CA Business Service Insight

Implementierungshandbuch

8.2.5

Diese Dokumentation, die eingebettete Hilfesysteme und elektronisch verteilte Materialien beinhaltet (im Folgenden als "Dokumentation" bezeichnet), dient ausschließlich zu Informationszwecken des Nutzers und kann von CA jederzeit geändert oder zurückgenommen werden.

Diese Dokumentation darf ohne vorherige schriftliche Genehmigung von CA weder vollständig noch auszugsweise kopiert, übertragen, vervielfältigt, veröffentlicht, geändert oder dupliziert werden. Diese Dokumentation enthält vertrauliche und firmeneigene Informationen von CA und darf vom Nutzer nicht weitergegeben oder zu anderen Zwecken verwendet werden als zu denen, die (i) in einer separaten Vereinbarung zwischen dem Nutzer und CA über die Verwendung der CA-Software, auf die sich die Dokumentation bezieht, zugelassen sind, oder die (ii) in einer separaten Vertraulichkeitsvereinbarung zwischen dem Nutzer und CA festgehalten wurden.

Ungeachtet der oben genannten Bestimmungen ist der Benutzer, der über eine Lizenz für das bzw. die in dieser Dokumentation berücksichtigten Software-Produkt(e) verfügt, berechtigt, eine angemessene Anzahl an Kopien dieser Dokumentation zum eigenen innerbetrieblichen Gebrauch im Zusammenhang mit der betreffenden Software auszudrucken, vorausgesetzt, dass jedes Exemplar diesen Urheberrechtsvermerk und sonstige Hinweise von CA enthält.

Dieses Recht zum Drucken oder anderweitigen Anfertigen einer Kopie der Dokumentation beschränkt sich auf den Zeitraum der vollen Wirksamkeit der Produktlizenz. Sollte die Lizenz aus irgendeinem Grund enden, bestätigt der Lizenznehmer gegenüber CA schriftlich, dass alle Kopien oder Teilkopien der Dokumentation an CA zurückgegeben oder vernichtet worden sind.

SOWEIT NACH ANWENDBAREM RECHT ERLAUBT, STELLT CA DIESE DOKUMENTATION IM VORLIEGENDEN ZUSTAND OHNE JEGLICHE GEWÄHRLEISTUNG ZUR VERFÜGUNG; DAZU GEHÖREN INSBESONDERE STILLSCHWEIGENDE GEWÄHRLEISTUNGEN DER MARKTTAUGLICHKEIT, DER EIGNUNG FÜR EINEN BESTIMMTEN ZWECK UND DER NICHTVERLETZUNG VON RECHTEN. IN KEINEM FALL HAFTET CA GEGENÜBER IHNEN ODER DRITTEN GEGENÜBER FÜR VERLUSTE ODER UNMITTELBARE ODER MITTELBARE SCHÄDEN, DIE AUS DER NUTZUNG DIESER DOKUMENTATION ENTSTEHEN; DAZU GEHÖREN INSBESONDERE ENTGANGENE GEWINNE, VERLORENGEGANGENE INVESTITIONEN, BETRIEBSUNTERBRECHUNG, VERLUST VON GOODWILL ODER DATENVERLUST, SELBST WENN CA ÜBER DIE MÖGLICHKEIT DIESES VERLUSTES ODER SCHADENS INFORMIERT WURDE.

Die Verwendung aller in der Dokumentation aufgeführten Software-Produkte unterliegt den entsprechenden Lizenzvereinbarungen, und diese werden durch die Bedingungen dieser rechtlichen Hinweise in keiner Weise verändert.

Diese Dokumentation wurde von CA hergestellt.

Zur Verfügung gestellt mit „Restricted Rights“ (eingeschränkten Rechten) geliefert. Die Verwendung, Duplizierung oder Veröffentlichung durch die US-Regierung unterliegt den in FAR, Absätze 12.212, 52.227-14 und 52.227-19(c)(1) bis (2) und DFARS, Absatz 252.227-7014(b)(3) festgelegten Einschränkungen, soweit anwendbar, oder deren Nachfolgebestimmungen.

Copyright © 2013 CA. Alle Rechte vorbehalten. Alle Marken, Produktnamen, Dienstleistungsmarken oder Logos, auf die hier verwiesen wird, sind Eigentum der entsprechenden Rechtsinhaber.

Technischer Support – Kontaktinformationen

Wenn Sie technische Unterstützung für dieses Produkt benötigen, wenden Sie sich an den Technischen Support unter http://www.ca.com/worldwide. Dort finden Sie eine Liste mit Standorten und Telefonnummern sowie Informationen zu den Bürozeiten.

Inhalt

Kapitel 1: Einführung 9

Rollen 9
- Servicekatalog-Manager 10
- Vertragsmanager 10
- Business-Logik-Experte 11
- Datenexperte 11
- Systemadministrator 12

Lösungsvorgang 13
- Planung 14
- Entwurf 15
- Implementierung 16
- Installation und Bereitstellung 16

Kapitel 2: Planung und Entwurf 17

Sitzung zur Anforderungserfassung (Alle Rollen) 18
Erfassen von Beispieldaten (Datenexperte) 20
Design 21
- Entwurfeinführung (Vertragsmanager, Business-Logik-Experte, Datenexperte) 22
- Vertragsmodellierung (Vertragsmanager) 23
- Vertragsmodellierungsphasen-Ausgaben (Vertragsmanager, Datenexperte) 43
- Daten-Modellierung (Datenexperte, Business-Logik-Experte) 44
- Datenmodellierungsphasen-Ausgaben (Datenexperte und Business-Logik-Experte) 60

Kapitel 3: Implementierung 61

Implementierung - Einführung 61
Den Rahmen aufstellen (Vertragsmanager) 64
Einrichten der Vorlagenbibliothek (Vertragsmanager) 64
Wie erstellt man Verträge (Vertragsmanager) 65
- Erstellen von Verträgen aus einem Service 67
- Erstellen von Service Level-Vorlagen 68
- Vertragslebens-Zyklus und Versionierung 69

Datenerfassung (Datenexperte) 71
- Adapter-Funktionalität 72
- Adapter-Umgebung 73
- Hauptdateien 76
- Arbeitsdateien 77

Adapter-Kommunikation 83
Adapter-Registrierungseinstellungen 86
Ausführungsmodi des Adapters 88
Konfigurations- und Wartungstools 90
Konfigurieren von Adaptern und Übersetzungen 90
Automatische Übersetzungen mit Übersetzungsskripten 129
Erweiterte Themen der Adapter-Funktion 130
Business-Logik-Skripting (Business-Logik-Experte) 138
Workflow beim Business-Logik-Skripting 140
Business-Logik-Module 141
Innerhalb der Business-Logik 142
Aktivieren der Verträge (Vertragsmanager) 183
Volle Neuberechnung von Vertragsmetriken 185
Erstellen von lieferbaren Ergebnissen (Vertragsmanager) 186
Definieren von Sicherheitseinstellungen (Administrator) 186
Erstellen von Berichten 187
Konfigurieren von Dashboard-Seiten 200
Hinzufügen von Service Level-Alarmprofilen 203

Kapitel 4: Installation und Bereitstellung 207

Einführung 208
Vorbereitungen 210
Installation 212
Importieren oder Exportieren zwischen Umgebungen (Datenexperte) 213
Integration - Mail-Servereinrichtung (Systemadministrator) 215
Festlegen von Systemvoreinstellungen (Systemadministrator) 216
Sicherheitseinstellungen (Systemadministrator) 216
Angeben von Einstellungen für SSA-Synchronisation 217

