dati dal titolare dell'origine dati per identificare i problemi seguenti:

Tipo:

Database, file, flusso o altro.

Ubicazione:

Dove si trova? Come accedervi (dettagli di connessione)? Ostacoli di protezione? Piattaforma?

Convenzione di denominazione:

Come sono denominati i file? In caso di origine dati basata su file, come sono denominate le tabelle? I nomi dei campi di tabella?

Disponibilità:

Quando è accessibile? Quando l'accesso è obbligatorio (considerazioni di caricamento)? Aggiornamento continuo oppure una volta ogni X ore? Per quanto tempo è permanente?

Comportamento:

I dati vengono accumulati? I dati vengono sovrascritti? Viene conservato un archivio?

Periodo:

Per quanto i dati cronologici sono disponibili?

Volumi:

Volume corrente dei dati? Tasso di crescita? Stime?

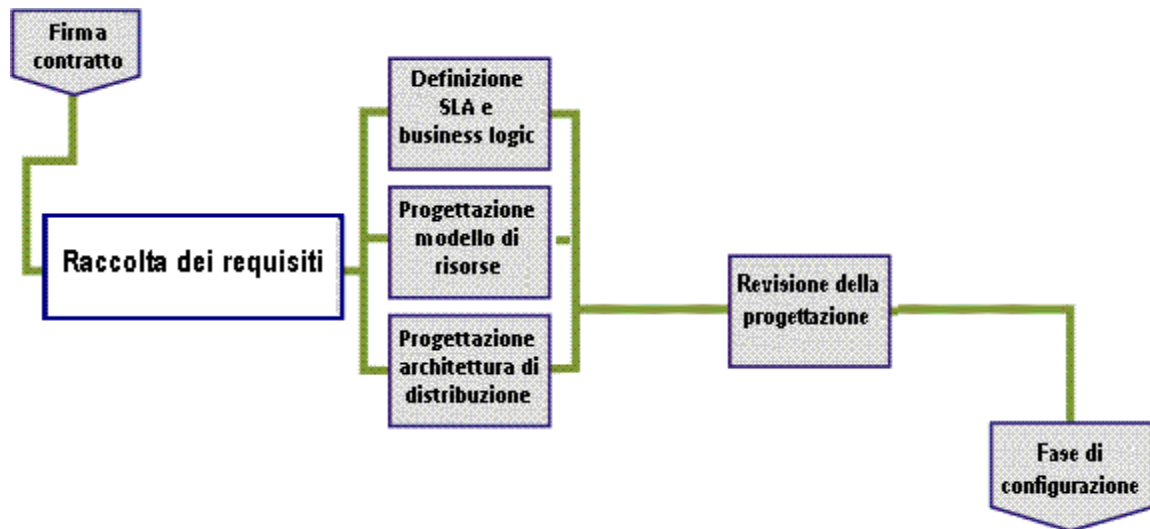
Struttura e formato:

Come sono organizzati i dati nell'origine dati? Quali sono i campi di dati? Quali sono i nomi di tabella? Che cosa separa i campi di dati?

Flusso:

L'adapter riceve i dati oppure i dati vengono raccolti dall'adapter?

Progettazione



Introduzione alla progettazione (responsabile contratti, esperto di business logic, esperto delle sorgenti dati)

In questa sezione vengono descritti il processo e la logica di base della fase di progettazione del processo di soluzione. La fase di progettazione segue la fase di pianificazione ed è a sua volta seguita dalla fase di implementazione, descritta nel prossimo capitolo.

L'obiettivo della fase di progettazione per il team di implementazione è poter completare il mapping dei contratti reali e delle relative clausole e dei dati sulle prestazioni esistenti sul sistema CA Business Service Insight.

Prima di avviare il processo, il team di implementazione deve essere a conoscenza dei vari metodi e opzioni disponibili in modo da garantire che non solo si tenga conto di tutti i requisiti, ma che la progettazione risultante sia ottimizzata al massimo, consentendo al tempo stesso l'espansione o modifica successiva.

Il processo di progettazione coinvolge il team di implementazione nell'esecuzione dei seguenti passaggi:

- Esaminare i contratti e convertirli in oggetti necessari CA Business Service Insight, processo definito in questa guida come "modellazione dei contratti"; la responsabilità è del responsabile contratti.
- Accettare gli insiemi di dati esistenti e determinare quali elementi pertinenti e come estrarli in luce dei risultati desiderati, processo definito in questa guida come "modellazione di dati"; la responsabilità è dell'esperto delle sorgenti dati e dell'esperto di business logic.

La sezione Modellazione dei contratti illustra:

- Terminologia utilizzata (essenziale per la corretta implementazione)
- Modelli e moduli di business logic
- Modello del livello di servizio e modalità di utilizzo
- Importanza della creazione di un catalogo dei servizi solido
- Metriche di gestione finanziaria (penali, incentivi, costi ed esempi) applicate ai contratti del cliente e altre metriche contrattuali

La sezione Modellazione dei dati illustra:

- Eventi applicati a CA Business Service Insight e il relativo flusso nel sistema CA Business Service Insight
- Metriche e relativa modalità di registrazione
- Risorse CA Business Service Insight e modalità di identificazione
- Suggerimenti su come automatizzare la raccolta e la definizione di queste risorse

Tutti i punti precedenti vengono illustrati in modo approfondito nelle sezioni seguenti.

È importante tenere presente che le scelte inadeguate effettuate in questa fase possono influire negativamente sul funzionamento di CA Business Service Insight e possono essere difficili o impossibili da invertire in un secondo momento.

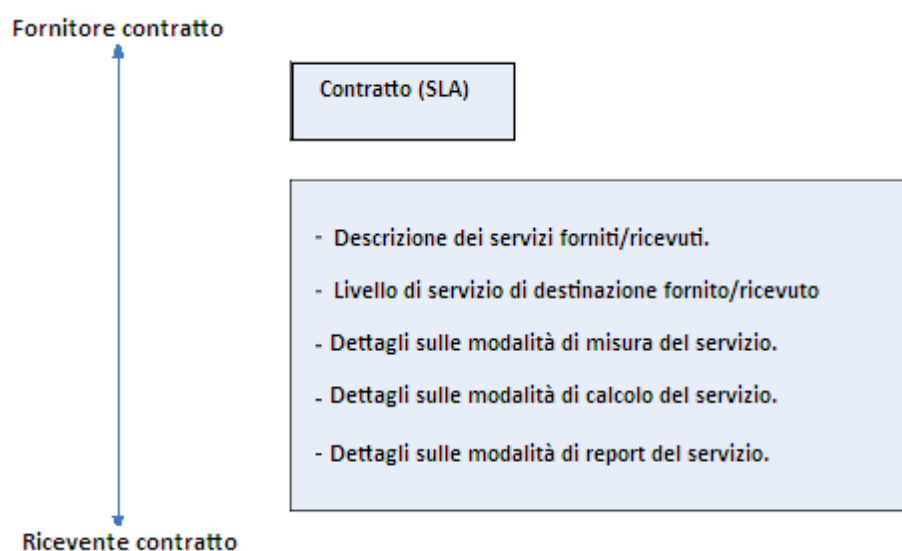
Modellazione dei contratti (responsabile contratti)

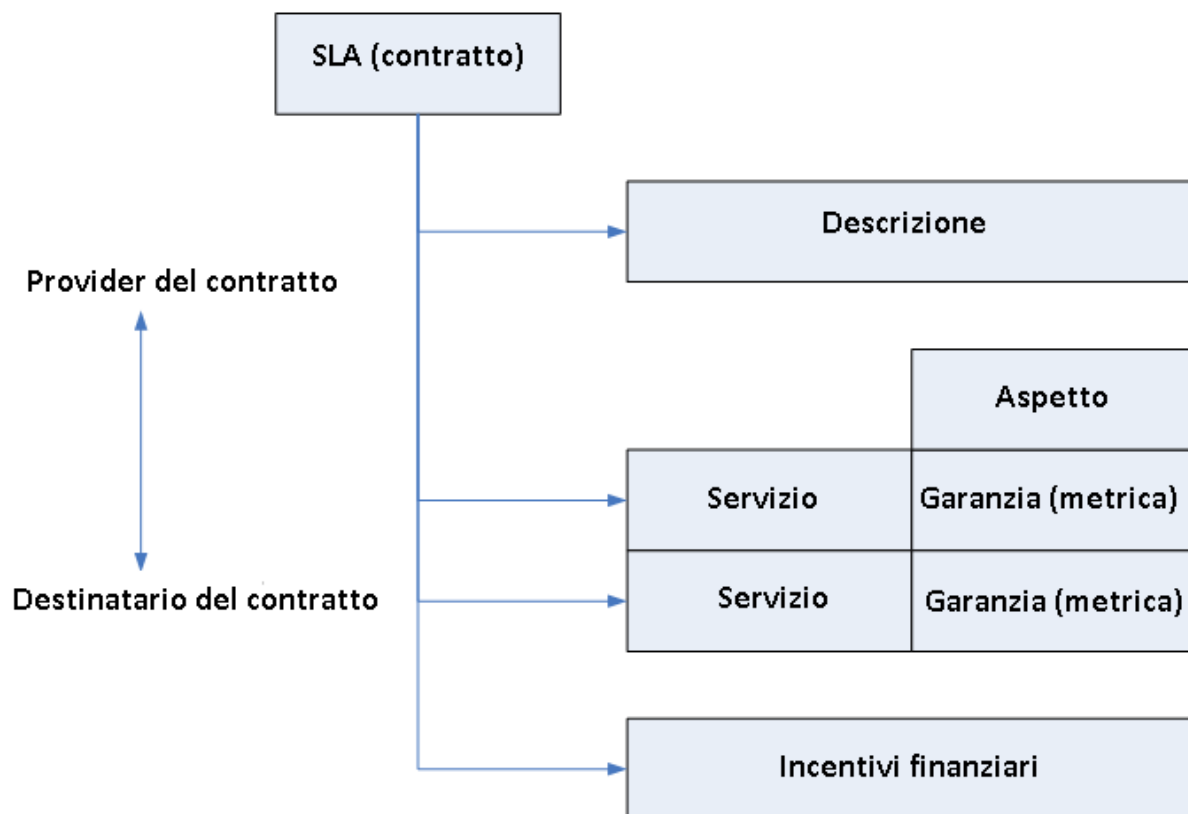
Le attività seguenti per la modellazione dei contratti vengono eseguite dal responsabile contratti.

Terminologia di contratto

Un contratto è una raccolta di obiettivi. L'obiettivo è un obiettivo operativo o contrattuale (metrica) o un obiettivo finanziario (incentivo). Il processo di modellazione dei contratti prevede l'acquisizione del contenuto del contratto intero nell'ambito dell'implementazione e la conversione nel modello CA Business Service Insight.

Il seguente diagramma schematizza il processo.

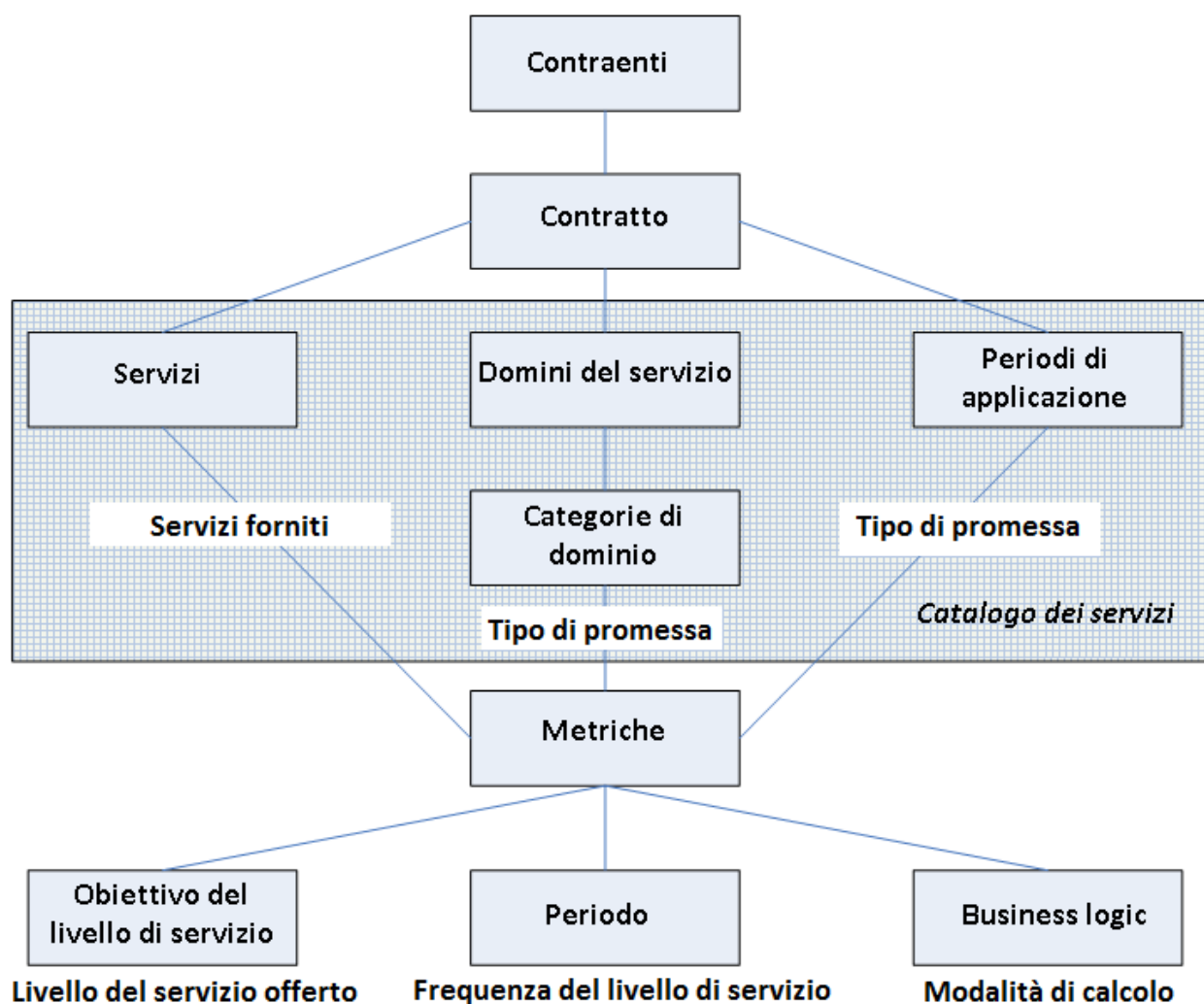




Nota: è necessario tener conto dei possibili piani futuri per il sistema anche se verranno modellati solo gli aspetti che rientrano nell'ambito di implementazione. L'ambito deve includere i requisiti generali di gestione dei contratti dell'organizzazione per progettare un modello sufficientemente ampio che comprende sviluppi futuri. In tal modo, vengono ridotte al minimo e semplificate eventuali modifiche che devono essere eseguite in futuro.

La conversione dei contratti del cliente nel modello CA Business Service Insight è un processo in cui il responsabile contratti raggruppa le garanzie (metriche) offerte all'interno dei limiti comuni in gruppi logici, componenti di servizio comuni, aspetti del contratto e così via. Il raggruppamento logico consente una gestione del livello di servizio molto flessibile.

Il modello di contratto CA Business Service Insight comprende le entità elencate nella figura seguente:



Contratto

Definisce il contratto e la raccolta delle metriche. Un contratto può essere di tre tipi diversi a seconda della relazione tra i contraenti interessati. Può essere SLA (accordo sui livelli di servizio tra l'organizzazione e i suoi clienti), OLA (accordo sui livelli operativi tra i reparti nell'organizzazione) o UC (contratto di fornitura tra l'organizzazione e i suoi fornitori, dove generalmente il contratto UC riguarda un servizio fornito dall'organizzazione a un altro cliente tramite un accordo SLA).

Contraenti

Definisce il cliente dei servizi forniti e la parte con cui viene firmato il contratto. Un singolo contraente entità può essere associato più contratti. Nota: all'interno del contratto è possibile definire il provider e il destinatario dei servizi relativi di tale contratto.

Metriche

Definisce un singolo obiettivo del livello di servizio o garanzia. Ciascuna metrica è una richiesta di misurazione che restituisce il risultato del livello di servizio effettivo su cui viene elaborato il report.

Una metrica comprende i seguenti elementi:

Tipo

Una metrica può presentare uno dei seguenti otto tipi:

SLO	Obiettivo del livello di servizio.
Informativa	Obiettivo del livello di servizio senza una destinazione.
KPI	Indicatore chiave di prestazioni, utilizzato per supportare diversi concetti industriali. In telecomunicazioni KPI=SLO indica l'obbligo del cliente, mentre in IT indica un obbligo tecnico.
KQI	Indicatore chiave di qualità, metrica generale per misurare ai risultati aggregati da diversi indicatori.
Intermedia	Misurazione intermedia da utilizzare in altre metriche. Questi risultati di metrica non possono essere riferiti in report.
Consumo	Utilizzato dalle metriche finanziarie per calcolare il consumo di risorse/servizio. Utilizzato in combinazione con le metriche dell'elemento di costo
Incentivo	Metrica finanziaria, utilizzata per i calcoli degli incentivi (termine positivo per penale).
Elemento di costo	Metrica finanziaria, utilizzata per calcolare i risultati basati sui consumi. Prezzi diversi per periodo o per numero di unità.

Periodo di riferimento

Periodo contrattuale di creazione report o periodo per il quale il risultato del livello di servizio effettivo calcolato viene confrontato con la destinazione. Il periodo di riferimento in CA Business Service Insight può essere definito con una granularità di ora, giorno, settimana, mese, trimestre o anno. Per lo scopo di analisi della causa principale, oltre al periodo di riferimento, la metrica viene calcolata in altri sei granularità, ma i risultati per questi periodi sono contrassegnati solo come risultati operativi.

Nota: questa operazione viene eseguita solo se vengono specificate queste granularità di calcolo per la metrica.

Periodo di applicazione

Periodo entro il periodo di riferimento durante il quale è applicabile la garanzia o l'obbligo specifico, incluse le definizioni del periodo di applicazione eccezionali, quali le finestre previste per la manutenzione, le festività nazionali, e così via.

Business logic

Formula che definisce la modalità di valutazione dei dati non elaborati per calcolare il valore del livello di servizio effettivo per tale aspetto del servizio e i presupposti specifici da prendere in considerazione per il calcolo. Durante la fase di progettazione, è identificato solo lo schema del calcolo. Viene definito in modo più dettagliato durante la fase di configurazione.

Destinazione

Obbligo contrattuale sul livello di servizio. La destinazione potrebbe essere obbligatoria oppure no, a seconda del tipo di metrica in questione. Nel caso in cui una destinazione non è definita, la metrica fornisce risultati solo per la creazione di report, senza la possibilità di confrontarli con un obbligo contrattuale o garanzia. Le destinazioni possono essere statiche o dinamiche.

Nota: per maggiori informazioni su alcuni concetti presentati finora, consultare la sezione [Appendice B - Esempi di casi di studio](#) (a pagina 215).

Ciascuna metrica è associata alle entità di sistema seguenti contenuta nella cartella dei modelli di sistema:

Componente del servizio

Descrive quale è obbligatorio fornire.

Gruppo dei componenti di servizio

Raccolta dei componenti del servizio. Un singolo componente di servizio potrebbe far parte di più gruppi. Un gruppo dei componenti di servizio è un'entità facoltativa; se utilizzato, è in grado di fornire una maggiore flessibilità per i report.

Dominio del servizio

Aspetto specifico dei componenti di servizio che deve essere misurato ai fini del monitoraggio del livello di servizio (ad esempio, le prestazioni o la disponibilità).

Categoria di dominio

Unità di misura specifica. Definisce l'unità di misura predefinita e se l'obiettivo di destinazione è un valore minimo o massimo. Sostanzialmente, è la dimensione specifica del dominio del servizio che viene misurato (ossia, percentuale del tempo disponibile, numero di interruzioni, velocità effettiva media, e così via).

Ognuna delle entità di cui sopra, nonché le rispettive relazioni a tutti gli obiettivi del livello di servizio da monitorare in CA Business Service Insight, deve essere identificata durante la fase di modellazione dei contratti.

Funzionamento del processo di modellazione

Durante il processo di modellazione, è necessario definire tutte le metriche da configurare nel sistema con le relative entità correlate, in base al contratto in considerazione e ai requisiti di creazione del report.

Prima o durante questa fase è buona prassi decidere uno standard di denominazione da utilizzare per i nomi della metrica in modo da rendere il sistema chiaro e ordinato e facile da utilizzare. Considerare inoltre la descrizione della metrica che può essere utilizzata nella sezione Dichiarazione dell'obiettivo della metrica.

Il processo di analisi del contratto include i seguenti passaggi:

1. Identificazione degli obiettivi contrattuali.

Per ogni obiettivo, identificare:

- Un nome appropriato (tenere in considerazione gli standard di denominazione)
- Destinazione (facoltativo)
- Periodo di riferimento
- Unità di misura
- Componente del servizio
- Dominio del servizio
- Categoria di dominio (e unità di misura)
- Periodo di applicazione (quando: 24 x 7, solo orari di ufficio?)
- Schema di calcolo (quali variabili sono necessarie e come viene calcolato il livello di servizio)

Nota: alcuni elementi potrebbero non essere chiari dal primo passaggio, ma possono essere ordinati in seguito durante il perfezionamento del catalogo di servizio.

2. Identificazione, dal contratto, di eventuali obiettivi finanziari e determinazione di ciascun obiettivo finanziario:
 - È un obiettivo di penale, incentivo o costo?
 - Quali sono le condizioni che lo attivano?
 - Per quali componenti del servizio si verifica?
 - Dominio del servizio
 - Categoria di dominio (l'unità in questo caso potrebbe essere una valuta, un importo di costo o una percentuale di pagamento, e così via)

Una volta annotati e definiti tutti gli obiettivi, è consigliabile rivedere l'elenco completo delle metriche e provare a ottimizzare/normalizzare il catalogo.

Durante questo passaggio è importante assicurarsi che tutti i componenti del servizio, i domini del servizio e le categorie di dominio siano definiti in modo adeguato e che possano formare un catalogo chiaro e conciso di ciò che è disponibile nel sistema. Sono inclusi TUTTI i contratti e le metriche nel sistema. In questo modo viene garantita la creazione di un catalogo dei servizi molto solido all'interno del sistema per consentire espansioni successive del sistema.

I domini del servizio e le categorie di dominio NON devono essere definiti fino a un livello granulare molto basso. Devono essere descrittivi, ma a un livello superiore rispetto alla metrica.

Ad esempio, se si dispone delle tre metriche seguenti:

- % of outage reports delivered on time
- % of exception reports delivered on time
- % of management reports delivered on time

La categoria migliore in cui è probabile che rientrino tutte e tre le metriche è il dominio del servizio delle prestazioni e la categoria di dominio della percentuale di report consegnati in tempo. La categoria di dominio non deve specificare il tipo di report. Queste tre metriche probabilmente hanno lo stesso metodo di calcolo e utilizzare la stessa business logic (ad eccezione, forse, di un singolo parametro per distinguere i diversi tipi di report).

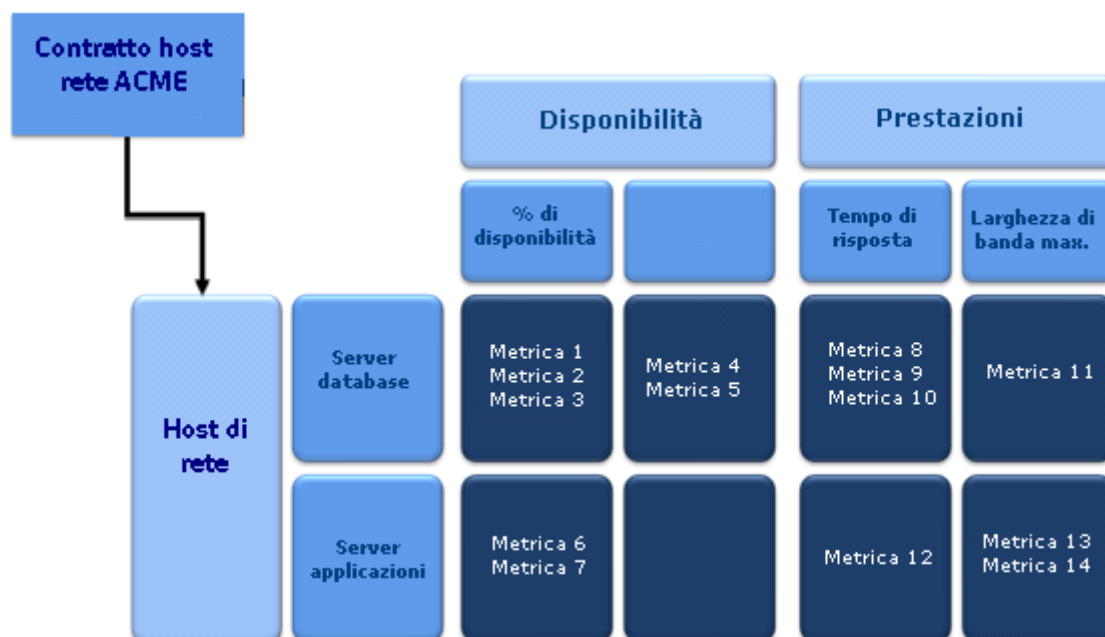
I domini del servizio e le categorie di dominio appropriati possono essere presi dallo standard ITIL (Information Technology Infrastructure Library). Questi sono solo esempi suggeriti, tuttavia, e ogni organizzazione specifica potrebbe avere i propri standard definiti per tale scopo. Consultare la sezione [Appendice A - Esempi di domini del servizio e di categorie di dominio](#) (a pagina 213) per alcuni domini del servizio e categorie di servizio suggeriti.

Nota: potrebbe essere utile a questo punto avere una riunione con le parti interessate principali, in modo da ottenere l'impegno da ciascuna che il catalogo selezionato supporta le proprie esigenze attuali e future.

Ulteriori esempi che illustrano punti importanti vengono presentati nelle appendici. In questi esempi, un singolo obiettivo di contratto viene descritto con la relativa modellazione. Durante la modellazione di situazioni effettive, è necessario conoscere tutti gli obiettivi in modo che le entità del catalogo rappresentino tutti gli obiettivi in modo più ampio ed esaustivo.

Una volta completato il processo di identificazione di tutte le metriche e delle entità correlate, il responsabile contratti ha una matrice in cui sono indicate le metriche come illustrato nel diagramma seguente.





Nelle sezioni successive vengono descritti altri problemi da considerare nel processo di modellazione.

Domande per il responsabile contratti

Di seguito sono elencate le domande che il responsabile contratti deve considerare in modo da garantire che tutti gli aspetti siano tenuti in conto:

Come si può sapere se è stato selezionato il componente del servizio corretto?

Se il componente del servizio può essere applicato a più contratti e misurato in più aspetti, è stato definito correttamente.

Ad esempio, il Sistema X è un sistema fornito diversi clienti e può essere misurato per disponibilità, affidabilità, prestazioni, e così via.

Come si può sapere se è stato selezionato il dominio del servizio corretto?

Se il dominio del servizio può essere misurato o calcolato in diversi modi, esso descrive un aspetto generale del servizio fornito ed è pertanto il dominio del servizio corretto.

Ad esempio, la disponibilità può essere misurata in vari modi, uno dei quali è la percentuale di tempo disponibile. Altri metodi possono essere la percentuale di disponibilità durante o fuori l'orario di ufficio, il numero di errori, il tempo medio tra due guasti (MTBF), il tempo medio di riparazione, il tempo massimo di inattività, il tempo medio di inattività, il tempo totale di inattività, e così via. Tutti questi sono modi diversi con cui valutare la disponibilità di un determinato sistema.

Casi da considerare durante il processo di modellazione

Di seguito sono riportati alcuni esempi di casi comuni e più particolari che descrivono i concetti da tenere presente durante il processo di modellazione. Tali concetti possono comportare una definizione più precisa delle metriche e un framework stabile.

Metriche senza destinazione

Poiché il campo destinazione all'interno della definizione di metrica è facoltativo, nei casi in cui non è definito, sono disponibili report sul livello di servizio per la metrica. Tuttavia, non è possibile alcun report su livello di servizio rispetto a destinazione e deviazione, poiché non esiste alcuna destinazione con cui confrontare il livello di servizio effettivo calcolato. Questi tipi di metriche vengono definiti nei casi in cui sono necessari report per le informazioni non incluse negli obblighi contrattuali effettivi.

La definizione di questo tipo di metrica offre all'utente tutte le caratteristiche funzionali possibili di drill-down per i report, e offre al manager del livello di servizio la possibilità di applicare le misurazioni a una destinazione in qualsiasi fase in futuro.

Ad esempio:

La garanzia contrattuale prevede di fornire il 99% di disponibilità della rete e report sul numero dei tempi di inattività per ogni mese.

Vengono definite due metriche: una con una destinazione del 99% per la disponibilità, l'altra per il conteggio dei tempi di inattività ogni mese senza una destinazione. È possibile creare un report per entrambe le metriche, ma solo la prima presenta i calcoli della deviazione in virtù del relativo obbligo contrattuale.

Nota: un altro metodo possibile per risolvere questo tipo di situazione è utilizzare gli output di business logic e i report in formato libero su tali dati. Tuttavia, si perdono la caratteristica funzionale drill-down del report sui dati e anche la possibilità di utilizzare la procedura guidata di creazione report semplice. Il vantaggio dell'utilizzo degli output di business logic è, dall'altro lato, è il risparmio dell'energia del motore con un numero minore di metriche.

Per ulteriori informazioni su questo metodo, consultare il case study [Output - Tabelle utente](#) (a pagina 155).

Metriche con destinazioni

Nel caso in cui viene definita una destinazione per una metrica, ci sono due modi possibili di specificare la destinazione. Può essere specificata come destinazione statica o dinamica. Una destinazione statica è lo scenario più frequente, in cui la destinazione può essere un valore valido concordato per la durata del contratto.

Ad esempio:

La disponibilità della rete non deve essere inferiore al 98% ogni mese.

La destinazione in questo caso è 98%.

In alternativa, la destinazione può dipendere dalle prestazioni dei mesi precedenti, oppure cambiare solo il valore in tutto l'anno. Esistono molte altre situazioni che possono essere rilevate, ma in generale vengono tutte implementate mediante una formula. CA Business Service Insight supporta questa funzionalità richiamando una funzione aggiuntiva dal modello di business logic standard. La funzione di destinazione può accedere ad altri parametri dal contesto della business logic e può supportare qualsiasi possibile scenario richiesto.

Ad esempio, il tempo di risoluzione dei ticket nell'assistenza tecnica che dipende dal carico dell'assistenza tecnica: il tempo medio di risoluzione per i ticket di priorità alta è 1 giorno se non sono presenti più di 1000 ticket durante lo stesso mese. Se sono presenti più di 1000 ticket inviati all'assistenza tecnica in un mese, il tempo medio di risoluzione per i ticket di priorità alta è 2 giorni.

In questo caso, la metrica viene definita con una destinazione dinamica che viene valutata all'interno dello script di business logic in base al numero di ticket emessi in tale mese.

Nota: per informazioni sul metodo di implementazione della destinazione dinamica, consultare il case study Implementazione di destinazioni dinamiche.

Parametro della metrica

Un parametro della metrica è un valore che può essere trattato dalla business logic della metrica e un valore che può essere facilmente modificato dalla definizione di metrica senza dover modificare il codice effettivo. Viene utilizzato al posto del valore hardcoded e può essere facilmente modificato.

È importante identificare i parametri della metrica per individuare facilmente i moduli di business logic e creare contenuti riutilizzabili. Inoltre, i parametri della metrica sono accessibili tramite la Procedura guidata contratto che consente a un utente finale di eseguire modifiche facilmente.

Ad esempio:

- Gli incidenti di Severity 1 devono essere risolti entro 24 ore.

Nel precedente obbligo, la destinazione è un tasso di risoluzione di 24 ore e il livello di gravità (Severity1) può essere definito come un parametro.

- Il numero dei tempi di inattività durante un mese non deve superare le 3 volte, dove il tempo di inattività è la volta in cui il sistema non è stato disponibile per più di 5 minuti.

Nell'obbligo precedente, il tempo che viene considerato come tempo di inattività può essere definito come un parametro.

Parametro contrattuale

Un parametro contrattuale è un valore che può essere trattato da tutte le metriche all'interno di un contratto. Un parametro contrattuale può essere utilizzato all'interno della metrica utilizzando lo stesso metodo del parametro della metrica, ma definito invece come parametro dinamico.

È consigliabile utilizzare un parametro contrattuale quando è necessario l'utilizzo dello stesso valore per più di una metrica. Un altro vantaggio importante per utilizzare un parametro contrattuale è la semplificazione della manutenzione dei contratti. Poiché i parametri tendono a modificarsi spesso e richiedere l'aggiornamento nel sistema, è più facile accedere a una singola posizione nel contratto e modificare i valori dei parametri simultaneamente invece che accedere a ciascuna metrica nel contratto e modificare i valori dei parametri a livello di metrica.

Di conseguenza, la modellazione più consigliata è definire i parametri a livello di contratto come parametri contrattuali e accedere ai relativi valori tramite parametri dinamici a livello di metrica.

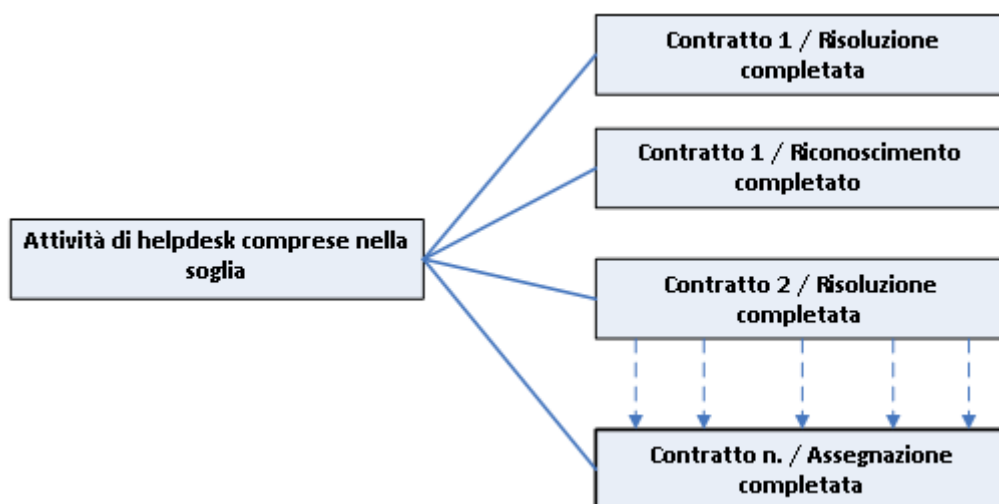
Ad esempio, consultare il case study [Prestazioni dell'assistenza tecnica](#) (a pagina 223).

Modelli e moduli di business logic

I modelli di business logic sono un modo semplice per l'archiviazione di un metodo di calcolo per una metrica. Componente di business logic completo, rappresentano un modo semplice per la creazione di un riferimento per altri componenti di business logic. I nuovi componenti di business logic creati da un modello copiano il codice e creano una nuova istanza. Tuttavia, in generale, la flessibilità nell'utilizzo dei modelli è alquanto ridotta e i moduli di business logic devono essere utilizzati appena possibile.

I moduli di business logic sono componenti di codice indipendenti che consentono il riutilizzo della stessa base di codice da altre business logic. I moduli possono includere anche altri moduli, pertanto possono esservi più livelli gerarchici. Quando si utilizzano i moduli, il codice è contenuto in un unico posto e viene riutilizzato da tutti gli altri componenti collegati. Il riutilizzo delle sezioni di codice facilita la manutenzione eliminando la duplicazione di codice e rendendo possibile l'applicazione di modifiche della logica a livello di sistema in modo rapido e semplice.

Durante la fase di progettazione, è necessario identificare i moduli di business logic principali e i relativi parametri. Quando la modellazione dei contratti è stata completata e il responsabile contratti ha una visione chiara della logica da utilizzare, è possibile identificare i calcoli che sono in comune e possono essere definiti in singoli moduli.



Il diagramma sopra riportato descrive un modulo che calcola la percentuale di operazioni riuscite dell'assistenza tecnica per soddisfare una destinazione entro soglie specificate. Per implementarlo come descritto, è necessario definire due parametri, noti come parametri della metrica: uno che definisce il tipo di attività dell'assistenza tecnica, l'altro per la soglia da confrontare (consultare la definizione *Parametro della metrica* nella sezione [Casi di studio da considerare durante il processo di modellazione](#) (a pagina 32)).

Con un'attenta considerazione dei tipi di calcoli implementati nel sistema, probabilmente si noterà che è possibile eseguire alcuni tipi simili tramite la modifica di una piccola parte di codice e utilizzando un parametro in modo che agisca da switch tra loro. In questo modo, è possibile ridurre al minimo la quantità di codice da creare e ottimizzare la quantità di codice da riutilizzare.

Modelli del livello di servizio

Un modello del livello di servizio è un insieme di componenti di servizio e delle metriche associate definite per misurarli. Questi modelli del livello di servizio possono essere creati come richiesto e spesso vengono definiti in base agli aspetti del servizio misurati con maggiore frequenza.

Il problema principale nella definizione dei modelli del livello di servizio è che tutti i parametri che possono essere utilizzati per modificare il comportamento delle metriche devono essere identificati e visibili. In questo modo si offre la massima flessibilità per il sistema e si semplifica la configurazione per gli utenti del sistema. Quando si utilizza il modello del livello di servizio per creare nuove metriche, ognuno dei parametri di configurazione è visibile nell'interfaccia della procedura guidata, pertanto non è richiesta alcuna ulteriore o minima personalizzazione per attivare il contratto. I parametri visibili all'utente tramite la procedura guidata sono inclusi nella dichiarazione dell'obiettivo. Pertanto, considerare nella dichiarazione dell'obiettivo della metrica tutti i parametri che si desiderano rendere disponibili per la modifica da parte dell'utente finale.

Per assicurare la massima efficienza con i modelli del livello di servizio, si deve puntare all'esecuzione di tutte le business logic attraverso la funzionalità del modulo, come descritto in precedenza.

Di seguito è riportato uno scenario di esempio per l'utilizzo dei modelli del livello di servizio, in cui i componenti del servizio di hosting dell'applicazione sono forniti ai clienti a tre livelli, minimo, medio e massimo, a seconda dell'importo che il cliente desidera pagare. Le metriche aggiuntive possono essere specificate in ogni livello superiore di hosting, insieme con le destinazioni più rigorose. Ciascuno di questi livelli potrebbe essere una buona base per la creazione di un modello del livello di servizio, come illustrato nello scenario di esempio seguente:

- Hosting applicazione - Minimo (5 metriche)
- Hosting applicazione - Medio (8 metriche)
- Hosting applicazione - Massimo (12 metriche)

Ora, quando un nuovo cliente viene aggiunto al sistema, è facile scegliere la definizione richiesta in base alle necessità del cliente. L'utilizzo dell'interfaccia della procedura guidata abilita la personalizzazione di ciascuna metrica compresa per tale cliente specifico. Nota: è anche possibile scegliere solo alcune metriche dalle definizioni, o anche aggiungere metriche da due definizioni differenti al contratto del nuovo cliente, se è richiesta una personalizzazione particolare.

Importanza di un catalogo dei servizi solido

Come descritto in precedenza, il catalogo dei servizi, è un componente principale del sistema e deve essere configurato in modo strutturato. Consente di utilizzare il sistema in modo efficiente a tutti gli utenti e di evitare confusione. Consente inoltre di espandere il sistema con l'organizzazione e di gestire futuri miglioramenti e aggiunte con un impatto minimo sugli elementi già creati.

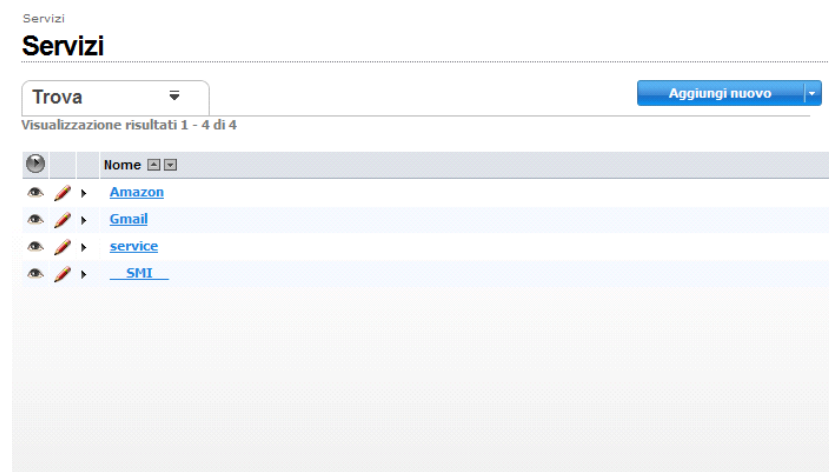
Il sistema offre un elevato grado di flessibilità nella creazione e gestione del catalogo di servizi componenti e degli accordi SLA. Tuttavia, poiché la qualità dipende dalla progettazione, la pianificazione per il futuro è essenziale alle prime fasi di progettazione.

La definizione del catalogo dei servizi CA Business Service Insight in un modo efficiente e strutturato rende il sistema flessibile per aggiungere servizi e domini al catalogo dei servizi in un secondo momento. Consente inoltre l'aggiunta di contratti, metriche, notifiche e report in seguito. Inoltre, consente un approccio strutturato alla business logic e apre la strada alla possibilità di utilizzare i moduli e i modelli standard per la gestione dei dati di business e dei calcoli associati.

Insieme al catalogo, i modelli del livello di servizio definiti nel sistema offrono un modo eccezionale al responsabile contratti di creare nuovi contratti molto facilmente e con poche o nessuna nozione dei livelli di base dei dati richiesti. Una volta impostato correttamente, dovrebbe essere possibile configurare un nuovo contratto per un cliente solo tramite la modifica dei parametri nel modello del livello di servizio. Tuttavia, tutto questo si basa sul catalogo e sulle definizioni impostate nel modo più efficiente. Questi parametri sono tutti visibili tramite la dichiarazione dell'obiettivo per ciascuna metrica nel modello del livello di servizio e possono essere modificati o nel modello o dalla procedura guidata quando si utilizza la definizione.

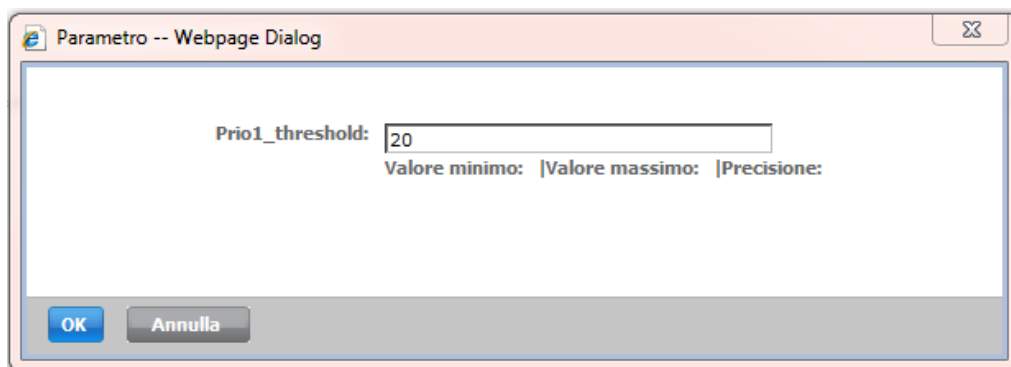
Esempio:

Quando si utilizza un modello del livello di servizio per distribuire metriche di supporto al cliente a un contratto, è possibile selezionare le metriche richieste da una definizione esistente.



All'interno di questo modello del livello di servizio è presente una metrica denominata % *of Incidents responded to on time*. È possibile notare la presenza di un livello soggettivo, poiché l'aspetto "on time" (in tempo) potrebbe essere soggetto a domande. Il seguente esempio illustra la misurazione effettuata in questa metrica:

La dichiarazione dell'obiettivo visualizzata nella parte inferiore della scheda Generale della metrica (oppure nella scheda Dichiarazione dell'obiettivo) mostra tutti i parametri esposti in questa metrica. Nella figura precedente, la definizione specificata di "*on time*" (in tempo) è 20 minuti. Questo parametro è personalizzabile per consentire una nostra reinterpretazione di questa metrica predefinita. Per modificare questo valore, è possibile fare clic sul collegamento del parametro 20 minuti.



In questo modo, la nuova metrica creata dal modello del livello di servizio può essere personalizzata senza la necessità di modificare la business logic di base della metrica. Nota: questo presuppone inoltre la business logic è scritta in modo da includere questi parametri nel calcolo del livello di servizio.

Questo semplice esempio mostra chiaramente l'importanza della creazione di un insieme di modelli del livello di servizio solido e flessibile per il catalogo di sistema in modo da consentire il riutilizzo futuro dei contratti da tali modelli.

Gestione finanziaria (penali, incentivi e costi)

Nelle versioni precedenti di CA Business Service Insight, erano presenti entità di contratto note come Penali che sono state implementate utilizzando le formule analoghe a quelle di Excel. I risultati delle penali erano basati esclusivamente sull'input dalle metriche di contratto e legati alle funzioni di base per calcolare l'importo di penale risultante. Dalla versione 4.0 in poi, questi sono stati sostituiti dalle metriche finanziarie che possono essere create dall'utente, con una maggiore flessibilità. Queste metriche finanziarie possono essere utilizzate per fornire informazioni Incentivo e Costo del contratto.

Nota: un incentivo sostituisce il vecchio termine di penale della versione 3.0 e precedenti di CA Business Service Insight, e può essere positivo o negativo a seconda delle prestazioni. Un incentivo negativo, tuttavia, è praticamente una penale. È importante anche tenere presente che se viene implementata una penale con un tipo di metrica Incentivo, *occorre* far restituire alla funzione Result() un valore negativo. In questo modo sono consentite eventuali funzioni di riepilogo che possono combinare insieme i risultati di metriche diverse, così da adeguare i totali nella giusta direzione. In altre parole, un incentivo aumenta il valore mentre una penale lo diminuisce.

CA Business Service Insight versione 4.0 consente inoltre di creare una metrica di consumo per misurare l'utilizzo dei componenti del servizio e delle risorse, e di combinarla con una metrica di elemento di costo per stabilire il costo tale servizio o risorsa. La combinazione di queste con la funzionalità avanzata di previsione consente di creare metriche di gestione finanziaria molto esaustive.

Le metriche finanziarie sono anche in grado di acquisire l'output da altre metriche di contratto e di determinare i valori correlati di penale o incentivo in base alle prestazioni di queste metriche contrattuali. È inoltre possibile utilizzare altri tipi di informazioni per determinare il loro risultato, ad esempio le informazioni sul prezzo unitario e i modelli di previsione che abilitano la funzionalità di report, tra cui costi previsti rispetto a costi effettivi.

Esempio di elemento di costo:

Una particolare applicazione di rischio presenta un costo associato in base al numero di utenti simultanei del sistema. Viene calcolato mensilmente ed esiste un valore previsto fornito per questa applicazione. Il prezzo unitario di questa applicazione (costo per utente simultaneo) è riportato nella tabella seguente (presupporre che l'applicazione rientri nell'indice 1):

Index	Oct-01-2006	Nov-01-2006	Dec-01-2006	Jan-01-2007	Feb-01-2007	Mar-01-2007	Apr-01-2007	May-01-2007	Jun-01-2007
1	1.0	1.0	1.0	0.75	0.75	0.75	0.6	0.6	0.6
2	0.9	0.9	0.9	0.7	0.7	0.7	0.5	0.5	0.5

È inoltre disponibile il numero di utenti simultanei previsti per il periodo (nuovamente, indice 1).

Index	Oct-01-2006	Nov-01-2006	Dec-01-2006	Jan-01-2007	Feb-01-2007	Mar-01-2007	Apr-01-2007	May-01-2007	Jun-01-2007
1	3000	5000	8000	5000	5000	5000	5000	5000	5000
2	2500	2500	6000	4000	4000	4000	4000	4000	4000
3	1000	2000	3000	1000	1000	1000	1000	1000	1000

Con la modellazione di questa metrica di costo utilizzando le tabelle dei costi, è possibile determinare il costo dell'applicazione mensile, in base al numero effettivo di utenti simultanei. Questa informazione proviene dall'origine dati e può essere moltiplicata per le cifre di costo unitario precedenti per dare le cifre di costo. È inoltre possibile confrontarla con i valori previsti per fornire un'analisi del costo previsto rispetto al costo effettivo. La business logic in questo caso è necessaria per determinare il numero effettivo di utenti simultanei durante il periodo e moltiplicarlo utilizzando i valori di costo unitario. Inoltre, la funzione di previsione nella business logic fa riferimento alle informazioni di previsione. Questo è un esempio di applicazione di una metrica di elemento di costo.

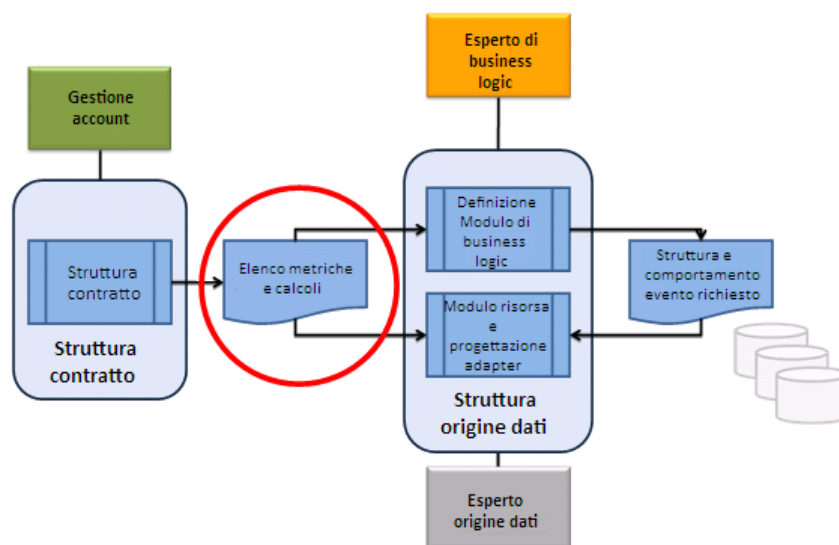
Esempio di scenario di penale:

L'accordo SLA del cliente presenta una clausola di inadempimento per garantire che la rete sia disponibile il 98% del tempo per un determinato mese durante le ore lavorative. Qualsiasi livello di servizio mensile al di sotto di questo valore incorre nel pagamento di una penale basato su una formula ($\text{Penalty} = \$1000 \text{ per ogni percentuale intera inferiore alla destinazione (ad esempio, } 96.5\% = (98 - \text{Round}(96.5)) * 1000 = (98 - 97) * 1000 = - \$1000$)).

Per implementare la condizione della penale, è possibile creare una metrica di incentivo finanziario che accetta l'input da una metrica esistente (*Network Availability* $\geq 98\%$). Il processo di registrazione per questa metrica utilizza il processo RegisterByMetric() per ricevere i valori del livello di servizio da tale metrica per eseguire i confronti. I valori del livello di servizio vengono inviati per il periodo di riferimento di quella metrica a questa metrica finanziaria come eventi che vengono quindi utilizzati come parte del calcolo per determinare l'importo della penale per lo stesso periodo utilizzando la formula dallo scenario.

Nota: per ulteriori casi di studio, consultare la sezione [Case study della modellazione di metriche finanziarie](#) (a pagina 226).

Output della fase di modellazione dei contratti (responsabile contratti, esperto delle sorgenti dati)



Come mostrato nel precedente diagramma, per passare alla fase di modellazione dei dati della soluzione, al responsabile contratti viene richiesto alla fine della fase di modellazione dei contratti di fornire l'elenco delle metriche e i relativi calcoli necessari. L'elenco include le seguenti informazioni:

- Definizione di un catalogo dei servizi completo, che comprende:
 - Elenco completo dei componenti del servizio da offrire
 - Suddivisione dei domini del servizio e delle categorie di dominio
 - Definizione di tutti i periodi di applicazione necessari per ciascuna metrica
 - Elenco dei moduli di business logic per gestire ogni tipo di calcolo (inclusi i parametri necessari per l'implementazione)
- Elenco completo delle metriche da implementare, tra cui:
 - Standard di denominazione ben definito per le metriche
 - Categorizzazione di ciascuna metrica in base ai componenti del catalogo di servizi predefinito

Questo documento è uno strumento utile per assicurare che tutte le definizioni principali per un contratto sono state eseguite e che tutte le metriche siano state definite completamente.

Le informazioni in questo foglio di calcolo sono l'input che l'esperto delle sorgenti dati necessita per continuare con la prossima fase di modellazione dei dati.

Una volta completati questi elementi, è possibile passare alla fase successiva, in cui è possibile iniziare la modellazione con i dati effettivi dalle origini. Senza un modello di contratto completato, è molto difficile sapere esattamente che cosa viene richiesto dall'origine dati per soddisfare gli obblighi contrattuali.

Modellazione dei dati (Esperto delle sorgenti dati, Esperto di business logic)

La sezione sulla modellazione dei dati descrive la seconda parte del processo di progettazione. La fase di modellazione dei dati è il processo di acquisizione dei dati dalle origini del cliente, identificazione dei componenti specifici dei dati richiesti, decisione sulle modalità necessarie di normalizzazione dei dati e valutazione della procedura di assegnazione dei dati pertinenti alle metriche corrispondenti (tramite il processo di registrazione).

Questa fase include le seguenti attività:

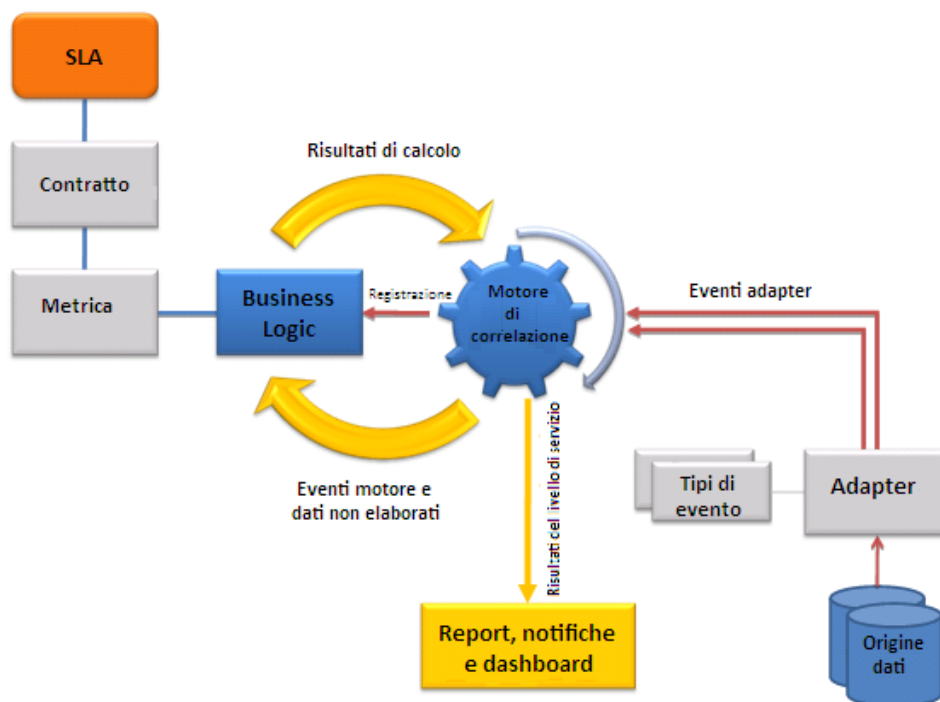
- **Esperto di business logic:**
 - Identificazione e definizione della struttura degli eventi di input richiesti per il calcolo da definire successivamente come Tipi di evento
 - Creazione dei campi richiesti in Tipo di evento per fornire i dati necessari per tutti i calcoli in cui è utilizzato
 - Definizione del processo di registrazione delle metriche per massimizzare il flusso degli eventi
 - Determinazione del modo per creare il modello di risorsa dai dati disponibili per soddisfare tutti i requisiti di report e logica
- **Esperto delle sorgenti dati:**
 - Identificazione del tipo di origine dati e determinazione delle modalità di normalizzazione tramite l'adapter nei tipi di evento definiti e immessi nel database CA Business Service Insight
 - Determinazione degli identificatori di tipo di evento da associare ai dati

Eventi e relativo flusso

Il flusso di dati nel sistema è sotto forma di eventi. Quando si verifica un evento, viene creato dall'adapter un messaggio informativo in base all'origine dati, in un formato utilizzabile da CA Business Service Insight per i relativi calcoli del livello di servizio. I dati non elaborati sono sempre formati da eventi.

L'elemento centrale della progettazione deve quindi essere sul flusso di eventi all'interno del sistema.

Prima di modellare i requisiti dei dati, l'esperto di business logic e l'esperto delle sorgenti dati devono avere una profonda conoscenza degli eventi e del relativo flusso all'interno del sistema CA Business Service Insight. Nel seguente diagramma viene illustrato ad alto livello questo flusso di eventi base.



Il precedente diagramma descrive il modo in cui gli eventi vengono recuperati dall'origine dati con gli adapter e normalizzati in una struttura di eventi standard definita come tipo di evento. Questi eventi vengono inviati dagli adapter a CA Business Service Insight. Questi eventi sono definiti Eventi di dati non elaborati.

I calcoli di business logic in ciascuna metrica sono basati su un sottoinsieme di eventi di dati non elaborati. La business logic di conseguenza richiede questo sottoinsieme per l'esecuzione della *registrazione*.

In base all'istruzione di registrazione, il motore di correlazione invia solo gli eventi di dati non elaborati pertinenti ai calcoli di business logic.

Altri tipi di eventi inviati alla business logic sono gli eventi del motore. Tutti i concetti coinvolti in questo processo vengono trattati in dettaglio in questo capitolo.

Questa sezione è incentrata sulle seguenti parti del diagramma:

- Origine dati
- Adapter
- Tipi di evento
- Registrazione della metrica

Il modello di dati CA Business Service Insight è stato progettato per ottimizzare l'efficienza di questo flusso di dati nel sistema.

In generale, CA Business Service Insight funziona su due livelli: il livello dell'infrastruttura e il livello del modello di business. Semplificando la suddivisione, il livello dell'infrastruttura include gli adapter, le risorse e gli oggetti del tipo di evento, mentre il livello di business include i contratti, le metriche e gli oggetti di servizio. Tra i due livelli esiste un livello shim virtuale, denominato livello di correlazione.

Un identificatore di eventi è l'oggetto Tipo di evento. Il tipo di evento determina il modo in cui gli eventi vengono definiti e segnalati a CA Business Service Insight. Definisce anche la struttura del campo dati Evento in modo che possa essere interpretato dalla business logic durante l'elaborazione.

Un altro identificatore di eventi è Risorsa, che è l'entità più piccola utilizzata nel calcolo. Ad esempio, durante il calcolo della disponibilità del server, la definizione logica dell'entità più piccola su cui è richiesta la creazione di un report potrebbe essere un server specifico oppure un cliente durante la creazione di un report sulla gestione del ticket del cliente. La risorsa è una definizione di un'entità CA Business Service Insight derivata dall'origine dati e dai requisiti di calcolo. Per ogni risorsa viene specificato il tipo di risorsa, vale a dire l'identificatore di risorsa, che determina esattamente quale tipo di risorsa è definito. Ogni risorsa deve avere un tipo di risorsa associato, che consente inoltre l'aggiunta di attributi personalizzati da associare a ogni risorsa. Per ulteriori informazioni su questi attributi, consultare la sezione [Risorse e relativa gestione](#) (a pagina 51).

La correlazione si verifica tra gli eventi dell'adapter in ingresso e le metriche di contratto. Il centro di questo processo di correlazione è l'allocazione di risorse e la registrazione di metriche.

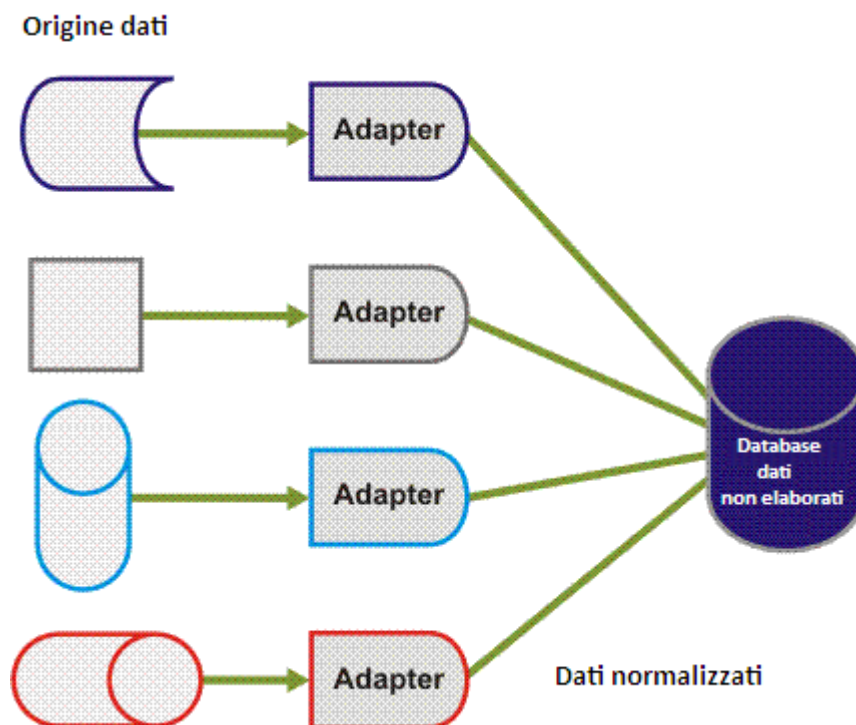
L'allocazione di risorse e la registrazione di metriche specificano quali flussi di eventi della risorsa vengono misurati e da quale metrica.

Nota: in questo caso, con la registrazione della metrica potrebbe verificarsi un grado di riutilizzo e co-dipendenza con altre metriche poiché è possibile utilizzare l'output di una metrica come input per un'altra. Allo stesso modo, esistono eventi intermedi che non vengono utilizzati come l'output di una metrica per la misurazione del livello di servizio, ma piuttosto come fase di calcolo intermedio che può essere utilizzato da altre metriche.

Modello di dati - Panoramica

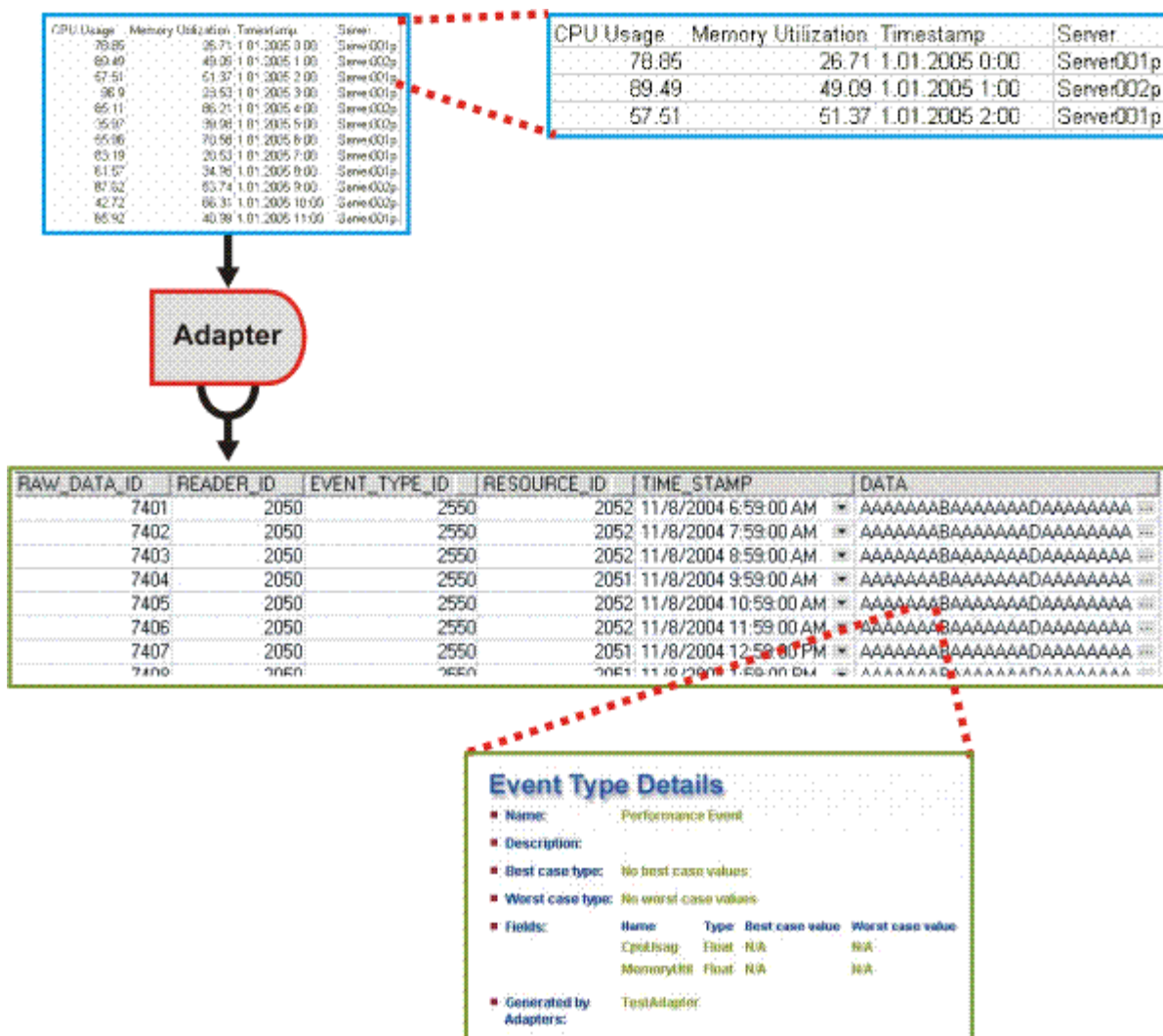
Il modello di dati CA Business Service Insight è stato progettato per rispondere e vincere la sfida seguente.

I dati non elaborati vengono recuperati dagli adapter da varie origini dati distinte e vengono gestiti in diversi formati. È necessario recuperare e rendere omogenei questi dati diversi in un'unica tabella di database. Di conseguenza, gli adapter devono leggere e normalizzare i dati in un modello dei dati unificati, come illustrato nella seguente figura.



Come parte del processo, tutti i campi di dati vengono inseriti nello stesso campo di tabella del database, ma crittografati. Ogni riga inserita nel database CA Business Service Insight presenta un identificatore Tipo di evento associato. La definizione del tipo di evento contiene le descrizioni dei campi di dati. Inoltre, consente al motore di correlazione di interpretare i campi di dati correttamente e di identificare quando sono richiesti dalla business logic per i calcoli.

Nella figura seguente viene illustrata una rappresentazione grafica della sezione di recupero dei dati e popolamento del database di questo processo. Inoltre, viene illustrata una sezione ingrandita per mostrare che cosa rappresentano i dati in termini reali, invece del modo di visualizzazione dei dati non elaborati.



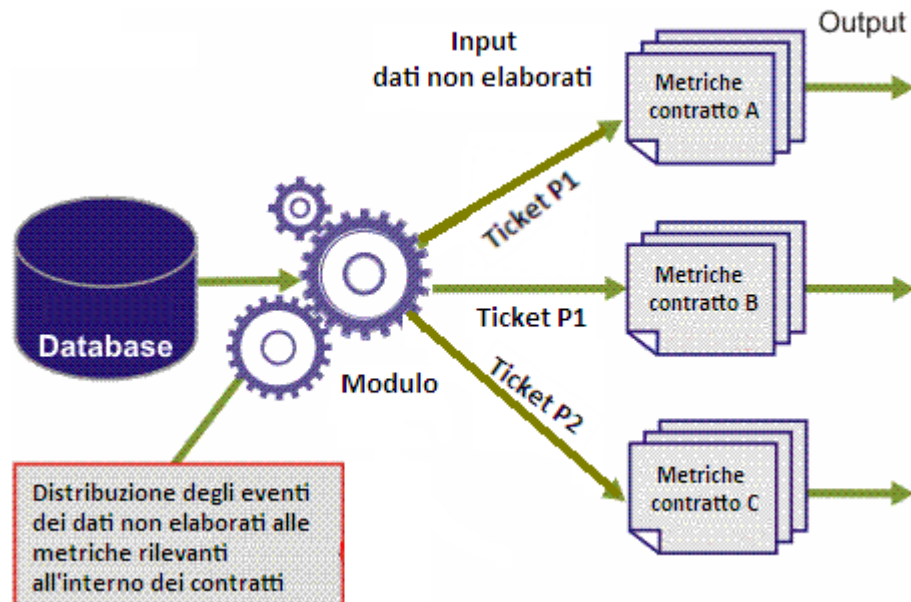
Il sistema CA Business Service Insight include anche tutti i contratti e le metriche che richiedono la valutazione rispetto ai dati non elaborati per produrre le informazioni sulle prestazioni del livello di servizio risultante. Ciascuna metrica deve ricevere solo il sottoinsieme dei dati pertinenti al calcolo. I dati non elaborati contengono un numero potenzialmente vasto di record di tipi diversi. L'utilizzo della metrica per filtrare gli eventi pertinenti in base ai rispettivi valori non è molto efficiente. Pertanto, il motore CA Business Service Insight distribuisce i dati non elaborati pertinenti a ogni metrica specifica.

Esempio:

Per le due metriche seguenti in un contratto:

- Priority 1 (P1) tickets average resolution time
- Priority 2 (P2) tickets average resolution time

Alla prima metrica è richiesto di valutare solo i ticket con priorità 1, mentre alla seconda metrica solo i ticket con priorità 2. Pertanto, al motore viene richiesto di distribuire i record di conseguenza. Inoltre, il tempo di risoluzione all'interno di un contratto è calcolato per i ticket P1 aperti per il contraente A, mentre nel secondo contratto i ticket P1 per il contraente B, e nel terzo i ticket P2 per il contraente C. Pertanto, al motore viene richiesto di selezionare il tipo di ticket e il cliente cui è stato segnalato, come illustrato nella seguente figura.



Come spiegato in precedenza, i record dei dati non elaborati presentano identificatori associati che consentono al motore di identificare i record e gli eventi pertinenti a ciascuna business logic della metrica. I due identificatori sono Tipo di evento e Risorsa.

Conversione e normalizzazione dell'adapter

La funzione dell'adapter è la lettura dei dati dall'origine dati e la relativa normalizzazione nel formato di un evento. Ogni evento in CA Business Service Insight è costituito dai seguenti campi:

- ID risorsa
- ID tipo di evento
- Timestamp (Data/ora)
- Campi Valore in base a Tipo di evento

È necessario che l'adapter colleghi campi originali dell'origine dati ai campi Risorsa corrispondenti in CA Business Service Insight. A tale scopo, viene utilizzata una tabella di conversione che contiene il valore trovato nell'origine dati e l'ID risorsa corrispondente in CA Business Service Insight.

Il processo di associazione dell'ID risorsa e dell'ID tipo di evento al valore dell'origine dati pertinente è definito *conversione dell'adapter*. Durante questo processo, viene creata la tabella di conversione con i valori corrispondenti. Questa tabella viene utilizzata dall'adapter per popolare l'ID tipo di evento e l'ID risorsa nel record di evento in corso di creazione. Vengono create tabelle indipendenti per la conversione di risorse e tipi di evento.

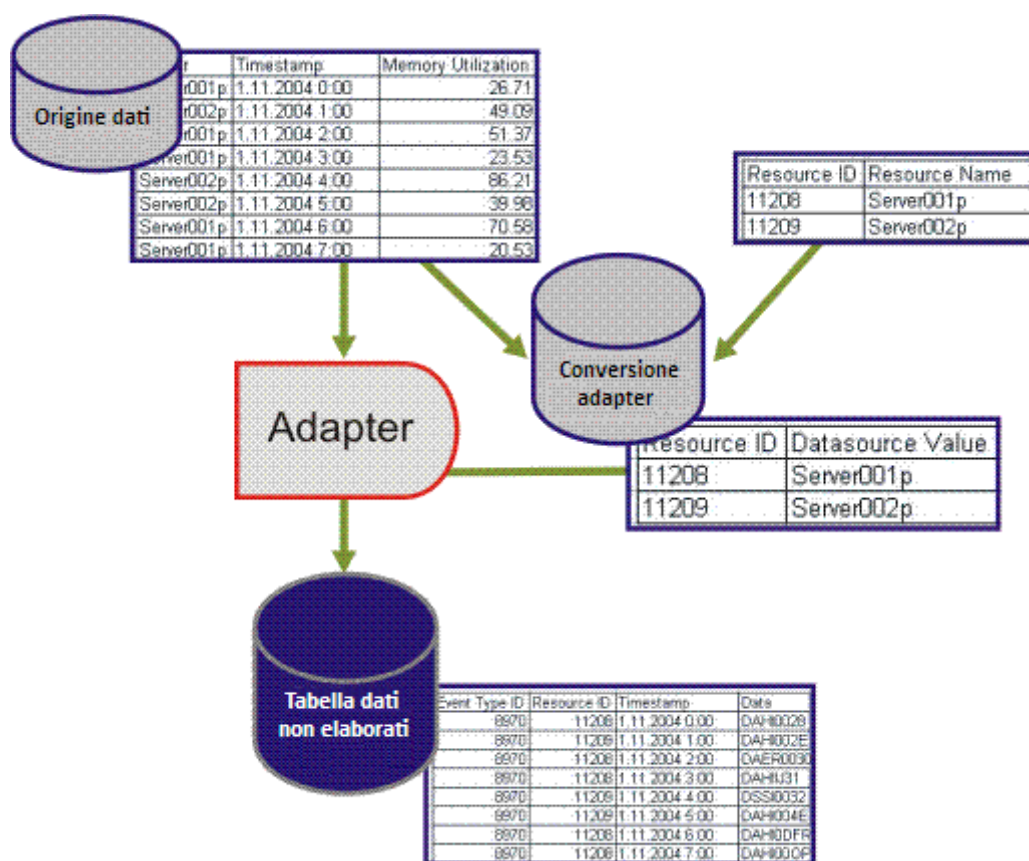
Come indicato in precedenza, l'ID risorsa e l'ID tipo di evento vengono utilizzati per scopi di registrazione, mentre il campo dati e i valori di data/ora vengono utilizzati per i calcoli effettivi.

Il campo Data/ora viene utilizzato anche dal motore per determinare l'ordine e l'intervallo degli eventi inviati per i calcoli.

La definizione del tipo di evento viene eseguita manualmente in CA Business Service Insight in base all'origine dati di input e all'output richiesto.

Nota: la definizione della risorsa può essere eseguita manualmente o automaticamente tramite gli script di conversione (per ulteriori informazioni, consultare la sezione [Risorse e relativa gestione](#) (a pagina 51)).

Nella figura seguente viene illustrata l'interazione tra l'origine dati, la tabella di conversione dell'adapter, l'adapter e la tabella dei dati non elaborati CA Business Service Insight.



Registrazione della metrica

Affinché al motore di correlazione siano noti quali dati potrebbero essere richiesti, la metrica deve registrare la sua presenza e requisiti con il motore di correlazione.

La registrazione della metrica è la richiesta da parte di una metrica di ricevere eventi, e solo quelli che devono essere inclusi nel calcolo. La richiesta viene eseguita per lo stato di Tipo di evento e Risorsa dell'identificatore di evento.

La registrazione può essere eseguita sulla base di una singola risorsa o di un gruppo di risorse.

Esempio:

Per la seguente metrica informativa, Number of down-times of Server X, presupponendo che l'origine dati fornisca una notifica quando un server è attivo o inattivo e che tale la notifica indichi se è attivo o inattivo in un determinato momento, quindi la registrazione è:

Tipo di evento: evento server attivo/non attivo

Risorsa: server X

In base a questa registrazione, il motore filtra ed esclude tutti gli eventi che presentano le definizioni di attivo/inattivo nel campo Tipo di evento e server X nel campo Risorsa.

Una volta attivato un contratto nel sistema CA Business Service Insight, tutte le metriche registrano gli eventi pertinenti necessari per i calcoli. In base a tali richieste, il motore di correlazione contrassegna gli eventi pertinenti a ogni business logic. Una volta avviati i calcoli, gli eventi pertinenti vengono inviati a ciascuna metrica per il calcolo.

Risorse e relativa gestione

Per consentire la registrazione dinamica, le risorse possono essere allocate singolarmente in base al proprio nome univoco/identificatore o alla loro relazione con gruppi logici.

Esempio:

Per la metrica Overall number of down-times of data center servers, la registrazione è:

Tipo di evento: eventi attivi/non attivi

Risorsa: tutte le risorse che sono contrassegnate nei server di data center. Potrebbe trattarsi di un gruppo di risorse.

Informazioni sul ciclo di vita delle risorse

Una risorsa è un'entità fisica o logica in grado di modificare le proprie caratteristiche nel tempo. Può essere allocata contemporaneamente a determinati componenti del servizio o contraenti, e così via, quindi essere riassegnata in un determinato punto in futuro. Ciascuna di queste modifiche o ri-allocazioni viene acquisita da CA Business Service Insight per eseguire calcoli in qualsiasi momento, in base alla configurazione e all'allocazione delle risorse a un'ora esatta.

È possibile apportare modifiche a una risorsa o alle relative allocazioni in qualsiasi momento, ma è necessario creare una nuova versione di tale risorsa. Ogni nuova versione deve disporre di una data di inizio validità impostata per il momento in cui verranno apportate le modifiche. Le modifiche verranno quindi eseguite in futuro, a meno che non siano riscontrate ulteriori modifiche in una versione successiva di quella stessa risorsa. Tutte le modifiche saranno visibili e disponibili solo per il motore di calcolo una volta attivata e resa effettiva la nuova versione. Questo processo viene definito Conferma della risorsa.

In CA Business Service Insight è previsto anche un metodo per la gestione di più allocazioni di risorse in un unico passaggio. Questo metodo è possibile tramite l'utilizzo di un gruppo di modifiche. I gruppi di modifiche consentono di apportare modifiche a grandi volumi di risorsa in un'unica transazione, analogamente al funzionamento di un database transazionale. È possibile apportare qualsiasi modifica a tutte le risorse che vengono allocate in un gruppo di modifiche tramite l'esecuzione dell'operazione su un gruppo di modifiche intero e, quindi, la conferma del gruppo di modifiche in un unico passaggio.

Durante l'uso delle risorse e delle relative modifiche, è importante tenere in considerazione i seguenti punti in relazione al motore di calcolo:

- Durante l'attivazione (conferma) delle modifiche di una risorsa specifica o un gruppo di risorse (gruppo di modifiche), considerare quali elementi verranno interessati nel sistema. Siccome la modifica del modulo di risorsa potrebbe generare un ricalcolo, è importante ottimizzare la data di attivazione delle risorse e il numero di modifiche attivate in un'unica operazione.
- **Aggiornamento di massa:** è possibile applicare la stessa modifica a molte risorse (aggiornamento di massa). Siccome la modifica del modulo di risorsa potrebbe causare il ricalcolo, è importante ottimizzare l'operazione.

L'esempio precedente affrontava una risorsa non direttamente, ma mediante la sua allocazione logica per la sua funzione o ubicazione (in questo caso, alla funzione, un server di data center).

La richiesta di registrazione potrebbe essere molto scomoda se gli eventi vengono richiesti per ciascun server gestito nel data center. Un problema è il numero di risorse cui viene fatto riferimento. L'altro riguarda la modifica dell'infrastruttura del data center su base periodica, tanto che un server che faceva parte di un data center potrebbe non essere più compreso, oppure potrebbe essere aggiunto un nuovo server. Pertanto, è necessario che l'elenco sia dinamico.

In base all'esempio precedente, è evidente che le risorse devono essere associate a un gruppo logico in modo che possano essere gestite tramite questa entità logica. Inoltre, la gestione potrebbe essere necessaria per il gruppo logico stesso se viene costantemente modificato.

L'allocazione delle risorse è il metodo di CA Business Service Insight per contrassegnare le risorse. Una risorsa può essere assegnata a uno o più gruppi, tipi di risorsa, contraenti o servizi. Le allocazioni delle risorse vengono gestite tramite il Controllo versione di CA Business Service Insight.