Anhang A: Beispiele für Servicedomänen und Domänenkategorien 219

Anhang B: Beispiele für Fallstudien 221

Vertragsbildungsbeispiele 221
Fallstudie 1: Systemverfügbarkeit 222
Fallstudie 2: Systemverfügbarkeit 2 223
Fallstudie 3: System-Reaktionszeit 225
Fallstudie 4: Helpdesk-Leistung 229
Fallstudie 5: Systemsicherung 231
Beispiel für eine Finanzmetrikmodellierung 232
Fallstudie 6: Modellieren von Finanzbedingungen eines Vertrags/Services 232

Beispiele für die Datenmodellierung 240
Fallstudie 7: Serverleistung 240
Fallstudie 8: Helpdesk-Leistung 243
Beispiel für die Verwendung von anwenderspezifischen Attributen 250
Fallstudie 9: Mehrere dynamische Ziele 250
Beispiele eines Übersetzungsskripts 254
Fallstudie 10: Grundlegende automatische Übersetzung 254
Fallstudie 11: Aktualisieren der Ressourcenmodelle 257
Beispiele für das Business-Logik-Skripting 260
Fallstudie 12: Verwenden der Zähler-Logik, um die Anzahl der Fehler zu berechnen 261
Fallstudie 13: Umgang mit dynamischen Komponentengruppen 266
Fallstudie 14: Umgang mit der Zeitakkumulationsuhr 271
Beispiele für das Schreiben einer effektiven Business-Logik 277
Fallstudie 15: Geclusterte Metriken 280
Fallstudie 16: Designmuster der Business-Logik 284
Fallstudie 17: Integrierte Funktionalität 289
Fallstudie 18: Registrierung 292
Fallstudie 19: Adapterassistent für dateibasierte Datenquelle 295
Adaptererstellung 297
Einsatz von Adaptern 307
Adapterplanung 308
Fallstudie 21: LDAP-Integration – Beispiel 311
Fallstudie 22: Mit PERL generierte Berichte 318

Anhang C: Eigenschaften der Adapterkonfiguration 321

Gesamtstruktur 321
Abschnitt "General" 322
CA Business Service Insight-Schnittstellenabschnitt 328
Abschnitt "DataSourceInterface" 331
Abschnitt zur SQL-Schnittstelle 334
InputFormatCollection-Abschnitt 338
Abschnitt "TranslationTableCollection" 343
Abschnitt "TranslatorCollection" 345

Anhang D: Definieren von Business-Logik-Formeln (Business-Logik-Experte) 349

Was beim Erstellen von Business-Logik-Formeln zu vermeiden ist 349
Geclusterte Metriken und Ressourceneffektivität 349

Terminologieglossar 353

Kapitel 1: Einführung

In diesem Handbuch wird der Implementierungsprozess sowie die dazugehörigen Rollen, Zuständigkeiten und Interaktionen erläutert, die in Planung und Entwurf, Implementierung, Installation und Bereitstellung von CA Business Service Insight involviert sind.

Dieses Kapitel enthält folgende Themen:

Rollen (siehe Seite 9)
Lösungsvorgang (siehe Seite 13)

Rollen

Eine Rolle ist eine Person, die eine Reihe von Aufgaben ausführt, die während eines normalen Implementierungsvorgangs ausgeführt werden. Das Wort "Rolle" kann sich auch auf die Reihe von Aufgaben selbst beziehen. CA Business Service Insight hat mehrere vordefinierten Rollen, die der Systemadministrator bearbeiten kann. Außerdem können neue Rollen und bestimmte Berechtigungen erstellt werden.

Die folgenden Rollen sind für die Implementierung erforderlich:

- Servicekatalog-Manager
- Vertragsmanager
- Business-Logik-Experte
- Datenexperte
- Systemadministrator

Die Zuständigkeiten und Qualifikationen jeder Rolle werden unten beschrieben.

Hinweis: Die gleiche Person kann einige vom Systemadministrator definierte Rollen ausführen.

Servicekatalog-Manager

Zuständigkeiten:

- Verwalten des organisatorischen Servicekatalogs
- Definieren des organisatorischen Serviceangebotskatalogs
- Verwalten der Anforderungen für Vertragsdefinitionen und die Berichterstellung

Erforderliche Qualifikationen:

- Vertrautheit mit dem Servicebereitstellungsprozess E2E der Organisation
- Vertrautheit mit CA Business Service Insight-Konzepten

Vertragsmanager

Zuständigkeiten:

- Chatten Sie mit dem Endanwender.
- Verantwortung für den Aufbau des Servicekatalogs gemäß den definierten Anforderungen
- Erstellen neuer Verträge und Verwalten von vorhandenen Verträgen
- (Optional, jedoch empfohlen) Verantwortung für die spezielle Kunden-SLA-Implementierung

Qualifikationen:

- Gutes Verständnis für Prinzipien und Logik der Metriken
- Fachanwender von Funktionen auf CA Business Service Insight-GUI-Basis

Business-Logik-Experte

Zuständigkeiten:

- Implementieren von Metriken
- Schreiben von Business-Logik-Skripten und Verwalten von vorhandenen Business-Logik-Skripten

Qualifikationen:

- Grundlegender Entwicklungshintergrund und sehr aktives Wissen in Bezug auf Skriptsprachen, wie VB-Skript
- Gutes Verständnis für den CA Business Service Insight-Datenfluss
- Experte auf dem Gebiet CA Business Service Insight-Business-Logik-Skripting
- Gutes Verständnis für die CA Business Service Insight-Architektur und -Komponenten
- Fachanwender von Funktionen auf CA Business Service Insight-GUI-Basis

Datenexperte

Zuständigkeiten:

- Entwerfen der CA Business Service Insight-Infrastruktur
- Erstellen neuer Adapter und Verwalten von vorhandenen Adaptern, Verbinden mit Betriebsinfrastruktur, um Messungen zu erhalten
- Erstellen und Verwalten der CA Business Service Insight-Infrastruktur

Qualifikationen:

- Gutes Verständnis für die Umgebung und Struktur der Kundendatenquellen
- Grundkenntnisse in Bezug auf Datenbanken, XML und Scripting-Sprachen
- Gutes Verständnis für CA Business Service Insight-Datenfluss und Datenerfassungsprozesse
- Fachkenntnisse auf dem Gebiet CA Business Service Insight-Adapter
- Gutes Verständnis für CA Business Service Insight-Architektur und -Komponenten
- Fachanwender von Funktionen auf CA Business Service Insight-GUI-Basis
- Ein Entwicklungshintergrund ist von Vorteil

Systemadministrator

Zuständigkeiten:

- Verwalten von Installationen und Upgrades
- Erfüllen von Hardware- und Softwaresystemanforderungen
- Verwalten von Sicherheitseinstellungen: Anwender- und Rollendefinitionen
- Überwachen und Verwalten des Systems

Qualifikationen:

- Gutes Verständnis für Infrastruktur und Netzwerk der Kunden
- Gutes Verständnis für CA Business Service Insight-Architektur und -Komponenten
- Grundkenntnisse in Bezug auf DB, XML und Scripting-Sprachen
- Gutes Verständnis für CA Business Service Insight-Datenfluss
- Anwender von Funktionen auf CA Business Service Insight-GUI-Basis

Lösungsvorgang

Der Lösungsvorgang umfasst drei grundlegende Bereiche:

- Planung und Entwurf
- Implementierung
- Installation und Bereitstellung

Die zuvor beschriebenen Rollen sind - gemäß ihren Fachgebieten - in unterschiedliche Phasen des Implementierungsprozesses involviert. Diese Rollen interagieren miteinander, um den ganzen Prozess von Anfang bis Ende, von der Vertragsdefinition zum Ausgabebericht, abschließend durchzuführen.

Dieses Handbuch ist wie eine Rolle aufgebaut und prozessorientiert. Es basiert auf einem allgemeinen Phasenablauf, auf den der Implementierungsprozess aufbaut. Es beschreibt, wie jede Rolle eingebunden wird und wie die Rollen miteinander interagieren.

Die folgenden Abschnitte bieten eine Übersicht über die in den Implementierungsprozess involvierten Phasen und die Tätigkeiten einer jeden Rolle.

Das restliche Handbuch ist in Phasen untergliedert, wobei kenntlich gemacht ist, welche Rollen in welcher Phase involviert sind. Jedes Kapitel führt für jede Tätigkeit die dafür zuständige Rolle auf.

Nachfolgend finden Sie eine Kurzbeschreibung der grundlegenden Aufgaben einer jeden Phase sowie der Funktionen der unterschiedlichen Rollen. Weitere Einzelheiten sind in den jeweiligen Kapiteln aufgeführt.

Planung

Das Ziel der Planungsphase ist, alle Aspekte, die im Rahmen der Implementierung abzuwickeln sind, zu identifizieren und zu quantifizieren.

In dieser Phase erfasst der Vertragsmanager Unternehmensanforderungen vom Servicekatalog-Manager, um die Erwartungen von CA Business Service Insight nach einer erfolgreichen Implementierung zu verstehen. Es ist in dieser Phase sehr wichtig sowohl die aktuellen Anforderungen als auch künftige Pläne zu verstehen, um sicherzustellen, dass die Implementierung ein ungehindertes und reibungsloses Wachstum sowie eine einfache und problemlose Weiterentwicklung ermöglicht.

Gemäß den definierten Implementierungsanforderungen wird erwartet, dass der Vertragsmanager den bestehenden zugehörigen Prozess (sofern vorhanden) durch Prüfung der vorhandenen Prozesseingaben und -ausgaben evaluiert. In dieser Phase ist es notwendig, alle benötigten Informationsquellen zu identifizieren, und Proben, Verträge, Ausgabeberichte und Eingabequellen zu beziehen, um die vorhandenen Ausgaben berechnen zu können. Diese Informationen müssen sehr genau angegeben werden, damit der Vertragsmanager jede Eingabe identifizieren kann, so dass die erwartete Leistung hergeleitet werden kann.

In dieser Phase prüft der Vertragsmanager die Verträge, um sicherzustellen, dass der Servicekatalog und die Vorlagen, die als Teil der Implementierung bereitgestellt wurden, den bestehenden und den künftigen Bedarf unterstützen, und dass die aktuelle Implementierung alle Vertragsbereiche abdeckt.

Der Vertragsmanager prüft die Berichte und deren Formate, um sicherzustellen, dass alle festgelegten Messungen durchgeführt werden, um sie mit der vorhandenen Implementierung generieren zu können.

Der Datenexperte prüft dann die Eingabedatenmuster anhand der erforderlichen Messungen und Kalkulationen, die vom Vertragsmanager bereitgestellt werden. Dadurch kann der Datenexperte alle zu bewältigenden Eingabeprobleme identifizieren, wie anwenderspezifische Formate oder Mängel. Er kann dann entscheiden, ob zusätzliche Datenquellen benötigt werden.

Der Zweck dieser Untersuchung ist es, sicherzustellen, dass die Quellen die zur Berechnung der erforderlichen Metrik benötigten Daten enthalten, sowie die Anfangsdaten zum Retrieval-Prozess, wie Kommunikationsmethoden mit den Datenquellen und die Datenquellenstruktur.

Entwurf

In der Entwurfsphase werden die Eingaben und Ausgaben in die CA Business Service Insight-Modellstruktur aus Verträgen, Metriken und Ressourcen übersetzt. Das bedeutet, tatsächliche Daten so umzuwandeln und zu konzeptualisieren, dass sie in das CA Business Service Insight-Framework passen.

Das Systemdesign umfasst Folgendes:

Vertragsmodellierung

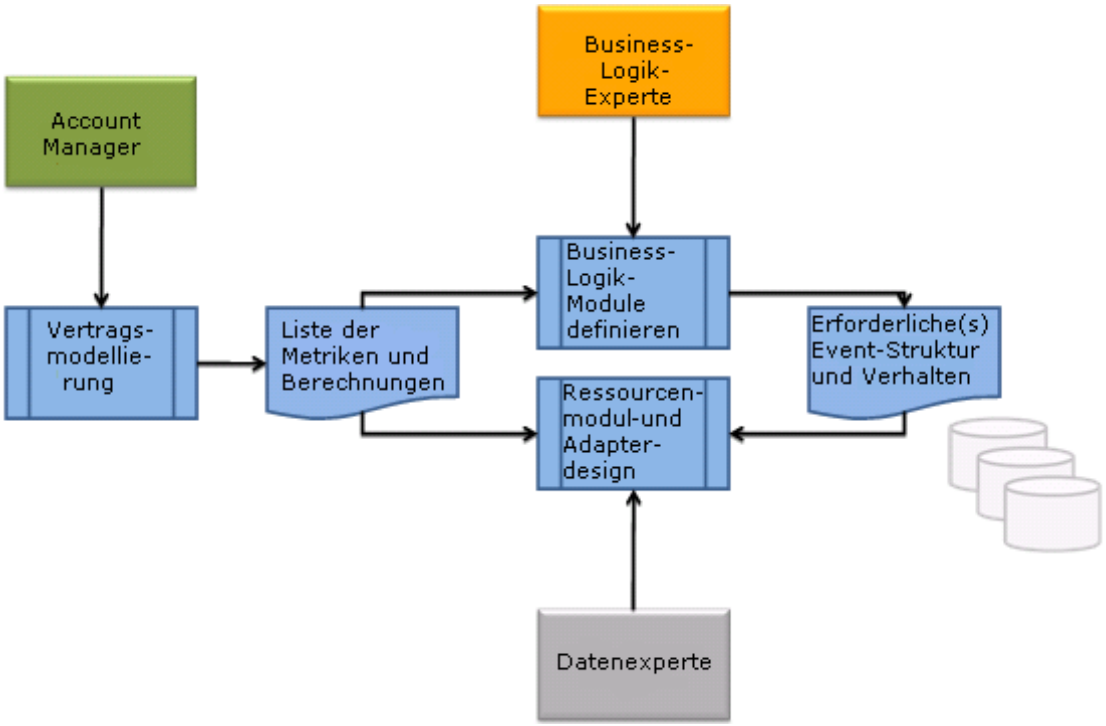
Kunden-SLAs werden als CA Business Service Insight-Verträge interpretiert und Servicekatalogelemente (wie der Vorlagenordner) werden definiert. Dies wird hauptsächlich vom Vertragsmanager ausgeführt.

Datenmodellierung

Ressourcendaten werden untersucht und in das CA Business Service Insight-Ressourcenmodell integriert. Dies wird vom Datenexperten und vom Business-Logik-Experten ausgeführt.

Möglicherweise gibt es mehrere Methoden für die Vertrags- oder Datenmodellierung. Oftmals gibt es allerdings eine optimale Vorgehensweise, bei der nicht nur das Problem gelöst wird, sondern die auch mehr Flexibilität oder Produktivität bietet und somit einen starken Kontext für ein weiteres Wachstum bildet.

Um dem Datenexperten bei der Auswahl der geeignetsten Methoden zu helfen, werden Fallstudien mit Lösungsvorschlägen verwendet.



Wie im vorherigen Workflow-Diagramm dargestellt, liefert der Vertragsmanager als Ergebnis des Vertragsmodellierungsprozesses eine Liste von Metriken, die im System und in ihrer Kalkulationsübersichtsdefinition konfiguriert werden sollten.

Beispiel:

Kunde A, durchschnittliche Reaktionszeit, CNP-System.