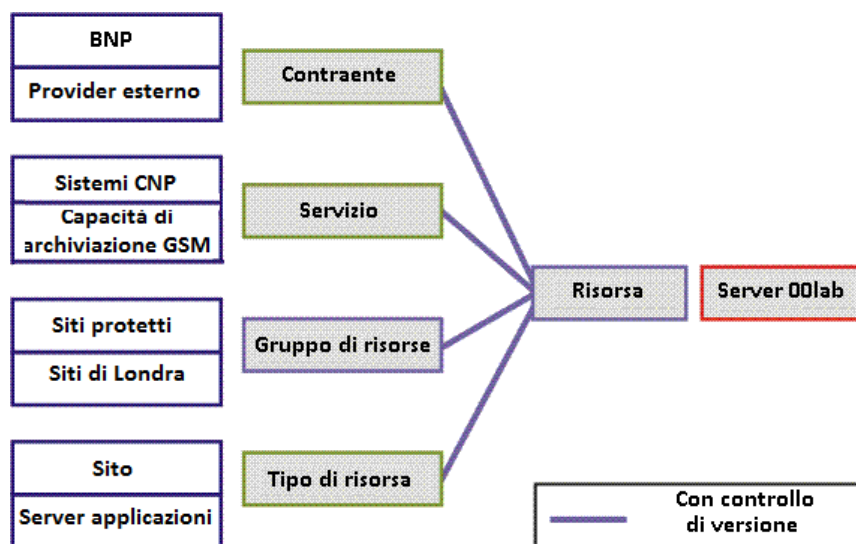
Le risorse disponibili per l'inclusione nei calcoli sono determinate dalle risorse attualmente in vigore all'interno del sistema (in relazione all'intervallo di tempo calcolato al momento).

Ora, tornando all'esempio precedente:

Overall number of down-times of data center servers (numero complessivo di tempi di inattività dei server dei centri dati)

Il data center può essere rappresentato nel sistema come un servizio al quale vengono quindi assegnati tutti i server compresi nel centro dati. Può anche essere definito come un gruppo di risorse denominato server di data center. Questi sono due dei metodi alternativi per scegliere l'allocazione risorse in questo caso specifico, ma esistono altre opzioni disponibili.

Nel seguente diagramma viene illustrato a quali entità una risorsa potrebbe essere associata e il loro utilizzo logico.



Un gruppo di risorse può riflettere qualsiasi aspetto della risorsa necessario per i calcoli, ad esempio la relativa ubicazione o la tecnologia contenuta.

L'allocazione di risorse a queste entità ha come scopo principale quello di verificare la corrispondenza con i requisiti di calcolo e la dinamicità più a lungo possibile del modello.

Attributi personalizzati

Un'altra caratteristica disponibile per una risorsa è la possibilità di specificare attributi personalizzati. Ogni tipo di risorsa può disporre di tali attributi personalizzati aggiunti ad essi e, a loro volta, ogni risorsa definita di quel determinato tipo di risorsa, eredita tali attributi personalizzati.

Utilizzando l'esempio precedente, per ciascun server di datacenter, associare anche il relativo indirizzo IP. Se ciascun server di datacenter viene definito con il tipo di risorsa "server di datacenter", verificare quindi che un attributo personalizzato denominato IP_Address venga aggiunto al tipo di risorsa server di datacenter. In questo modo, ogni risorsa (server) può essere quindi associata al relativo indirizzo IP attraverso l'attributo personalizzato.

Nota: per ulteriori informazioni e un esempio di case study, fare riferimento a [Case study per l'utilizzo degli attributi personalizzati](#) (a pagina 245).

Che cos'è una metrica di gruppo?

Una metrica di gruppo è una metrica impostata per calcolare i livelli di servizio per un gruppo di risorse. I calcoli vengono eseguiti per ciascuna risorsa singolarmente, senza la necessità di duplicare la metrica per ogni registrazione di una nuova risorsa.

Il raggruppamento può essere effettuato solo su un gruppo di risorse che non ha alcun livello di servizio di destinazione associato o che ha lo stesso livello di servizio di destinazione. Ad esempio, la disponibilità di tutti i server applicazioni deve essere non inferiore al 99,9% per server. In questo caso, una metrica è di gruppo per un gruppo di risorse contenente tutti i server applicazioni. Il motore calcola un risultato del livello di servizio separato per ciascun server, in cui la destinazione per ogni server è 99,9%.

Oltre a questo tipo di raggruppamento, CA Business Service Insight supporta un raggruppamento di tipo rollup, che consente la creazione di report per più livelli di risorsa (e gruppo) da una singola metrica. In questo modo è possibile calcolare il livello di servizio su più livelli della gerarchia di risorse ed eseguire la funzionalità di drill-up/down tra le entità correlate di questa struttura della risorsa. Dalla scheda Raggruppamento metriche è possibile attivare questa opzione selezionando le opzioni pertinenti in questa pagina.

☐ Metrica non di gruppo.

☒ Il raggruppamento della metrica è il seguente:

Raggruppamento applicato agli elementi di:

☒ Dinamico

☐ Solo primo livello: membri diretti

☐ Tutti i livelli: include solo risorse

☒ Tutti i livelli: include risorse e gruppi di risorse

☐ Statico. Vengono inclusi solo i seguenti elementi:

Creazione del modello di dati del sistema

Come parte del processo di modellazione dei dati, i componenti richiesti sono identificati in base all'origine dati e ai requisiti di calcolo.

Di seguito è riportato un elenco dei componenti da identificare nel processo di modellazione dei dati e delle rispettive definizioni.

Nome evento	Nome dell'evento come viene visualizzato in CA Business Service Insight; deve essere il più descrittivo possibile.
Comportamento evento	Comportamento dell'evento specificato; quando viene ricevuto dall'origine dati, quali condizioni, e così via.
Campo Data/ora	Campo dell'origine dati utilizzato come data/ora dell'evento.

Campo Tipo di evento	Campo dell'origine dati da convertire in un tipo di evento, che descrive il tipo di report. È importante che il numero di diversi tipi di evento sia ridotto al minimo per quanto possibile, perché la definizione del tipo di evento è manuale e dovrebbe essere eseguita solo una volta.
Campi Dati	Campi dell'origine dati da recuperare come campi di dati.
Campo Risorsa	Campo dell'origine dati da convertire in una risorsa. Contiene un'entità su cui è necessario il report con un ciclo di vita relativamente fisso. La risorsa è un'entità con un ciclo di vita definito, in cui è possibile gestire le modifiche in modo dinamico all'interno del sistema. Il riferimento al ciclo di vita di una risorsa indica la frequenza con cui nuove risorse vengono aggiunte e ne viene modificata l'allocazione in un servizio diverso o in qualsiasi altra entità di allocazione, come descritto in precedenza. Il campo da convertire come risorsa in CA Business Service Insight deve presentare poche modifiche di allocazione e di altro tipo.
Registrazione per	Infine, in base a tutte le definizioni precedenti: Definisce il criterio di registrazione. Il criterio di registrazione definisce il tipo di evento e la risorsa registrati dalla metrica. È possibile eseguire una richiesta di risorsa direttamente mediante la registrazione della risorsa o l'entità di allocazione, quale servizio, contraente, gruppo di risorse, tipo di risorsa, e così via. Tale definizione è determinata dalle caratteristiche funzionali di registrazione. Un altro metodo consiste nella registrazione per metrica, in cui la metrica recupera gli output (risultato del livello di servizio) da un'altra metrica e li utilizza come input. È possibile inoltre utilizzare il risultato da più metriche come input.

Linee guida per la definizione delle registrazioni

Fare riferimento alle seguenti linee guida per la definizione delle registrazioni:

- Non impostare mai la registrazione solo per tipo di evento. Anche se per il calcolo non è richiesto il filtro delle risorse, aggiungere il filtro delle risorse almeno sul tipo di risorsa.

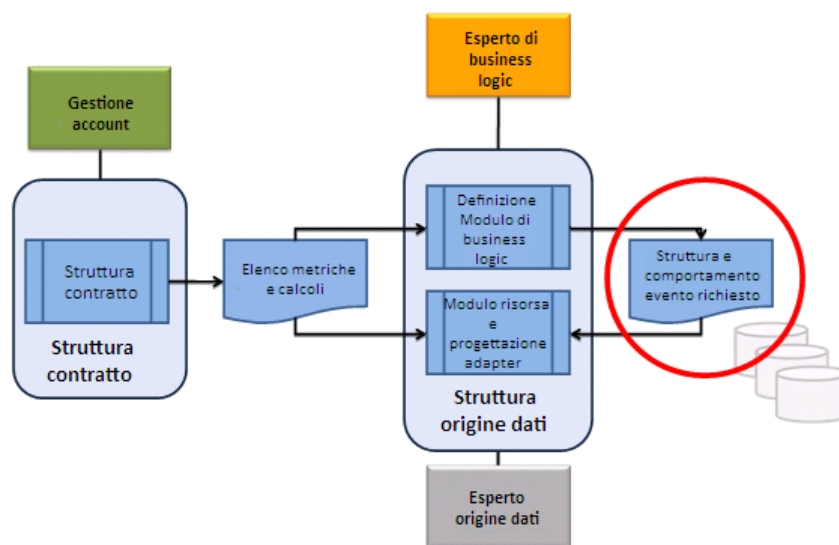
Ad esempio, nel calcolo dei tempi di risposta medi generali dei server applicazioni, è necessario eseguire il report sull'evento del tempo di risposta solo quando è associato a uno di questi server applicazioni (vale a dire, quando il tipo di risorsa associato all'evento è Application Server (Server applicazioni)). In tal caso, il sistema potrebbe disporre di altri tipi di risorse, quali siti o router con lo stesso tipo di evento per inviare dati, quindi per distinguerli. Il tipo di risorsa su cui eseguire il report (server applicazioni) deve essere aggiunto al comando di registrazione come ^{terzo} parametro,

perché quando una risorsa viene modificata, il motore contrassegna la metrica associata a tale risorsa indicando che è necessario il ricalcolo. Nel caso in cui la metrica viene registrata solo per il tipo di evento, il motore visualizza la metrica come registrata a tutte le risorse e pertanto la contrassegna per il ricalcolo all'attivazione di ogni risorsa. Per evitare questa situazione, è necessario utilizzare il ^{terzo} parametro facoltativo del tipo di risorsa.

- Il metodo più efficiente di registrazione è per Contraente e servizio. La disposizione delle risorse in questo modo consente l'espressione della relazione logica tra il livello di dati e il livello di business nel sistema. La registrazione delle risorse attraverso queste entità non richiede la modifica delle formule se utilizzate in diversi contratti o per diversi servizi. Il contratto del contesto di metrica e il servizio definiscono il contraente e il servizio pertinenti. È possibile riutilizzare facilmente le formule di business logic definite in questo tipo di registrazione perché non richiedono modifiche nella registrazione.

Nota: è possibile gestire le registrazioni anche dalla scheda *Registrazione* all'interno di ciascuna metrica. Questa interfaccia offre una procedura guidata attraverso il processo per la metrica selezionata.

Output della fase di modellazione dei dati (esperto delle sorgenti dati ed esperto di business logic)



Per passare alla fase di implementazione della soluzione, è necessario che l'esperto delle sorgenti dati disponga dei seguenti elementi:

- Conoscenza approfondita delle origini dati con cui interfacciarsi, tra cui:
 - Dati cronologici di esempio
 - Metodo di comunicazione
 - Conoscenza dei campi chiave da ogni origine dati da utilizzare per i componenti chiave del modello di risorsa
- Per ogni origine dati è previsto un elenco di tipi di evento che descrive la struttura e il comportamento, comprese le seguenti definizioni:
 - Nome evento
 - Comportamento evento
 - Campo Data/ora
 - Campo Tipo di evento
 - Campi Dati
 - Campo Risorsa

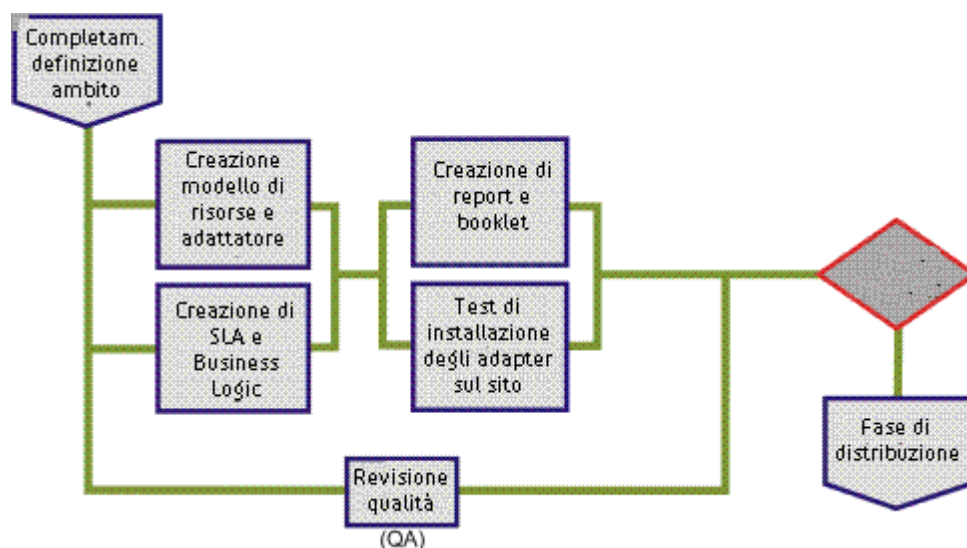
Nota: è necessario creare un documento di riepilogo delle informazioni acquisite.

Capitolo 3: Implementazione

Questa sezione contiene i seguenti argomenti:

- [Implementazione - Introduzione](#) (a pagina 59)
- [Configurazione del framework \(responsabile contratti\)](#) (a pagina 62)
- [Configurazione della libreria dei modelli \(responsabile contratti\)](#) (a pagina 62)
- [Modalità di creazione dei contratti \(responsabile contratti\)](#) (a pagina 63)
- [Raccolta dei dati \(esperto delle sorgenti dati\)](#) (a pagina 68)
- [Implementazione di business logic \(esperto di business logic\)](#) (a pagina 134)
- [Attivazione dei contratti \(responsabile contratti\)](#) (a pagina 177)
- [Creazione dei risultati finali \(responsabile contratti\)](#) (a pagina 180)

Implementazione - Introduzione



In questo capitolo vengono descritti il processo e la logica di base della fase di implementazione del progetto. Come mostrato nella figura precedente, la fase di implementazione segue la fase di progettazione, che è a sua volta seguita dalla fase di installazione e distribuzione.

L'obiettivo della fase di implementazione è rendere il responsabile contratti in grado di completare l'effettiva creazione di tutti gli elementi e gli oggetti nel sistema CA Business Service Insight definiti in precedenza durante la fase di progettazione. Durante la fase di implementazione, il team è impegnato a preparare il sistema per il deployment completo e l'installazione.

Questa fase non deve essere avviata prima di aver completato l'intera fase di progettazione e aver preso in considerazione e definito correttamente tutti i relativi obiettivi. Se la fase di progettazione non è stata completata correttamente oppure non è stato chiaramente definito ogni contratto, metrica, adapter e così via, ci saranno problemi con il sistema o gli elementi mancanti che dovrebbero essere implementati durante la fase di implementazione. La fase di implementazione deve avviarsi solo dopo aver confermato la revisione della progettazione.

Inoltre, è importante che la fase di implementazione sia completata correttamente prima di procedere con l'installazione e il deployment del sistema. Questa operazione deve essere eseguita solo dopo l'esecuzione di un controllo di qualità.

Il processo di configurazione richiede l'esecuzione dei seguenti passaggi da parte del responsabile contratti:

- Configurazione del catalogo dei servizi
- Creazione di contratti
- Attivazione di contratti
- Configurazione delle impostazioni di protezione
- Creazione di report, notifiche, dashboard

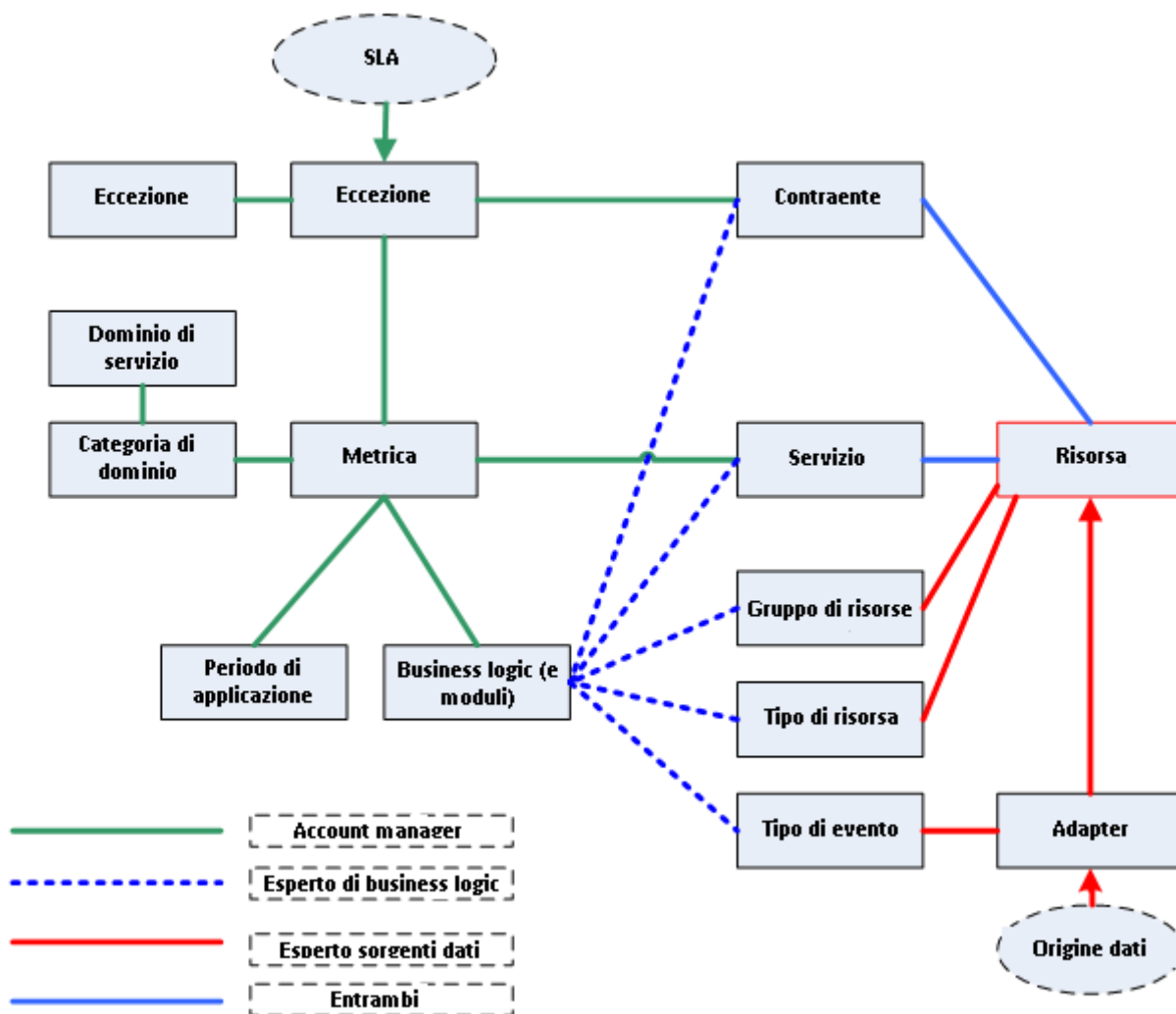
L'esperto delle sorgenti dati deve eseguire la configurazione dell'adapter e l'integrazione con le origini dati. L'esperto delle sorgenti dati deve inoltre eseguire il passaggio di conversione della risorsa per collegare le strutture dell'origine dati alle entità definite nel sistema di CA. Questo passaggio è fondamentale per l'intero processo e può richiedere l'assistenza da parte del responsabile contratti.

Inoltre, è necessario che l'esperto di business logic scriva la business logic per tutte le metriche, secondo i piani della fase di progettazione. Il passaggio potrebbe includere la creazione di tutti i moduli necessari e la configurazione dei parametri associati per fornire le funzionalità di calcolo previste.

Tutti i punti precedenti vengono illustrati in modo approfondito all'interno di questo capitolo.

Nota: per il responsabile contratti è importante tenere presente che le scelte inadeguate effettuate in questa fase possono influire negativamente sul funzionamento di CA Business Service Insight e possono essere difficili o impossibili da invertire in un secondo momento.

Nella figura seguente viene schematizzato il flusso di lavoro logico complessivo.



Configurazione del framework (responsabile contratti)

Il framework consente:

- Definizione dei servizi, gruppi di servizi, domini del servizio e unità
- Creazione e gestione di modelli, inclusi i modelli di business logic e del periodo di applicazione, e i moduli di business logic
- Gestione degli attributi personalizzati per i tipi di risorsa

In questa fase, tutte le entità a livello di sistema identificate durante la fase di progettazione vengono create nella sezione Framework dell'applicazione. Solo quando il sistema contiene tali entità di framework è possibile procedere con la creazione dei contratti e delle metriche associate.

La creazione del framework implica l'aggiunta dei seguenti nuovi elementi:

- Servizi
- Gruppi di servizi
- Domini del servizio e categorie di dominio
- Unità di misura
- Modelli del periodo di applicazione
- Contraenti
- Attributi personalizzati

Per ulteriori informazioni su ciascuno degli elementi sopra elencati, consultare la guida in linea.

Configurazione della libreria dei modelli (responsabile contratti)

Le librerie dei modelli consentono la definizione e gestione di:

- Librerie di modello
- Cartelle dei modelli
- Modelli del livello di servizio
- Modelli di contratto
- Le impostazioni di protezione per i diritti di accesso utente

Per ulteriori informazioni su ciascuno degli elementi sopra elencati, consultare la guida in linea.

Modalità di creazione dei contratti (responsabile contratti)

In questa fase, vengono creati nel sistema i contratti e le rispettive entità correlate definite nella fase di progettazione.

Attenersi alla seguente procedura:

1. Aggiungere un nuovo contratto e applicare i dettagli generali del contratto.
2. Per ogni contratto, definire le metriche e applicare i rispettivi dettagli generali.

In questa fase vengono applicate solo i dettagli generali del contenuto di un contratto, senza includere la business logic e il raggruppamento delle metriche del contratto.

La seguente descrizione di questi passaggi pone l'accento su alcuni punti importanti da considerare in questa fase. Tali passaggi vengono descritti in dettaglio nella guida in linea.

Passaggio 1: aggiunta di un nuovo contratto e applicazione dei dettagli generali del contratto.

La definizione del contratto deve includere:

- Impostazione del nome del contratto.
- Selezione dei contraenti associati.
- Associazione ai servizi correlati.
- Impostazione delle date di inizio validità per il contratto. Le date di inizio validità dei contratti sono l'intervallo di date per il quale il motore di correlazione calcola il livello di servizio per tale contratto; i risultati per i report sono disponibili solo per queste date. Durante l'impostazione delle date, è importante tenere conto dei requisiti in termini di disponibilità dei report correlati al contratto e dei dati non elaborati disponibili.
- Impostazione del fuso orario e della valuta del contratto. Questa definizione è ai fini della creazione di report e consente di creare report su questo contratto per seguire il fuso orario relativo. La definizione della valuta consente al motore di creazione report di determinare in quale valuta visualizzare i risultati di penale per le formule di penale.

Passaggio 2: aggiunta di metriche e dei dettagli generali.

Quando il contratto è in vigore è possibile creare le metriche al suo interno. Nel processo di definizione delle metriche, è necessario eseguire le operazioni seguenti:

- Impostazione del nome della metrica.
- Applicazione dei dettagli della metrica (servizio, dominio, unità, periodo di applicazione, fuso orario, e così via).

- Impostazione delle soglie di dashboard (consultare la sezione [Configurazione delle pagine del dashboard](#) (a pagina 194)).
- Associazione delle metriche associate (se applicabile) e relazione tra di esse.
- Definizione della granularità alla quale la metrica viene calcolata dal motore.
- Impostazione della dichiarazione dell'obiettivo e dei parametri.

In questa fase, la definizione della metrica non include ancora i seguenti elementi:

- Formula/modulo di business logic e registrazione
- Raggruppamento della definizione

Questi elementi sono inclusi nelle metriche del contratto solo dopo aver creato l'infrastruttura del sistema e aver sviluppato e provato le formule di business logic.

Nota: un metodo alternativo per l'approccio appena descritto è sviluppare prima i modelli del livello di servizio all'interno del sistema, invece di definire direttamente i contratti. In questo modo viene creato il modello che può essere utilizzato per creare ulteriori contratti. In alcuni casi, i modelli del livello di servizio preesistenti possono essere importati nel sistema da un'altra istanza di CA. Questo potrebbe consentire a un vantaggio rispetto alla creazione dei contratti ex novo. Per ulteriori informazioni sulla creazione dei modelli del livello di servizio, fare riferimento alla sezione [Creazione dei modelli del livello di servizio](#) (a pagina 66).

In alcuni casi, è consigliabile creare inizialmente un contratto di esempio, per verificare che tutto funzioni correttamente e come previsto. È quindi possibile creare i modelli del livello da questo contratto e archivarli nel catalogo dei servizi, che fornisce un solido punto di partenza per tutti i contratti nel sistema.

Creazione di contratti da un servizio

La creazione di contratti nel sistema utilizzando un catalogo dei servizi, sulla base di un modello o senza un modello (basato su più modelli del livello di servizio), offre una maggiore efficienza e un alto livello di riusabilità, applicabili a molti contratti diversi. In generale, i modelli del livello di servizio contengono una raccolta delle metriche predefinite applicabili a determinati componenti del servizio. Se necessario, è possibile inoltre disporre di più servizi (e metriche associate) in un unico modello del livello di servizio. In genere, i contenuti di un modello del livello di servizio vengono definiti in base alle modalità di utilizzo e possono variare a seconda dei requisiti.

Ad esempio, considerare un servizio di hosting applicazione offerto da un'organizzazione. L'organizzazione può offrire ai propri clienti tre diversi pacchetti di servizio, Minimo, Medio e Massimo, a seconda dei contenuti inclusi nel pacchetto. Un buon utilizzo, ad esempio, dei modelli del livello di servizio può essere la creazione di un modello per ciascun pacchetto.

Una volta creato, è possibile utilizzare tali definizioni per creare un nuovo contratto cliente in modo molto efficiente. Ad esempio, il cliente ABC decide di sottoscrivere il pacchetto di hosting applicazione massimo. È possibile creare questo nuovo contratto nel sistema direttamente tramite il modello del livello di servizio come segue:

Nella pagina Contratti, fare clic su **Aggiungi nuovo**, quindi su **Using Service Catalog** (Utilizzo catalogo dei servizi), e selezionare **Based on Template** (Basato su modello) o **Without Template** (Senza modello). Seguire la **Procedura guidata contratto** per completare la creazione del contratto. Se viene selezionata l'opzione **Based on Template** (Basato su modello), è necessario specificare le impostazioni del modello.

Nella **Procedura guidata contratto** viene visualizzato un elenco di modelli del livello di servizio ed è possibile specificare quali modelli del livello di servizio si desiderano includere nel contratto. Nota: è possibile selezionare metriche specifiche da più modelli del livello di servizio, oppure selezionare semplicemente l'intera definizione per l'inclusione. In questo esempio, tutte le metriche sono prese dal pacchetto di hosting Massimo. Nota: la selezione del livello superiore comporta la selezione automatica di tutti i nodi figlio. Notare inoltre che è possibile assegnare le metriche a un altro servizio se necessario. Tuttavia, per valore predefinito devono essere assegnate agli stessi componenti del servizio della definizione.

Una volta selezionate tutte le metriche necessarie, fare clic sul pulsante **Avanti** per trasferire tali le metriche al nuovo modello di contratto e richiedere il nome del contratto e i dettagli generali da inserire. Fare clic su **Salva e continua** per creare il contratto.

Una volta completato, sono disponibili le opzioni seguenti:

- Continuare con la **Procedura guidata contratto**, definire i parametri ed eseguire la **Procedura guidata di impostazione metriche**, che consente di visualizzare l'interfaccia della procedura guidata e di personalizzare le metriche dei contratti. La procedura guidata consente di rivedere e modificare ciascuna metrica cambiando i campi disponibili, quali **Metric Parameters** (Parametri della metrica) nella dichiarazione dell'obiettivo, **Nome metrica**, **Periodo di applicazione** e la descrizione. Una volta completate le operazioni per ciascuna metrica, la procedura guidata indirizza l'utente alla pagina **Metriche contratto**, in cui è possibile chiudere e salvare il nuovo contratto.
- Aprire la pagina **Contratto** per visualizzare o modificare il contratto.

Creazione dei modelli del livello di servizio

La creazione di un modello del livello di servizio è un processo abbastanza semplice. Da un contratto esistente nella pagina Dettagli contratto selezionare tutte le metriche da includere e fare clic su Salva come modello del livello di servizio.

Nella finestra successiva viene richiesto di denominare il modello del livello di servizio. È quindi possibile salvare il modello del livello di servizio. Una volta salvato, tutti i periodi di applicazione associati alle metriche selezionate vengono inclusi nella scheda Periodo di applicazione. Da questo punto potrebbe essere necessario personalizzare ulteriormente il modello del livello di servizio per renderlo completamente flessibile. A tale scopo è possibile includere l'aggiunta di parametri alle metriche e l'esposizione di questi parametri tramite la dichiarazione dell'obiettivo per ciascuna metrica. È possibile includere forse anche la creazione di parametri del modello del livello di servizio (analogamente ai parametri contrattuali) cui si può fare quindi riferimento con alcune o con tutte le metriche. Una volta completato, il modello del livello di servizio è disponibile per la distribuzione di altri contratti.

Ciclo di vita di un contratto e il controllo delle versioni

Una volta completata la configurazione del contratto, è necessario confermarlo. Questa azione consente al motore di calcolo di iniziare il calcolo dei livelli di servizio per tutte le metriche nel contratto. La conferma del contratto modifica lo stato da Preliminare (in cui è modificabile) a In vigore (in cui non è modificabile). Per eventuali ulteriori modifiche richieste su questo contratto è necessaria la creazione di una nuova versione del contratto. Se per la nuova versione vengono selezionate le stesse date di inizio validità, a partire da tali date le modifiche vengono completate e la nuova versione viene confermata, sovrascrivendo completamente la versione precedente. Questa operazione inoltre attiva il motore per ricalcolare le metriche diverse rispetto alla versione precedente. Le versioni in vigore possono inoltre sovrapporsi parzialmente, ad esempio quando viene apportata una modifica alle destinazioni di alcune metriche a metà del contratto durante le date di inizio validità. In questo caso, la versione precedente viene ancora utilizzata finché la data di inizio validità della seconda versione non diventa effettiva. In questo momento, la seconda versione assume lo stato in vigore per i calcoli.

Nella seguente tabella vengono mostrate le modifiche utente rispetto all'ambito dell'impatto del ricalcolo e all'intervallo di tempo. Una modifica può influire su una determinata metrica in una versione specifica del contratto o sulle metriche presenti in diversi contratti e versioni del contratto.

Modifica	Impatto di ambito	Intervallo di impatto
Modifiche apportate in Metriche		
Modifica Dettagli metrica - formula di business logic	Impatto su tutte le metriche all'interno di una versione specifica del contratto	Dall'inizio della versione effettiva del contratto

Modifica Dettagli metrica -valore di destinazione	Impatto su tutte le metriche all'interno di una versione specifica del contratto	Dall'inizio della versione effettiva del contratto
Modifica Dettagli metrica - periodo di riferimento	Impatto su tutte le metriche all'interno di una versione specifica del contratto	Dall'inizio della versione effettiva del contratto
Modifica Dettagli metrica - Parametri della metrica	Impatto su tutte le metriche all'interno di una versione specifica del contratto	Dall'inizio della versione effettiva del contratto
Modifica Dettagli metrica - fuso orario	Impatto su tutte le metriche all'interno di una versione specifica del contratto	Dall'inizio della versione effettiva del contratto
Modifica Dettagli metrica - raggruppamento	Impatto su tutte le metriche all'interno di una versione specifica del contratto	Dall'inizio della versione effettiva del contratto
Modifica Dettagli contratto - date di inizio validità	Impatto su tutte le metriche all'interno di una versione specifica del contratto	Dall'inizio della versione effettiva del contratto
Modifica Dettagli contratto - periodi di applicazione	Impatto su tutte le metriche all'interno di una versione specifica del contratto	Dall'inizio della versione effettiva del contratto
Modifica parametri contrattuali	Impatto su tutte le metriche all'interno di una versione specifica del contratto	Dall'inizio della versione effettiva del contratto
Modifica moduli SLALOM	Impatto su tutte le metriche associate al modulo SLALOM modificato, in tutti i contratti e le versioni del contratto	Dall'inizio della versione effettiva del contratto
Operazioni generali		
Modifica del modello di risorsa della metrica/gruppo di modifiche (CA Business Service Insight 4.0)	Impatto su tutte le metriche che registrano la risorsa, in tutti i contratti e le versioni del contratto	Dallo stato più recente prima della data in cui è stata modificata la risorsa
Recupero di eventi posticipati per la metrica Eventi con data/ora precedenti (dati non elaborati o eventi riutilizzabili)	Impatto su tutte le metriche che usano la risorsa, in tutti i contratti e le versioni del contratto	Dallo stato più recente prima della data in cui è stato modificato l'evento
Aggiunta di dati di correzione della metrica	Impatto su tutte le metriche che usano la risorsa, in tutti i contratti e le versioni del contratto	Dallo stato più recente prima della data in cui è stata modificata la correzione

Modifica, aggiornamento o eliminazione dei periodi di eccezione della metrica (attivazione e disattivazione)	In base alle impostazioni di eccezione e all'implementazione specifica, l'impatto si verifica su una determinata metrica in una versione specifica del contratto o su metriche presenti in diversi contratti e versioni del contratto	Il più vicino possibile al periodo di eccezione
Aggiornamento dell'attributo personalizzato	Impatto su tutte le metriche che registrano la risorsa, in tutti i contratti e le versioni del contratto	Dallo stato più recente prima della data in cui è stata modificata la risorsa

Infine, alcuni punti chiave da tenere presente sulle versioni di contratto:

- Se la nuova versione ha la stessa data di inizio validità, solo le metriche modificate vengono ricalcolate dall'inizio della versione.
- Se la nuova versione presenta date diverse, tutte le metriche nella nuova versione vengono calcolate dall'inizio di tale versione e tutte le metriche della versione precedente vengono ricalcolate da un determinato punto in tale versione finché la nuova non diventa effettiva. L'esatta quantità di ricalcolo dipende dalla configurazione degli stati.
- È consigliabile creare contratti con date di inizio validità per un anno e rinnovarli alla scadenza. In questo modo si evitano periodi di ricalcolo superiori a un anno.
- Vengono calcolate le versioni non in vigore dei contratti (la data corrente è successiva alla fine delle date di validità del contratto); pertanto, vengono considerate metriche attive nel sistema, dal momento che calcolano i dati del livello di servizio associati per la creazione di report.

I valori associati alle variabili globali all'interno delle metriche non sono passati tra le versioni del contratto, ossia, la routine OnLoad nella business logic viene invocata all'inizio di ogni versione del contratto.

Nota: per alcuni scenari di prova e casi di studio, fare riferimento alla sezione [Casi di studio sulla modellazione dei contratti](#) (a pagina 215).

Raccolta dei dati (esperto delle sorgenti dati)

Nella fase di raccolta dei dati del processo di implementazione vengono utilizzati gli adapter. Negli argomenti di seguito viene delineato il processo.

Funzionalità dell'adapter

Gli adapter sono moduli di raccolta dei dati da origini dati e trasferimento al sistema CA Business Service Insight. Gli adapter filtrano i dati provenienti dall'origine dati e li modificano in modo tale che quando raggiungono il sistema contengono solo le informazioni necessarie per i calcoli del livello di servizio nella struttura corretta.

La piattaforma degli adapter offre la flessibilità di:

- Ricevere i dati in linea o non in linea con la frequenza necessaria
- Ricevere i dati a vari livelli: non elaborati, calcolata o aggregati
- Ricevere i dati da una vasta gamma di tipi di strumento

In sostanza, ogni adapter è costituito da due componenti:

- **Componente generico dell'adapter:**

Esistono due tipi di componente generico dell'adapter, un componente adapter per file ASCII e un componente adapter SQL basato su ODBC (Open Database Connectivity). Questi componenti consentono la connessione a un'origine dati e l'analisi come un file ASCII o l'esecuzione di una query SQL su tale origine dati.

- **File di configurazione dell'adapter:**

Ogni adapter richiede un file di configurazione per conoscere il punto e la modalità di connessione, i dati da recuperare e il modo di trasformare e convertire i dati in transazioni ed eventi generici di CA. CA Business Service Insight fornisce a ogni tipo di adapter generico un file modello di configurazione XML predefinito che può essere ottimizzato per rappresentare le specifiche del cliente in merito all'origine dati con cui connettersi. Il file di configurazione XML definisce quali campi devono essere recuperati, come devono essere identificati, come devono essere convertiti nel database di sistema normalizzato, e così via.

Nota: nell'interfaccia utente è presente una procedura guidata di configurazione dell'adapter che consente la personalizzazione di base di questo modello XML in linea. Tale procedura ha lo stesso scopo della creazione del file di configurazione XML per l'adapter. È possibile trovare ulteriori informazioni su questa funzionalità più avanti in questo capitolo.

La piattaforma dell'adapter include un meccanismo di riavvio/ripristino e può gestire problemi nei dati ricevuti da strumenti di terze parti, ad esempio arresto improvviso dello strumento, problemi di rete, dati mancanti, dati duplicati, dati errati, lacune nei dati, convalida dei dati e così via. Le funzionalità offerte da ogni adapter, integrità dei dati, completo rilevamento e registrazione dei messaggi di tutti gli adapter, vengono trattate più avanti in questa guida in modo più dettagliato.

Gli adapter CA Business Service Insight possono essere eseguiti su richiesta o come un'applicazione (visibile o non visibile). La tecnologia dell'adapter CA Business Service Insight supporta meccanismi di protezione avanzata, quali la crittografia, l'handshake e i processi di autenticazione.

È importante tenere presente in questa fase che la procedura guidata di configurazione dell'adapter è un meccanismo di automatizzazione dei processi e delle attività seguenti. Mentre alcuni elementi citati potrebbero non essere sempre visibili quando si utilizza la procedura guidata, sono comunque presenti "dietro le quinte" dell'interfaccia della procedura guidata.

Ambiente dell'adapter

Le seguenti entità si riferiscono all'adapter e ai relativi parametri di configurazione ed esecuzione.

Origine dati:

Origine dati cui l'adapter si connette e da cui recupera i dati nel formato originale.

File di lavoro:

I file di output generati dall'adapter e scritti nei processi (per ulteriori informazioni, consultare la sezione [File di lavoro](#) (a pagina 73)).

Listener dell'adapter CA Business Service Insight:

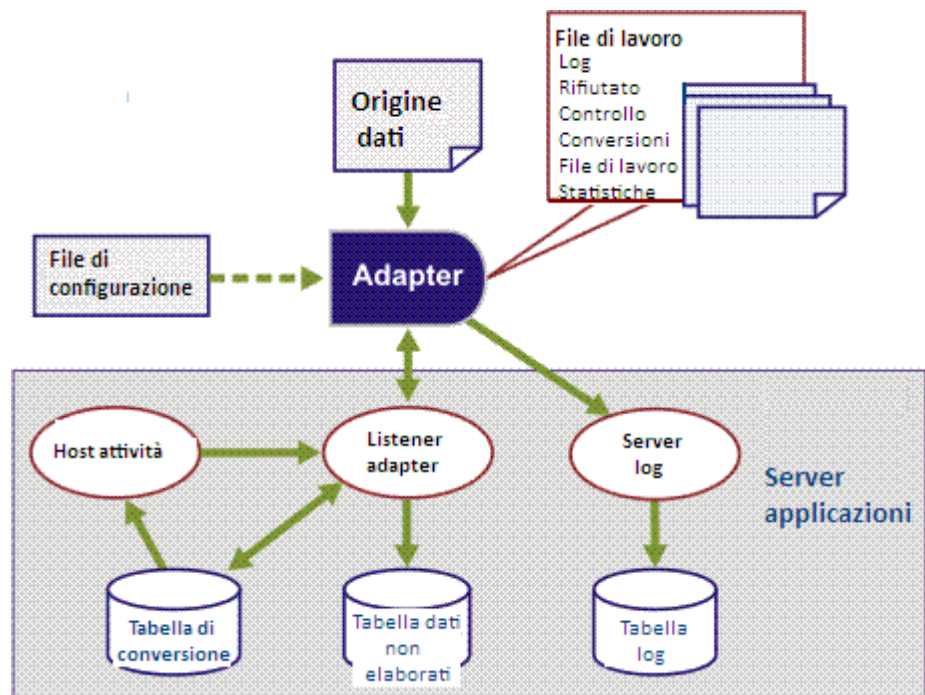
Tre tipi di messaggio vengono trasferiti tra l'adapter e il listener dell'adapter:

- **Controllo:** messaggi di avvio\interruzione\sospensione inviati dal listener all'adapter e vice versa quando l'adapter cambia stato.
- **Conversioni:** richieste inviate dall'adapter per i contenuti della tabella di conversione e i valori specifici di conversione. Il listener restituisce le tabelle e il valore convertito. Il listener dell'adapter riceve l'indicazione dal servizio di host delle attività del fatto che la voce di conversione è stata convertita. Il messaggio viene quindi inviato all'adapter.
- **Dati non elaborati:** eventi unificati di dati non elaborati inviati dall'adapter. Questi eventi sono inviati in pacchetti e includono le conferme di ricezione.

Server di registrazione dei log CA Business Service Insight

L'adapter può essere configurato per l'invio dei messaggi di log per il log di sistema, nonché per la scrittura in un file locale. Se la porta e l'indirizzo IP del server di registrazione dei log sono specificati e configurati nelle impostazioni del registro dell'adapter, l'adapter può anche inviare i messaggi al server di registrazione dei log.

Nel diagramma seguente viene descritto il processo dell'adapter in relazione a ogni entità con cui interagisce.



Di seguito è riportata una descrizione dell'interazione del processo dell'adapter con queste entità:

File di configurazione:

Contiene le impostazioni per alcuni o tutti i parametri di configurazione dell'adapter. L'adapter utilizza il file di configurazione per determinare il metodo di connessione utilizzato dall'adapter e le metriche di l'analisi per creare l'output dell'evento. Si tratta di un file XML e il formato contiene sei elementi di base:

General:

Vari attributi dell'adapter (directory di lavoro, file di output, contrassegno di debug).

OblicoreInterface:

Attributi per la connessione con il server CA Business Service Insight.

DataSourceInterface:

Attributi per la connessione con l'origine dati (percorso del file e modello, stringhe di connessione, query SQL, e così via)

InputFormatCollection:

Metrica di analisi per l'analisi e la modifica dei dati di origine.

TranslatorCollection:

Metriche per la creazione dell'evento unificato composto dai campi di dati analizzati e modificati.

TranslationTableCollection:

Metriche per i dati di mapping tra i dati originali e le entità CA Business Service Insight.

Ciascuna delle sei sezioni contiene tutte le informazioni pertinenti che consentono all'adapter di connettersi all'origine dati, recuperare le informazioni richieste, analizzarle nelle strutture di eventi unificati CA Business Service Insight e archiviarle nella tabella dei dati non elaborati CA Business Service Insight.

File principali

L'adapter è composto da due file principali: i file eseguibili e i file di configurazione. Il file eseguibile è un file generico. Esistono due file eseguibili: adapter SQL e adapter per file di testo.

Un file di configurazione XML è adatto per ogni adapter in modo da archiviare i requisiti dell'origine dati specifici. Il file di configurazione consente di specificare le informazioni relative all'origine dati (nome, ubicazione, metodo di connessione e struttura) e la struttura degli eventi di output da generare con l'adapter.

Il file di configurazione include parametri e valori impostati per gli attributi all'interno di un file XML strutturato predefinito.

Durante la creazione di un nuovo adapter, è necessario utilizzare i file eseguibili pertinenti (secondo il tipo di origine dati di destinazione, file per le origini dati di file flat, SQL per origini dati di database), quindi modificare il file di configurazione in base alle esigenze. Le due strutture contengono elementi di configurazione leggermente diversi a seconda che si tratti di un adapter per file di testo o di un adapter SQL. Di norma, viene impostato automaticamente durante la creazione dell'adapter mediante l'utilità Adapter Manager.

Altri file correlati agli adapter sono i file di lavoro creati dall'adapter nel processo di lettura delle origini dati e di scrittura degli eventi nel sistema CA Business Service Insight.

Nota: per informazioni sulla modifica del file di configurazione, consultare la sezione [Specifiche di configurazione dell'adapter](#) (a pagina 317).

File di lavoro

I file di lavoro dell'adapter vengono creati durante la prima esecuzione dell'adapter e vengono aggiornati costantemente durante ogni esecuzione dell'adapter.

Ogni adapter dispone di propri file di lavoro. I nomi dei file di lavoro possono essere impostati nel file di configurazione dell'adapter (facoltativo) o conservare i nomi predefiniti. Il percorso del file di lavoro viene impostato dalla cartella di lavoro e può essere impostato nel file di configurazione. Nota: il percorso specificato potrebbe essere relativo alla directory attuale in cui si trova l'adapter. Il percorso specificato deve esistere già (o è necessario crearlo) per la corretta esecuzione dell'adapter.

Nota: la cartella non viene creata automaticamente se inesistente.

Tutti i parametri pertinenti nel file di configurazione sono contenuti nella sezione General. Solo il percorso del file di log è impostato nel registro di sistema o passato tramite la riga di comando.

AdapterStatistics.txt

L'adapter scrive le informazioni statistiche in questo file con intervalli di un minuto. L'ultima riga viene scritta quando l'adapter si interrompe. Ogni riga del file contiene i numeri di:

- Eventi ricevuti
- Eventi ignorati
- Eventi con errori
- Messaggi inviati
- Pacchetti inviati

Durante ogni esecuzione dell'adapter vengono avviate le statistiche.

RejectedEvents.txt

Questo file contiene tutti gli eventi che l'adapter non è riuscito a inviare a CA Business Service Insight perché il valore Evento, definito con la necessità di conversione, non presenta un ID corrispondente nella tabella di conversione. Questo significa che la conversione pertinente non è stata eseguita. Ogni evento che ha almeno un valore in attesa di conversione viene scritto nel file rejectedEvents.

All'inizio di ogni esecuzione, l'adapter prima tenta di inviare a CA Business Service Insight gli eventi dal file rejectedEvents perché cerca di trovare l'ID corrispondente per il valore pertinente nella tabella di conversione. Se il valore viene trovato, l'adapter invia l'evento e lo elimina dal file. Se non viene trovato un valore corrispondente, l'evento rimane nel file rejectedEvents.

È possibile configurare il limite massimo per il numero di eventi rifiutati che può essere raggiunto impostando il parametro RejectedEventsUpperLimit (Limite massimo di eventi rifiutati) nel file di configurazione dell'adapter. Quando viene raggiunto il limite, l'adapter interrompe la lettura di nuovi record e inserisce lo stato Bloccato. È possibile vederlo quando viene visualizzato l'output di debug sullo schermo durante l'esecuzione dell'adapter. Se viene visualizzata una stringa continua di lettere maiuscole B, l'adapter è bloccato e richiede la conversione di alcune delle voci in sospeso prima del caricamento di ulteriori dati.

Gli eventi in sospeso vengono scritti nel file in formato XML. Di seguito è riportato un esempio di un evento dal file:

```
<rejectedEvent
  createDate="1062330841"
  translator="Translator">
  <event
    inputFormat="InputFormat">
      <field name="resource" type="3" value="Server333p"/>
      <field name="timestamp" type="4" value="1036108800"/>
      <field name="memory_utilization" type="2" value="26.71"/>
      <field name="cpu_utilization" type="2" value="78.85"/>
    </event>
  </rejectedEvent>
```

Log dell'adapter

Il log dell'adapter è il file in cui l'adapter scrive i messaggi di log.

Si consiglia di esplorare il file di log dell'adapter tramite l'utility Log Browser.

È possibile impostare il livello di report per questo file di log modificando un parametro nel file di configurazione dell'adapter, LogDebugMode (Modalità debug log). Se impostato su "sì", l'adapter scrive i messaggi di indicazione normale nel log, nonché i record originali, i risultati di analisi e l'evento destinato.

Questo parametro è in genere impostato su "yes" (sì) durante la verifica e il monitoraggio dell'adapter.

Per impostazione predefinita, la dimensione del file è limitata a 1 MB. Quando il limite viene raggiunto, il nome dell'adapter viene modificato con l'aggiunta di _old e viene creato un nuovo file di log. L'adapter può contenere potenzialmente fino a 2 MB di messaggi di log; 1 MB per il file precedente e 1 MB per il file corrente.

Il limite delle dimensioni del file di log può essere configurato come una voce nel registro di sistema per ogni adapter con un massimo di 10 MB. La voce nel registro di sistema è denominata LogFileMaxSize ed è definita nell'adapter specifico con un valore multiplo di KB.

DataSourceControl.xml

Il file DataSourceControl.xml viene utilizzato dall'adapter per il controllare l'accesso all'origine dati e verificare che durante ogni esecuzione continui sempre dall'ultimo punto in cui è stata effettuata la lettura.

L'adapter per file conserva il nome dell'ultimo file letto, l'ultima riga letta e la posizione raggiunta nel file. Con la prossima esecuzione, l'adapter accede al file dal percorso utilizzando le informazioni contenute nel file DataSourceControl.xml. Grazie a questo meccanismo, l'adapter può leggere solo i nuovi record in ogni esecuzione.

L'adapter non utilizza direttamente i file di origine, ma prima copia il file corrente in un file di lavoro. Di conseguenza, le stesse informazioni vengono conservate per il file di lavoro e il file di origine. Se viene aggiunto il file di origine, nel file di lavoro vengono copiati solo i nuovi record.

Se l'adapter è configurato per eliminare il file di origine una volta elaborato, impostando il parametro nel file di configurazione DeleteAfterProcessing (Elimina dopo elaborazione) su "yes" (sì), le informazioni non vengono salvate nel file di origine. Al termine, vengono letti tutti i nuovi file esistenti nella cartella di lavoro che corrispondono al modello di file definito nel file di configurazione.

Solo quando DeleteAfterProcessing (Elimina dopo elaborazione) è impostato su "no" viene eseguito il controllo di nuovi record nell'ultimo file. Se non ne sono presenti, passa al file successivo in un ordine lessicografico. Pertanto, durante la denominazione dei file di origine dei dati, per accertarsi che siano letti nella sequenza corretta, applicare un modello di denominazione in sequenza crescente. Ad esempio, aggiungere i file con un valore di data inverso (yyyymmdd-hhmmss) per supportarlo. Ad esempio:

- DataSourceABC20070517-14:00:00.csv
- DataSourceABC20070517-15:30:00.csv
- DataSourceABC20070517-17:00:00.csv e così via.

Di seguito è riportato un esempio di DataSourceControl.xml per un adapter per file.

```
<AdapterControl Save="end" LastSaveTime="2005/05/20 13:06:39">
<Data><WorkData><LineNumber>0</LineNumber>
<FilePattern>c:\adapters\callsadapter\*adapterpca.csv</FilePattern>
[set the File Name variable]</FileName>
<BasicPosition>0</BasicPosition>
</WorkData>
<NonDeletedFiles>
<File NamePattern="c:\adapters\callsadapter\*adapterpca.csv">
[set the File Name variable]2005adapterpca.csv</FileName>
<LastLine>25/04/2005,5925,NN4B,12,12,0,10,0,11</LastLine>
<LastPosition>15427</LastPosition></File>
</NonDeletedFiles>
</Data>
</AdapterControl>
```

Un adapter SQL mantiene l'ultimo valore dei campi chiave della query per ogni query eseguita. I campi chiave sono identificatori univoci per i record nella tabella del database di destinazione. L'adapter utilizza questi valori durante la compilazione della query per l'esecuzione successiva. In questo modo l'adapter legge solo i nuovi record.

Ad esempio, considerare la seguente istruzione SQL utilizzata per il recupero dei dati del ticket di problema.

```
Select ticket_id, status, organization, open_date, respond_date, resolved_date,
record_modified_date from t_ticket_data;
```

In questo esempio, per acquisire tutti gli ultimi i record dall'origine dati, il campo chiave della query selezionato per assicurare il recupero delle informazioni più recenti è record_modified_date, perché genera nuovi ticket creati dall'ultima esecuzione dell'adapter, nonché gli aggiornamenti per i ticket esistenti. Con la selezione di questo campo come campo chiave della query, l'adapter aggiunge automaticamente la seguente sezione alla fine della query durante l'esecuzione:

```
where record_modified_date > :previous_value order by record_modified_date asc
```

Pertanto, recupera solo i record più recenti. Nota: occorre tener conto di alcune considerazioni durante la selezione del campo chiave della query, che dipendono sempre dal comportamento dell'origine dati e dal risultato che l'utente desidera ottenere con i dati recuperati. Notare inoltre che i campi selezionati nell'esempio precedente non sono sempre la scelta migliore per *ogni* situazione.

Un esempio di file DataSourceControl.xml per un adapter SQL è indicato di seguito.

```
<AdapterControl Save="end" LastSaveTime="2005/05/20 15:59:02">
<Data><QueryCollection>

<Query QueryName="ChangeManagementOpenQuery">
<KeyField Name="Incident Ref"><LastValue>32357</LastValue></KeyField>
<KeyField Name="Date Logged"><LastValue>18/04/2005
16:56:26</LastValue></KeyField>
<LastFileName/>
</Query>

<Query QueryName="ChangeManagementPendingQuery">

<KeyField Name="Incident Ref"><LastValue>0</LastValue></KeyField>
<KeyField Name="Date Resolved"><LastValue>1900-01-
01</LastValue></KeyField><LastFileName/>

</Query>
```

send.txt

Tutti gli eventi che vengono creati e sono pronti per essere inviati a CA Business Service Insight vengono inizialmente scritti nel file da inviare.

SendControl.xml

Il file sendcontrol.xml contiene tutte le righe inviate e riconosciute da CA.

Il file consente all'adapter di seguire il protocollo di riconoscimento della finestra scorrevole per il trasferimento di dati a CA. È possibile trovare ulteriori informazioni su questo meccanismo nella sezione [Comunicazione dell'adapter](#) (a pagina 80).

```
<SendControl  
PacketMaxSize="50"  
LastAckSequence="47"  
LastAckIndex="-1"/>
```

File di output IllegalEvents.txt

L'adapter scrive sul file di output IllegalEvents tutti i record letti, ma con un errore durante l'analisi. In genere, è causato dalla logica di convalida inserita nel file di configurazione dell'adapter. L'adapter salva questi record se il parametro SaveIllegalEvents (Salva eventi non validi) nel file di configurazione è impostato su "yes" (sì). Nota: il percorso dell'opzione deve essere impostato con IllegalEventsDirectoryName. La cartella deve esistere già, perché non viene creata automaticamente. Se la cartella è inesistente, l'adapter restituisce un errore durante l'esecuzione.

All'interno di un adapter per file, il file contenente i record di errore ha lo stesso nome del file di origine, mentre nell'adapter SQL reca il nome della query.

Translations.xml

Il file translation.xml contiene le tabelle di conversione dell'adapter.

Quando l'adapter viene eseguito in modalità in linea, il file contiene una copia della tabella di conversione dal database. Se la tabella di conversione è configurata come remota, l'adapter carica la tabella di conversione dal database in questo file, sostituendolo. Se autonoma, legge il file locale.

Quando l'adapter viene eseguito in modalità non in linea, utilizza solo il file come tabella di conversione (per ulteriori informazioni sulle modalità in linea/non in linea, fare riferimento alla sezione [Modalità dell'adapter](#) (a pagina 85)).

```
<TranslationTableCollection>

<AssystResourceTable>
<Entry TranslationStatus="Ignored" resource="Authority"/>
<Entry TranslationStatus="Translated" resource="LONDON" TranslationTo="1006"/>

</AssystResourceTable>

<AssystSupplierEventTypeTable>
<Entry TranslationStatus="Translated" severity="1" TranslationTo="1545"/>
<Entry TranslationStatus="Translated" severity="2" TranslationTo="1550"/>
<Entry TranslationStatus="Translated" severity="3" TranslationTo="1551"/>

</AssystSupplierEventTypeTable></TranslationTableCollection>
```

Comunicazione dell'adapter

Gli adapter interagiscono con l'origine dati da un lato e con il listener dell'adapter CA Business Service Insight e il server di registrazione dei log dall'altro, come illustrato nel seguente diagramma.



L'adapter comunica con l'origine dati per recuperare i dati tramite una connessione ODBC e può trovarsi in locale o in remoto dall'origine dati, a condizione che l'adapter sia in grado di stabilire la connessione ODBC.

L'adapter comunica con il server applicazioni CA Business Service Insight utilizzando il protocollo TCP/IP e pertanto può trovarsi in locale o in remoto, a condizione che sia in grado di stabilire la connessione TCP/IP.

L'adapter deve avere due porte aperte, una per il listener dell'adapter e una per il server di registrazione dei log. Le porte del listener dell'adapter devono essere univoche per ogni adapter e non devono essere in conflitto con altre operazioni di rete o applicazioni che potrebbero utilizzare tali porte. Ad esempio, si consiglia di non utilizzare la porta 1521, poiché in genere viene utilizzata dal protocollo Oracle TNS (Transparent Network Substrate) per la comunicazione con il database, e così via. Inoltre, è necessario considerare eventuali firewall locali che potrebbero bloccare il traffico.

Nota: verificare con l'amministratore locale in caso di incertezza su quali porte siano disponibili per l'utilizzo, oppure se è necessario richiedere l'apertura delle porte per consentire la comunicazione.

La porta e l'indirizzo del listener dell'adapter sono impostati nel file di configurazione dell'adapter. La porta e l'indirizzo IP del server di registrazione dei log sono impostati tramite le voci dell'adapter nel registro di sistema.

È possibile configurare l'operazione client/server rispetto al listener dell'adapter, in modo che l'adapter sia configurato per operare come un client o come un server. La configurazione dell'operazione client/server viene eseguita sul lato dell'adapter laterale nei parametri del file di configurazione. A tale scopo, le variabili Port, Address e ConnectionInitiator devono essere impostate di conseguenza.

Se ConnectionInitiator è impostato per essere l'adapter, è obbligatoria solo una porta di destinazione. Se è impostato per essere CA Business Service Insight, sono richiesti una porta e un indirizzo IP del listener dell'adapter su CA Business Service Insight. Per impostazione predefinita, il server viene impostato per essere l'adapter. In alcuni casi una funzione importante è l'abilitazione dell'attivazione di una regola firewall (funzionalità nota come attivazione delle porte). A volte solo un firewall consente una richiesta interna su una porta, se un messaggio è stato inviato dall'interno del firewall sulla stessa porta. Quindi, il firewall viene attivato per consentire la comunicazione.

Nota: rivolgersi al proprio amministratore di rete per ulteriori informazioni sulle condizioni locali che potrebbero influire sulle comunicazioni dell'adapter.

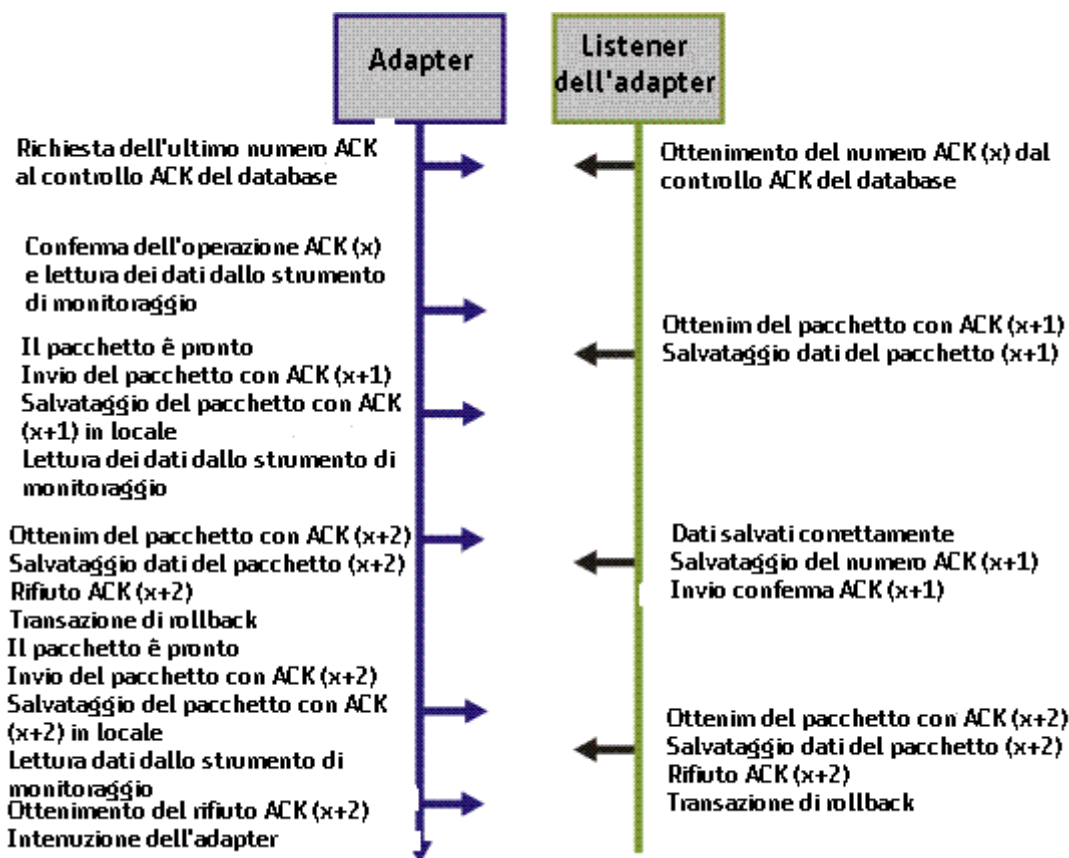
Da un punto di vista della protezione, è consigliabile che l'adapter sia impostato per essere il client poiché in questo modo si garantisce la destinazione degli eventi quando si opera in un ambiente di deployment multiplo per test e produzione.

Per verificare la corretta trasmissione di record dei dati dall'adapter al listener dell'adapter CA Business Service Insight, l'adapter comprende un algoritmo di finestra scorrevole/ACK a livello TCP/IP. Questo algoritmo in pratica invia i dati in pacchetti e quindi attende la conferma di ricezione dal listener dell'adapter prima di passare al pacchetto successivo. Ogni pacchetto contiene diversi messaggi di dati non elaborati. È possibile configurare il numero di messaggi in un pacchetto impostando il parametro Packet Size (Dimensioni pacchetto). Ogni pacchetto ha una sequenza contenuta nel messaggio di riconoscimento. Tutti i parametri appropriati che controllano il processo sono contenuti nella sezione Interface CA Business Service Insight del file di configurazione. In generale, tuttavia, non è necessario modificare tali parametri.

Il listener dell'adapter scrive i dati non elaborati nel pacchetto in una singola transazione.

Nota: l'operazione ACK può essere eseguita solo sui messaggi di dati non elaborati inviati a CA Business Service Insight.

Nella figura seguente viene illustrato il processo di comunicazione dell'adapter.



Impostazioni del registro di sistema dell'adapter

Nei casi in cui mancano informazioni dalla riga di comando, l'adapter utilizza alcune definizioni archiviate nel registro di sistema sul server in cui l'adapter è installato.

Le voci del registro di sistema vengono scritte mediante l'utilità Adapter Manager se l'utilità è stata utilizzata per l'installazione dell'adapter. Se non è stata impiegata per l'installazione dell'adapter, è possibile aggiungere manualmente tali voci al registro di sistema.

Nota: in caso di installazione di un adapter in un ambiente UNIX, è necessario aggiungere tali voci manualmente in quanto non esiste alcuna utilità Adapter Manager per questo ambiente.

Di seguito viene riportato un elenco con le voci del registro di sistema utilizzate dall'adapter e dall'utilità Adapter Manager.

Voci di server generale

Le seguenti voci vengono scritte nel registro di sistema
 \HKEY_LOCAL_MACHINE\SOFTWARE\Oblicore\Adapters:

Proprietà disponibili:

Nome	Tipo	Descrizione
AdaptersDir	Stringa	Directory principale per tutti gli adapter.*
FileAdapterConfTemplate	Stringa	Percorso del modello di configurazione dell'adapter per file.* Adapter Manager utilizza queste informazioni per copiare il modello di configurazione nella cartella del nuovo adapter in cui viene specificato, come parte del processo di creazione dell'adapter.
GenericFileAdapter	Stringa	Eseguibile dell'adapter per file.* Adapter Manager crea un collegamento all'eseguibile o lo copia nella cartella del nuovo adapter in cui viene specificato, come parte del processo di creazione dell'adapter.
GenericSqlAdapter	Stringa	Eseguibile dell'adapter SQL.* Adapter Manager crea un collegamento all'eseguibile o lo copia nella cartella del nuovo adapter in cui viene specificato, come parte del processo di creazione dell'adapter.
LogServerAddress	Stringa	Indirizzo di rete del server di registrazione dei log. (Facoltativo)** Porta del server di registrazione dei log, in genere 4040. (Facoltativo)** Nel caso in cui questi parametri vengono impostati, l'adapter invia i messaggi di log al server di registrazione dei log CA Business Service Insight.

Nome	Tipo	Descrizione
LogServerPort	Stringa	
SqlAdapterConfTemplate	Stringa	Percorso del modello di configurazione dell'adapter SQL.* Adapter Manager utilizza queste informazioni per copiare il modello di configurazione nella cartella del nuovo adapter in cui viene specificato, come parte del processo di creazione dell'adapter.

* Utilizzato solo dall'utility Adapter Manager

** Utilizzato dall'adapter

Voci di adapter singolo

Le seguenti voci vengono scritte nel registro di sistema
 \HKEY_LOCAL_MACHINE\SOFTWARE\Oblicore\Adapters\<Adapter Name>:

Proprietà disponibili:

Nome	Tipo	Descrizione
ConfigurationFileName	Stringa	Nome del file di configurazione dell'adapter.**
Directory	Stringa	Directory dell'adapter.*
LogFileName	Stringa	Nome del file di log dell'adapter.**
Path	Stringa	Percorso dell'eseguibile dell'adapter.*
RunAs	Numero	Modalità di esecuzione.* Servizio/applicazione console/applicazione Windows
Tipo	Numero	Tipo di adapter.* File/SQL
LogFileMaxSize	Numero	Il valore è un numero espresso in KB.** L'intervallo consentito è 1.000-100.000 e il valore predefinito è 1.000.

* Utilizzato solo dall'utility Adapter Manager

** Utilizzato dall'adapter

Modalità di esecuzione dell'adapter

L'adapter può essere eseguito da:

Servizio:

È possibile installare l'adapter come un servizio standard di Windows. Questa opzione consente al sistema di controllare il proprio stato (In esecuzione, Sospeso, Interrotto, Automatico) come un servizio standard di Windows.

L'adapter viene installato come un servizio di Windows mediante l'esecuzione dell'eseguibile dell'adapter dalla riga di comando utilizzando **-i** per installarlo come servizio e **-u** per disinstallarlo.

Applicazione:

Esecuzione dell'eseguibile dell'adapter dalla riga di comando. La riga di comando dell'adapter può essere eseguita nei modi riportati di seguito:

Opzioni della riga di comando:

```
TextFileAdapter.exe -i | -u | -v | -d [-t] [-f configurationFileName] [-l  
logFileName] [-n serviceName]  
[-a OblicoreAddress] [-p OblicorePort] [-la LogGerheaDed] [-lp  
LogServerPort]
```

Parametro	Funzionalità
-i	Installazione del servizio
-u	Rimozione del servizio
-v	Visualizzazione della versione
-d	Esecuzione dell'adapter come applicazione di console
-t	Controllo solo del file di configurazione e interruzione
-f	Impostazione del nome del file di configurazione
-l	Impostazione del nome del file di log
-n	Impostazione del nome del servizio
-a	Impostazione dell'indirizzo del server applicazioni
-p	Impostazione del numero di porta del server applicazioni (1024-49151)
-la	Impostazione dell'indirizzo del server di registrazione dei log
-lp	Impostazione del numero di porta del server di registrazione dei log (1024-49151)

Questo tipo di esecuzione viene comunemente utilizzato nei progetti. Consente l'esecuzione dell'adapter tramite un file .bat e inoltre consente l'uso dell'utilità di pianificazione di Windows per controllare gli intervalli di esecuzione dell'adapter. Per pianificare l'adapter utilizzando l'utilità di pianificazione di Windows, è necessario configurare la modalità di esecuzione su Esegui una volta.

RunOnce: (facoltativo [sì/no]). Quando nel file di configurazione l'impostazione è "no", una volta avviata l'esecuzione l'adapter viene eseguito in modo continuo. Se l'impostazione è "yes" (sì), l'adapter per file è in esecuzione, legge i record e si interrompe automaticamente quando non è visualizzato alcun nuovo record. L'adapter per file legge l'intero file, quindi attende alcuni secondi e tenta di leggere i nuovi record (a seconda delle impostazioni SleepTime). Se non sono presenti nuovi record, si interrompe. Un adapter SQL esegue ogni query solo una volta. Se RepeatUntilDry è impostato su "no", si interrompe immediatamente. Se RepeatUntilDry è impostato su "yes" (sì), attende (a seconda di SleepTime). Tenta di eseguire nuovamente la query (in base alla durata di sospensione della query) e, se non sono presenti nuovi record, si interrompe.

Per informazioni sugli attributi SleepTime e RepeatUntilDry, fare riferimento alla sezione [Specifiche di configurazione dell'adapter](#) (a pagina 317).

La sezione Interface CA Business Service Insight del file di configurazione è costituita da due attributi che specificano le due modalità di connessione a CA Business Service Insight: in linea e non in linea.

In modalità in linea, l'adapter si connette a CA Business Service Insight, recupera le tabelle di conversione e comandi di controllo da CA Business Service Insight, quindi invia eventi, stati e richieste di conversione a CA Business Service Insight. In modalità non in linea, l'adapter utilizza il file della tabella di conversione locale e scrive gli eventi in un file di output.

La modalità non in linea viene comunemente utilizzata durante la prima fase di sviluppo e verifica dell'adapter.

L'attributo ConsoleDebugMode impostato su "yes" (sì) consente la visualizzazione dei messaggi di debug sulla console.

Per ulteriori informazioni sui diversi indicatori, fare riferimento alla sezione [Specifiche di configurazione dell'adapter](#) (a pagina 317), in particolare all'attributo ConsoleDebugMode.

Strumenti di configurazione e manutenzione

Gli strumenti necessari come parte del processo di configurazione e manutenzione degli adapter sono principalmente utilità autonome, quali le utilità CA Business Service Insight che sono semplici eseguibili di Windows.

Nota: durante la configurazione degli adapter in un ambiente UNIX, queste utilità non sono disponibili, quindi è necessario eseguire la configurazione manualmente.

Quando si utilizza un server applicazioni CA Business Service Insight, le utilità vengono installate come parte dell'installazione di applicazioni e vengono collocate nella cartella Program Files\CA\Cloud Insight\Utilities.

Come parte dell'installazione, viene creato il collegamento dal menu Avvio. È consigliabile utilizzare questo collegamento per eseguire le utilità.

Nel caso in cui non viene utilizzato il server applicazioni CA Business Service Insight, è possibile copiare queste utilità su qualsiasi computer Windows come file standard oppure installarle con il pacchetto CA Business Service Insight fornito e selezionare l'installazione personalizzata. L'installazione di routine non è obbligatoria e in genere è sufficiente copiare i file .exe in un percorso pratico sulla workstation. Mediante questa opzione, tuttavia, è possibile riscontrare alcuni file .dll mancanti che devono essere copiati, se disponibili, dal server alla cartella locale sulla workstation.

Configurazione di adapter e conversioni

Questa fase include l'esecuzione dei seguenti passaggi:

1. Configurazione dell'adapter tramite la Procedura guidata di configurazione dell'adapter o la modifica manuale del file di configurazione XML (descritta nel capitolo successivo).
2. Distribuzione dell'adapter.
3. Verifica dell'adapter.
4. Esecuzione delle conversioni.
5. Scrittura degli script di conversione per supportare il processo di conversione automatica (facoltativo).

Nota: il deployment dell'adapter può essere effettuato automaticamente durante l'utilizzo della Procedura guidata di configurazione dell'adapter quando il nuovo servizio di deployment adapter viene eseguito come un servizio in background sul server applicazioni per gestire questa attività.

Deployment di un nuovo adapter (Procedura guidata di configurazione dell'adapter)

Quando viene creato un nuovo adapter tramite la Procedura guidata di configurazione dell'adapter, verificare che i servizi del listener dell'adapter e di deployment adapter siano in esecuzione.

Deployment di un nuovo adapter (manuale)

Prerequisiti per la creazione di un nuovo adapter

Prima di iniziare la creazione di un nuovo adapter, è necessario che i seguenti elementi siano presenti:

- **Cartella principale dell'adapter:** se CA Business Service Insight è installato sul server, questa cartella esiste nella cartella principale Program Files\CA\Cloud Insight, altrimenti deve essere creata.
- **Cartella dell'adapter singolo:** creare una cartella nella cartella principale dell'adapter per l'adapter specifico.
Nota: in caso di utilizzo dell'utility Adapter Manager, l'utilità crea automaticamente la cartella durante l'aggiunta del nuovo adapter.
- **Eseguibili dell'adapter:** TextFileAdapter.exe, eseguibile dell'adapter per file di testo; SQLAdapter.exe, eseguibile dell'adapter SQL. È possibile trovarli sul server applicazioni nella cartella Program Files\CA\Cloud Insight\Adapters.
Nota: durante il processo di creazione dell'adapter tramite l'utility Adapter Manager, quando possibile, è necessario selezionare l'opzione per creare un collegamento ai file eseguibili, invece di creare una copia. In questo modo si garantisce che, in caso di applicazione di un aggiornamento o di una service release (SR) a CA Business Service Insight, tutti i componenti binari vengono aggiornati correttamente.
- **Modelli di configurazione:** modelli del file di configurazione dell'adapter. Posizionare il file nella cartella principale dell'adapter. È possibile trovarli sul server applicazioni nella cartella Program Files\CA\Cloud Insight\Adapters. I modelli di configurazione vengono utilizzati per creare il file di configurazione in modo da evitarne la creazione ex novo. È inoltre frequente l'utilizzo di un file di configurazione dell'adapter esistente per tale scopo.
- **Utility Adapter Manager:** eseguibile autonomo. È sufficiente creare una copia di AdaptersManager.exe dalla cartella Utilities alla cartella in Program Files\CA\Cloud Insight\Utilities sul server applicazioni. Non è necessario creare l'adapter tramite questa utility, che può essere utilizzata solo su un server Windows.

- **Due porte TCP/IP aperte:** una porta per AdapterListener sul server applicazioni e un'altra per LogServer. La porta LogServer è in genere 4040.
- Verificare che i componenti di servizio dell'applicazione CA Business Service Insight siano in esecuzione. Ai fini dell'esecuzione dell'adapter, è necessario che i seguenti componenti del servizio siano in esecuzione:
 - **AdapterListener**
 - **TaskHost**
 - **LogServer**

Attenersi alla seguente procedura:

1. Eseguire l'utility Adapter Manager (consultare la sezione precedente Utilità Adapter Manager).
2. Preparare tutti i prerequisiti citati e creare un file batch con la riga di comando dell'eseguibile (consultare la sezione [Modalità di esecuzione dell'adapter](#) (a pagina 85)).

Modifica del file di configurazione dell'adapter

Durante la creazione di un adapter, la maggior parte del lavoro consiste nella modifica del file di configurazione.

Questo lavoro implica l'impostazione degli attributi nel file XML per il controllo del comportamento dell'adapter in modo sia eseguito come richiesto. Il file di configurazione è un file XML con ogni sezione corrispondente a un passaggio nel suo flusso di lavoro interno.

Le sezioni sono:

- **Sezione General:** vari attributi dell'adapter, cartella di lavoro, specifiche dei file di output, contrassegni di debug, valori predefiniti, e così via.

Oblicore:

- **Sezione OblicoreInterface:** attributi di connessione con il server CA Business Service Insight (porta TCP/IP, modalità di protezione, e così via).
- **Sezione DataSourceInterface:** attributi di connessione con l'origine dati (percorso del file e modello, stringhe di connessione, query SQL, e così via).
- **Sezione InputFormatCollection:** regole di analisi per l'analisi e la modifica dei formati dei dati originali (delimitatori, tipi di campo, ordine di dati, espressioni regolari, e così via).
- **Sezione TranslatorCollection:** regole per l'evento unificato CA Business Service Insight composto dai campi di dati analizzati e modificati.
- **Sezione TranslationTableCollection:** le regole di mapping dei dati tra la terminologia dei dati originali e le entità di dati CA Business Service Insight.

Queste sezioni vengono descritte in modo dettagliato nel capitolo [Specifiche di configurazione dell'adapter](#) (a pagina 317).

Nota: l'ordine dei nodi XML in ogni sezione non è importante.

Adapter per file di testo

Gli adapter per file di testo utilizzano il componente generico (eseguibile) FileAdapter e il file di configurazione con cui analizzare i file ASCII.

Flusso di lavoro dell'adapter per file di testo:

- Copiare o rinominare il file di origine con un file di lavoro.
- Leggere il record logico.
- Analizzare il record in base ai delimitatori o alle espressioni regolari.
- Trovare il corretto inputFormat.
- Creare il record dell'evento.
- Convertire il record dell'evento.
- Aggiornare il file di controllo.

Esempio del file di configurazione

Nel seguente esempio viene utilizzata un'origine dati di file ASCII semplice con i relativi requisiti di output dell'evento e vengono riviste le impostazioni necessarie per il file di configurazione dell'adapter.

Il file di configurazione può essere visualizzato e modificato solo mediante l'utilità XMLPad.

Per una panoramica della struttura e del contenuto del file di configurazione, consultare le sezioni attinenti.

Nota: le impostazioni revisionate sono solo le impostazioni di attributo più importanti e principali. Per le specifiche complete dell'attributo, consultare la sezione [Specifiche di configurazione dell'adapter](#) (a pagina 317).

Case study dell'adapter per file

Considerare il seguente sistema di monitoraggio del server che genera file di log con le informazioni in base alla seguente struttura:

Server	Data/ora	Utilizzo memoria	Utilizzo CPU
Server001p	1.11.2004 0:00	26.71	78.85
Server002p	1.11.2004 1:00	49.09	89.49
Server001p	1.11.2004 2:00	51.37	57.51
Server001p	1.11.2004 3:00	23.53	98.9
Server002p	1.11.2004 4:00	86.21	85.11
Server002p	1.11.2004 5:00	39.98	35.97
Server001p	1.11.2004 6:00	70.58	55.96

I file vengono trasferiti alla cartella dell'adapter CA Business Service Insight in una sottocartella denominata Data (Dati). I nomi dei file contengono tutti il prefisso ServerData e il suffisso di data e ora. I file sono anche file CSV con estensione .csv.

È richiesto il seguente evento di output:

- **Campo di conversione risorsa:** Server
- **Campo di data/ora:** Timestamp
- **Campi di dati:** Memory Utilization, CPU Usage

Inoltre, si presuppone che l'origine dei dati sia in Belgio (fuso orario CET con differenza di fuso orario +1, che include periodi di ora legale con l'aggiunta di un'ora in estate).

Sezione General del file di configurazione

- **MajorVersion e MinorVersion:** devono essere impostati per impostazione predefinita sulla versione di applicazione.
- **WorkingDirectoryName (facoltativo):** specifica il percorso predefinito per i file di output dell'adapter (controllo origine dati, conversioni, invii). In questo caso, è impostato su Output, la cartella viene quindi creata nella cartella principale degli adapter.

I seguenti indicatori controllano il modo degli adapter di leggere e convertire i record:

- **RunOnce:** se impostato su Sì, l'adapter esegue una sola volta, legge tutti i dati e quindi si interrompe.
- **DataSourceDifferenceFromUTC:** indica la differenza di tempo tra l'ora UTC (Tempo universale coordinato, sempre con differenza di orario zero; equivalente a GMT) e il fuso orario dell'origine dati. Il fuso orario dell'origine dati è il fuso orario con vengono denotati i campi di tempo. Questo attributo è necessario perché l'adapter normalizza tutte le date in base all'ora UTC. Tutte le date in base all'ora UTC rende l'applicazione flessibile per visualizzare così le ore in base ai requisiti dell'utente. I seguenti attributi forniscono all'adapter le informazioni su come trasformare i campi di tempo dall'input in UTC:
 - **DefaultOffset:** differenza da UTC durante i periodi senza ora legale. In questo caso, è impostato su 1 per l'orario dell'Europa centrale (CET).
 - **TimeFormat:** formato in base al quale devono essere analizzate le informazioni sull'ora legale (descritto in seguito). Il formato ora europeo è dd/mm/yyyy hh:mi:ss, mentre le specifiche di formato CA Business Service Insight sono impostate come %d/%m/%Y %H:%M:%S.
 - **DayLightSaving:** un periodo di ora legale del fuso orario dell'origine dati. Questo elemento è facoltativo (ovvero, se non è selezionato, non è presente l'ora legale) e possono esistere più di una volta; una volta per ogni periodo di ora legale immesso. Quando sono presenti alcuni elementi, devono essere ordinati per ora e i periodi non devono sovrapporsi. In genere, durante la configurazione degli adapter, questo elemento è specificato come numero di anni avanti. In questo caso, è configurato solo un anno come esempio.
- **From:** data di inizio del periodo. Specificata utilizzando il formato ora impostato in precedenza, 25/03/2007 02:00:00.
- **To:** data di fine del periodo. Specificata utilizzando il formato ora impostato in precedenza, 28/10/2007 03:00:00.
- **Shift:** ore di differenza aggiunte a DefaultOffset nel periodo di ora legale. Digitare 1 come ore di differenza per spostare in avanti di un'ora durante questo periodo di ora legale specificato tra le due date indicate.