Die Liste der Metriken wird als Eingabe an den Business-Logik-Experten bereitgestellt. Sie definiert die erforderliche Event-Struktur und das Event-Verhalten, mit dem das erforderliche Kalkulationsskript festgelegt werden kann.

Die Liste der Metriken, die Event-Struktur und die Verhaltensanforderungen stellen die Eingabe vom Datenexperten für den Entwurf des Ressourcenmodells und der Datenadapter dar.

Implementierung

In der Implementierungsphase wird das CA Business Service Insight-System gemäß den Ergebnissen der Entwurfsphase konfiguriert. In der Implementierungsphase werden die theoretischen Entwurfsphasenergebnisse übernommen und zum Aufbau der betrieblichen Anforderungen für CA Business Service Insight verwendet.

Einige dieser Elemente, die erstellt oder bearbeitet werden müssen, sind:

Vertragsmanager

- Service-Setup
- Vertragserstellung
- Berichte und Dashboard-Aufbau

Datenexperte

- Adapterkonfiguration
- Einrichtung der Infrastruktur

Business-Logik-Experte

- Business-Logik-Module

Installation und Bereitstellung

Die Installations- und Bereitstellungsphase befasst sich mit der Installation und Integration des Produktionssystems, den Leistungstests und -überwachungen sowie der Anwenderschulung. Der Systemadministrator und der Datenexperte führen in dieser Phase die meisten Aktivitäten aus.

Kapitel 2: Planung und Entwurf

Dieses Kapitel enthält folgende Themen:

Sitzung zur Anforderungserfassung (Alle Rollen) (siehe Seite 18)
Erfassen von Beispieldaten (Datenexperte) (siehe Seite 20)
Design (siehe Seite 21)

Sitzung zur Anforderungserfassung (Alle Rollen)

Die Sitzung zur Anforderungserfassung ist der erste kritische Schritt in der Implementierung von CA Business Service Insight, und an ihr sollten alle an der Implementierung beteiligten Hauptmitarbeiter teilnehmen.

Die für diese Sitzung benötigten Informationen müssen von Mitarbeitern bereitgestellt werden, die mit den ausgewählten SLAs sehr vertraut sind. Dies umfasst die SLA-Business-Logik, die gegenwärtige Verwaltung der SLAs, alle Berichte, die gegenwärtig für sie generiert werden, und welche Berichterstellungsanforderungen zukünftig erwartet werden.

Folgende Mitarbeiter müssen an der Sitzung teilnehmen:

- Der Servicekatalog-Manager
- Der Vertragsmanager
- Der Datenexperte
- Der Business-Logik/Skript-Experte(n)
- Andere relevante Mitarbeiter, die mit den ausgewählten SLAs vertraut sind
- Der Projektmanager (sofern einer eingesetzt wurde)

CA empfiehlt Folgendes:

- An der Sitzung sollten unbedingt Mitarbeiter teilnehmen, die befugt sind, Entscheidungen über unklare Definitionen zu treffen, die bei der Prüfung der ausgewählten SLAs aufkommen können.
- Es ist ratsam, dem Implementierungsteam vor dieser Sitzung eine Kopie der für dieses Projekt ausgewählten SLAs zukommen zu lassen. Die Anwesenden sollten mit allen relevanten Datenquellen vertraut sein, die zu den ausgewählten SLAs gehören. Darüber hinaus sollten sie Datenauszüge dieser Datenquellen bereitstellen und nach Bedarf deren innere Struktur erklären können.

Die Hauptziele der Planungssitzung sind die Definition und die Festlegung folgender Punkte:

- Inhalt und Umfang der geplanten Arbeit
- Erfolgskriterien
- Rollen und Zuständigkeiten aller beteiligter Parteien
- Implementierungsprozess und -plan
- Erforderliche Hardware-/Softwareanforderungen

Dies wird erfüllt durch:

- Überprüfen der SLAs, die bei der Implementierung verwendet werden sollen
- Überprüfen der "Business-Logik" für jedes Ziel innerhalb der ausgewählten SLAs

- Überprüfen von Berichten, Alarmen und Dashboard-Anforderungen hinsichtlich der ausgewählten SLAs
- Überprüfen relevanter Datenquellen, die bei der Implementierung verwendet werden sollen
- Überprüfen der relevanten Infrastruktur-Topologie
- Erfassen relevanter Datenauszüge von relevanten Datenquellen
- Festlegen der Hauptmitarbeiter und deren Zuständigkeiten
- Definieren von Kommunikationspfaden für Dauer der Implementierung, Review-Meetings, den QS-Prozess uvm.
- Überprüfen des Implementierungprozesses und -plans
- Überprüfen der aktuellen Sachlage, d. h. wie diese SLAs verwaltet werden, und was fehlt
- Definieren der Erfolgskriterien für die Implementierung
- Definieren der Anforderungen an die benötigte Hardware/Software
- Definieren der Zeitlinie für die Implementierung

Erfassen von Beispieldaten (Datenexperte)

Für die standortferne, anfängliche Konfiguration des Adapters ist es aus Gründen der Datenerfassung wichtig, Proben ehemaliger Daten zu erhalten. Diese Proben müssen dieselbe Struktur aufweisen wie die Daten, die später beim eigentlichen System eingehen werden, von dem CA Business Service Insight Daten abrufen muss. Zusätzlich zu den Proben sollte sich der Datenexperte beim Datenquelleninhaber über die Datenquelle informieren, um Folgendes klären zu können:

Typ:

Datenbank, Datei, Stream oder andere.

Standort:

Wo ist es? Wie gelangt man dort hin (Verbindungsdetails)? Sicherheitshindernisse? Plattform?

Namenskonvention:

Wie sind Dateien benannt? (Handelt es sich um eine dateibasierte Datenquelle) Wie sind Tabellen benannt? Tabellenfeldnamen?

Verfügbarkeit:

Wann kann darauf zugegriffen werden? Wann sollte darauf zugegriffen werden (Auslastung)? Kontinuierliche Aktualisierung oder einmal pro Zeitraum von X? Wie lange sind sie persistent?

Verhalten:

Sammeln sich Daten an? Werden Daten überschrieben? Ist ein Archiv angelegt?

Spanne:

Wie viele historische Daten sind verfügbar?

Volumen:

Aktuelles Datenvolumen? Wachstumsrate? Prognosen?

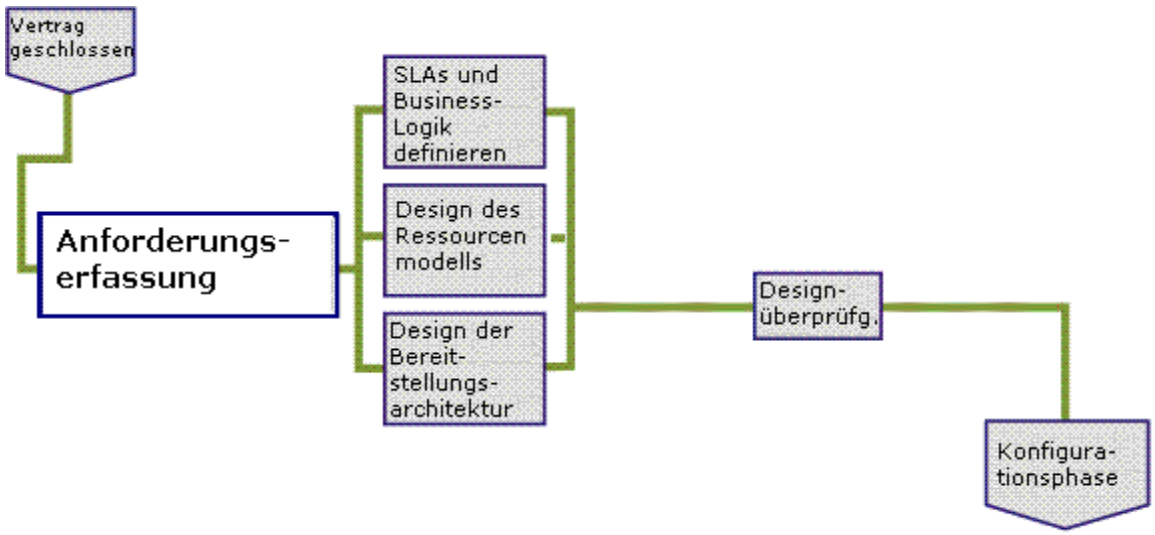
Struktur und Format:

Wie sind die Daten innerhalb Datenquelle organisiert? Welches sind die Datenfelder? Wie heißen die Tabellen? Was trennt die Datenfelder?

Streaming:

Erhält der Adapter Daten oder werden die Daten vom Adapter gesammelt?

Design


Vertrag geschlossen
Anforderungs-erfassung
SLAs und Business-Logik definieren
Design des Ressourcen modells
Design der Bereit-stellungs-architektur
Design-überprüfg.
Konfigura-tionsphase

Entwurfeinführung (Vertragsmanager, Business-Logik-Experte, Datenexperte)

In diesem Abschnitt werden der Prozess und die Argumentation hinter der Entwurfsphase des Lösungsprozesses erläutert. Die Entwurfsphase folgt der Planungsphase. Ihr folgt anschließend die Implementierungsphase, die im nächsten Kapitel beschrieben wird.

Ziel der Entwurfsphase ist es, das Implementierungsteam in die Lage zu versetzen, die Zuweisung von tatsächlichen Verträgen und deren Klauseln sowie von vorhandenen Leistungsdaten zum CA Business Service Insight-System abzuschließen.

Bevor mit diesem Prozess begonnen wird, muss das Implementierungsteam die verschiedenen verfügbaren Methoden und Optionen kennen, um sicherzustellen, dass nicht nur alle Anforderungen berücksichtigt werden, sondern auch, dass der resultierende Entwurf optimiert ist, während gleichzeitig künftiges Wachstum oder Änderungen ermöglicht werden.

Beim Entwurfsprozess muss das Implementierungsteam die folgenden Schritte ausführen:

- Prüfen der Verträge und Umwandeln in die notwendigen CA Business Service Insight-Objekte. Dies wird in diesem Handbuch als "Vertragsmodellierung" bezeichnet. Dafür zuständig ist der Vertragsmanager.
- Begutachten vorhandener Datensätze und die Entscheidung, welche Elemente auf welche Weise angesichts der gewünschten Ergebnisse extrahiert werden sollen. Dies wird in diesem Handbuch als "Datenmodellierung" bezeichnet. Dafür zuständig sind der Datenexperte und der Business-Logik-Experte.

Im Abschnitt über die Vertragsmodellierung wird Folgendes erläutert:

- Verwendete Terminologie (entscheidend für die ordnungsgemäße Implementierung)
- Business-Logik-Vorlagen und -Module
- Service Level-Vorlage und deren Verwendung
- Die Bedeutung der Erstellung eines soliden Servicekatalogs
- Finanzmanagement-Metrik (Vertragsstrafen, Bonus sowie Kosten und Beispiele), wie auf Kundenverträge und andere Vertragsmetriken angewendet

Im Abschnitt über die Datenmodellierung wird Folgendes erläutert:

- Ereignisse, die für CA Business Service Insight gelten, sowie deren Weg durch das CA Business Service Insight-System
- Metriken und wie sie registriert werden
- CA Business Service Insight-Ressourcen und wie sie identifiziert werden
- Vorschläge, wie die Erfassung und Definition dieser Ressourcen automatisiert werden können

Alle oben genannten Punkte werden in den folgenden Abschnitten weiterführend erläutert.

Es ist wichtig, sich bewusst zu sein, dass sich falsche Entscheidungen in dieser Phase ungünstig auf den Betrieb von CA Business Service Insight auswirken können und zu einem späteren Zeitpunkt nur schwer oder nicht mehr rückgängig zu machen sind.

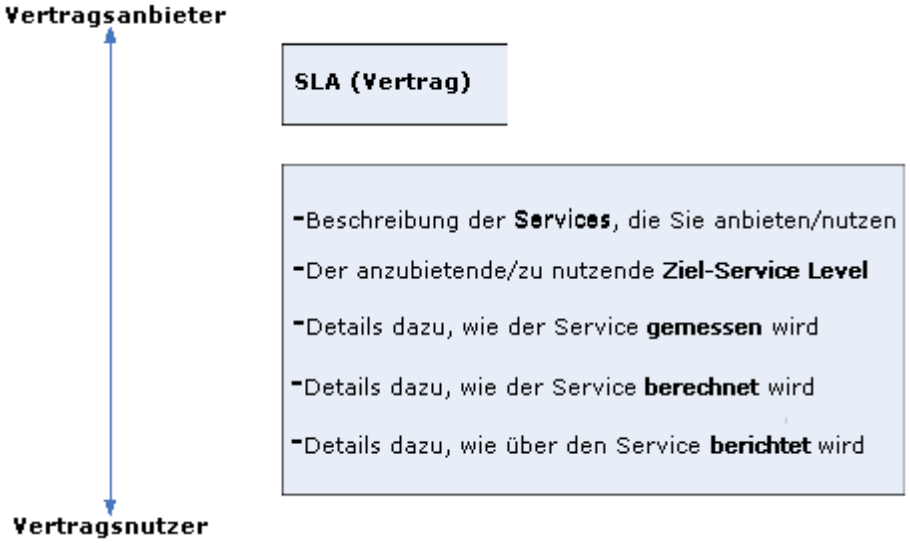
Vertragsmodellierung (Vertragsmanager)

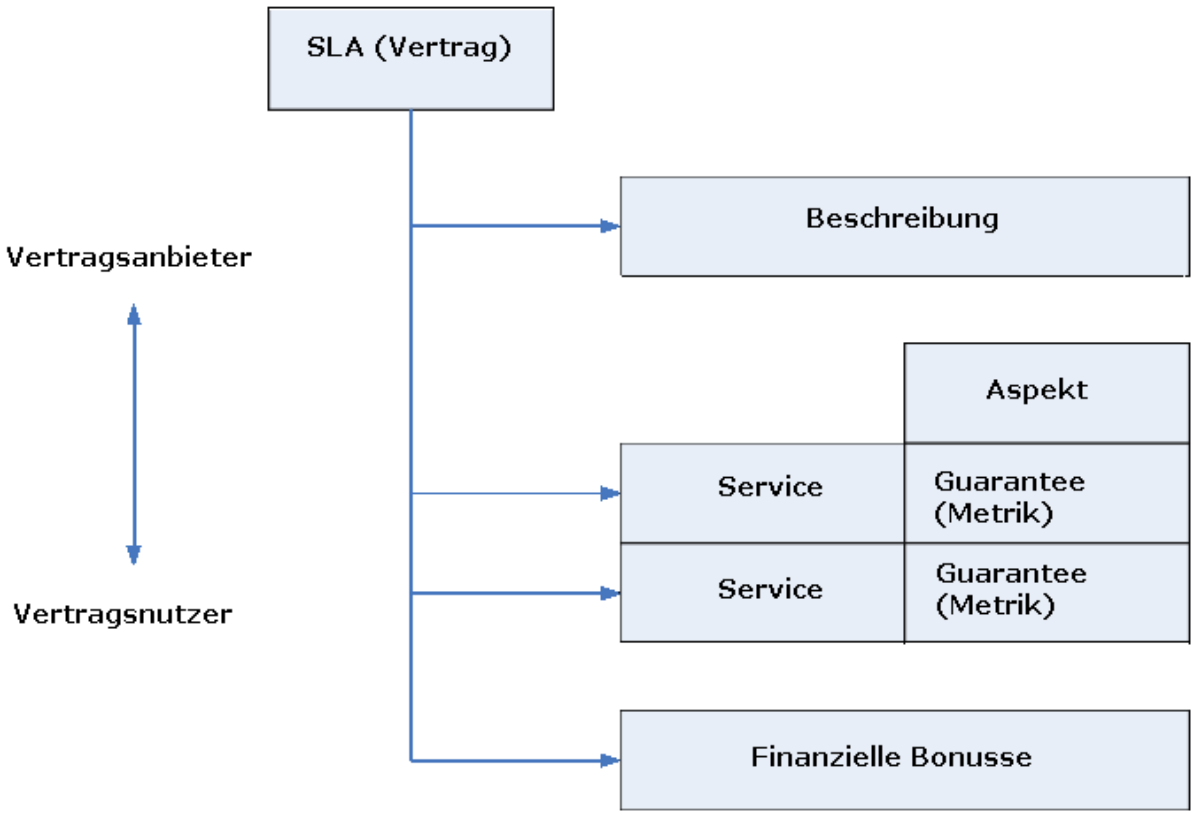
Die folgenden Aufgaben für die Vertragsmodellierung werden vom Vertragsmanager ausgeführt.