Sezione OblicoreInterface del file di configurazione

Questa sezione definisce la connessione al server CA Business Service Insight.

- Modalità: in modalità in linea, l'adapter si connette a CA Business Service Insight, recupera le tabelle di conversione e comandi di controllo da CA Business Service Insight, quindi invia eventi, stati e richieste di conversione a CA Business Service Insight. Configurare l'adapter in questa modalità e impostare il valore su In linea.
- Porta: il numero di porta TCP/IP utilizzata dall'adapter per comunicare con il server CA Business Service Insight (dove si trova AdapterListener).

Nel caso in cui non sorgono problemi con questa operazione, configurare l'adapter per utilizzare la porta 5555 (selezionato in modo arbitrario). Inoltre, è necessario specificare questo numero sul server nell'interfaccia utente anche per l'adapter, per consentire la comunicazione.

Sezione DataSourceInterface del file di configurazione

La sezione DataSourceInterface è costituita dagli attributi che specificano la connessione e il tipo di connessione tra l'adapter e l'origine dati. Esistono due tipi di interfaccia, file e SQL. La principale differenza tra i due è che per file è obbligatorio l'insieme dei file, mentre per SQL è obbligatorio l'insieme delle query.

La sezione DataSourceInterface consente inoltre di definire il modo in cui l'adapter gestisce il file di origine (se elimina il file originale quando è stato creato solo per l'adapter, o se mantiene i dati quando è necessario per altri usi, e così via).

Per consentire agli adapter per file di leggere e analizzare i file ASCII, viene utilizzata l'interfaccia dei file come illustrato nella seguente figura. Selezionare i seguenti valori per le impostazioni come segue:

La sezione Files nel nodo DataSource Interface fa riferimento alla connessione all'origine dati. Configurare gli attributi seguenti.

Nota: questa sezione è completamente diversa per un adapter SQL.

- **DeleteFileAfterProcessing:** imposta il modo in cui l'adapter gestisce il file di origine e determina il modo in cui l'adapter controlla la lettura in modo da leggere solo i nuovi record. In questo caso, i file di origine vengono conservati sul server e il valore viene impostato su "no".

Nel caso in cui viene creato un file solo per l'adapter e tale file può essere eliminato dopo l'elaborazione, deve essere impostato su "yes" (sì). Il file viene quindi rinominato, elaborato ed eliminato.

Quando è impostato su "no", il file viene copiato e l'elaborazione viene eseguita utilizzando il file copiato. Se i nuovi record vengono aggiunti alla fine di questo file, l'adapter copia questi nuovi record nel file di lavoro durante il successivo ciclo. Se i nuovi record non vengono aggiunti al file, l'adapter cerca il prossimo file con lo stesso modello e nome maggiore (in ordine lessicografico) del file corrente. Se l'adapter trova il file, l'elaborazione continua su questo file. L'adapter non ripristina il file precedente, anche se i nuovi record vengono aggiunti.

Impostare su "no" quando è necessario mantenere l'integrità del file di origine e quando il file deve essere aggiunto.

- **InputFormat:** indica il nome specificato per l'elemento InputFormat nel successivo InputFormatCollection che gestisce i dati dalla connessione a questa origine dati. Il formato di input è la struttura dei campi dei dati di input provenienti dall'origine dati dopo l'analisi eseguita dall'adapter. Il metodo di analisi viene specificato nell'attributo dei delimitatori come indicato di seguito. Durante la gestione di più connessioni tramite diversi formati di interfaccia, questo campo diventa più importante e determina quale struttura del formato di input gestisce i dati di ciascuna origine dati.
- **Path:** ubicazione fisica dei file di origine dati. Ad esempio: C:\Program Files\CA\Cloud Insight\Adapters\ServersAdapter\data\.

- **NamePattern:** indica il nome del file di origine dati. È possibile utilizzare caratteri jolly se più file utilizzano lo stesso formato di input. Se un nome di file viene specificato senza caratteri jolly, l'adapter ricerca solo il file con questo nome. Con l'utilizzo dei caratteri jolly, l'adapter cerca tutti i file corrispondenti al modello, li mette in ordine lessicografico, quindi li legge uno a uno. Durante la prossima esecuzione, cerca i nuovi record nell'ultimo file prima di passare al successivo.

In questo esempio, se viene utilizzato il carattere jolly *, il valore di attributo è `ServerData*.csv`. (L'adapter legge tutti i file con nomi che iniziano con `ServerData` e con estensione `.csv`.)

Importante: È consigliabile aggiungere la data e l'ora alla fine dei nomi di file utilizzando il formato seguente `YYYYMMDD-HHMISS` per assicurarsi che i file siano ordinati correttamente, letti nell'ordine corretto e che nessuno dei file sia escluso. È possibile inoltre aggiungere la porzione di tempo se sono presenti più file generati ogni giorno.

- **Delimiters:** metodo di determinazione della modalità di analisi del file. È possibile specificare uno o più caratteri come delimitatori, in base alle righe di dati da analizzare nei campi. Se non specificato, il valore predefinito è `/t`.

Il file dell'origine dati in questo esempio è un file CSV (separato da virgola). Il modo più semplice di analisi di tali file è specificare la virgola come delimitatore.

Altri metodi disponibili per l'analisi sono:

- **RegExForParser:** utilizza un'espressione regolare per impostare la regola di analisi.
- **LogicLineDefinition:** utilizzato quando una riga nel file è costituita da più righe.
- **TitleExists** (facoltativo): indica se la prima riga non vuota nel file di origine dati è una riga di titolo. Il titolo può essere utilizzato dall'adapter durante l'analisi dei dati. In questo esempio, ogni file di dati contiene la riga di titolo, pertanto è necessario specificare "yes" (sì) per questo attributo.

Sezione InputFormatCollection del file di configurazione

Questa sezione consente di specificare la struttura dei dati recuperati dall'origine dati, come dividere una riga di dati in campi e quali sono i tipi di campo e i formati. In questa sezione è possibile eseguire il filtraggio dei dati iniziali e modifiche sui dati utilizzando rispettivamente i campi InputFormatSwitch e Compound.

In questa sezione è possibile definire le metriche di convalida per il record di input tramite InputFormatValidation e ValidationCase, che determinano la validità o meno di un record.

Il nodo InputFormatCollection può contenere uno o più nodi InputFormat.

Il flusso di lavoro generale di questa sezione seguito dall'adapter è:

- Viene eseguita la corrispondenza della riga di dati con InputFormat specificato in DataSourceInterface.
- I dati vengono divisi in campi in base alla specifica InputFormat corrispondente. L'ordine di InputFormatFields deve corrispondere con l'ordine dei campi analizzati dall'origine dati.
- I campi composti sono valori assegnati con la combinazione e suddivisione dei campi di dati specificati in precedenza. Le definizioni del campo composto devono seguire la definizione del campo normale.
- I dati elaborati vengono confrontati con le condizioni TranslatorSwitch per determinare quale convertitore viene utilizzato per generare l'evento di output al di fuori del record di input.
- I dati elaborati vengono inviati al convertitore corrispondente oppure ignorati.

Utilizzare i seguenti parametri:

- **InputFormatName (Nome formato di input):** nome per questo formato, segnalato dalla sezione DataSourceInterface.
- **InputFormatFields (Campi formato di input):** può contenere uno o più nodi di campo in base al numero dell'origine dati dei campi di input.
- **InputFormatField (Campo formato di input):** indica un campo dati della riga di dati originali o un campo composto.

```
<InputFormatField Name="timestamp" Type="time"
```

```
TimeFormat="%d/%m/%Y %H:%M:%S"/>
```

- **Name (Nome):** nome per questo campo, cui fanno riferimento altri elementi, ad esempio l'elemento composto o TranslatorFields come campo di origine.
- **Type (Tipo):** tipo di dati del campo (stringa, numero intero, reale, ora).
- **Source (Origine):** valore di origine per questo campo; il valore predefinito utilizzato è "event", mentre i possibili valori sono:

- **event (evento):** il valore di campo è ottenuto dall'evento proveniente dall'origine dati. I valori di campi vengono acquisiti nello stesso ordine di provenienza dall'origine dati.
- **compound (composto):** il valore di campo è generato da altri campi, in base alla modifica dei valori o delle costanti di altri campi. I campi modificati devono essere già definiti.
- **title (titolo):** il valore di campo è ottenuto dai nomi di campo titolo. Il campo di riferimento deve essere già definito.
- **filename (nome file):** il valore di campo è ottenuto dal nome del file dell'origine dati; solo per adapter per file di testo.
- **constant (costante):** il valore di campo è costante e viene acquisito da ConstantValue la cui proprietà deve essere visualizzata accanto.
- **TranslatorSwitch (Switch convertitore):** determina quale convertitore viene utilizzato per convertire la riga di dati in un evento unificato.
 - **DefaultTranslator (Convertitore predefinito):** convertitore da utilizzare nel caso in cui nessun criterio può essere soddisfatto. Se il valore è impostato su "Ignore" (Ignora), nessun convertitore viene utilizzato e la riga viene ignorata e non inviata a CA Business Service Insight.

Sezione TranslatorCollection del file di configurazione

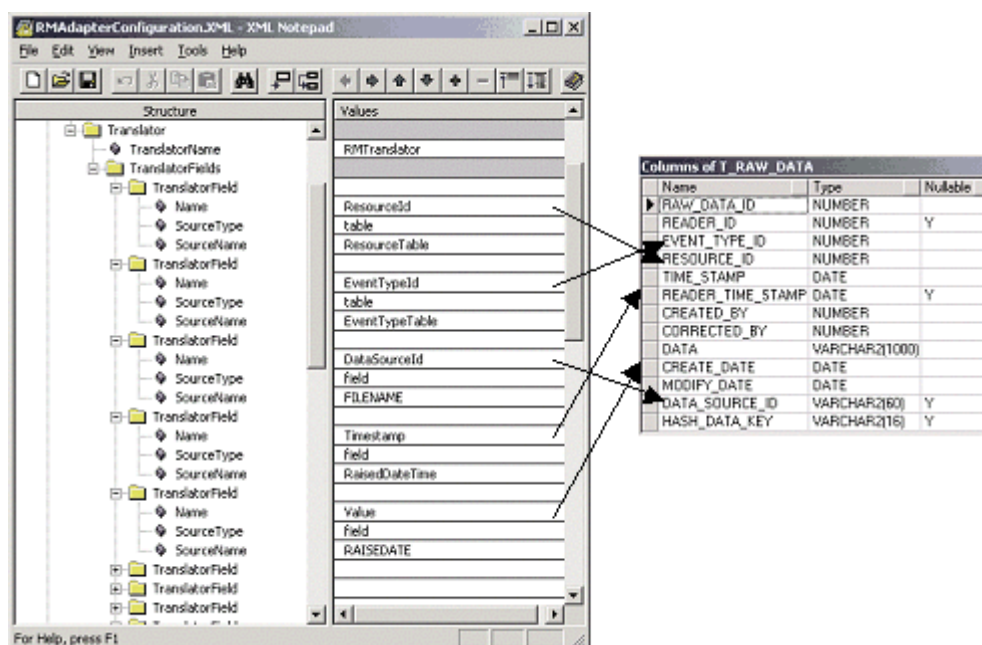
La sezione TranslatorCollection definisce come i record dell'origine dati analizzati e modificati, estratti nelle sezioni precedenti, verranno convertiti in un evento CA Business Service Insight.

Questa sezione definisce inoltre le modalità di gestione degli eventi duplicati e di utilizzo del meccanismo di univocità dell'evento (per ulteriori informazioni, consultare il capitolo [Univocità dell'evento](#) (a pagina 127)).

Quando la modalità dell'interfaccia è impostata su in linea, l'evento CA Business Service Insight dispone di una struttura unificata che contiene i seguenti campi:

- **Timestamp**: ora in cui si è verificato l'evento.
- **ResourceId**: ID risorsa associato all'evento (la risorsa misurata in tale evento).
- **EventTypeId**: tipo di evento associato all'evento, descrive il tipo di evento (tipo di misurazione della risorsa, tipo di azione del ticket, e così via).
- **DataSourceID** (facoltativo): qualsiasi valore. Fornisce ulteriori criteri di filtro per gli eventi di dati non elaborati.
- **Value** (multipli): valori dell'evento (risultato della misurazione, numero di ticket, e così via). Questo campo spesso viene visualizzato più di una volta.

La struttura del convertitore corrisponde alla struttura del tipo di evento all'interno di CA Business Service Insight, e anche alla tabella di database T_RAW_DATA in cui è archiviato l'evento, come illustrato nella figura seguente:



- **Translator:** descrive come convertire il set di campi ricevuti nell'evento di output.
- **TranslatorName:** nome utilizzato da InputFormat per inviare i gruppi di campo a tale convertitore. In questo esempio viene utilizzato il nome Translator1. Consultare la figura precedente per i valori che possono essere utilizzati in questo scenario.
- **TranslatorFields:** contiene un elenco di elementi TranslatorField, ognuno dei quali contiene i seguenti attributi:
 - **Name:** nome di campo. Nell'interfaccia in linea deve essere Timestamp, ResourceId, EventTypeId, ResourceId o Value.
 - **SourceType:** specifica l'origine del valore di campo. Può trattarsi di uno dei seguenti tipi:
 - **Field:** il valore di questo campo è ottenuto dal campo nel formato di input. L'attributo SourceName contiene il nome del campo InputFormat.
 - **Table:** il valore del campo è ottenuto dalla tabella di conversione. L'attributo SourceName contiene il nome della tabella di conversione che fa riferimento a un nome definito nella sezione successiva di TranslationTableCollection. Questo tipo viene utilizzato per i valori selezionati per la conversione in risorse e tipi di evento nell'evento.
 - **Lookup:** il valore del campo è ottenuto dalla tabella di conversione. L'attributo SourceName contiene il nome della tabella. Il valore da convertire dall'attributo LookupValue e non dal formato di input. Solitamente viene utilizzato quando è necessario un valore costante ai fini delle conversioni.
 - **Constant:** il valore di campo è costante e il relativo valore è nell'attributo ConstantValue. Quando si utilizza un valore costante, è necessario specificare il tipo di campo utilizzando l'attributo Type.
 - **SourceName:** contiene il nome del campo o della tabella di conversione.
 - **LookupValue:** contiene il valore di ricerca quando SourceType="lookup".
 - **ConstantValue:** contiene il valore costante quando SourceType=constant. Quando il campo è di tipo ora, il valore costante è una stringa formattata in base a TimeFormat (impostato nella sezione General dell'adapter) oppure "Now" o "NowUtc", in cui "Now" è l'ora corrente nelle impostazioni internazionali dell'origine dati e "NowUtc" è l'ora corrente in UTC.

Questa sezione contiene inoltre le tabelle di mapping che definiscono il mapping dei valori di origine dati nei campi Evento di CA Business Service Insight e contiene la definizione della tabella con il valore dell'origine dati di riferimento da convertire.

- **LoadingMode:** il valore predefinito per l'interfaccia in linea è "remote" (remota), mentre per l'interfaccia non in linea è "standalone" (autonoma).

Di seguito il metodo di caricamento delle tabelle di conversione specificato:

- **Standalone:** l'adapter carica le tabelle di conversione in locale. Non è presente alcuna connessione al server CA Business Service Insight per la conversione. Le modifiche nelle tabelle di conversione vengono archiviate solo nel file locale.

- **Remote:** l'adapter invia una richiesta di caricamento di tutte le tabelle dal server CA Business Service Insight. Le modifiche apportate nelle tabelle di conversione vengono memorizzate anche localmente.
- **TranslationTable:** collega il valore dell'evento alla tabella di mapping.
 - **Name:** nome della tabella di conversione utilizzato e indicato dal convertitore.
 - **DestinationType:** [resource, event_type, contract_party, service, time_zone, value]. Contiene il tipo di tabella di conversione. In questo esempio, la colonna **Server** nel file di origine dati viene convertita in una risorsa CA Business Service Insight. Pertanto, il tipo di tabella di conversione è Risorsa, e contiene le conversioni dei valori in ID risorsa in CA Business Service Insight.
 - **TranslationField:** nome di campo da cui eseguire la conversione e che è ottenuto dai campi del formato di input. Può contenere al massimo cinque campi.

Ciascuna tabella di conversione definita nel file di configurazione deve avere una definizione corrispondente nell'interfaccia utente CA Business Service Insight.

Di seguito viene riportata la rappresentazione XML di un file di configurazione di esempio:

```
<?xml version="1.0" encoding="utf-8"?>
<AdapterConfiguration>
  <General MajorVersion="4" MinorVersion="0" RunOnce="yes" LogDebugMode="yes"
    ConsoleDebugMode="yes" WorkingDirectoryName="output"
    RejectedEventsUpperLimit="10000">
    <DataSourceDifferenceFromUTC DefaultOffset="0" TimeFormat="%d/%m/%Y %H:%M">
      <DaylightSaving From="20/04/2001 00:00" To="15/10/2001 00:00"
        Shift="1"/>
    </DataSourceDifferenceFromUTC>
  </General>
  <OblicoreInterface Mode="online">
    <OnlineInterface Port="5555" SecurityLevel="none"/>
  </OblicoreInterface>
  <DataSourceInterface>
    <Files>
      <File DeleteFileAfterProcessing="no" InputFormat="InputFormat1"
        NamePattern="servers*.csv" Path=" C:\Program
        Files\Oblicore\Adapters\ServersAdapter\data\" TitleExists="yes" SleepTime="60"
        Delimiters=","/>
    </Files>
  </DataSourceInterface>
  <InputFormatCollection>
    <InputFormat InputFormatName="InputFormat1">
      <InputFormatFields>
        <InputFormatField Name="resource" Type="string"/>
        <InputFormatField Name="timestamp" Type="time"
          TimeFormat="%d.%m.%Y %H:%M"/>
        <InputFormatField Name="memory_util" Type="real"/>
      </InputFormatFields>
    </InputFormat>
  </InputFormatCollection>
</AdapterConfiguration>
```

```
        <InputFormatField Name="cpu_util" Type="real"/>
    </InputFormatFields>
    <TranslatorSwitch DefaultTranslator="Translator1"/>
</InputFormat>
</InputFormatCollection>
<TranslatorCollection>
<Translator TranslatorName="Translator">
    <TranslatorFields>
        <TranslatorField Name="ResourceId" SourceType="table"
SourceName="ResourceTable"/>
        <TranslatorField Name="EventTypeId" SourceType="lookup"
SourceName="EventTable" LookupValue="PerformanceEvent"/>
        <TranslatorField Name="Timestamp" SourceType="field"
SourceName="timestamp"/>
        <TranslatorField Name="Value" SourceType="field"
SourceName="memory_util"/>
        <TranslatorField Name="Value" SourceType="field"
SourceName="cpu_util"/>
    </TranslatorFields>
    </Translator>
</TranslatorCollection>
<TranslationTableCollection LoadingMode="remote">
    <TranslationTable Name="ResourceTable" DestinationType="resource">
        <TranslationField>resource</TranslationField>
    </TranslationTable>
    <TranslationTable Name="EventTable" DestinationType="event_type">
        <TranslationField>resource</TranslationField>
    </TranslationTable>
</TranslationTableCollection>
</AdapterConfiguration>
```

Adapter SQL

Gli adapter SQL utilizzano praticamente il componente generico di adapter SQL (eseguibile dell'adapter SQL) con le impostazioni appropriate nel file di configurazione. L'adapter SQL può connettersi a tutte le origini dati che supportano Open Database Connectivity (ODBC) e Object Linking and Embedding Database (OLEDB). La connessione viene stabilita tramite la stringa di connessione. Il driver appropriato deve essere installato sul server dove è installato l'adapter.

Esempi di origini dati:

- Oracle
- SQL Server
- Accesso
- Excel
- File di testo, file CSV (questi possono essere inoltre collegati tramite l'adapter TEXT, ma la connessione ODBC spesso fornisce ulteriori funzionalità di filtro/query).

Flusso di lavoro dell'adapter SQL:

- Aprire la connessione.
- Sostituire le variabili locali con gli ultimi valori di campo chiave.
- Eseguire il completamento automatico, creare le clausole where delle query, quindi concatenare alla fine delle query.

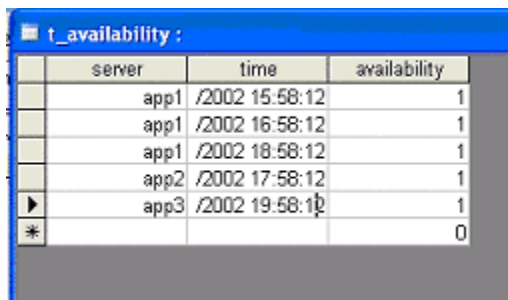
Eseguire la query e ottenere recordset.

Per ogni record nel recordset restituito dalla query:

- Trovare il corretto InputFormat.
- Creare il record dell'evento.
- Convertire il record.
- Aggiornare il valore dei campi chiave nel file di controllo.
- Chiudere la connessione.
- Sospendere.

Esempio di file di configurazione dell'adapter SQL

Specificato un database di Microsoft Access (.mdb) con la seguente tabella:



	server	time	availability
	app1	/2002 15:58:12	1
	app1	/2002 16:58:12	1
	app1	/2002 18:58:12	1
	app2	/2002 17:58:12	1
	app3	/2002 19:58:12	1
*			0

L'unica differenza tra il file di configurazione dell'adapter SQL e il file di configurazione dell'adapter per file è la sezione DataSourceInterface.

La sezione DataSourceInterface in un adapter per file memorizza l'insieme dei file, mentre il file dell'adapter SQL presenta gli elementi ConnectionString e QueryCollection.

La principale differenza tra i due file di configurazione è il metodo di recupero e di analisi. Il resto del file è identico.

L'interfaccia SQL definisce la connessione al database e le query utilizzate per recuperare i dati.

Di seguito sono riportati i dettagli:

Nota: la sezione è definita secondo il database dell'origine dati precedente.

Elemento **ConnectionString**

ConnectionString: stringa di connessione per l'accesso al database di origine.

ConnectionString può essere definito nell'elemento DataSourceInterface e/o negli elementi Query. La definizione ConnectionString nell'elemento DataSourceInterface è l'impostazione predefinita e viene utilizzata solo in una query senza la definizione ConnectionString. In questo modo all'adapter viene consentita la connessione a più database in cui ogni query può disporre della propria stringa di connessione. Per ulteriori informazioni sul meccanismo della stringa di connessione, consultare la sezione seguente.

Sezione **QueryCollection** del file di configurazione

Query: indica gli attributi della query.

- **QueryName:** nome della query.
- **InputFormat:** InputFormat associato alla query. L'adapter utilizza InputFormat per estrarre i dati dall'origine.

Nota: l'ordine dei campi InputFormat deve corrispondere all'ordine dei campi selezionati di query.

- **SleepTime:** tempo in secondi durante il quale l'adapter è inattivo in attesa di nuovi dati.
- **SelectStatement:** include l'istruzione selezionata da eseguire sul database di origine.

Nota: è necessario immettere i campi chiave della query come primi valori selezionati nell'istruzione.

- **AutoCompleteQuery:** quando è impostato su "yes", l'adapter concatena automaticamente un'istruzione WHERE alla query specificata come segue:
 - Creazione di un'istruzione WHERE che recupera solo i nuovi valori in base ai campi chiave.
 - Ordinamento dell'istruzione dei risultati in base ai campi chiave.
- **QueryKeyFields:** definisce i campi per avviare il prossimo recupero dei dati.
 - **KeyField:**
 - **Name:** il nome del campo ottenuto dai campi della query.
 - **Sort:** ordinamento dei dati (crescente/decrescente).
 - **ValueLeftWrapper:** concatena i caratteri prima del valore del campo. Il valore predefinito è apostrofo (').

- **ValueRightWrapper:** concatena i caratteri dopo il valore del campo. Il valore predefinito è apostrofo (').
- **Critical:** interrompe l'esecuzione delle altre query se questa query specifica non riesce.
- **SleepTime:** durata della sospensione tra le operazioni necessarie. Il valore predefinito è apostrofo (').

Nota: spesso i campi di data richiedono l'utilizzo di caratteri speciali per racchiudere la data stessa. I caratteri necessari per identificare il campo come un campo di data dipendono dall'origine dati. Il carattere #, come mostrato nella figura all'inizio della sezione, può essere utilizzato per racchiudere il campo valore in Excel. Un'altra origine dati, tuttavia, può richiedere metodi diversi. Per ulteriori informazioni, fare riferimento alla sezione [Specifiche di configurazione dell'adapter](#) (a pagina 317).

- **SelectInitialValues:** istruzione SELECT che fornisce i valori iniziali dei campi chiave per la prima istruzione WHERE (quando il file di controllo è vuoto).

Esempio di query (ODBC basato su Excel) creata con AutoCompleteQuery="yes":

```
SELECT INC_CLOSE_DATE, INCIDENT_REF, Severity, `Resolve Time`, `Date Logged`,
`Date Resolved` FROM `AllCalls$` WHERE INC_CLOSE_DATE>CDate('7/03/2005
13:06:21') order by INC_CLOSE_DATE
```

L'istruzione SELECT deve essere eseguibile sul database di destinazione in cui è in esecuzione la query. Potrebbero esserci differenze tra le origini e il driver ODBC utilizzato per la connessione. Ad esempio, in Oracle è possibile selezionare i valori dalla tabella speciale DUAL (select 'aaa', '1-jan-1970' da DUAL), ma in Excel è possibile solo selezionare i valori direttamente senza una tabella. (select 'aaa')

Di seguito viene riportato il layout completo di un file di configurazione XML:

```
<?xml version="1.0" encoding="utf-8"?>
<AdapterConfiguration>
  <General MajorVersion="3" MinorVersion="0" RunOnce="yes" LogDebugMode="yes"
ConsoleDebugMode="yes" WorkingDirectoryName="d:\Oblicore\Training
Kit\Exercises\Adapters\SQL Adapters\Ex1\Solution">
    <DataSourceDifferenceFromUTC DefaultOffset="1" TimeFormat="%Y/%m/%d-
%H:%M:%S"/>
  </General>
  <OblicoreInterface Mode="online">
    <OnlineInterface Port="2000" SecurityLevel="none"/>
  </OblicoreInterface>
  <DataSourceInterface>
    <ConnectionString>

      Driver={Microsoft Access Driver (*.mdb)};Dbq=d:\Oblicore\Training
Kit\Exercises\Adapters\SQL Adapters\Ex1\db1.mdb;

    </ConnectionString>
  </DataSourceInterface>
  <QueryCollection>
    <Query QueryName="Query" InputFormat="InputFormat" SleepTime="5">
      <SelectStatement AutoCompleteQuery="yes">
```

```

        select time,server,availability from t_availability

    </SelectStatement>
    <QueryKeyFields>
        <KeyField Name="time" Sort="asc" ValueLeftWrapper="#"
ValueRightWrapper="#" />
        <KeyField Name="server" Sort="asc" />
    <SelectInitialValues>

        select 'AAA', '1/1/1970'

    </SelectInitialValues>
    </QueryKeyFields>
</Query>
</QueryCollection>
</DataSourceInterface>
<InputFormatCollection>
    <InputFormat InputFormatName="InputFormat">
        <InputFormatFields>
            <InputFormatField Name="timestamp" Type="time" TimeFormat="%m/%d/%Y
%i:%M:%S %p" />
            <InputFormatField Name="server" Type="string" />
            <InputFormatField Name="status" Type="integer" />
        </InputFormatFields>
        <TranslatorSwitch DefaultTranslator="Translator" />
    </InputFormat>
</InputFormatCollection>
<TranslatorCollection>
    <Translator TranslatorName="Translator">
        <TranslatorFields>
            <TranslatorField Name="ResourceId" SourceType="table"
SourceName="ResourceTable" />
            <TranslatorField Name="EventTypeId" SourceType="constant"
ConstantValue="1500" />
            <TranslatorField Name="Timestamp" SourceType="field"
SourceName="timestamp" />
            <TranslatorField Name="Value" SourceType="field"
SourceName="status" />
        </TranslatorFields>
    </Translator>
</TranslatorCollection>
<TranslationTableCollection LoadingMode="remote">
    <TranslationTable Name="ResourceTable" DestinationType="resource">
        <TranslationField>server</TranslationField>
    </TranslationTable>
</TranslationTableCollection>
</AdapterConfiguration>

```

Stringa di connessione dell'adapter SQL

La stringa di connessione dell'adapter SQL è un meccanismo progettato per fornire le seguenti caratteristiche funzionali:

- Utilizzare diverse stringhe di connessione nello stesso adapter.
- Utilizzare un modello di stringa di connessione, in cui è possibile modificare il nome del file senza modificare il file di configurazione.
- Eliminare l'origine dati del file quando l'adapter ha terminato la lettura.
- Spostare l'origine dati del file in un'altra posizione quando l'adapter ha terminato la lettura.

Oltre alla semplice definizione della stringa di connessione come stringa nell'elemento `ConnectionString`, la stringa di connessione può essere definita da segmenti, in cui ogni segmento contiene i valori specifici concatenati alla stringa di connessione. Questo consente all'adapter di generare la stringa di connessione in modo dinamico.

Esistono due tipi di segmento. Il primo è di tipo testo e contiene il testo che viene concatenato alla stringa di connessione così com'è. Il secondo è un file e contiene un nome di file con o senza caratteri jolly. Il segmento di file può essere visualizzato una sola volta. Contiene altri attributi che definiscono le operazioni da eseguire nel file di lettura.

`ConnectionString` può essere definito nell'elemento `QueryCollection` e/o negli elementi `Query`. La definizione `ConnectionString` nell'elemento `QueryCollection` è l'impostazione predefinita e viene utilizzata solo in una query senza la definizione esplicita `ConnectionString`.

Elementi e attributi

Elemento `ConnectionString`

Questo elemento può contenere gli elementi figlio del segmento. Se contiene almeno un elemento di segmento, la stringa di connessione è concatenata a questo. In caso contrario, la stringa di connessione viene acquisita dal testo dell'elemento `ConnectionString` (come nella versione corrente).

Elemento `Segment`

Questo elemento contiene un attributo obbligatorio denominato *Type*, che presenta due valori validi: testo e file. Quando *Type*="file", l'elemento `Segment` deve contenere almeno un elemento `File`. Ogni elemento `File` viene considerato una query diversa.

Elemento `File`

Questo elemento contiene gli attributi che definiscono quali file è necessario utilizzare nella stringa di connessione e quali operazioni è necessario eseguire con il file quando l'adapter ha terminato la lettura.

- **Path**: definisce il percorso del file (directory).
- **NamePattern**: definisce il nome di file con il percorso specificato. Può contenere caratteri jolly.
- **UsePath**: valori validi: sì/no. Il valore predefinito è "yes" (sì). Se impostato su "yes" (sì), il percorso del file viene concatenato alla stringa di connessione.
- **UseFileName**: valori validi: sì/no. Il valore predefinito è "yes" (sì). Se impostato su "yes" (sì), il nome del file viene concatenato alla stringa di connessione (come richiesto per i file di Excel).
- **UpdateSchemaFile**: valori validi: sì/no. Il valore predefinito è "no". Se impostato su "yes" (sì), l'adapter aggiorna il file schema.ini con il nome del file corrente.

Nota: utilizzare questo attributo solo quando si desidera che l'adapter modifichi il file schema.ini.

Variabili interne

Sono state aggiunte due ulteriori variabili interne che possono essere utilizzate negli elementi `SelectStatement` e `SelectInitialValues`. Tali variabili sono:

- `:file_name`: sostituita con il nome e l'estensione del file corrente.
- `:file_name_no_ext`: sostituita dal nome del file corrente senza l'estensione.

Esempi

Esempio 1: stringa di connessione semplice per Oracle:

```
<ConnectionString> Provider=msdsql;dsn=0; uid=0; pwd=0 </ConnectionString>
```

or

```
<ConnectionString>
  <Segment Type="text" Text="Provider=msdasql;" />
  <Segment Type="text" Text="dsn=0; " />
  <Segment Type="text" Text="uid=0; " />
  <Segment Type="text" Text="pwd=0; " />
</ConnectionString>
```

Esempio 2: stringa di connessione semplice per Excel con un file singolo:

```
<ConnectionString>Driver={Microsoft Excel Driver (*.xls)}; DriverId=790;
                    Dbq=d:\Oblicore\Availabilty_2003_01.XLS
</ConnectionString>
```

or

```
<ConnectionString>
  <Segment Type="text" Text=" Driver={Microsoft Excel Driver (*.xls)};" />
  <Segment Type="text" Text=" DriverId=790;" />
  <Segment Type="text" Text=" Dbq=" />
  <Segment Type="File">
    <File Path="d:\Oblicore " NamePattern="Availabilty_2003_01.XLS">
  </Segment>
</ConnectionString>
```

Esempio 3: stringa di connessione semplice per l'utilizzo con più file di Excel:

```
<ConnectionString>
  <Segment Type="text" Text=" Driver={Microsoft Excel Driver (*.xls)};" />
  <Segment Type="text" Text=" DriverId=790;" />
  <Segment Type="text" Text=" Dbq=" />
  <Segment Type="File">
    <File Path="d:\Oblicore ",NamePattern="Availabilty_*.XLS" />
  </Segment>
</ConnectionString>
```

Esempio 4: utilizzo di una voce DSN ODBC standard:

Utilizzando la voce DSN ODBC standard è possibile connettersi a qualsiasi origine con una voce DSN creata nella gestione ODBC sul server applicazioni. La voce DSN ODBC standard può essere trovata nella sezione degli strumenti di amministrazione del server.

```
<ConnectionString>dsn=SampleDataSource;usr=scott;pwd=tiger;</ConnectionString>
```

Lettura dei record dall'origine dati dei file

Se è presente un'interfaccia ODBC per l'origine dati, è possibile configurare un adapter SQL per eseguire query nei file. Per configurare l'adapter alla lettura da file multipli, è necessario utilizzare gli elementi Segment nell'elemento ConnectionString. Per un esempio, consultare la sezione precedente che descrive la stringa di connessione.

Di seguito viene riportato come l'adapter SQL utilizza i file:

1. In ogni query, l'adapter tenta di leggere il file finché è impossibile recuperare ulteriori record.
2. L'adapter quindi si sposta sul file successivo di cui tenta di eseguire la lettura.
3. Quando non sono presenti altri file, l'adapter sospende l'esecuzione della query secondo l'attributo SleepTime attributo.

File schema.ini

Quando viene utilizzato un driver ODBC di testo, il formato del file di testo è determinato dal file delle informazioni di schema (schema.ini). Questo file delle informazioni di schema dovrebbe trovarsi nella stessa directory dell'origine dati del testo.

I file schema.ini sono costituiti da voci, con ogni voce che descrive la struttura di un singolo file di testo. La prima riga di ogni voce è il nome del file di origine di testo, racchiuso tra parentesi quadre.

Quando l'adapter utilizza i file definiti con caratteri jolly, il nome del file viene modificato continuamente. Poiché il nome del file schema.ini non può contenere caratteri jolly, l'adapter, deve modificare il file schema.ini prima di stabilire la connessione al database.

È pertanto necessario aggiungere una riga di indicazione prima della riga della voce del file. Questa riga di indicazione contiene il modello del nome dall'elemento della stringa di connessione nel file di configurazione dell'adapter, racchiuso tra parentesi quadre. L'adapter sostituisce la riga successiva con il nuovo nome del file racchiuso tra parentesi quadre.

Nota: il file schema.ini può contenere più voci. Spetta all'utente aggiungere le righe nella posizione corretta.

Esempio di schema.ini

```
[sqltes*.txt]
[sqltest2.txt]
ColNameHeader = False
CharacterSet = ANSI
Format = TabDelimited
Coll=id Char Width 30
```

```
Col2=idname Char Width 30
-----
[report_200*.txt]
[report_2003_09.txt]
ColNameHeader = False
CharacterSet = ANSI
Format = TabDelimited
Col1=date Char Width 30
Col2=service Char Width 30
Col3=responseTime Char Width 30
-----
```

File DataSourceControl

Per ogni query, il file di controllo dell'origine dati contiene il modello del nome di file e il nome del file di input corrente per continuare con lo stesso file durante il riavvio dell'adapter.

Di seguito è riportato un esempio di file di controllo in formato XML:

```
<AdapterControl Save="end" LastSaveTime="2005/03/23 09:16:15">
  <Data>
    <QueryCollection>
      <Query QueryName="TroubleTickets (on D:\Data\Incidents*.xls)">
        <KeyField Name="INC_CLOSE_DATE">
          <LastValue>7/03/2005 13:06:21</LastValue>
        </KeyField>
        <LastFileName>IncidentsMarch.xls</LastFileName>
      </Query>
    </QueryCollection>
  </Data>
```

Conservazione dei file di input

Quando l'adapter ha terminato la lettura del file corrente, cerca il file successivo. Il file successivo letto è il primo file che soddisfa il modello del nome e il cui nome è maggiore (in ordine lessicografico) rispetto al nome del file precedente. L'adapter non torna ai file precedenti anche se vengono aggiunti i nuovi record.

L'adapter utilizza l'attributo InitialFileName solo quando questi due attributi sono uguali a "no" e il file di controllo non contiene il nome dell'ultimo file.

Verifica delle query

L'adapter verifica la stringa di connessione e la query solo se esiste il file definito. Se è stato definito utilizzando i caratteri jolly, l'adapter esegue la verifica solo nel primo file.

Possono verificarsi errori quando l'adapter tenta di leggere un nuovo file. In questo caso, l'adapter si interrompe immediatamente se l'attributo critico è uguale a "yes" (sì). Se è uguale a "no", l'adapter non continua a eseguire questa query, ma continua con le altre query. In alternativa, l'adapter lascia il file corrente e si sposta nel file successivo.

Variabili interne dell'adapter

Sono presenti due variabili interne che possono essere utilizzate nel file di configurazione di riferimento per il contesto corrente del nome di file. Queste sono :file_name e :file_name_no_ext, che fanno riferimento, rispettivamente, al nome del file corrente e al nome di file corrente senza l'estensione di file.

Queste variabili possono essere utilizzate nell'elemento SelectStatement, nell'elemento SelectInitialValues e nell'attributo QueryKeyField\Function.

L'adapter sostituisce la variabile con il nome del file nelle query.

Ad esempio:

- select date, service, value from ":filename"
- select id and name from :file_name_no_ext

Creazione di un adapter tramite l'interfaccia utente CA Business Service Insight

Ogni adapter configurato per l'esecuzione nell'ambiente CA Business Service Insight deve essere registrato nell'interfaccia utente oltre a essere definito nel registro di sistema. La ragione di questo passaggio è principalmente stabilire le impostazioni del listener dell'adapter in modo che possa ricevere eventi dall'adapter. Durante questo passaggio vengono definite tutte le impostazioni dell'adapter, quali le tabelle di conversione e i tipi di evento.

Attenersi alla seguente procedura:

1. Creare l'adapter.
2. Selezionare Risorse/Adapter.
3. Controllare gli adapter esistenti nell'elenco per assicurarsi che nessuno sia definito sulla stessa porta del proprio l'adapter, ossia, la stessa porta definita nel file di configurazione dell'adapter nella directory OblicoreInterface\OnlineInterface\Port.
4. Fare clic su Aggiungi nuovo, quindi selezionare il metodo da utilizzare per la creazione dell'adapter. Esistono varie possibilità:
 - a. **Creazione manuale:** consente di configurare il listener dell'adapter per connettersi all'adapter definito (o da definire) manualmente.
 - b. **Tramite la procedura guidata:** consente la creazione dell'adapter tramite le schermate successive dell'interfaccia della procedura guidata. Consultare la sezione successiva per ulteriori informazioni su questo metodo.
 - c. **Da un modello**
 - d. **Da un file di configurazione esistente:** consente di caricare un modello dell'adapter preconfigurato in grado di compilare automaticamente i campi della procedura guidata di configurazione dell'adapter con le opzioni impostate nel file di configurazione specificato.
 - e. La schermata restituita varia in base all'opzione selezionata.
5. Compilare i campi:

Indirizzo di rete, per immettere l'indirizzo IP dell'adapter. localhost nel caso sia locale nel server applicazioni; in caso contrario, immettere la porta definita.
6. Fare clic su Salva.

Per creare le tabelle di conversione necessarie:

Nota: tali passaggi devono essere eseguiti per ogni tabella di conversione definita nel file di configurazione:

1. Nel menu Progettazione fare clic su Conversione, quindi su Tabelle di conversione e fare clic sul pulsante Aggiungi nuovo.
2. Tutte le impostazioni della tabella di conversione devono corrispondere alla definizione della tabella equivalente nel file di configurazione dell'adapter:

- *Nome*: deve corrispondere all'attributo Name nel nome della tabella di conversione definito nel file di configurazione.
- *Campi di origine*: deve avere tutti i campi da TranslationField della tabella di conversione. Aggiungere tutti i campi. Può essere presente una combinazione di due o più campi che compongono il valore di conversione. Di seguito è riportato un esempio della possibile visualizzazione:



3. Tipo di destinazione: deve nuovamente corrispondere all'attributo DestinationType della definizione della tabella di conversione nel file di configurazione. (resource, event_type, e così via).
 4. Adapter registrati: aggiungere gli adapter che devono utilizzare questa tabella di conversione. Più adapter possono utilizzare una singola tabella di conversione.
 5. Fare clic su Salva.
 6. Importare la definizione dei campi Tipo di evento per ciascun tipo di evento.
Per essere in grado di importare la definizione dei campi dal file di configurazione di un adapter specifico, l'adapter deve essere eseguito e collegato a CA Business Service Insight almeno una volta. Quando l'adapter si connette a CA Business Service Insight, invia il file di configurazione per CA Business Service Insight che consente al sistema di utilizzare la definizione dei campi contenuta.
- Nota:** in alternativa, è possibile specificare le definizioni di campo manualmente per il tipo di evento.
7. Attivare l'adapter:
 - a. Nel menu Progettazione, fare clic su Acquisizione dati, quindi su Adapter.
 - b. Fare clic sul pulsante Avvia adapter.

Nella seguente tabella vengono riportati i diversi stati degli adapter:

Stato	Descrizione
Non attivo	Stato iniziale. L'adapter non è attivo e non è ancora stato avviato.
Listener non attivo	Il servizio (dispatcher) del listener dell'adapter non è stato avviato.
Inizio	L'adapter è in fase di avvio.
Avviato	L'adapter è stato avviato.
Interruzione	Interruzione in corso.
Sospensione in corso...	Sospensione in corso.
In Pausa	In pausa.

Non in esecuzione	L'adapter non è in esecuzione sul computer dell'adapter.
Errore	Errore nel file di configurazione dell'adapter; l'adapter non può essere avviato.
Errore di connessione	Errore nella connessione dell'adapter (porta/host errato), oppure nessun segnale rilevato prima di eseguire l'adapter per la prima volta. Stato durante la prima esecuzione dell'adapter.
Bloccato	È stato raggiunto il numero massimo di eventi rifiutati.

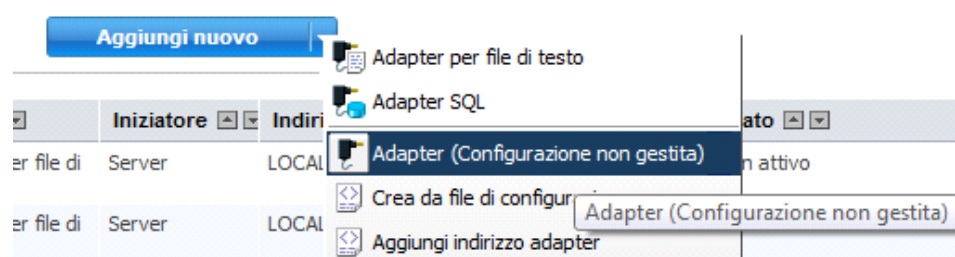
Creazione di un adapter tramite la procedura guidata di configurazione dell'adapter

La procedura guidata di configurazione dell'adapter è una nuova funzionalità dell'interfaccia utente CA Business Service Insight che offre una guida per la creazione di un nuovo adapter, grazie a un'interfaccia più intuitiva rispetto all'Editor XML. La procedura guidata fornisce all'utente una serie di schede contenenti tutte le informazioni necessarie per creare un adapter e alla fine consente di scaricare una copia del file di configurazione XML compilato. Tuttavia, esistono alcuni limiti nelle operazioni consentite con la procedura guidata.

Attualmente non è possibile:

- Fare riferimento alla stessa struttura di input (formato di input) da diverse interfacce di origine dati
- Fare riferimento alla stessa struttura di output (convertitore) da input diversi
- Configurare uno switch del formato di input e utilizzarlo per decidere quale formato di input utilizzare dall'interfaccia dell'origine dati
- Definire un campo ID origine dati nella struttura di output
- Definire più file nella stringa di connessione per eseguire la stessa query su file di testo o di Excel
- Utilizzare i vincoli temporali UTC o UtcNow
- Specificare i valori che iniziano o terminano con uno spazio

Durante la creazione di un nuovo adapter tramite la procedura guidata sono presenti quattro opzioni tra cui scegliere, come illustrato nella figura seguente:



Le prime due opzioni consentono di creare un adapter per file di testo o un adapter SQL tramite l'interfaccia della procedura guidata. L'opzione successiva Adapter (Configurazione non gestita) deve essere selezionata durante l'aggiunta di un adapter preconfigurato creato utilizzando l'Editor XML. Questa opzione non consente di modificare ulteriormente la configurazione dell'adapter tramite la procedura guidata in un secondo momento. L'ultima opzione Crea da file di configurazione consente di caricare la configurazione di un adapter preconfigurato, che verrà importata dal sistema nell'interfaccia della procedura guidata per ulteriori modifiche. Per questa opzione è necessario che nessuno dei limiti della procedura guidata descritta sia implementato in tale configurazione specifica.

Oltre questo punto, le opzioni della procedura guidata di configurazione dell'adapter offrono la stessa funzionalità descritta per la configurazione manuale dell'adapter. Lo scopo è fornire un'interfaccia più semplice e intuitiva per la modifica delle impostazioni di configurazione. Poiché si applicano gli stessi principi e le stesse funzioni dell'alternativa manuale, consultare le relative sezioni per ulteriori informazioni.

Esecuzione e verifica dell'adapter

L'impostazione del file di configurazione dell'adapter non può essere completata in un unico ciclo. Potrebbero essere necessarie varie ripetizioni, durante le quali l'adapter viene eseguito, e i risultati vengono controllati per verificare che la configurazione dell'adapter sia corretta.

Di seguito vengono descritti alcuni dei problemi più frequenti da correggere:

- Problemi di connessione (tra l'origine dati e l'adapter o tra l'adapter e il listener)
- Errori nel file di configurazione, quali:
 - Struttura errata
 - Utilizzo non valido di attributi
 - Utilizzo non valido di maiuscole/minuscole (negli adapter viene fatta distinzione tra maiuscole e minuscole; "Sì" e "sì" sono diversi, e così via)
 - Assegnazione di un valore errato
- Errori di modifica dei dati, quali struttura dell'evento di output, valori di eventi non validi, errore nelle query.

Attenersi alla seguente procedura:

1. Impostare l'adapter su RunOnce = "yes" e LogDebugMode = "yes", e impostare RejectedEventsUpperLimit su un numero ragionevole (consultare la sezione [Modalità di esecuzione dell'adapter](#) (a pagina 85)).

Nella figura seguente vengono illustrate le impostazioni di configurazione richieste per la verifica.

2. È inoltre possibile utilizzare la modalità non in linea ai fini delle impostazioni del file di configurazione.
3. Una volta che il file di configurazione è stato caricato correttamente, modificare nuovamente l'impostazione su in linea (consultare la sezione Modalità di esecuzione dell'adapter).
4. Ogni iterazione comprende i seguenti passaggi:
 - a. Aggiornare/correggere il file di configurazione dell'adapter.
 - b. Eliminare tutti i file di output dell'adapter.
 - c. Eseguire l'adapter facendo doppio clic sul collegamento dell'eseguibile dell'adapter o del file .bat che è stato creato.
 - d. Aprire il file registro dell'adapter utilizzando il browser di log (utility Log Browser.exe che si trova nella cartella Utilities CA Business Service Insight) e verificare che non vi siano messaggi di errore.

- Per ogni record rifiutato dall'adapter deve essere visualizzata sulla console la lettera "R". Tenere presente che gli eventi rifiutati sono previsti finché non siano state completate tutte le conversioni necessarie.

- [illegible]

- AVVISO:** l'eliminazione dei file di output dell'adapter è molto rischiosa. Deve essere eseguita solo in questa fase ai fini della verifica. Ad esempio, durante l'eliminazione del file di controllo, l'adapter perde traccia dei file già letti e potrebbe pertanto perdere i dati o leggere nuovamente i file. L'unico file che può essere eliminato durante la modalità operativa senza conseguenze sul funzionamento è il file di log.

Se è presente un messaggio di errore, fare doppio clic sul messaggio e leggerlo. Il problema è in genere causato da un errore nel file di configurazione.

Conversioni di eventi e risorse

Nel passaggio precedente, erano presenti alcuni eventi rifiutati creati durante l'esecuzione dell'adapter. Questi eventi rifiutati vengono memorizzati nel file `RejectedEvents.txt`, ma vengono anche archiviati come voci di conversione in sospeso nel database CA Business Service Insight. Il passaggio successivo nel processo di caricamento dei dati non elaborati nel sistema è fornire una conversione dei dati misurati in modo che CA Business Service Insight possa utilizzare questi dati come richiesto.

Gli eventi di dati non elaborati possono essere rifiutati a causa del tipo di evento oppure della risorsa non definita nel sistema. Gli eventi in sospeso sono definiti per il tipo di tabella di conversione da cui provengono. Gli esempi più comuni presentano i tipi di evento provenienti da una tabella di conversione e le risorse provenienti da un'altra tabella di conversione.

Quando una nuova risorsa viene rilevata dall'adapter (ad esempio, per esempio, un nuovo server è aggiunto allo strumento di monitoraggio della rete e viene visualizzato come una nuova voce nell'origine dati da questo strumento di monitoraggio), può essere aggiunta al modello di risorse del sistema. È necessario completare due passaggi in modo che questa nuova risorsa diventi segnalabile in CA Business Service Insight.

Innanzitutto, è necessario creare la risorsa come un'entità (risorsa) CA Business Service Insight e quindi immettere una conversione. Viene così fornito il collegamento tra la rappresentazione di stringa trovata nell'origine dati e l'entità definita come risorsa in CA Business Service Insight. È possibile eseguire i due passaggi di questo processo in una singola azione tramite l'interfaccia utente in un processo noto come **Aggiungi e converti**, con cui è possibile creare la nuova risorsa e la voce di conversione necessaria in una singola procedura. Durante l'aggiunta e la conversione, è possibile selezionare più voci purché abbiano le stesse impostazioni di allocazione. L'esecuzione della conversione è il processo di creazione della corrispondenza tra il valore dell'origine dati e l'entità CA Business Service Insight. Durante la creazione delle conversioni, viene aggiunta una voce alla tabella di conversione con i valori corrispondenti. Quindi, nelle query successive all'origine dati, l'adapter saprà automaticamente come gestire il nuovo valore.

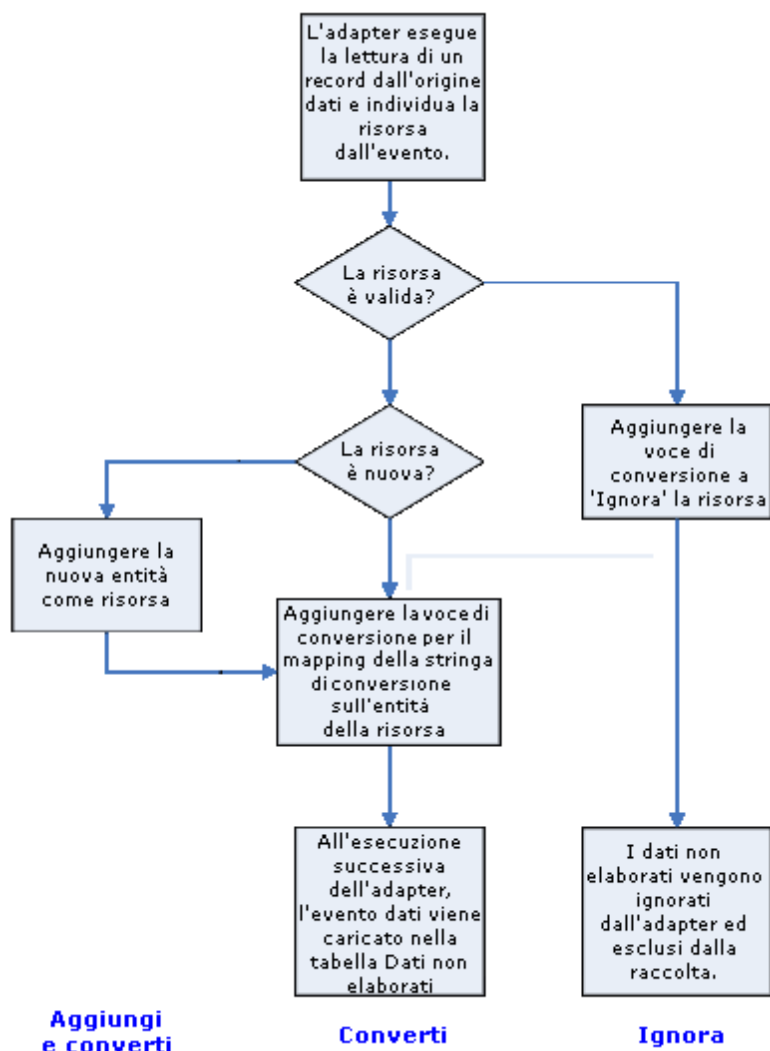
In questa fase, l'adapter è stato già eseguito e ha inviato le richieste di conversione per ogni valore nei campi di conversione. Gli eventi associati a tali valori sono stati rifiutati e sono in attesa di essere inviati a CA Business Service Insight una volta completata la conversione. Le conversioni possono essere eseguite in modo manuale o automatico tramite uno script di conversione.

È possibile completare le azioni di conversione seguenti sulle voci di conversione in sospeso:

Converti: consente di creare una voce nella tabella di conversione, trovare la corrispondenza tra il valore dell'origine dati e l'entità pertinente CA Business Service Insight. L'entità CA Business Service Insight su cui vengono effettuate le conversioni deve esistere già. (Considerare, ad esempio, il caso di una risorsa con errore di ortografia da un'origine dati. Potrebbero essere presenti diversi nomi specificati nell'origine dati che si riferiscono effettivamente a una singola entità logica. Vale a dire, Server001 e SERVER001. Nelle risorse CA Business Service Insight viene fatta distinzione tra maiuscole e minuscole).

- **Aggiungi e converti:** consente di creare una nuova entità in CA Business Service Insight e contemporaneamente di aggiungere una voce di conversione per tale entità nella tabella di conversione. Si tratta dell'azione più comune eseguita sulle voci di conversione in sospeso perché il meccanismo di conversione è utilizzato per generare l'infrastruttura in CA Business Service Insight.
- **Ignora:** quando viene ignorata una voce, tutti gli eventi associati vengono ignorati e non inviati alla tabella dei dati non elaborati CA Business Service Insight. Gli eventi ignorati verranno persi. Ad esempio, se l'origine dati contiene informazioni su tutti i server in un data center, ma sono necessari solo i dati del server applicazioni per il calcolo del livello di servizio, di conseguenza tutti i server raggiungeranno l'adapter per la conversione, ma solo i server applicazioni verranno convertiti. Tutti gli altri vengono ignorati poiché CA Business Service Insight garantisce che l'evento non necessario sia ignorato. Una voce ignorata può essere convertita in una seconda fase se necessario, ma i dati vengono acquisiti da quel momento in poi.
- **Elimina:** la voce di conversione viene rimossa e anche l'evento rifiutato associato viene eliminato. Se lo stesso valore viene successivamente inviato dall'origine dati, viene creata una nuova voce in sospeso.

Nel seguente diagramma di flusso vengono riepilogati i casi di utilizzo di queste opzioni:



Conversioni manuali

Le conversioni manuali sono necessarie quando l'entità esiste già in CA Business Service Insight. Può verificarsi in varie situazioni. Ad esempio, quando viene eseguito uno script di conversione per creare automaticamente le risorse da un'origine esterna, ma le conversioni non possono essere automatizzate. Oppure, quando l'origine dati presenta voci incerte e una risorsa è stata definita due volte in modo diverso (vale a dire, Server001p e server001P). Può anche essere causato da risorse create manualmente.

Per eseguire una conversione manuale quando la risorsa esiste già:

1. Nel menu Progettazione, fare clic su Conversioni, quindi su Voci di conversione.
2. Per impostazione predefinita, vengono visualizzate tutte le voci in sospeso.
3. Selezionare la voce di conversione in sospeso da convertire selezionando la casella di controllo accanto ad essa.
4. Fare clic su Converti.
5. Selezionare l'entità pertinente (risorsa, tipo di evento, e così via) dall'elenco. Se nessun elemento viene visualizzato nell'elenco, potrebbe essere necessario modificare i criteri di ricerca predefiniti nella casella.
6. Selezionare la risorsa/tipo di evento dall'elenco visualizzato di entità facendo clic sulla riga contenente l'elemento. Una volta selezionata, la riga resta evidenziata.
7. Fare clic su Converti. La voce di conversione è archiviata nel sistema.

Per eseguire una conversione manuale quando la risorsa NON esiste ancora:

1. Nel menu Progettazione, fare clic su Conversioni, quindi su Voci di conversione.
2. Per impostazione predefinita, vengono visualizzate tutte le voci in sospeso.
3. Selezionare la voce di conversione in sospeso da trasformare in una risorsa CA Business Service Insight e convertire selezionando la casella di controllo accanto ad essa.
4. Fare clic su Aggiungi e converti.
5. Verificare che sia specificato il nome della risorsa desiderato. È inoltre possibile personalizzare il campo *Nome visualizzato* per modificare il modo in cui la risorsa viene visualizzata in un report. Se la risorsa deve essere gestita come parte di un gruppo di modifiche, è necessario indicare in questo campo il gruppo di modifiche specifico.
6. La data di inizio validità della risorsa deve essere configurata come la data in cui è possibile eseguire il report dalla risorsa nel sistema. Nota: la risorsa NON verrà visualizzata nei report prima della data specificata in questo campo.
7. Fare clic sulla scheda Dettagli e selezionare le seguenti opzioni per l'associazione della risorsa: In vigore (true/false), Tipo di risorsa, appartenenza a Gruppo di risorse, associazione a Servizio e Contratto.
8. Fare clic su Salva e converti. La risorsa viene aggiunta al catalogo dei servizi delle risorse e la voce di conversione viene archiviata nel sistema. Una volta trattate tutte le voci in sospeso, è possibile verificare che i dati siano caricati nel sistema:
9. Accedere alle risorse in CA Business Service Insight e verificare che la nuova risorsa sia stata creata.
10. Eseguire nuovamente l'adapter.
11. Controllare che il file degli eventi rifiutati sia vuoto, il contenuto e le dimensioni.
12. Controllare che il file degli eventi da inviare sia vuoto, il contenuto e le dimensioni.
13. Verificare con lo strumento dei dati non elaborati gli eventi che sono stati aggiunti ai dati non elaborati.

Configurazione dell'infrastruttura

La configurazione dell'infrastruttura include la definizione delle entità di modellazione dei dati come identificato durante la fase di progettazione.

Questa fase non include la definizione di tutte le entità di infrastruttura e viene conclusa una volta completata la configurazione dell'adapter.

Durante questa fase le entità seguenti vengono inserite nel sistema CA Business Service Insight:

- Tipi di evento
- Tipi di risorsa
- Risorse e allocazioni risorse
- Gruppi di risorse

Nota: se l'adapter è stato eseguito correttamente, è possibile utilizzare la funzionalità di importazione automatica quando si definiscono i campi Tipo di evento.

Conversioni automatiche con script di conversione

La conversione automatica consiste nei processi automatizzati di creazione e conversione e nell'infrastruttura, in base a un'origine dati esterna tramite script che eseguono le azioni di conversione.

La conversione automatica viene effettuata tramite script di conversione. Gli script di conversione accelerano il processo di mapping di nuove risorse IT e business su CA Business Service Insight. Lo script di conversione identifica automaticamente quando viene ricevuta una nuova voce di conversione e converte le risorse, consentendo un mapping rapido ed efficiente della risorsa. L'automazione supporta l'interfaccia nei CMDB, consentendo al sistema di identificare le risorse in base alla relativa definizione configurata. La conversione automatica ha diversi benefici, tra cui la gestione facilitata delle conversioni e la prevenzione di errori. È possibile utilizzare gli script di conversione per creare nuove risorse e allocare le modifiche.

Inoltre, gli script di conversione possono essere utilizzati per:

- Convertire voci in oggetti esistenti CA Business Service Insight.
- Aggiungere nuovi oggetti CA Business Service Insight e convertirli in base alle voci di conversione esistenti.
- Creare oggetti e convertire le voci sulla base delle tabelle all'esterno di CA Business Service Insight, ad esempio, tabelle di risorse da un altro CMS esterno.
- Controllare se un oggetto esiste.
- Creare risorse ed eseguire le allocazioni di risorsa, quali tipi di risorsa, gruppi di risorse, contraenti e componenti del servizio.
- Allocare/deallocare risorse in Gruppi di modifiche.

Poiché il processo di conversione è supportato anche dalla procedura manuale nell'interfaccia utente, è necessario decidere quale processo di conversione selezionare. In questo caso, considerare i seguenti pro e contro riguardanti la conversione automatica:

- Maggiore complessità del progetto: gli script di conversione richiedono appropriate competenze e abilità di sviluppo di script.
- Maggiore tempo di sviluppo e controllo di qualità per la verifica.
- Implementazione non necessaria nei casi in cui il processo di conversione è solo un lavoro iniziale richiesto una sola volta.
- Possibile pianificazione per l'implementazione come parte di un approccio di seconda fase.
- Ridotta manutenzione di routine.
- Ridotti errori di conversione umana.
- Per ulteriori informazioni sulla creazione e l'esecuzione degli script di conversione, consultare la guida di *SDK* nel capitolo dedicato all'integrazione di CMDB.

Argomenti avanzati della funzionalità dell'adapter

Le seguenti sezioni riguardano argomenti avanzati della funzionalità dell'adapter.

Univocità dell'evento

L'univocità dell'evento è un meccanismo dell'adapter che offre un processo in grado di impedire l'inserimento di dati non elaborati duplicati nella tabella T_Raw_Data.

Senza l'univocità dell'evento, quando l'adapter viene eseguito su un'origine dati e carica gli eventi nel database, non viene attuata alcuna convalida per un evento identico. La funzionalità di univocità dell'evento risolve questo aspetto fornendo la possibilità di specificare se controllare l'univocità degli eventi prima dell'inserimento e quali azioni intraprendere qualora l'evento sia univoco. Il processo di verifica può, tuttavia, avere un impatto negativo sulle prestazioni dell'adapter.

La soluzione consente all'utente di definire una chiave, che può essere basata su campi dell'evento. Tale chiave rappresenta l'univocità dell'evento, vale a dire che se due eventi hanno la stessa chiave, sono eventi identici.

È anche possibile specificare l'azione da eseguire nel caso in cui una chiave duplicata è già presente nel database. Tali azioni vengono descritte di seguito.

Nota: la chiave può essere definita come una combinazione di vari campi dal convertitore.

File di configurazione dell'adapter con univocità dell'evento

TranslatorCollection/Translator/OnDuplication

Questo campo specifica le azioni da eseguire se l'evento esiste già nel database.

I valori possibili sono:

- **Add:** inserisce (ulteriori) eventi nel database.
- **Update:** comporta l'eliminazione (in alcuni casi) dell'evento esistente e l'inserimento di uno nuovo dopo la convalida della modifica dell'evento.
- **UpdateAlways:** comporta l'eliminazione (in alcuni casi) dell'evento esistente e l'inserimento di uno nuovo senza verificare le modifiche.
- **Ignore:** non inserisce un nuovo evento nel database.

Il valore predefinito è Add.

Ignore rispetto a Add

Può verificarsi una riduzione minore delle prestazioni durante l'esecuzione dell'adapter in modalità Ignore. Tuttavia, in modalità Add, il sistema attiva il ricalcolo in ogni caso di un evento duplicato.

Update rispetto a UpdateAlways

L'utilizzo dell'opzione Update riduce le prestazioni dell'adapter, mentre UpdateAlways riduce le prestazioni di calcolo e al tempo stesso riattiva il ricalcolo per ogni caso.

TranslatorCollection > Translator > TranslatorFields > TranslatorField > IsKey

Questo attributo specifica se il campo cui appartiene deve contribuire con il proprio valore alla chiave univoca per i dati non elaborati. Viene incluso se value = "yes" e non incluso se value = "no". Se non è specificato alcun valore, il valore predefinito è "yes" (sì) per i campi standard (ResourceId, EventTypeId e Timestamp) e "no" per tutti gli altri. La chiave deve essere selezionata attentamente per accertarsi che i dati non elaborati conservino l'integrità richiesta e assicurarsi che i calcoli siano completamente accurati.

Comportamento del listener dell'adapter con univocità dell'evento

Alla ricezione di un nuovo evento dall'adapter, il listener controlla il valore del campo `OnDuplication`. Quando il valore è "add", viene eseguito il normale processo di inserimento. Se il valore non è "add", il listener controlla l'esistenza di un evento con lo stesso `UniqueKey` e lo stesso ID del lettore nel database. Se il database contiene già un evento come descritto, il nuovo evento non viene inserito nel database quando il valore di `OnDuplication` è "ignore".

Quando il valore di `OnDuplication` è "update", viene eseguito un controllo delle modifiche nell'evento. Se tutti i campi sono identici, il nuovo evento non viene inserito nel database.

Quando il valore di `OnDuplication` è "updateAlways", il controllo precedente viene ignorato e viene eseguito comunque un aggiornamento.

In entrambe le modalità `update` e `updateAlways`, vengono considerati i passaggi seguenti:

- Se `ResourceId`, `EventTypeId` e `Timestamp` hanno subito modifiche, il ricalcolo viene eseguito per tutte le metriche che utilizzano la formula dell'evento precedente.
- L'evento viene aggiornato con i valori appropriati.

Dati della transazione di aggregazione

I dati della transazione vengono spesso raccolti in modo da corrispondere alle soglie o per essere in grado di calcolare le eventuali percentuali periodiche di successo. Ad esempio, ogni cinque minuti viene eseguita una transazione virtuale rispetto al sistema e al risultato (tempo di risposta in millisecondi) e archiviata come mostrato di seguito:

```
[...]  
1/1/2004 00:00 432  
1/1/2004 00:05 560  
1/1/2004 00:10 329  
1/1/2004 00:15 250  
1/1/2004 00:20 275  
1/1/2004 00:25 2860  
1/1/2004 00:30 140  
[...]
```

In altri casi, invece di utilizzare transazioni virtuali, è possibile accedere alle transazioni effettive eseguite in un sistema. In questi casi, è possibile eseguire centinaia o migliaia di transazioni ogni ora.

In entrambi i casi descritti, è necessario evitare, per quanto possibile, il caricamento di un tale volume di informazioni in CA Business Service Insight.

L'aggregazione dei dati per periodi di tempo è il modo migliore per ridurre le quantità dei dati. Quando viene fissata la soglia in base alla quale il successo è misurato, l'aggregazione può essere eseguita consentendo all'adapter di contare il numero di transazioni nel periodo aggregato che hanno avuto esito positivo. Ad esempio, se la soglia di successo nell'esempio precedente è impostata su 500 millisecondi, solo cinque transazioni su sette sono state completate correttamente entro il periodo visualizzato. Il problema con questo approccio è la soglia fissata: cosa succederebbe se, successivamente, un utente desiderasse modificare la soglia? In una situazione simile, i dati non elaborati devono essere riletti e verificati dall'adapter rispetto alla nuova soglia.

Di conseguenza, l'adapter deve aggregare i dati delle transazioni in un modo ottimale e flessibile senza perdere dati significativi.

Una soluzione limitata è consentire all'adapter di verificare le transazioni rispetto a diverse soglie. Esistono due modi per tale scopo:

- Un evento con più verifiche: Event Type = {TotalTransactions, TransBelow250, TransBelow500, TransBelow750, [...]}
- Più eventi con una verifica: Event Type = {TotalTransactions, Threshold, TransBelowThreshold}.

Entrambe le opzioni presentano lo stesso problema, ossia le soglie possono essere modificate in futuro solo all'interno di un piccolo gruppo di valori predefiniti.

Soluzione consigliata

Presupposto: tutte le soglie potenziali sono un multiplo di un dato numero. Ad esempio, tutti i valori di soglia (in millisecondi) sono multipli di 250, quindi 0, 250, 500, 1750 e 3000 millisecondi sono tutti possibili soglie.

In base a questo presupposto, la soluzione suggerita è aggregare le transazioni arrotondando tutti i valori al multiplo comune e contando quante transazioni rientrano in ogni valore arrotondato. Event Type = {RangeFrom, RangeTo, TransactionCount}

Ad esempio, gli eventi seguenti verranno generati per aggregare i dati visualizzati sopra, in cui il multiplo comune è 250 millisecondi:

```
{1}, 250, 2
{251, 500, 3}
501, 750, {1}
751, 1000, {1}
```

Commenti:

La data/ora di tali eventi è la stessa. Ad esempio, tutti gli eventi aggregati potrebbero verificarsi il 1/1/2007 00:00 e potrebbe esserci un altro insieme di eventi per il prossimo campione il 1/1/2007 01:00, supponendo un'aggregazione ogni ora.

RangeTo viene calcolato arrotondando una transazione al multiplo comune (consultare la sezione successiva).

RangeFrom è RangeTo meno il multiplo, più 1. È specificato solo per motivi di chiarezza, ed è possibile ignorarlo.

Ad esempio, l'aggregazione per ora dovrebbe essere come segue (sostituire MULTIPLE con il valore del multiplo):

```
select trunc (time_stamp, 'hh') "TimeStamp",
       round (response_time/MULTIPLE, 0)*MULTIPLE-MULTIPLE+1 "RangeFrom",
       round (response_time/MULTIPLE, 0)*MULTIPLE "RangeTo",
       count (*) "TransactionCount"
from   t_log
group by trunc (time_stamp, 'hh'),
         round (response_time/MULTIPLE, 0)*MULTIPLE
```

Nella business logic, la seguente condizione può essere applicata agli eventi:

```
If eventDetails("RangeTo")<=Threshold Then
    SuccessfulTransactions=SuccessfulTransactions+eventDetails("TransactionCount")
End If
```

Alcune riflessioni conclusive:

- Poiché le transazioni tendono a distribuirsi normalmente, il numero di eventi di aggregazione deve essere relativamente basso.
- La selezione di multipli comuni superiori consente di derivare meno eventi aggregativi.

- Il volume di eventi di aggregazione deve essere sempre uguale o inferiore al volume dei dati non elaborati.

Esecuzione di un adapter protetto da firewall

Per eseguire un adapter protetto da firewall, è consigliata la soluzione seguente:

Aprire due porte; la prima porta è la porta assegnata all'adapter in CA Business Service Insight; la seconda porta, facoltativa ma consigliata, è la porta del server di registrazione dei log. La porta del server di registrazione dei log predefinita è la porta 4040. L'apertura della porta del server di registrazione dei log consente all'adapter di inviare report al log CA Business Service Insight e inoltre di generare notifiche. È facoltativa perché l'adapter invia sempre i report a un file di log locale.

Per entrambe le porte il protocollo in uso è TCP/IP.

Caricamento dei dati da diverse directory

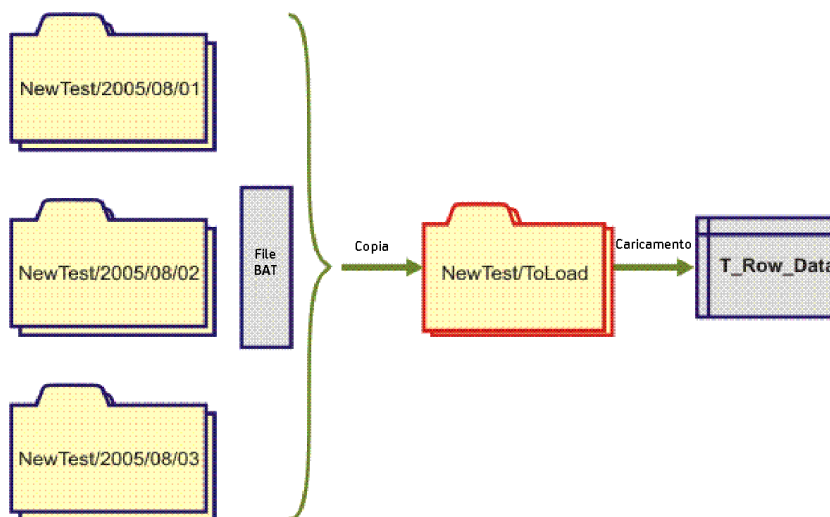
Questa sezione descrive una soluzione che può essere implementata se i file di origine dati (input a un adapter CA Business Service Insight) si trovano in directory diverse ogni giorno, o per tutti i periodi di tempo impostati (in base a una convenzione di denominazione specifica).

Ad esempio, considerare la struttura di directory `c:\NewData\YYYY\MM\DD`. In questo esempio, ogni giorno c'è una nuova directory in cui vengono posizionati i file relativi di quel giorno.

L'adapter CA Business Service Insight deve esplorare la directory più recente e caricare i nuovi file.

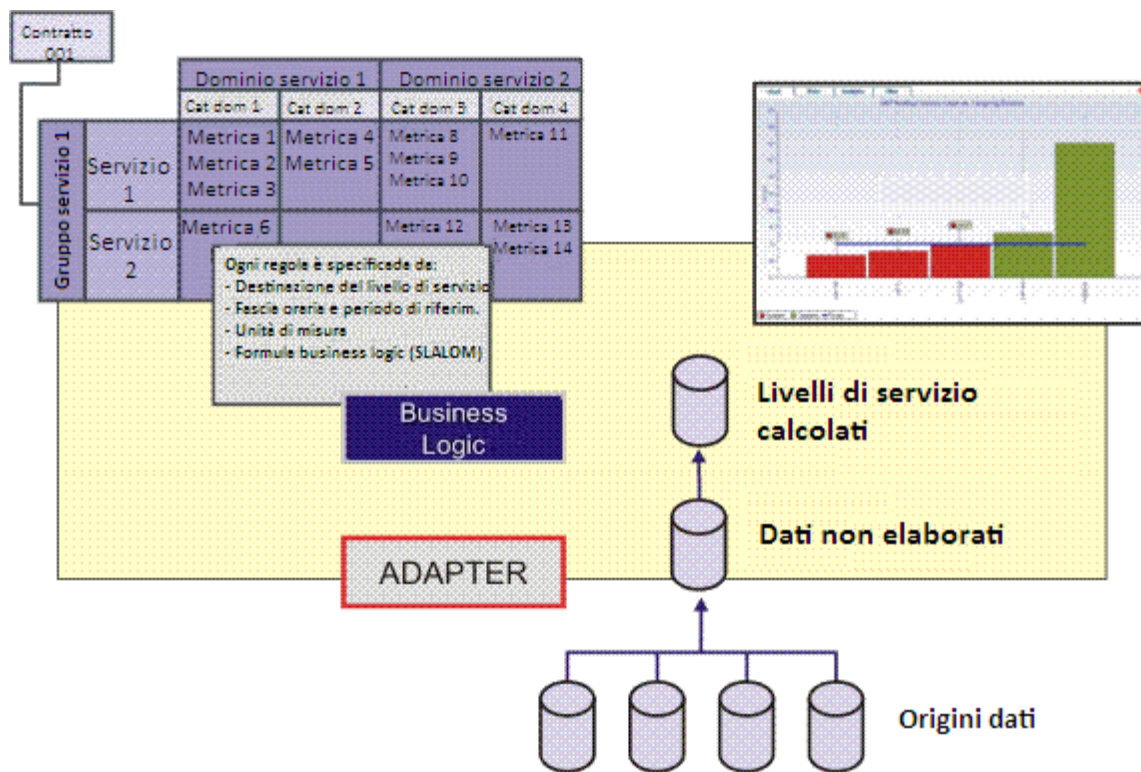
Una soluzione alternativa disponibile è aggiungere alcuni comandi all'inizio del file .bat che esegue l'adapter. Questi comandi copiano le origini dati che devono essere caricate dalla cartella specifica (in base alle relative convenzioni) in una singola cartella dedicata creata appositamente a tale scopo. L'adapter carica sempre le informazioni da questa cartella.

Nella figura seguente viene descritta questa soluzione:



Implementazione di business logic (esperto di business logic)

Nella figura seguente viene illustrata la posizione della business logic all'interno di CA Business Service Insight. Rientra sotto ciascuna metrica inclusa nei contratti.



In questa fase di implementazione, vengono configurati gli adapter pertinenti e i record dei dati non elaborati devono essere già disponibili nella tabella T_RAW_DATA in CA Business Service Insight. Ora è possibile applicare la business logic agli eventi per generare il risultato del livello di servizio effettivo per ciascuna metrica.

L'implementazione di business logic è il processo di scrittura del codice che opera a livello logico sui dati non elaborati (dati non elaborati inviati dall'adapter) per calcolare i livelli di servizio.

Ciascuna metrica dispone di una propria formula di business logic con cui calcolare il livello di servizio effettivo, anche se molte delle metriche nel sistema hanno in genere una logica comune che può essere applicata a diversi insiemi di eventi di dati non elaborati.

Ad esempio, una metrica che calcola il tempo di risoluzione del ticket Severity 1 e un'altra metrica che calcola il tempo di risoluzione del ticket Severity 2 valutano un insieme diverso di record: l'una utilizza solo i ticket Severity 1 e l'altra solo i ticket Severity 2. Tuttavia, molto probabilmente entrambe applicano lo stesso metodo per calcolare il tempo di risoluzione. Lo script del tempo di risoluzione verrà creato e verificato una volta, definito come un modulo di business logic, e verrà quindi utilizzato da entrambe le metriche includendo questo modulo nelle aree di business logic delle metriche.

Pertanto, in genere durante lo sviluppo di script di business logic i moduli o modelli principali vengono sviluppati in modo da renderli disponibili per tutte le metriche nel sistema. Inoltre, ogni categoria di dominio in genere riflette un tipo diverso di misurazione e pertanto ogni categoria di dominio in genere segue un modulo o modello di business logic diverso.

Flusso di lavoro di implementazione di business logic

La fase di implementazione di business logic comprende i seguenti passaggi:

- Definire una formula

Creare la formula in base ai requisiti di calcolo definiti in fase di progettazione. Le formule definite sono tutte formule univoche da utilizzare nelle loro varie permutazioni nelle metriche del contratto, ognuna come un modulo di business logic.

Ad esempio, se il contratto contiene tre metriche per calcolare il tempo medio di risoluzione dei ticket e una metrica per la priorità di ogni ticket, viene quindi sviluppata un'unica formula per il calcolo del tempo di risoluzione dei ticket e con la priorità del ticket impostata come parametro. La formula, una volta verificata, viene definita come modulo e associata a tutte le relative metriche.

- Verificare la formula

I test vengono eseguiti per verificare che la formula sia definita correttamente e senza errori e che i calcoli producano il risultato previsto. È importante coprire tutti i casi estremi e le condizioni limite come parte della verifica. L'ambito di business logic è dove viene eseguita la formula ai fini della verifica. Quando viene definita all'inizio, la formula viene verificata interamente. Quindi, una volta applicata a tutte le metriche come un modulo, è importante eseguire ogni metrica nell'ambito almeno una volta per verificare che riceva gli eventi (vale a dire che la registrazione sia corretta) e restituisca un risultato appropriato.

- Definire il modulo SLALOM associato

Ogni modulo è un calcolo di business logic univoco e con la definizione del relativo parametro può essere applicato a tutte le metriche pertinenti. Durante la definizione del modulo, è importante che il modulo sia verificato attentamente e documentato in dettaglio: quali azioni esegue il modulo (descrizione del calcolo), i parametri previsti (nome, significato e utilizzo), e così via.