Vertragsterminologie

Ein Vertrag ist eine Zusammenstellung von Zielen. Ein Ziel ist entweder ein Vertrags- oder Betriebsziel (Metrik) oder ein Finanzziel (Bonus). Der Prozess der Vertragsmodellierung beinhaltet, den gesamten Vertragsinhalt des Implementierungsumfangs zu erfassen und in das CA Business Service Insight-Modell zu konvertieren.

Die folgenden Diagramme stellen diesen Prozess dar.

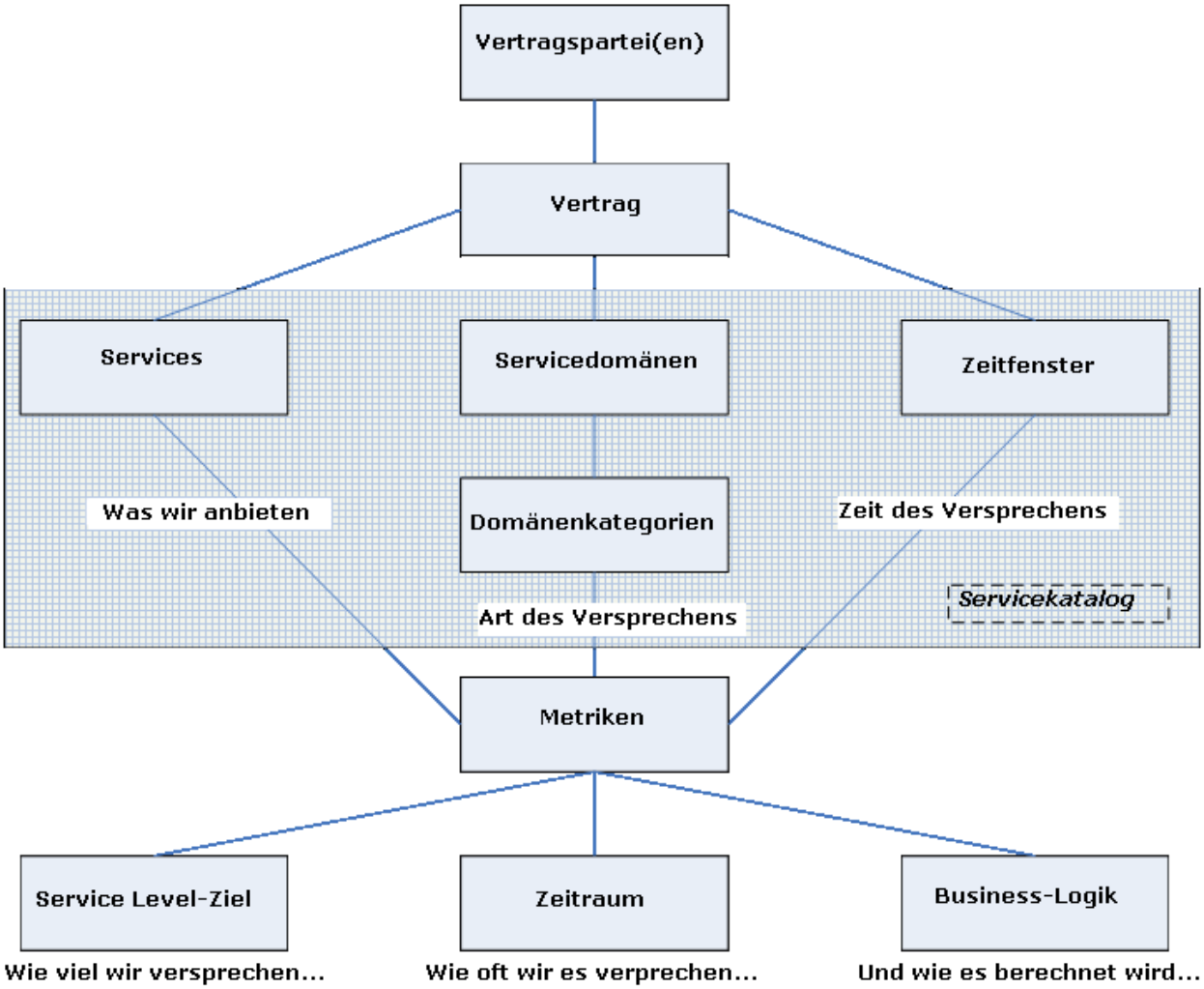




Hinweis: Mögliche künftige Pläne für das System müssen berücksichtigt werden, auch wenn nur Aspekte modelliert werden, die den Implementierungsumfang betreffen. Der Umfang muss allgemeine Anforderungen zur Vertragsverwaltung der Organisation beinhalten, um so ein Modell von ausreichender Bandbreite zu generieren, das künftige Entwicklungen einschließt. Dies minimiert und vereinfacht alle zukünftig erforderlichen Änderungen.

Das Konvertieren von Kundenverträgen in das CA Business Service Insight-Modell ist ein Prozess, bei dem der Vertragsmanager die innerhalb der üblichen Grenzen angebotenen Garantien (Metriken) in logische Gruppen, allgemeine Servicekomponenten, Vertragsaspekte usw. zusammenfasst. Diese logische Zusammenfassung ermöglicht sehr flexibles Service Level-Management.

Das CA Business Service Insight-Vertragsmodell umfasst die in der folgenden Abbildung dargestellten Entitäten:



Vertrag

Definiert die Vereinbarung und die Erfassung der Metriken. Bei dem Vertrag kann es sich, je nach Beziehung zwischen den beteiligten Vertragsparteien, um einen von drei Typen handeln. Es kann entweder ein SLA (Service Level Agreement zwischen der Organisation und ihren Kunden) sein, ein OLA (Operational Level Agreement zwischen den Abteilungen einer Organisation) oder ein UC (Underpinning Contract, ein Vertrag mit einer Drittpartei zwischen der Organisation und ihren Lieferanten, wobei der UC im Allgemeinen für einen Service gilt, den die Organisation über einen SLA für einen anderen Kunden leistet).

Vertragspartei(en)

Definiert den Kunden für die Serviceleistungen sowie die Partei, mit der er den Vertrag unterzeichnet hat. Eine einzelne Vertragspartei kann mehr als nur einen Vertrag abgeschlossen haben. Beachten Sie, dass Sie den Anbieter und den Empfänger der Serviceleistungen, die der jeweilige Vertrag umfasst, festlegen können.