- Associare tutte le metriche al modulo di business logic pertinente

Per ciascuna metrica nei contratti definiti, è necessario specificare un collegamento al modulo di business logic pertinente. Deve quindi essere eseguito nell'ambito di business logic per accertarsi che il collegamento sia implementato correttamente e che la registrazione funzioni correttamente per la ricezione di eventi pertinenti e la restituzione dei risultati previsti.

Moduli di business logic

È necessario tener presente alcune considerazioni importanti durante la compilazione dei moduli di business logic, soprattutto nel caso di utilizzo di più moduli in una singola metrica:

- Per garantire che l'utilizzo di un modulo sia chiaro, è necessario aggiungere un commento nella parte superiore della business logic per la metrica.
- Se viene utilizzata una piccola parte di codice all'interno dello spazio di business logic della metrica e vengono inclusi uno o più moduli nel codice, verificare che per qualsiasi gestore eventi predefinito o subroutine sia stata rimossa quella sezione di codice dalla business logic della metrica principale. È necessario accertarsi che ogni subroutine e gestore eventi siano definiti solo *una volta* in ognuno dei moduli utilizzati da una metrica particolare. In questo modo si evita di confondere il motore di calcolo e produrre risultati imprevisti.

Nota: Se, ad esempio, l'utente definisce la funzione OnPeriodStart() nel proprio modulo e vi inserisce il codice specifico, e mantiene OnPeriodStart() predefinito senza codice nella schermata di business logic principale della metrica, di conseguenza durante l'esecuzione il motore non sa quale subroutine eseguire. Potrebbe non eseguire il codice previsto dall'utente per l'esecuzione.

- È necessario essere estremamente attenti nell'impostazione di parametri per la subroutine OnRegistration. In alcuni casi, questa operazione può interrompere l'attivazione automatica creata all'interno del sistema per ricalcolare le metriche sulla base dell'aggiunta di nuovi dati non elaborati.

All'interno della business logic

La business logic si trova in uno script azionato dagli eventi e basato sulla sintassi di VBScript. Ogni evento che raggiunge la business logic, attiva un gestore eventi.

I seguenti tipi di eventi vengono inviati dal motore per essere valutati dalla business logic:

- Eventi di dati non elaborati. Input di dati non elaborati effettivi su cui la business logic basa i propri risultati. Il motore invia solo gli eventi di dati non elaborati pertinenti in base alla registrazione della formula.
- Eventi del motore. Eventi inviati dal motore che in modo implicito riflettono le proprietà della definizione della metrica, quali la definizione del periodo di applicazione e il periodo di riferimento.
- Eventi di dati intermedi. Eventi generati da altri script di business logic che possono essere utilizzati all'interno di un altro.

La formula di business logic contenuta all'interno della definizione di metrica procede alla valutazione degli eventi e restituisce un risultato del livello di servizio su cui si basano i report. A seconda dei risultati del livello di servizio e della definizione di dominio, il motore produce inoltre un risultato di deviazione (se un obiettivo del livello di servizio è stato applicato alla metrica). I risultati generati vengono archiviati in una tabella di database denominata T_PSL. A questa tabella vengono inviate le query dalla procedura guidata di creazione report durante la generazione di report, pertanto, tutti i dati dei report vengono precalcolati per garantire che le prestazioni di reporting siano ottimizzate.

Flusso di eventi

Come descritto in precedenza, gli input per la business logic sono gli eventi del motore e gli eventi di dati non elaborati.

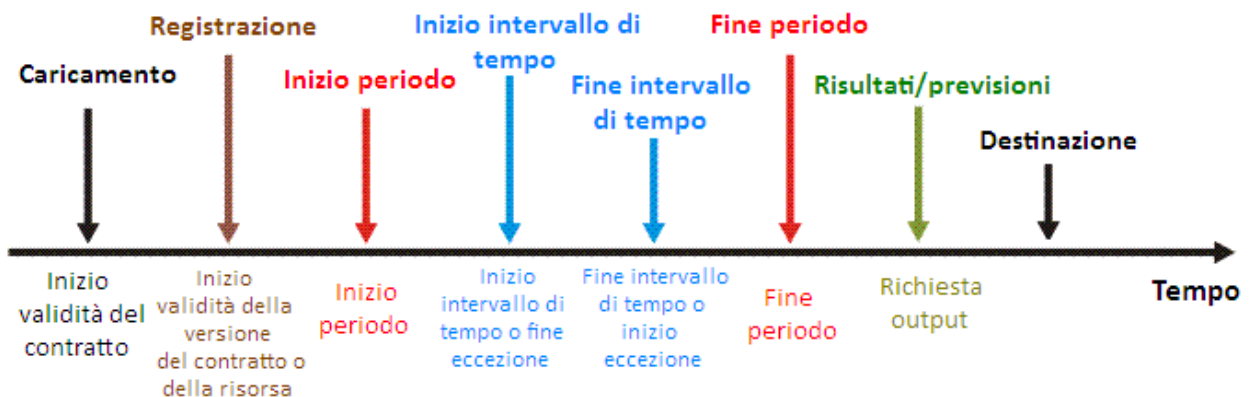
Gli eventi di dati non elaborati ricevuti dalla business logic vengono determinati dalla funzione di registrazione in cui il codice richiede un insieme specifico di eventi di dati non elaborati definiti in base agli identificatori Tipo di evento e Risorsa.

Nella business logic, la registrazione associa inoltre una subroutine definita dall'utente che viene eseguita per gestire l'evento di dati non elaborati una volta ricevuto. (Per impostazione predefinita, è necessario rinominare OnXXXEvent con un nome più significativo.)

Gli eventi del motore vengono generati dal motore in base alle definizioni di contratto e metrica associate. Una volta generato e ricevuto l'evento del motore, il motore esegue il gestore eventi pertinente. Ciascun evento del motore ha un gestore eventi implicito. Questi gestori di evento corrispondono a funzioni e procedure definite nella parte superiore del VBScript. Sono obbligatori il gestore eventi che gestisce la registrazione e la funzione Result per l'implementazione nel codice. Tutti gli altri gestori eventi sono facoltativi. Tuttavia, la business logic non elabora gli eventi del motore per cui non sono implementati i gestori eventi. Pertanto, è consigliabile mantenerli, anche se non vengono utilizzati, per consentire miglioramenti futuri.

Nota: durante la scrittura dello script di business logic è importante implementare tutti gli eventi del motore per comprendere tutte le possibilità finali. Ad esempio, anche se durante la prima fase di implementazione la definizione di un periodo di applicazione non era valida, ma lo sarà in futuro, tutte le formule richiederanno la rettifica. Si consiglia pertanto all'esperto di business logic di definire in anticipo il comportamento nel periodo di applicazione e fuori il periodo di applicazione, anche se inizialmente non è applicabile, in modo che quando il comportamento è introdotto, le necessarie modifiche al sistema sono banali.

Di seguito sono riportati diversi eventi del motore e i relativi gestori eventi:



- **OnLoad (Time):** facoltativo, invocato una sola volta all'inizio dei calcoli quando il contratto è in fase di attivazione. Può essere utilizzato per inizializzare le variabili globali.
- **OnRegistration (Dispatcher):** obbligatorio, procedura per richiedere i relativi eventi di dati non elaborati e associarli alle procedure di gestione definite dall'utente. Gli eventi vengono invocati e associati alle procedure utilizzando i metodi dell'oggetto Dispatcher.
OnRegistration viene invocato una volta dal motore di calcolo all'inizio del calcolo della metrica e ogni volta che una risorsa registrata per la metrica entra in vigore, cosicché il motore valuta il gruppo di modifiche apportate a tale risorsa che possono influenzare l'insieme di eventi trasmessi per la formula. Il motore presenta la definizione della richiesta dell'evento definita in base agli identificatori Tipo di evento e Risorse e nel caso in cui una risorsa (o gruppo di modifiche che contiene più risorse) cambia qualcosa legato a questo insieme. Una volta diventata effettiva, viene attivato un gestore eventi OnRegistration.
- **OnPeriodStart (Time):** facoltativo, invocato all'inizio del periodo di tempo dell'agente (impostato in base all'unità di tempo). Il primo OnPeriodStart viene attivato una volta che il contratto è diventato effettivo, quando il saldo dei periodi inizia con unità di tempo di un'ora intera. Questo gestore eventi viene in genere utilizzato per inizializzare periodicamente le variabili, ad esempio un contatore che contiene il numero di incidenti aperti entro il periodo di calcolo che deve essere quindi inizializzato a zero all'inizio di ogni periodo.
- **OnPeriodEnd (Time,IsCompleteRecord):** facoltativo, invocato alla fine del periodo di tempo dell'agente (impostato in base all'unità di tempo). È sempre invocato alla fine arrotondata dell'unità di tempo in ore intere (ad esempio, 24:00).
IsCompleteRecord è true quando il periodo della metrica è terminato (in base al tempo reale del server applicazioni) ed è false quando si effettua un calcolo intermedio. Questo gestore eventi viene in genere utilizzato per i calcoli finali del periodo terminale in modo da preparare il terreno per il risultato finale che verrà fornito dalla funzione Result.
- **OnTimeslotEnter (Time):** facoltativo, invocato durante l'ingresso in un periodo di applicazione basato sulla definizione della metrica associata. Ad esempio, se la metrica è associata a una definizione del periodo di applicazione negli orari di ufficio, in cui ogni giorno alle 8:00 inizia il periodo di applicazione, quindi questo gestore eventi verrà attivato a tale ora ogni giorno.
- **OnTimeSlotExit (Time):** facoltativo, invocato durante l'uscita da un periodo di applicazione basato sulla definizione della metrica associata. Ad esempio, se la metrica è associata a una definizione del periodo di applicazione negli orari di ufficio, in cui termina ogni giorno alle 17:00, quindi questo gestore eventi verrà attivato a tale ora ogni giorno.
- **Target():** facoltativo, invocato quando la metrica è specificata con una destinazione dinamica. Consente di determinare la destinazione del livello di servizio di una metrica durante l'esecuzione della business logic. Tali destinazioni possono includere il consumo dei componenti del servizio o le modifiche dell'infrastruttura. Presenta quattro valori, come la funzione Result, uno per ciascuna modalità. Questa funzione viene eseguita *dopo* la funzione Result durante la normale esecuzione.

- **Forecast():** facoltativo, invocato una sola volta durante l'esecuzione di una conferma della versione di contratto; il motore di calcolo calcola il contratto una sola volta in modalità di previsione. La modalità di previsione esegue un ciclo completo del motore di calcolo nel contratto (dalla data di inizio e alla data di fine della versione del contratto). Lo scopo di questo ciclo è esclusivamente il calcolo dei valori di previsione. (Non viene eseguito alcun calcolo nella tabella T_RAW_DATA). Questa funzione viene invocata *al posto* della funzione Result durante questa modalità di esecuzione.
- **Result():** obbligatorio, invocato dal motore di calcolo per ottenere il risultato del calcolo per un determinato periodo. La funzione Result viene invocata sempre dopo il gestore eventi OnPeriodEnd.

Di seguito sono riportati i passaggi del meccanismo di calcolo (servizio PslWriter) per l'elaborazione di una singola formula di business logic:

- PslWriter carica la formula in memoria ed esegue ogni istruzione presente nella sezione delle dichiarazioni (la sezione delle dichiarazioni corrisponde a tutto il codice all'esterno di una funzione o subroutine). Nota: inoltre, tutti i moduli e le librerie associati sono inclusi e compilati in questo modulo di codice singolo per l'esecuzione.
- PslWriter richiede il gestore eventi OnLoad.
- PslWriter richiede il gestore eventi OnRegistration.
- Il calcolo del periodo inizia con l'invocazione di OnPeriodStart.
- Per ogni evento di dati non elaborati registrato durante OnRegistration che rientra nell'intervallo di tempo del periodo, PslWriter invoca il gestore eventi definito dall'utente associato a tale evento.
- Se l'inizio o la fine del periodo di applicazione rientra in un intervallo di tempo del periodo, verrà invocato OnTimeslotEnter oppure OnTimeslotExit.
- Se è presente una modifica dell'infrastruttura pertinente entro l'intervallo di tempo del periodo, verrà invocato OnRegistration.
- Il calcolo del periodo termina con l'invocazione di OnPeriodEnd.
- Se è stata specificata una destinazione dinamica, verrà invocata questa funzione.
- PslWriter invoca la funzione Result per ottenere il risultato finale del periodo di calcolo.

Nota: quando la versione del contratto viene prima confermata ed è stata selezionata una previsione, viene invocata la funzione Forecast invece della funzione Result. Tuttavia, questo caso si verifica solo una volta per ogni versione.

Durante ogni ciclo di calcolo, il motore valuta su cosa gli eventi del motore e gli eventi di dati non elaborati pertinenti si basano al momento del calcolo. Innanzitutto li ordina in base all'ora e in seguito li invia alle formule pertinenti per il calcolo. L'ora dell'evento di dati non elaborati corrisponde alla relativa data/ora, mentre l'ora dell'evento del motore corrisponde all'ora di attivazione. Entrambi i tipi di evento vengono quindi combinati in una sequenza temporale e inviati per il calcolo.

Gli intervalli degli eventi sono in base alla metrica locale associata, ma il parametro Time dei gestori eventi (es. OnPeriodStart (Time)) e la data/ora dell'evento di dati non elaborati è in base al valore di ora UTC. Il motore li confronta in base ai relativi valori di ora UTC, in modo da utilizzare un punto di riferimento costante.

Esempio:

Se la differenza di fuso orario di una metrica rispetto all'ora UTC è di due ore (ad esempio, GMT+2) e la data/ora in cui un evento ha aperto un incidente è alle ore 10:00, all'interno del motore l'ora utilizzata dal gestore eventi viene spostata di conseguenza e in realtà inizia alle ore 8:00 UTC. Supponendo che l'adapter è configurata per l'utilizzo di questo stesso fuso orario, quindi gli eventi di dati non elaborati verranno spostati indietro di 2 ore UTC anche nel database. Se gli eventi vengono trasferiti alla business logic, l'agente di calcolo responsabile del calcolo degli eventi per il periodo che inizia alle 10:00 utilizzerà effettivamente l'ora UTC per gli eventi, che diventa le 8:00 da quel momento in poi. Tuttavia, se si utilizza il messaggio out.log nel codice per stampare la data/ora, verrà mostrata la data/ora spostata all'ora UTC, quindi le 8:00, nonostante il periodo specificato (in base alla metrica) sia alle 10:00.

Nei requisiti di calcolo seguenti è importante convertire la data/ora dell'evento prima di utilizzarlo:

Se la metrica deve calcolare il numero di eventi chiusi nello stesso giorno in cui sono stati aperti, quindi è necessario confrontare l'ora di apertura e chiusura di ogni incidente. Se l'ora di apertura e di chiusura di un incidente sono comprese nello stesso giorno (e all'interno di un periodo di applicazione definito), l'incidente verrà contato.

Durante il processo di spostamento l'incidente all'ora UTC dall'ora locale originale, è possibile che il giorno cambi (ad esempio utilizzando GMT+2). Un incidente aperto alle ore 1:00, verrà spostato indietro alle ore 23:00 del giorno precedente in base all'ora UTC. Quindi, non verrà conteggiato anche se dovrebbe esserlo. Questa è una situazione in cui l'ora deve essere prima convertita in base all'ora locale della metrica e quindi confrontata. In questo caso, utilizzare il metodo GetLocaleTime(utcTime). Questo metodo converte una data ora specificata in un fuso orario UTC nel fuso orario della metrica locale.

Di seguito è riportato il codice del gestore eventi:

```
Sub OnIncidentEvent(eventDetails)
    If dateDiff("d",Tools.GetLocaleTime(eventDetails.time),_
        Tools.GetLocaleTime(eventDetails("ClosedAt")))=0 then
        CountIncidents=CountIncidents+1
    End If
End Sub
```

Registrazione

La registrazione è il processo con cui la business logic invia una richiesta al motore di calcolo per l'insieme di eventi di dati non elaborati affinché diventi una parte del calcolo.

Il processo di registrazione può essere gestito in due modi: tramite la procedura guidata di registrazione o manualmente tramite l'oggetto dispatcher all'interno della business logic.

La procedura guidata di registrazione è un semplice processo di selezione tra le opzioni disponibili. Sono disponibili tutte le stesse opzioni presenti durante registrazione manuale, senza la possibilità di utilizzare parametri. Se è necessario utilizzare parametri, occorre eseguire la registrazione manuale. Il flusso di base della procedura guidata richiede prima di stabilire quale tipo di registrazione si desidera eseguire, quindi di impostare i tipi di risorse e gli eventi in cui deve essere eseguita la registrazione e, infine, quale gestore eventi verrà utilizzato per elaborare gli eventi raccolti.

Una volta completate le registrazioni, verranno visualizzate nella scheda Registrazione della metrica. Nota: inoltre, è possibile avere più di una dichiarazione di registrazione per una metrica.

In effetti, la procedura guidata di registrazione utilizza la stessa funzionalità della registrazione manuale e tutte le opzioni vengono trattate nella sezione seguente.

Quando viene eseguita manualmente all'interno della business logic, la registrazione della formula è gestita dal gestore eventi *OnRegistration*. Deve essere implementato nella formula e attivato quando viene generato un evento del motore di registrazione. L'evento di registrazione viene generato una volta sola quando il contratto è attivato, quindi ogni volta che una risorsa corrispondente o un gruppo di modifiche diventa attivo. Una modifica della risorsa interessata è pertinente se riguarda gli eventi che la metrica è destinata a ricevere. Ad esempio, se la registrazione viene eseguita per Contraente (*RegisterByContractParty*), significa che tutti gli eventi del tipo definito le cui risorse sono associate al contraente della metrica sono una parte del calcolo. In questo caso, ogni volta che una nuova risorsa è associata o dissociata da tale contraente, il metodo di registrazione verrà attivato per notificare al motore la modifica.

I metodi di registrazione vengono forniti dall'oggetto dispatcher che viene passato a *OnRegistration* come argomento. I diversi metodi forniscono vari modi in cui definire i criteri di filtro in base alla definizione del tipo di evento e tutti i criteri di allocazione risorse, quali risorse di un gruppo di risorse, o risorse di un determinato tipo.

I metodi di registrazione per contraente e servizio sono altamente consigliati perché semplificano l'utilizzo della business logic come un modulo o un modello. In questo modo, il contraente e il servizio pertinente vengono recuperati dalla definizione della metrica associata e durante il riutilizzo della formula per diversi contratti e/o componenti del servizio, non è necessario modificare la registrazione.

Un altro metodo di registrazione diffuso è RegisterByResourceGroup, comodo da utilizzare con le risorse raggruppate logicamente, ma potrebbe non essere sempre associato a contraenti o a servizi. L'assegnazione delle risorse ai gruppi può in questo caso essere gestita dal catalogo delle risorse (singolarmente o tramite i gruppi di modifiche) e potrebbe anche essere aggiornata automaticamente da script di conversione.

In generale, il metodo di registrazione viene determinato durante la fase di progettazione e guidato direttamente dal modello di dati definito.

Nota: per verificare se l'oggetto dispatcher è stato utilizzato correttamente, la funzione OnRegistration viene invocata anche durante il controllo della sintassi del modulo SLALOM. Per questo motivo, non è necessario presupporre che OnLoad sia stato eseguito prima della funzione OnRegistration né utilizzare alcune delle proprietà dell'oggetto di contesto, quali "TimeUnit", "IntervalLength", "IsPeriod", "CorrectionsApply" e "ExceptionsApply" nel gestore eventi OnRegistration.

I metodi di registrazione sono inoltre responsabili dell'associazione di eventi con una procedura che verrà attivata in base alla data/ora dell'evento. La procedura definita può avere qualsiasi nome, ma ha sempre l'oggetto EventDetails (Dettagli evento) come parametro. L'oggetto EventDetails (Dettagli evento) riflette gli eventi di dati non elaborati ricevuti e contiene tutti i dettagli dell'evento come campi di dati, come illustrato nella seguente registrazione:

```
Sub OnRegistration(dispatcher)
dispatcher.RegisterByContractPartyAndService "OnMemUseEvent", "MemUse", "Server"
'the parameters of the method are: <procedure name>, <Event Type>, <Resource
Type>
End Sub
```

L'istruzione di registrazione riportata indica che gli eventi di dati non elaborati del tipo di evento "MemUse" e associati al tipo di risorsa "Server" verranno inviati al gestore eventi "OnMemUseEven" nella business logic.

La seguente procedura dovrà anche essere definita prima nella business logic:

```
Sub OnMemUseEvent(eventDetails)
If InTimeSlot And eventDetails("MemoryUsage")>MaxMemoryUse Then
    MaxMemoryUse = eventDetails("MemoryUsage")
End If
End Sub
```

Facendo riferimento all'oggetto EventDetails (Dettagli evento) e utilizzando il parametro MemoryUsage (Utilizzo memoria), l'istruzione estrae il valore del campo MemoryUsage dall'evento che è stato passato alla funzione. Questi campi sono gli stessi definiti nel tipo di evento e nei nomi dei campi viene fatta distinzione tra maiuscole e minuscole.

Registrazione delle metriche di gruppo

Le metriche di gruppo consentono di eseguire la definizione di una metrica per ogni membro di un gruppo di risorse, di applicare la stessa definizione e logica a un insieme di elementi. Un raggruppamento può essere impostato in modo statico su un insieme predefinito di risorse o in modo dinamico sui membri del gruppo di risorse mentre il gruppo può essere modificato nel tempo e includere o escludere i membri dal gruppo.

Nota: per una descrizione dettagliata, consultare la sezione Appendice D - Definizione delle formule di business logic (esperto di business logic).

Le metriche di gruppo sono utilizzate quando è necessario calcolare un risultato del livello di servizio per ogni elemento in un gruppo di risorse. Gli elementi in un gruppo di risorse possono essere risorse o altri gruppi di risorse, pertanto il metodo di registrazione in un script di business logic per una metrica di gruppo deve essere `RegisterByResourceGroup` o `RegisterByResource`, in cui il nome della risorsa o del gruppo di risorse specificato è definito come elemento nel gruppo. Questa operazione viene eseguita utilizzando la proprietà `ClusterItem` dell'oggetto di contesto che contiene il nome dell'elemento di gruppo corrente.

Esempio:

```
dispatcher.RegisterByResource "<ProcedureName>", "<Event Type name>",  
Context.ClusterItem
```

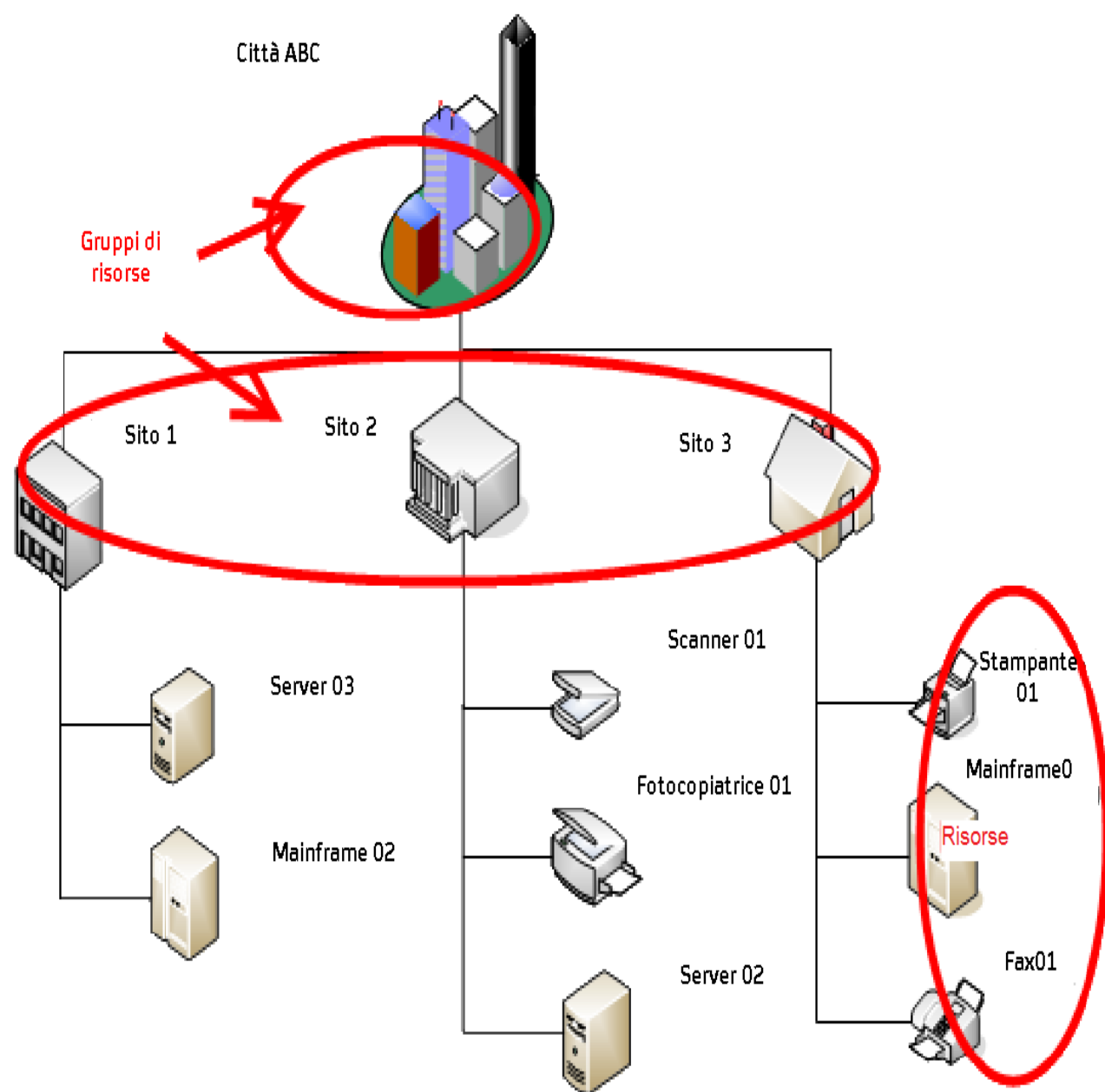
Nel caso in cui viene utilizzato questo metodo di registrazione, la metrica calcola un risultato per tutte le risorse nel gruppo di risorse in cui la metrica è di gruppo,

oppure

```
dispatcher.RegisterByResourceGroup "<ProcedureName>", "<Event Type name>",  
Context.ClusterItem
```

Nel caso in cui viene utilizzato questo metodo di registrazione, la metrica calcola un risultato per tutti i gruppi di risorse contenuti nel gruppo di risorse per cui la metrica è inserita in un gruppo.

È possibile eseguire il raggruppamento a livelli diversi a seconda del modo in cui è stato creato il modello di risorse. Spesso le organizzazioni dispongono di diversi livelli di raggruppamento che desiderano rappresentare. Ad esempio, in una determinata città, potrebbe esserci un numero di siti e all'interno di ogni sito potrebbe esserci un numero di dispositivi di infrastruttura (stampanti, scanner, server e così via). Con questo tipo di raggruppamento è possibile strutturare una gerarchia di risorse definita che contiene più livelli e raggruppamenti di questi elementi hardware, presupponendo che il dispositivo di infrastruttura sarà la risorsa. La struttura descritta potrebbe essere come segue:



Come è possibile vedere dal diagramma, ci sono più livelli di gruppi. Il gruppo di livello superiore "Città ABC" contiene tre diversi siti (che sono anche i gruppi di risorse). Il gruppo di risorse "Risorse sito 3" contiene tre diverse risorse. In base all'esempio precedente, per raggruppare le metriche nei tre diversi siti, utilizzare la registrazione seguente:

```
dispatcher.RegisterByResourceGroup "<ProcedureName>", "<Event Type name>",  
Context.ClusterItem
```

In questo caso Context.ClusterItem fa riferimento al gruppo di risorse chiamato "Siti città ABC", che contiene tre altri gruppi di risorse chiamati "Risorse sito 1", "Risorse sito 2" e così via, e può essere visualizzato come segue nella scheda Raggruppamento della metrica.

Generale	Dettagli	Note	Raggruppamento	Soglie	Metriche associate	Granularità	Dichiarazione dell'obiettivo

☐ Metrica non di gruppo.

☒ Il raggruppamento della metrica è il seguente:

Raggruppamento applicato agli elementi di:

☒ Dinamico

☒ Solo primo livello: membri diretti

☐ Tutti i livelli: include solo risorse

☐ Tutti i livelli: include risorse e gruppi di risorse

☐ Statico. Vengono inclusi solo i seguenti elementi:

Considerare inoltre che il raggruppamento è impostato su dinamico poiché così verranno incluse automaticamente eventuali modifiche apportate al gruppo. Il raggruppamento statico può essere utile per sottoinsiemi di gruppi di risorse o se non si desidera che il raggruppamento cambi nel tempo.

Per creare una metrica con report sulle risorse del gruppo del sito 3, utilizzare la seguente istruzione di registrazione:

```
dispatcher.RegisterByResource "<ProcedureName>", "<Event Type name>",  
Context.ClusterItem
```

In questo caso, Context.ClusterItem fa riferimento alle singole risorse, quindi registra solo per risorsa. La scheda Raggruppamento della metrica contiene un riferimento al gruppo "Risorse sito 3".

È possibile configurare il raggruppamento in modo che operi su diversi livelli della gerarchia all'interno di una singola metrica. Ad esempio, presupporre la situazione descritta in precedenza e raggruppare questa metrica nuovamente sul gruppo "Siti città ABC". È possibile includere in una metrica i membri della risorsa da diversi livelli della gerarchia. In questo caso, vi sono tre opzioni per includere le risorse in questo gruppo:

- **Solo primo livello: membri diretti:** uguale ai metodi di raggruppamento precedenti descritti prima.
- **Tutti i livelli: include solo risorse:** comprende tutte le risorse contenute nei gruppi di risorse dei tre siti in un unico livello e calcola il rispettivo livello di servizio singolarmente.
- **Tutti i livelli: include risorse e gruppi di risorse:** comprende tutte le risorse contenute nei gruppi di risorse dei tre siti, nonché i tre gruppi di risorse, e calcola il rispettivo livello di servizio singolarmente.

Agenti

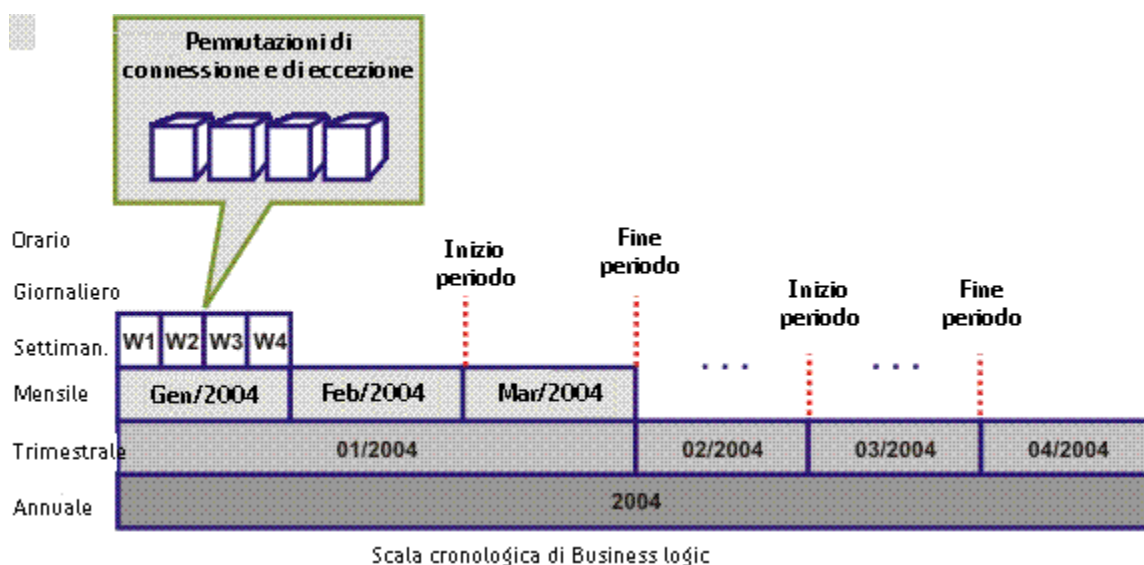
Ciascuna metrica presenta una definizione di un periodo di riferimento. Il periodo di riferimento è il periodo durante il quale la metrica deve dare l'output di un risultato del livello di servizio e tale risultato deve essere confrontato con la destinazione definita. Il risultato prodotto per il periodo di riferimento è il risultato di business, in altre parole, è il risultato contrattuale per cui il provider si è impegnato a fornire un determinato livello di servizio di destinazione. Ciascuna istanza di business logic per ogni periodo di tempo è nota come agente ed è direttamente correlata alle granularità associate a ciascuna metrica.

Ad esempio, se la metrica è definita con un periodo di riferimento di un mese, la metrica è eseguita per fornire un risultato mensile.

Per fornire la caratteristica funzionale di drill-down in cui l'utente può eseguire il drill-down in un risultato mensile per vedere il risultato giornaliero, la metrica deve avere una definizione di altre unità di tempo in cui deve essere calcolata. Le unità di tempo definite nella sezione Dettagli generali della metrica nella scheda Granularità.

Per ogni unità di tempo definita nella scheda Granularità della metrica e per il periodo di riferimento, viene eseguita una nuova istanza di business logic dal motore. Ciascuna di tali istanze esegue lo stesso codice di business logic, ma OnPeriodStart e OnPeriodEnd verranno avviati in modo diverso per ciascuna istanza. Ogni istanza giornaliera verrà attivata all'inizio del giorno e alla fine del giorno. Ogni unità di tempo verrà attivata in base all'inizio dell'unità di tempo e ai punti di fine.

Tutte le istanze di business logic vengono eseguite per generare il risultato di unità di tempo pertinente. Inoltre, ogni periodo deve avere un risultato che prenda in considerazione eventuali eccezioni. Qualsiasi periodo che non ne tiene conto, se le eccezioni sono definite, deve consentire all'utente di scegliere se visualizzare il risultato del livello di servizio con o senza le eccezioni prese in considerazione. A causa di questo requisito, ciascuna metrica potrebbe avere quattordici agenti (istanze) eseguiti dal motore, due agenti per i risultati di business e dodici per i sei ulteriori periodi operativi.



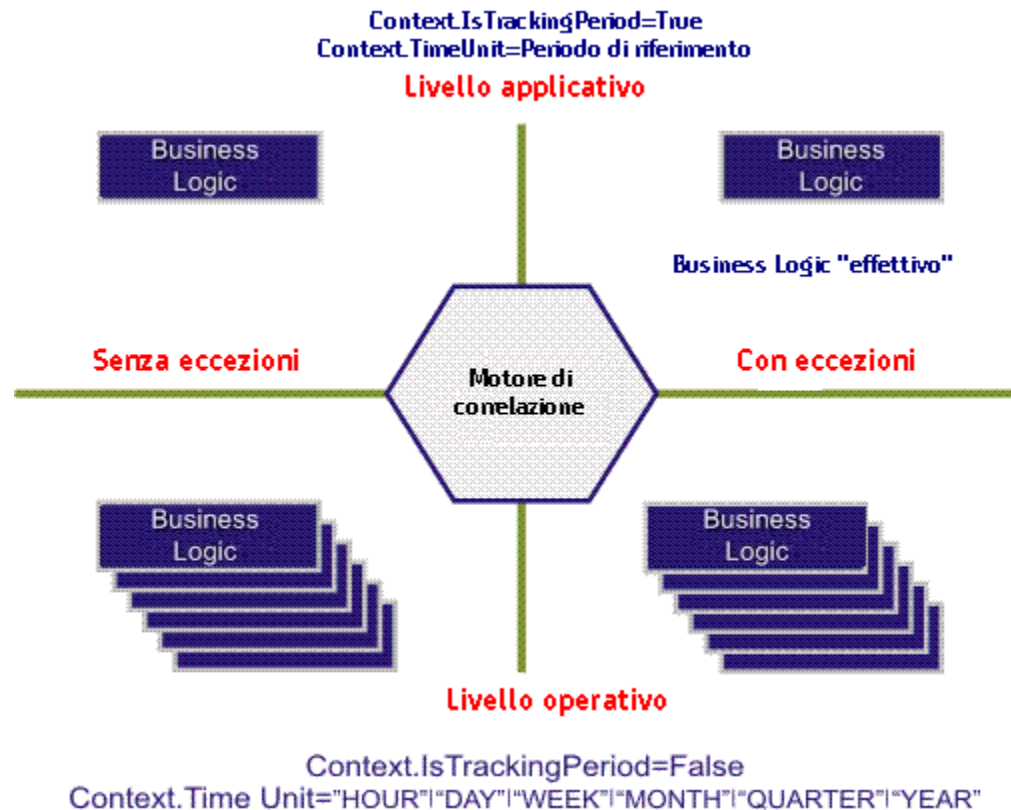
Questo significa che la definizione di granularità ha un forte impatto sulle prestazioni del motore di calcolo perché ogni periodo viene calcolato per un agente diverso. Pertanto, nei casi in cui la caratteristica funzionale di drill-down non è necessaria per intero o in parte, è consigliabile che alcuni agenti siano disattivati. Questa operazione ha un impatto particolarmente notevole nel caso di granularità inferiori quali i risultati orari. Ha anche un enorme impatto per la metrica di gruppo, poiché il motore esegue tutti i calcoli di cui sopra per ogni ClusterItem rilevato. In effetti, tratta ogni ClusterItem come nuova metrica. Ad esempio, durante il calcolo di una metrica in un gruppo di risorse di 50 elementi, il motore effettuerà un lavoro 49 volte superiore rispetto alla stessa metrica non di gruppo.

Ad esempio, se la metrica è impostata per calcolare il tempo di risoluzione in numero di giorni, il risultato orario è irrilevante e deve essere deselezionato nella scheda delle granularità per evitare al motore di eseguire calcoli inutili.

L'attributo `TimeUnit` dell'oggetto `Contesto` (ad esempio, `context.TimeUnit` nella business logic) restituisce l'unità di tempo dell'agente attualmente in esecuzione, in cui i possibili valori restituiti sono: `HOURL`, `DAY`, `WEEK`, `MONTH`, `QUARTER`, `YEAR`.

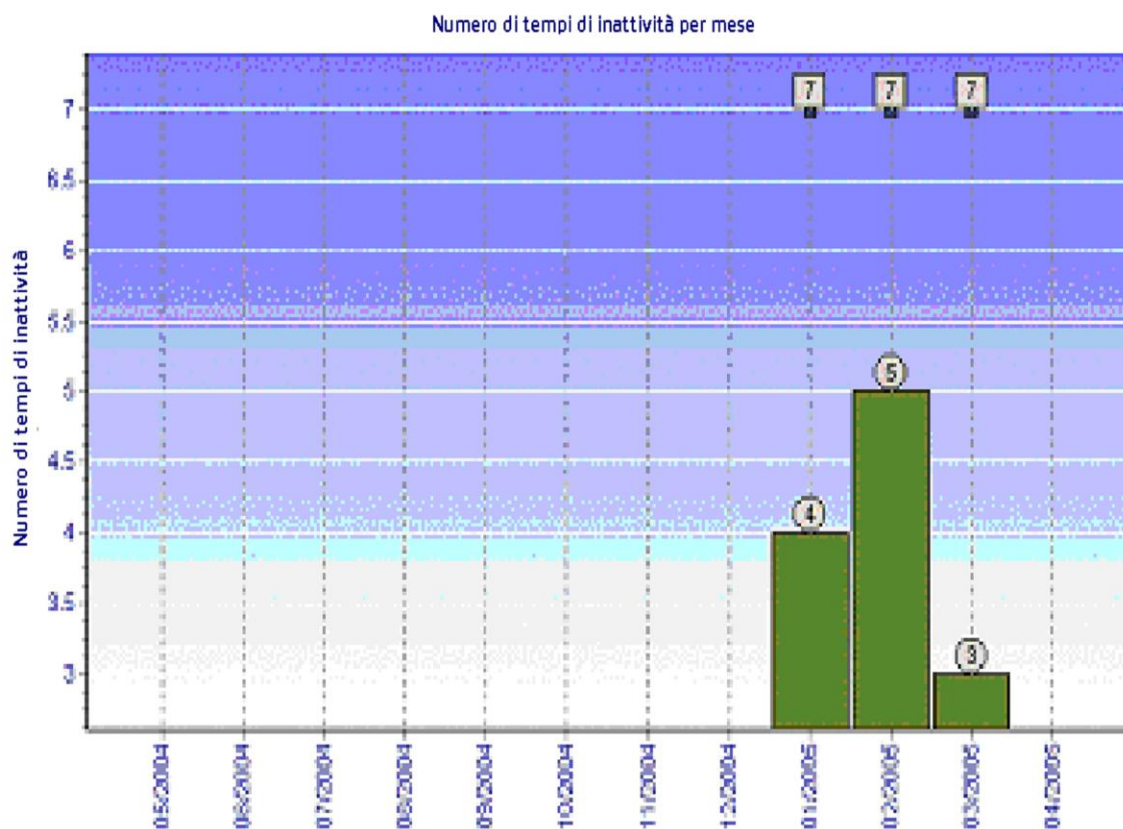
Ad esempio, per ogni agente giornaliero `Context.TimeUnit` restituirà `"DAY"`.

L'attributo `IsTrackingPeriod` dell'oggetto `Contesto` restituisce `true` per l'agente che calcola l'unità di tempo del periodo di riferimento. È importante notare inoltre che gli agenti visualizzati nella scheda `Granularità delle metriche` si aggiungono all'agente del periodo di riferimento della metrica. Pertanto, anche per una metrica con un periodo di riferimento mensile è possibile disattivare l'agente mensile e verrà comunque calcolato il livello di servizio mensile, ma solo come risultato di business (risultati non operativi).

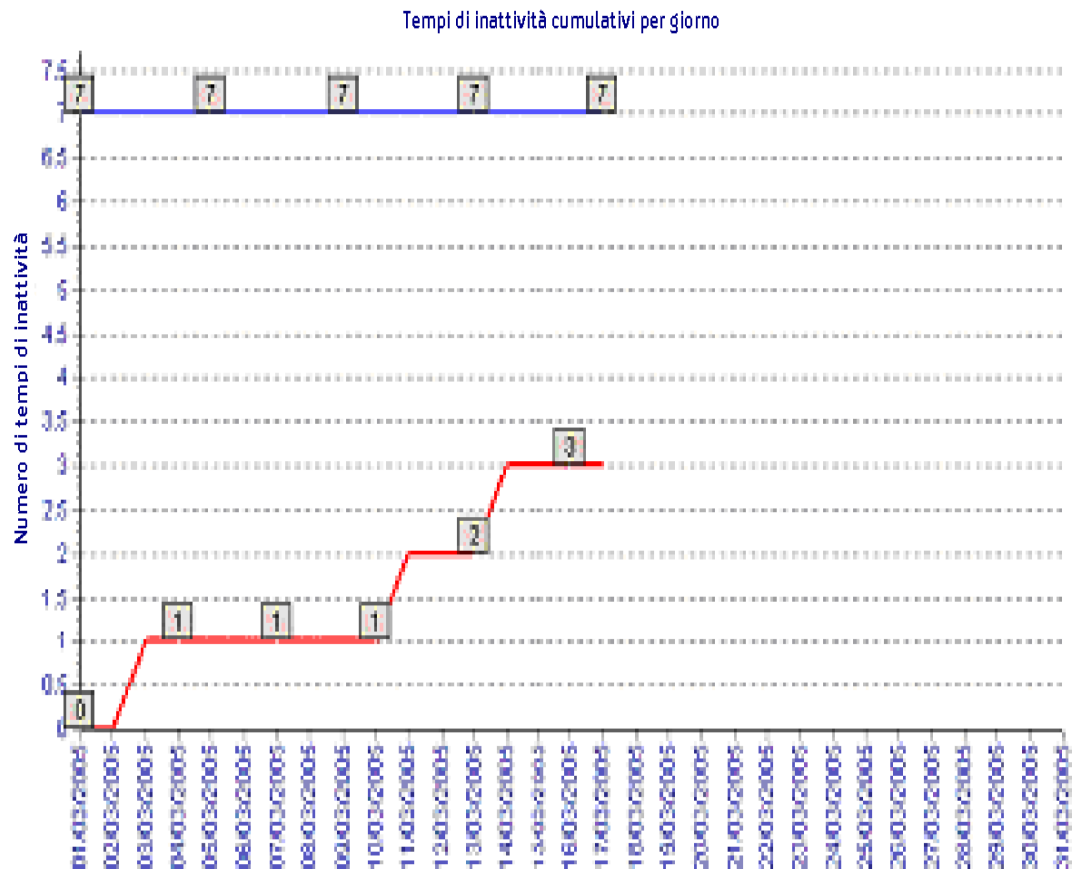


Come descritto sopra, i diversi agenti in genere eseguono lo stesso codice di business logic, ma vi sono casi in cui è necessario applicare una logica leggermente diversa.

Ad esempio, per il caso mensile, il risultato deve essere il numero di tempi di inattività per ogni mese separatamente, come mostrato di seguito:



Per il drill-down giornaliero è necessario poter visualizzare i tempi di inattività complessivi, in cui ogni giorno è la somma di tutti i giorni dall'inizio del mese. Questo metodo deve essere applicato a tutte le unità di tempo inferiori a un mese, come mostrato di seguito:



La differenza tra le due unità di tempo è che per l'agente che calcola il periodo di riferimento il contatore del tempo di inattività verrà inizializzato su 0 all'inizio di ogni periodo, ma per l'agente giornaliero il contatore verrà inizializzato solo nel caso in cui il giorno è il primo giorno del mese.

Di seguito è riportato il gestore eventi OnPeriodStart:

```
Sub OnPeriodStart(time)
    If InStr("MONTH,QUARTER,YEAR", Context.TimeUnit) > 0 _
    Or (Context.TimeUnit = "DAY" And Day(time) = 1) _
    Or (Context.TimeUnit = "HOUR" And Day(time) = 1 And Hour(time) = 0) Then
        DownTimeCounter = 0
    End If
End Sub
```

Che cosa è il ricalcolo?

Il ricalcolo viene eseguito quando il motore di correlazione identifica che un risultato periodico finale di metrica non è più valido ed è pertanto necessario calcolare nuovamente i risultati.

Il ricalcolo potrebbe essere causato da:

- Ricezione di nuovi eventi che si sono effettivamente verificati in passato (precedenti al periodo in cui il motore ha calcolato i risultati *fino al momento presente* per tale metrica)
- Modifica di una risorsa che è registrata nella metrica (nuova data di versione e modifiche apportate).
- Conferma di una nuova versione del contratto che si sovrappone al tempo calcolato in precedenza con modifiche apportate ad alcune metriche (solo le metriche modificate vengono ricalcolate)

Il metodo più efficiente di registrazione è per Contraente e servizio. La disposizione delle risorse per contraente e per servizi è un modo per esprimere la relazione logica tra il livello di dati e il livello di business nel sistema. La registrazione delle risorse attraverso queste entità non richiede la modifica delle formule se utilizzate in diversi contratti o per diversi servizi. Il contesto corrente della metrica identifica il contratto e il servizio, che definisce il contraente pertinente e il servizio associato. È possibile riutilizzare facilmente le formule di business logic definite in questo tipo di registrazione perché non richiedono modifiche nella registrazione.

Output - Tabelle utente

Lo script di business logic standard non dispone dell'accesso alle tabelle di output esterne. Esistono solo due destinazioni di output:

- La tabella PSL (T_PSL), per cui il motore scrive i risultati del livello di servizio e in cui viene scritto il risultato specificato nella funzione Result. I valori del livello di servizio scritti in questa tabella possono essere solo di tipo numerico. I risultati scritti nella tabella T_PSL sono i risultati su cui viene eseguito il report con la procedura guidata di creazione report. Nessun controllo viene eseguito sulla struttura o sul metodo con cui questi risultati vengono scritti. Questa operazione viene eseguita automaticamente dal motore di correlazione.
- La tabella di output SLALOM (T_SLALOM_OUTPUT). La scrittura in questa tabella viene eseguita utilizzando i metodi specificati dall'oggetto Tools all'interno della formula di business logic. Durante la definizione dell'output in questa tabella, viene fornito un nome di tabella logico indicato come Tabella utente. Queste tabelle sono utilizzate per le informazioni di output durante il calcolo del livello di servizio. Queste informazioni possono essere utilizzate per generare report in formato libero. Possono esistere più tabelle utente.

L'utilizzo della tabella esterna T_SLALOM_OUTPUTS è obbligatorio quando sono necessari altri output oltre il risultato periodico del livello di servizio, quando l'output aggiuntivo non è disponibile con l'aggiunta di un'altra metrica, o quando l'aggiunta di un'altra metrica riduce il calcolo delle prestazioni, utilizzando lo stesso insieme di record, solo per fornire un output diverso.

Ad esempio, considerare il caso in cui una metrica è impostata per calcolare la percentuale di ticket risolti in meno di un giorno e deve essere generato un report con l'elenco di tutti i ticket non risolti in meno di un giorno, è necessario che la formula restituisca come output ogni ticket rilevato come non riuscito in una tabella esterna, oltre ad aggiungerlo alle statistiche di calcolo.

Con questo requisito, la tabella del livello di servizio di output normale non può fornire questo output, perché:

- I risultati del servizio sono tutti numerici
- È possibile solo un risultato del livello di servizio per ogni periodo.

I record vengono scritti nelle tabelle utente di output solo per l'agente che è in esecuzione per il periodo di riferimento della metrica e che calcola eccezioni e correzioni. Ad esempio, se la metrica è definita con un periodo di riferimento mensile, quindi le istruzioni di output (Tools.SaveRecord e Tools.SaveFields) NON vengono eseguite quando il motore sta eseguendo la formula per le altre granularità quali HOUR, DAY, WEEK, QUARTER e YEAR.

Un ulteriore vantaggio di avere questa tabella esternamente è che può essere utilizzata per i requisiti di più report. Altri requisiti di report tipici possono essere generati da queste tabelle oltre ai requisiti contrattuali. Ad esempio, una metrica che calcola il numero di ticket chiusi in un mese potrebbe anche calcolare il tempo di risoluzione del ticket e fornire l'output di tutte queste informazioni nella tabella di output. In questo modo è consentita un'analisi più dettagliata dei singoli ticket chiusi durante il periodo, insieme alle informazioni aggiuntive che potrebbero essere necessarie per un requisito di report diverso.

Le informazioni contenute in tali tabelle sono inoltre gestite dal meccanismo di ricalcolo del motore. Durante il processo di ricalcolo, i record vengono contrassegnati come non attivi e viene scritta una nuova riga al loro posto. È un punto importante da notare poiché tutte le istruzioni SQL dei report in formato libero devono includere un segno di spunta nel campo IS_ACTIVE nella tabella T_SLALOM_OUTPUTS (in cui is_active = 1) dal momento che solo questi record sono attuali.

Nota: durante l'esecuzione dell'ambito di business logic nel processo di debug delle formule, i messaggi vengono in realtà scritti nella tabella T_DEBUG_SLALOM_OUTPUTS invece che nella tabella T_SLALOM_OUTPUTS.

Durante la documentazione dei dati tramite T_SLALOM_OUTPUTS, i dati inseriti sono sempre testo (poiché i campi di T_SLALOM_OUTPUTS sono tutti varchar2). Di conseguenza, i valori di data vengono convertiti in testo applicando il formato del sistema operativo che potrebbe variare durante il ciclo di vita dell'applicazione. T_SLALOM_OUTPUTS potrebbe pertanto risentire delle incongruenze nei formati di data. Inoltre, la business logic gestisce le date in UTC, mentre ci si potrebbe aspettare che T_SLALOM_OUTPUTS contenga la data/ora locale, pertanto in alcuni casi potrebbe essere necessario per utilizzare la funzione di conversione *Tools.GetLocaleDate(date)* per risolvere questo problema.

La seguente funzione converte le date in ora locale e mantiene la coerenza del formato di data formattando le date al formato "dd/mm/yyyy hh24:mi:ss":

```
Function FormatDate(time)
    Dim LocalTime
    LocalTime=Tools.GetLocaleTime(time)
    FormatDate=Day(LocalTime) & "/" & Month(LocalTime) & "/" &
    Year(LocalTime) & " " & _
    Hour(LocalTime) & ":" & Minute(LocalTime) & ":" & Second(LocalTime)
End Function
```

Esistono due metodi che possono essere utilizzati per scrivere nella tabella esterna dall'interno della formula di business logic:

- Tools.SaveRecord <parameter list>
- Tools.SaveFields <parameter list>

Entrambi i metodi dell'oggetto Tools sono descritti di seguito in modo più dettagliato:

Tools.SaveRecord tableName, key, [val1], [val2], ...

Questo metodo consente di salvare un record in una tabella denominata T_SLALOM_OUTPUTS. Il parametro tableName (Nome tabella) specifica la tabella (virtuale) all'interno di T_SLALOM_OUTPUTS in cui devono essere scritte le informazioni. Ogni record nella tabella utente ha una chiave univoca che specifica il record in cui devono essere scritte le informazioni. Inoltre, ogni record presenta fino a 20 campi con valore di tipo stringa. Il metodo SaveRecord riceve un nome di tabella utente e una chiave. Accetta inoltre tutti i campi valore nella tabella utente. Questi parametri di valore sono facoltativi e possono essere omessi. Se un record con la stessa chiave esiste già, verrà aggiornato. Solo i campi valore trasferiti come parametri vengono aggiornati. Se un record con questa chiave non esiste, verrà creato.

```
Tools.SaveFields tableName, key, [fieldName1,fieldVal1], [fieldName2,fieldVal2]
```

Questo metodo è simile a SaveRecord, ma invece di enumerare tutti i valori, fornisce coppie di nomi di campo e i relativi valori di campo. I numeri del campo possono sostituire i nomi di campo. I nomi di campo devono essere definiti manualmente in precedenza nella tabella T_SO_FIELD_NAMES. Questa tabella viene utilizzata per registrare la struttura delle tabelle di output.

È consigliabile che l'esperto di business logic definisca la struttura della tabella di output prima della scrittura in T_SLALOM_OUTPUTS, perché così la struttura e il significato dei campi sono già ben definiti e renderanno l'esecuzione della query molto più semplice.

La struttura della tabella comprende:

- **table_name:** ogni tabella logica ha un nome univoco
- **field_name:** ogni campo in una tabella è univoco
- **field_id:** ogni campo è numerato con un numero di serie a partire da 1

L'utilizzo del metodo SaveFields è preferibile poiché conserva la documentazione della struttura e il significato dei valori inseriti.

Esempio:

Considerare un caso in cui è necessario generare un elenco di tutti gli incidenti con il tempo di risoluzione superiore alla soglia definita, oltre al risultato della metrica che calcola la percentuale di ticket risolti entro tale soglia. La scrittura nelle tabelle di output tabelle verrà eseguita nella procedura del gestore eventi OnXXXEvent dopo la convalida della soglia.

Viene illustrato nell'esempio seguente:

```
Sub OnIncidentEvent(eventDetails)
```

```
    If eventDetails("RESOLUTION_TIME")<=SThreshold Then
        CountSuccessIncident s= CountSuccessIncidents+1
    Else
        building the record unique key
        KeyString= eventDetails("IncidentID") &eventDetails.Time
```

```
Tools.SaveFields "IncidentsTable", KeyString, "CASE_ID",  
eventDetails("CASE_ID"),_  
"CUSTOMER_REF",eventDetails("CUSTOMER_REF"),_  
"TICKET_CREATOR",eventDetails("TICKET_CREATOR"),_  
"CREATION_TIME",eventDetails("CREATION_TIME"),_  
"SEVERITY",eventDetails("SEVERITY"),_  
" RESOLUTION_TIME ",eventDetails("RESOLUTION_TIME "),_  
"CLOSE_TIME",eventDetails("CLOSE_TIME")  
End If  
End Sub
```

Di seguito sono riportati alcuni suggerimenti relativi all'uso delle tabelle
T_SLALOM_OUTPUTS:

- Si consiglia di scrivere solo i dati necessari nelle tabelle di output (non di più).
- Scrivere nelle tabelle di output solo per le granularità di metrica necessarie.
- Le dimensioni di tutti i campi valore sono solo di 50 caratteri per impostazione predefinita (eccetto la colonna VAL_1 che è di 512). Potrebbe essere necessario troncare alcuni valori per i campi per assicurarsi che si adattino alle dimensioni dei dati; in caso contrario, vengono restituiti errori di runtime nella business logic.
- Sono disponibili solo 20 colonne per i dati (VAL_1 ... VAL_20)

Nota: la scrittura nelle tabelle di output può avere effetto sulle prestazioni del motore, poiché la scrittura di una tabella è un'operazione computazionale complessa, rispetto a un calcolo in memoria. Pertanto, considerare con attenzione se e in quali casi è necessario utilizzare questa funzionalità, quindi ridurre al minimo il numero di volte in cui si esegue l'accesso alle tabelle.

Report sulle informazioni dalle tabelle definite dall'utente

Non è possibile utilizzare la procedura guidata di creazione report CA Business Service Insight standard per i report sulle informazioni scritte nelle tabelle di output. Per generare report su queste informazioni è necessario creare un report in formato libero. In sostanza, implica la creazione di una query SQL nella parte superiore di questa tabella. Siccome la tabella contiene molte tabelle logiche che vengono scritte da varie formule, è consigliabile che un utente che abbia dimestichezza con SQL (esperto delle sorgenti dati) crei una visualizzazione di T_SLALOM_OUTPUTS in modo da distinguere facilmente le diverse tabelle logiche contenute e verificare inoltre che le informazioni siano recuperate come pianificato.

La definizione della visualizzazione avrà già tutto l'insieme dei campi di tipo stringa per le informazioni di tipo pertinente e conterrà inoltre tutti i filtri necessari (tabella logica, record attivi, metrica pertinente, ecc.).

Di seguito è riportato un esempio di creazione della visualizzazione:

```
create or replace view kpi_view as  
select  
to_date(t...) as fieldName,
```

```
        to_number(t..), ...
from t_slalom_outputs t,
     t_rules r,
     t_sla_versions sv,
     t_slas s,
where table_name = 'TableName'
      and is_active = 1
      and t.rule_id = r.psl_rule_id
      and r.sla_version_id = sv.sla_version_id
      and sv.sla_id = s.sla_id
      and sv.status = 'EFFECTIVE'
```

Creazione dei moduli di business logic

Gli utenti possono definire moduli di business logic indipendenti utilizzabili da più obiettivi del livello di servizio (metriche). Per applicare modifiche alla business logic a livello di sistema, l'utente modifica il componente logico di base che può essere quindi applicato a tutti gli accordi SLA pertinenti con un solo clic.

Un modulo di business logic è un componente di codice in cui la business logic può essere definita e gestita facilmente, riducendo la ridondanza. Un singolo modulo può essere utilizzato da più metriche.

Durante la fase di configurazione, le formule sono configurate per definire i moduli di business logic principale (consultare il capitolo Progettazione alla sezione Modelli e moduli di business logic).

Esistono tre tipi di moduli di business logic:

- **Business Logic completo:** impiegato quando è necessario utilizzare una formula intera come un modulo. I metodi Result e OnRegistration devono essere implementati nello script del modulo.
- **Classe:** modulo che contiene una definizione di una singola classe VB.
- **Libreria:** modulo che contiene un insieme di procedure, funzioni e classi.

I moduli possono essere utilizzati dai seguenti elementi:

- Metriche
- Altri moduli
- Modelli di business logic
- Metriche all'interno dei modelli di contratto e dei modelli del livello di servizio.

I moduli possono utilizzare i parametri attivati da Parameters("ParamName") del contesto della metrica.

Nota: per evitare errori di runtime, impostare sempre un valore predefinito quando si utilizzano i parametri nei moduli di business logic. La formula genera un messaggio di log errore per i parametri inesistenti.

```
If Parameters.IsExist("ReportedDowntimesNum") Then
    maxBufferSize = Parameters("ReportedDowntimesNum")
Else
    maxBufferSize = 3
    out.log "ReportedDowntimesNum parameter not set", "E"
End If
```

Esempio di modulo di business logic

È presente un oggetto del livello di servizio definito come "attività dell'assistenza tecnica entro la soglia". Il seguente sistema di assistenza tecnica presenta un ciclo di vita dei ticket con stati:

- Aperta
- Assigned (Assegnato)
- Chiuso.

Le due metriche che possono essere definite per descrivere le prestazioni dell'assistenza tecnica sono:

- **Nome metrica: Successful Ticket resolution on time.**

Dichiarazione dell'obiettivo: è necessario risolvere entro 4 ore almeno il 99% dei ticket.

Business logic: la risoluzione deve essere calcolata dallo stato aperto allo stato chiuso.

- **Nome metrica: Successful ticket assignment on time.**

Dichiarazione dell'obiettivo: è necessario assegnare entro 30 minuti almeno il 99% dei ticket.

Business logic: l'assegnazione deve essere calcolata dallo stato aperto allo stato assegnato.

Poiché la stessa logica può essere identificata per entrambe le metriche, è possibile creare un modulo per soddisfare entrambe le metriche.

Il modulo richiede i seguenti parametri nel contesto di metrica:

- First status (Primo stato)
- Second status (Secondo stato)
- Threshold (Soglia).

Una volta creato il modulo di business logic, le metriche definite possono utilizzare un modulo come parte della definizione.

In seguito è possibile modificare la logica. Ad esempio, è necessario considerare un nuovo stato Customer Pending (In attesa del cliente). Lo stato Customer Pending (In attesa del cliente) viene impostato per un ticket quando l'assistenza tecnica è in attesa di ulteriori informazioni sul ticket dal cliente. All'interno della business logic, per la durata del periodo in cui il ticket è in attesa del cliente, non deve essere considerato come parte del calcolo. Pertanto, solo il modulo di business logic deve essere modificato per riflettere il nuovo stato e la relativa logica. Viene creata una nuova versione del modulo che comprende la nuova logica.

Durante la conferma di una modifica, viene visualizzata una richiesta con un elenco delle metriche che utilizzano il modulo. È possibile applicare la modifica a tutte le metriche insieme o selezionare di applicare la modifica solo a metriche specifiche nell'elenco.

Se si selezionano solo metriche specifiche dall'elenco, viene richiesto di creare un nuovo modulo per le metriche selezionate. Il vecchio modulo utilizzato per le metriche selezionate viene sostituito dal nuovo modulo di business logic e il ricalcolo viene eseguito secondo la nuova logica.

Creazione dei parametri

I parametri offrono agli utenti di business un modo rapido e intuitivo con cui visualizzare e modificare i propri valori tramite l'interfaccia utente grafica (GUI) senza dover modificare lo script della formula.

L'utilizzo di parametri all'interno della business logic consente la creazione di formule generali che possono avere un ampio uso nel sistema e permette di ottimizzare l'utilizzo dei moduli.

I parametri possono essere definiti a livello di contratto o a livello di metrica. I parametri della metrica vengono visualizzati e configurati nella scheda Dichiarazione dell'obiettivo della pagina Dettagli metrica. La business logic dispone ha accesso solo ai parametri della metrica; pertanto, per accedere a un parametro contrattuale dall'interno di una metrica, viene creato un altro tipo di parametro localmente all'interno della metrica. Viene chiamato parametro dinamico e ottiene il valore come riferimento dai parametri contrattuali. I valori di riferimento consentiti nel parametro dinamico sono solo quelli definiti nel contratto padre della metrica.

Tipi di parametri:

- Date (Data): scelta obbligatoria data (data + ora).
- Number (Numero): dimensioni massime di 15 cifre con una precisione massima di 5 cifre.
- Text (Testo): dimensioni massime di 100 caratteri.
- Table (Tabella): dimensioni massime di 120 voci.

Per accedere ai valori del parametro dal codice della formula, è necessario utilizzare l'oggetto Parameters e fare riferimento al nome di parametro.

Esempio:

```
Parameters("Threshold")
```

(Nota: è un metodo rapido per invocare un valore; normalmente, questa operazione viene eseguita come `Parameters.Item("Threshold")`).

Oppure, per un parametro di tipo tabella:

```
Parameters("Table")("Word")("Price")
```

(in cui "Word" e "Price" valori sono rispettivamente i nomi di riga e di colonna del parametro di tabella).

I parametri di tabella devono essere utilizzati solo in casi particolari:

1. Definire una variabile globale (ad esempio, `dim myTableParam`)
2. Nella funzione `OnLoad`, compilare la variabile dall'oggetto parametri (ad esempio, `"Set myTableParam = Parameters("myParametersTable")`).
3. In seguito utilizzare solo l'oggetto creato, `myTableParam`. Il parametro se stesso non deve essere utilizzato al di fuori della funzione `OnLoad` e deve fare riferimento solo all'oggetto di variabile globale creato da esso.
(ad esempio, `"dim myVal: myVal = myTableParam ("myRow")("myColumn")`).

Questa operazione consente di liberare la memoria che il parametro richiede e impedisce al motore di creare la mappa del parametro all'invocazione di ogni parametro e per ogni `OnXXXEvent`, che può essere invocato migliaia di volte per metrica.

Il codice riportato di seguito illustra l'utilizzo corretto di un parametro di tabella:

```
Option Explicit
```

```
Dim sum
```

```
Dim myParamTable
```

```
Sub OnLoad(TIME)
```

```
    Set myParamTable = Parameters("MyTableParam")
```

```
End Sub
```

```
Sub OnRegistration(dispatcher)
```

```
    dispatcher.RegisterByResource " OnEvent", "EventType", "ResourceType"
```

```
End Sub
```

```
Sub OnPeriodStart(TIME)
```

```
    sum = 0
```

```
End Sub
```

```
Sub OnEvent(eventDetails)
```

```
    If Context.IsWithinTimeslot Then
```

```
        sum = sum + 1 * myParTimeSlotamTable("firstRow")("secondColumn")
```

```
    End If
```

```
End Sub
```

```
Function Result
    Result = ( sum * myParamTable("secondRow")("thirdColumn") )
End Function
```

I seguenti metodi sono disponibili per ogni oggetto di parametro creato nel codice.

Parameter	Description
IsExist	È un parametro esistente. Produce un errore per i parametri del contratto.
IsNumeric	È un parametro di tipo numerico.
IsText	È un parametro di tipo testo.
IsDate	È un parametro di tipo data.
IsTable	È un parametro di tipo tabella.
IsEntryExist	È una voce esistente nella tabella.
IsEntryNumeric	È una voce nella tabella di tipo numerico.
IsEntryText	È una voce nella tabella di tipo testo.
IsEntryDate	È una voce nella tabella di tipo data.
Dump	Restituisce un elenco di tutti i parametri.
Item	Fa riferimento al valore del parametro.

Implementazioni delle destinazioni dinamiche

Le destinazioni dinamiche sono gestite dalla business logic tramite un gestore eventi nello script di business logic standard, simile alla funzione Result() che viene utilizzata per restituire il valore del livello di servizio dalla metrica. La destinazione dinamica deve essere specificata nella scheda Dettagli metrica come mostrato di seguito.

The screenshot shows the 'Dettagli metrica' (Metric Details) form. The form is divided into several sections. The 'Obiettivo del livello di servizio' (Service Level Objective) section includes a dropdown for 'Valore di destinazione' (Destination value) and a checkbox for 'Destinazione dinamica' (Dynamic destination), which is checked and circled in red. Other fields include 'Indicatore principale' (Main indicator), 'Servizio' (Service), 'Dominio del servizio' (Service domain), 'Categoria di dominio' (Domain category), 'Unità di misura' (Unit of measure), 'Periodo di applicazione' (Application period), 'Fuso orario' (Time zone), 'Periodo di riferimento' (Reference period), 'Unità di consumo' (Consumption unit), 'Prezzo unitario' (Unit price), 'Calcolo previsione' (Forecast calculation), and 'Previsione' (Forecast). Buttons for 'Aggiungi nuovo' (Add new) and 'Visualizza dettagli' (View details) are also present.

Quando è specificata una destinazione dinamica, la destinazione è ottenuta dalla funzione Target() nella business logic, al posto del valore statico specificato nella scheda Dettagli della metrica. La funzione Target è simile alla seguente.

Funzione Target

```
'TODO: AGGIUNGERE il codice qui PER gestire il calcolo della destinazione  
dinamica  
Target = Null  
End Function
```

Questa funzione deve essere implementata in base ai requisiti della metrica per restituire il valore di destinazione desiderato per un periodo specifico. La funzione può restituire qualsiasi numero che la business logic può assegnare.

Esempio concreto di destinazioni dinamiche

Per un call centre, la destinazione per una metrica che misura il *tempo medio di accettazione della chiamata* potrebbe dipendere dal volume di chiamate. Se sono presenti da 0 a 800 chiamate, la destinazione deve essere inferiore ai 15 secondi; se sono presenti da 801 a 1500 chiamate, la destinazione deve essere inferiore ai 20 secondi; se sono presenti più di 1500 chiamate, la destinazione deve essere inferiore ai 25 secondi. È possibile la seguente implementazione: (supponendo che TotalCalls è un contatore incrementato per ogni evento chiamata ricevuto e che TotalCalls non può essere minore di 0)

Funzione Target

```
If TotalCalls >0 and TotalCalls <= 800 Then  
    Target = 15  
ElseIf Total Calls > 800 and TotalCalls <= 1500 Then  
    Target = 20  
Else  
    Target = 25  
End If  
End Function
```

Altro esempio di utilizzo delle destinazioni dinamiche

Considerare la situazione in cui la destinazione per una metrica può variare a seconda della granularità del calcolo. Potrebbe essere il caso in cui è presente una destinazione giornaliera con disponibilità al 98% per un gruppo di server, ma la destinazione mensile è con disponibilità al 99,5%. La soluzione per questo caso richiede l'utilizzo della funzione di destinazione dinamica associata all'invocazione di *Context.TimeUnit* per determinare l'agente attuale in fase di calcolo. Quindi è possibile adeguare la destinazione di conseguenza.

Funzione Target

```
If Context.TimeUnit = "DAY" Then  
    Target = 98  
ElseIf  
    Target = 99,5  
End If  
End Function
```

Backup degli stati

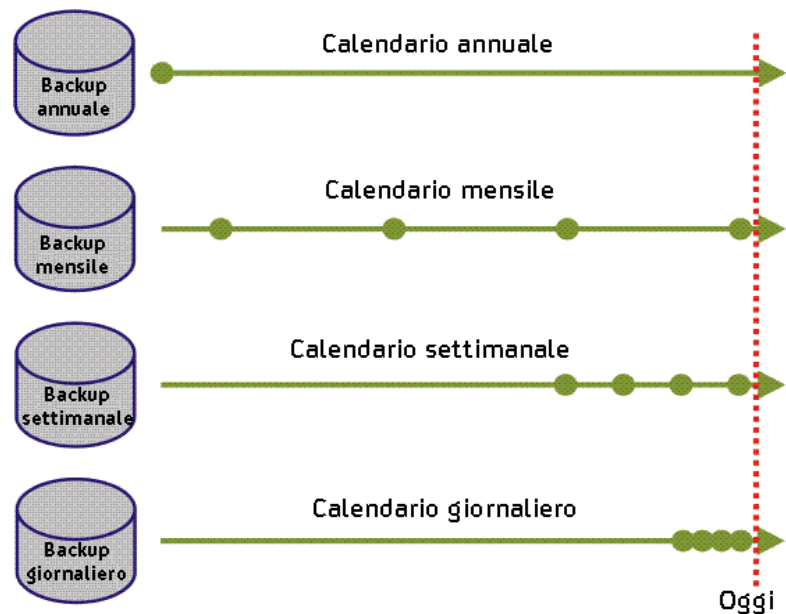
Durante il continuo processo di calcolo dei livelli di servizio per ciascuna metrica, il motore è spesso costretto a eseguire un calcolo parziale per un periodo non ancora completato. Per assicurare che non sia necessario tornare all'inizio del calcolo quando nuovi dati arrivano in ritardo, esegue un tipo di backup del proprio stato attuale prima di passare alla prossima attività di calcolo. A questo punto effettua una snapshot delle variabili e dei valori correnti a quel punto nel calcolo e salva questo stato nel database.

Il processo di backup della business logic è un meccanismo per cui il codice di business logic, inclusi i valori delle variabili, è codificato in un flusso binario e salvato nel database. Questo meccanismo è anche necessario per accelerare le prestazioni del motore di calcolo nei casi di ricalcolo. Viene eseguito il backup dello stato periodicamente e viene utilizzato nel ricalcolo e come misura di efficienza per i calcoli continui.

Ad esempio, se è richiesto un ricalcolo per un mese retroattivo, invece di ricalcolare i risultati a partire dall'inizio del contratto, viene utilizzato il backup di stato più recente prima della data di ricalcolo e i calcoli vengono eseguiti da tale stato in avanti.

Il motore di calcolo utilizza l'euristica predefinita per determinare se è necessario il backup e utilizza la caratteristica funzionale di backup per archiviare lo stato codificato nel database.

Nella figura seguente, i punti rossi rappresentano un backup di stato. Tanto più è indietro nel tempo, tanto minore è il numero di backup degli stati che vengono presi in considerazione. La logica alla base di questo meccanismo è il presupposto per cui il ricalcolo è in genere necessario per il periodo di tempo più recente rispetto al mese precedente.



Ottimizzazione del sistema di ricalcolo

Il processo di calcolo del livello di servizio richiede una notevole quantità di risorse della CPU, di memoria e di archiviazione. Di seguito è riportato un elenco di suggerimenti che possono ridurre il carico del computer e migliorare la velocità di calcolo.

Nota: alcune di queste indicazioni richiedono impostazioni interne non disponibili nell'interfaccia utente. In questi casi, contattare il Supporto tecnico di CA per ulteriori informazioni e istruzioni.

■ Modificare la configurazione di salvataggio degli stati

A seconda dei ritardi noti dell'adapter, è possibile impostare i parametri di stato per adattarsi al meglio alle proprie esigenze; pertanto, è possibile modificare il numero e la granularità dei punti di salvataggio degli stati.

■ Configurare il sistema per calcolare solo le unità di tempo realmente necessarie

Le metriche possono avere al massimo sette periodi di granularità diversa: *anno*, *trimestre*, *mese*, *settimana*, *giorno*, *ora* e il periodo di riferimento. In alcune implementazioni, non sono necessarie tutte le granularità. È possibile disattivare la granularità non necessarie per i contratti non confermati tramite l'interfaccia utente. Fare riferimento alla scheda delle granularità di ciascuna metrica. Per modificare le granularità di metrica per i contratti confermati, contattare il Supporto tecnico di CA.

Nota: evitare di calcolo dei periodi operativi che sono analoghi al periodo di riferimento.

■ Non calcolare i valori "senza correzione", "senza eccezioni"

Per impostazione predefinita, il motore di calcolo calcola quattro diversi valori: *fornito*, *fornito con correzioni*, *fornito con eccezioni* e *fornito con correzioni ed eccezioni*. È possibile apportare modifiche per calcolare solo il valore fornito.

Nota: per ulteriori informazioni, contattare il Supporto tecnico di CA.

■ Modificare l'ordine di calcolo

L'ordine di calcolo PSL delle metriche può essere definito con priorità per iniziare con le metriche più importanti e quindi continuare con le altre metriche.

Nota: per ulteriori informazioni, contattare il Supporto tecnico di CA.

■ Creare più istanze PslWriter

Con la creazione di più istanze PSL (del motore di calcolo), è possibile dividere il carico di lavoro e aumentare la velocità di calcolo.

Nota: per ulteriori informazioni, contattare il Supporto tecnico di CA.

■ Pianificare le implementazioni per ridurre il tempo di ricalcolo

Per ottimizzare il tempo di ricalcolo, è possibile:

- Eseguire l'adapter più volte per ridurre gli eventi posticipati ed evitare backlog di grandi dimensioni dei dati da elaborare con il motore

- Disattivare gli agenti inutilizzati nella scheda di granularità delle metriche.
- Duplicare le metriche e calcolare diversi agenti utilizzando la stessa metrica, per bilanciare il carico di calcolo
- Utilizzare le metriche intermedie per eseguire calcoli comuni e condividere i risultati con tutte le altre metriche che richiedono gli stessi dati.

■ **Pianificare le implementazioni per ridurre la quantità di dati**

Per ottimizzare la quantità di dati, utilizzare l'adapter per caricare solo dati aggregati (elaborati). L'aggregazione delle informazioni dell'origine dati prima che siano inviate alla tabella dei dati non elaborati CA Business Service Insight aumenta l'efficienza di lettura di input PSL.

■ **Seguire i consigli di configurazione PSL CA Business Service Insight**

Il motore PSL può essere riconfigurato per migliorare le prestazioni in base all'ambiente e ai requisiti specifici dell'applicazione. Alcuni parametri possono essere impostati dall'interfaccia utente e altri possono essere impostati solo dalla tabella di configurazione del sistema del database.

Nota: consultare le indicazioni di supporto per le proprie configurazioni PSL.

Log e notifiche

Ci sono i casi in cui la business logic è obbligatoria per segnalare report al log o attivare un messaggio di notifica. È necessaria quando i messaggi devono essere inviati in base all'elaborazione dell'evento. È possibile inviare le informazioni raccolte durante il processo di calcolo e che possono essere utili come una notifica. Ad esempio, è possibile inviare un messaggio di notifica quando un evento specifico è sotto la soglia del tempo di risoluzione o nell'analisi delle tendenze quando è stato raggiunto un certo numero di tentativi consecutivi non riusciti.

Out è un oggetto di business logic globale che consente alla formula di inviare le notifiche e i messaggi di log. Dispone di due metodi associati che si presentano come segue:

`Alert(<Event type>, <Resource name>, <value1, value2>, ...<value16>)`

Questo comando invia una notifica di un determinato tipo di evento. Tuttavia, questo tipo di evento deve essere creato manualmente ai fini di questa notifica. Il numero e il tipo di valori devono corrispondere alla definizione del tipo di evento.

`Log(<Message>, <Level>)`

Questo comando invia un messaggio al log di sistema. Il primo parametro è il messaggio informativo rilevato e può essere in testo libero. È inoltre possibile aggiungere i valori delle variabili a questa stringa per fornire significato contestuale al messaggio. Il parametro Level (Livello) può assumere uno dei seguenti valori:

Descrizione

Viene segnalato un messaggio di avviso.

Viene segnalato un messaggio di errore.

Viene segnalato un messaggio informativo solo quando è in esecuzione nell'ambito di business logic. Durante l'esecuzione in PslWriter, nessun messaggio viene segnalato. Si tratta del livello predefinito. Viene utilizzato principalmente per motivi di debug.

Esempio:

Il seguente esempio è tratto da un caso in cui le informazioni sull'infrastruttura dell'evento sono state previste prima dei dettagli dell'incidente effettivo. È stato configurato un meccanismo di notifica per comunicare all'amministratore questa condizione per richiedere la correzione del problema.

`Out.Alert "Site Unknown Alert", Context.ClusterItem, Context.Rule`

```
Out.Log("Fault Event Received for a Site with no infrastructure details: " &  
Context.ClusterItem)
```

Commenti relativi alla causa della violazione e commenti relativi agli eventi

Il commento relativo alla causa della violazione può essere impostato nella business logic per spiegare i risultati del livello di servizio. I commenti relativi alla causa della violazione sono associati alle metriche.

È inoltre possibile generare le note dell'evento che sono commenti associati agli eventi di dati non elaborati, invece che alla metrica. Entrambi i tipi di commento possono essere visualizzati dai dati del report.

Due metodi dell'oggetto Tools di business logic consentono la creazione di record relativi alla causa della violazione e alle note dell'evento:

- **Tools.AddRootCauseComment** (Text, Timestamp, [resourceId])
- Salva un commento relativo alla causa. Queste informazioni possono essere utili in seguito nei report generati. Il commento salvato relativo alla causa descrive una situazione specifica durante l'esecuzione della formula di business logic in un momento determinato. Il parametro Information (Informazioni) indica che il commento deve essere scritto. Il metodo riceve la data/ora da salvare con il commento. Accetta inoltre un parametro ResourceId (ID risorsa) che specifica la risorsa associata al contesto di metodo. Questo parametro è facoltativo e può essere omesso.
- **Tools.AddEventAnnotation** (EventId, Text)
- Questi metodi possono essere utilizzati in qualsiasi punto all'interno della business logic e il contesto in cui vengono applicati deve essere valutato. L'aggiunta di un commento relativo alla causa potrebbe essere più pertinente alla fine di un periodo di calcolo, quando è noto il motivo per cui il livello di servizio deve essere inferiore rispetto a quanto previsto per quel periodo. Presupporre, ad esempio, che durante un mese si verificano quattro interruzioni, tutte causate da un singolo dispositivo. Il commento relativo alla causa può quindi essere compilato con alcune informazioni sulle interruzioni, in modo che quando i report vengono visualizzati per questo periodo è possibile vedere che cosa ha contribuito alla violazione di un livello di servizio, durante tale periodo. Il comando AddRootCauseComment è adatto per la routine del gestore eventi OnPeriodEnd o un'altra funzione analoga eseguita verso la fine del periodo calcolato.
- Tuttavia l'aggiunta di una nota dell'evento è più adatta per l'elaborazione di eventi di dati non elaborati e preferisce l'utilizzo in OnXXXEvent (il gestore eventi personalizzato specificato nell'istruzione di registrazione). All'interno di questo gestore eventi, tutti i campi specifici all'evento effettivo sono disponibili tramite l'oggetto eventDetails.
- Di seguito è riportato un esempio della possibile visualizzazione all'interno della business logic:

```
Sub OnPeriodEnd(Time)
    pctAvailable = (TotalTime-OutageTime) / TotalTime
    Tools.AddRootCauseComment "Violations caused by the
    following items: " & ViolationCollection, Time
End Sub
```

```
...  
...  
Sub OnIncidentEvent(eventDetails)  
    OutageTime = OutageTime + eventDetails("TimeToResolve")  
  
    If eventDetails("TimeToResolve") > 6 Then  
        ViolationCollection = ViolationCollection &  
eventDetails("HardwareId")  
        Tools.AddEventAnnotation eventDetails.EventId, "Incident " &  
eventDetails("IncidentId") & " not resolved within target  
time 6 hours"  
    End If  
End Sub
```

Separazione delle eccezioni dai periodi di applicazione

La business logic CA Business Service Insight non riceve eventi di eccezione. Riceve invece OnTimeslotExit quando inizia un periodo di eccezione e OnTimeslotEnter quando termina un periodo di eccezione. La business logic pertanto non è grado di distinguere tra periodi di eccezione e periodi esterni al periodo di applicazione. Inoltre, non è in grado di distinguere tra i tipi di eccezione. Di conseguenza, non è possibile implementare la logica diversa per il comportamento nel periodo di eccezione e per il comportamento nel periodo esterno al periodo di applicazione.

Un modo di implementazione delle eccezioni speciali (vale a dire, un'eccezione che non si comporta come un periodo esterno al periodo di applicazione) consiste nella definizione di tipi di evento dedicati, invece di utilizzare il meccanismo integrato di CA Business Service Insight per la gestione delle eccezioni. Questi eventi vengono generati in seguito a lettura da un'origine dati dedicata, utilizzando un adapter.

Un foglio di calcolo di Excel (o qualsiasi altra origine dati) può archiviare queste eccezioni, quindi un adapter può caricare i dati e generare una risposta: eventi Exception Enter (Inserimento eccezione) e Exception Exit (Chiusura eccezione). In alternativa, le eccezioni possono essere aggiunte utilizzando le correzioni. Oltre alla correzione, è necessario definire una risorsa fittizia e associarla a tali eventi ai fini della registrazione. Questa risorsa serve solo allo scopo di segnaposto come richiesto dal comando.

Per essere in grado di gestire i periodi di eccezione riportati da questi eventi dedicati, la formula di business logic deve registrare questi eventi di eccezione oltre alla registrazione normalmente necessaria degli eventi di dati non elaborati da utilizzare nel calcolo.

È consigliabile che l'esperto di business logic includa un campo per il tipo di eccezione nel tipo di evento, consentendo la diversa gestione dei vari tipi di eccezioni speciali.

Questo approccio presenta le seguenti caratteristiche:

- Separa due insiemi di eccezioni, quelle standard e quelle speciali.
- Le eccezioni speciali non dispongono di un'interfaccia utente per la manutenzione.
- I report basati sulle eccezioni generate come eventi da un adapter non hanno commenti sulla loro esistenza. Nei casi in cui il meccanismo di correzione è stato utilizzato, è presente un commento. L'utilizzo del metodo di correzione è consigliato per mantenere l'integrità dei risultati generati dal sistema.
- Nessuna specifica con/senza eccezioni considera tali eccezioni.
- Quando viene utilizzato il meccanismo di correzione, si applica la specifica con/senza correzione.

Una volta implementata, è consigliabile che l'esperto di business logic applichi la logica a tutte le metriche del sistema.

È presente un altro metodo per applicare un'eccezione in una singola risorsa, se necessario. Questo metodo richiede l'utilizzo di risorse con stato In vigore.

L'impostazione di una risorsa allo stato Non in vigore indica che durante questo periodo il motore di calcolo ignorerà tutti gli eventi di dati non elaborati inviati per la risorsa. Impostando un periodo di tempo in cui la risorsa non è in vigore, vengono create nuove versioni della risorsa, una all'inizio del periodo di eccezione e un'altra alla fine del periodo di eccezione.

Tuttavia, se la risorsa è parte di una metrica di gruppo e la risorsa è in vigore e non in vigore nello stesso periodo di calcolo, nel risultato verrà preso in considerazione solo l'ultimo periodo in cui la risorsa era in vigore, come indicato sopra. In questo caso è consigliabile utilizzare la funzione degli attributi personalizzati. È possibile gestire un attributo aggiuntivo per la risorsa che indica lo stato della risorsa e la formula di business logic eseguirà una query per lo stato della risorsa in ogni punto pertinente nello script.

Considerazioni sulla memoria e sulle prestazioni

Di seguito viene riportato un insieme di situazioni da prendere in considerazione durante la progettazione delle soluzioni di business logic. Le situazioni descritte sono i casi in cui le prestazioni del motore di calcolo potrebbero subire un impatto negativo:

- **Parametri**

Se il valore di un parametro è obbligatorio nel codice, è consigliata la creazione di una variabile globale per assegnare il valore del parametro. Inoltre, quando il valore del parametro è obbligatorio, utilizzare invece la variabile globale. In questo modo si impedisce una situazione in cui il motore crea la mappa dei parametri per l'invocazione di ogni parametro.

- **Utilizzo delle mappe nelle metriche di gruppo**

Gli oggetti di mappa globale grande nella business logic per le metriche di gruppo devono essere utilizzati con la massima attenzione. Mentre il motore calcola una metrica di gruppo, è occupato per il caricamento di variabili globali da stati precedenti per ogni elemento nel cluster separatamente.

- **Registrazione di business logic**

È consigliato filtrare gli eventi di dati non elaborati unicamente con i metodi della registrazione. L'aggiunta del filtro interno utilizzando un'istruzione If all'interno del codice comporterà l'aumento del tempo di elaborazione. Ma soprattutto, è richiesto un overhead aggiuntivo dal motore per il recupero e l'elaborazione dei record di dati non elaborati che non sono necessari.

- **Evitare l'utilizzo di Dispatcher.RegisterByEventType**

Consente di migliorare le prestazioni. L'utilizzo di questo metodo di registrazione indica che è in corso la registrazione di tutte le risorse nel sistema e non solo nelle risorse che presentano eventi di quel tipo specifico. Pertanto, ogni modifica nella risorsa influisce sui calcoli della metrica. Un altro svantaggio nell'utilizzo di questo metodo di registrazione è in fase di esecuzione della metrica quando accede ai dati non elaborati. È quindi necessario filtrare ed escludere dai dati non elaborati solo gli eventi con il tipo di evento specifico e ignorare gli altri eventi.

- **Dispatcher.Register**

Quando si utilizza Dispatcher.Register, verificare *sempre* di aver specificato il ^{terzo} parametro. La registrazione senza il ^{terzo} parametro è esattamente come eseguire le registrazioni per tipo di evento (Dispatcher.RegisterByEventType). In altre parole, assicurarsi di aver utilizzato almeno un altro parametro oltre ai primi due.

- **Granularità di calcolo**

È importante attivare solo gli agenti necessari ai fini del calcolo e del drill-down. Il calcolo di tutte le unità di tempo dell'agente richiede notevoli risorse del processore.

Attivazione dei contratti (responsabile contratti)

L'attivazione di un contratto viene eseguita con la conferma dello stesso. Per ulteriori informazioni, consultare la Guida in linea.

L'attivazione del contratto avvia il motore per iniziare i calcoli per le metriche del contratto e produrre quindi i risultati per il contratto.

Prima di attivare il contratto, verificare se tutte le condizioni seguenti sono state soddisfatte:

- Tutte le metriche devono avere la business logic definita (nella metrica o come modulo collegato), che è stata verificata e non contiene errori.
- Tutte le metriche hanno una soglia di dashboard definita. Per ulteriori informazioni, consultare la Guida in linea. È importante che le soglie siano già definite in modo che durante il processo di calcolo il dashboard sarà in grado di valutare i limiti per la metrica.
- Le date di inizio validità siano conformi al periodo di tempo corretto e che siano disponibili record di dati non elaborati. Verificare inoltre che i risultati siano come previsto utilizzando l'ambito di business logic.

Una volta confermato il contratto, controllare se l'attivazione è stata iniziata correttamente e che i calcoli procedano come previsto.

Attenersi alla seguente procedura:

1. Verificare che tutti i componenti di servizio CA Business Service Insight siano stati avviati (specialmente il motore di calcolo, che comprende sia il PslWriter, sia il servizio di calcolo per le penali). È consigliabile che tutti i componenti del servizio siano in esecuzione quando è richiesto il calcolo.
2. Eseguire la diagnosi del contratto. La funzionalità di diagnostica del contratto consente di visualizzare i risultati di una serie di test eseguiti su tutte le metriche del contratto (e le formule di penale se utilizzate). Viene fornita la gravità del risultato del test con una procedura di risoluzione suggerita. È consigliabile eseguire una diagnosi a ogni attivazione di un contratto, nonché dopo che il calcolo del contratto è stato completato.
3. Generare il report in formato libero sullo stato del calcolo. Questo report è incluso come parte dell'installazione iniziale CA Business Service Insight e si trova nella cartella del pacchetto Admin Reports (Report di amministrazione). Fornisce informazioni sull'avanzamento del calcolo e può essere utilizzato in questo punto per verificare se il motore PSL è in fase di esecuzione e se il calcolo è stato completato. Consultare il report per valutare se esistono eventuali potenziali problemi nel calcolo.

Il report include i seguenti campi di colonna:

Campo	Descrizione
Contratto	Contiene il nome del contratto. L'elenco contiene i nomi dei contratti in vigore e non in vigore.
Metrica	Contiene il nome della metrica nel contratto. L'elenco contiene tutte le metriche comprese in ogni contratto.
Periodo di riferimento	Contiene il periodo di calcolo della metrica. L'elenco visualizza una voce per ogni unità di tempo del calcolo della metrica in base agli agenti attivi e alla definizione di granularità della metrica. Nei casi in cui il periodo di calcolo è il periodo di riferimento, questo verrà annotato.
Updated Up To	Contiene l'ultima data di aggiornamento del risultato. Indica che il risultato della metrica specifica è disponibile fino a questa data. Ad esempio, se è visualizzato 01/01/2006, indica che tutti i risultati per la metrica in tale unità di tempo sono aggiornati fino a questa data e che i report sono disponibili fino a questa data.
Last Cycle Begin At	Contiene l'ora in cui inizia il ciclo di calcolo durante il quale viene aggiornato il risultato della metrica.
Requires Recalculation From	Nei casi in cui il motore contrassegna una determinata metrica per il ricalcolo e non è stata ancora aggiornata, la data risultante verrà visualizzata in questo campo che specifica l'ora a partire dalla quale i risultati non sono pertinenti e quindi richiedono l'aggiornamento. Può verificarsi in tutti i casi di ricalcolo.
Last Update At	L'ora in cui il motore ha aggiornato il record con l'ultimo risultato.
Handled by Engine #	Contiene il numero del motore assegnato per la gestione del calcolo della metrica specifica.

Questo report può inoltre fornire le seguenti informazioni basate sui dati non elaborati disponibili:

- Il tempo impiegato dal motore per calcolare una singola metrica. È possibile visualizzare, ordinando tutte le metriche calcolate in un singolo ciclo per la relativa ora di aggiornamento, il tempo impiegato dal motore per aggiornare ciascuna metrica. Tutti i record con lo stesso valore di Last Cycle Begin At vengono calcolati durante lo stesso ciclo e l'ora di aggiornamento è il valore Last Update At. È possibile valutare il tempo impiegato dal motore per calcolare l'intera metrica con le relative unità di tempo dell'agente di riferimento, nonché ogni unità di tempo.
- Il tempo impiegato dal motore per calcolare un contratto intero. Questa operazione viene eseguita esaminando l'ora di aggiornamento della prima metrica del contratto e l'ora di ultimo aggiornamento dell'ultima metrica del contratto e calcolando l'intervallo di tempo.

Ricalcolo completo delle metriche del contratto

Il processo di ricalcolo nel contesto corrente è l'inizializzazione di un ricalcolo completo di sistema da parte dell'esperto delle sorgenti dati o dell'esperto di business logic. Non corrisponde al ricalcolo del motore eseguito durante un normale processo di calcolo. Questo tipo di operazione viene in genere eseguita durante o dopo il processo di configurazione del contratto quando potrebbero essere stati identificati diversi malfunzionamenti.

È consigliabile che il ricalcolo completo sia iniziato solo quando il sistema è in uno stato stabile (vale a dire, non durante la creazione del sistema) prima dell'implementazione nel sito.

Il ricalcolo viene attualmente effettuato con l'esecuzione di uno script SQL sul database. Questo script pulisce la tabella PSL e tutte le tabelle di supporto associate che fanno parte del processo di calcolo. Questo script deve essere approvato e convalidato dal team del Supporto tecnico di CA prima di essere eseguito, poiché modifiche strutturali occasionali apportate al sistema possono comportare modifiche allo schema del database e/o agli oggetti dipendenti.

Nota: prima di eseguire lo script, tutti i componenti del servizio devono essere interrotti e dopo l'esecuzione, i componenti del servizio possono essere riavviati per attivare nuovamente i calcoli.

Le seguenti situazioni potrebbero richiedere un ricalcolo completo:

- Problemi con le definizioni nel contratto: se in questa fase viene rilevato che la destinazione di una metrica è stata definita in modo scorretto o una soglia della metrica è errata.
- Errori nel contratto.
- Necessità di rivalutare le soglie del dashboard o di rigenerare le notifiche SLA.

Nota: contattare il Supporto tecnico di CA per discutere questo caso se necessario e per ottenere inoltre le copie degli script per la versione installata di CA Business Service Insight.

Creazione dei risultati finali (responsabile contratti)

In questa fase, ogni output del sistema viene configurato per soddisfare i requisiti, incluse la creazione e la formattazione di tutti i report.

La configurazione dei risultati finali include le seguenti operazioni:

- Definizione delle impostazioni di protezione (autorizzazioni utente e gruppi).
- Creazione di report salvati.
- Creazione di booklet.
- Creazione dei modelli di esportazione del contratto.
- Creazione delle visualizzazioni Service Delivery Navigator (SDN)
- Configurazione delle mappe del dashboard
- Creazione dei profili di notifica del livello di servizio.

Definizione delle impostazioni di protezione (amministratore)

Durante la definizione delle impostazioni di protezione, è necessario configurare le autorizzazioni degli utenti di CA e le autorizzazioni associate. Le impostazioni includono la definizione di quali informazioni devono essere accessibili a un utente (quali entità all'interno del sistema l'utente può visualizzare o modificare). Le autorizzazioni possono essere impostate in diversi livelli da gruppi di utenti, ruoli o singolarmente per utente. Le informazioni accessibili sono definite in relazione ai contraenti e possono essere impostate direttamente sull'utente o ereditate dal gruppo di utenti cui appartiene l'utente.

In questa fase vengono configurati i ruoli principali e i gruppi di utenti associati a essi, in modo che quando viene aggiunto un nuovo utente, è necessario essere associato solo a un gruppo per ereditare le relative impostazioni.

Le azioni consentite sono configurate nei ruoli e vengono inviate all'utente direttamente con l'associazione dell'utente a un ruolo o dal gruppo di utenti cui appartiene. Le azioni consentite relative al dashboard vengono definite anche nel ruolo associato.

È consigliabile che l'amministratore determini quali gruppi di utenti e ruoli devono essere definiti e le rispettive autorizzazioni necessarie per configurare una struttura che consenta l'aggiunta facile di utenti.

Creazione di report

Per creare un report, attenersi alla procedura riportata di seguito:

1. Creare tutte le cartelle richieste di report: è necessario creare tutte le cartelle in anticipo in modo che siano disponibili durante il salvataggio di ogni nuovo report salvato. In genere ogni contratto dispone di una propria cartella che include una cartella esecutiva per i report di livello superiore.
2. Definire i criteri di filtro del report tramite la procedura guidata di creazione report: ogni report salvato viene avviato con la generazione del report tramite la procedura guidata di creazione report. Nella Procedura guidata di creazione report vengono selezionati i criteri di filtro richiesti e viene generato un report. Gli amministratori IT possono impostare i parametri del report per designare i campi definiti dall'utente che devono essere compilati dall'utente/visualizzatore del report, consentendo la generazione di risultati di report specifici per gli interessi di tale utente.

Durante la generazione di un report per l'impostazione come report salvato, è molto importante definire i criteri di filtro in modo più flessibile possibile. In questo modo si evitano azioni non necessarie mentre il sistema si espande e si evolve. L'obiettivo deve essere di assicurare che ogni report visualizzi sempre le informazioni correnti e aggiornate. Ad esempio, se un report attualmente visualizza tre componenti del servizio, in futuro, all'aggiunta di nuovi componenti del servizio, è importante che questo servizio sia aggiunto automaticamente al report e non richieda la configurazione di un nuovo report. Oppure, in caso di report mensile, il report attualmente visualizza tre mesi, quindi nel prossimo mese deve visualizzare quattro mesi, incluso l'ultimo mese. In molti casi, i report devono sempre visualizzare i dati degli ultimi sei mesi. Questi tipi di report a finestra scorrevole sono estremamente utili rispetto ai report con durata fissa perché non richiedono un'ulteriore attenzione una volta creati.

Di seguito sono riportati alcuni suggerimenti per impostare i criteri di filtro per i report salvati:

Scheda Criteri

- Utilizzando l'opzione Solo dati di business i dati visualizzati sono limitati solo ai dati associati al periodo di riferimento della metrica.
 - Esprimere sempre la preferenza per l'*opzione di raggruppamento o l'opzione di esecuzione report* invece di selezionare la metrica specifica su cui eseguire il report.
 - Impostare un intervallo di tempo dinamico. Se viene impostato come un intervallo di tempo fisso, il report mostra sempre gli stessi risultati. Ad esempio, Last 3 Months (Ultimi tre mesi) è un intervallo dinamico.

Scheda Varie

- Verificare se nel report devono essere presentati periodi incompleti oppure no. In tal caso, selezionare l'opzione Periodi incompleti dall'elenco. In genere, durante la configurazione dei report, il periodo incompleto viene escluso perché non è un risultato finale e, di conseguenza, non un risultato di business.
- Progettare il layout del report. Una volta generato il report, è possibile progettare il relativo layout tramite lo strumento di progettazione nella pagina del report. La progettazione potrebbe avere requisiti specifici in base al destinatario previsto del report. È consigliabile creare più layout di progettazione, uno per ogni possibile scenario, da applicare al resto dei report da generare. Quindi, quando si definiscono i criteri del report, nella scheda *Visualizza*, scegliere *Usa progettazione da*.

Nota: per suggerimenti su come utilizzare lo strumento di modifica per progettare il layout dei report, consultare la sezione successiva.

Scheda Generale

- Salvare il report. Una volta generato e progettato il report, può quindi essere salvato nella relativa cartella.
- Durante il processo di salvataggio del report, viene associato agli utenti cui è consentito l'accesso. È pertanto importante disporre di un gruppo di utenti già definito in modo che sia possibile associare i report agli utenti. Gli utenti con le autorizzazioni appropriate possono essere associati al report in una seconda fase utilizzando le opzioni della cartella.
- Associare il report correlati in modo da semplificare la navigazione tra report simili o la creazione di una relazione tra i report per utenti di tali report.
- Nella pagina Cartella dei report, è possibile creare report, report di raggruppamento, report composti, report in formato libero, booklet, collegamenti e cartelle, nonché eseguire ricerche dei report.

Dal menu Report, fare clic su Cartella dei report. Viene visualizzata la pagina Cartella dei report, in cui è riportato un elenco dei report.

Report sulla deviazione

Il valore Deviazione è calcolato automaticamente dal motore CA Business Service Insight per le metriche che dispongono di una destinazione. Il valore rappresenta la differenza tra il livello di servizio effettivo e la destinazione. La formula di calcolo della deviazione, calcolata automaticamente da CA, varia in base alla definizione del livello di servizio: se la destinazione del livello di servizio è un valore massimo (quando il livello di servizio effettivo è "non superiore a"), o un valore minimo (quando il livello di servizio è "non inferiore a"). Di seguito è riportato un esempio di variazione della formula:

Dichiarazione di destinazione	Soglia del livello di servizio	Formula di deviazione
Il servizio deve essere disponibile almeno al 99% dell'orario pianificato.	La destinazione è la soglia minima	$Service_{Dev} = (-1) \times \frac{Service_{SL} - Service_{SLT}}{Service_{SLT}}$
Il MTTR medio non deve superare le 4 ore per mese.	La destinazione è la soglia massima	$Service_{Dev} = \frac{Service_{SL} - Service_{SLT}}{Service_{SLT}}$

In cui $Service_{Dev}$ = deviazione del servizio

In cui $Service_{SLT}$ = prestazioni previste del servizio

In cui $Service_{SL}$ = prestazioni effettive del servizio

Il seguente esempio illustra un calcolo della deviazione minima:

Il servizio deve essere disponibile almeno al 99% del periodo di applicazione pianificato.
Il livello di servizio effettivo è al 98% durante il periodo di applicazione pianificato.

$$Service_{Dev} = (-1) \times \frac{98\% - 99\%}{99\%} = 1.0101\%$$

I report sulla deviazione vengono utilizzati per le visualizzazioni di livello superiore delle garanzie di natura esterna (altro tipo di calcolo) e per l'aggregazione in una singola barra con basi comuni.

Se, ad esempio, nel contratto è presente la matrice seguente:

Servizio - Assistenza tecnica	Dominio del servizio - Gestione ticket Priority 1	Dominio del servizio - Gestione ticket Priority 2	Dominio del servizio - Gestione ticket Priority 3
	Tempo medio di risoluzione P1	Tempo medio di risoluzione P2	Tempo medio di risoluzione P3
	% di ticket P1 risolti entro T1	% di ticket P2 risolti entro T1	% di ticket P3 risolti entro T1
	% di ticket P1 con risposta entro T1	% di ticket P2 con risposta entro T1	% di ticket P3 con risposta entro T1
	Tempo medio di risposta P1	Tempo medio di risposta P2	Tempo medio di risposta P3

I risultati di generazione di un report del dominio del servizio filtrato per servizio di assistenza tecnica sono illustrati nel grafico seguente.

Il report sopra riportato consente al manager del livello di servizio di vedere la qualità delle prestazioni dell'assistenza tecnica in base alle diverse priorità, indipendentemente dal contratto o dal tipo di obbligo.

Tutte le metriche di assistenza tecnica sono aggregate in una singola barra in base alla priorità.

Ad esempio, la barra Priority 1 (Priorità 1) mostra le tre metriche definite all'interno della metrica e aggrega la rispettiva deviazione in un unico valore. Il metodo di aggregazione può essere selezionato nella procedura guidata di creazione report tra media/minimo/massimo.

Con questo report il manager può dedurre, ad esempio, la necessità di investire più risorse nei ticket con priorità 1 o modificare i contratti a essi associati.

Questo esempio mostra che la modellazione fornisce sia il report su un singolo obbligo che indica se un contratto è stato rispettato o violato, ma anche un report con gestione più ampia che consente al manager del livello di servizio di gestire le risorse in modo più efficiente e quindi di migliorare i propri componenti del servizio.

Report in formato libero

I report in formato libero consentono agli utenti di generare report basati su query SQL del database CA Business Service Insight o da qualsiasi altra origine dati esterna cui è possibile accedere tramite una connessione dal server CA Business Service Insight. Sono inoltre inclusi altri tipi di origine dati cui è possibile accedere tramite ODBC, ad esempio Excel, Access, Lotus Notes, file di testo e così via. I report in formato libero vengono comunemente utilizzati per configurare report statistici basati su dati creati con comandi di business logic Tools.SaveRecord e Tools.SaveFields.

I report in formato libero si connettono tramite una stringa di connessione a un database selezionato ed eseguono una query SQL sul database utilizzando una stringa di query. I parametri possono essere aggiunti a entrambe le stringhe per creare report dinamici che consentono all'utente di immettere o selezionare valori specifici da includere nella query, quali il nome utente e la password per la connessione al database.

I report in formato libero vengono visualizzati nelle schede Diagramma, Dati e Filtro analogamente ai report generati tramite la procedura guidata di creazione report.

Nota: i report in formato libero possono includere diagrammi solo se tutte le colonne, esclusa la prima colonna, hanno un valore numerico. I dati nella prima colonna vengono utilizzati per i titoli dell'asse X. I nomi di colonna sono utilizzati per altri titoli.

A causa del fatto che i report in formato libero utilizzano l'accesso diretto a un database e una query SQL aperta, la manutenzione è difficile. È necessario prestare attenzione per non compromettere i dati di base che vengono utilizzati come origine per i report in formato libero. Quando i report vengono generati da un'origine dati esterna, è consigliabile impostare un processo di notifica per garantire che tali origini dati non siano modificate senza consultare prima il responsabile contratti per i report sui dati in formato libero.

Informazioni generali da considerare durante la creazione di report in formato libero.

- A causa di problemi di formattazione della data internazionale, spesso è utile specificare il formato esatto di data con la creazione di una formula personalizzata nella business logic. Questa formula consente di convertire la data in una stringa con lo stesso formato ogni volta e di evitare problemi con la formattazione data americana rispetto a quella europea. Deve essere utilizzata per le date che verranno scritte nella tabella T_SLALOM_OUTPUTS tramite i comandi Tools.SaveFields o Tools.SaveRecord. Di seguito è riportato un esempio della formula:

Funzione FormatDate(DateField)

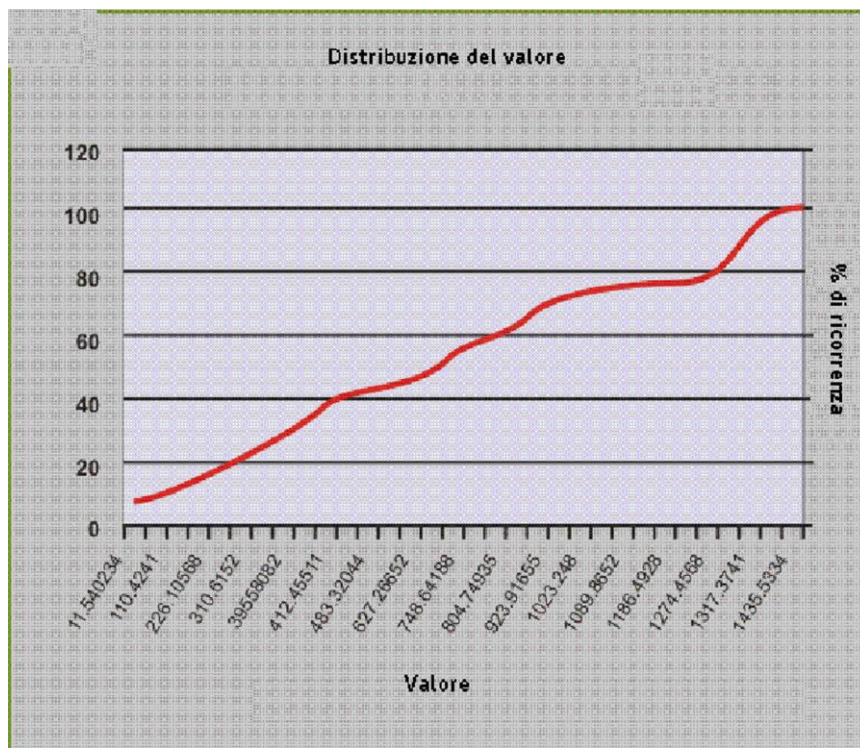
```
If DateField = "" Then
    FormatDate = ""
Else
    Dim PeriodYear, PeriodMonth, PeriodDay, PeriodHour,
    PeriodMinute, PeriodSecond
    PeriodYear    = DatePart("yyyy",DateField)
    PeriodMonth   = DatePart("m",DateField)
```

```
PeriodDay      = DatePart("d",DateField)
PeriodHour     = DatePart("h",DateField)
PeriodMinute   = DatePart("n",DateField)
Periodsecond   = DatePart("s",DateField)
FormatDate     = PeriodDay&"/"&PeriodMonth&"/"&PeriodYear& _
" "&PeriodHour&":"&PeriodMinute&":"&Periodsecond
End If
End Function
```

- Quando si utilizza il parametro di tempo dai gestori eventi di business logic, o qualsiasi altra data/ora dall'oggetto eventDetails, considerare sempre l'impatto delle ore di differenza di fuso orario. Tenere presente che le date, quando vengono scritte nella tabella, potrebbero essere nell'ora UTC invece che nell'ora locale. Potrebbe essere necessario utilizzare la funzione Tools.GetLocaleTime() per correggere il problema.
- È possibile usare l'utility di Report Designer (quando un report in formato libero produce un grafico) per personalizzare l'aspetto.
- Le opzioni di esportazione PDF per il report in formato libero sono personalizzabili utilizzando la sezione Report Parameters (Parametri report) della schermata di configurazione in formato libero, per elementi quali il layout di pagina (orientamento orizzontale, verticale).
- L'HTML può essere incorporato nei report per eseguire diverse funzionalità, ad esempio l'aggiunta di collegamenti ipertestuali o la modifica dei colori di riga e di colonna, i tipi di carattere o le impostazioni di visualizzazione.
- È possibile creare visualizzazioni di database (funzionalità Oracle) sulla tabella T_SLALOM_OUTPUTS per semplificare l'SQL richiesto per i report.
- Quando si specificano i parametri per i report, è possibile che siano impostati in vari modi: testo libero, numerico (convalida), password (caratteri nascosti, utili per le password delle connessioni di database), data (utilizzando l'utility di selezione data) ed elenchi (creati specificando un'istruzione SQL per popolare l'elenco).
- I valori di parametro specificati nella sezione SQL del report in formato libero devono essere sostituiti dal nome del parametro, preceduto dal simbolo @. Ad esempio, @PARAM1. L'utilizzo delle virgolette che racchiudono il valore di questo parametro è obbligatorio se i valori sono stringhe.

Report in formato libero con istogramma generico

La seguente query può essere utilizzata in un report in formato libero per presentare la distribuzione dei valori in una tabella in percentuale, come visualizzato nel diagramma seguente:



Nel diagramma precedente è possibile visualizzare quale porzione (in percentuale) dei valori è inferiore a 11,5 (0%), inferiore a 804,74 (~ 50%) e inferiore a 1435,53 (100%).

Se l'accordo SLA specifica le destinazioni, ad esempio "x% dei valori deve essere inferiore a y", i risultati di questo formato libero facilitano la ricerca dei valori x, y che assicurano la conformità all'accordo SLA.

I parametri seguenti sono utilizzati nella query:

- **@Query:** istruzione SELECT come segue: `select some_value val from some_table`. L'alias val è obbligatorio. La query fornisce i valori per cui viene analizzata la distribuzione.
- **@Buckets:** il numero di valori sull'asse X. I valori origine sono arrotondati a tali numeri. Ad esempio, se si specifica `@Buckets = 100`, quindi i valori dei dati di origine vengono divisi in 100 gruppi e la query consente di calcolare il numero di valori compreso all'interno di ogni gruppo. Più elevato è `@Buckets`, più accurato è il risultato, ma il diagramma non è graficamente interessante.
- **@Relation:** la direzione di accumulo. Valori possibili; `<=` (quando la destinazione è non inferiore a) o `>=` (in caso contrario).

La query può essere eseguita con l'origine dati o con `T_SLALOM_OUTPUTS` per ottenere risultati ottimali.

La seguente query restituisce il grafico come indicato:

```
select val,100*records/(select count(*) from (@Query))
from
(
  select x.bucket_val val,
    sum(y.records) records
  from
  (
    select round(val/bucket_size,0)*bucket_size bucket_val,
      count(*) records
    from
    (
      select (max(val)-min(val))/@Buckets bucket_size
      from
      (
        @Query
      )
    ) params,
    (
      @Query
    ) source
  group by round(val/bucket_size,0)*bucket_size
  order by round(val/bucket_size,0)*bucket_size
) x,
(
  select round(val/bucket_size,0)*bucket_size bucket_val,
```

```

        count(*) records
    from
    (
        select (max(val)-min(val))/@Buckets bucket_size
        from
        (
            @Query
        )
    ) params,
    (
        @Query
    ) source
    group by round(val/bucket_size,0)*bucket_size
    order by round(val/bucket_size,0)*bucket_size
) y
where y.bucket_val @Relation x.bucket_val
group by x.bucket_val
order by x.bucket_val
)

```

Di seguito è riportato un elenco di parametri di esempio (in formato XML) che può essere utilizzato:

```

<custom>
  <connection>
    <params/>
  </connection>
  <query>
    <params>
      <param name="@Query" disp_name="Data Type" type="LIST">
        <value>PDP Context Activation Success</value>
        <list>
          <item>
            <value>select success_rate as val from
PDP_Context_Activation_Success.CSV</value>
            <text>PDP Context Activation Success</text>
          </item>
          <item>
            <value>select throughput as val from [gprs
throughput volume by apn.csv]</value>
            <text>Throughput of a Single APN</text>
          </item>
          <item>
            <value>select throughput as val from [Generic GPRS
Throughput.CSV]</value>
            <text>Generic Throughput</text>
          </item>
        </list>
      </param>
      <param name="@Buckets" disp_name="X Axis Values" type="LIST">

```

```

        <value>100</value>
        <list>
            <item>
                <value>25</value>
                <text>25</text>
            </item>
            <item>
                <value>50</value>
                <text>50</text>
            </item>
            <item>
                <value>100</value>
                <text>100</text>
            </item>
            <item>
                <value>250</value>
                <text>250</text>
            </item>
            <item>
                <value>500</value>
                <text>500</text>
            </item>
            <item>
                <value>1000</value>
                <text>1000</text>
            </item>
        </list>
    </param>
    <param name="@Relation" disp_name="Violation of threshold means"
type="LIST">
        <value>providing too little</value>
        <list>
            <item>
                <value>>=</value>
                <text>providing too little</text>
            </item>
            <item>
                <value><=</value>
                <text>providing too much</text>
            </item>
        </list>
    </param>
</params>
</query>
</custom>

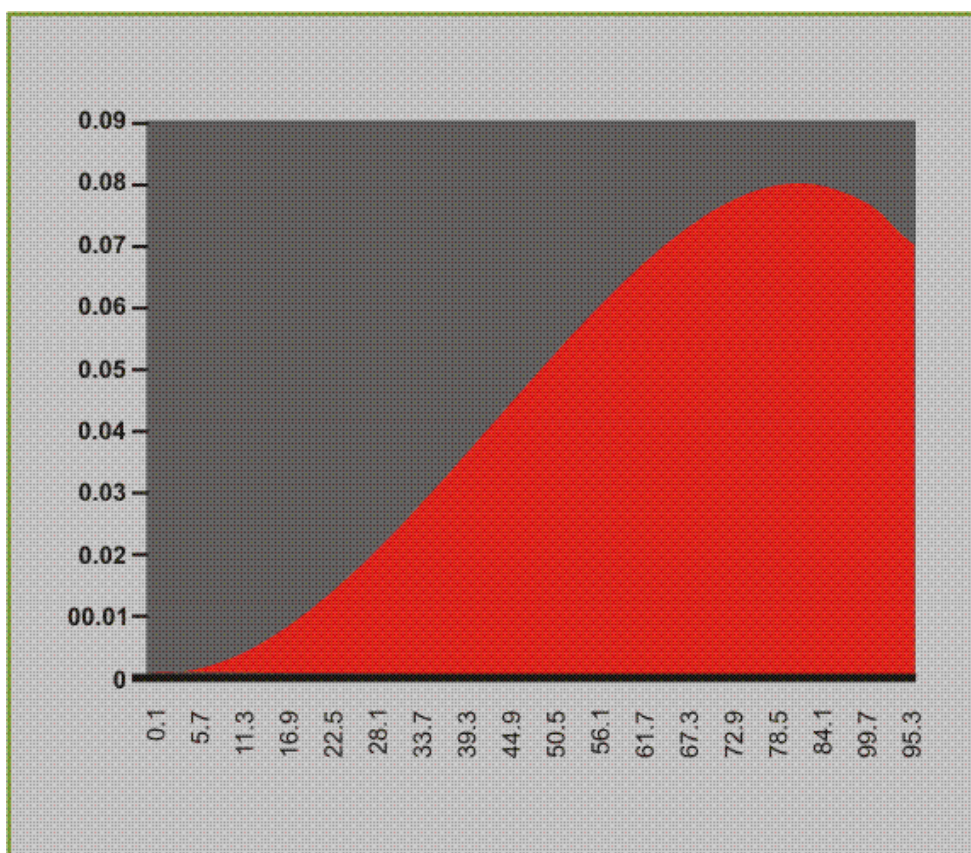
```

Commenti

- La query è stata ottimizzata per Oracle e SQL Server. Per altre origini dati ODBC, potrebbe essere necessario aggiungere "as" (come) prima della colonna degli alias e applicare altre modifiche.
- Durante l'esportazione dei risultati in Excel, è consigliabile che l'utente generi un report dei dati XY (grafico a dispersione) per rappresentarlo. Quando il grafico è puntato, sarà inoltre possibile visualizzare la spaziatura verticale dei valori.

Report in formato libero con distribuzione normale (gaussiana) generica

La query riportata di seguito potrebbe essere utilizzata in un report in formato libero per rappresentare la distribuzione normale di valori in una tabella, come illustrato nel diagramma seguente:



I parametri seguenti sono utilizzati nella query:

- @Query: istruzione SELECT come segue:

```
select min (something) min_val,  
       max (something) max_val,  
       avg (something) avg_val,  
       stddev (something) stddev_val  
from table_name
```

x_val è obbligatorio. La query fornisce i valori per cui viene analizzata la distribuzione.

- @Buckets: il numero di valori sull'asse X. I valori origine sono arrotondati a tali numeri. Ad esempio, se si specifica @Buckets = 100, quindi i valori dei dati di origine vengono divisi in 100 gruppi e la query consente di visualizzare il valore di distribuzione normale per ogni gruppo. Più elevato è @Buckets, più accurato è il risultato, ma il calcolo è più pesante. 100 è una selezione consigliata per @Buckets.

Nuovamente, la query può essere eseguita con i dati di origine o con T_SLALOM_OUTPUTS per ottenere risultati ottimali.

La seguente query restituisce il grafico come indicato:

```
select (max_val-min_val)/@Buckets*serial,  
       (1/stddev_val*sqrt (2*3,14159265359))*exp ( -1/(2*power (stddev_val,  
2))*power (((max_val-min_val)/@Buckets*serial-avg_val), 2))  
from    (  
        @Query  
    )  
    (  
        select      rownum serial  
        from        t_psl  
        where rownum < @Buckets  
    )  
order by serial
```

I parametri XML corrispondenti:

```
<custom>  
  <connection>  
    <params/>  
  </connection>  
  <query>  
    <params>  
      <param name="@Query" disp_name="Data Type" type="LIST">  
        <value>PDP Context Activation Success</value>  
        <list>  
          <item>  
  
          </item>  
        </list>  
      </param>  
    </params>  
    <value>  
select min(success_rate) min_val,max(success_rate) max_val,avg(success_rate)  
avg_val,stddev(success_rate) stddev_val  
      from PDP_Context_Activation_Success.CSV  
    </value>
```



```

        <text>PDP Context Activation Success</text>
      </item>
      <item>
        <value>
select min(throughput) min_val,max(throughput) max_val,avg(throughput)
avg_val,stddev(throughput) stddev_val
  from [gprs throughput volume by apn.csv]
</value>
        <text>Throughput in Kb</text>
      </item>
    </list>
  </param>
  <param name="@Buckets" disp_name="X Axis Values" type="LIST">
    <value>100</value>
    <list>
      <item>
        <value>25</value>
        <text>25</text>
      </item>
      <item>
        <value>50</value>
        <text>50</text>
      </item>
      <item>
        <value>100</value>
        <text>100</text>
      </item>
      <item>
        <value>250</value>
        <text>250</text>
      </item>
      <item>
        <value>500</value>
        <text>500</text>
      </item>
      <item>
        <value>1000</value>
        <text>1000</text>
      </item>
    </list>
  </param>
</params>
</query>
</custom>

```

Commenti

- La query è stata ottimizzata per Oracle e SQL Server. Per altre origini dati ODBC, potrebbe essere necessario aggiungere "as" (come) prima della colonna degli alias e altre modifiche.
- Durante l'esportazione dei risultati in Excel, è consigliabile che l'utente generi un report dei dati XY (grafico a dispersione) o un report di area per rappresentarlo.

Configurazione delle pagine del dashboard

La sezione seguente include i suggerimenti sui contenuti da configurare per gli utenti CA Business Service Insight. I suggerimenti sono di livello superiore e devono tener conto dei requisiti specifici del cliente. Le pagine vengono illustrate di seguito nel contesto di un utente specifico descritto tramite il proprio ruolo nell'organizzazione.

Pagine del responsabile esecutivo

Si presume che un responsabile esecutivo possa essere interessato alla visualizzazione di livello superiore tra reparti, paesi, account. Un responsabile esecutivo in genere non è un ruolo operativo e richiede visualizzazioni in grado di fornire informazioni per prendere decisioni in termini di strategia. Pertanto, lo stato contrattuale invece dello stato corrente potrebbe essere più attinente per la visualizzazione nelle mappe del responsabile esecutivo e nel dashview di aggregazione.

Ad esempio, i seguenti dashview possono essere inclusi nella pagina Panoramica:

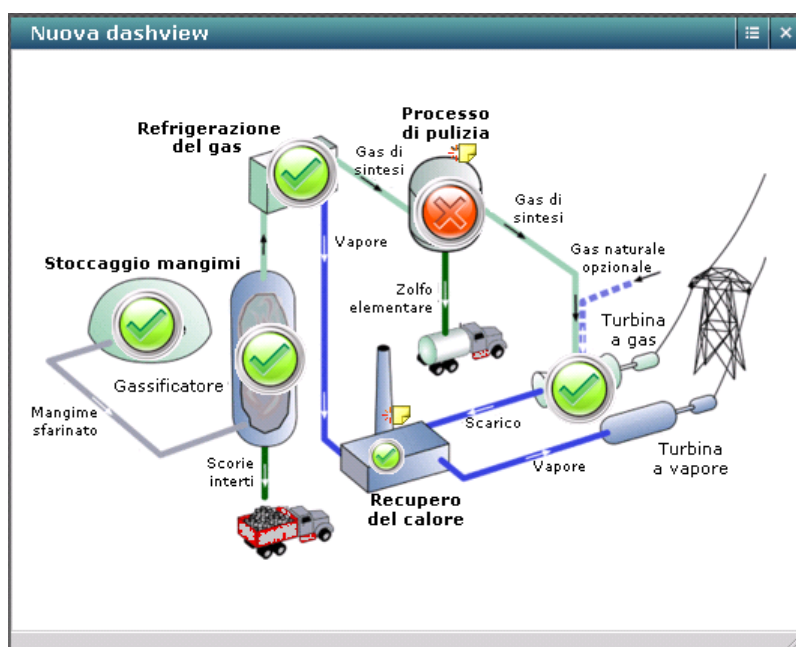
Dashview	Descrizione
Critical accounts (Account critici)	Include tutti i contraenti contrassegnati come sensibili. Il responsabile esecutivo seleziona i contratti o i contraenti che dal suo punto di vista sono considerati sensibili.
Overall performance (Prestazioni complessive)	Comprende widget personalizzati di qualità complessiva, widget di visualizzazione di aggregazione che include tutti i KQI degli account.
Dipartimenti	Utilizza come immagine di sfondo l'organigramma e inserisce i widget nei relativi reparti. In genere il gruppo di componenti del servizio è utile in questo caso a seconda di quale reparto rappresenti nell'organizzazione.
Geografie	Utilizza come immagine di sfondo una carta geografica e individua i widget dei gruppi di contraenti nelle rispettive località.
Financial performance (Prestazioni finanziarie)	Comprende widget che includono informazioni di aggregazione sulle metriche finanziarie
URL	Include le pagine dal portale aziendale, ad esempio i lead nella forza vendita

Report consigliati per la pagina:

Report	Descrizione
Report sulla deviazione	Il report sui dieci contratti peggiori per un determinato periodo fornisce informazioni sulle aree con le peggiori prestazioni in termini di delivery del servizio.
Report finanziario per il mese corrente	Riassume lo stato finanziario aggregato nel tempo

Pagina Process (Processo):

Questa pagina deve contenere un dashview che presenta un diagramma di processo con widget che rappresentano ogni catena nel processo, come nell'esempio seguente:



Pagine del responsabile contratti

Le pagine configurate per il responsabile contratti visualizzano la qualità di fornitura del servizio per ciascun account gestito. È un dashview dei contratti pertinenti per aggiornare il responsabile sul livello di servizio attuale fornito secondo gli obblighi contrattuali sotto la sua responsabilità. Inoltre, tale pagina visualizza i report disponibili per ogni componente del servizio incluso nel contratto.

La pagina Panoramica include i seguenti dashview:

Dashview	Descrizione
My accounts (Account personali)	Widget di tutti i contratti o contraenti che sono sotto la responsabilità del responsabile contratti.
My Services (Servizi personali)	Componenti del servizio negli account gestiti dal responsabile contratti.
Financial performance (Prestazioni finanziarie)	Widget che includono le informazioni di aggregazione sulle metriche finanziarie per gli account gestiti.
URL	Portale, ad esempio il sistema CRM

Creare una pagina di account per ogni account gestito che fornisce una visualizzazione degli obblighi specifici dell'account. È consigliabile raggruppare le visualizzazioni di calcolo dello stato corrente distinte dallo stato contrattuale per consentire al responsabile account di essere proattivi e agire velocemente. Ad esempio:

Pagina di account	Descrizione
Account obligations (Obblighi account)	Comprende le metriche inclusa nel contratto
Financial performance (Prestazioni finanziarie)	Widget che includono le informazioni di aggregazione sulle metriche finanziarie per gli account gestiti.

Pagine di gestione dei servizi

Pagine da configurare per il responsabile di un servizio o di un dominio specifico, in cui è riportata una vista dettagliata degli obiettivi di servizio tra i vari contratti e gli stati dei relativi obiettivi. Inoltre, questa pagina contiene i report che evidenziano i parametri del livello di servizio critico misurati.

I dashview inclusi nelle pagine di gestione dei servizi sono simili alle pagine di gestione degli account, in cui le informazioni vengono solo visualizzate e aggregate a livello di servizio invece che a livello di contratto o di contraente.

Pagine iniziali

Una pagina del dashboard può essere assegnata come pagina iniziale, per consentire a un gateway personalizzabile l'interazione con il sistema, fornendo l'accesso rapido a informazioni e azioni.

Aggiunta dei profili di notifica del livello di servizio

Il profilo di notifica consente la definizione delle condizioni in cui viene inviata una notifica a un destinatario predefinito utilizzando un dispositivo specificato.

Prima di creare profili, è necessario considerare quali condizioni sono abbastanza importanti per richiedere la notifica e chi dovrebbero essere i destinatari previsti. È importante definire i profili che possono facilitare al responsabile contratti e all'amministratore di sistema la gestione in modo efficace di eventuali questioni e problemi di servizio rilevati. Tutte le notifiche devono indirizzare le risorse necessarie in modo specifico per risolvere i problemi più urgenti ed evitare una situazione di violazione.

Al momento dell'aggiunta delle notifiche del livello di servizio sono presenti alcuni campi che possono essere utilizzati per determinare lo stato della metrica e per decidere se una condizione di notifica deve essere attivata o no. È possibile utilizzare qualsiasi dettaglio della metrica standard per i criteri di filtro e condizione (ad esempio, contratto, servizio, nomi di metrica, destinazioni, periodi di applicazione, risultati del livello di servizio e così via). È possibile notificare anche le stime del livello di servizio in seguito; in tal caso, vengono forniti alcuni valori aggiuntivi, come la deviazione dal caso migliore/peggiore, e possono essere stimati anche i livelli di servizio. Sono estremamente utili per decidere se la violazione di un livello di servizio può essere ripristinata oppure no in un determinato periodo.

Le notifiche basate sul log di sistema e le sezioni generali di CA Business Service Insight sono altre funzionalità estremamente utili e consentono di inviare una notifica in base a qualsiasi azione che si verifica nel sistema (che viene scritta nel log di sistema). Poiché quasi tutte le azioni vengono registrate, fornisce un ambito molto ampio per i profili di notifica. I profili comuni di notifica del log di sistema comprendono:

- L'avviso di un adapter che esegue la notifica per l'esperto delle sorgenti dati.
- Qualsiasi condizione di errore immessa nel log, con notifica tramite posta elettronica inviata all'amministratore di sistema.
- Gli errori personalizzati creati mediante la business logic relativa a determinate condizioni dalle origini dati possono generare una notifica tramite posta elettronica al gruppo responsabile del servizio all'interno dell'organizzazione e al responsabile contratti. (Ad esempio, un incidente con priorità alta non è stato risolto entro l'intervallo di tempo > invio di una notifica all'assistenza tecnica per l'inoltro del problema)
- I tentativi ripetuti di accesso non valido possono essere inviati tramite posta elettronica all'amministratore di sistema per determinare se è un utente tenta di entrare illecitamente nel sistema
- Nuove voci di conversione provenienti dall'adapter (ad esempio, sono state trovate in un'origine dati nuove voci di risorsa, che potrebbero necessitare della conversione nel sistema per consentire all'accordo SLA di essere calcolato correttamente).

Di seguito è riportata la finestra Dettagli profilo di notifica, che visualizza la configurazione di un profilo di notifica personalizzato che fa riferimento a un tipo di evento personalizzato degli eventi aperti con ticket dell'assistenza tecnica. Se il criterio corrisponde, viene inviata una notifica critica al coordinatore dell'assistenza tecnica per assicurarsi che il ticket venga gestito in modo efficiente. Può essere utile nelle situazioni in cui un'applicazione è sotto osservazione alcune volte e richiede un livello di cura molto particolare.

Dettagli profilo di notifica

Nome: *	Critica	
Descrizione:		
Tipo: *	Messaggi di log	
Formula di condizione:	#Informazioni #Gravità =	Editor formula
Oggetto messaggio: *		
Messaggio:	#Codice #File #Informazioni	Editor messaggio
Destinatari: *	Incluso Disponibile i18n-testAlertRecipientName	
	Visualizza dettagli	Aggiungi destinatario della notifica
Gravità: *	Critica	
Stato: *	Abilitato	
Intervallo minimo tra notifiche: *	60	minuti

Di seguito è riportato un esempio di implementazione di una notifica sulla violazione del livello di avviso a seconda che l'accordo SLA possa ancora essere soddisfatto, considerando l'attuale livello di servizio e il tempo rimanente nel periodo di riferimento di business della metrica.

Esempio: notifica di violazione SLA reversibile

Questa notifica viene attivata quando un obiettivo (metrica) è attualmente violato, ma siccome lo scenario del caso migliore indica la conformità con l'obiettivo, la violazione può essere evitata se viene fornito un livello di servizio più appropriato da questo momento in poi.

Tipo: stima del livello di servizio

Formula condizionale:

```
#Deviazione>0 And #DeviazioneCasoMigliore<=0 And  
#LivelloServizioCasoMigliore<>#LivelloServizioCasoPeggior e And #ElementoGruppo = ""
```

Messaggio:

Nota: l'obiettivo #Metrica del contratto #Contratto è attualmente violato, ma la violazione è reversibile.

Mentre la destinazione del livello di servizio è #Destinazione, sono previsti i seguenti livelli di servizio:

- Livello di servizio da adesso: #LivelloServizio (Deviazione: #Deviazione%).
- Livello di servizio scenario peggiore: #LivelloServizioCasoPeggior e (Deviazione: #DeviazioneCasoPeggior e%).
- Livello di servizio scenario migliore: #LivelloServizioCasoMigliore (Deviazione: #DeviazioneCasoMigliore%).

Capitolo 4: Installazione e deployment

Questa sezione contiene i seguenti argomenti:

[Introduzione](#) (a pagina 202)

[Preparazione](#) (a pagina 204)

[Installazione](#) (a pagina 206)

Introduzione

Dopo aver completato la fase di configurazione e aver verificato correttamente i calcoli del sistema, il sistema è pronto per la distribuzione nell'ambiente di produzione. Questo capitolo è finalizzato a trattare tutti i passaggi che potrebbero essere necessari durante questa fase. Questi passaggi potrebbero variare a seconda delle installazioni; tuttavia, gli elementi seguenti devono essere controllati per verificare se sono stati effettuati tutti i preparativi per l'integrazione del sistema in un ambiente di produzione reale.

- I prodotti hardware e software di produzione sono disponibili e pronti per l'installazione.
- Il server di posta elettronica è configurato per supportare l'invio di posta interna/esterna.
- Agente FTP presente (se richiesto per i trasferimenti di file dei dati).
- I feed dell'origine dati sono disponibili e funzionanti

Sono presenti due fasi nell'installazione. Durante la prima fase, la fase di installazione, l'amministratore di sistema deve installare e integrare i componenti CA Business Service Insight nell'ambiente di produzione. Durante la seconda fase, viene verificata la funzionalità del sistema e il sistema deve essere posto in uno stato di monitoraggio per verificare che i processi di sistema e le funzionalità dell'interfaccia utente funzionino tutti correttamente.

Durante il processo di installazione potrebbe essere necessario lavorare sui server di produzione tramite strumenti di connessione remota che devono essere installati nel server Web e nel server applicazioni per consentire un'installazione completa e sicura.

L'installazione del database Oracle deve essere eseguita da un esperto amministratore di database (DBA) Oracle se possibile per assicurare che la configurazione sia eseguita correttamente e sia conforme ai requisiti aziendali in uso. In alternativa, la creazione del database può essere affidata all'amministratore di sistema, tramite l'utility di creazione dei dati fornita con dell'installazione CA Business Service Insight. In questo caso, l'installazione deve essere verificata dal DBA per assicurare che sia stata completata correttamente. Se il sistema è stato già configurato su un sistema di sviluppo e deve essere eseguita la migrazione, quindi è necessario spostare il contenuto del database nel nuovo database di produzione.

A volte, quando l'intero sistema è stato trasferito nel nuovo ambiente di produzione, è consigliabile eliminare tutti i dati calcolati e non elaborati archiviati nel sistema e ricaricare il sistema ex novo (tenere in uso il catalogo dei servizi, le risorse e le conversioni ovviamente). È un modo eccellente per verificare il funzionamento dell'intero sistema nell'ambiente di produzione.

Per riepilogare, in questa fase sono inclusi i seguenti passaggi:

- Installare e connettere il sistema.
- Caricare il database del sistema configurato se necessario.

- Integrare le connessioni per posta elettronica, Exchange, SMS e così via.
- Verificare le funzionalità e ottimizzare le prestazioni.
- Installare, configurare e provare le connessioni remote.
- Reiniziare il sistema pronto per l'operazione reale.
- Attivare gli adapter dei dati per avviare il polling delle origini dati e il caricamento dei dati non elaborati.
- Accendere il motore CA Business Service Insight e avviare l'inizio dei calcoli.
- Completare la documentazione necessaria, ad esempio, per l'amministrazione del sistema, il database e altre procedure di manutenzione.
- Verificare di aver fornito la formazione completa agli utenti.

Preparazione

Per assicurare un'installazione efficiente, è importante preparare in anticipo quanto segue:

1. **Esportazione di database dell'ambiente di sviluppo.** L'esportazione del database deve essere eseguita utilizzando gli script approvati da CA che creano un file di estrazione (dump) per importarlo nuovamente nel software sul sistema di produzione.

Nota: è necessario che l'esportazione sia eseguita da un utente di database che disponga di privilegi sufficienti e che questo stesso utente sia accessibile durante l'importazione nel database. A tale scopo, è possibile utilizzare l'account **oblicore** visto che viene creato sempre in ciascun sistema o, in alternativa, l'account **sys**. Tuttavia, verificare che la password **sys** sia disponibile nel database di destinazione per consentire il processo di importazione.

2. **Script di database:** se il DBA desidera modificare gli script di creazione del database, quindi devono essere verificati in anticipo da un DBA di CA. Questi script devono essere verificati e pronti in anticipo per consentire un'agevole l'installazione. È frequente che organizzazioni diverse abbiano commenti e modifiche che devono essere approvati dall'amministratore di database di CA per garantire che la configurazione del database sia conforme ai criteri locali.

- **Connettività e accessibilità del server:** sia i server applicazioni, sia i server Web devono avere un client di connessione remota installato per supportare l'accessibilità esterna (se l'accesso fisico non è possibile al momento dell'installazione o in futuro per problemi di supporto). L'accessibilità esterna è importante anche per consentire il monitoraggio continuo del sistema durante un periodo di liquidazione. Se è necessario l'accesso remoto esterno, questa operazione potrebbe richiedere più tempo per l'organizzazione a causa delle implicazioni di protezione aggiuntiva; pertanto, deve essere affrontata in anticipo.

Le connessioni remote possono essere stabilite tramite i seguenti strumenti:

- PcAnywhere
- Host proxy/Master
- Microsoft Terminal Server
- Desktop remoto
- VNC
- Qualsiasi altro strumento di connessione remota supportato dall'organizzazione.

Note:

1. Microsoft Terminal Server si connette al server tramite la simulazione di un accesso al server, a differenza di PcAnywhere, VNC, proxy e altri strumenti.
2. Alcuni prodotti software Oracle non possono essere installati tramite Microsoft Terminal Server, e devono essere installati localmente.

3. **Accessibilità dell'origine dati:** l'origine dati deve essere accessibile per la modifica manuale e la connessione ODBC deve essere configurata in modo da consentire agli adapter di connettersi all'origine dati.
4. **Protezione del server:** sia nel server applicazioni, sia nel server Web è obbligatorio un profilo utente con diritti di amministratore locale. Se la piattaforma di database è un sistema operativo standard Windows, è comunque obbligatorio un profilo utente con diritti di amministratore locale. Per piattaforme UNIX, l'amministratore di database deve disporre dei privilegi appropriati per creare il database sul server.
5. **Accesso a Internet:** consente all'**amministratore di sistema** di connettersi a Internet per qualsiasi sistema operativo o aggiornamenti dell'applicazione se necessario.
6. **Un PC desktop utente standard, per verificare la funzionalità dell'applicazione.**
Nota: l'applicazione richiede l'utilizzo di alcuni controlli ActiveX da installare, spesso bloccati dai criteri di protezione dell'organizzazione.
7. **Programma di formazione con lo staff adeguato:** formazione introduttiva che consente allo staff di eseguire la formazione di accettazione utente e iniziare a lavorare con il sistema

Installazione

Il processo di installazione viene descritto in modo dettagliato nella Guida all'installazione e include le seguenti attività:

1. Installazione del database:

L'installazione del database è il compito dell'esperto delle sorgenti dati o del DBA Oracle, e in situazioni particolari deve essere sotto la direzione di un tecnico di CA. L'installazione del database include i passaggi seguenti:

- Installazione di Oracle sul server di database.
- Installazione della patch Oracle sul server di database (se necessario; deve sempre essere utilizzate le service release di supporto più recenti del software di database Oracle).
- Installazione del client Oracle per il server applicazioni/Web.
- Installazione della patch di client Oracle sul server applicazioni/Web (se necessario; verificare sempre che questa versione corrisponda al server).

2. Installazione dell'applicazione CA Business Service Insight:

L'installazione dell'applicazione viene eseguita dall'amministratore di sistema. Nei casi in cui l'installazione è già stata eseguita in un ambiente di prova (nella fase di configurazione) durante l'integrazione e la verifica dell'adapter, è richiesto solo lo stato di inizializzazione o l'importazione del database. Potrebbe quindi essere richiesta l'installazione nell'ambiente di produzione.

L'installazione dell'applicazione include i seguenti passaggi:

- Creazione del database CA Business Service Insight.
- Installazione del server applicazioni.
- Installazione del server Web.
- L'installazione dell'adapter.

Installare sempre la service release di CA più recente nel server applicazioni/Web. Durante l'installazione dell'applicazione e delle service release, gli script SQL sono inclusi per assicurare che il database sia aggiornato alla struttura corrente per la release. Questi script sono memorizzati nella directory di installazione dell'applicazione nel caso in cui il database richieda nuovamente l'aggiornamento in un secondo momento. Ad esempio, se è necessario importare il database da un backup che è stato creato in una versione precedente di service release. In questo caso, è necessario eseguire l'ultimo script di aggiornamento delle service release per assicurarsi che la struttura del database sia in linea con i componenti binari installati come parte della stessa service release.

Facoltativamente, è possibile installare anche altri strumenti di supporto, ad esempio sviluppatore PLSQL o altre utility SQL per agevolare la modifica dei dati di livello inferiore che potrebbe essere richiesta.

Importazione o esportazione tra ambienti (esperto delle sorgenti dati)

Questo scenario riguarda la migrazione degli adapter configurati e verificati in un ambiente di sviluppo/prova a un ambiente reale o di produzione. Si presuppone che l'ambiente di produzione sia stato già installato con un'installazione CA Business Service Insight standard e che il database sia presente.

Il processo è composto dai seguenti passaggi:

Per esportare il database e importarlo nel nuovo ambiente

1. Interrompere tutte le operazioni sul sistema di sviluppo e selezionare un punto di logico nel sistema per eseguire l'esportazione che può essere utilizzata per il sistema di produzione. Arrestare tutti i componenti del servizio CA Business Service Insight e i componenti COM+ CA Business Service Insight. Esportare utilizzando gli script di esportazione standard del sistema CA Business Service Insight. (Se necessario, contattare il Supporto tecnico di CA).
2. Eseguire la copia dell'estrazione del database e spostarla nel server di produzione pronto per l'importazione. Nota: le versioni Oracle dei database di sviluppo e produzione devono corrispondere. Importare il database tramite gli script di importazione del sistema standard (forniti con gli script di esportazione)
3. Una volta completata l'operazione, verificare la presenza di eventuali problemi di importazione. Se non sono presenti errori, assicurarsi di aver eseguito gli script SQL della service release più recente (se necessario)
4. Eseguire il collegamento Little Wizard (Procedura guidata breve) per configurare il database con le impostazioni del nuovo sistema di produzione.
5. Avviare tutti i componenti di servizio CA Business Service Insight e accedere al sistema di produzione per confermare che il sistema è stato importato correttamente.

Per eseguire la migrazione degli adapter

1. Installare l'adapter utilizzando l'utility Adapter Manager con le impostazioni simili all'adapter in fase di importazione (verificare che la denominazione sia la stessa perché il passaggio successivo funzioni correttamente).
2. Copiare il file di configurazione dall'adapter dal sistema di sviluppo alla cartella del nuovo adapter nel sistema di produzione. Sovrascrivere la configurazione predefinita fornita. (È necessario verificare che il file venga sovrascritto. In caso contrario, significa che la denominazione non è la stessa, causando problemi durante l'esecuzione.)
3. Aggiornare il file di configurazione dell'adapter. La comunicazione al server CA Business Service Insight e all'origine dati deve essere aggiornata per adattarsi al nuovo ambiente. La sezione OblicoreInterface deve essere aggiornata con la porta dell'adapter corretta. La sezione DataSourceInterface deve essere aggiornata con il corretto ConnectionString o modello di nome del file o percorso, se necessario.
4. Verificare che tutti i nomi delle origini dati (DNS) ODBC siano configurati e funzionanti sul nuovo computer.
5. Verificare la connettività dell'adapter.
6. Verificare l'esecuzione dell'adapter.
7. **Conversioni:** nei casi in cui sono presenti script di conversione che devono essere attivati. Verificare che siano sincronizzati con l'adapter e che le conversioni siano eseguite come previsto. Quando viene utilizzata la conversione manuale e non è completata in precedenza, in questa fase è necessario eseguire ulteriori conversioni.
8. **Pianificazione dell'adapter:** pianificare l'adapter in modo che sia eseguito come previsto. Se l'adapter è stato definito come un'applicazione in modalità di esecuzione unica, può essere pianificato utilizzando l'utilità di pianificazione di Windows. Può essere visualizzato in Pannello di controllo > Operazioni pianificate. Per ulteriori informazioni consultare la sezione [Modalità di esecuzione dell'adapter](#) (a pagina 85).

Integrazione - Configurazione del server di posta elettronica (amministratore di sistema)

Per attivare la funzionalità di notifica del sistema, è necessario sapere quale server di posta elettronica e quale casella di posta vengono utilizzati per inviare messaggi di posta elettronica a CA Business Service Insight. Il server di posta elettronica deve consentire l'inoltro della posta, poiché verrà inviata come SMTP dal server CA Business Service Insight a questo server di posta, utilizzando l'account specificato. Dopo aver completato la configurazione del server di posta elettronica, è possibile utilizzare le funzioni di posta elettronica nelle notifiche e nelle funzionalità di approvazione del contratto in CA Business Service Insight.

Fare clic sul menu Amministratore e selezionare Impostazioni sito, Notifiche. Configurare le definizioni di posta elettronica nella sezione delle impostazioni di notifica, inclusi il server di posta elettronica, l'indirizzo di invio e il nome del mittente, insieme alle informazioni sul provider SMS per l'utilizzo dei gateway SMS.

Parte della verifica di integrazione comprende di verificare che il server applicazioni possa raggiungere il server di posta dell'organizzazione e utilizzare questo per inviare notifiche CA Business Service Insight.

Per eseguire il test della connettività tra il server di posta e il server applicazioni, eseguire il comando seguente sul server applicazioni:

```
C:\Documents and Settings>telnet ORGANIZATION-MAIL 25
```

ORGANIZATION-MAIL

Indica il server di posta definito sul client di posta elettronica. Contattare l'amministratore di sistema per ottenere questa impostazione se non si è sicuri.

Per verificare il server di posta elettronica di client Outlook:

1. In Microsoft Outlook accedere a Strumenti > Account di posta elettronica e selezionare Visualizza o cambia gli account di posta elettronica esistenti.
2. Fare clic su Modifica.
3. Copiare il server di Microsoft Exchange. Questo server è il server di posta dell'organizzazione.

Se è presente una risposta dal server con questo comando, la connessione è stata completata correttamente. La risposta deve essere simile a quanto segue:

Se si riceve un altro messaggio, significa che non è presente alcuna connettività tra i due server. Contattare l'amministratore di sistema.

Impostazione delle preferenze di sistema (amministratore di sistema)

Impostazioni moduli

Listener dell'adapter: ☐ Mostra messaggi informativi

Motore di correlazione:

Numero di istanze ACE: 1

Frequenza di calcolo livello di servizio: * 11440 Minuti

Al completamento del calcolo: * Continua

Modalità di pianificazione: * Normale

☒ Limite intervallo di calcolo

Periodo consecutivo massimo da calcolare: * 100 Giorni

☐ Limita la memoria utilizzata dal motore di correlazione.

Avvia recupero memoria al superamento di: MB

Interrompi recupero memoria al raggiungimento di un valore inferiore a: MB

Servizio per il calcolo delle penali:

Frequenza di calcolo penali: * 60 Minuti

Eliminazione log di sistema:

☐ Inizializza parametri attività

Elimina messaggi di debug dopo: * 7 Giorni

Elimina messaggi di informazione dopo: * 15 Giorni

Elimina messaggi di avviso dopo: * 15 Giorni

Elimina messaggi di errore dopo: * 30 Giorni

Data ultima attivazione:

Il processo di impostazione delle preferenze di sistema include l'applicazione dei valori pertinenti alle variabili di sistema. Dal menu Amministrazione, fare clic su Impostazioni sito, quindi su Moduli per aprire la finestra di dialogo Engine Preferences (Preferenze motore).

Per informazioni sui suggerimenti relativi ai vari parametri, consultare la Guida dell'amministratore.

Impostazioni di protezione (amministratore di sistema)

Le impostazioni di protezione includono la creazione di utenti, gruppi utenti e ruoli. Per impostazione predefinita, tutti gli utenti sono associati all'organizzazione specificata durante il processo di installazione dell'applicazione. Tuttavia, è inoltre possibile creare altre organizzazioni, se necessario.

La maggior parte delle definizioni richieste è stata già completata durante la fase di configurazione, pertanto, in genere è necessario solo ottimizzare la definizione di impostazioni aggiuntive che potrebbero essere identificate da quel momento.

Per ulteriori informazioni sulle impostazioni di protezione, consultare la Guida in linea o la Guida per l'amministratore.

Definizione delle impostazioni per la sincronizzazione SSA

Quando si utilizza CA Spectrum Service Assurance (SSA) per il rilevamento dei servizi per CA Business Service Insight, è possibile configurare le impostazioni per abilitare la sincronizzazione automatica. Le funzionalità di automazione consentono di tenere l'elenco dei servizi aggiornato e i dati di servizio correnti.

Nota: è necessario disporre di accesso all'API restful in SSA per poter modificare queste impostazioni.

Attenersi alla seguente procedura:

1. Nel menu Amministratore, fare clic su Impostazioni sito, quindi su Impostazioni SSA.

Viene visualizzata la finestra di dialogo Impostazioni SSA.

2. Immettere le seguenti informazioni nell'area Autenticazione utente SSA:

URL del server

Specifica l'URL del server SSA di destinazione.

ID utente

Specifica l>ID utente per il server SSA.

Password

Specifica la password per l>ID utente del server SSA.

3. Immettere le seguenti informazioni nell'area Opzioni di sincronizzazione:

Sincronizzazione automatica

Specifica che si desidera eseguire la sincronizzazione automatica, in base alla frequenza di sincronizzazione (vedere il paragrafo seguente).

Set Synchronization Frequency (Imposta frequenza di sincronizzazione)

Specifica la frequenza per la ricerca di nuovi servizi. È possibile indicare un valore in ore o in giorni.

Sincronizzazione manuale

Consente di eseguire la sincronizzazione manuale dalla finestra di dialogo.

4. Immettere le seguenti informazioni nell'area Opzioni dei dati predefiniti:

Imposta i servizi predefiniti su:

Consente di impostare il valore predefinito su gestito o non gestito per i servizi rilevati.

Abilita il calcolo delle metriche di confronto

Consente di indicare l'azione predefinita per l'impostazione delle metriche di confronto per i servizi SSA.

5. Fare clic su OK.

Le impostazioni SSA sono attivate.

Appendice A: Domini del servizio e categorie di dominio di esempio

Nella seguente tabella vengono elencati i domini del servizio e le categorie di dominio comunemente utilizzati.

Dominio del servizio	Categoria di dominio	Commenti
Disponibilità del sistema	% di tempo disponibile	
	Numero di disservizi/tempi di inattività	
	Tempo medio di riparazione	
	Tempo medio tra errori	
Disponibilità del servizio	Minuti di inattività	
	Numero di giorni di interruzione	
Gestione finanziaria	Costo del servizio	
Prestazioni del processo	% di processi completati in tempo	
	Numero di cicli di processo completati	
Incident Management (Gestione incidenti)	% di incidenti risolti < T1	
	% di incidenti con risposta < T1	
	Numero di incidenti	
	% di incidenti convertiti in problema	
Soddisfazione del cliente	% di soddisfazione del cliente	
	Punteggio medio CSI	
Prestazioni di assistenza tecnica	% di chiamate abbandonate	
	Tempo medio di accettazione della chiamata	
	% FLLR	(percentuale di risoluzione di primo livello)
Qualità dei dati	% di accuratezza	
	% di tempestività	
	Numero totale di errori/difetti	

Appendice B: Esempi di case study

Questa sezione contiene i seguenti argomenti:

[Esempi di modellazione di contratto](#) (a pagina 215)

[Esempio di modellazione della metrica finanziaria](#) (a pagina 226)

[Esempi di modellazione dei dati](#) (a pagina 234)

[Esempio di utilizzo degli attributi personalizzati](#) (a pagina 245)

[Esempi di script di conversione](#) (a pagina 249)

[Esempi di implementazione di business logic](#) (a pagina 255)

[Esempi di scrittura di business logic efficace](#) (a pagina 271)

[Case study 19: procedura guidata di configurazione dell'adapter per origine dati basata su file](#) (a pagina 288)

[Case study 21: esempio di integrazione con LDAP](#) (a pagina 306)

[Case study 22: generazione di report con PERL](#) (a pagina 313)

Esempi di modellazione di contratto

Per ciascuno dei seguenti case study, utilizzare i seguenti elementi per la categorizzazione degli obiettivi descritti:

- Servizio
- Dominio del servizio
- Categoria di dominio
- Periodo di applicazione
- Destinazione
- Periodo di riferimento
- Unità di misura
- Schema di calcolo
- Parametri contrattuali o della metrica

Case study 1: disponibilità del sistema

Considerare la seguente garanzia contrattuale:

"Il sistema X deve essere disponibile per almeno il 99,6% del mese durante le ore lavorative".

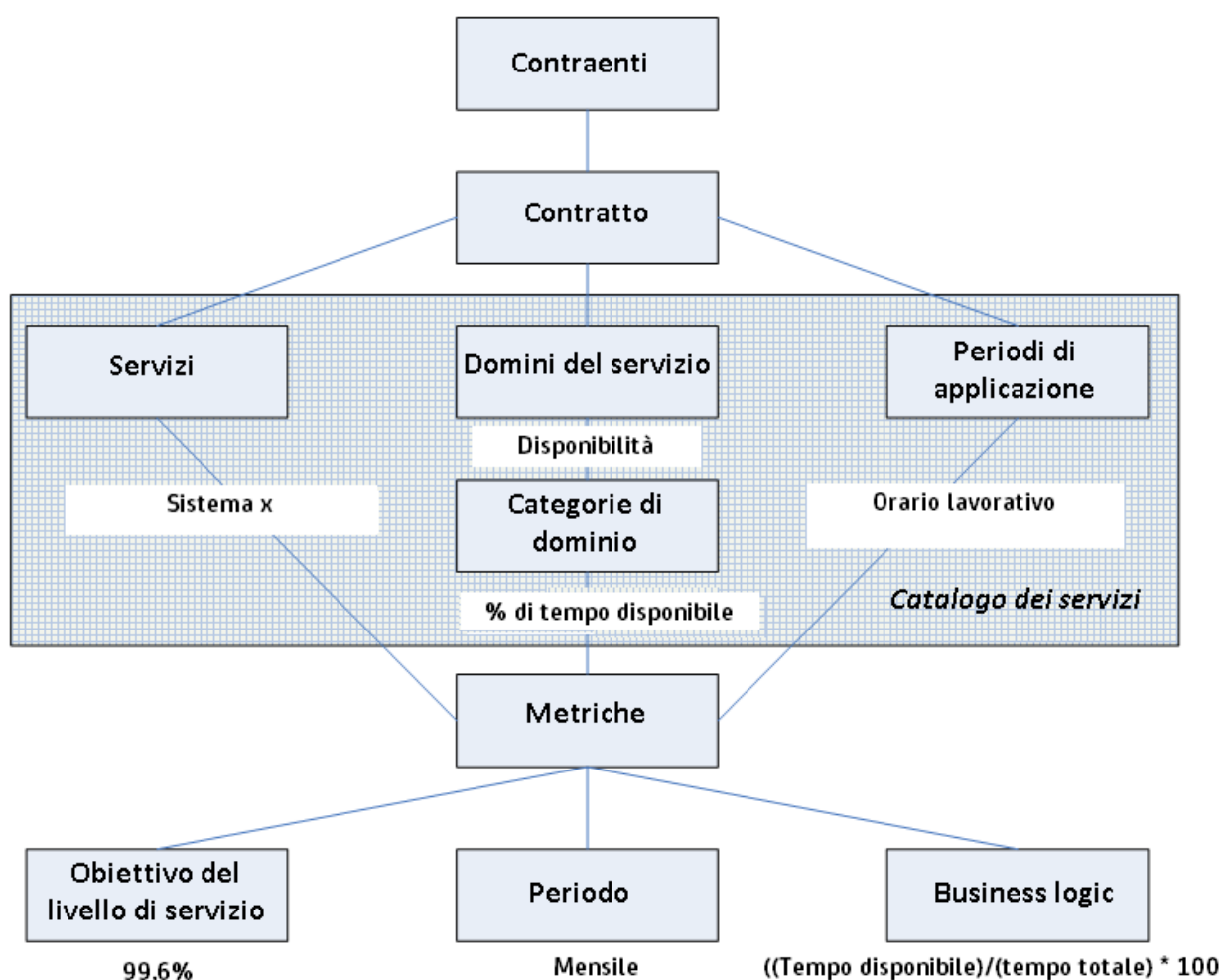
Potrebbe essere descritta utilizzando le seguenti entità CA Business Service Insight:

- **Nome metrica:** System X % of time available
- **Periodo di riferimento:** 1 mese
- **Periodo di applicazione:** ore lavorative (è necessaria un'ulteriore definizione successiva)
- **Business logic:** $((\text{tempo disponibile})/(\text{tempo totale})) * 100$
Nota: questo case study riguarda solo i casi in cui il monitoraggio rientra nelle ore lavorative (che corrisponde al periodo di applicazione della metrica)
- **Destinazione:** 99,6%

Oltre alle precedenti informazioni sulla metrica, dalla descrizione citata è possibile dedurre anche le seguenti voci dal catalogo dei servizi del sistema:

- **Servizio:** sistema X.
- **Dominio del servizio:** disponibilità.
- **Categoria di dominio:** % del tempo disponibile.
- **Gruppo di servizi:** qualsiasi gruppo che identifica più di un sistema per cui è possibile eseguire il monitoraggio. A questo punto, è difficile per stabilire se può essere creato un gruppo adatto.

Pertanto, ora è possibile riprodurre il diagramma dalla sezione di modellazione del contratto del presente documento in cui queste entità sono rappresentate in un diagramma.



Case study 2: disponibilità del sistema 2

La disponibilità del sistema CNP deve conservare i seguenti livelli:

Ambiente	Giorni feriali	Fine settimana
Produzione	99.9%	98.9%
Sviluppo	90%	nd
Test/QA	Nessuna garanzia	nd
Rete	99.9%	nd

Soluzioni suggerite:

Metrica	Produzione media del sistema CNP durante i giorni feriali
Destinazione	99,9
Periodo di riferimento	1 mese
Unità di misura	%
Servizio	Produzione del sistema CNP
Dominio del servizio	Disponibilità
Categoria di dominio	Disponibilità applicazione
Periodo di applicazione	giorni feriali
Metrica	Produzione media del sistema CNP durante i fine settimana
Destinazione	98.9
Periodo di riferimento	1 mese
Unità di misura	%
Servizio	Produzione del sistema CNP
Dominio del servizio	Disponibilità
Categoria di dominio	Disponibilità applicazione
Periodo di applicazione	fine settimana
Metrica	Sviluppo medio del sistema CNP durante i giorni feriali
Destinazione	90
Periodo di riferimento	1 mese
Unità di misura	%
Servizio	Sviluppo del sistema CNP
Dominio del servizio	Disponibilità
Categoria di dominio	Disponibilità applicazione
Periodo di applicazione	giorni feriali
Metrica	Test/QA medio del sistema CNP durante i giorni feriali
Destinazione	nessuno
Periodo di riferimento	1 mese
Unità di misura	%

Servizio	Test/QA del sistema CNP
Dominio del servizio	Disponibilità
Categoria di dominio	Disponibilità applicazione
Periodo di applicazione	giorni feriali
Metrica	Disponibilità di rete del sistema CNP
Destinazione	99,9
Periodo di riferimento	1 mese
Unità di misura	%
Servizio	Rete del sistema CNP
Dominio del servizio	Disponibilità
Categoria di dominio	Disponibilità di rete
Periodo di applicazione	Sempre

Case study 3: tempi di risposta del sistema

I seguenti case study illustrano le metriche dei tempi di risposta. Un contratto può essere modellato in diversi modi, ciascuno con i suoi vantaggi.