Metriken

Definiert ein einzelnes Service Level-Ziel bzw. eine einzelne Garantie. Jede Metrik ist eine Anfrage für eine Messung, die das eigentliche Service Level-Resultat erzeugt, für das die Berichterstellung durchgeführt wird.

Eine Metrik umfasst die folgenden Elemente:

Typ

Bei einer Metrik kann es sich um einen der folgenden acht Typen handeln:

SLO	Service Level-Ziel
Informativ	Service Level-Ziel ohne ein Ziel
KPI	Leistungskennzahl (Key Performance Indicator), dient zur Unterstützung unterschiedlicher Industriekonzepte. Bei Telco ist die KPI=SLO und steht für die Kundenverpflichtung, während sie bei IT für eine technische Verpflichtung steht.
KQI	Qualitätskennzahl (Key Quality Indicator), eine umfassende Metrik zur Messung aggregierter Ergebnisse von unterschiedlichen Kennzahlen.
Interim	Interim-Messung, die in anderen Metriken eingesetzt wird. Für diese metrischen Ergebnisse ist keine Berichterstellung möglich.
Verbrauch	Wird von Finanzmetriken verwendet, um den Service-/Ressourcenverbrauch zu kalkulieren. Wird zusammen mit Preisobjekt-Metriken verwendet
Bonus	Finanzmetrik, wird für Bonusberechnungen (positiver Begriff für Vertragsstrafe) verwendet.
Preiselement	Finanzmetrik, wird zur Berechnung von Ergebnissen verwendet, die auf dem Verbrauch basieren. Unterschiedliche Preise pro Zeitraum oder pro Anzahl von Einheiten.

Kontrollzeitraum

Berichterstellungszeitraum des Vertrags oder Zeitraum, für den das tatsächlich berechnete Service Level-Ergebnis mit dem Ziel verglichen wird. Kontrollzeiträume in CA Business Service Insight können mit einer Granularität von Stunde, Tag, Woche, Monat, Quartal oder Jahr definiert werden. Für die Ursachenanalyse wird die Metrik zusätzlich zum Kontrollzeitraum auch in den anderen sechs Granularitäten berechnet, die Ergebnisse für diese Zeiträume werden jedoch nur als operative Ergebnisse gekennzeichnet.

Hinweis: Dies wird nur durchgeführt, wenn diese Kalkulationsgranularitäten für die Metrik festgelegt sind.

Zeitfenster

Zeit innerhalb des Kontrollzeitraums, während der die spezifische Garantie oder Verpflichtung anwendbar ist, einschließlich außerordentlicher Zeitfensterdefinitionen wie vorhergesagter Wartungsfenster, gesetzlicher Feiertage usw.

Business-Logik

Formel, die angibt, wie Rohdaten auszuwerten sind, um den tatsächlichen Service Level-Wert für diesen Serviceaspekt zu kalkulieren, sowie die spezifischen Prämissen, die bei der Kalkulation berücksichtigt werden müssen. Während der Entwurfsphase wird nur der Kalkulationsentwurf identifiziert. Er wird in der Konfigurationsphase detaillierter ausgearbeitet.

Ziel

Vertragliche Service Level-Verpflichtung. Das Ziel kann, muss jedoch nicht obligatorisch sein. Dies richtet sich nach dem jeweiligen Metriktyp. Wenn kein Ziel definiert ist, liefert die Metrik nur Ergebnisse zu Berichterstellungszwecken. Sie kann nicht mit einer vertraglichen Verpflichtung oder einer Garantie verglichen werden. Ziele können statisch oder dynamisch sein.

Hinweis: Nähere Erläuterungen zu einigen der bisher genannten Konzepte finden Sie unter Anhang B - Fallbeispiele (siehe Seite 221).

Jede Metrik bezieht sich auf die folgenden systemweiten Entitäten, die im Systemvorlagenordner enthalten sind:

Servicekomponente

Beschreibt, was verpflichtungsgemäß bereitgestellt werden muss.

Servicekomponentengruppe

Sammlung von Servicekomponenten. Eine einzelne Servicekomponente kann Bestandteil mehrerer Gruppen sein. Eine Servicekomponentengruppe ist eine optionale Entität und kann - sofern sie verwendet wird - größere Flexibilität für die Berichterstellung bieten.

Servicedomäne

Spezifischer Aspekt von Servicekomponenten, die zur Service Level-Überwachung gemessen werden müssen (beispielsweise Leistung oder Verfügbarkeit).

Domänenkategorie

Bestimmte Maßeinheit. Gibt die Standardmaßeinheit an, und ob das anvisierte Ziel ein Minimal- oder ein Maximalwert ist. Grundsätzlich ist dies die spezifische Dimension der Servicedomäne, die gemessen wird (d. h. Anteil der verfügbaren Zeit in %, Anzahl von Ausfällen, durchschnittliche Durchsatzrate usw.).

Jede der oben aufgeführten Entitäten sowie ihre Beziehungen zu allen Service Level-Anforderungen, die in CA Business Service Insight überwacht werden sollen, müssen während der Vertragsmodellierungsphase identifiziert werden.

Funktionsweise des Modellierungsprozesses

Während des Modellierungsprozesses müssen alle Metriken, die im System konfiguriert werden sollen, zusammen mit den zugehörigen Entitäten gemäß dem berücksichtigten Vertrag und den Berichterstellungsanforderungen definiert werden.

Vor oder während dieser Phase ist es üblich, sich auf einen Namensgebungsstandard für die Metrikbezeichnungen zu einigen, sodass das System übersichtlich und leicht zu navigieren ist. Bedenken Sie auch die Beschreibung der Metrik, die im Abschnitt "Zielvorgabe" der Metrik verwendet werden kann.

Der Prozess der Vertragsanalyse umfasst die folgenden Schritte:

1. Identifizieren der Vertragsziele

   Identifizieren Sie für jedes Ziel:

   - Einen geeigneten Namen (beachten Sie die Namensgebungsstandards)
   - Ziel (optional)
   - Kontrollzeitraum
   - Maßeinheit
   - Servicekomponente
   - Servicedomäne
   - Domänenkategorie (und Maßeinheit)
   - Zeitfenster (wann: 24 x 7, nur zu Geschäftszeiten?)
   - Kalkulationsentwurf (welche Variablen werden benötigt und wie wird der Service Level kalkuliert)

   Hinweis: Einige dieser Punkte sind möglicherweise nicht gleich im ersten Durchgang klar, sie lassen sich jedoch bei der Präzisierung des Servicekatalogs klären.

2. Identifizierung aller Finanzziele anhand des Vertrags und folgende Festlegungen für jedes Finanzziel:
   - Ist es eine Vertragsstrafe, ein Bonus oder ein Kostenziel?
   - Durch welche Bedingungen wird es ausgelöst?
   - Für welche Servicekomponenten trifft es zu?
   - Servicedomäne
   - Domänenkategorie (die Einheit kann hier eine Währung, ein Kostenbetrag, der Prozentsatz einer Zahlung usw. sein)

Sobald alle Ziele notiert und definiert wurden, sollten Sie die komplette Liste der Metriken überprüfen und versuchen, den Katalog zu optimieren/zu standardisieren.

Während dieses Schrittes ist es wichtig, sich zu vergewissern, dass alle Servicekomponenten, Servicedomänen und Domänenkategorien vernünftig definiert sind und dass sie einen klaren und präzisen Katalog dessen ergeben, was im System geboten wird. Dies schließt ALLE Metriken und Verträge im System ein. Dadurch wird im System ein äußerst solider Servicekatalog generiert, um so ein künftiges Systemwachstum zu ermöglichen.