Nel seguente esempio vengono esaminati vari metodi di modellazione:

Soluzione A di modellazione suggerita	Valore massimo
Tempi di risposta transazione sistema CNP	Non può essere superiore a 750 millisecondi per mese
Nome della metrica	Maximum transaction response time
Destinazione	750
Periodo di riferimento	1 mese
Unità di misura	Millisecondi
Periodo di applicazione	Sempre
Servizio	Sistema CNP
Dominio del servizio	Prestazioni
Categoria di dominio	Maximum transaction response time

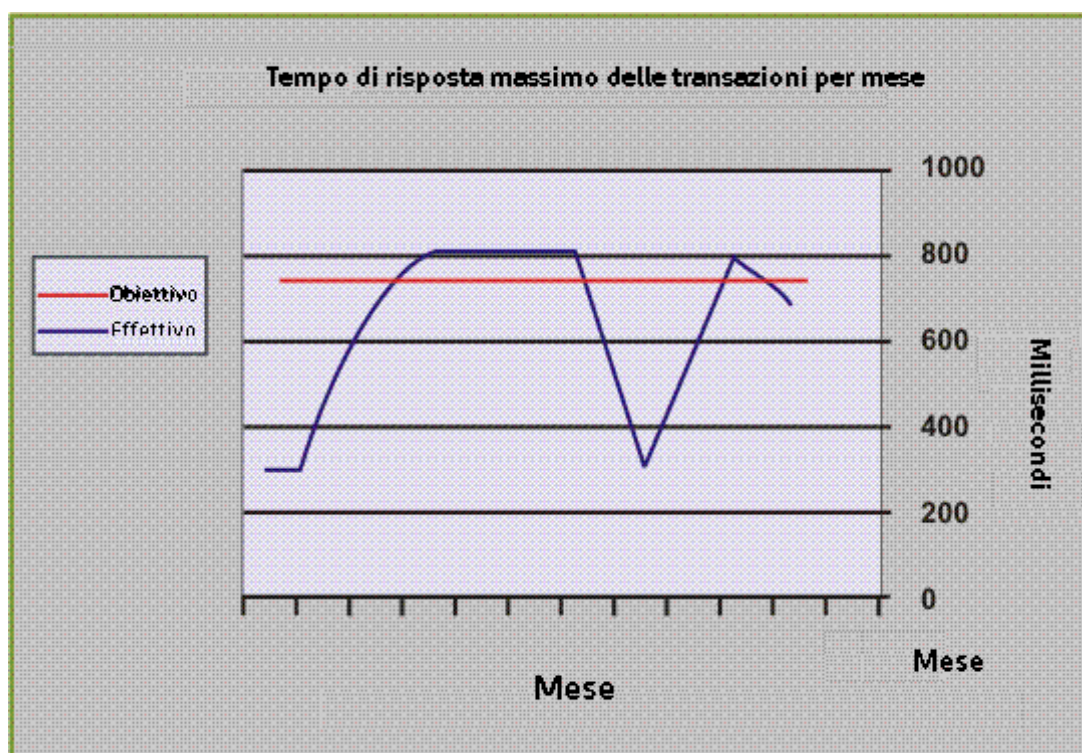
In base alle matrice sopra riportata, qual è il livello di servizio effettivo calcolato?

In base alla definizione della categoria di dominio, sembra che il livello di servizio effettivo sia calcolato come un valore massimo. Questo implica che per tutte le transazioni eseguite durante un mese, viene acquisita la transazione con il valore massimo e questo valore viene confrontato con la destinazione.

Nota: il calcolo del livello di servizio è basato su un'aggregazione di dati non elaborati in un determinato periodo di tempo. Per ogni periodo di tempo, la metrica fornisce un risultato singolo. La destinazione di una metrica non viene confrontata con una singola transazione, ma con un risultato mensile che è un'aggregazione periodica di tutte le transazioni entro tale periodo. Il responsabile contratti deve controllare che il risultato rispecchi da un lato il contratto e dall'altro la qualità del servizio.

Nota: la misurazione del tempo di risposta come valore massimo è un obbligo molto rigido e difficile da raggiungere nella prassi. Misurare un livello massimo significa che una singola transazione di 751 ms su un milione di transazioni eseguite nel corso di un mese è sufficiente per causare una violazione del contratto. Tutte le barre nei report saranno quindi rosse e non rifletteranno la vera qualità del servizio che è stata fornita.

Nella figura seguente viene illustrato un report tipico in queste circostanze.



Qualsiasi transazione che supera la destinazione verrà considerata una violazione del contratto, ma come base per comprendere l'effettiva qualità del servizio fornito è molto scarsa, poiché riflette solo una singola transazione e non si conosce nulla in merito al resto delle transazioni, ad esempio, si è trattato di un errore o di un trend? Se non è stato un incidente isolato, quindi quanti errori si sono verificati, o qual è il rapporto tra le transazioni non riuscite e il numero totale di transazioni eseguite durante il mese? Potrebbe esserci alcuni mesi con tali occorrenze e di conseguenza una violazione del contratto, ma qual è la tendenza? Sta migliorando o peggiorando? Tutte queste sono le domande che il manager del livello di servizio potrebbe chiedere e che il report deve essere in grado di fornire le risposte.

Nota: quando si definiscono la metrica e lo schema di calcolo associato, è molto importante prevedere come i risultati verranno visualizzati in un report. Il report deve fornire due elementi fondamentali:

- Si è verificata una violazione del contratto?
- Deve fornire ai manager sufficiente trasparenza dei dati e la possibilità di esaminare la causa dell'errore e anche offrire ai gestori gli strumenti necessari per comprendere a fondo i componenti del servizio.

Soluzione B di modellazione suggerita

Tempi di risposta transazione sistema CNP

Metrica

Destinazione

Periodo di riferimento

Unità di misura

Categoria di dominio

Tempo medio di risposta

Non deve essere superiore a 750 millisecondi per mese

Average transaction response time

750

1 mese

Millisecondi

Average transaction response time

Il calcolo del tempo medio di risposta fornisce un'idea migliore della qualità di servizio mensile e allo stesso tempo può ancora possibile riflettere quei mesi con tempi di risposta al limite o non conformi al contratto.

Soluzione C di modellazione suggerita

Tempi di risposta transazione sistema CNP

Metrica

Destinazione

Periodo di riferimento

Unità di misura

Parametro della metrica

Percentuale di transazioni completate correttamente entro la soglia.

Non deve essere superiore a 750 millisecondi per mese

Successful transaction response time

100

1 mese

Percentuale di successo

750 MS

Servizio	Sistema CNP
Dominio del servizio	Prestazioni
Categoria di dominio	Successful transaction response time
Periodo di applicazione	Sempre

Utilizzando questo metodo, verrà calcolata la percentuale di transazioni completate correttamente entro la soglia di 750 ms durante il periodo specificato, fornita dalla formula:

$$((\text{Numero di transazioni in 750 ms})/(\text{numero totale di transazioni}))*100$$

L'espressione della garanzia come una percentuale di successo offre la possibilità di mantenere una garanzia ristretta (destinazione al 100%), mentre consente anche il valore effettivo che rappresenta la scarsa o elevata qualità del servizio.

Utilizzando questo metodo, la destinazione non è un limite superiore di 750 MS, ma è il rapporto da conservare. Nei casi in cui la garanzia deve essere ristretta, la destinazione deve essere 100%, che non lascia spazio neanche un singolo errore. Nota: in tal caso, è stata introdotta un'ulteriore variabile, il parametro di metrica. Questo parametro deve essere implementato come parametro di metrica per consentire modifiche semplici, se necessarie.

Un modello alternativo a questo metodo può essere quello di forzare un modello di tipo di escalation:

Le seguenti tre soluzioni definiscono tre metriche invece di una singola, come nelle soluzioni precedenti.

Metrica	Successful transaction response time
Destinazione	95
Periodo di riferimento	1 mese
Unità di misura	Percentuale di successo
Parametro della metrica	750 MS
Metrica	Successful transaction response time
Destinazione	99
Periodo di riferimento	1 mese
Unità di misura	Percentuale di successo
Parametro della metrica	850 MS
Metrica	Successful transaction response time
Destinazione	100

Periodo di riferimento	1 mese
Unità di misura	Percentuale di successo
Parametro della metrica	1000 MS

Nel caso in cui sia necessario segnalare l'obbligo contrattuale, nonché il numero di transazioni che superano la soglia di 750 ms, è necessario definire un'ulteriore metrica per contare il numero di transazioni non riuscite.

Nota: ogni metrica genera un singolo risultato per un determinato periodo di tempo. Se è impostata per calcolare la percentuale delle transazioni, non è in grado di specificare il numero di transazioni.

L'unico modo per produrre report aggiuntivi da una metrica è utilizzare gli output dalla business logic. (Fare riferimento alla sezione [Output - Tabelle utente](#) (a pagina 155) in cui sono descritti i risultati di output dalla business logic).

Metrica	Number of Failed Transactions
Destinazione	Nessuna destinazione
Periodo di riferimento	1 mese
Unità di misura	Numero di transazioni
Parametro della metrica	750ms
Servizio	Sistema CNP
Dominio del servizio	Prestazioni
Categoria di dominio	Numero di transazioni
Periodo di applicazione	Sempre

Case study 4: prestazioni dell'assistenza tecnica

Case study che illustra una situazione di assistenza tecnica

L'assistenza tecnica deve ottenere il 100% di successo per tutti i seguenti elementi:

Tipo di ticket	Resolution Time (Tempo di risoluzione)
Priority 1	1 ora
Priority 2	2 ore
Priority 3	4 ore

Modellazione suggerita, soluzione A

Metrica	Tempo di risoluzione per priorità 1
Destinazione	100
Periodo di riferimento	1 mese
Unità di misura	Percentuale di successo
Parametro contrattuale	Matrice del tempo di risoluzione
Servizio	Assistenza tecnica
Dominio del servizio	Prestazioni di assistenza tecnica
Categoria di dominio	Tempo di risoluzione del ticket
Periodo di applicazione	Sempre

La matrice precedente è applicabile a tre metriche. Per ogni priorità, viene definita una metrica distinta con tutte le proprietà all'interno delle stesse categorie.

Modellazione suggerita, soluzione B

La definizione di metrica è la stessa, come illustrato nella soluzione A.

Opzione 1:

Servizio	Assistenza tecnica
Dominio del servizio	Priorità 3 di gestione ticket
Categoria di dominio	Tempo di risoluzione del ticket
Categoria di dominio	Tempo di risposta al ticket
Periodo di applicazione	Sempre

Opzione 2:

Servizio	Assistenza tecnica
Dominio del servizio	Gestione ticket
Categoria di dominio	Tempo di risoluzione del ticket con priorità 3
Categoria di dominio	Tempo di risposta al ticket
Periodo di applicazione	Sempre

Case study 5: backup del sistema

Il backup viene eseguito come indicato di seguito:

Intervallo di tempo	Numero di backup
settimanale	6
mensile	27
annuale	350

Soluzioni suggerite:

Metrica	Numero di backup settimanale
Destinazione	6
Periodo di riferimento	1 settimana
Unità di misura	Backup
Servizio	Backup
Dominio del servizio	Backup
Categoria di dominio	Numero di backup per settimana
Periodo di applicazione	Sempre
Metrica	Numero di backup mensile
Destinazione	27
Periodo di riferimento	1 mese
Unità di misura	Backup
Servizio	Backup
Dominio del servizio	Backup
Categoria di dominio	Numero di backup per settimana
Periodo di applicazione	Sempre
Metrica	Numero di backup annuale
Destinazione	350
Periodo di riferimento	1 anno
Unità di misura	Backup
Servizio	Backup
Dominio del servizio	Backup
Categoria di dominio	Numero di backup per settimana

Periodo di applicazione	Sempre

Esempio di modellazione della metrica finanziaria

Il seguente case study presenta un esempio di modellazione finanziaria.

Case study 6: modellazione delle condizioni finanziarie di un contratto/servizio

Esistono tre tipi generali di metriche utilizzati per la modellazione delle condizioni finanziarie di un servizio o di un contratto. Tali variabili sono:

- Costi fissi
- Costi a consumo
- Addebiti di penale/incentivi

Fare riferimento all'esempio seguente:

Una nuova azienda è in fase di avvio e richiede la fornitura del servizio di posta elettronica insieme all'installazione e alla manutenzione delle caselle di posta. Dal momento che verrà assunto nuovo personale, il numero delle caselle di posta ovviamente aumenterà. La previsione di un servizio di posta elettronica per un contratto comporta un costo fisso di 1000 \$, con un costo aggiuntivo per ogni casella di posta che verrà addebitato ogni mese. Il costo per ogni casella di posta è un modello di determinazione dei prezzi a più livelli, come indicato di seguito:

Numero delle caselle di posta	Costo per casella di posta
1-1,000	\$ 1,00
1.001 - 5.000	\$0.80
5,001 +	\$0.50

Pertanto, più caselle di posta vengono aggiunte, più il costo aggiuntivo si riduce. Ad esempio, 1500 caselle di posta costeranno $(1000 \times \$ 1) + (500 \times \$ 0,80) = \$ 1400$. Utilizzando questo modello, è possibile creare due metriche per riflettere questa situazione nel contratto.

- Costo del servizio di posta elettronica (fisso)
- Costo di utilizzo della casella di posta (variabile/in base al consumo)

Inoltre, è presente una stima da parte del team di gestione sul numero di membri del personale nel corso dell'anno (2007), come indicato di seguito. La tendenza dipende dalla crescita aziendale iniziale con l'assunzione e quindi dall'apertura di nuovi uffici in altre regioni:

Gen	Feb	Mar	Apr	Mag	Giu	Lug	Ago	Sett	Ott	Nov	Dic
50	100	500	900	1600	1700	1800	2500	2600	3500	3600	5800

Per modellare queste metriche, procedere come segue:

Creare nel contratto la metrica di costo fisso (utilizzare il tipo Elemento di costo), con le seguenti informazioni:

Generale	Dettagli	Note	Raggruppamento	Soglie	Metriche associate	Granularità	Dichiarazione dell'obiettivo	Registrazione	Business logic
Indicatore principale: Servizio: __SMI__ Dominio del servizio: * __SMI__ ↳ Categoria di dominio: * AccuracyChangesInPercent Unità di misura: * US Dollars Periodo di applicazione: * Always (Settimanale) Fuso orario: * GMT Obiettivo del livello di servizio: Valore di destinazione non superiore a \$ Destinazione dinamica: ☐ Periodo di riferimento: * 1 Mesi Unità di consumo: Prezzo unitario: Aggiungi nuovo Calcolo previsione: ☐ Previsione: Aggiungi nuovo Stato di misurazione: * Attivo									

Per specificare il costo fisso per il contratto del servizio, implementarlo come parametro nella business logic (in cui è necessario che il costo fisso sia restituito dalla funzione Result). Tale parametro può essere quindi visualizzato tramite la dichiarazione dell'obiettivo della metrica, come mostrato di seguito:

Generale	Dettagli	Note	Raggruppamento	Soglie	Metriche associate	Granularità	Dichiarazione dell'obiettivo	Registrazione	Business logic



SMI SMI Price Item US Dollars per 1 Mes

Campi	Tipo di metrica:	Price Item	↑
Parametri	Contraente:	i18n-test-ContractParty	↑
	Contratto:	i18n-test	↑

La restituzione del valore di parametro per questa metrica consiste solo nella restituzione del valore del costo di servizio tramite la funzione Result.

Generale	Dettagli	Note	Raggruppamento	Soglie	Metriche associate	Granularità	Dichiarazione dell'obiettivo	Registrazione	Business logic



```

Option Explicit

Sub OnRegistration(dispatcher)
'TODO: ADD code here TO REGISTER EVENTS NEEDED TO calculate service LEVEL
End Sub

Sub OnResourceStructureChanged(TIME)
End Sub

Sub OnMetricChanged(TIME)
'TODO: this code is executed at the beginning of every contract version
End Sub

Sub OnLoad(TIME)
'TODO: ADD code here TO handle calculation START event
End Sub

Sub OnPeriodStart(TIME)
'TODO: ADD code here TO handle period START event
End Sub

Sub OnPeriodEnd(TIME, isComplete)
'TODO: ADD code here TO handle period END event
End Sub
    
```

Successivamente, creare la metrica variabile di determinazione del prezzo (nuovamente, utilizzare il tipo Elemento di costo) per stabilire i costi in base al consumo del numero di caselle di posta utilizzate. Denominare questa metrica *Mailbox Consumption Cost* e crearla con le seguenti informazioni:

Generale	Dettagli	Note	Raggruppamento	Soglie	Metriche associate	Granularità	Dichiarazione dell'obiettivo	Registrazione	Business logic
Indicatore principale: Servizio: __SMI__ Dominio del servizio: * __SMI__ Categoria di dominio: * AccuracyChangesInPercent Unità di misura: * US Dollars Periodo di applicazione: * Always (Settimanale) Fuso orario: * GMT Obiettivo del livello di servizio: Valore di destinazione non superiore a \$ Destinazione dinamica: ☐ Periodo di riferimento: * 1 Mesi Unità di consumo: Hours Prezzo unitario: Calcolo previsione: ☐ Previsione: Stato di misurazione: * Attivo									

In questo esempio, è necessario immettere i parametri di consumo nelle informazioni relative alla metrica. Questi verranno inseriti nella tabella Prezzo unitario. Per modellare la tabella precedente per il numero di caselle di posta a fronte dei costi, creare una colonna per il limite superiore di caselle di posta e una colonna per i prezzi unitari.

Aggiungi colonna						
		Nome	Tipo	Indice	Predefinito	
		Consumption	Numero decimale	Sì		v
		Jul-01-2009	Numero decimale	No		v

Immettere quindi i valori per ogni livello. In questo caso, il limite superiore di caselle di posta determina la fascia di costo associata. Poiché sono presenti tre livelli, vengono aggiunti alla tabella in questo modo:

Aggiungi riga 

		Mailboxes	UnitCost
 		1000	100
 		5000	80
 		9999999	5

Oltre a questa operazione, implementare la funzione di previsione sul consumo delle caselle di posta. A tale scopo, creare la tabella Previsione con il layout mensile preimpostato.

Aggiungi colonna

		Nome	Tipo	Indice	Predefinito	
 		Consumption	Numero decimale	Sì		
 		Jul-01-2009	Numero decimale	No		 
 		Aug-01-2009	Numero decimale	No		 
 		Sep-01-2009	Numero decimale	No		 
 		Oct-01-2009	Numero decimale	No		 
 		Nov-01-2009	Numero decimale	No		 
 		Dec-01-2009	Numero decimale	No		 
 		Jan-01-2010	Numero decimale	No		 
 		Feb-01-2010	Numero decimale	No		 
 		Mar-01-2010	Numero decimale	No		 
 		Apr-01-2010	Numero decimale	No		 
 		May-01-2010	Numero decimale	No		 
 		Jun-01-2010	Numero decimale	No		 
 		Jul-01-2010	Numero decimale	No		 
 		Aug-01-2010	Numero decimale	No		 

Quest'ultimo viene quindi compilato con i valori dalle tabelle specificate nella descrizione del caso.

Consumo	Gen-01-2007	Feb-01-2007	Mar-01-2007	Apr-01-2007	Mag-01-2007	Giu-01-2007	Lug-01-2007	Ago-01-2007	Set-01-2007	Ott-01-2007	Nov-01-2007	Dic-01-2007
1	50	100	500	900	1600	1700	1800	2500	2600	3500	3600	5800

A questo punto, è possibile aggiungere la dichiarazione dell'obiettivo della metrica. In tal caso, non sono necessari valori di parametro in quanto derivati dalle tabelle Prezzo unitario e Previsione.

Generale	Dettagli	Note	Raggruppamento	Soglie	Metriche associate	Granularità	Dichiarazione dell'obiettivo	Registrazione	Business logic																																								
Nome carattere: 11px <Servizio> <Dominio del servizio> Price Item <Unità di misura> per <Periodo di riferimento> Mesi																																																	
<table border="1"> <thead> <tr> <th>Campi</th> <th></th> <th></th> <th></th> </tr> </thead> <tbody> <tr> <td>Tipo di metrica:</td> <td>Price Item</td> <td></td> <td>↑</td> </tr> <tr> <td>Contraente:</td> <td>i18n-test-ContractParty</td> <td></td> <td>↑</td> </tr> <tr> <td>Contratto:</td> <td>i18n-test</td> <td></td> <td>↑</td> </tr> <tr> <td>Data di inizio validità:</td> <td>01/07/2009 00:00</td> <td></td> <td>↑</td> </tr> <tr> <td>Data di fine validità:</td> <td>31/08/2011 00:00</td> <td></td> <td>↑</td> </tr> <tr> <td>Valuta:</td> <td>1</td> <td></td> <td>↑</td> </tr> <tr> <td>Simbolo di valuta:</td> <td>1</td> <td></td> <td>↑</td> </tr> <tr> <td>Indicatore principale:</td> <td> </td> <td></td> <td>↑</td> </tr> <tr> <td>Servizio:</td> <td> </td> <td></td> <td>↑</td> </tr> </tbody> </table>										Campi				Tipo di metrica:	Price Item		↑	Contraente:	i18n-test-ContractParty		↑	Contratto:	i18n-test		↑	Data di inizio validità:	01/07/2009 00:00		↑	Data di fine validità:	31/08/2011 00:00		↑	Valuta:	1		↑	Simbolo di valuta:	1		↑	Indicatore principale:			↑	Servizio:			↑
Campi																																																	
Tipo di metrica:	Price Item		↑																																														
Contraente:	i18n-test-ContractParty		↑																																														
Contratto:	i18n-test		↑																																														
Data di inizio validità:	01/07/2009 00:00		↑																																														
Data di fine validità:	31/08/2011 00:00		↑																																														
Valuta:	1		↑																																														
Simbolo di valuta:	1		↑																																														
Indicatore principale:			↑																																														
Servizio:			↑																																														

Infine, completare la business logic, come indicato di seguito:

Option Explicit

```
Dim PPUp1, PPUp2, PPUp3, PPUkey, FCmap, periodFC, TierPPU
Dim currentMonth, TotalMailboxes, MailboxesThisPeriod, TotalPrice
```

```
Sub OnRegistration(dispatcher)
    'registrazione solo di esempio
    dispatcher.RegisterByMetric "OnMailboxAddedEvent", "NewMailboxEventType", _
        "MailboxResource", "MONTH", "MetricName", "MetricContract", _
        "MetricContractParty"
End Sub
```

```
Sub OnLoad(TIME)
    'Inizializzare le mappe dei livelli di prezzo e le mappe di previsione
    Set PPUp1 = Context.Field ("Price Per Unit")(1)
    Set PPUp2 = Context.Field ("Price Per Unit")(2)
    Set PPUp3 = Context.Field ("Price Per Unit")(3)
    Set FCmap = Context.Field ("Forecast")(1)
End Sub
```

```
Sub OnPeriodStart(TIME)
    'TODO: AGGIUNGERE il codice qui PER gestire l'evento di INIZIO periodo
    currentMonth = GetMonth (time)
    If Context.IsInForecast Then
        periodFC = getForecastValue (currentMonth)
    End If
    MailboxesThisPeriod = 0
    TotalPrice = 0
End Sub
```

```
Sub OnPeriodEnd(TIME, isComplete)
    ' Calcolare il prezzo corrente di tutte le caselle di posta utilizzando il
    modello
    ' di determinazione dei prezzi a più livelli
    ' Viene impiegato un approccio globale attraverso ciascun livello per
    ' determinare il costo totale.
    TotalPrice = getMailboxCost (TotalMailboxes)
End Sub
```

```
Sub OnTimeslotEnter(TIME)
End Sub
```

```
Sub OnTimeslotExit(TIME)
End Sub
```

```
Sub OnMailboxAddedEvent(eventDetails)
    MailboxesThisPeriod = MailboxesThisPeriod + 1
    TotalMailboxes = TotalMailBoxes + 1
End Sub
```



```

End Sub

Funzione previsione
    Forecast = getMailboxCost (periodFC)
End Function

Funzione Target
    Target = Null
End Function

Function Result
    result = TotalPrice
End Function

Function getforecastvalue(q)
    getforecastvalue = FCmap (q)
End Function

Function getmonth(time)
    'questa funzione consente di recuperare il mese
    Dim lTime
    lTime = Tools.GetLocaleTime(time)
    getmonth = monthname (datepart ("m", lTime), True) & _
        "-0" & datepart ("d", lTime) & "-" & datepart ("yyyy", lTime)
End Function

Function getMailboxCost(num_boxes)
    'Funzione che calcola il costo delle caselle di posta utilizzando il modello
    di determinazione dei prezzi a più livelli
    Dim returnValue
    If num_boxes <= PPUMap1 ("Mailboxes") Then
        'Primo livello
        returnValue = num_boxes * PPUMap1 ("UnitCost")
        'Out.Log "Tier1: " & num_boxes
    Else If num_boxes > PPUMap1 ("Mailboxes") And num_boxes <= PPUMap2
("Mailboxes") Then
        'solo secondo livello
        returnValue = (PPUMap1 ("Mailboxes") * PPUMap1 ("UnitCost")) + _
            ((num_boxes - PPUMap1 ("Mailboxes")) * PPUMap2 ("UnitCost"))
        'Out.Log "Tier2: " & num_boxes
    Else If num_boxes > PPUMap2 ("Mailboxes") Then
        'terzo livello
        returnValue = (PPUMap1 ("Mailboxes") * PPUMap1 ("UnitCost")) + _
            ((PPUMap2 ("Mailboxes") - PPUMap1 ("Mailboxes")) * PPUMap2 ("UnitCost"))
        + _
            ((num_boxes - PPUMap2 ("Mailboxes")) * PPUMap3 ("UnitCost"))
        'Out.Log "Tier3: " & num_boxes
    End If
    getMailboxCost = returnValue

```

```
'Out.Log "Cost is: " & returnValue  
End Function
```

Nota: questo script di business logic gestisce sia i risultati del calcolo di previsione (utilizzando la tabella Previsione) sia del costo di consumo finanziario. Entrambi utilizzano la stessa formula `getMailboxCost()` che calcola la determinazione dei prezzi a più livelli in base alla tabella Prezzo unitario definita per questa metrica.

Esempi di modellazione dei dati

I due casi seguenti rappresentano il processo di modellazione dei dati di base, nonché alcuni aspetti più dettagliati che potrebbero essere implicati.

Poiché il processo di modellazione dei dati viene eseguito dopo il processo di modellazione del contratto, i requisiti di calcolo derivati dalle garanzie del contratto sono già noti, essendo stati identificati e definiti come parte del processo di modellazione del contratto.

È necessario che il modello di dati prenda in considerazione tutti i requisiti di calcolo.

Nei due case study seguenti, un requisito singolo o alcuni requisiti select vengono selezionati per dimostrare il processo di modellazione dei dati.

Case study 7: prestazioni del server

Si tratta di un tipico case study relativo alle prestazioni del server.

In base alla struttura dell'origine dati seguente:

Indicazione	Server	Misura	Timestamp (Data/ora)
disponibilità	Appserv01	1	03/01/2001 07:44
tempo di risposta	Appserv01	354,6	03/01/2001 09:58
carico CPU	Dbserve02	83%	03/01/2001 12:12
disponibilità	Appserv01	0	03/01/2001 14:26
carico CPU	Dbserve02	94.30%	03/01/2001 16:40
capacità	Firewall01	10%	03/01/2001 18:54
tempo di risposta	Dbserve02	476.89	03/01/2001 21:08
disponibilità	Appserv02	1	03/01/2001 21:24
tempo di risposta	Appserv01	774,4	03/01/2001 21:48
carico CPU	Dbserve01	83%	03/01/2001 21:52

In aggiunta vi sono i requisiti di calcolo seguenti:

Calcolare la percentuale di disponibilità di ciascun server applicazioni.

È necessario calcolare la disponibilità di ciascun server separatamente. Pertanto, per calcolare la disponibilità di ciascun server, è necessario ricevere gli eventi solo per tale server specifico. Inoltre, le origini dati contengono altri indicatori di prestazioni che non sono pertinenti ai calcoli della disponibilità (tempo di risposta, capacità, e così via), perciò è necessario filtrare gli indicatori di disponibilità e i relativi server.

Poiché i criteri di filtro in CA Business Service Insight sono Tipo di evento e Risorse, convertire i criteri di filtro dai valori dell'origine dati in una definizione di Risorsa e Tipo di evento.

In questo caso, l'indicatore è un valore ideale da convertire come un tipo di evento in CA Business Service Insight in quanto descrive logicamente il tipo di evento. Esiste un numero limitato di tipi, ad esempio, disponibilità, tempo di risposta, capacità e carico CPU. Ciò significa che per le metriche che calcolano la disponibilità del server, la registrazione viene eseguita per il tipo di evento disponibilità.

In questo caso, quando è presente un numero elevato di server ed è necessario calcolare la disponibilità di ciascun server, ne consegue la necessità di definire ciascun server come una risorsa. È quindi necessario eseguire il raggruppamento in un gruppo di risorse e la metrica verrà raggruppata in base a tale gruppo di risorse.

Modellazione suggerita:

Nome evento	Evento di disponibilità.
Comportamento	Riportato come modifica di stato di 0 o 1.
Campo Data/ora	Data/ora (solo campo ora nell'origine dati).
Campo Risorsa	Server (ogni server visualizzato nell'origine dati viene convertito in una risorsa CA Business Service Insight).
Campo Tipo di evento	Indicatore (ciascun valore in questo campo viene convertito in un tipo di evento in CA Business Service Insight. Esistono quattro i tipi di evento).
Campi Dati	La misura è 0 o 1 (solo per i record di disponibilità).

È necessario definire le seguenti allocazioni di risorsa:

Attributo Tipo di risorsa	Server applicazioni
Allocazione a Contraente	Ogni server applicazioni viene allocato al contraente, in cui il server pertinente esegue le proprie applicazioni. In questo modo viene consentita la registrazione per contraente che recupera tutti i server di conseguenza.
Allocazione a Servizio	Come sopra.
Allocazione a Gruppo di risorse	Facoltativa. In genere è necessario raggruppare le risorse quando il raggruppamento è obbligatorio.

Registrazione per

Infine, in base a tutte le definizioni precedenti:

Per la metrica di gruppo, che calcola la disponibilità di ciascun server singolarmente, la registrazione è per Risorsa.

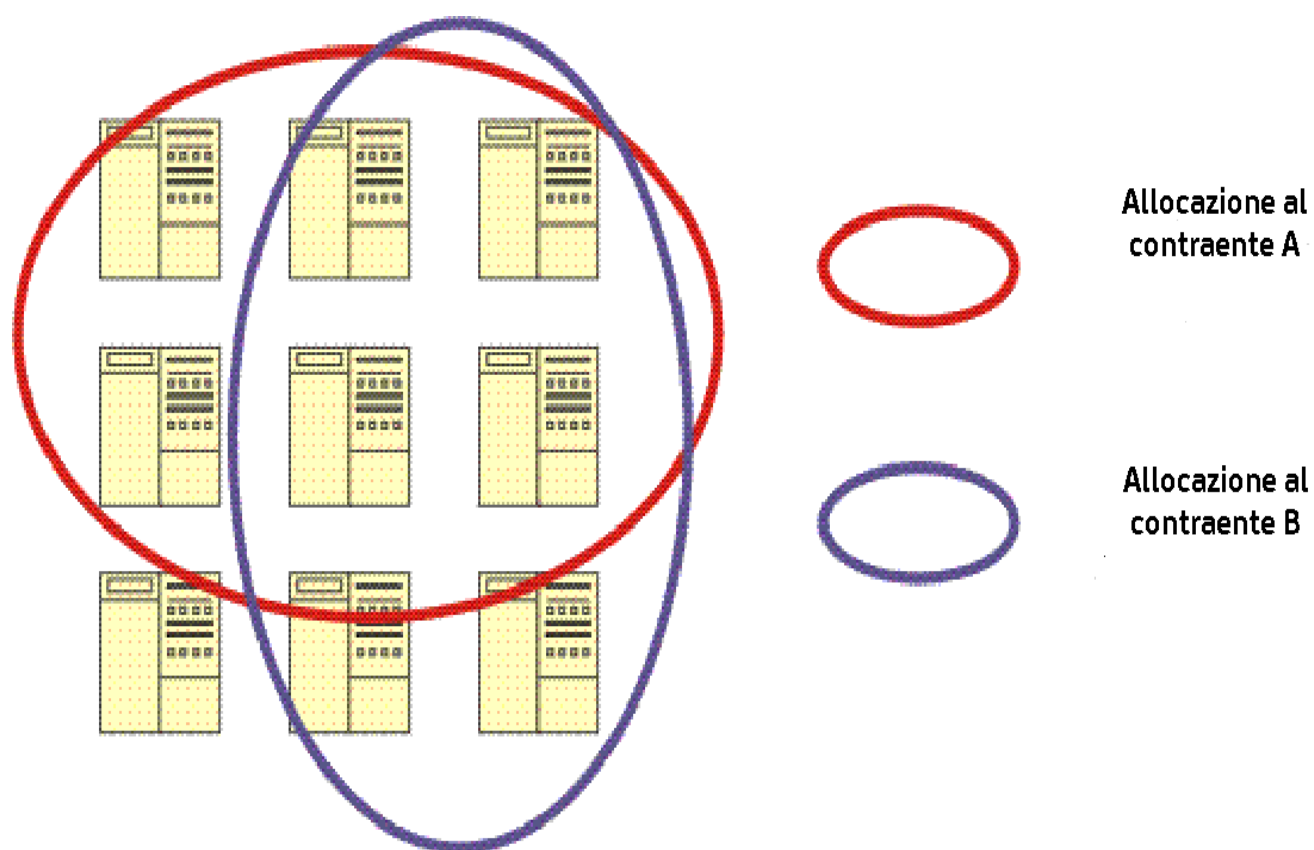
Per poter soddisfare i requisiti elencati viene aggiunto il criterio seguente:

Calcolare il tempo di risposta medio dei server applicazioni per ogni contraente separatamente.

Per questo requisito, è necessario ricevere gli eventi dei tempi di risposta per tutti i server applicazioni che fanno parte di un gruppo di server che eseguono le applicazioni per il contraente specifico. La ricezione degli eventi di tempo di risposta avviene tramite la registrazione del tipo di evento creato dal campo di indicazione con il valore "tempo di risposta". In questo modo viene assicurata solo la ricezione di eventi che segnalano i tempi di risposta dei server.

Per ricevere solo gli eventi che generano report sui server pertinenti per un contraente specifico, è necessario registrare le risorse mediante l'allocazione al contraente.

È possibile allocare una risorsa a più contraenti, servizi o gruppi. Di conseguenza, potrebbe verificarsi che un evento inviato per il calcolo dei tempi di risposta come parte del contratto di un contraente A sia anche parte dei calcoli per il contraente B.



Case study 8: prestazioni dell'assistenza tecnica

Si tratta di un tipico case study relativo alle prestazioni dell'assistenza tecnica.

In base all'origine dati illustrata di seguito:

N. ticket	TK. Priorità	Data/ora emissione	Data/ora chiusura	Data/ora risoluzione	Rif. cliente	Ubicazione
3800968	1	19/12/2003 07:51	05/01/2004 11:31	22/12/2003 12:00	CM3	Londra

N. ticket	TK. Priorità	Data/ora emissione	Data/ora chiusura	Data/ora risoluzione	Rif. cliente	Ubicazione
38000509	1	18/12/2003 09:21	05/01/2004 11:33	22/12/2003 12:00	CM1	Ipswitch
38084199	2	07/01/2004 11:20	14/01/2004 09:10	09/01/2004 12:00	CM2	Ipswitch
38188329	3	21/01/2004 09:27	27/01/2004 09:09	24/01/2004 12:00	CM3	Leeds
37964069	3	12/12/2003 14:04	05/01/2004 11:35	19/12/2003 12:00	CM286	Birmingham
3796448	1	12/12/2003 14:18	05/01/2004 11:39	19/12/2003 12:00	CM263	Luton
37965039	2	12/12/2003 14:57	14/01/2004 15:05	18/12/2003 12:00	CM264	Leeds
37970699	2	15/12/2003 09:26	05/01/2004 11:22	22/12/2003 12:00	CM288	Londra
37997259	1	17/12/2003 15:58	05/01/2004 11:27	23/12/2003 12:00	CM302	Ipswitch
38000259	1	18/12/2003 09:11	06/01/2004 14:44	29/12/2003 12:00	CM340	Londra
38021049	1	22/12/2003 09:32	06/01/2004 14:28	31/12/2003 12:00	CM341	Londra

L'origine dati illustrata in precedenza elenca le informazioni per i ticket di assistenza tecnica gestiti per ciascun cliente nelle diverse ubicazioni dei servizi cliente. Una singola ubicazione potrebbe anche essere condivisa tra i clienti.

I tre requisiti seguenti sono obbligatori per la creazione di report utilizzando questa origine dati:

- % di ticket con priorità 1 risolti in quattro ore per il cliente CM3.
- % di ticket con priorità 1 risolti in quattro ore per il cliente CM3 in ciascuna ubicazione.
- % di ticket con priorità 1 chiusi in un giorno per il cliente CM3 in ciascuna ubicazione.

Tali requisiti indicano che è necessario filtrare gli eventi per:

- Priorità
- Cliente
- Ubicazione

È necessario specificare quali di questi criteri di filtro vengono convertiti in Tipo evento e quali sono la risorsa pertinente.

Come è possibile selezionare un tipo di evento?

Dei tre criteri di filtro richiesti, la priorità del ticket è quello più appropriato per la conversione in Tipo di evento per i seguenti motivi:

- Descrive il tipo di evento da gestire (ticket con priorità 1).
- Il numero di priorità diverse esistente è alquanto ridotto (priorità 1, 2, 3).
- Il tipo di evento stesso è relativamente costante (solo raramente l'assistenza tecnica modifica le priorità con cui vengono gestiti i ticket).

Come viene selezionata una risorsa?

Gli altri due criteri di filtro richiesti sono il cliente e l'ubicazione e sono le entità più piccole che necessitano della creazione di report. Pertanto, la combinazione di cliente e ubicazione costituisce la risorsa.

Il cliente e l'ubicazione sono entità alquanto fisse con un ciclo di vita definito, in base al quale è possibile aggiungere nuovi clienti o nuovi siti. Inoltre, la relazione tra un sito e un cliente può variare.

Ai fini delle conversioni, è possibile utilizzare più campi dall'origine dati. Se nel precedente case study il campo del server è stato convertito in una risorsa CA Business Service Insight, in questo caso la risorsa viene creata utilizzando la combinazione dei due campi. Ogni permutazione pertanto genera una nuova risorsa.

L'elenco di risorse è indicato di seguito:

Campo Cliente	Campo Sito	Risorsa di output
CM3	Londra	CM3@London
CM1	Ipswich	CM1@Ipswich
CM2	Ipswich	CM2@Ipswich
CM3	Leeds	CM3@Leeds
CM286	Birmingham	CM286@Birmingham
CM263	Luton	CM263@Luton
CM264	Leeds	CM264@Leeds
CM288	Londra	CM288@London
CM302	Ipswich	CM302@Ipswich
CM340	Londra	CM340@London
CM341	Londra	CM341@London

La risorsa di output è la combinazione dei campi cliente e sito utilizzando il simbolo + per combinarli. È importante conoscere il nome della risorsa in quanto viene estratta dall'origine dati e viene visualizzata successivamente nei report. È necessario che le funzionalità dei report soddisfano le aspettative.

Nota: uno degli errori più comuni durante la modellazione dell'assistenza tecnica o di un ticket o di un sistema di incidenti consiste nella definizione di un solo incidente come risorsa.

I seguenti presupposti errati spesso sono causa di errori:

1. L'unico incidente è quello che viene segnalato nel report. Non impostare di segnalare l'entità come l'entità per la quale vengono generati i calcoli in modo che il singolo incidente non sia la base dei calcoli per il sito del cliente. In generale, lo SLA si basa sulle prestazioni complessive, non sulle prestazioni di gestione di un singolo ticket.
2. La garanzia è per livello di ticket. Si tratta di un errore perché le garanzie sono periodiche, l'oggetto dei calcoli è un'aggregazione nel corso del tempo.

Impostazione delle allocazioni di risorse

Per il primo requisito di calcolo:

1. % di ticket con priorità 1 risolti in quattro ore per il cliente CM3.
 - È necessario ricevere eventi di ticket attribuibili a uno specifico cliente. Il cliente è parte della risorsa, che inoltre indica il sito del cliente. Pertanto, per acquisire tutti gli eventi relativi alle risorse e attribuibili a tale cliente, sono disponibili due opzioni:
 - Nei casi in cui il cliente nell'origine dati rappresenta un contraente, è possibile associare le risorse al contraente pertinente. Si consente così la registrazione per contraente. Laddove possibile, questa opzione è sempre preferibile.
 - Creare un gruppo di risorse per ciascun cliente nell'origine dati e raggruppare tutte le risorse pertinenti all'interno di esso. Con questo metodo, la registrazione avviene per gruppo di risorse.

Per i due requisiti di calcolo successivi:

2. % di ticket con priorità 1 risolti in quattro ore per il cliente CM3 in ciascuna ubicazione.
3. % di ticket con priorità 1 chiusi in un giorno per il cliente CM3 in ciascuna ubicazione.

Per questi requisiti, raccogliere le risorse in gruppi di risorse in quanto è necessario che le metriche siano di gruppo, dato che il calcolo è richiesto per ogni sito singolarmente.

Nota: anche se le allocazioni di risorse per il modello corrente e i requisiti non sono necessarie, è importante crearle tenendo conto dei futuri requisiti. In tal modo si impedisce il sovraccarico nello sviluppo del sistema futuro.

Selezione del campo Data/ora

Come indicato in precedenza, il campo Data/ora è molto importante per la modalità di gestione degli eventi da parte del motore di correlazione. Le metriche calcolano sempre il livello di servizio per il periodo di tempo e gli eventi elaborati entro questo periodo sono quelli i cui valori Data/ora rientrano in tale periodo.

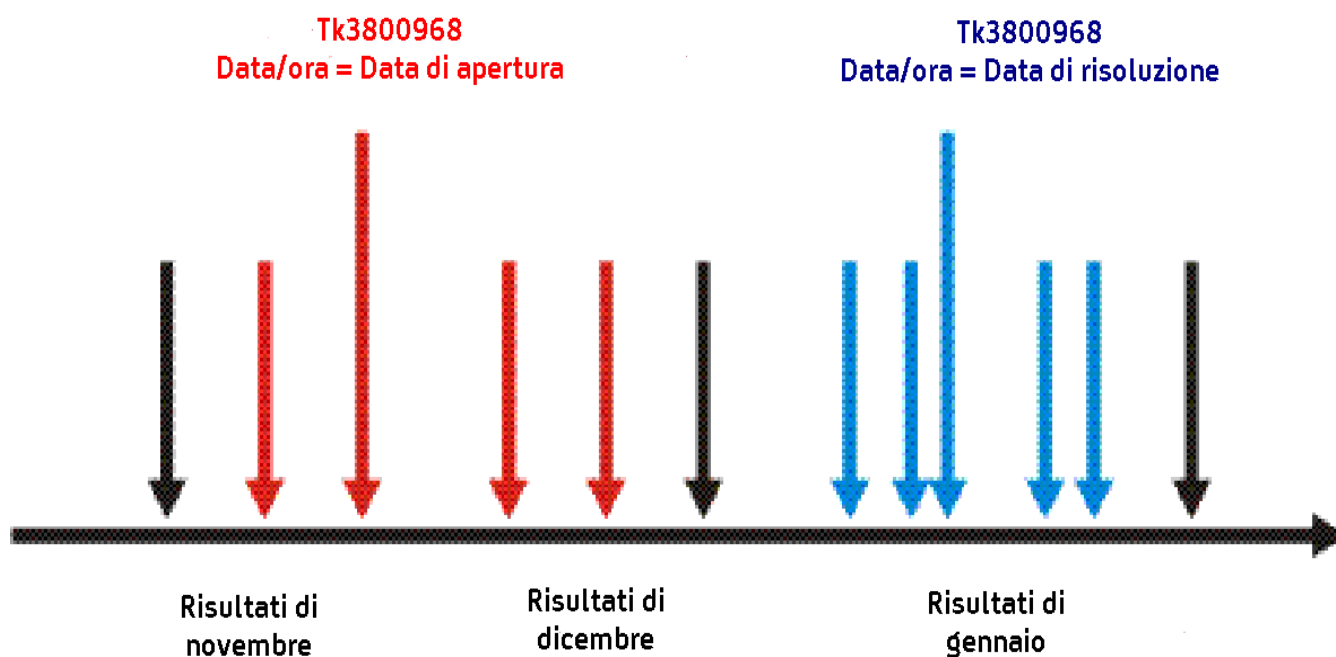
Nel primo case study, l'origine dati dispone di un solo campo ora. Tuttavia, in questo caso, sono presenti tre campi che è possibile impostare come data/ora. In base all'esame dei primi due record:

N. ticket	TK. Priorità	Data/ora emissione	Data/ora chiusura	Data/ora risoluzione	Rif. cliente	Ubicazione
3800968	1	19/12/2003 07:51	05/01/2004 11:31	22/12/2003 12:00	CM3	Londra
38000509	1	18/12/2003 09:21	05/01/2004 11:33	22/12/2003 12:00	CM1	Ipswitch

Per calcolare i tempi di risoluzione, sono obbligatorie sia l'ora di emissione del ticket, sia l'ora di risoluzione. Ai fini dell'evento, è possibile associarvi solo un valore Data/ora. Quindi, è possibile utilizzare gli altri come un valore all'interno dei campi valore.

Se il valore Raised at (Data/ora emissione) è utilizzato come Data/ora, il ticket viene incluso nei risultati di dicembre. Se il valore Resolved at (Data/ora risoluzione) è utilizzato come Data/ora, il ticket viene incluso nei risultati di gennaio. Entrambe le opzioni sono utilizzabili. È necessario che la selezione soddisfi le aspettative quando i ticket devono essere visualizzati nei report.

È un aspetto molto importante da considerare durante l'implementazione, in quanto determina quando gli eventi vengono utilizzati per i calcoli. Ad esempio, se un ticket viene emesso a novembre, ma non chiuso fino a dicembre, quando contribuisce rispetto al risultato dello SLA? Rientra nei dati di novembre o in quelli di dicembre?



In tal caso, poiché il ticket viene segnalato all'origine dati solo una volta chiuso, è possibile acquisire i dati solo dopo la chiusura del ticket. In genere, la data di chiusura è successiva alle date di emissione e risoluzione. Nel caso in cui la data di emissione è stata selezionata come Data/ora, è necessario quindi elaborarla come parte dei risultati di dicembre. La data di chiusura è stata a gennaio, il che significa che dicembre è già passato quando il ticket è stato segnalato. Pertanto, i risultati per dicembre sono stati già pubblicati. Il motore di correlazione quindi notifica che l'evento è nel passato, in quanto il valore Data/ora appartiene a dicembre, e attiva il ricalcolo. Di conseguenza, i risultati di dicembre vengono modificati in modo retroattivo.

È necessario comprendere a fondo tali conseguenze in modo da poter definire quale campo ora sia necessario selezionare come Data/ora. In genere, viene selezionata la data di chiusura in modo che non sia necessario modificare i report finali in modo retroattivo.

D'altra parte, l'utilizzo della data di chiusura come valore Data/ora ritarda l'immissione dei calcoli da parte dei ticket. Un ticket risolto potrebbe essere chiuso solo a una determinata ora successivamente.

Tuttavia, tenere presente che l'utilizzo della data di chiusura potrebbe inoltre attivare un processo nell'organizzazione in grado di accelerare i tempi di chiusura dei ticket.

Il suggerimento finale è quindi:

Nome evento	Ticket con priorità 1 (può essere definito anche per altre priorità se necessario)
Comportamento	Segnalato nel report quando il ticket viene chiuso
Campo Data/ora	Closed at (Data/ora chiusura)
Campo Risorsa	campo del cliente + campo del sito
Campo Tipo di evento	Priorità
Campi Dati	Tutti
Attributo Tipo di risorsa	Sito contraente
Allocazione a Contraente	Ogni sito è allocato al contraente pertinente
Allocazione a Servizio	Come sopra
Allocazione a Gruppo di risorse	Il gruppo di risorse viene creato per ciascun contraente per consentire il raggruppamento
Registrazione per	Per le metriche di gruppo, la registrazione è per Risorsa e la metrica è associata a un gruppo di risorse chiamato Siti CM3 cliente Per le metriche della data/ora di chiusura, la registrazione è per Contraente e per Servizio

Esempio di utilizzo degli attributi personalizzati

Il seguente case study presenta un esempio per più destinazioni dinamiche.

Case study 9: più destinazioni dinamiche

Considerare l'esempio di uno scenario in cui tutti i dispositivi dell'infrastruttura hardware nell'ambiente del cliente abbiano singole destinazioni impostate per i rispettivi requisiti di disponibilità. Utilizzando il metodo di modellazione standard, il compito sarebbe alquanto difficile da eseguire e richiederebbe numerosi raggruppamenti logici per i dispositivi e la gestione tramite il modello di risorsa. Per rendere lo scenario più complesso, le destinazioni per tali dispositivi possono cambiare nel tempo. I valori di destinazione vengono aggiornati in CA Business Service Insight da uno script di conversione man mano che le informazioni vengono archiviate in un CMDB esterno (consultare la sezione [Esempi ottimali di script di conversione](#) (a pagina 249) per un esempio di script di conversione)

In questa istanza la metrica potrebbe essere come indicato di seguito:

% of availability per hardware device.

Una procedura per modellarlo in modo efficace consiste nell'utilizzo della funzionalità Attributi personalizzati associata a un'altra funzionalità importante, Destinazioni dinamiche. Entrambe possono essere utilizzate con una metrica di gruppo per ottenere i risultati desiderati. L'aggiunta dell'obiettivo del livello di servizio direttamente alla risorsa consente alla business logic di confrontare il livello di servizio di ogni risorsa (dispositivo hardware) rispetto alla propria destinazione. Una metrica di gruppo consente la conformità del singolo servizio per ogni componente hardware utilizzando una metrica singola.

Pertanto, è necessario prima creare l'attributo personalizzato aggiungendolo al tipo di risorsa di tali dispositivi (tutti i dispositivi sono una risorsa di tipo Infrastructure Device (Dispositivi di infrastruttura)). L'attributo personalizzato creato è denominato DeviceTarget e può essere aggiunto dal menu in *Service Catalog (Catalogo dei servizi) > Attributi personalizzati*. Nota: quando si crea l'attributo personalizzato, è necessario collegarlo ai tipi di risorsa che lo richiedono.

Dettagli attributo personalizzato

Nome: *

Descrizione:

Tipo di valore: *

Valore predefinito:

Allegati:

[Aggiungi nuovo](#)

	Entità	Tipo	Obbligatorio	Valore predefinito
	Risorse	Infrastructure Devices	Sì	

A questo punto, durante la visualizzazione delle risorse nel sistema, è possibile notare che il nuovo attributo personalizzato è disponibile per il tipo di risorsa cui è stato collegato.

Risorse

Trova

[Aggiungi nuovo](#)

Visualizzazione risultati 1 - 9 di 9
(Limitato a 1000)

Risultati applicati a 31/08/2011 16:45

	Nome	Nome visualizzato	Descrizione	Data di creazione	Tipo di risorsa	Gruppo di risorse	Servizio	Contrainte	Fuso orario	Ultimo gruppo di mod.	DeviceTarget
	i18n-test GroupName	i18n-test GroupName	i18n-test Group Description	17/08/2011 19:10	i18n-test Resource Type				GMT	Default	
	i18n-test ResourceGroup	i18n-test ResourceGroupName	i18n-test ResourceGroupDescriptor	17/08/2011 19:12						Default	
	i18n-test ResourceName	i18n-test ResourceName	Added by translation script.	17/08/2011 19:33	Default					Default	
	Server111	Server111		29/08/2011 16:52	Infrastructure Devices					i18n-testChnageSetName	
	Server222	Server222		29/08/2011 16:52	Infrastructure Devices					i18n-testChnageSetName	
	Server333	Server333		29/08/2011 16:52	Infrastructure Devices					i18n-testChnageSetName	
	Server444	Server444		29/08/2011 16:53	Infrastructure Devices					i18n-testChnageSetName	
	Test_1	Test_1		29/08/2011 17:07	Infrastructure Devices					Device Target	
	Infraservers	Infraservers		29/08/2011 17:13						Device Target	

Inoltre, le singole risorse presentano un nuovo campo che può essere aggiornato.

Dettagli risorsa

Generale **Dettagli**

Aggiungi nuovo Filtra

Date 09/08/2011 00:00

In vigore:

Tipi di risorsa: *

Gruppi di risorse:

Servizi:

Contraenti:

DeviceTarget: *

Attributo personalizzato aggiunto

In questo esempio, il campo dovrebbe essere normalmente compilato/aggiornato dallo script di conversione.

Dal momento che ogni risorsa dispone di una propria destinazione specificata, è possibile definire la logica per eseguire il calcolo richiesto (dopo la conferma delle modifiche apportate alle risorse). Il codice di esempio riportato di seguito illustra la modalità di estrazione del valore di attributo personalizzato dalla risorsa (in grassetto).

Option Explicit

```
Dim TotalTime
Dim OutageTime
Dim PeriodStart
```

```
Sub OnRegistration(dispatcher)
    dispatcher.RegisterByResource "OnDeviceOutageEvent", "DeviceOutageEvent", _
        Context.ClusterItem
End Sub
```

```
Sub OnLoad(TIME)
    TotalTime = 0
    OutageTime = 0
End Sub
```

```
Sub OnPeriodStart(TIME)
    TotalTime = 0
    OutageTime = 0
    PeriodStart = TIME
End Sub
```

```
Sub OnPeriodEnd(TIME, isComplete)
    TotalTime = Tools.NetTime(PeriodStart, TIME)
End Sub
```

```
Sub OnDeviceOutageEvent(eventDetails)
    OutageTime = OutageTime + Tools.NetTime (eventDetails ("OutageStartTime"), _
        eventDetails ("OutageEndTime"))
End Sub
```

```
Funzione Target
    Target = eventDetails.CustomAttribute ("DeviceTarget")
End Function
```

```
Function Result
    If TotalTime > 0 Then
        Result = (TotalTime - OutageTime) / TotalTime
    Else
        Result = Null
    End If
End Function
```


Esempi di script di conversione

I seguenti case study includono un esempio di conversione automatica di base e aggiornamenti del modello di risorsa.

Case study 10: conversione automatica di base

L'esempio di script di conversione fornito in questa sezione è un esempio abbastanza semplice di elaborazione delle voci in sospeso nella schermata Voci di conversione.

Il gestore `OnTranslationEvent` effettua un semplice controllo sul primo carattere della risorsa ed esegue un'azione in base al valore: se è "a", la voce di conversione è impostata su Ignora, se è "b" viene eliminata, se è "c" viene convertita o, negli altri casi, viene mantenuta invariata per la conversione manuale. Nota: i contatori nell'intero codice consentono di tenere traccia delle azioni eseguite durante l'esecuzione dello script. È molto utile per il debug o la documentazione delle azioni eseguite dallo script ogni volta che viene eseguito, in particolare quando lo script è automatizzato. Il comando *Tools.Commit* alla fine della funzione è molto importante per tenere traccia, dal momento che senza di esso nessuna modifica apportata dallo script sarebbe salvata nel database.

La funzione `TranslateResource()`, quando viene invocata, verifica semplicemente se nel sistema esiste una risorsa con lo stesso nome di quella immessa con la voce di conversione in sospeso (con il prefisso E2E-). Se è inesistente, lo script aggiunge questa risorsa e quindi esegue la conversione. Se è già esistente, viene creata una voce di conversione dalla stringa della risorsa alla risorsa CA Business Service Insight esistente.

La funzione finale Result dello script restituisce semplicemente una descrizione delle attività eseguite dallo script. Di seguito è riportato il codice:

Option Explicit

```
dim translated
dim ignored
dim deleted
dim manually
dim ActionDate
```

```
Sub OnLoad()
    'tools.log "Translation Script: In OnLoad procedure", "I"
End Sub
```

```
Sub OnTranslationEvent(entryDetails)
```

```
    Dim dump
    dump = entryDetails.Dump
    tools.log dump
```

```
    Dim resource, entryId
    entryId = entryDetails.EntryId
    resource = entryDetails.FieldValue(1)
    ActionDate = entryDetails.LastActionDate
```

```
    If mid(resource,1,1) = "a" Then
        tools.IgnoreEntry entryId
        ignored = ignored + 1
        tools.log "ignored" & entryId & " " & resource
    Else If mid(resource,1,1) = "b" Then
        tools.DeleteEntry entryId
        deleted = deleted + 1
        tools.log "deleted" & entryId & " " & resource
    Else If mid(resource,1,1) = "c" Then
        TranslateResource resource, entryId
        tools.log "translated" & entryId & " " & resource
    Else
        tools.SetManualTranslationEntry entryId, 1
        manually = manually + 1
        tools.log "manually" & entryId & " " & resource
    End if
```

```
    Tools.commit
End Sub
```

```
Sub TranslateResource(resource, entryId)
```

```
    Dim newName
    Dim vector
    newName = "E2E-" & resource
```

```
if NOT tools.IsResourceExists(newName) Then
    Dim resourceDetails
    set resourceDetails = tools.GetResourceDetailsByDate(resource,ActionDate)
    resourceDetails("ResourceName") = newName
    resourceDetails("ResourceTypes") = "E2E Transactions"
    tools.AddResourceByMap resourceDetails
end if

tools.TranslateEntry entryId, newName
End Sub

Sub Main()
end Sub

Function Result
    Result = translated & "entries were translated, "& _
        ignored & "entries were ignored," & _
        deleted & "entries were deleted and "& _
        manually & "entries were set to manually update!"
End Function
```

Case study 11: aggiornamenti del modello di risorsa

Un altro possibile utilizzo degli script di conversione consiste nell'aggiornamento del modello di risorsa CA Business Service Insight con valori ricavati da un'origine dati esterna. Anche se non più strettamente correlato alla conversione, questo esempio delinea una funzione molto utile per gli aggiornamenti automatici del sistema.

Nella sezione precedente sugli attributi personalizzati è stato descritto uno scenario in cui i dispositivi di infrastruttura hardware di un'organizzazione vengono archiviati in un CMDB esterno, insieme alla destinazione di disponibilità prevista per ogni dispositivo. Queste informazioni devono essere replicate nel modello di risorsa CA Business Service Insight in modo da tenere aggiornato il mapping dell'infrastruttura (insieme alle destinazioni dei dispositivi).

In questo caso, è necessario che lo script esegua le seguenti attività:

- Aggiungere nuovi dispositivi hardware che attualmente non esistono nel sistema.
- Aggiornare la destinazione di disponibilità prevista dei dispositivi già esistenti (se la destinazione è stata modificata).

Di seguito è riportato lo script:

Option Explicit

```
'* * * * *  
* * * * *  
'Variabili globali e costanti  
'* * * * *  
* * * * *
```

```
Dim added  
Dim updated  
Dim ChangeSetName
```

```
added = 0  
updated = 0
```

```
Const RESOURCE_TYPE = "Infrastructure Devices"  
Const RESOURCE_GROUP = "InfraServers"  
Const CHANGESSET_NAME = "Infrastructure Devices"  
Const CHANGESSET_EFFECTIVE_DATE = "01/01/07"
```

```
'* * * * *  
* * * * *  
'OnLoad secondario:  
'Preparazione delle entità di infrastruttura basilari  
'* * * * *  
* * * * *
```

```
Sub OnLoad()
```

```

Tools.log "Translation Script : In OnLoad procedure", "D"

'Cercare la versione di risorsa preliminare esistente
Dim ChangeSetMap
Set ChangeSetMap=Tools.SearchChangeSets(CHANGESET_EFFECTIVE_DATE,
CHANGESET_NAME)

'Se non esiste alcuna versione creare una nuova versione
If ChangeSetMap.EMPTY Then
    Tools.AddChangeSet CHANGESET_NAME, CHANGESET_EFFECTIVE_DATE, CHANGESET_NAME
    Tools.Log "Changes set '" & CHANGESET_NAME & "' added."
End If

Set ChangeSetMap = Nothing
End Sub

Sub OnTranslationEvent(EntryDetails)
End Sub

Sub Main()
    Dim conn, rs, resource, deviceTarget, resource_id, resMap, custAttrib,
    custAttribValue
    Set conn = CreateObject("ADODB.Connection")
    Set rs = CreateObject("ADODB.RecordSet")
    conn.open "DSN=HardwareDevices"
    rs.Open "select * from tblServers", conn

    Do While Not rs.EOF
        resource = rs("serverName")
        deviceTarget= rs("DeviceTarget")
        'Aggiungere risorse all'ultima versione se non esiste già
        If Not Tools.IsResourceExists(resource) Then
            resource_id = Tools.AddResource(resource, CHANGESET_NAME,
"Infrastructure Device", RESOURCE_TYPE, RESOURCE_GROUP, Null, Null)
            Tools.UpdateResourcesCustomAttribute resource, CHANGESET_NAME,
"DeviceTarget", deviceTarget
            Tools.Log "AddingResource: " & resource & " with target: " &
deviceTarget & " ; assigned ID= " & resource_id
            added = added + 1
        Else
            Set resMap = Tools.GetResourceDetails(resource,CHANGESET_NAME, False)
            Set custAttrib = resMap("CustomAttributes")
            custAttribValue =
CDBl(custAttrib("DeviceTarget")("CustomAttributeValue"))
            If CDBl(deviceTarget) <> custAttribValue Then
                Tools.UpdateResourcesCustomAttribute resource, CHANGESET_NAME,
"DeviceTarget", deviceTarget
                Tools.Log "Updating Resource target for : " & resource & " from: " &
deviceTarget & " to " & custAttribValue
            End If
        End If
    Loop
End Sub

```

```
        updated = updated + 1
    End If
End If

Tools.Commit
rs.MoveNext
Ciclo

rs.Close
conn.Close
Set rs = Nothing
Set conn = Nothing
End Sub

Function Result
    ' Confermare la transazione
    Tools.CommitChangeSets CHANGESSET_NAME
    Result = added & " resources added and " & updated & " resources updated."
End Function

'* * * * *
* * * * *
```

Esempi di implementazione di business logic

Di seguito sono riportate alcune linee guida generali per l'implementazione della business logic:

Variabili globali

- Assicurarsi di inserire le iniziali dei valori globali dichiarati. Il meccanismo di stato PSL non è in grado di salvare le variabili con valori null

Codifica generale

- Prima di utilizzare uno degli oggetti elencati di seguito, verificare che sia esistente (tramite il metodo adatto `IsExist`):
 - Tutti i tipi di parametro (ad esempio, tabella, elenco, ecc.)
 - Attributo personalizzato
 - Risorsa
- Assicurarsi di immettere un modulo di business logic con tutti i parametri richiesti
- Prima di modificare il nome della risorsa verificare da quali metriche è in uso e aggiornarle di conseguenza

Mappe

- Utilizzo di mappe nelle metriche di gruppo: le mappe richiedono un maggiore sforzo di calcolo del motore; tenere presente che, in caso di raggruppamento su una metrica, lo sforzo di calcolo si moltiplica per il numero di elementi di gruppo.
- È necessario utilizzare grandi mappe globali all'interno della business logic di metriche di gruppo solo dopo un'attenta valutazione. Mentre il motore calcola una metrica di gruppo, è occupato con il caricamento di variabili globali dagli stati per ogni elemento nel gruppo separatamente
- Assicurarsi di cancellare le mappe e i vettori dopo aver terminato il lavoro con essi
- Se è necessario utilizzare mappe di grandi dimensioni, assicurarsi di gestire la mappa in modo efficiente suddividendola in intervalli logici.

Case study 12: utilizzo della logica di contatore per calcolare il numero di errori

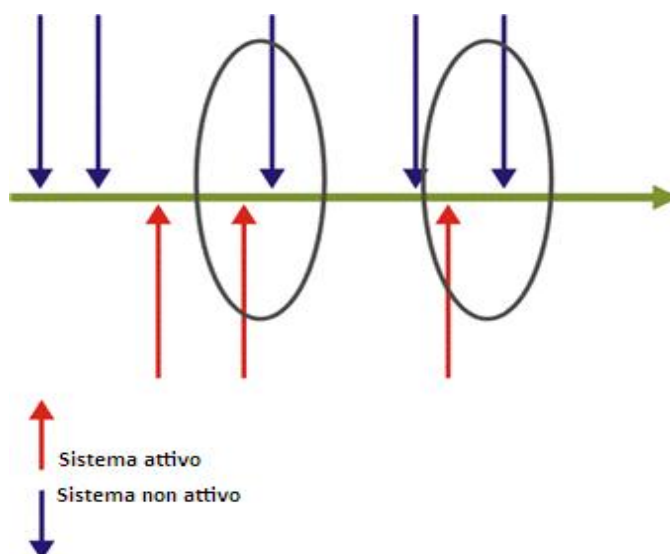
L'esempio seguente calcola il numero di errori che si sono verificati in un dato periodo di calcolo. La formula e i metodi utilizzati per implementarlo possono essere considerati come esempio di una formula richiesta quando è necessario calcolare qualcosa.

Vengono utilizzati i seguenti presupposti di calcolo:

- Eventi di input:
 - Evento di disponibilità, Stato (0/1)
 - L'evento di disponibilità viene ricevuto a intervalli di qualche minuto. Non è possibile presupporre la frequenza dell'evento (l'evento può verificarsi ogni cinque minuti, oppure una volta ogni ora) e inoltre possono esservi eventi ridondanti (ad esempio: 0, 0, 0, 1, 1, 0, ecc.).
- Periodo di applicazione: ore lavorative; gli errori che si verificano al di fuori del periodo di applicazione non vengono considerati.
- Il risultato richiesto è il numero di errori che si sono verificati durante il periodo.

Per conteggiare gli errori che si sono verificati durante il periodo di calcolo, è necessario archiviare una variabile di contatore periodico e anche una variabile che archivia lo stato del sistema. Poiché si potrebbero ricevere informazioni su eventi ridondanti (ad esempio, un evento Attivo seguito da un altro evento Attivo), è necessario inoltre calcolare il numero di ubicazioni in cui si è verificata una modifica dello stato di sistema da Attivo a Non attivo senza calcolare ogni volta che viene ricevuto un evento Non attivo, in quanto potrebbe essere un evento ridondante che rappresenta un errore già considerato.

Nella figura seguente è rappresentato graficamente il conteggio di stati Attivo e Non attivo del sistema.



Punti importanti da tenere in considerazione:

- **Costanti:** è consigliato l'utilizzo di definizioni di costante, invece che di valori costanti all'interno del codice. In questo modo, se il valore necessita di modifiche, è possibile modificare il valore solo nella definizione di costante senza dover eseguire la ricerca nel codice intero. Inoltre, è semplice modificarlo con l'uso di un parametro, se necessario. Nell'esempio precedente, i valori che rappresentano lo stato del sistema ricevuto nell'evento sono definiti come costanti in modo da rendere il codice più comprensibile, nonché per distinguere quando il numero zero viene utilizzato come un numero ai fini del conteggio dai casi in cui rappresenta uno stato del sistema.
- **Variabili globali:**
 - **Contatore:** la definizione della variabile del contatore è una definizione globale. Nella formula viene dichiarata nella parte superiore del codice e all'esterno di qualsiasi subroutine o funzione. È importante definire la variabile del contatore, in questo caso, come una variabile globale. In questo modo viene consentito l'utilizzo in diverse subroutine/funzioni all'interno del codice e inoltre viene consentito al sistema di mantenere i propri valori nell'intero periodo di calcolo e fornire i propri risultati al termine del periodo.

In questo esempio, è richiesto l'utilizzo della variabile del contatore in tre punti nel codice:

 - Per essere inizializzata all'inizio di un periodo.
 - Per essere anticipata in caso di un evento non riuscito nel gestore eventi.
 - Per essere restituita con la funzione Result al termine del periodo.
 - **Stato precedente del sistema:** questa variabile contiene lo stato del sistema e viene aggiornata ogni volta che viene ricevuto un nuovo evento con lo stato del sistema. Questa variabile deve inoltre essere globale poiché viene utilizzata in molte subroutine/funzioni nel codice, ad esempio:
 - Per essere inizializzata all'inizio del calcolo.
 - Per essere aggiornata quando viene ricevuto un nuovo evento.
- **Considerazioni sul periodo di applicazione:** per verificare se un errore interno avviene entro il periodo di applicazione, è possibile utilizzare il valore della proprietà *context.IsWithinTimeSlot*. Il contesto è un oggetto globale che può essere utilizzato da qualsiasi punto nel codice e, in questo caso, viene utilizzato per indicare quando viene ricevuto un errore e se si trova all'interno o all'esterno del periodo di applicazione. Se alla data/ora dell'evento il flag restituisce TRUE, quindi l'evento si è verificato all'interno di un periodo di applicazione, mentre se restituisce FALSE, indica che l'evento si è verificato al di fuori del periodo di applicazione. Secondo la logica richiesta, gli errori che si verificano al di fuori del periodo di applicazione vengono ignorati e pertanto non conteggiati.
- **Variabili di inizializzazione:**

- **Variabile del contatore:** contiene un valore periodico e viene pertanto inizializzata nel gestore *OnPeriodstart* per assicurarsi che ciascun periodo di calcolo conteggi gli errori separatamente. Ogni periodo inizia per zero e restituisce solo il numero di errori all'interno di questo periodo.

Nei casi in cui è richiesta per calcolare gli errori accumulati all'interno di ciascun periodo (vale a dire che il risultato di ogni periodo comprende tutti gli errori avvenuti fino al termine di tale periodo e compresi tutti i periodi precedenti), è necessaria per inizializzare il contatore solo nel gestore *OnLoad* e rimuoverla dal gestore *OnPeriodStart*. Pertanto, il contatore continua il conteggio e l'accumulo tra i periodi come indicato di seguito:

```
Sub OnLoad(time)
    FingerprInted=0
End Sub
```

- **Variabile dello stato di sistema:** deve essere inizializzata una volta all'inizio del calcolo e aggiornata, da quel punto in poi, su ogni evento di stato. Questa variabile memorizza il relativo valore per l'intero calcolo e non è limitata a un certo periodo di calcolo. È inoltre necessario che mantenga il relativo valore tra i periodi di calcolo. Poiché lo stato del sistema è sconosciuto prima di ricevere l'evento iniziale, è necessario assumere un presupposto in merito allo stato del sistema. Il presupposto comune è che il sistema sia Attivo. È necessario concordare e controllare tale presupposto dal momento che può comportare i casi seguenti:

Se, una volta avviato il calcolo lo stato del sistema era effettivamente Non attivo e il primo evento ricevuto indica tale stato Non attivo, verrà considerato un errore in quanto lo stato presupposto era Attivo. Questo errore viene conteggiato per il primo periodo di calcolo, anche se non si verifica necessariamente durante tale periodo.

```
Option Explicit
```

```
'Definizioni di costanti
Const UP=1
Const DOWN=0
```

```
'Variabile globale per il conteggio di errori
Dim FingerprInted
Dim SystemStatus
```

```
Sub OnRegistration(dispatcher)
    ' I parametri del metodo sono: <procedure name>, <Event Type>
    dispatcher.RegisterByContractPartyAndService
    "OnAvailabilityEvent", "AvailabilityEvent"
End Sub
```

```
Sub OnLoad(time)
    SystemStatus = UP 'presuppone il primo stato del sistema
End Sub
```

```
Sub OnPeriodStart(time)
```

```
FingerprInted = 0
End Sub

Sub OnAvailabilityEvent(eventDetails)
    ' In caso di un errore all'interno del periodo di applicazione, viene
    conteggiato
    If context.IsWithinTimeSlot and SystemStatus=UP and _
        eventDetails("Status")=DOWN Then
        FingerprInted = FingerprInted + 1
    End If
    ' aggiornare lo stato del sistema per il confronto successivo
    SystemStatus = eventDetails("Status")
End Sub

Function Result
    Result = FingerprInted
End Function
```

Case study 13: gestione del gruppo di componenti dinamico

Spesso è necessario archiviare i valori di un gruppo di risorse i cui membri possono essere dinamici e modificarsi durante il periodo di calcolo. Nel seguente calcolo di esempio dei requisiti, è necessario eseguire un calcolo intermedio su ogni risorsa per raggiungere il risultato finale di aggregazione.

Di seguito vengono riportati alcuni esempi di questo tipo:

- **Incidenti nei siti:** calcolo della percentuale di siti con incidenti la cui risoluzione ha richiesto un tempo superiore a X ore o conteggio dei siti con incidenti le cui medie di tempo di risoluzione sono state maggiori di X.

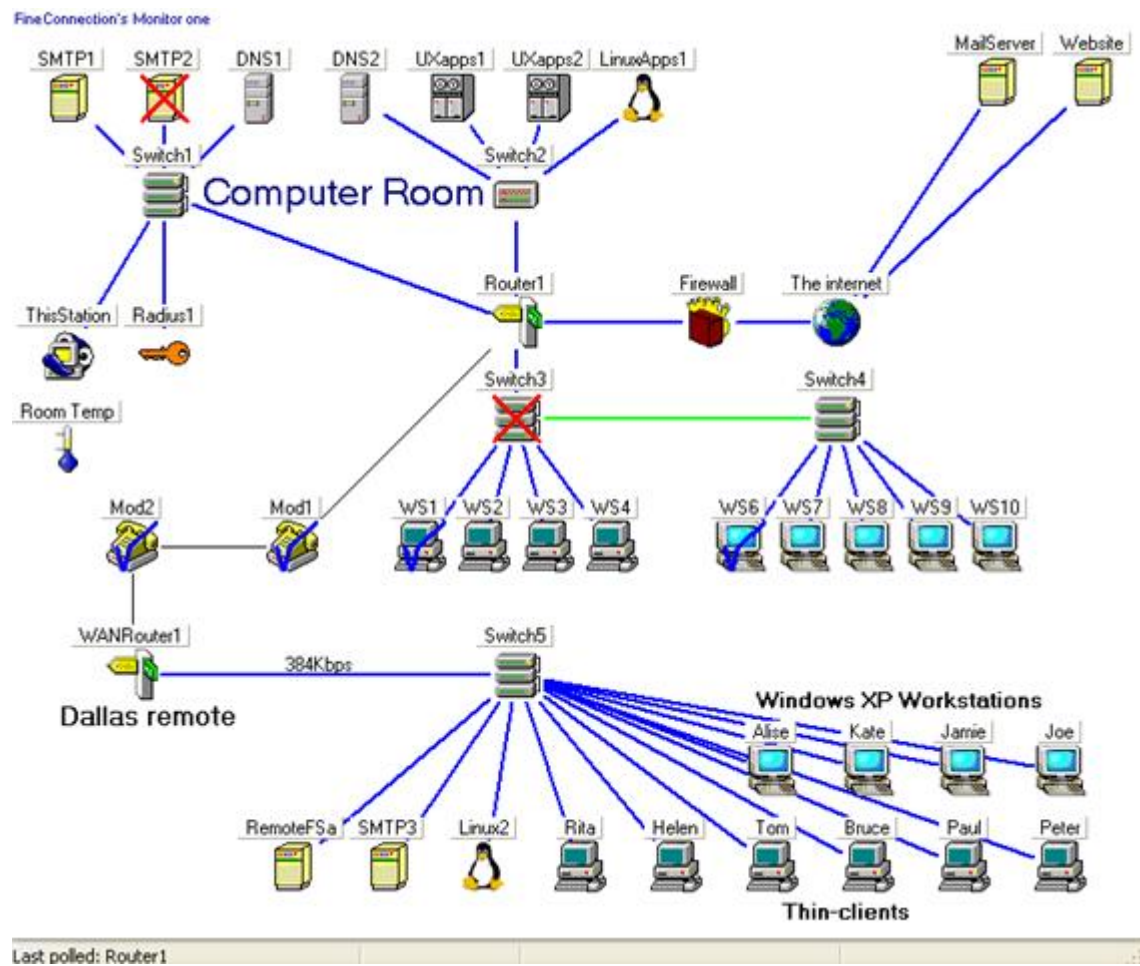
Negli esempi, i siti sono risorse con incidenti associati.

- **Disponibilità del server:** conteggio del numero di server il cui tempo di disponibilità è stato maggiore di X%.

Un server è una risorsa per cui è necessario valutare la percentuale di disponibilità.

- **Tipi di transazione:** calcolo della percentuale dei tipi di transazioni non riuscite più di X volte.

In questo caso, il tipo di transazione è una risorsa con eventi di errore associati. Per ogni tipo di transazione, viene memorizzato un contatore di errori come risultato intermedio e viene conteggiato il numero di diversi tipi di transazione con più di X errori.



Esempio:

Per il seguente requisito di calcolo:

Calcolare la percentuale di disponibilità per un sistema che è composto da un cluster di server. Il sistema viene considerato disponibile solo quando tutti i server di base sono disponibili.

La progettazione della soluzione viene implementata come indicato di seguito:

Viene valutata la disponibilità del sistema tramite la disponibilità delle relative risorse di gruppo di base. In questo caso, è necessario gestire e archiviare lo stato di tutte le risorse di gruppo per valutare lo stato del sistema in ogni punto nel tempo.

Per eseguire questa operazione, la formula deve disporre di una mappa (la mappa `ServerAvailabilityIndicators` nel codice di esempio riportato di seguito) che presenta una voce per ogni risorsa monitorata. Poiché tutti gli oggetti di mappa richiedono un campo chiave per fare riferimento al valore associato, questa mappa avrà l'indicatore di risorsa come campo chiave (che è l'ID risorsa) e il valore sarà lo stato del componente. Quando lo stato di un componente cambia, è necessario modificare la voce corrispondente nella mappa. In seguito a ogni evento di disponibilità, la formula aggiornerà lo stato della relativa risorsa nella mappa ed eseguirà una nuova valutazione dello stato del sistema di conseguenza.

Poiché le risorse monitorate possono cambiare (alcune potrebbero essere rimosse e altre aggiunte durante il periodo di calcolo), è necessario gestire le modifiche all'interno della formula con l'aggiunta di una funzione che identifica la modifica e aggiorna la mappa del componente monitorato con l'aggiunta di una nuova voce per il nuovo componente o con l'eliminazione di una voce se un componente è stato rimosso.

`OnRegistration` è il gestore eventi che gestisce un evento di registrazione attivato quando si verifica una modifica nelle risorse monitorate. Tale modifica può verificarsi in seguito alla conferma di una risorsa (o un gruppo di modifiche) che potrebbe apportare modifiche alle risorse incluse nel calcolo, in base al metodo di registrazione della formula.

Durante ciascuna registrazione è necessario aggiornare la mappa che contiene gli stati della risorsa con le modifiche necessarie. Ciò significa confrontare la mappa utilizzata per contenere gli stati della risorsa con la mappa che contiene le risorse quando è stata eseguita la registrazione (in base al metodo di registrazione) e includere tutte le risorse aggiunte o eliminare le risorse rimosse.

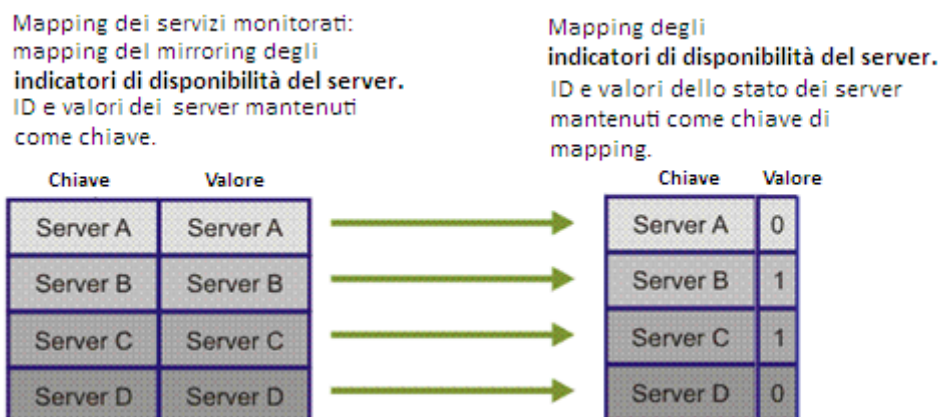
È necessario pertanto che la procedura `OnRegistration` esegua una funzione che confronti le risorse monitorate rispetto alle nuove risorse allocate in modo da strutturare le risorse monitorate di conseguenza.

L'oggetto di contesto comprende un insieme di metodi che confrontano i metodi di registrazione. Viene restituita una mappa con tutte le risorse che costituiscono una parte delle risorse in base al metodo di registrazione.

Nell'esempio seguente la registrazione della formula è `ByContractParty` e lo stesso metodo è pertanto utilizzato da `Context.ResourcesOfContractParty`. Viene restituita una mappa con tutte le risorse che costituiscono una parte di questo insieme al momento di registrazione.

Per confrontare le due mappe, è necessario scorrere le mappe in parallelo. È possibile scorrere le mappe mediante l'istruzione `For Each`. Questa istruzione consente di scorrere i valori di una mappa e, pertanto, è necessaria un'altra mappa speculare alla mappa degli stati in modo da poter scorrere le risorse e non i rispettivi valori di stato. Questa istruzione consente di scorrere i valori di una mappa e non le relative chiavi. Di conseguenza, è necessaria un'ulteriore mappa contenente gli ID sia come chiave e sia come valore. Inoltre, mantenere la mappa speculare per riflettere la mappa degli stati continuamente, in modo che abbia sempre lo stesso gruppo di risorse. Ciò significa che, quando la mappa degli stati viene aggiornata, è necessario aggiornare anche la mappa speculare.

La seguente figura mostra il concetto di mappe di mirroring.



Esempio:

```
Dim ServerAvailabilityIndicators
Dim MonitoredServers
Set ServerAvailabilityIndicators=CreateObject("SlalomMap.Map")
Set MonitoredServers=CreateObject("SlalomMap.Map")

Sub OnRegistration(dispatcher)
    dispatcher.RegisterByContractParty "OnAvailabilityEvent",_
        "Availability Event", "SAP Production Server"
    Dim AllocatedServers
    Set AllocatedServers = Context.ResourcesOfContractParty("SAP Production
Server")
    UpdateMonitoredServers AllocatedServers
End Sub

Sub UpdateMonitoredServers(allocatedServers)
    Dim Server
    For Each Server In allocatedServers
        If Not MonitoredServers.Exist(Server) Then
            MonitoredServers(Server) = Server
            ServerAvailabilityIndicators(Server)=True
        End If
    Avanti

    For Each Server In MonitoredServers
        If Not allocatedServers.Exist(Server) Then
            MonitoredServers.Erase Server
            ServerAvailabilityIndicators.Erase Server
        End If
    Avanti
End Sub
```

Esempio:

La funzione seguente illustra come viene utilizzata la mappa di mirroring per scorrere la mappa degli stati quando richiesto per valutare lo stato dell'intero sistema in base allo stato di ciascuna risorsa monitorata.

In questo esempio, il sistema viene considerato disponibile se tutte le risorse sono disponibili. La presenza di un singolo componente non attivo è sufficiente per considerare il sistema non attivo:

```
Function SystemAvailability
    Dim Server
    For Each Server In MonitoredServers
        If ServerAvailabilityIndicators(Server) = DOWN then
            SystemAvailability=DOWN
            Exit Function
        End if
    Avanti
```


End Function

Un esempio di business logic completa con la gestione di risorse dinamiche viene descritto nel seguente esempio di modello di progettazione.

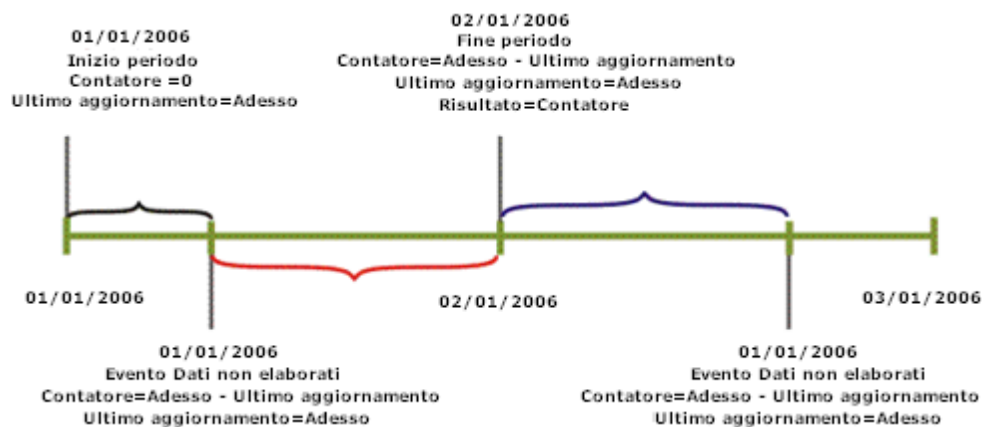
Case study 14: gestione clock di controllo del tempo

Il modello di progettazione descritto in questa sezione è adatto quando il risultato richiesto è una funzione di un periodo di tempo trascorso tra eventi, ad esempio:

- Tempo disponibile: calcolato come il tempo trascorso tra un errore e un altro.
- Tempo di risoluzione: calcolato come il tempo trascorso tra l'ora di apertura di un incidente e quella di chiusura.

Per accumulare tempo, è necessario assegnare una variabile in cui è possibile accumulare tempo (in secondi) e implementare una funzione che controlli le condizioni e il tempo accumulato dall'ultimo aggiornamento avvenuto. Questa funzione viene quindi eseguita per ogni evento ricevuto dalla formula.

La seguente illustrazione raffigura il clock di controllo del tempo.



La variabile LastUpdateTime memorizza l'ultima volta in cui è stato eseguito l'aggiornamento, a prescindere se il contatore di tempo fosse aggiornato oppure no. La funzione contiene la condizione che determina se è necessario aggiornare o accumulare tempo. Ad esempio, il tempo non deve essere considerato se supera il periodo di applicazione, lo stato del sistema è Non attivo o un incidente è nello stato in sospeso.

Sebbene la situazione delineata in questo caso spesso utilizza la funzione Tools.NetTime() per calcolare la durata, potrebbero esservi casi in cui è preferibile utilizzare la funzione VB standard DateDiff().

La funzione Tools.NetTime comporta un sovraccarico di verifica del periodo di applicazione ogni volta che viene utilizzata. Si consiglia di evitare l'uso di NetTime nelle procedure degli eventi di dati in quanto tali procedure sono invocate per ogni nuovo evento in arrivo e, pertanto, invocano la chiamata NetTime. Se il proprio periodo di applicazione è 24 ore per 7 giorni, si consiglia di utilizzare la funzione DateDiff per evitare il sovraccarico di verifica del periodo di applicazione.

Esempio 1:

La seguente routine dei contatori di aggiornamento accumula il periodo di tempo totale nella variabile PeriodNetTime. In AvailabilityTime la routine accumula il tempo in cui lo stato di sistema era Attivo, indicando che il sistema era disponibile.

L'oggetto Tools contiene il metodo NetTime, NetTime(beginTime, endTime)0. Questo metodo restituisce il numero di secondi tra beginTime ed endTime che rientrano nel periodo di applicazione della metrica corrente. Qualsiasi intervallo di tempo tra queste due variabili è parte di un periodo di applicazione escluso, perciò, è utilizzato molto comunemente per i calcoli di durata in cui viene impiegato un periodo di applicazione. Ad esempio: per i ticket con priorità 1 risolti in quattro ore lavorative, anche se un ticket veniva emesso alla fine di un giorno lavorativo e poteva non essere risolto fino al giorno successivo, è ancora incluso nello SLA per via delle ore del periodo di applicazione escluse.

```
Sub UpdateCounters (time)
    PeriodNetTime = PeriodNetTime + tools.NetTime (LastUpdateTime, time)
    If SystemStatus = UP Then
        AvailabilityTime = AvailabilityTime + tools.NetTime (LastUpdateTime, time)
    End If
    LastUpdateTime = time
End Sub
```

Esempio 2:

L'esempio seguente calcola la disponibilità dell'applicazione tramite la gestione degli eventi Attivo e Non attivo di diversi componenti critici, nonché gli eventi di manutenzione di tali componenti. Se tutti i componenti sono in manutenzione, il tempo non viene considerato come tempo di disponibilità prevista.

La subroutine UpdateCounters anticipa i contatori di tempo quando necessario e viene eseguita con ogni evento ricevuto per la formula (Evento di dati non elaborati/Evento del motore). Inoltre, aggiorna il tempo disponibile previsto nei casi in cui il tempo è compreso entro il periodo di applicazione e i componenti non sono all'interno del periodo di inattività pianificato. La formula aggiorna il tempo di disponibilità corrente solo quando dispone anche dello stato di sistema Attivo.

DateDiff è una funzione VB standard che restituisce il tempo tra due date, ma non esclude eventuali informazioni sul periodo di applicazione.

'Forzare la dichiarazione di variabile

Option Explicit

'Variabili globali

Dim ExpectedAvailabilityTime

Dim ActualAvailabilityTime

Dim LastUpdateTime

Dim AvailabilityIndicators

Dim MonitoredComponents

Dim DowntimeStatuses

'Mappare la creazione di oggetti

Set AvailabilityIndicators=CreateObject("SlalomMap.Map")

Set MonitoredComponents=CreateObject("SlalomMap.Map")

Set DowntimeStatuses=CreateObject("SlalomMap.Map")

'Dopo il caricamento e quando si verifica una modifica all'infrastruttura

Sub OnRegistration(dispatcher)

 dispatcher.RegisterByResourceGroup "OnComponentDownEvent", "Component
Down", "Application Components"

 dispatcher.RegisterByResourceGroup "OnComponentUpEvent", "Component
Up", "Application Components"

 dispatcher.RegisterByResourceGroup "OnMaintenanceStartEvent", "Maintenance
Start", "Application Components"

 dispatcher.RegisterByResourceGroup "OnMaintenanceEndEvent", "Maintenance
End", "Application Components"

 UpdateCounters Context.RegistrationTime

 Dim AllocatedComponents

 Set AllocatedComponents = Context.ResourcesOfResourceGroup("Application
Components")

 'assicurarsi che la formula controlli solo tutte le risorse pertinenti

 UpdateMonitoredComponents AllocatedComponents

End Sub

Sub OnLoad(time)

 'Quando il sistema è attivo per la prima volta, presupporre disponibilità OK

 LastUpdateTime = time

End Sub

```
Sub OnPeriodStart(time)
    'inizializzazione contatori per rieseguire il calcolo periodico
    ExpectedAvailabilityTime = 0
    ActualAvailabilityTime = 0
End Sub

Sub OnTimeslotEnter(time)
    UpdateCounters time
End Sub

Sub OnTimeslotExit(time)
    UpdateCounters time
End Sub

Sub OnComponentDownEvent(eventDetails)
    UpdateCounters eventDetails.Time
    'scrivere lo stato di disponibilità della risorsa segnalata
    AvailabilityIndicators(eventDetails.ResourceId) = _
        AvailabilityIndicators(eventDetails.ResourceId)+1
End Sub

Sub OnComponentUpEvent(eventDetails)
    UpdateCounters eventDetails.Time
    'scrivere lo stato di disponibilità della risorsa segnalata
    AvailabilityIndicators(eventDetails.ResourceId)= _
        AvailabilityIndicators(eventDetails.ResourceId)-1
End Sub

Sub OnMaintenanceStartEvent(eventDetails)
    UpdateCounters eventDetails.Time
    'scrivere lo stato di disponibilità della risorsa segnalata
    DowntimeStatuses(eventDetails.ResourceId)= _
        DowntimeStatuses(eventDetails.ResourceId)+1
End Sub

Sub OnMaintenanceEndEvent(eventDetails)
    UpdateCounters eventDetails.Time
    'scrivere lo stato di disponibilità della risorsa segnalata
    DowntimeStatuses(eventDetails.ResourceId)= _
        DowntimeStatuses(eventDetails.ResourceId)-1
End Sub

Sub OnPeriodEnd(time,isComplete)
    UpdateCounters time
End Sub

Function Result
    If ExpectedAvailabilityTime <> 0 Then
```

```
        Result = 100 * (ActualAvailabilityTime / ExpectedAvailabilityTime)
    Else
        Result = Null
    End If
End Function

Sub UpdateCounters(time)
    If Context.IsWithinTimeslot And Not AllComponentsAreInPlannedDowntime Then
        'aggiornare il contatore di secondi nel periodo di tempo (in caso di
        disponibilità prevista)
        ExpectedAvailabilityTime = ExpectedAvailabilityTime +
        DateDiff("s",LastUpdateTime,time)
        If SystemIsAvailable Then
            'aggiornare il contatore di secondi di disponibilità
            ActualAvailabilityTime = ActualAvailabilityTime +
            DateDiff("s",LastUpdateTime,time)
        End If
    End If
    LastUpdateTime=time
End Sub

Sub UpdateMonitoredComponents(allocatedComponents)
    Dim Component
    'aggiungere alla mappa dei componenti monitorati tutti i nuovi componenti da
    monitorare
    For Each Component In allocatedComponents
        If Not MonitoredComponents.Exist(Component) Then
            MonitoredComponents(Component) = Component
            AvailabilityIndicators(Component) = 0
            DowntimeStatuses(Component) = 0
        End If
    Avanti

    'rimuovere dalla mappa dei componenti monitorati tutti i componenti non più
    pertinenti
    For Each Component In MonitoredComponents
        If Not allocatedComponents.Exist(Component) Then
            MonitoredComponents.Erase Component
            AvailabilityIndicators.Erase Component
            DowntimeStatuses.Erase Component
        End If
    Avanti
End Sub

Function SystemIsAvailable
    Dim SystemAvailability
    SystemAvailability = True

    Dim Component
```

```
Dim ComponentAvailability
For Each Component In MonitoredComponents
    ComponentAvailability = AvailabilityIndicators(Component) = 0 _
        Or DowntimeStatuses(Component) > 0
    'la disponibilità di sistema è valutata con la disponibilità
    SystemAvailability = SystemAvailability And ComponentAvailability
Avanti
SystemIsAvailable = SystemAvailability
End Function

Function AllComponentsAreInPlannedDowntime
    Dim ComponentsInPlannedDowntime
    ComponentsInPlannedDowntime = 0
    Dim Component
    For Each Component In MonitoredComponents
        If DowntimeStatuses(Component) > 0 Then
            ComponentsInPlannedDowntime = ComponentsInPlannedDowntime + 1
        End If
    Avanti

    If ComponentsInPlannedDowntime = MonitoredComponents.Count Then
        AllComponentsAreInPlannedDowntime = True
    Else
        AllComponentsAreInPlannedDowntime = False
    End If
End Function
```

Esempi di scrittura di business logic efficace

Si consigliano le best practice di scrittura per una business logic efficace in relazione ai seguenti oggetti:

■ Metriche di gruppo

- Durante la valutazione del volume di sistema, conteggiare una metrica di gruppo come il numero di elementi nella metrica e tenere presente che tutto viene moltiplicato.
- Il ricalcolo di un elemento di gruppo comporta il ricalcolo dell'intero gruppo. Pertanto è necessario tenerne conto durante la pianificazione del raggruppamento, del modo in cui gli adapter vengono configurati e durante le modifiche alla struttura delle risorse
- Gli stessi eventi di dati non elaborati che rientrano in diversi elementi di gruppo hanno un elevato costo delle prestazioni (cambio di contesto)

■ Variabili globali

- Ottenere i parametri e i valori di attributo in ciascun punto del codice in cui sono necessari. Evitare di tenerli nella variabile globale, in particolare nelle metriche di gruppo (aumenta la dimensione degli stati)
- Evitare la logica che elabora mappe di grandi dimensioni. Al contrario, elaborare ogni evento con il metodo OnXXEvent
- Rimuovere gli elementi dalle mappe appena possibile. Ad esempio, quando un ticket viene chiuso e non al termine del periodo

■ Modelli di progettazione

Il pacchetto del contenuto predefinito contiene diversi modelli di progettazione per scenari comuni. L'utilizzo di questi modelli di progettazione consente di migliorare le prestazioni

■ Funzionalità integrata

ACE dispone di una funzionalità integrata e di strumenti per diversi scopi, come indicato di seguito:

- Funzionalità del periodo di applicazione
 - NetTime
 - IsWithinTimeslot
- Ora di eventi
 - TimeOfLastEvent
 - TimeOfLastEventHandler
- Oggetto di contesto
 - Contiene molti metodi distinti a seconda dell'ambiente
 - Utilizzare questi metodi ed evitare Safe ODBC

■ **Output di business logic**

Mantenere la struttura in T_SLALOM_OUTPUTS. Ciò significa che se si dispone di più tabelle logiche in T_SLALOM_OUTPUTS con una struttura simile, è molto utile inserire campi logici analoghi nello stesso campo fisico. Viene così consentita una facile indicizzazione per migliorare le prestazioni dei report

■ **Riusabilità dell'evento**

Da impiegare quando:

- Diverse metriche eseguono il calcolo di prima fase, di per sé richiesto per il contratto, e inviano gli eventi a una metrica di riepilogo che calcola il risultato (ad esempio, il calcolo finanziario, KPI di livello elevato)
- Una singola metrica esegue un'aggregazione preliminare sui dati non elaborati e invia gli eventi a molte altre metriche. Ammessa quando la metrica invia meno eventi rispetto a quelli che riceve o esegue calcoli significativi che altrimenti vengono eseguiti più volte

Da evitare quando:

- Aumenta significativamente il numero di metriche
- Vengono implementati più di tre livelli
- La complessità dell'implementazione aumenta e la manutenzione diventa più difficile

■ **Nuovi calcoli**

- Evitare ricalcoli pesanti come parte del normale funzionamento del sistema
- I motivi per eseguire nuovi calcoli sono:
 - Dati non elaborati con data/ora nel passato
 - Univocità dell'evento che modificherebbe i dati non elaborati nel passato
 - Correzioni
 - Eccezioni
 - Modifiche nei moduli di business logic
 - Modifiche apportate a un contratto
 - Eventi di riusabilità dell'evento con data/ora nel passato
 - Modifiche nella struttura delle risorse
- Prendere in considerazione l'utilizzo della funzionalità di blocco dei dati calcolati

■ **Moduli di business logic**

- È necessario scrivere i moduli di business logic in modo che, una volta completata la revisione, non debbano essere modificati
- La linea guida è: un modulo equivale a una logica generica

- I moduli di business logic molto specifici non possono essere utili a molte metriche e non favoriscono il codice né la riusabilità della logica
- I moduli di business logic troppo generici sono difficili da gestire. Inoltre, se un modulo di business logic implementa molte logiche complesse, la correzione in un flusso (utilizzata da parte delle metriche) comporta il ricalcolo di tutte le metriche

■ **Registrazione**

- Eseguire il filtraggio completo degli eventi mediante la registrazione. Lasciando la funzione di filtro al codice comporta un notevole impatto sulle prestazioni
- Semplificare il processo
- Per il codice che non è la registrazione stessa, utilizzare i metodi `dispatcher.IsRunTimeMode` e `OnResourceStructureChange`, in particolare quando sono presenti più modifiche nelle risorse
- Evitare l'utilizzo del metodo `RegisterByEventType`
- Nei moduli di business logic, utilizzare un modulo generico (per contraente, servizio, tipo di risorsa) oppure utilizzare i parametri o lasciare l'esecuzione della registrazione mediante l'interfaccia utente (opzione consigliata per la riusabilità dell'evento)

Case study 15: metriche di gruppo

In genere, nella descrizione di un determinato componente software, questa può essere divisa in due parti, WHAT e HOW. Per WHAT si intende la descrizione del significato di questa parte di codice. Per HOW si intende la procedura eseguita. Esiste una tendenza a concentrarsi sulla parte WHAT, trascurando la parte HOW. Il motivo di ciò è semplice e in molti casi giustificato. In questo modo, è possibile ridurre l'accoppiamento tra i componenti ed evitare di tediare l'utente con informazioni in molti casi non pertinenti. In molti casi, tuttavia, il costo di trascurare la parte HOW si traduce in scarse prestazioni.

Il presente case study descrive il modo in cui il motore calcola le metriche di gruppo (risposta alla parte HOW) e descrive il costo delle prestazioni implicate in determinate implementazioni. Inoltre, vengono descritti diversi metodi per ridurre questo costo modificando l'implementazione.

Che cosa sono le metriche di gruppo

Le metriche di gruppo sono metriche che comprendono nella propria definizione un certo gruppo di risorse. Questo gruppo è indicato come Gruppo di metrica e ogni risorsa in tale gruppo è definito come elemento di gruppo. Durante il calcolo di una metrica di gruppo viene eseguito un calcolo distinto per ciascun elemento di gruppo. I calcoli per ciascun elemento di gruppo sono simili l'uno all'altro, ad eccezione di:

- Context.ClusterItem: valore della proprietà Context.ClusterItem che è uguale all'elemento di gruppo in corso di calcolo.
- Registros per elemento di gruppo: quando una metrica è registrata a eventi per elemento di gruppo, ciascun elemento di gruppo riceve dati non elaborati ed eventi di riusabilità che vengono inviati a questo elemento di gruppo.

Procedura di calcolo delle metriche di gruppo

L'aspetto più importante da comprendere sul calcolo di una metrica di gruppo è che tutti gli elementi di gruppo vengono calcolati in parallelo. Per parallelo non significa che vengono calcolati con thread diversi, ma che durante l'elaborazione dei vari eventi che devono essere gestiti dai vari elementi di gruppo, gli eventi vengono elaborati in sequenza e per ciascun evento vengono chiamati gli elementi di gruppo pertinenti che elaborano l'evento. Ad esempio, sono presenti molti eventi che devono essere gestiti da numerosi elementi di gruppo. Esistono due modi per tale scopo:

Esempio: Opzione 1

Per ciascun elemento di gruppo C

Per ogni evento E che deve essere gestito da C
Consentire a C di gestire E

Esempio: Opzione 2

Per ogni evento E

Per ciascun elemento di gruppo C che deve gestire E
Consentire a C di gestire E

Il motore gestisce le metriche di gruppo utilizzando l'opzione 2.

Un altro punto importante da comprendere è che l'esecuzione di VBScript in PsIWriter viene effettuata da un componente denominato Script Control. È presente solo una singola istanza di questo componente per ogni metrica e tale istanza viene riutilizzata per il calcolo dei vari elementi di gruppo. Poiché gli elementi di gruppo vengono calcolati in parallelo come descritto in precedenza, e poiché il componente Script Control può contenere solo i dati di un singolo elemento di gruppo in ogni momento, è necessario commutare i dati di frequente all'interno del componente Script Control.

A scopo illustrativo, viene riportato di seguito uno pseudo-codice più dettagliato per l'esecuzione del calcolo.

```
1-   For each metric M
2-       Let X be the earliest event not yet handled by M
3-       Let T be the timestamp of the latest state before X
4-       Let L be the list of all events registered by M (all cluster items)
      starting from timestamp T until the current time
5-       For each event E in L
6-           For each cluster item C that should handle E
7-               Let C' be the cluster item that is currently loaded into
the script control
8-               Take the values of the global variables from the script
control and store them aside for C'
9-               Take the values of the global variables stored aside for C
and load them into the script control
10-            Handle event E
```

L'intero processo di rilevamento talvolta di un evento non ancora preso in considerazione e di esecuzione del calcolo da questo punto in poi è denominato Nuovo calcolo. Il processo di sostituzione dei valori delle variabili globali (i passaggi 8 e 9 nel codice sopra riportato) è denominato Cambio di contesto.

I due principali problemi che possano essere facilmente riscontrati nel codice riportato sopra sono:

- Il nuovo calcolo viene eseguito per tutti gli elementi di gruppo insieme. Poiché il punto nel tempo T (passaggio 3 nel codice sopra riportato) è rilevato una sola volta e quindi tutti gli elementi di gruppo eseguono il nuovo calcolo in base a quel momento. Ciò significa che, quando un singolo elemento di gruppo contiene un nuovo evento, tutti gli elementi di gruppo vengono ricalcolati.
- Il cambio di contesto viene eseguito molto spesso. Questo può essere facilmente riscontrato poiché il cambio di contesto (passaggi 8 e 9 nel codice sopra riportato) avviene nel ciclo interno.

Ricalcolo delle metriche di gruppo

- Descrizione del problema

Come già indicato, tutti gli elementi di gruppo in una metrica di gruppo vengono ricalcolati come un insieme. Questo significa che se si dispone di una metrica di gruppo con 1000 elementi di gruppo e per uno di essi è necessario il ricalcolo dell'ultimo anno a causa di qualche nuovo evento ricevuto, tutti i 1000 elementi di gruppo vengono ricalcolati per l'ultimo anno.

- Soluzioni possibili

Le soluzioni suggerite di seguito possono ridurre le conseguenze del problema, ma non sono sempre applicabili e ognuna presenta i propri svantaggi. L'aspetto più importante è comprendere il problema e il costo stimato e confrontare tale costo con il costo delle soluzioni proposte.

- Quando il numero degli elementi di gruppo è piccolo, è possibile prendere in considerazione la definizione di ognuno di essi come una metrica distinta. Il lato negativo di questo approccio è, ovviamente, il costo di manutenzione per la gestione di diverse metriche, nonché il fatto che non è possibile eseguire un report per l'intera metrica e quindi eseguire il drill-down in un elemento di gruppo specifico
- Quando il numero degli elementi di gruppo è elevato, ma solo uno o un numero ridotto di essi viene ricalcolato di frequente, è possibile considerare di inserire questo elemento di gruppo in una metrica separata e lasciare tutti gli altri elementi di gruppo nell'altra metrica
- Utilizzare di frequente il blocco del calcolo per il contratto o la metrica pertinente, in modo che questa metrica non sia mai sottoposta a ricalcoli lunghi
- Eseguire alcune modifiche negli adapter e nelle origini dati, in modo che non vi siano ricalcoli lunghi (ad esempio, non inviare eventi il cui valore di data/ora è antecedente di un mese)

Cambio di contesto

- Descrizione del problema

Come già indicato, il cambio di contesto viene eseguito nel ciclo interno. In altre parole, per ogni evento che deve essere gestito da ciascun elemento di gruppo. Quando una metrica riceve molti eventi e ogni evento è gestito da molti elementi di gruppo, questa quantità può essere molto elevata. Inoltre, l'operazione di cambio di contesto è relativamente costosa (rispetto alla gestione dell'evento in sé nella business logic) e si presenta un problema.

Il costo dell'operazione di cambio di contesto è proporzionale alle dimensioni dei dati commutati. I dati commutati durante il cambio di contesto sono i valori di tutte le variabili globali nella business logic (denominata anche "stato"). Pertanto, quanto maggiori sono il numero di variabili globali di cui si dispone e le dimensioni di tali variabili globali, tanto più costosa è l'operazione di cambio del contesto.

In particolare, non è consigliabile utilizzare le mappe di business logic nelle metriche di gruppo, soprattutto se le dimensioni delle mappe possono essere grandi.

- Soluzioni possibili

- Ridurre il tempo di ciascun cambio di contesto

L'idea è di ridurre le dimensioni dello stato (variabili globali). È possibile eseguire questo approccio con la riscrittura della business logic, in modo che non contenga mappe. Ovviamente non è sempre possibile, ma qualora lo fosse, è consigliabile.

- Ridurre la quantità di cambi di contesto

Se il gruppo è piccolo, è possibile creare una metrica separata per ogni elemento di gruppo.

Evitare le metriche di gruppo con molti elementi di gruppo che registrano gli stessi eventi. L'idea in questo caso è la seguente:

Se ciascun evento è gestito da un unico elemento di gruppo, la quantità di cambi di contesto è proporzionale al numero di eventi

Se ciascun evento è gestito da tutti gli elementi di gruppo, la quantità di cambi di contesto è proporzionale al numero di eventi per il numero di elementi di gruppo

Creare una metrica non di gruppo in grado di calcolare i risultati per tutti gli elementi di gruppo originari (che a questo punto sono semplici risorse e non elementi di gruppo). Fare in modo che la metrica invii il risultato di ogni elemento di gruppo come un evento. Creare un'altra metrica che sia di gruppo e che riceva gli eventi dalla prima metrica e generi un report sul valore ricevuto in tali eventi come risultato. L'idea in questo caso è che le grandi quantità di eventi di dati non elaborati verranno gestite da una metrica non di gruppo e che la metrica di gruppo gestirà un evento singolo per periodo per elemento di gruppo.

Case study 16: modelli di progettazione di business logic

Sono disponibili diversi modelli di progettazione, utilizzabili in molte business logic. I modelli di progettazione sono testati e il loro impiego, quando applicabile, consente di guadagnare molto tempo e nella maggior parte dei casi di creare una business logic più efficace. Questo case study esamina in particolare un modello di progettazione.

Modello di progettazione dei contatori di aggiornamento

Questo modello di progettazione è utile quasi in ogni business logic progettata per misurare il tempo tra determinati eventi. Esempi di tale business logic sono business logic per la misurazione di disponibilità, tempi di inattività, tempo medio tra due errori, tempo medio di ripristino, tempo medio di risposta, tempo medio di risoluzione, percentuale di componenti con disponibilità inferiore a X, numero di casi non risolti in tempo, ecc.

La parte in comune tra tutte queste business logic è che il rispettivo risultato dipende dal valore di data/ora di vari eventi ricevuti.

Una regola pratica per stabilire se la business logic possa trarre vantaggio dal modello di progettazione sarebbe: se la business logic dipende dal valore di data/ora dei diversi eventi ricevuti, molto probabilmente è necessario utilizzare questo modello di progettazione.

Struttura di questo modello di progettazione

È possibile suddividere il codice di business logic che utilizza questo modello in due parti: un framework e un'implementazione. Il framework contiene il codice che nella maggior parte dei casi è fisso e non cambia per le diverse business logic. Questa parte è la stessa per il calcolo della disponibilità e del numero di ticket non risolti in tempo. L'implementazione contiene il codice specifico per ciascuna business logic.

Si consiglia di impostare queste due parti del codice in moduli di business logic distinti e di riutilizzare il modulo del framework in metriche diverse.

Di seguito viene riportato il codice del framework:

```
Dim g_PrevEventTimestamp

Sub OnLoad(time)
    g_PrevEventTimestamp = time
    InitializeStatusVariables
End Sub

Sub OnRegistration(Dispatcher)
    ' Se è presente una copia separata di variabili di stato
    ' per ogni risorsa registrata a seconda delle risorse
    ' registrate è necessario impostare i valori iniziali sulle
    ' variabili di stato delle risorse appena aggiunte qui
End Sub
```

```

Sub OnPeriodStart(time)
    InitializeCounters
End Sub

Sub OnPeriodEnd(time, completePeriod)
    HandleEvent (time)
End Sub

Sub OnTimeslotEnter(time)
    HandleEvent (time)
End Sub

Sub OnTimeslotExit(time)
    HandleEvent (time)
End Sub

Function Result()
    Result = CalculateResult()
End Function

Sub HandleEvent(Time)
    Dim diff
    diff = TimeDiff( "s",Time,g_PrevEventTimestamp)
    UpdateCounters(diff)
    g_PrevEventTimestamp = Time
End Sub

```

Di seguito è riportata la struttura di un modulo di implementazione:

```

' Definire le variabili di stato qui. Può trattarsi di una
' variabile globale semplice o di tante variabili globali complesse
' a seconda della business logic
Dim g_StatusVar_1, g_StatusVar_2, ... ,g_StatusVar_n

```

```

' Definire i contatori qui.
' Può trattarsi di una variabile globale semplice o di tante
' variabili globali complesse a seconda della business logic
Dim g_Counter_1, g_Counter_2, ... , g_Counter_n

```

```

Sub InitializeStatusVariables ()
    ' Impostare i valori iniziali per le diverse variabili di stato
End Sub

```

```

Sub InitializeCounters ()
    ' Impostare i valori iniziali per i vari contatori
    g_Counter_1 = 0

```

```
        g_Counter_2 = 0
        '...
        g_Counter_n = 0
    End Sub

    Function CalculateResult ()
        ' Calcolare il risultato. Il risultato deve dipendere
        ' dai valori dei contatori. Non deve dipendere
        ' dal valore delle variabili di stato. Non deve
        ' modificare i valori dei contatori o le variabili
        ' di stato
    End Function

    Sub UpdateStatus(method, eventDetails)
        ' Aggiornare il valore delle variabili di stato in base
        ' ai parametri (e possibilmente al valore precedente delle
        ' variabili di stato)
    End Sub

    Sub UpdateCounters(diff)
        ' Aggiornare i valori dei contatori in base al loro
        ' valore precedente, al valore delle variabili di stato
        ' e al valore del parametro diff.
        ' In molti casi questo calcolo si basa anche sul
        ' valore di Context.IsWithinTimeslot
    End Sub

    Sub OnEvent_1(eventDetails)
        HandleEvent (eventDetails.Time)
        UpdateStatus("OnEvent_1",eventDetails)
    End Sub

    Sub OnEvent_2(eventDetails)
        HandleEvent (eventDetails.Time)
        UpdateStatus("OnEvent_2",eventDetails)
    End Sub

    '...

    Sub OnEvent_n(eventDetails)
        HandleEvent (eventDetails.Time)
        UpdateStatus("OnEvent_n",eventDetails)
    End Sub
```


Per spiegare meglio questo modello di progettazione, di seguito viene fornito un esempio dell'implementazione del modello per il calcolo della disponibilità. Questo esempio presuppone la presenza di eventi per stati UP (Attivo) e DOWN (Non attivo) gestiti da due diversi gestori di eventi nella business logic. La disponibilità è definita come la percentuale di tempo durante il quale il sistema è attivo rispetto al tempo totale nel periodo di applicazione. Si presuppone che lo stato del sistema sia lo stato dell'ultimo evento ricevuto (UP (Attivo) oppure DOWN (Non attivo)).

Di seguito viene riportato il codice dell'implementazione (il codice del framework non è stato modificato):

```
' Variabile di stato
Dim g_Status
' Contatori.
Dim g_UpTime, g_TotalTime

Sub InitializeStatusVariables ()
    G_Status = "UP"
End Sub

Sub InitializeCounters ()
    g_UpTime = 0
    g_TotalTime = 0
End Sub

Function CalculateResult ()
    If g_TotalTime = 0 Then
        CalculateResult = Null
    Else
        CalculateResult = g_UpTime/g_TotalTime*100
    End If
End Function

Sub UpdateStatus(method, eventDetails)
    If method = "OnUP" Then
        G_Status = "UP"
    Else
        G_Status = "DOWN"
    End If
End Sub

Sub UpdateCounters(diff)
    If Context.IsWithinTimeslot Then
        G_TotalTime = g_TotalTime + diff
        If g_Status = "UP" Then
            G_UpTime = g_UpTime + diff
        End If
    End If
End Sub
```

```
Sub OnUp(eventDetails)
    HandleEvent (eventDetails.Time)
    UpdateStatus("OnUp",eventDetails)
End Sub

Sub OnDown(eventDetails)
    HandleEvent (eventDetails.Time)
    UpdateStatus("OnDown",eventDetails)
End Sub
```

Esistono diverse varianti di questo modello. Una delle varianti più comuni si ha quando un contatore di tempo separato deve essere sottoposto a manutenzione per diverse entità. Ad esempio, durante la misurazione del tempo di soluzione, un contatore separato deve essere sottoposto a manutenzione per ogni ticket aperto. In questo caso, durante la gestione di un evento pertinente solo a un ticket, è più efficiente aggiornare solo il contatore di tale ticket. Quando viene gestito un evento comune (ad esempio OnPeriodEnd o OnTimeslotEnter), è necessario aggiornare i contatori di tutti i ticket.

Nota: la variante del modello richiede la conservazione di una copia separata della variabile globale g_PrevEventTimestamp per ciascun ticket.

È possibile riscontrare alcuni esempi ottimali di utilizzo di questo modello nel contenuto predefinito. Tenere presente, tuttavia, che nel contenuto predefinito questo modello è utilizzato in modo un po' diverso e che la distinzione tra framework e implementazione non è così ovvia.

Case study 17: funzionalità integrata

ACE dispone di una funzionalità integrata e di strumenti per diversi scopi. L'utilizzo della funzionalità integrata è preferibile per la scrittura in VBS. Poiché VBS è un linguaggio interpretato, la riproduzione in VBS compromette le prestazioni.

Di seguito viene riportato un elenco delle funzioni integrate e delle corrette modalità di utilizzo:

IsWithinTimeslot

Questa è la funzione integrata più semplice. Il suo scopo è consentire alla business logic di indicare se il sistema è attualmente all'interno di un periodo di applicazione oppure no. In questo modo viene rimossa la necessità di gestire una variabile nelle funzioni di inizio e di termine del periodo di applicazione per eseguire la stessa operazione. Ad esempio, invece di eseguire il seguente codice:

```
Dim amIWithinATimeslot
Sub OnTimeslotEnter(time)
    amIWithinATimeslot = 1
End sub
Sub OnTimeslotExit(time)
    amIWithinATimeslot = 0
End sub
Sub OnEvent(eventDetails)
    If amIWithinATimeslot = 1 Then
        count = count + 1
    End if
End sub
```

È possibile eseguire questo codice molto più semplice:

```
Sub OnEvent(eventDetails)
    If context.IsWithinTimeslot Then
        count = count + 1
    End if
End sub
```

Se si desidera utilizzare o mantenere le informazioni sulla data/ora di inizio e di termine del periodo di applicazione, questa funzionalità non è in grado di soddisfare tale esigenza. Ma di norma non è necessaria e il codice è sufficiente.

TimeOfLastEvent

Questa funzione restituisce la data/ora dell'ultimo evento di dati non elaborati o di dati intermedi che è stato gestito. Ciò significa che non è necessario salvare queste informazioni nel gestore eventi, in quanto sono disponibili direttamente tramite questa funzione. Ad esempio:

```
Function result
    Dim LastEventTimestamp
    LastEventTimestamp = Context.TimeOfLastEvent
```

End function

TimeOfLastEventHandler

Questa funzione restituisce la data/ora dell'ultimo gestore evento chiamato dal motore di aggregazione e correlazione. Questo include non solo i gestori di eventi di dati non elaborati e di dati intermedi, ma anche tutti gli eventi di sistema chiamati. È particolarmente utile nei gestori eventi che non ricevono l'ora, ad esempio, per la funzione Result. Ad esempio:

```
Function result
    Dim LastEventHandlerTimestamp
    LastEventHandlerTimestamp= Context.TimeOfLastEventHandler
End function
```

NetTime

Questa funzione consente di specificare due valori di data/ora e di ricevere il tempo netto (in secondi) con cui il sistema è all'interno del periodo di applicazione per la regola corrente, compreso tra i due valori di data/ora. Questa in particolare è una funzionalità poco pratica e non deve essere implementata nel VBS. L'implementazione nel VBS comporterebbe la conservazione di un elenco ogni inizio e termine del periodo di applicazione o il calcolo della differenza tra ogni ora in cui viene immesso il termine del periodo di applicazione direttamente, in modo da calcolare l'intervallo di tempo tra di essi. In condizioni estreme, potrebbe verificarsi un gran numero di volte e questo non favorirebbe le prestazioni di calcolo. La funzione interna esegue la stessa operazione dopo una significativa ottimizzazione, con una maggiore efficienza. Ad esempio:

```
Function result
    Dim MyNetTime
    MyNetTime = Tools.NetTime(MyBeginTimestamp, MyEndTimestamp)
End function
```

Oggetto di contesto

L'oggetto di contesto comprende una varietà di parametri che forniscono informazioni su:

- Metrica corrente.
- Contratto contenuto.
- Stato di calcolo corrente.
- Dominio del servizio, categoria di dominio e relativi valori (ad esempio, soglia).
- Informazioni di raggruppamento: il gruppo in generale e l'elemento di gruppo specifico in gestione.
- Funzioni che restituiscono elenchi delle risorse in base ai requisiti utente.
- Funzioni che consentono di convertire i nomi di risorsa in ID risorsa e viceversa.

L'accesso a queste informazioni direttamente dal database tramite Safe ODBC è estremamente inefficiente e non ha senso poiché le informazioni sono subito disponibili dall'oggetto di contesto. Se possibile, utilizzare sempre la funzionalità integrata per ottenere informazioni.

Case study 18: registrazione

La business logic è spesso scritta in modo da conservare una mappa della struttura della risorsa di metrica per l'uso durante i calcoli. Poiché la struttura della risorsa cambia nel tempo, la business logic deve aggiornare la struttura nella mappa quando la struttura della risorsa cambia.

Il metodo OnRegistration viene chiamato quando la struttura della risorsa subisce modifiche, poiché è responsabile della gestione del comportamento del motore associato alle modifiche nelle registrazioni e nel raggruppamento dovute a modifiche alla struttura della risorsa. Il fatto che questo metodo viene chiamato per ogni modifica della struttura della risorsa lo rende un passaggio comodo per aggiornare la mappa precedentemente citata. Tuttavia, la compilazione della mappa non è pertinente al processo di registrazione. Questo significa che la compilazione della mappa riduce le prestazioni della funzione OnRegistration. Non è importante durante il runtime, perché non si verifica di frequente. Tuttavia, il metodo OnRegistration è invocato anche nel processo di elaborazione dell'infrastruttura da parte del motore, durante il quale il sistema calcola se le modifiche alla struttura della risorsa sono pertinenti alla registrazione di ciascuna metrica specifica di cui l'istanza è responsabile. Durante questo processo, il metodo OnRegistration viene invocato per ogni modifica nella struttura della risorsa, anche se la modifica alla struttura non è pertinente alla metrica corrente. Ciò significa che il metodo può essere invocato varie volte per metrica.

Se questa logica viene implementata nel metodo OnRegistration, una lieve riduzione delle prestazioni in fase di runtime potrebbe diventare una riduzione delle prestazioni molto significativa durante l'elaborazione dell'infrastruttura.

Per risolvere questo problema, la compilazione delle mappe o di qualsiasi altro processo di inizializzazione che deve essere eseguito quando si verifica una modifica nella struttura della risorsa, ma non pertinente alla registrazione, può essere effettuata in due modi:

Utilizzando la proprietà IsRunTimeMode dell'oggetto dispatcher.

Questa proprietà consente all'utente di determinare se l'esecuzione corrente è un calcolo eseguito o meno, nonché di racchiudere la logica non pertinente alla registrazione in un'istruzione if che garantisce l'esecuzione solo durante il runtime.

Nell'esempio riportato di seguito, la parte contrassegnata in blu è la parte della business logic pertinente alla registrazione e deve essere sempre eseguita. La parte contrassegnata in verde non è pertinente alla registrazione e può essere racchiusa nella nuova istruzione if.

```
Sub OnRegistration (dispatcher)
    Dim MyResource
    MyResource = context.clusteritem
    Dispatcher.RegisterByResource "OnEvent", "My Event Type", MyResource
    Dim ThisResourceMap
    Set GlobalResourceVector= CreateObject("SlalomVector.Vector")
    Dim resource
    Set ThisResourceMap = Context.ResourcesOfResourceGroup(Context.ClusterItem)
    Per ogni risorsa in ThisResourceMap
        Risorsa GlobalResourceVector.Add
    Avanti
End Sub
```

È possibile migliorare questo codice modificandolo nel modo seguente:

```
Sub OnRegistration (dispatcher)
    Dim MyResource
    MyResource = context.clusteritem
    Dispatcher.RegisterByResource "OnEvent", "My Event Type", MyResource
    If Dispatcher.IsRunTimeMode Then
        Dim ThisResourceMap
        Set GlobalResourceVector= CreateObject("SlalomVector.Vector")
        Dim resource
        ThisResourceMap = Context.ResourcesOfResourceGroup(Context.ClusterItem)
        Per ogni risorsa in ThisResourceMap
            Risorsa GlobalResourceVector.Add
        Avanti
    End If
End Sub
```

Utilizzo del metodo OnResourceStructureChanged.

Questo metodo viene eseguito subito dopo il metodo OnRegistration e fornisce la stessa funzionalità del metodo originale, ma viene eseguito solo durante il runtime. Questo metodo non è invocato durante l'elaborazione dell'infrastruttura e, pertanto, le prestazioni non sono compromesse.

Nell'esempio riportato di seguito, la parte contrassegnata in blu è la parte della business logic pertinente alla registrazione e deve restare nel metodo OnRegistration. La parte contrassegnata in verde non è pertinente alla registrazione e può essere inserita nella nuova funzione.

```
Sub OnRegistration (dispatcher)
    Dim MyResource
    MyResource = context.clusteritem
    Dispatcher.RegisterByResource "OnEvent", "My Event Type", MyResource
    Dim ThisResourceMap
    Set GlobalResourceVector= CreateObject("SlalomVector.Vector")
    Dim resource
    Set ThisResourceMap = Context.ResourcesOfResourceGroup(Context.ClusterItem)
    Per ogni risorsa in ThisResourceMap
        Risorsa GlobalResourceVector.Add
    Avanti
End Sub
```

È possibile migliorare questo codice modificandolo nel modo seguente:

```
Sub OnRegistration (dispatcher)
    Dim MyResource
    MyResource = context.clusteritem
    Dispatcher.RegisterByResource "OnEvent", "My Event Type", MyResource
End Sub

Sub OnResourceStructureChanged(time)
    Dim ThisResourceMap
    Set GlobalResourceVector= CreateObject("SlalomVector.Vector")
    Dim resource
    Set ThisResourceMap = Context.ResourcesOfResourceGroup(Context.ClusterItem)
    Per ogni risorsa in ThisResourceMap
        Risorsa GlobalResourceVector.Add
    Avanti
End Sub
```

Case study 19: procedura guidata di configurazione dell'adapter per origine dati basata su file

In questo case study sono esaminati i metodi ottimali per l'integrazione di un'origine dati basata su file. Lo scenario di esempio gestisce un file di dati CSV generato dal sistema di origine. Con la maggior parte delle integrazioni basate su file CA consiglia di seguire alcune linee guida di base, per assicurare il minimo rischio per l'integrazione. Sono indicate di seguito:

- Se è disponibile l'opzione, è necessario richiedere che i dati siano recapitati al file system del server applicazioni CA Business Service Insight. In tal modo il meccanismo di recapito non dipende dall'adapter che cerca di recuperare i dati da un archivio remoto (riduce al minimo i problemi di autorizzazione con gli account utente, i problemi di sincronizzazione, ecc.)
- La convenzione di denominazione dei file è fondamentale in quanto l'adapter ordina i file in base alla denominazione alfanumerica. Se è possibile controllare questa opzione, CA consiglia di richiedere due parti:
 - Una convenzione di denominazione sensibile basata sul contenuto del file di origine (specialmente se più file derivano dall'origine)
 - Un indicatore di data/ora con ordine inverso per garantire che i file siano ordinati a partire dall'ultimo file più recente. (ad esempio, `<file_name>_yyyymmdd_hhmiss.csv`). L'estensione del valore di data/ora utilizzato dipende dalla frequenza dei dati forniti

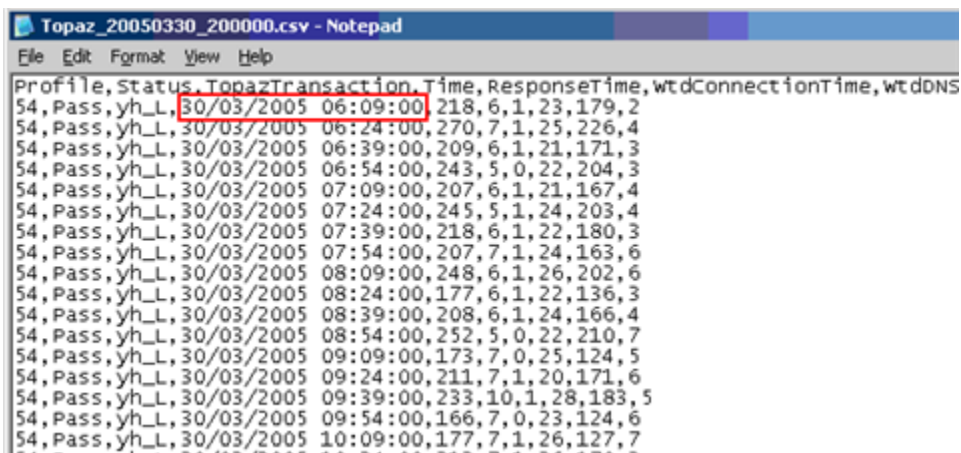
In questo scenario, i file di origine derivano da un'origine dati Topaz (oggi nota come Mercury Global Monitor, posseduta da HP). È stata creata mediante un'API del prodotto per includere i file necessari per i KPI richiesti. I file vengono recapitati al server applicazioni CA Business Service Insight direttamente da un processo automatico esterno. I file di origine vengono denominati: *Topaz_yyyymmdd_hhmiss.csv*.

Nota: la data/ora del file corrisponde all'ora di creazione, pertanto tutte le voci all'interno del file conducono fino a questo momento.

Di seguito è possibile consultare un esempio di dati all'interno del file.

	A	B	C	D	E	F	G	H	I	J	K
1	Profile	Status	TopazTransaction	Time	ResponseTime	WtdConnectionTime	WtdDNSTime	WtdSSLTime	WtdNetworkTime	WtdDownloadTime	
2	54	Pass	yh_L	30/03/2005 06:09	218	6	1	23	179	2	
3	54	Pass	yh_L	30/03/2005 06:24	270	7	1	25	226	4	
4	54	Pass	yh_L	30/03/2005 06:39	209	6	1	21	171	3	
5	54	Pass	yh_L	30/03/2005 06:54	243	5	0	22	204	3	
6	54	Pass	yh_L	30/03/2005 07:09	207	6	1	21	167	4	
7	54	Pass	yh_L	30/03/2005 07:24	245	5	1	24	203	4	
8	54	Pass	yh_L	30/03/2005 07:39	218	6	1	22	180	3	
9	54	Pass	yh_L	30/03/2005 07:54	207	7	1	24	163	6	
10	54	Pass	yh_L	30/03/2005 08:09	248	6	1	26	202	6	
11	54	Pass	yh_L	30/03/2005 08:24	177	6	1	22	136	3	
12	54	Pass	yh_L	30/03/2005 08:39	208	6	1	24	166	4	
13	54	Pass	yh_L	30/03/2005 08:54	252	5	0	22	210	7	
14	54	Pass	yh_L	30/03/2005 09:09	173	7	0	25	124	5	
15	54	Pass	yh_L	30/03/2005 09:24	211	7	1	20	171	6	

Nota: CA consiglia di controllare i file CSV nel Blocco note (invece che semplicemente in Excel) per assicurare che il formato data sia come previsto. Excel tende a formattare le date in base alle impostazioni internazionali del computer e potrebbe non corrispondere al formato effettivo visualizzato dall'adapter.



```

Topaz_20050330_200000.csv - Notepad
File Edit Format View Help
Profile,Status,TopazTransaction,Time,ResponseTime,wtdConnectionTime,wtdDNS
54,Pass,yh_L,30/03/2005 06:09:00,218,6,1,23,179,2
54,Pass,yh_L,30/03/2005 06:24:00,270,7,1,25,226,4
54,Pass,yh_L,30/03/2005 06:39:00,209,6,1,21,171,3
54,Pass,yh_L,30/03/2005 06:54:00,243,5,0,22,204,3
54,Pass,yh_L,30/03/2005 07:09:00,207,6,1,21,167,4
54,Pass,yh_L,30/03/2005 07:24:00,245,5,1,24,203,4
54,Pass,yh_L,30/03/2005 07:39:00,218,6,1,22,180,3
54,Pass,yh_L,30/03/2005 07:54:00,207,7,1,24,163,6
54,Pass,yh_L,30/03/2005 08:09:00,248,6,1,26,202,6
54,Pass,yh_L,30/03/2005 08:24:00,177,6,1,22,136,3
54,Pass,yh_L,30/03/2005 08:39:00,208,6,1,24,166,4
54,Pass,yh_L,30/03/2005 08:54:00,252,5,0,22,210,7
54,Pass,yh_L,30/03/2005 09:09:00,173,7,0,25,124,5
54,Pass,yh_L,30/03/2005 09:24:00,211,7,1,20,171,6
54,Pass,yh_L,30/03/2005 09:39:00,233,10,1,28,183,5
54,Pass,yh_L,30/03/2005 09:54:00,166,7,0,23,124,6
54,Pass,yh_L,30/03/2005 10:09:00,177,7,1,26,127,7

```

Prima di avviare un processo di creazione dell'adapter, è di cruciale importanza aver eseguito tutte le analisi e le verifiche necessarie nell'origine dati e i KPI correlati per assicurare che siano noti i seguenti punti:

- Quali campi sono obbligatori per la business logic
- Qual è il formato delle date nel file
- Qual è il fuso orario di valori di data/ora nel file di origine (è necessario creare i fusi orari nel sistema prima di avviare il processo di creazione dell'adapter)
- Se esistono campi data con voci vuote o NULL
- Qual è il comportamento dell'origine dati (vengono aggiunti tutti i record, alcuni record sono un aggiornamento di un evento precedente)
- Quali drill-down sono richiesti per i KPI (questa opzione potrebbe influire sulla selezione della risorsa)
- Quali tipi di evento sono obbligatori per il filtraggio effettivo di eventi nella business logic

Quando questi punti sono noti, è possibile iniziare il processo di creazione.

A questo punto, è possibile eseguire il processo di creazione dell'adapter in base a questo scenario, tramite la procedura guidata di configurazione dell'adapter.

In questo scenario è possibile utilizzare TopazTransaction come risorsa, Profile come tipo di evento e il campo Time come data/ora. Inoltre, vengono inseriti i campi Status, ResponseTime e WtdDownloadTime nella struttura dell'evento per l'utilizzo con la business logic.

Creazione dell'adapter

Innanzitutto, assicurarsi che il sistema sia pronto per la creazione del nuovo adapter e la corretta distribuzione sul server verificando che i servizi seguenti siano avviati sul server applicazioni.

- Servizio di distribuzione dell'adapter
- Servizio del listener dell'adapter

Successivamente, accedere alla pagina Adapter e creare un nuovo adapter. Selezionare l'opzione Aggiungi Nuovo > Adapter per file di testo dalla pagina Adapter.

Passaggio Generale

Nel passaggio Generale della Procedura guidata di configurazione dell'adapter, compilare i seguenti campi:

- *Nome*: immettere un nome adatto per l'adapter
- *Indirizzo adapter*: LOCALHOST è l'opzione predefinita (per la distribuzione del server applicazioni), ma è possibile inserire altri indirizzi utilizzando il pulsante accanto se necessario.
- *Formato ora*: è il formato ora predefinito utilizzato per i campi di data/ora all'interno dell'origine dati. La corretta selezione assicura che i campi siano rilevati automaticamente in seguito nella procedura guidata di configurazione dell'adapter. È possibile inserire nuovi formati ora in questa fase se necessario, utilizzando il pulsante accanto al campo.
- *Fuso orario*: è il fuso orario in cui i record dei dati vengono registrati. È richiesto in modo che l'adapter possa spostare correttamente il campo event_timestamp (and altri campi di data/ora) indietro all'ora UTC per il corretto time-shift interno. È necessario che questo campo sia stato compilato in precedenza come da lista di controllo dell'origine dati nella sezione precedente.

Note:

In questa pagina è presente inoltre un pulsante Avanzate che consente di selezionare alcuni parametri di configurazione per l'adapter. È possibile lasciare la maggior parte dei parametri ai valori predefiniti, salvo modifiche necessarie.

La porta dell'adapter viene assegnata automaticamente all'adapter come la porta di CA Business Service Insight in seguito, ma è possibile sostituirla in questa fase, se necessario. Altri parametri importanti che si possono modificare includono: impostazioni internazionali, modalità in linea/non in linea, dettagli di connessione, opzioni di monitoraggio e registrazione, impostazioni di esecuzione una volta/sempre, limiti di errore, nomi dei file e commenti.



L'esame di ciascuna delle impostazioni citate esula dall'ambito di questo case study, ma è possibile trovarlo nella sezione [Specifiche di configurazione dell'adapter](#) (a pagina 317).

Fare clic su Avanti per procedere al passaggio successivo della procedura guidata di configurazione dell'adapter. Il passaggio successivo consente l'accesso all'Interfaccia origine dati dell'adapter.

Passaggio Interfaccia origine dati

Nel passaggio Interfaccia origine dati della Procedura guidata di configurazione dell'adapter, compilare i seguenti campi:

- *Nome origine dati*: nome del file di origine specifico (è possibile disporre di più file di origine in un adapter)
- *Percorso file*: percorso sul server applicazioni (o altro server) che contiene i file dell'origine dati. Per un server diverso dal server applicazioni, utilizzare una notazione UNC (ad esempio, //server01/sourcefolder)
- *Modello nome*: utilizzare un carattere jolly per filtrare i file che si trovano nel percorso file caricati dall'adapter.

La scheda Definizione analisi consente inoltre di definire la struttura del file in corso di importazione. È possibile utilizzare i campi come indicato di seguito:

- *Titolo*: casella di controllo con valori booleani per specificare se è presente o no una riga di titolo nel file di dati (ad esempio, la prima riga del file CSV è un nome di titolo, seguito dai valori delle righe successive)
- *Delimitatori*: specificare il delimitatore nel file che separa i singoli campi.

Nota: sono presenti anche altri due pulsanti Avanzate che forniscono opzioni di configurazione aggiuntive. Un pulsante si trova nella scheda Dettagli e l'altro si trova nella scheda Definizione analisi. Il pulsante Avanzate nella scheda Dettagli consente l'accesso alle seguenti opzioni:

Origine dati attiva: casella di controllo booleana che consente di attivare/disattivare questa particolare sezione dell'origine dell'adapter

Dopo elaborazione: consente di specificare se il file viene eliminato o conservato dopo l'elaborazione

Nome file iniziale: consente di impostare il nome di file iniziale da cui avviare l'elaborazione (nel caso in cui non si desiderano caricare tutti i file in una directory specifica)

Verifica nuovi dati ogni: definisce l'intervallo di tempo compreso tra la verifica per la presenza di nuovi file quando l'adapter è impostato per essere eseguito continuamente

Il pulsante Avanzate nella scheda Definizione analisi consente di definire le opzioni di chiusura del record quali record multilinea, ecc.

Una volta impostati i dettagli e le definizioni di analisi, è possibile caricare un esempio di file di origine nella procedura guidata per verificare le opzioni di configurazione e visualizzare in anteprima i contenuti del file.

Facendo clic sul pulsante Sfoglia è possibile caricare un file di esempio nella finestra di anteprima riportata di seguito. Individuare il file di esempio, quindi premere il pulsante Verifica file. Questo passaggio è facoltativo.

Nota: la funzione di esplorazione viene visualizzata dal computer locale sul quale si sta lavorando. Se non è il server applicazioni, è necessario disporre di una copia dei file di origine presenti sul server per eseguire questa operazione. Non è possibile esplorare direttamente il server utilizzando questa funzione.

Una volta caricato il file, i contenuti dovrebbero essere visualizzati nella finestra di anteprima come illustrato di seguito.

Nota: il nome Campo viene visualizzato nell'intestazione, insieme al tipo di campo rilevato (numero intero, stringa, ora, ecc.). È necessario verificare che siano stati rilevati correttamente e secondo le proprie esigenze prima di procedere. Verificare assolutamente quanto segue:

- L'indicatore di data/ora viene rilevato come "ora"; questa operazione è possibile in base al formato ora specificato nel primo passaggio della procedura guidata
- La risorsa rilevata è di tipo stringa

Se si è soddisfatti dell'anteprima del file, fare clic su Avanti. Viene visualizzato il passaggio Mapping.

Passaggio Mapping

Il passaggio successivo (e ultimo) della procedura guidata di configurazione dell'adapter esegue le attività di conversione e consente di mappare i campi di input dall'origine dati ai campi di output, che costituiscono l'evento CA Business Service Insight. Esistono due opzioni per continuare in questo passaggio, a seconda che il tipo di evento sia stato già creato nel sistema oppure no. Vi è anche una serie di altre opzioni che è possibile selezionare per la configurazione, verificando che siano rispettati gli standard di denominazione. La maggior parte di queste opzioni è facoltativa, ma richieste per semplificare il processo e ridurre i passaggi. I passaggi obbligatori sono elencati di seguito.

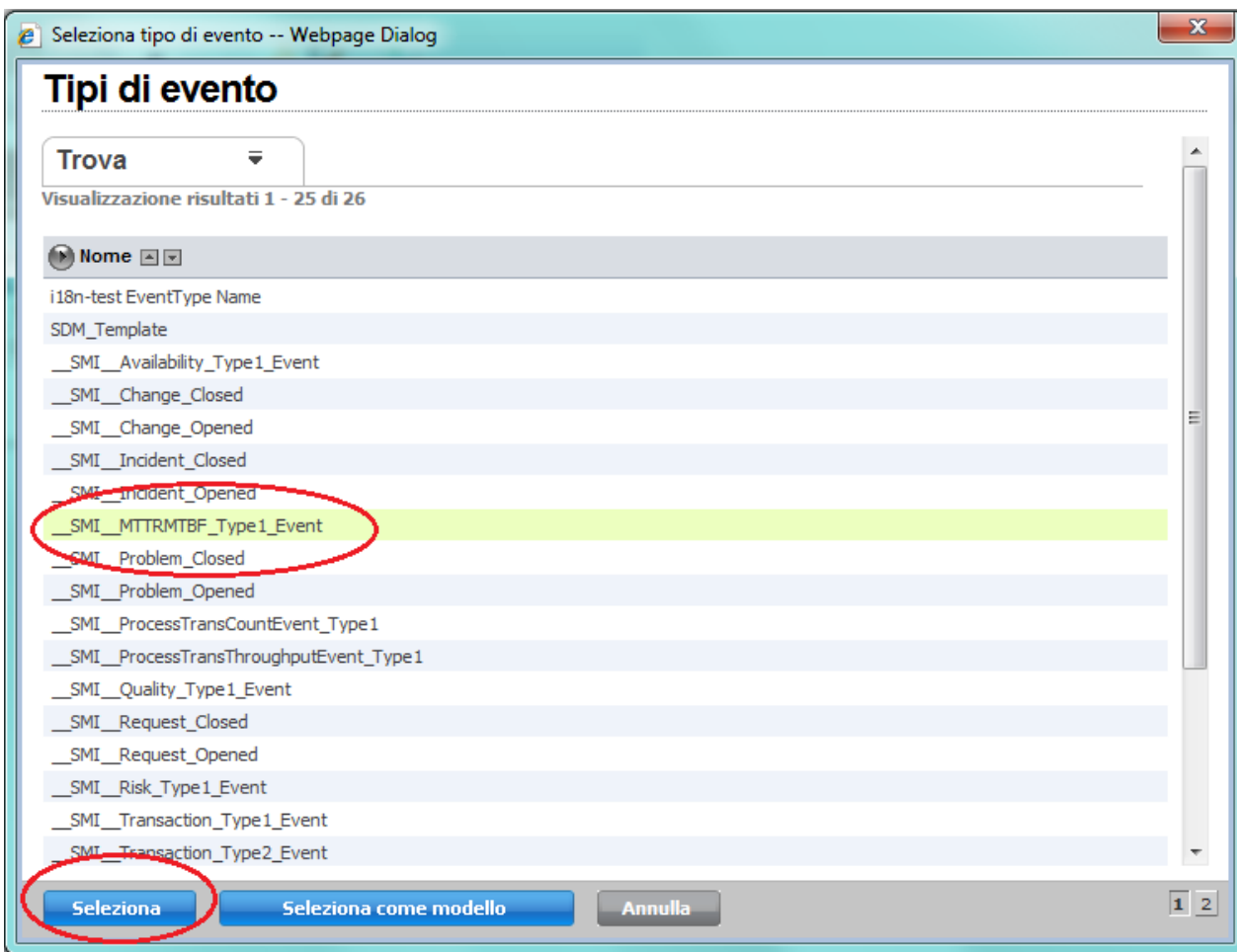
Le fasi da configurare nel passaggio Mapping sono indicate di seguito (incluse le fasi facoltative):

1. Immettere un nome per il campo Formato di input (utile per gli adapter con più input)
2. Aggiungere eventuali altri campi richiesti (ad esempio, Valore costante, Origine dati, Titolo, Nome file o tipi di campo Composto)
3. Creare i criteri Convalida input richiesti
4. Selezionare Tipo di evento se esistente o Crea nuovo tipo di evento. (obbligatorio)
5. Eseguire il mapping dei campi da Input su Output (obbligatorio)
6. Immettere un nome per il campo Output Format (Formato di output)
7. Eseguire il mapping per ResourceId, Data/ora e Tipo di evento
8. Creare i criteri Convalida output richiesti
9. Specificare l'impostazione OnDuplication per gli eventi

Quando si desidera creare un nuovo tipo di evento, viene visualizzata una nuova finestra popup compilata automaticamente in base al formato di input nella schermata principale. È comunque necessario immettere il nome per Tipo di evento e assegnare un Tipo di risorsa a Tipo di evento.

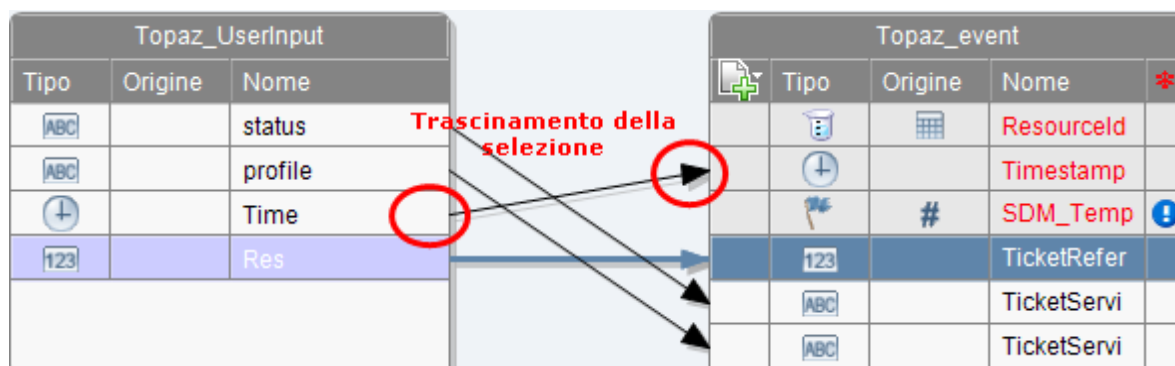
Una volta completato, è possibile salvare e chiudere e il mapping viene terminato automaticamente.

Se viene selezionata l'opzione Seleziona tipo di evento, viene visualizzato un elenco dei tipi di evento esistenti nel sistema fra cui scegliere. Tuttavia, nel corso dell'operazione, solo i campi da Input con il nome e il tipo di dati corrispondenti dal campo Tipo di evento sono collegati automaticamente. Per i campi restanti è necessario eseguire il mapping manuale.

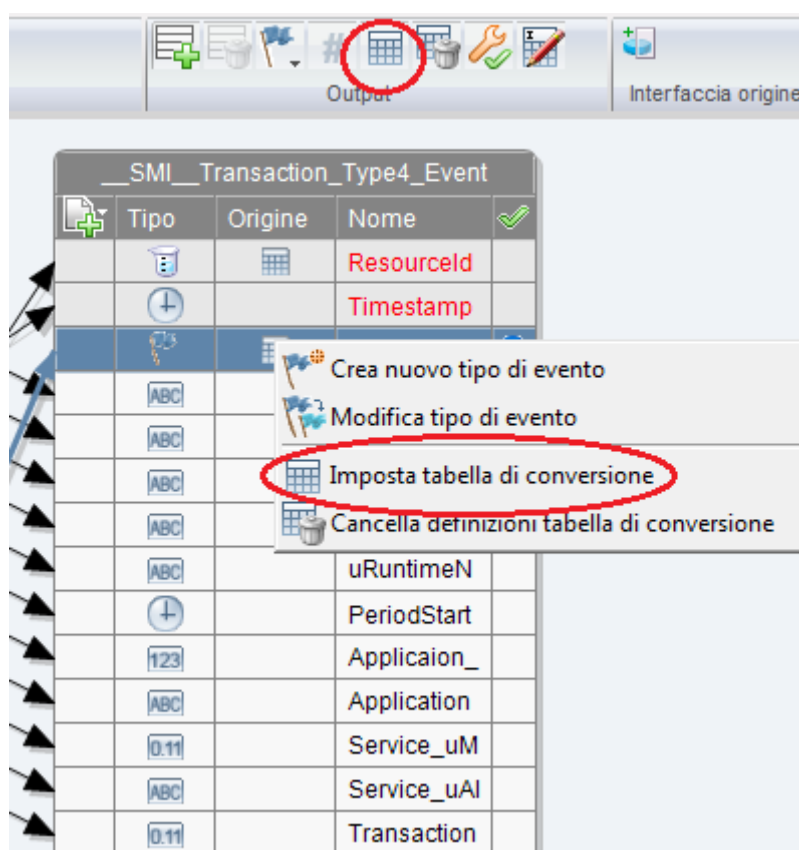


Il passaggio successivo richiede la configurazione dei campi ResourceId, Data/ora e Tipo di evento. Se i campi esistono già (se creati nei passaggi precedenti), è possibile collegarli a questi campi come richiesto.

L'interfaccia supporta il tipo di associazione con selezione e trascinamento. Se i campi associati non esistono dopo l'impostazione, assicurarsi che il tipo di ognuno corrisponda. (ad esempio, ora/stringa/numero intero, ecc.).



- È necessario mappare il campo ResourceId secondo i requisiti, identificati durante l'analisi, da un valore Stringa nel campo Input
- È necessario impostare il campo Data/ora sulla variabile Ora che definisce l'ora in cui si verifica l'evento e viene utilizzata ai fini del calcolo
- I valori predefiniti del campo Tipo di evento sono impostati su un valore costante in base al campo Nome origine dati della schermata precedente. È possibile sostituirlo se richiesto. Può essere necessario se più eventi devono essere inviati da questo adapter, in base a un determinato valore del campo (ad esempio, si desidera inviare un evento diverso in base al valore di un campo Input). È possibile eseguire questa operazione facendo clic con il pulsante destro del mouse sulla riga Tipo di evento (o facendo clic sul pulsante sopra come indicato nell'immagine riportata di seguito) e selezionando Imposta tabella di conversione



In seguito, viene visualizzata una finestra popup.

Dettagli tabella di conversione

Nome: Eventtype Id

Tipo di origine: Tabella

Tipo: Numero intero

Nome tabella: * SMI_EVENTS

Tipo di destinazione: Tipo di evento

Origine di conversione: Nome campo event_name Valore campo EventName

Aggiungi nuova tabella di conversione

OK Annulla

Se la tabella di conversione per questo campo valore esiste già, è possibile selezionarla in questa fase, o altrimenti è possibile impostare una nuova tabella di conversione. È possibile utilizzare il pulsante nella schermata precedente a tale scopo.

Dettagli tabella di conversione

Nome: * Topaz_Events_Table

Descrizione: A translation table storing the different Topaz Events that are recieved from the source system

Tipo di destinazione: * Risorsa di tipo Numero intero

Campi di origine: * Nome campo event_type

Script di conversione assegnati: Nome Stato

Adapter registrati:

Una volta creata la tabella, è necessario specificare quale dei campi di input è mappato sui campi di origine specificati in precedenza.

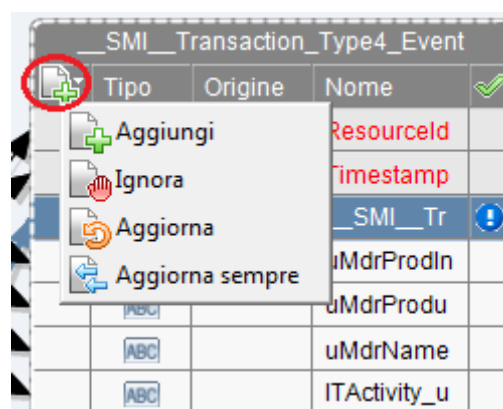
Se si desidera specificare una tabella di conversione differente per le risorse dell'adapter, in questo passaggio è possibile eseguire anche questa operazione. L'operazione avviene attraverso un processo analogo a quello sopra descritto per il campo Tipo di evento.

Nota: per impostazione predefinita, la procedura guidata di configurazione dell'adapter esegue l'allocation di tutte le risorse a una tabella di conversione denominata `Default_Translation_Table` se non diversamente specificato. Potrebbe essere corretta per le implementazioni semplici, ma per le implementazioni più complesse (e per scopi di separazione dei dati), CA consiglia di utilizzare un'altra tabella. È obbligatoria anche quando i campi di origine della sezione di mapping dell'adapter sono diversi o contengono più di un valore.

L'ultima fase del passaggio Mapping consiste nella configurazione dell'impostazione `OnDuplication` per l'adapter. Questa impostazione descrive l'azione effettuata quando viene ricevuto un secondo evento, con i valori corrispondenti a tutti i campi chiave. È possibile definire questa chiave univoca per ogni output dell'adapter (per ulteriori informazioni, consultare le sezioni seguenti). Per impostazione predefinita, il valore `OnDuplication` è impostato su `Aggiungi`, pertanto è necessario modificarlo solo se è richiesta un'azione diversa. I valori disponibili sono:

- *Aggiungi*: il nuovo evento viene aggiunto solo come un nuovo evento distinto
- *Ignora*: il nuovo evento viene ignorato (non elaborato) dall'adapter
- *Aggiorna*: il nuovo evento sostituisce l'evento caricato in precedenza solo nel caso in cui i campi Valore degli eventi siano diversi
- *Aggiorna sempre*: il nuovo evento sostituisce l'evento caricato in precedenza

L'utilizzo di opzioni diverse da `Aggiungi` può influire sulle prestazioni dell'adapter e deve essere considerato con attenzione prima di implementarlo, specialmente in caso di volumi di dati molto elevati.



Se il valore è impostato su un'opzione diversa da Aggiungi, la struttura di output visualizza una serie di caselle di controllo che consentono di definire la chiave univoca. La chiave è costituita da ogni elemento su cui viene eseguito un controllo. È necessario selezionare la chiave in base ai requisiti dopo un'attenta analisi.

__SMI__Transaction_Type4_Event				
	Tipo	Origine	Nome	
☒			ResourceId	
☒			Timestamp	
☒			__SMI__Tr	
	ABC		uMdrProdIn	
☒	ABC		uMdrProdu	
☒	ABC		uMdrName	
☐	ABC		ITActivity_u	
☐	ABC		uRuntimeN	
☐			PeriodStart	
☐	123		Applicaion_	
☐	ABC		Application	
☐	0.11		Service_uM	
☐	ABC		Service_uAI	
☐	0.11		Transaction	

Dopo aver completato la configurazione della sezione di mapping, fare clic sul pulsante Fine nella parte inferiore destra della schermata. Viene restituito l'elenco degli adapter nel sistema e l'utente dovrebbe poter visualizzare l'adapter creato con lo stato Non attivo.

Adapter

Trova

Aggiungi nuovo

Visualizzazione risultati 1 - 4 di 4

	Nome	Modalità	Tipo	Iniziatore	Indirizzo adapter	Porta	Stato
    	__SMI__APM	In linea	Adapter per file di testo	Server	LOCALHOST	6202	Non attivo
    	__SMI__SDM	In linea	Adapter per file di testo	Server	LOCALHOST	6203	Non attivo
    	__SMI__SSA	In linea	Adapter per file di testo	Server	LOCALHOST	6204	Non attivo
    	topaz_adapter	In linea	Adapter per file di testo	Server	LOCALHOST	6232	Non attivo

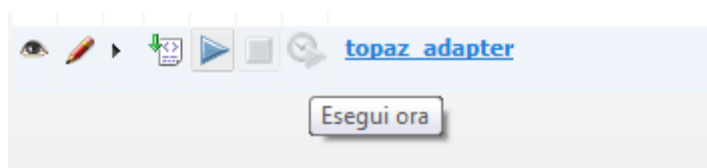
Distribuzione degli adapter

Dopo aver completato la configurazione dell'adapter, è necessario distribuire l'adapter sul server applicazioni. Facendo clic sul pulsante Avvia adapter, il servizio di distribuzione degli adapter inizia a distribuire l'adapter al file system. Include le operazioni seguenti:

- Creazione delle cartelle richieste per eseguire l'adapter
- Copia del file di configurazione XML nella cartella
- Creazione di un collegamento per eseguire l'adapter

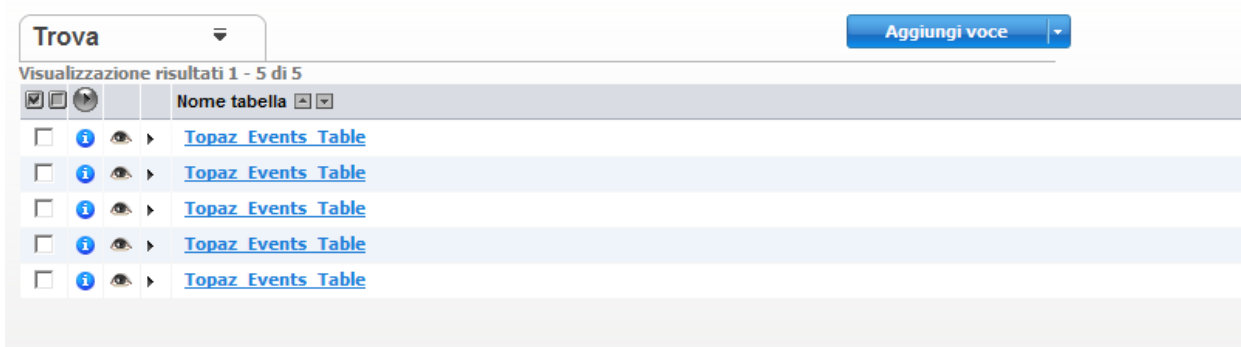
Una volta eseguite le operazioni, le modifiche apportate dall'utente si rispecchiano nel server.

Da questo punto è possibile eseguire l'adapter dalla GUI, facendo clic sul pulsante Esegui ora che è diventato attivo. In questo modo, si consente al servizio di distribuzione dell'adapter di eseguire l'adapter sul server e di iniziare la raccolta dei dati.



Una volta eseguito l'adapter, l'utente dovrebbe essere in grado di visualizzare le risorse in sospeso e gli eventi nella sezione Pending Translations (Conversioni in sospeso) del sistema.

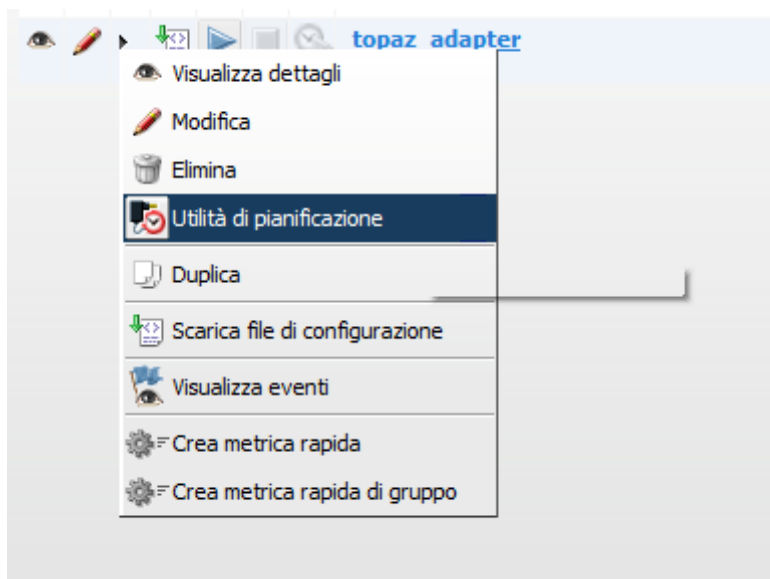
Voci di conversione



Le voci in sospeso possono quindi essere convertite in base alla normale configurazione di sistema. Una volta completata la conversione, è necessario eseguire nuovamente l'adapter per caricare i dati non elaborati nel sistema.

Pianificazione dell'adapter

Oltre all'esecuzione dell'adapter, è possibile anche impostare una pianificazione per l'adapter dalla GUI. Tuttavia, è necessario che l'adapter sia nello stato Interrotto a tale scopo. Una volta interrotto, è possibile impostare la pianificazione visualizzando il menu di scelta rapida dell'adapter e selezionando Utilità di pianificazione.



Le opzioni di pianificazione sono uguali a quelle fornite dall'Utilità di pianificazione di Windows. Nel frattempo, il servizio di distribuzione dell'adapter ha comunque creato effettivamente la pianificazione come un elemento nell'Utilità di pianificazione.

Utilità di pianificazione -- Webpage Dialog

Utilità di pianificazione

Ora di inizio: * 00 00 in base a GMT fuso orario

Schema di ricorrenza: *

☒ Giornaliero
☐ Settimanale
☐ Mensile

☒ Ogni 1 giorni
☐ Ogni giorno della settimana

Intervallo di ricorrenza: *

Inizio: 05/09/2011
dd/MM/yyyy

☒ Nessuna data di fine
☐ Termina entro:
dd/MM/yyyy

☐ Ripeti attività

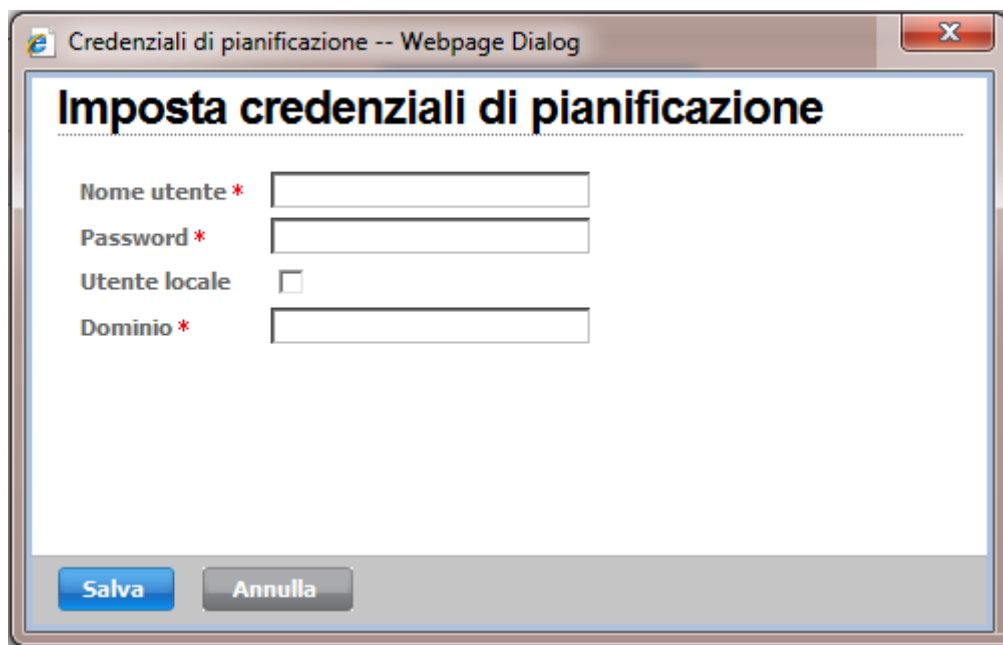
Ogni

☐ Minuti 05
☐ Ore 01

Fino a (durata): 00 Ore 00 Minuti

OK Annulla

Dopo aver aggiunto la pianificazione e quando l'adapter viene quindi distribuito, all'utente viene richiesto di immettere le credenziali utente dell'account sul server che esegue l'attività. Le credenziali da inserire sono le stesse dell'account di servizio creato per eseguire il sistema CA Business Service Insight (OblicoreSrv è l'impostazione predefinita), oppure di un altro account amministrativo, se necessario.



The image shows a Windows-style dialog box titled "Credenziali di pianificazione -- Webpage Dialog". The main heading inside is "Imposta credenziali di pianificazione". Below the heading, there are four input fields with labels: "Nome utente *" (with a red asterisk), "Password *" (with a red asterisk), "Utente locale" (with an unchecked checkbox), and "Dominio *" (with a red asterisk). At the bottom of the dialog, there are two buttons: "Salva" (Save) and "Annulla" (Cancel).

L'utilità di pianificazione integrata è una funzionalità particolarmente utile in quanto consente all'utente dell'interfaccia utente di controllare la pianificazione dell'adapter senza dover accedere direttamente al desktop del server applicazioni (ovviamente, in attesa delle autorizzazioni utente appropriate).

Case study 21: esempio di integrazione con LDAP

Requisiti dell'organizzazione

Utilizzare gli utenti già esistenti definiti nel server LDAP dell'organizzazione. Inoltre, il portale dell'organizzazione è utilizzato per la registrazione e l'accesso a CA Business Service Insight tramite la funzionalità CA Business Service Insight di accesso invisibile all'utente per portali Single Sign-On (SSO).

Definire uno script di conversione Visual Basic (VB) per la creazione automatica di utenti nel sistema CA Business Service Insight (sincronizzazione LDAP); lo script di conversione viene utilizzato per la connessione al server LDAP dell'organizzazione e l'estrapolazione dell'elenco utenti. I metodi del pacchetto di strumenti CA Business Service Insight sono utilizzati per la creazione di utenti, gruppi e ruoli.

Esempio di codice VB di connessione LDAP

```
Option Explicit On
Imports System.DirectoryServices

Public Function GetLDAPUsers(ByVal ldapServerName As String, ByVal pFindWhat
As String) As ArrayList
    Dim oSearcher As New DirectorySearcher
    Dim oResults As SearchResultCollection
    Dim oResult As SearchResult
    Dim RetArray As New ArrayList
    Dim mCount As Integer
    Dim mIdx As Integer
    Dim mLDAPRecord As String
    Dim ResultFields() As String = {"securityEquals", "cn"}
    Try
        With oSearcher
            .SearchRoot = New DirectoryEntry("LDAP://" & ldapServerName & _
"/dc=lippogeneral,dc=com")
            .PropertiesToLoad.AddRange(ResultFields)
            .Filter = "cn=" & pFindWhat & "*"
            oResults = .FindAll()
        End With
        mCount = oResults.Count
        If mCount > 0 Then
            For Each oResult In oResults
                mLDAPRecord =
oResult.GetDirectoryEntry().Properties("cn").Value & " " &
oResult.GetDirectoryEntry().Properties("mail").Value
                RetArray.Add(mLDAPRecord)
            Next
        End If
    Catch e As Exception
        MsgBox("Error is " & e.Message)
    Return RetArray
End Try
```

```

        Return RetArray
    End Function

Sub CheckAddUser
    Dim map
    Set map = Tools.GetUserDetails("acme@Test")
    'Controllare se l'utente esiste già
    'mappa Tools.AddUserByMap
    'Controllare duplicati
    map("UserName") = "acme2"
    map("UserPassword") = "acme2"
    map("UserPasswordExpirationInterval") = "50"
    map("UserDescription") = "New description"
    map("UserStatus") = "INACTIVE"
    Tools.AddUserByMap map
    Tools.Commit
End Sub

```

Metodi dello script di conversione VB CA Business Service Insight

- Metodi per l'organizzazione
 - AddOrganization / IsOrganizationExists
- Metodi per ruoli
 - IsRoleExists / SearchRoles
- Metodi per utenti
 - AddUserByMap / GetUserName
 - GetOrganizationName / IsUserExists
 - GetUserDetails / SearchUsers
 - GetUserFullName / UpdateUserByMap
- Metodi per gruppi di utenti
 - AddUserGroupByMap / IsUserGroupExists
 - DeleteUserGroup / SearchUserGroups
 - GetUserGroupDetails / UpdateUserGroupByMap

Creare un codice di accesso invisibile all'utente e integrarlo nel portale dell'organizzazione per l'accesso a CA Business Service Insight.

Esempio di codice C# del gateway CA Business Service Insight (viene integrato nel portale dell'organizzazione)

```

using System;
using System.Data;
using System.Configuration;
using System.Collections;

```

```
using System.ComponentModel;
using System.Drawing;
using System.Web;
using System.Web.Security;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Security.Cryptography.X509Certificates;
using OblicoreAuthenticationWebService;

namespace Oblicore.SSO
{
    /// <summary>
    /// This sample page is a sample gateway to Oblicore Guarantee(tm)
    application interface
    /// The page should be called prior navigating to any page at Oblicore
    Guarantee website
    /// or any page using Web Services supplied by Oblicore
    /// The OblicoreGateway page should perform the following actions:
    ///      1) Call Oblicore Authentication Web service in order to
    authenticate current user
    ///      2) Call SilentLogin.asp page at Oblicore website to login
    silently at Oblicore website
    ///      and create user session context
    ///      3) Redirect to desired page
    /// </summary>
    public partial class _Default : System.Web.UI.Page
    {

        /// <summary>
        /// Oblicore user credentials
        /// </summary>
        struct UserCredentials
        {
            public string UserName;
            public string Organization;
        }

        private void Page_Load(object sender, System.EventArgs e)
        {

            if (Request["OGSESSIONID"]!=null)
            {
                //We have been redirected back to this page from
                SilentLogin.asp after authentication.
                //Save OGSESSIONID in cookie for further use
            }
        }
    }
}
```

```

        HttpCookie SessionCookie = new
HttpCookie("OGSESSIONID",Request["OGSESSIONID"]);
        Response.Cookies.Add(SessionCookie);
        //Redirect to desired page
        Response.Redirect("/");
    }
    else
    {
        //First time we enter the page.
        //Let's perform authentication.

        string sAuthToken = string.Empty;

        // Obtain OG user name and organizations from portal user
directory
        UserCredentials ucOblicoreUser =
GetOblicoreUserCredentials();

        //Initialize Oblicore Authentication WebService
        //Project should include Web Reference to the service
        //Service is located on Oblicore Guarantee website at
/WebServices/OblicoreAuth.asmx
        OblicoreAuth oAuthService = new OblicoreAuth();
        // oAuthService.ClientCertificates.Add(x509);
        oAuthService.Url = "https://" + "localhost" +
"/WebServices/OblicoreAuth.asmx";
        try
        {
            //Invoke authentication Web Service.
            //The AuthenticateUser method returns encrypted
token, which should be passed to
            //SilentLogin.asp page, located in root folder of
Oblicore Guarantee website
            sAuthToken =
oAuthService.AuthenticateUser(ucOblicoreUser.UserName,ucOblicoreUser.Organization
);
        }
        catch (Exception ex)
        {
            //Proceed authentication error if any
            Response.Write("The error has occurs during
Oblicore authentication: " + ex.Message);
            Response.End() ;
        }

        //Call SilentLogin.asp page along with passing it
authentication folder
        //SilentLogin.asp page is located Oblicore Guarantee
website root folder

```

```
        //After logging in, SilentLogin.asp page will redirect us  
        back to the current page along with passing OGSESSIONID parameter
```

```
        //Response.Redirect(ConfigurationSettings.AppSettings["OGURL"].ToString() +  
        "/SilentLogin.asp?AuthToken="+Server.UrlEncode(sAuthToken)+"&DesiredPage="+GetCur  
        rentPageURL());
```

```
        Response.Redirect("https://vit-  
05/SilentLogin.asp?AuthToken=" + Server.UrlEncode(sAuthToken) +  
        "&DesiredPage=/Oblicore.asp"); // + GetCurrentPageURL());  
    }  
}
```

```
    /// <summary>  
    /// Obtain Oblicore Guarantee user name and organization from portal  
    user directory  
    /// The method is supposed to call ActiveDirectory or another  
    repository using portal API  
    /// to obtain current user name and organization in terms of Oblicore  
    Guarantee
```

```
    /// </summary>  
    /// <returns>Oblicore Guarantee user credentials struct</returns>  
    private UserCredentials GetOblicoreUserCredentials()  
    {  
        UserCredentials ucOblicoreUser = new UserCredentials();  
        //currently alwasy assume user is sadmin and organization is  
        Oblicore (default)  
        ucOblicoreUser.UserName = "sadmin";  
        ucOblicoreUser.Organization = "Oblicore";  
        return ucOblicoreUser;  
    }
```

```
    /// <summary>  
    /// Retrieves current page URL  
    /// </summary>  
    /// <returns>Full URL of current page</returns>  
    private string GetCurrentPageURL()  
    {  
        string s =  
        (Request.ServerVariables["HTTPS"]==null||Request.ServerVariables["HTTPS"].ToLower  
        ()=="off")?"http://":"https://";  
        s += Request.ServerVariables["SERVER_NAME"] +  
        Request.ServerVariables["URL"];  
        if (Request.QueryString.ToString() != string.Empty)  
        {  
            s += "?" + Request.QueryString.ToString();  
        }  
    }
```

```

        return s;
    }

    #region Web Form Designer generated code
    override protected void OnInit(EventArgs e)
    {
        //
        // CODEGEN: This call is required by the ASP.NET Web Form
Designer.
        //
        InitializeComponent();
        base.OnInit(e);
    }

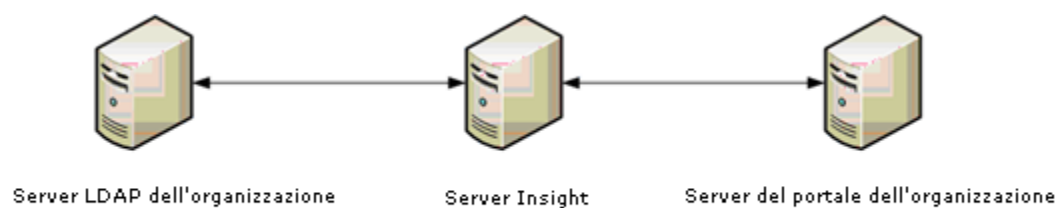
    /// <summary>
    /// Required method for Designer support - do not modify
    /// the contents of this method with the code editor.
    /// </summary>
    private void InitializeComponent()
    {
        this.Load += new System.EventHandler(this.Page_Load);
    }
    #endregion
}

```

■ Diagramma di flusso

Il seguente diagramma mostra il processo di sincronizzazione degli utenti e il flusso di accesso al portale. Lo script di conversione viene configurato per l'esecuzione periodica. Lo script mantiene l'elenco di utenti LDAP aggiornato e aggiunge/rimuove gli utenti a seconda delle esigenze.

L'utente esegue l'accesso al portale dell'organizzazione. È possibile configurare il portale in modo da reindirizzare l'utente al server CA Business Service Insight o visualizzare un elenco di applicazioni disponibili. Il server CA Business Service Insight utilizza le credenziali fornite durante il primo accesso al portale.



Elenco utenti LDAP:

John
Albert
Sandy
Allison

**Utenti Insight,
inclusi gli utenti
LDAP sincronizzati:**

Sadmin
John
Albert
Sandy
Allison



Case study 22: generazione di report con PERL

L'esempio seguente mostra la procedura di utilizzo dello script PERL per connettersi al servizio Web di report CA Business Service Insight e trasferire il parametro xml dei criteri in una busta SOAP tramite un flusso HTTP.

Nota: il codice XML trasferito nella busta SOAP non può contenere i simboli < o >; è ammesso invece il codice HTML di questi simboli (ad esempio, <=> >=<)

```
#!/usr/bin/perl
```

```
#use strict;
```

```
use LWP::UserAgent;  
use HTTP::Request::Common;  
use XML::Simple;  
use Data::Dumper;
```

```
my $userAgent = LWP::UserAgent > new(agent => 'Mozilla/5.0');
```

```
### Creating a Oblicore Session Context - Oblicore Session ID is stored in $scid  
###
```

```
my $message = "<?xml version=\"1.0\" encoding=\"utf-8\"?>  
<soap:Envelope xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\"  
xmlns:xsd=\"http://www.w3.org/2001/XMLSchema\"  
xmlns:soap=\"http://schemas.xmlsoap.org/soap/envelope/\">  
  <soap:Body>  
    <CreateSessionContext xmlns=\"http://www.oblicore.com\">  
      <userName>sadmin</userName>  
      <organizationName>Oblicore</organizationName>  
    </CreateSessionContext>  
  </soap:Body>  
</soap:Envelope>";
```

```
my $response = $userAgent > request(POST  
'http://obonob/WebServices/OblicoreAuth.asmx',Content_type => 'text/xml', Content  
=> $message);
```

```
print $response > error_as_HTML unless $response > is_success;
```

```
my $result = $response > as_string;
```

```
my $xmlstart = index $result, "<?xml";  
my $xmlend = length $result;
```

```
my $xmltext = substr $result, $xmlstart, ($xmlend - $xmlstart);
```

```
my $xml = new XML::Simple;  
my $data = $xml > XMLin($xmltext);
```

```
my $scid = ($data > {'soap:Body'} > {'CreateSessionContextResponse'} >
{'CreateSessionContextResult'});

print "Session ID : ".$scid."\n";

### Try to get contract list from Oblicore ###

my $criteria_xml =
"&lt;Criteria&gt;&lt;Name&gt;a*&lt;/Name&gt;&lt;Status&gt;EFFECTIVE&lt;/Status&gt;
&lt;/Criteria&gt;";

print "<?xml version=\"1.0\" encoding=\"utf-8\"?>
<soap:Envelope xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\"
xmlns:xsd=\"http://www.w3.org/2001/XMLSchema\"
xmlns:soap=\"http://schemas.xmlsoap.org/soap/envelope/\">
  <soap:Body>
    <GetContractsAdvanced xmlns=\"http://www.oblicore.com\">
      <sessionID>".$scid."</sessionID>
      <criteriaXML>".$criteria_xml."</criteriaXML>
    </GetContractsAdvanced>
  </soap:Body>
</soap:Envelope>";

my $message = "<?xml version=\"1.0\" encoding=\"utf-8\"?>
<soap:Envelope xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\"
xmlns:xsd=\"http://www.w3.org/2001/XMLSchema\"
xmlns:soap=\"http://schemas.xmlsoap.org/soap/envelope/\">
  <soap:Body>
    <GetContractsAdvanced xmlns=\"http://www.oblicore.com\">
      <sessionID>".$scid."</sessionID>
      <criteriaXML>".$criteria_xml."</criteriaXML>
    </GetContractsAdvanced>
  </soap:Body>
</soap:Envelope>";

#my $message = "<?xml version=\"1.0\" encoding=\"utf-8\"?>
#<soap:Envelope xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\"
#xmlns:xsd=\"http://www.w3.org/2001/XMLSchema\"
#xmlns:soap=\"http://schemas.xmlsoap.org/soap/envelope/\">
#  <soap:Body>
#    <GetContracts xmlns=\"http://www.oblicore.com\">
#      <sessionID>".$scid."</sessionID>
#    </GetContracts>
#  </soap:Body>
#</soap:Envelope>";

### is_well_formed($message)
```

```
print $message;

my $response = $userAgent > request(POST
'http://obonob/WebServices/Contracts.asmx', Content_Type => 'text/xml', Content
=> $message);

print $response > error_as_HTML unless $response > is_success;

my $result = $response > as_string;

print Dumper($result); # Output of contract list

### Close the Oblicore Session ###

my $message = "<?xml version=\"1.0\" encoding=\"utf-8\"?>
<soap:Envelope xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\"
xmlns:xsd=\"http://www.w3.org/2001/XMLSchema\"
xmlns:soap=\"http://schemas.xmlsoap.org/soap/envelope/\">
  <soap:Body>
    <ClearSessionContext xmlns=\"http://www.oblicore.com\">
      <sessionContextID>\".$scid.\"</sessionContextID>
    </ClearSessionContext>
  </soap:Body>
</soap:Envelope>";

my $response = $userAgent > request(POST
'http://obonob/WebServices/OblicoreAuth.asmx', Content_Type => 'text/xml',
Content => $message);

print $response > error_as_HTML unless $response > is_success;
```


Appendice C: Specifiche di configurazione dell'adapter

Questa sezione contiene i seguenti argomenti:

[Struttura generale](#) (a pagina 317)

[Sezione General](#) (a pagina 318)

[Sezione Interface CA Business Service Insight](#) (a pagina 324)

[Sezione DataSourceInterface](#) (a pagina 327)

[Sezione SQL Interface](#) (a pagina 330)

[Sezione InputFormatCollection](#) (a pagina 334)

[Sezione TranslationTableCollection](#) (a pagina 339)

[Sezione TranslatorCollection](#) (a pagina 341)

Struttura generale

La struttura generale del file XML di configurazione è la seguente:

```
< AdapterConfiguration>
  <General...>
  <OblicoreInterface...>
  <DataSourceInterface...>
  <InputFormatCollection...>
  <TranslatorCollection...>
  <TranslationTableCollection...>
</ AdapterConfiguration>
```

Ciascun nodo viene descritto nelle sezioni seguenti.

Sezione General

La sezione General è composta dagli attributi e dai sottonodi seguenti:

Struttura XML:

```
<General
  MajorVersion="2345"
  MinorVersion="01245"
  WorkingDirectoryName=" output"
  DataSourceControlFileName="DatasourceControl.xml"
  SendFileName="Send.txt"
  SendControlFileName="SendControl.xml"
  LogDebugMode="no"
  ConsoleDebugMode="no"
  RunOnce="yes"
  RepeatUntilDry="yes"
  RuntimeErrorLimit="1"
  RetryRejectedEvents="yes"
  RejectedEventsFileName="rejectedEvents.txt"
  RejectedEventsUpperLimit="1000"
  RegExUnlimitedSearch (V3.1 Patch)="no"
  HoldRejectedEventsSpan="24"
  GenerateStatistics="yes"
  StatisticsFileName="MyStatistics.txt" >
  KeepHistoricState="yes" >
  DefaultTimeFormat="%Y/%m/%d-%H:%M:%S"
  DefaultDecimalSymbol="."
  DataSourceIdMaxLength="60"
  DefaultDigitGroupingSymbol=","
  SaveIllegalEvents ="no"
  WriteEventErrorToLog ="yes"
  IllegalEventsDirectoryName="xxxpath"
  <DataSourceDifferenceFromUTC
    DefaultOffset="2"
    TimeFormat="%Y/%m/%d-%H:%M:%S">
    <DayLightSaving
      From="2001/04/09-12:00:00"
      To="2001/09/01-12:00:00"
      Shift="1"/>
  </ DataSourceDifferenceFromUTC >
</General>
```

- MajorVersion: specifica il numero di versione principale.
- MinorVersion: specifica il numero di versione secondaria.
- WorkingDirectoryName: (facoltativo)

Specifica il percorso predefinito per i file di output dell'adapter (controllo origine dati, conversioni, invii).

Valore predefinito: la directory di output (cartella) nella directory contenente il file di configurazione.

■ DataSourceControlFileName: (facoltativo)

Nome del file che l'adapter utilizzata per tenere traccia dell'ultimo record di dati recuperati dall'origine dati.

Valore predefinito: DataSourceControl.xml (se non altrimenti specificato, viene utilizzato il valore predefinito).

■ SendFileName: (facoltativo)

Nome del file che contiene gli eventi prima che vengano inviati a CA Business Service Insight.

Valore predefinito: Send.txt (se non altrimenti specificato, viene utilizzato il valore predefinito).

■ SendControlFileName: (facoltativo)

Nome del file che l'adapter utilizza per tenere traccia del processo di invio.

Valore predefinito: SendControl.xml (se non altrimenti specificato, viene utilizzato il valore predefinito).

■ DataSourceIdMaxLength:

Lunghezza massima per il campo DataSourceID nella tabella t_raw_data. Se l'utente inserisce una stringa con lunghezza superiore, riceve un errore nell'adapter.

Il valore deve essere minore o uguale alla dimensione reale della tabella.

Valore predefinito: 60

■ SaveIllegalEvents: se selezionato, questo attributo indica che gli eventi non validi devono essere scritti nel file.

Valore predefinito: "no"

■ WriteEventErrorToLog: guida se errori di dati vengono scritti nella tabella T_Log

Valori validi: [sì/no]

Valore predefinito: sì

■ IllegalEventsDirectoryName: (nessun valore predefinito)

■ SendFileName: (facoltativo)

Nome del file che contiene i record prima che vengano inviati a CA Business Service Insight.

Valore predefinito: Send.txt (se non altrimenti specificato, viene utilizzato il valore predefinito).

■ \$SendControlFileName: (facoltativo)

Nome del file che l'adapter utilizza per tenere traccia del processo di invio.

Valore predefinito: SendControl.xml (se non altrimenti specificato, viene utilizzato il valore predefinito).

- LogDebugMode: (facoltativo)

Valori validi: [sì/no]

Se impostato su "yes" (sì), al log vengono inviati i seguenti elementi: riga originale dall'origine dati, risultati di analisi, evento unificato CA Business Service Insight.

- ConsoleDebugMode: (facoltativo)

Valori validi: [sì/no]

Se impostato su "yes" (sì), sulla console vengono visualizzati i messaggi di debug.

- **Indicatori per la lettura e la conversione dei record:**

- **.**: per un record che è stato letto e convertito in un evento di output.
- **i**: per un record che è stato ignorato dal parser di espressione regolare.
- **I**: per un record che è stato letto e ignorato dalle tabelle di conversione.
- **R**: per un record che è stato letto ma rifiutato dalla tabella di conversione (non può essere convertito dalle tabelle di conversione).
- **X**: per un record per cui si è verificato un problema durante l'analisi. Verrà ignorato e andrà perso o verrà salvato nella directory degli eventi non validi.

Nota: quando il record letto passa attraverso più convertitori, l'indicazione del record verrà racchiusa tra parentesi. Ad esempio: [...] o [RRI].

- **Indicatori di stato dell'adapter:**

- **0**: l'adapter è attivo e non legge alcun record nell'ultimo secondo.
- **E**: stato di errore.
- **P**: stato di sospensione.
- **S**: comando di interruzione ricevuto da CA Business Service Insight.
- **B**: stato bloccato, la tabella degli eventi rifiutati è piena.
- **Indicatori delle tabelle di conversione:**
 - **L**: in attesa delle tabelle di conversione.
 - **T**: tabella di conversione inviata o ricevuta da CA Business Service Insight.
 - **t**: modifiche nella tabella di conversione inviata o ricevuta da CA Business Service Insight.
- **Indicatori di connessione:**
 - **<Connecting CA Business Service Insight:** tentativo di connettere CA Business Service Insight.
 - **<Connecting CA Business Service Insight>:** connessione stabilita.

- <Connecting DataSource: tentativo di connettere l'origine dati.
- <Connecting DataSource>: connessione stabilita.
- RunOnce: (facoltativo)
Valori validi: [sì/no]
Se impostato su "no", l'adapter viene eseguito di continuo.
Se impostato su "yes" (sì), l'adapter per file di testo viene eseguito, legge i record e si interrompe automaticamente. L'adapter per file legge i file interi, attende alcuni secondi, tenta di leggere i nuovi record (a seconda di SleepTime). Se non sono presenti tali record, si interrompe.
Un adapter SQL esegue ogni singola query solo una volta. Se RepeatUntilDry è impostato su "no", si interrompe immediatamente. Se RepeatUntilDry è impostata su "yes" (sì), attende (a seconda di SleepTime), prova nuovamente a eseguire le query (a seconda di SleepTime della query) e, se non sono presenti nuovi record, si interrompe.
- RepeatUntilDry: (facoltativo)
Valori validi: [sì/no] Valore predefinito: "yes" (sì)
Fare riferimento all'attributo RunOnce descritto sopra.
- RuntimeErrorsLimit: (facoltativo)
Determina il limite di errori (ad esempio, errori negli eventi di input) prima che l'adapter sia bloccato. Quando è uguale a 0, l'adapter non è bloccato.
Valore predefinito: 1 (indica che l'adapter è bloccato dopo il primo errore).
- RetryRejectedEvents: (facoltativo)
Valori validi: [sì/no] Valore predefinito: "yes" (sì)
Se impostato su "yes" (sì), i record che necessitano di conversione sono tenuti nel file degli eventi rifiutati, in attesa di conversione.
Non è consigliato impostare l'attributo su "no", perché si potrebbero perdere dati importanti.
- RejectedEventsFileName: (facoltativo)
Nome del file che l'adapter utilizza per tenere traccia dei record di dati recuperati dall'origine dati e in attesa di conversione.
Valore predefinito: rejected.txt (se non altrimenti specificato, viene utilizzato il valore predefinito).
- RejectedEventsUpperLimit- (facoltativo)
Determina il limite di record rifiutati che sono riportati nel file rejected.txt. Quando l'adapter raggiunge tale limite superiore, si interrompe ed entra in stato bloccato finché questi record non vengono convertiti.
Valore predefinito: 1000

- **RegexUnlimitedSearch:** guida l'adapter per eseguire la ricerca completa in base all'espressione regolare.
Valori validi: [sì/no]
Valore predefinito: "no"
- **HoldRejectedEventsSpan:** (facoltativo)
Determina il numero di ore per cui salvare il file degli eventi rifiutati. Se questo attributo non viene specificato, l'adapter non cancella alcun evento che deve essere gestito prima che l'adapter possa avviarsi nuovamente.

Non è consigliato utilizzare questo attributo, perché si potrebbero perdere dati importanti.
- **GenerateStatistics:** (facoltativo)
Valori validi: [sì/no] Valore predefinito: "yes" (sì)
Se impostato su "yes" (sì), l'adapter crea un file di statistiche contenente informazioni statistiche per ogni minuto.
- **StatisticsFileName:** (facoltativo)
Nome del file di statistiche.

Valore predefinito: statistics.txt (se non altrimenti specificato, viene utilizzato il valore predefinito).
- **KeepHistoricState:** (facoltativo)
Valori validi: [sì/no] Valore predefinito: "no"

Se impostato su "yes" (sì), l'adapter consente di salvare tutti i file, escluso il file di log, in una nuova directory denominata Historic state yyyyymmdd-hhmmss, in cui yyyyymmdd e hhmmss sono i formati di data e ora di creazione.
- **DefaultTimeFormat:** (facoltativo)
Formato ora predefinito. Se questo attributo è specificato, viene utilizzato come formato ora in qualsiasi posizione in cui l'attributo TimeFormat è omissivo. Se non è specificato, l'attributo TimeFormat in altri elementi è obbligatorio.
- **DefaultDecimalSymbol:** (facoltativo)
Simbolo decimale per campi di tipo reale.

Valore predefinito: (se non altrimenti specificato, viene utilizzato il valore predefinito).
- **DefaultDigitGroupingSymbol:** (facoltativo)
Simbolo di raggruppamento cifre predefinito per i campi di tipo numero intero e reale.

Valore predefinito: (se non altrimenti specificato, viene utilizzato il valore predefinito).

- DataSourceDifferenceFromUTC: indica la differenza di tempo tra l'ora UTC e il fuso orario dell'origine dati nel computer.
 - DefaultOffset: differenza da UTC durante i periodi senza ora legale.
 - TimeFormat: specifica il formato in base al quale devono essere analizzate le informazioni sull'ora legale (descritte in seguito).
 - DayLightSaving: specifica un periodo di ora legale del fuso orario dell'origine dati. Questo elemento è facoltativo (ovvero, se non è selezionato, non sono presenti periodi di ora legale) e può esistere più di una volta. Quando sono presenti alcuni elementi, devono essere ordinati per ora e i periodi non devono sovrapporsi.
 - From: data di inizio del periodo.
 - To: data di fine del periodo.
 - Shift: ore di differenza aggiunte a DefaultOffset nel periodo di ora legale.

Sezione Interface CA Business Service Insight

La sezione Interface CA Business Service Insight è costituita da attributi che specificano la connessione a CA Business Service Insight. Sono presenti due modalità, in linea e non in linea. In modalità in linea, l'adapter si connette a CA Business Service Insight, riceve le tabelle di conversione e i comandi di controllo da CA Business Service Insight e invia eventi, stati e richieste di conversione a CA Business Service Insight. In modalità non in linea, l'adapter utilizza il file delle tabelle di conversione locale e scrive gli eventi in un file di output.

Struttura XML per la modalità in linea:

```
<OblicoreInterface
  Mode="online">
  <OnlineInterface
    Port="3006"
    ConnectionInitiator="server"
    Address="localhost"
    SecurityLevel="high"
    SecurityKey="12345678901234567890123456789012"
    UseAcknowledgeProtocol="yes"
    PacketSize="50"
    PacketDeadline="60"
    PacketResendTimeout="60"
    WindowSize="10"
  />
</OblicoreInterface>
```

- Port: numero di porta TCP/IP che l'adapter utilizza per comunicare con il server CA Business Service Insight.
- ConnectionInitiator: (facoltativo)
Valori validi: server/adapter. Valore predefinito: server.
Definisce l'iniziatore della connessione, l'adapter o il listener dell'adapter su CA Business Service Insight.
L'iniziatore svolge il ruolo slave o client, mentre l'altro svolge il ruolo master o server.
- Address: indirizzo di rete del server obbligatorio quando l'iniziatore è l'adapter.
- SecurityLevel: definisce il livello di autenticazione tra l'adapter e il server CA Business Service Insight (handshake). Opzioni di livello:
 - none: non viene eseguita alcuna autenticazione.
 - high: autenticazione eseguita. SecurityKey deve essere specificato.
- Securitykey: stringa di 32 cifre. Deve essere la stessa stringa definita per l'adapter nel database CA Business Service Insight.
Il flusso del processo di handshake è il seguente:

- Il server CA Business Service Insight si connette all'adapter.
- Una stringa casuale viene inviata dall'adapter al server CA Business Service Insight.
- Il server CA Business Service Insight crittografa la stringa con una chiave preconfigurata e rinvia il risultato all'adapter.
- L'adapter crittografa la stessa stringa casuale inviata al server CA Business Service Insight utilizzando la stringa SecurityKey e confronta i risultati.
- Il server CA Business Service Insight compone una stringa in modo casuale e invia la stringa all'adapter.
- L'adapter crittografa la stringa e la rinvia al server CA Business Service Insight.
- Il server CA Business Service Insight crittografa la stessa stringa e confronta i risultati con i dati ricevuti dall'adapter.
- La connessione è stabilita.
- Se viene rilevato un errore in una delle fasi nel flusso, la connessione non viene stabilita.

■ UseAcknowledgeProtocol: (facoltativo)

Valori validi: [sì/no] Valore predefinito: "yes" (sì)

Se impostato su "yes" (sì), l'adapter utilizza il protocollo di riconoscimento. Se impostato su "no", l'adapter invia i messaggi/pacchetti e non attende che la conferma di ricezione da CA Business Service Insight.

Non è consigliato impostare l'attributo su "no", perché si potrebbero perdere eventi.

■ PacketSize: (facoltativo)

Valori validi: da 1 a 1000

Valore predefinito: 50

Numero massimo di eventi in un pacchetto.

■ PacketDeadline: (facoltativo)

Valori validi: da 1 a 3600

Valore predefinito: 60

Numero di secondi prima dell'invio di un pacchetto che non è pieno.

■ PacketResendTimeout: (facoltativo)

Valori validi: da 1 a 3600

Valore predefinito: 60

Quantità di tempo, in secondi, di attesa per il messaggio di conferma di ricezione prima del nuovo invio del pacchetto. Questo attributo è applicabile solo se l'attributo UseAcknowledgeProtocol è impostato su "yes" (sì).

- WindowSize: (facoltativo)

Valori validi: da 1 a 100

Valore predefinito: 10

Numero di pacchetti nella finestra. Questo attributo è applicabile solo se l'attributo UseAcknowledgeProtocol è impostato su "yes" (sì).

Struttura XML per la modalità non in linea:

```
<OblicoreInterface
  Mode="offline">
  <OfflineInterface
    OutputFileName="outputEvents.txt"
    OutputFileType="tsv"
    OutputTimeFormat="%Y/%m/%d %H:%M:%S"
    OutputDecimalSymbol="."
  />
</OblicoreInterface>
```

- OutputFileName: (facoltativo)

Nome del file in cui l'adapter scrive gli eventi di output.

Valore predefinito: AdapterOutput.txt (se non altrimenti specificato, viene utilizzato il valore predefinito).

- OutputFileType: (facoltativo)

Valori validi: csv/tsv/xml

Definisce il formato dell'evento.

- OutputTimeFormat: definisce il formato dei campi di data. Questo attributo può essere omesso se nella sezione General è stato definito l'attributo DefaultTimeFormat.

- OutputDecimalSymbol: (facoltativo)

Definisce il simbolo decimale per i campi di tipo reale.

Valore predefinito: . (punto)

Sezione DataSourceInterface

La sezione DataSourceInterface è composta da attributi che specificano la connessione e il tipo di connessione tra l'adapter e l'origine dati (strumento di misurazione, CRM, log di sistema, ecc.) ed è suddivisa in due tipi principali: File Interface e SQL Interface.

File Interface

È possibile utilizzare l'adapter per file di testo per recuperare dati da file di log, report pianificati o qualsiasi altro file basato su testo, e la sezione DataSourceInterface definisce regole di analisi delle informazioni dall'origine dati del file e la relativa estrazione in campi.

La sezione DataSourceInterface definisce inoltre il modo in cui l'adapter gestisce il file di origine (se elimina il file originale quando è stato creato solo per l'adapter, o se conserva i dati quando sono necessario per altri usi, e così via).

Struttura XML:

```
<DataSourceInterface WorkFileName="MyWorkingFile.txt" >
  <Files>
    <File
      IsActive="yes"
      InputFormat="events"
      Path="D:\adapters\sample_data\"
      NamePattern="adapterXXX*.log"
      DeleteFileAfterProcessing="no"
      Delimiters=", "
      IgnoreRedundantDelimiters ="no"
      TitleExists="no"
      SleepTime="10">
    </File>
  </Files>
</DataSourceInterface>
```

- **WorkFileName:** (facoltativo). Quando DeleteFileAfterProcessing è impostato su "no" (il file viene copiato con questo nome se impostato su "yes"), il file viene rinominato con questo nome. Se non specificato, verrà utilizzato il valore predefinito (WorkFile.txt).
- **Files:** raccolta di elementi di file (è possibile definire più file in un adapter).
- **File:** specifica gli attributi del file.
 - **IsActive:** (facoltativo) [si/no]. Definisce se il file viene attivato (se impostato su "no" il file non verrà letto).
 - **InputFormat:** InputFormat associato al file. L'adapter utilizza InputFormat per estrarre i dati dall'origine dati.
 - **Path:** percorso di ubicazione dei file dei dati di origine.

- **NamePattern**: indica il nome del file di origine dati. È possibile utilizzare caratteri jolly, se più file utilizzano lo stesso formato di input.
- **DeleteFileAfterProcessing** [sì/no]: modo in cui l'adapter gestisce il file di origine. Quando il file è stato creato solo per l'adapter e può essere eliminato dopo l'elaborazione, impostarlo su "yes" (sì). Il file viene rinominato, elaborato e quindi eliminato. Se impostato su "no", il file viene copiato e l'elaborazione ha luogo nella copia del file. Se vengono aggiunti nuovi record alla fine di questo file, l'adapter copia questi nuovi record nel file di lavoro durante il successivo ciclo. Se non vengono aggiunti nuovi record al file, l'adapter cerca il primo file con lo stesso modello e nome maggiore (in ordine lessicografico) rispetto al file corrente. Se l'adapter trova tale file, continua l'elaborazione su questo file. L'adapter non ripristina il file precedente, anche se vengono aggiunti nuovi record al file. Utilizzare "no" quando è necessario conservare l'integrità del file di origine.
- **InitialFileName**: il primo nome del file da cui l'adapter esegue la ricerca del file con il modello specificato. Utilizzare questo attributo quando NamePattern contiene caratteri jolly e non si desidera che l'adapter legga i file precedenti.
- **Delimiters**: (facoltativo). Uno o più caratteri che fungono da delimitatori, in base ai quali le righe di dati vengono analizzate nei campi; se non specificato, il valore predefinito è \t.
- **IgnoreRedundantDelimiters**: (facoltativo) [sì/no]. Quando è impostato su "yes" (sì), i delimitatori consecutivi verranno considerati come unico; in caso contrario, verrà creato un campo vuoto tra i delimitatori.
- **RegexForParser**: (facoltativo). Espressione regolare da utilizzare per estrarre i campi dell'origine dati che sostituisce l'attributo Delimiters appena descritto. Ad esempio:

```
<File
...
RegexForParser="^(.{8}) (.{6}) (?:[ ])*([0-9]+) "
/>
```

Per ulteriori informazioni, consultare la documentazione sulle espressioni regolari.

- **TitleExists**: (facoltativo) [sì/no] indica se la prima riga non vuota nel file di origine dati è una riga di titolo. Il titolo può essere utilizzato dall'adapter durante l'analisi dei dati.
- **SleepTime**: indica l'intervallo di recupero dei dati (in secondi), ad esempio, l'interruzione in secondi tra il recupero di dati dell'adapter dal file dei dati di origine.
- **LogicLineDefinition**: (facoltativo)

```
<File
...
<LogicLineDefinition
    FirstLine="Job server:.*"
    NumberOfLines="5"
/>
```


/>

Qualora il set di dati sia creato dal numero di righe e non da una riga, i seguenti attributi definiscono il punto di inizio dell'estrazione, il punto finale e il numero di righe che compongono i dati:

- AllFile: (facoltativo) [sì/no] se impostato su "yes" (sì), tutti i file vengono considerati come un record, una riga logica.
- FirstLine: (facoltativo) espressione regolare che definisce la prima riga della riga logica. Può essere specificato con/senza LastLine e/o NumberOfLines.
- LastLine: (facoltativo) espressione regolare che definisce l'ultima riga della riga logica. Può essere specificato con/senza FirstLine e/o NumberOfLines.
- NumberOfLines: (facoltativo) numero di righe in una riga logica. Può essere specificato con/senza FirstLine e/o LastLine.
- MatchCase: (facoltativo) [sì/no] definisce se nella corrispondenza con l'espressione regolare è presente la distinzione tra maiuscole e minuscole.

Sezione SQL Interface

È possibile utilizzare l'adapter SQL per recuperare i dati dai database utilizzando un'istruzione SQL.

L'interfaccia SQL definisce la connessione al database e le query utilizzate per recuperare i dati, come indicato di seguito:

Struttura XML:

```
< DataSourceInterface >
  <ConnectionString ConnectionTimeout="60" QueryTimeout="30">
    <![CDATA[ Driver={Microsoft Access Driver
(*.mdb)};DataBaseeq=d:\Oblicore\database1.mdatabase; ]]>
  </ConnectionString>
  <QueryCollection>
    <Query QueryName="cases" InputFormat="cases" SleepTime="3600">
      <SelectStatement AutoCompleteQuery="yes">
        select
dateclosed,callid,dateopened,companyname,priority,closedmn,responsemn
        from calls where dateclosed is not NULL
      </SelectStatement>
      <QueryKeyFields>
        <KeyField Name="dateclosed" Sort="asc"/>
        <KeyField Name="callid" Sort="desc"/>
        <SelectInitialValues>
          Select min(dateclosed) , 'min date' from calls
        </SelectInitialValues>
      </QueryKeyFields>
    </Query>
    <Query QueryName="contracts" InputFormat="contracts" SleepTime="3600">
      <ConnectionString>
        <Segment Type="text"
Text=" Driver={Microsoft Excel Driver (*.xls)}; DriverId=790;
DataBaseeq="/>
        <Segment Type="File">
          <File Path="d:\Oblicore " NamePattern="Availabilty_*.XLS">
        </Segment>
        <Segment Type="text" Text=";" />
      </ConnectionString>
      <SelectStatement AutoCompleteQuery="yes">...</SelectStatement>
      <QueryKeyFields>...</QueryKeyFields>
    </Query>
  </QueryCollection>
</DataSourceInterface>
```

- **ConnectionString:** stringa di connessione per l'accesso al database di origine.

ConnectionString può essere definito nell'elemento DataSourceInterface e/o negli elementi Query. La definizione ConnectionString nell'elemento DataSourceInterface è l'impostazione predefinita e viene utilizzata solo in una query senza definizione ConnectionString.

La stringa di connessione può essere definita in una stringa o per segmenti. Quando l'elemento di stringa di connessione non contiene elementi Segment, la stringa di connessione viene ottenuta dal testo dell'elemento ConnectionString. Se contiene almeno un elemento di segmento, la stringa di connessione è concatenata da questo.

Esistono due tipi di segmento. Il primo è di tipo testo e contiene il testo che viene concatenato alla stringa di connessione così com'è. Il secondo è un file e contiene un nome di file con o senza caratteri jolly. È possibile visualizzare il segmento di file una sola volta e contiene altri attributi che definiscono la procedura da eseguire con il file letto.

- **ConnectionTimeout:** (facoltativo). Numero intero positivo che indica, in secondi, il tempo di attesa per l'apertura della connessione. Il valore 0 indica un tempo di attesa indefinito finché la connessione non è aperta. Il valore predefinito è 15 secondi.

Nota: alcuni provider non supportano questa funzionalità.

- **QueryTimeout:** (facoltativo). Numero intero positivo che indica, in secondi, il tempo di attesa per l'esecuzione della query. Il valore 0 indica un tempo di attesa indefinito finché la query non è completata. Il valore predefinito è 30 secondi.

Nota: alcuni provider non supportano questa funzionalità.

- **Segment:** specifica gli attributi di segmento.
- **Type:** (facoltativo) (testo, file) tipo di segmento
- **Text:** il testo del segmento di testo.
- **File:** specifica gli attributi di file
- **Path:** percorso di ubicazione dei file dei dati di origine.
- **NamePattern:** specifica il nome del file di origine dati. Può contenere caratteri jolly.
- **InitialFileName:** (facoltativo). Indica all'utente il file da utilizzare per iniziare la ricerca e il modello da cercare.
- **UsePath:** (facoltativo) [yes/no] se impostata su "yes", il percorso del file viene concatenati alla stringa di connessione.
- **UseFileName:** (facoltativo) [yes/no] se impostato su "yes", il nome del file viene concatenato alla stringa di connessione (necessario per i file di Excel).
- **UpdateSchemaFile:** (facoltativo) [yes/no]. Se impostato su "yes", l'adapter aggiorna il file schema.ini con il nome del file corrente.

Nota: utilizzare questo attributo solo quando si desidera che l'adapter modifichi il file schema.ini (database per file di testo).

- QueryCollection: raccolta di query (è possibile definire più query in un adapter).
 - PreliminaryCheck-: (facoltativo [yes/no]). L'adapter controlla le query durante la connessione al database. Se questo attributo è impostato su "no", l'adapter verifica le query eseguendole senza alcuna modifica, continua l'esecuzione sui record restituiti in un secondo momento senza leggerli nuovamente. Quando è impostato su "yes", l'adapter sostituisce ogni stringa "where" nell'istruzione select con "where 1=2 and" e solo in seguito esegue le query. Utilizzare questa opzione per verificare tutte le query prima che l'adapter avvii l'esecuzione sui record e quando questa aggiunta riduce in modo notevole il tempo di query. Nota: alcuni provider non ottimizzano il processo di query e quando la query è valida, l'esecuzione richiede lo stesso tempo impiegato senza l'aggiunta.
 - ExternalCommand: (facoltativo). Comando esterno eseguito prima che l'adapter avvii un nuovo ciclo di lettura.
 - Command: nome del file batch da eseguire.
 - SleepTime: numero intero positivo che indica, in secondi, il tempo di attesa prima della nuova esecuzione del comando.
- Query: indica gli attributi della query.
 - QueryName: nome della query.
 - IsActive: (facoltativo) [yes/no]. Se impostato su "no", l'adapter non legge i record da questo file.
 - InputFormat: formato di input associato a questa query. L'adapter utilizza InputFormat per estrarre i dati dal record di origine.
 - SleepTime: intervallo, in secondi, in cui l'adapter non è attivo, in modo da attendere l'arrivo di nuovi dati.
 - Critical: (facoltativo) [yes/no]. Se impostato su "yes", l'adapter si interrompe immediatamente quando rileva un errore nella query. L'opzione "no" viene utilizzata quando si tratta di un errore noto risolto senza modificare il file di configurazione.
 - TransactionMode: (facoltativo) [implicito/esplicito]. Se impostato su "explicit" (esplicito), l'adapter avvia una transazione di database prima della query. Questa opzione risolve diversi problemi nelle query complesse e di grandi dimensioni.
 - RollbackSegmentName: (facoltativo). Definisce un determinato segmento di rollback utilizzato dall'adapter. Altrimenti, il database utilizza il segmento di rollback.
- SelectStatement: include l'istruzione selezionata da eseguire sul database di origine.
 - AutoCompleteQuery: (facoltativo) [yes/no]. Se impostato su "yes", l'adapter concatena automaticamente un'istruzione where alla query specificata come segue:

- Creazione di un'istruzione where che acquisisce solo i nuovi valori in base ai campi chiave.
- Ordinamento dell'istruzione dei risultati in base ai campi chiave.
- QueryKeyFields: definisce i campi per avviare il prossimo recupero dei dati:
 - KeyField:
 - Name: nome del campo ottenuto, dai campi della query.
 - Sort: (facoltativo) [crescente/decrescente]. Metodo di ordinamento dei dati (crescente o decrescente).
 - Function: (facoltativo). Utilizzare questo attributo se è necessario eseguire una funzione speciale su questo campo, per combinare il valore del campo nell'uso della funzione (:fieldname). Ad esempio, utilizzando la funzione di data Oracle con un nome di campo Ts-Function="to_date('ts','dd/mm/yyyy')"
 - ValueLeftWrapper: (facoltativo). Utilizzare questo attributo per concatenare i caratteri prima del valore del campo. Il valore predefinito per i campi di stringa e di data è ' (apostrofo).
 - ValueRightWrapper: (facoltativo). Utilizzare questo attributo per concatenare i caratteri dopo il valore del campo. Il valore predefinito per i campi di stringa e di data è ' (apostrofo).
 - NameLeftWrapper: (facoltativo). Utilizzare questo attributo per concatenare i caratteri prima del nome del campo. Il valore predefinito è una stringa null.
 - NameRightWrapper: (facoltativo). Utilizzare questo attributo per concatenare i caratteri dopo il nome del campo. Il valore predefinito è una stringa null.
 - IsPrimaryKey: (facoltativo) [yes/no]. Se impostato su "no", la chiave non viene utilizzata dall'adapter nell'istruzione where durante la query.
 - DefaultInitialValue: (facoltativo). Se la query SelectInitialValues non restituisce i record, l'adapter ricava il valore predefinito da qui. Se sono presenti alcuni campi chiave, tutti i campi chiave devono avere questo attributo.
 - SelectInitialValues: istruzione select che fornisce i valori iniziali per la prima istruzione where (quando il file di controllo è vuoto).

Nota: l'ordine dei valori deve essere lo stesso negli elementi Field per questo attributo QueryKeyFields e restituire un risultato per ogni campo.

Sezione InputFormatCollection

Questa sezione specifica la struttura dei dati recuperati dall'origine dati, la procedura per dividere una riga di dati in campi e quali sono i tipi di campo e i formati. In questa sezione è possibile eseguire il filtraggio dei dati iniziali e modifiche sui dati utilizzando rispettivamente i campi InputFormatSwitch e Compound.

Di seguito è riportato il flusso di lavoro generale di questa sezione:

- Viene trovata la corrispondenza della riga dei dati con uno o più InputFormats.
- I dati vengono divisi in campi in base alla specifica InputFormat corrispondente.
- I campi composti sono valori assegnati con la combinazione e suddivisione dei campi di dati.
- I dati elaborati vengono controllati in base alle condizioni TranslatorSwitch.
- I dati elaborati vengono inviati al convertitore corrispondente oppure ignorati.

Il nodo InputFormatCollection può contenere uno o più nodi InputFormat.

Struttura XML:

```
<InputFormatCollection>
  <InputFormat InputFormatName="MyInputFormat">
    <InputFormatFields>
      <InputFormatField Name="sid_id" Type="string"/>
      <InputFormatField Name="content" Type="string"/>
      <InputFormatField Name="date" Type="time"
TimeFormat="%d/%m/%Y %H:%M:%S"/>
      <InputFormatField Name="server" Type="string"
Source="compound">
        <Compound>
          <Segment SourceField="content"
RegularExpression=".*Job server: ([^\n]+).*" />
        </Compound>
      </InputFormatField>
    </InputFormatFields>
  <TranslatorSwitch DefaultTranslator="GeoTranslator">
    <TranslatorCase TranslatorName="NonGeoTranslator" Break="yes">
      <Condition SourceField="routing_info" Operator="EQ"
Value="c nano"/>
    </TranslatorCase>
  </TranslatorSwitch>
</InputFormat>
</InputFormatCollection>
```

- InputFormat:
 - InputFormatName: qualsiasi nome per questo formato, cui fare riferimento nella sezione DataSourceInterface.

- RequiredFields: (facoltativo) numero minimo di campi previsti in una riga di dati. Una riga contenente un numero inferiore di campi viene ignorata e l'errore viene registrato.
- InputFormatFields: può contenere uno o più nodi di campo in base al numero di campi di input nell'origine dati.
 - InputFormatField: indica un campo di dati della riga di dati originali o un campo composto.

```
<InputFormatField Name="timestamp" Type="time"  
TimeFormat="%d/%m/%Y %H:%M:%S"/>
```
 - Name: nome per questo campo da usare come riferimento per altri elementi.
 - Type: tipo di dati del campo (stringa, numero intero, reale, ora)
 - Source: (facoltativo) il valore predefinito è "event" (evento), i possibili valori sono:
 - event: valore di campo ottenuto da un evento proveniente dall'origine dati; i valori dei campi vengono acquisiti nello stesso ordine di arrivo dall'origine dati.
 - compound: il campo è composto. Ottiene il valore dopo aver apportato alcune modifiche su altri valori dei campi o costanti.
 - title: valore di campo ottenuto dai nomi di campo titolo. Il campo di riferimento deve essere già definito.
 - filename: valore di campo ottenuto dal nome del file dell'origine dati; solo per adapter per file di testo.
 - constant: valore di campo costante, acquisito dalla proprietà ConstantValue che deve essere visualizzata di seguito.
 - FieldTitleName: quando l'origine è il titolo, specifica il titolo del campo da utilizzare. È necessario definire il campo di origine prima.
 - ConstantValue: quando l'origine è costante, specifica il valore da corrispondere. Quando il campo è di tipo ora, il valore costante è una stringa formattata in base a TimeFormat oppure "Now" o "NowUtc", in cui "Now" è l'ora corrente nelle impostazioni internazionali dell'origine dati e "NowUtc" è l'ora corrente in UTC.
 - TimeFormat: formato di data utilizzato per l'analisi del campo. I seguenti codici di carattere possono essere utilizzati per il Formato data e ora:

Simbolo	Descrizione
%d	Giorno 1-31
%a	Nome abbreviato del giorno della settimana (lun, mar...)
%A	Nome completo del giorno della settimana (lunedì ...)
%m	Mese 1-12
%b	Nome abbreviato del mese (gen, feb...)
%B	Nome completo del mese (gennaio, febbraio...)
%y	Anno (2 cifre)
%Y	Anno (4 cifre)
%H	Ora in formato di 24 ore (00-23)
%I	Ora in formato di 12 ore (01-12)
%p	Indicatore AM/PM per il formato a 12 ore
%M	Minuti 0-59
%S	Secondi 0-59
%D	Data completa senza separatori (ggmmaaaa)
%T	Orario completo senza secondi né separatori (hhmm)
%n	Numero di secondi da 1970
%E	Data e ora in formato esadecimale
%%	Segno di percentuale

- DecimalSymbol: (facoltativo) simbolo decimale per i campi. Se non specificato, DefaultDecimalSymbol viene ricavato dalla sezione General.
- DigitGroupingSymbol: (facoltativo) simbolo di raggruppamento cifre predefinito per i campi di tipo numero intero e reale. Se non specificato, DefaultDigitGroupingSymbol viene ricavato dalla sezione General.
- Compound: obbligatorio quando source=compound. Specifica le modifiche di campo da raccogliere in un campo composto.
 - Segment: specifica una modifica di campo da aggiungere al campo composto creato. Solo l'attributo SourceField è obbligatorio.
 - SourceField: campo su cui basarsi. Il campo di riferimento deve essere già definito.

- **RegularExpression**: espressione regolare di modifica.
 - **MatchCase**: (facoltativo) [yes/no] definisce se nella corrispondenza con l'espressione regolare è presente la distinzione tra maiuscole e minuscole.
 - **SelectionStart**: posizione di inizio dell'estrazione di testo, a partire da 0.
 - **SelectionLength**: dimensioni dell'estrazione di testo.
 - **Prefix**: stringa aggiunta come prefisso al risultato della modifica.
 - **Suffix**: stringa aggiunta come suffisso al risultato della modifica.
 - **XpathExpression**: espressione Xpath di modifica.
- **InputFormatSwitch**: utilizzato per specificare i criteri di formato, quando le righe di dati arrivano in più formati.

Nota: durante l'utilizzo di **InputFormatSwitch**, l'ordine dei nodi **InputFormat** è importante; **InputFormat** di riferimento deve essere già definito.

DefaultInputFormat: specifica il nome di **InputFormat** cui indirizzare qualora nessun criterio sia soddisfatto

- **InputFormatCase**: indica un criterio da verificare sulle righe di dati per determinare a quali **InputFormat** devono essere indirizzati.
- **InputFormatName**: **InputFormat** da utilizzare quando il criterio viene soddisfatto.
- **LogicOperator**: (facoltativo) [e/o].
- **and**: è necessario soddisfare tutte le condizioni (impostazione predefinita).
- **or**: è necessario soddisfare almeno una condizione
- **Condition**: condizione da verificare su una riga di dati per determinarne il formato.

SourceField: campo da verificare.

Operator: tipo di testo delle seguenti opzioni:

- **EQ**: uguale a
- **NE**: diverso da
- **GT**: maggiore di
- **LT**: minore di
- **GE**: maggiore o uguale a
- **LE**: minore o uguale a
- **MATCH**: espressione regolare da soddisfare
- **UNMATCH**: espressione regolare da non soddisfare

ValueType: (facoltativo) [costante/campo/previousValue]

- **constant**: il contenuto dell'attributo **Value** è costante a prescindere dai dati di origine

- field: il contenuto dell'attributo Value è il nome di campo dallo stesso record.
- previousValue: il contenuto dell'attributo Value è il nome di campo dal record precedente nella stessa query con lo stesso formato di input.

Value: valore da soddisfare o espressione regolare.

MatchCase: (facoltativo) [yes/no] definisce se nella verifica esegue la distinzione tra maiuscole e minuscole. Se impostato su "yes", i due valori vengono convertiti in lettere minuscole prima della verifica.

- TranslatorSwitch: determina quale convertitore verrà utilizzato per convertire la riga di dati in un evento unificato CA Business Service Insight.
 - DefaultTranslator: convertitore da utilizzare qualora nessun criterio sia soddisfatto. Se il valore è "Ignore" (ignora), non viene utilizzato alcun convertitore e la riga viene ignorata.
 - TranslatorCase: specifica i criteri da verificare sui dati elaborati per determinare a quale convertitore devono essere indirizzati.

Break [yes|no]

yes: (impostazione predefinita) se i criteri sono soddisfatti, non verificare i criteri successivi.

no: in ogni caso, dopo la valutazione dei criteri e l'applicazione del convertitore se soddisfatti, procedere con i criteri successivi.

LogicOperator: (facoltativo) [and/or].

- and: è necessario soddisfare tutte le condizioni (impostazione predefinita).
- or: è necessario soddisfare almeno una condizione.

TranslatorName: convertitore cui reindirizzare se le condizioni vengono soddisfatte.

Condition: condizione da verificare sui dati elaborati per determinare se è stato utilizzato su di esso il convertitore pertinente. È uguale alla condizione in InputFormatSwitch.

Sezione TranslationTableCollection

La presente sezione contiene le tabelle di mapping dei valori delle origini dati nei campi evento di CA Business Service Insight.

Struttura XML:

```
<TranslationTableCollection
  LoadingMode="remote"
  TranslationTablesFileName="Translations.xml">
  <TranslationTable
    Name="ResourcesTranslateTable"
    DestinationType="resource">
    <TranslationField>nodeName</TranslationField>
  </TranslationTable>
  <TranslationTable
    Name="EventTypesTranslateTable"
    DestinationType="event_type">
    <TranslationField>eventType</TranslationField>
  </TranslationTable>
  <TranslationTable
    Name="valueUpDownTranslateTable"
    DestinationType="value"
    ValueType="string">
    <TranslationField>eventType</TranslationField>
  </TranslationTable>
</TranslationTableCollection>
```

- TranslationTablesFileName: (facoltativo) specifica il nome del file in cui sono memorizzate le tabelle a livello locale; se non specificato, viene utilizzato il valore predefinito ("Translation.XML").

- LoadingMode: (facoltativo) [standalone, remote].

Nota: il valore predefinito per l'interfaccia in linea è "remote" (remota), mentre per l'interfaccia non in linea è "standalone" (autonoma). Di seguito è riportato il metodo di caricamento delle tabelle di conversione:

- standalone: l'adapter carica le tabelle di conversione in locale. Nessuna connessione con il server CA Business Service Insight in relazione alla conversione. Le modifiche nelle tabelle di conversione vengono archiviate solo nel file locale.
- remote: l'adapter invia una richiesta di caricamento di tutte le tabelle dal server CA Business Service Insight. Le modifiche effettuate nelle tabelle di conversione verranno archiviate anche in locale.
- LoadTimeout: (facoltativo) se la modalità di caricamento è remota, è possibile specificare un timeout in secondi.
 - TranslationTable: collega il valore dell'evento alla tabella di mapping.

- Name: nome che verrà utilizzato e indicato dal convertitore. I nomi validi iniziano con una lettera o un carattere di sottolineatura e contengono lettere, numeri e caratteri di sottolineatura.
- DestinationType: [resource, event_type, contract_party, service, time_zone, value].
- ValueType: (facoltativo) [numero intero, reale, stringa] tipo di valore restituito dalla tabella. I valori di tipo stringa e reale sono validi solo per la tabella con DestinationType="value"
- TranslationField: nome del campo da cui eseguire la conversione; il nome del campo viene ottenuto dai campi del formato di input. È possibile avere fino a 5 campi.

Sezione TranslatorCollection

La sezione TranslatorCollection descrive come convertire i record dell'origine dati analizzati e modificati estratti nella sezione precedente in un evento CA Business Service Insight.

Quando la modalità dell'interfaccia è "online" (in linea), l'evento CA Business Service Insight dispone di una struttura unificata che contiene i seguenti campi:

- **Timestamp:** ora in cui si è verificato l'evento.
- **ResourceId:** ID risorsa CA Business Service Insight associato all'evento (la risorsa che è stata misurata, ecc.).
- **EventTypeId:** ID del tipo di evento CA Business Service Insight associato all'evento che descrive il tipo di evento (il tipo di misurazione applicato alla risorsa, tipo di azione di ticket, ecc.).
- **DataSourceID:** (facoltativo) nome dell'origine dati di input (nome file/nome tabella...).
- **Value:** il valore dell'evento (risultato della misurazione, numero di ticket, ecc.). Questo campo può essere visualizzato più di una volta.

Quando la modalità interfaccia è "offline" (non in linea), non sono presenti limitazioni sul numero dei campi o sul relativo nome.

Struttura XML:

```
<TranslatorCollection>
  <Translator TranslatorName="events" OnDuplication = "ignore">
    <TranslatorFields>
      <TranslatorField Name="ResourceId" SourceType="table"
        SourceName="ResourcesTranslateTable" Iskey="yes"/>
      <TranslatorField Name="EventTypeId" SourceType="constant"
        ConstantValue="1002" Iskey="yes"/>
      <TranslatorField Name="Timestamp" SourceType="field"
        SourceName="timestamp" Iskey="yes"/>
      <TranslatorField Name="Value" SourceType="table"
        SourceName="valueUpDownTranslateTable" Iskey="yes"/>
      <TranslatorField Name="Value" SourceType="field"
        SourceName="nodeName" Iskey="yes"/>
      <TranslatorField Name="Value" SourceType="constant"
        Type="integer" ConstantValue="1000" Iskey="yes"/>
      <TranslatorField Name="Value" SourceType="field"
        SourceName="timestamp" TimeShift="-3600"
        TimeShiftFieldName="createDate" Iskey="yes"/>
      <TranslatorField Name="Value" SourceType="lookup"
        SourceName="ServiceTable" LookupValue="word"
        Iskey="yes"/>
    </TranslatorFields>
  </Translator>
</TranslatorCollection>
```

</TranslatorCollection>

- Translator: descrive come convertire il set di campi ricevuti nell'evento di output.
 - TranslatorName: nome utilizzato da InputFormat per inviare i gruppi di campo a tale convertitore.
 - OnDuplication: membro che contiene il valore "ignore", "add", "update" o "updateAlways" per determinare le azioni da svolgere con l'evento di duplicazione (consultare la sezione Univocità dell'evento)
 - TranslatorFields: contiene un elenco di elementi TranslatorField, ognuno dei quali contiene i seguenti attributi:
 - Name: nome di campo. Nell'interfaccia in linea deve essere Timestamp, ResourceId, EventTypeId, Value o DataSourceId.
 - SourceType:
 - field: il valore di questo campo è ottenuto dal campo nel formato di input. L'attributo SourceName contiene il nome del campo.
 - table: il valore del campo è ottenuto dalla tabella di conversione. L'attributo SourceName contiene il nome della tabella.
 - lookup: il valore del campo è ottenuto dalla tabella di conversione. L'attributo SourceName contiene il nome della tabella. Il valore da convertire è ottenuto dall'attributo LookupValue e non dal formato di input.
 - constant: il valore del campo è costante e il relativo valore è nell'attributo ConstantValue.
 - SourceName: contiene il nome del campo o della tabella di conversione.
 - Type: [numero intero/reale/stringa/ora] richiesto solo quando il tipo di campo non è predefinito (dal nome di campo o da SourceType). Nell'interfaccia in linea è necessaria solo per il campo Value con SourceType=constant. Nell'interfaccia non in linea è necessario per ogni campo con SourceType=constant.
 - IsKey: rappresenta la chiave univoca dell'evento. Questa chiave è costituita da campi, che sono stati contrassegnati come TranslatorFields?IsKey = "yes". Consultare la sezione Univocità dell'evento.
 - LookupValue: contiene il valore di ricerca quando SourceType="lookup".
 - ConstantValue: contiene il valore costante quando SourceType=constant. Quando il campo è di tipo ora, il valore costante è una stringa formattata in base a TimeFormat oppure "Now" o "NowUtc", in cui "Now" è l'ora corrente nelle impostazioni internazionali dell'origine dati e "NowUtc" è l'ora corrente in UTC.
 - TimeFormat: contiene TimeFormat, obbligatorio solo per i campi con SourceType=constant e Type= time.
 - TimeShift: definisce le ore di differenza in secondi, solo per i campi di ora.

- TimeShiftFieldName: (facoltativo) contiene un nome di campo, dal formato di input, che contiene le ore di differenza in secondi. TimeShift e TimeShiftFieldName possono essere utilizzati insieme.

Appendice D: Definizione delle formule di business logic (esperto di business logic)

Questa sezione contiene i seguenti argomenti:

[Azioni da evitare durante la creazione di formule di business logic](#) (a pagina 345)

[Metriche di gruppo ed efficacia delle risorse](#) (a pagina 345)

Azioni da evitare durante la creazione di formule di business logic

Quando si crea una formula di business logic, ricordare sempre di:

- non assegnare mai il valore null a una variabile globale. L'assegnazione di un valore null potrebbe causare il malfunzionamento di PSLWriter durante il calcolo della metrica. Se è necessario un valore non inizializzato, utilizzare invece Vuoto.
- Evitare di utilizzare mappe e oggetti vettoriali in Metriche di gruppo. Se questi oggetti devono essere utilizzati, ridurre al massimo le loro dimensioni. Le metriche di gruppo contenenti mappe e vettori di grandi dimensioni richiedono molto tempo per il calcolo.

Metriche di gruppo ed efficacia delle risorse

Che cos'è una metrica di gruppo?

La metrica di gruppo consente di definire una metrica da eseguire per ogni singolo membro di un gruppo di risorse, di applicare la stessa definizione e logica a un insieme di elementi. Un raggruppamento può essere impostato in modo statico su un insieme predefinito di risorse o in modo dinamico sui membri del gruppo di risorse mentre il gruppo può essere modificato nel tempo e includere o escludere i membri dal gruppo.

Una risorsa o un gruppo di risorse può essere escluso e incluso nel gruppo nel corso del tempo e anche escluso dal gruppo e nuovamente incluso nel gruppo, per varie volte, durante lo stesso periodo di calcolo (giorno, mese, anno e così via).

Che cosa avviene nella business logic quando un elemento di gruppo viene rimosso dal gruppo di base?

Per l'elemento di gruppo vengono attivati il metodo OnPeriodEnd e la funzione Result. Se avviene durante il periodo di calcolo, il risultato viene scritto solo nel database al termine del periodo di calcolo (ad esempio, alla fine del mese, alla fine dell'anno).

Che cosa avviene nella business logic quando un elemento di gruppo viene unito al gruppo di base?

Per l'elemento di gruppo vengono iniziate le variabili globali e attivati i metodi OnLoad, OnRegistration e OnPeriodStart

Che cosa avviene nella business logic quando un elemento di gruppo viene unito al gruppo di base dopo esser stato rimosso dal gruppo nello stesso periodo di calcolo?

Il risultato impostato per il periodo in cui l'elemento di gruppo è stato parte del gruppo viene sovrascritto dal nuovo risultato. In altre parole, il risultato al termine del periodo di calcolo fa riferimento solo all'ultima fase del periodo calcolato durante il quale l'elemento di gruppo è parte del gruppo.

In che modo l'attributo in vigore di una risorsa influisce sulla business logic?

Durante il periodo in cui una risorsa non è in vigore, non vengono raccolti dati non elaborati per la risorsa non in vigore.

In che modo l'attributo in vigore di una risorsa influisce sul raggruppamento?

La modifica di una risorsa per renderla non in vigore ha sul raggruppamento lo stesso impatto dell'esclusione della risorsa dal gruppo. Il raggruppamento si comporta in modo analogo per l'efficacia e l'appartenenza al gruppo della risorsa.

Come è possibile implementare le eccezioni su una risorsa? Utilizzare l'efficacia della risorsa è il metodo corretto?

Vi sono alcuni casi di business in cui è necessario impostare un periodo di eccezione su una singola risorsa, ad esempio, un server potrebbe essere sottoposto a manutenzione e deve essere ignorato nei calcoli durante il periodo di manutenzione.

Poiché la business logic ignora gli eventi di dati non elaborati di una risorsa non in vigore, è possibile scegliere di implementare le eccezioni su una risorsa utilizzando il meccanismo di efficacia. Potrebbe essere utile per alcuni casi di utilizzo. Tuttavia, se la risorsa è parte di una metrica di gruppo e la risorsa diventa in vigore e non in vigore nello stesso periodo di calcolo, nel risultato verrà preso in considerazione solo l'ultimo periodo in cui la risorsa era in vigore, come indicato in precedenza. In tal caso, è consigliabile utilizzare la funzione degli attributi personalizzati. È possibile gestire un attributo aggiuntivo per la risorsa con l'indicazione di stato della risorsa. La formula di business logic quindi interroga lo stato della risorsa quando è pertinente nello script.

Glossario

Adapter

Interfaccia tra CA Business Service Insight e le origini dati di terze parti. Gli adapter convertono i dati acquisiti da queste origini dati in un formato che possa essere utilizzato per i calcoli del livello di servizio forniti ai contraenti.

Agente

Oggetto che rappresenta una metrica in una data unità di tempo.

Allocazione risorse

L'allocazione di una risorsa a un servizio indirizza il flusso di eventi della risorsa a tale servizio. Una risorsa può essere assegnata a un servizio, un contraente, un tipo o un gruppo.

Audit Trail

Record cronologico delle attività di sistema e utente di CA Business Service Insight, che vengono salvate nel sistema. L'audit può essere esaminato in un secondo momento per identificare le azioni degli utenti nel sistema o nei processi che si sono verificati sul sistema.

Audit trail del contratto

Registro di tutte le attività di un contratto.

Autore del contratto

Utente responsabile della creazione di un determinato contratto.

Booklet

File basato su RTF che potrebbe includere dati di contratto e report correlati in un comodo formato utilizzando modelli di progettazione predefiniti e configurabili.

Cartella dei report

La cartella dei report è un contenitore utilizzato per raggruppare i report correlati l'uno all'altro in qualche modo.

Catalogo dei servizi

Insieme di tutte le informazioni relative ai servizi, i domini, le unità, la libreria dei modelli e i modelli del livello di servizio in CA Business Service Insight.

Categoria di dominio

Aspetto specifico del livello di servizio concordato nel contratto. Ad esempio, è possibile avere un dominio di servizio chiamato Help Desk e una categoria di dominio definita come Response Time of the Help Desk, che descrive tale aspetto particolare del servizio di help desk. In genere, le categorie di dominio vengono definite dall'amministratore di sistema del provider di servizi. CA Business Service Insight fornisce inoltre categorie di dominio predefinite.

Commento manuale relativo alla causa

Commento, impostato manualmente mediante il visualizzatore di report, per illustrare o aggiungere ulteriori informazioni sul risultato di un livello di servizio per una metrica specifica. I commenti manuali relativi alla causa possono essere condivisi da tutti i visualizzatori di report della stessa metrica.

Commento relativo alla causa

Insieme di commenti nella business logic per spiegare i risultati del livello di servizio. I commenti relativi alla causa sono associati alle metriche.

Consumo

Tipo di metrica che calcola il consumo. Utilizzata principalmente per il modulo finanziario. Utilizzando questo tipo di metrica, è possibile visualizzare nei report il consumo e i prezzi separatamente. Il consumo può essere calcolato anche nella metrica Elemento di costo.

Contraente

Cliente o provider con cui è stato sottoscritto il contratto. Un contraente può rappresentare anche un'entità interna in un'organizzazione più ampia.

Contratto archiviato

Contratto i cui dati sono stati spostati in archivio. I contratti archiviati non sono inclusi nei calcoli e non possono essere visualizzati nei report.

Contratto di fornitura

Contratto di fornitura. Un contratto con un fornitore esterno che riguarda la fornitura di servizi che supportano l'organizzazione nella fornitura di servizi a loro volta.

Conversioni

Conversione dei dati raccolti dagli adapter dalle origini dati in entità che sono state definite in CA Business Service Insight.

Dashboard

Interfaccia utente per i gestori di alto livello che organizza e presenta le informazioni e i report in tempo reale in un modo semplice da leggere.

Data di inserimento della risorsa

Data in cui la risorsa viene inserita in CA Business Service Insight. Da non confondere con la data di validità della risorsa.

Data di validità della risorsa

Data a partire dalla quale CA Business Service Insight può utilizzare la risorsa. La data di validità della risorsa viene definita dalla versione della risorsa che include la risorsa. La risorsa non verrà riconosciuta dalla versione di risorsa che presenta una data di validità precedente alla versione di risorsa in cui la risorsa è stata creata.

Dichiarazione dell'obiettivo

Descrizione logica di una metrica contenente i parametri che definiscono la metrica. Le dichiarazioni dell'obiettivo sono visualizzate nella GUI di CA Business Service Insight per acquisire in modo efficace l'essenza della metrica.

Dispositivo di notifica

Dispositivi nella rete che ricevono i messaggi di notifica tramite posta elettronica, popup o sms, oppure tramite un'applicazione di terze parti.

Eccezione

Periodo di tempo che non viene considerato per il calcolo dei livelli di servizio. Ad esempio, i periodi di inattività.

Elemento di costo

Metrica di calcolo dei prezzi. I prezzi possono essere basati sul consumo o definito come prezzi fissi.

Eventi ricevuti

Elenco di eventi ricevuti da altre metriche visualizzato nella finestra Ambito di business logic facendo clic sulla scheda Eventi ricevuti.

Evento di dati

Eventi generati dagli adapter di CA Business Service Insight

Evento di dati non elaborati

Evento generato da dati non elaborati in CA Business Service Insight.

Evento di metrica

Evento generato da una metrica in CA Business Service Insight.

Granularità

La granularità determina le unità di tempo aggiuntive da cui l'autore della metrica desidera ricavare risultati per escludere il periodo di riferimento della metrica.

Gruppo di contraenti

Gruppo dei contraenti (clienti) definito in modo logico. Il riferimento a un gruppo invece che a un singolo contraente consente di ottimizzare la creazione del contratto. Un contraente può appartenere a più gruppi di contraenti.

Gruppo di modifiche

Insieme di modifiche alla mappa di risorse del provider di servizi.

Gruppo di risorse

Insieme di risorse raggruppate come un'unità logica. Un gruppo di risorse può contenere una o più risorse singole o un gruppo di risorse.

KPI

Key Performance Indicator. Misura significativa utilizzata singolarmente o in combinazione con altri indicatori chiave di prestazione, per monitorare il livello di conseguimento degli obiettivi quantificabili di una società o un servizio.

KQI

Key Quality Indicator. Misura significativa utilizzata singolarmente o in combinazione con altri indicatori chiave di prestazione, per monitorare il livello di conseguimento degli obiettivi di qualità di una società o un servizio.

Layout di dashboard

Layout del dashboard. La disposizione dei widget nel dashboard.

Metrica

Combinazione di vari parametri che definiscono il livello di servizio di destinazione per un determinato servizio in un determinato momento. Ciascuna metrica è associata a un singolo dominio del servizio. I campi e gli attributi di metrica includono la categoria di dominio, la formula del livello di servizio, il servizio, il periodo di applicazione, l'obiettivo del livello di servizio e il periodo di riferimento.

Metrica correlata

Metrica cui viene fatto riferimento come destinazione in una relazione.

Metrica derivata

Metrica creata da un modello di contratto o da un modello del livello di servizio.

Metrica di consumo

Metrica che fornisce la possibilità di visualizzare e i prezzi separatamente in un report.

Metrica di gruppo

Tipo di metrica applicata a più risorse o gruppi di risorse.

Metrica informativa

Metrica che esegue il calcolo informativo da utilizzare solo per le operazioni con i report.

Metrica intermedia

Metrica in grado di generare eventi per lo scopo del calcolo degli eventi. Le metriche intermedie non possono essere calcolate.

Modello del livello di servizio

Insieme predefinito di servizi e metriche, utilizzato per facilitare la creazione e la modifica dei contratti in base a processi di business comuni.

Modello di business logic

Le formule predefinite di business logic che possono essere utilizzate per definire nuove formule di business logic.

Modello di contratto

Contratto predefinito utilizzato per creare un nuovo contratto.

Moduli di business logic

Libreria di funzionalità script che può essere utilizzata in business logic.

Modulo di calcolo dello stato corrente

Processo autonomo che calcola le metriche di stato corrente. Non esistono limiti per il numero di istanze consentite per l'esecuzione su uno o più computer.

Note dell'evento

Informazioni aggiuntive su un evento specifico. Le note dell'evento vengono impostate automaticamente o manualmente (nei report).

Notifica

Notifiche agli utenti sugli eventi eseguiti nel sistema. Le notifiche consentono agli utenti di adottare azioni correttive per evitare la deviazione dai contratti, di incorrere in penali e così via.

OLA

Accordo sui Livelli Operativi. Un accordo sui livelli operativi è un accordo sui livelli di servizio (SLA) in cui il fornitore è l'organizzazione interna e i clienti sono le unità di business interne.

Opzioni internazionali

Dettagli specifici alla località del contraente. I parametri includono la lingua, la valuta, la differenza di tempo rispetto a GMT, l'ora legale e il formato di data.

Pacchetto

Raccolta di modelli del livello di servizio e modelli assemblati insieme da installare e decomprimere in un'altra istanza CA Business Service Insight.

Parametri di metrica

Parametri di una metrica.

Parametro

Valore che può essere impostato al di fuori della business logic per influire sulla formula di business logic. I parametri vengono utilizzati dalla business logic per determinare il risultato del livello di servizio. Un parametro può essere di tipo testo, elenco, numero, data o tabella. Sono esempi di parametri le soglie e i nomi delle risorse.

Parametro contrattuale

Parametro di un contratto standard, di un modello di contratto e di un modello del livello di servizio utilizzato dalle metriche nella business logic.

Penale

Risarcimento dovuto al contraente se non viene soddisfatto l'obiettivo del livello di servizio entro un determinato periodo di riferimento. Le penali possono essere attivate o disattivate in CA Business Service Insight.

Periodo di riferimento

Intervallo di tempo durante il quale CA Business Service Insight misura il livello di servizio fornito rispetto all'obiettivo del livello di servizio definito nel contratto. La misurazione eseguita durante questo periodo determina se è presente una deviazione rispetto all'obiettivo stabilito (come definito dalla business logic) e se sono stati applicati bonus o penali. Il periodo di business determina inoltre le modalità di generazione dei report.

Procedura guidata di creazione report

Interfaccia grafica utente (GUI) utilizzata per definire i parametri del report.

Procedura guidata di impostazione metriche

Semplice interfaccia sviluppata per consentire agli utenti di modificare le metriche alla prima aggiunta della metrica al contratto da un modello del livello di servizio.

Profilo di notifica

Insieme di parametri che definisce le condizioni in base alle quali si attiva l'invio di un messaggio di notifica, gli utenti cui viene inviato e le modalità di invio.

Raggruppamento di report

Report in cui più report sono affiancati in una pagina.

Rapporto

Tipo di unità.

Relazione

Entità che descrive una connessione di un determinato tipo stabilita tra due metriche e che presenta proprietà specifiche.

Report composto

Più report regolari che vengono visualizzati come più serie in un singolo diagramma.

Report di raggruppamento

Report che comprende più report standard, in posizione affiancata.

Report in formato libero

Report creato con un'istruzione SQL definita dall'utente in base al database di CA Business Service Insight o altri database esterni.

Report semplice

Mezzo per analizzare i risultati calcolati da CA Business Service Insight in base a un ampio set di criteri.

Risorsa

Singolo elemento dell'infrastruttura totale del provider di servizi. Gli esempi di risorse includono singoli server, switch, hub, router, l'help desk o qualsiasi altro elemento misurabile. Possono essere definite più risorse come parte di un gruppo di risorse, che viene quindi trattato come un'altra risorsa.

Ruolo

Insieme di responsabilità, attività e autorizzazioni. Il ruolo dell'utente definisce la modalità di visualizzazione di CA Business Service Insight. A ogni utente di CA Business Service Insight viene assegnato uno o più ruoli, che determinano quali azioni l'utente può eseguire o no in CA Business Service Insight. Le azioni che non possono essere eseguite per il ruolo vengono rimosse dall'interfaccia utente CA Business Service Insight quando l'utente accede all'applicazione.

Script di conversione

Script che può essere utilizzato per facilitare la manutenzione delle conversioni ed evitare errori.

SLO

Obiettivo del livello di servizio. Utilizzato per misurare gli accordi contrattuali.

Tabella di conversione

Meccanismo utilizzato per la conversione dei valori di dati provenienti da dati non elaborati in dati definiti nel sistema.

Tipi di evento

Un evento è una misurazione eseguita su una risorsa con strumenti di misurazione di terze parti e quindi convertito dall'adapter in un formato utilizzabile da CA Business Service Insight. I tipi di evento determinano il modo in cui gli eventi vengono definiti e formattati per CA Business Service Insight.

Tipo di metrica

Descrive il motivo per il calcolo di una determinata metrica e definisce l'elenco dei campi e il rispettivo comportamento in termini di campi obbligatori e valori predefiniti che una metrica di un determinato tipo deve presentare.

Tipo di risorsa

Categoria integrata di risorse. Una risorsa deve essere assegnata ad almeno un tipo di risorsa.

Unità

Catalogo di unità misurabili. Gli esempi includono la percentuale e la valuta.

Unità di misura

Unità di misura per tutte le metriche definite per una categoria di dominio.

Voce di conversione

Rappresenta una definizione dei valori di origine e di destinazione utilizzati dalla tabella di conversione.