Servicedomänen und Domänenkategorien sollten NICHT bis auf eine sehr niedrige granulare Ebene festgelegt werden. Sie sollten beschreibend sein, aber auf einer höheren Ebene als die Metrik.

Beispiel: Sie haben die drei folgenden Metriken:

- % von termingerechten Ausfallberichten
- % von termingerechten Ausnahmeberichten
- % von termingerechten Verwaltungsberichten

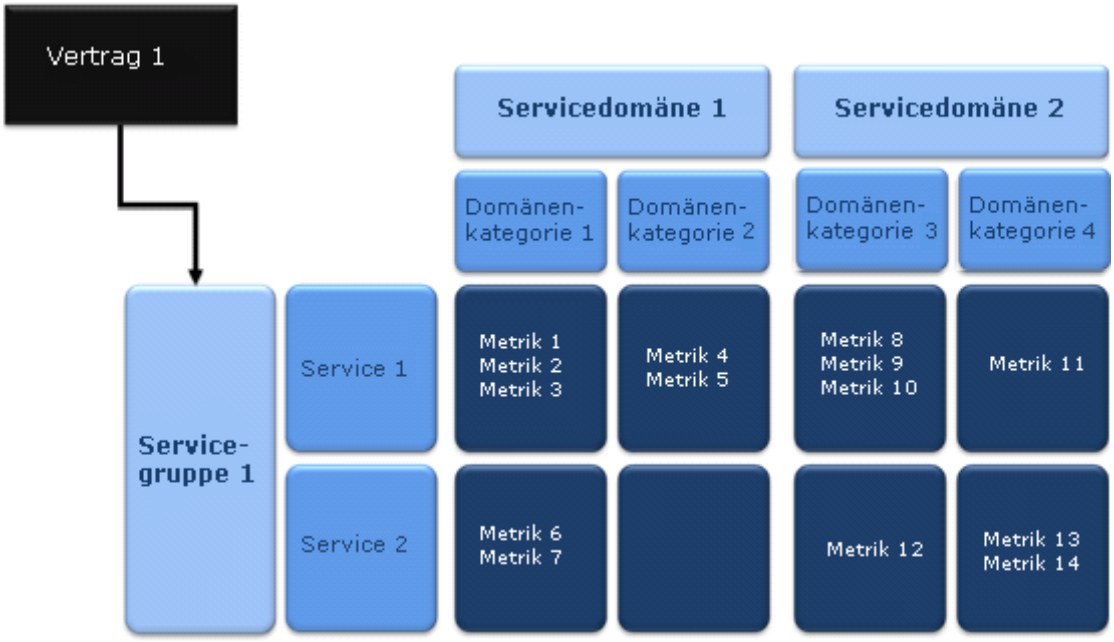
Die beste Kategorie, unter die alle drei Metriken wahrscheinlich fallen würden, ist die Servicedomäne Performance und die Domänenkategorie % Reports delivered on Time. Die Domänenkategorie sollte den Berichtstyp nicht erwähnen. Diese drei Metriken hätten wahrscheinlich dieselbe Berechnungsmethode und würden dieselbe Business-Logik verwenden (mit der möglichen Ausnahme eines einzelnen Parameters, um zwischen den unterschiedlichen Berichtstypen zu unterscheiden).

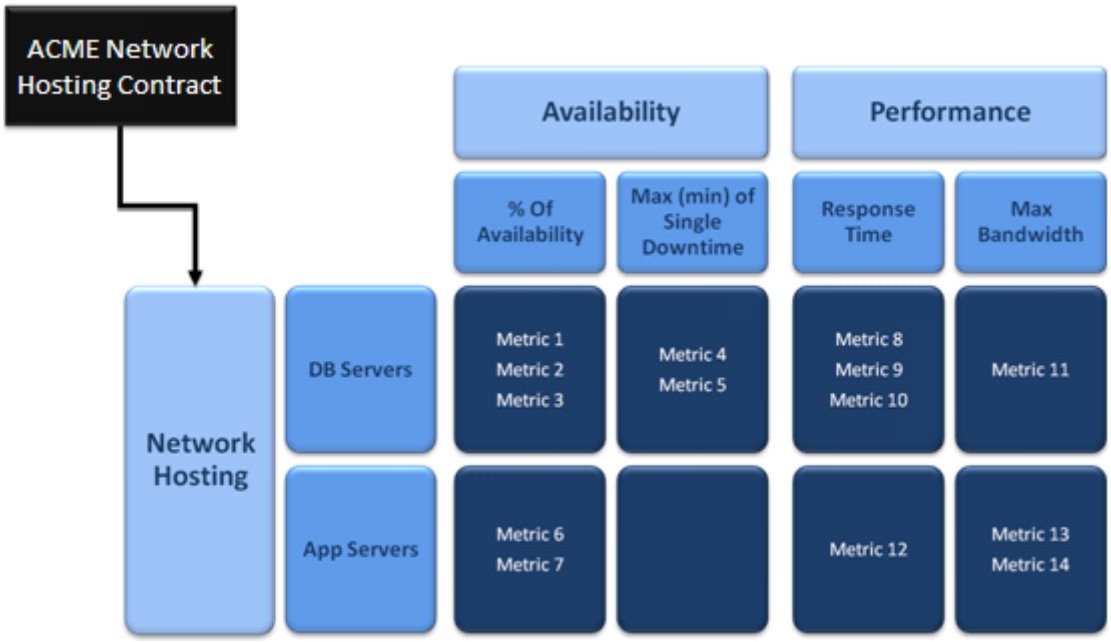
Geeignete Servicedomänen und Domänenkategorien können aus dem ITIL-Standard (Information Technology Infrastructure Library) übernommen werden. Dies sind jedoch nur Beispielvorschläge. Jede Organisation kann ihre eigenen definierten Standards dafür haben. Unter Anhang A - Beispiele zu Servicedomänen und Domänenkategorien (siehe Seite 219) finden Sie einige Vorschläge für Servicedomänen und Domänenkategorien.

Hinweis: Es kann an diesem Punkt nützlich sein, eine Sitzung mit allen wichtigen Projektbeteiligten abzuhalten, um sich von jedem bestätigen zu lassen, dass der gewählte Katalog ihre aktuellen und zukünftigen Bedürfnisse unterstützt.

Weitere Beispiele, die wichtige Punkte aufzeigen, sind in den Anhängen aufgeführt. In diesen Beispielen wird ein einzelnes Vertragsziel samt seiner Modellierung beschrieben. Wenn reale Situationen modelliert werden, müssen alle Ziele bekannt sein, sodass die Katalogentitäten alle Ziele in einer breiter gefächerten und umfassenden Weise repräsentieren.

Sobald der Prozess der Identifizierung aller Metriken sowie der zugehörigen Entitäten abgeschlossen ist, hat der Vertragsmanager eine Matrix, die die komplette Metrik beschreibt, wie im folgenden Diagramm dargestellt.





Zusätzliche Punkte, die es im Modellierungsprozess zu bedenken gilt, sind in den folgenden Abschnitten beschrieben.

Fragen für den Vertragsmanager

Nachfolgend sind die Fragen aufgeführt, die der Vertragsmanager bedenken muss, um sicherzustellen, dass alle Aspekte berücksichtigt wurden:

Woher weiß ich, dass ich die richtige Servicekomponente gewählt habe?

> Wenn die Servicekomponente auf mehr als einen Vertrag angewendet und in mehr als einem Aspekt gemessen werden kann, wurde sie richtig definiert.
>
> Zum Beispiel: Beim System X handelt es sich um ein System, das mehreren Kunden bereitgestellt wird und das anhand seiner Verfügbarkeit, Zuverlässigkeit, Leistung usw. gemessen werden kann.

Woher weiß ich, dass ich die richtige Servicedomäne gewählt habe?

> Wenn die Servicedomäne auf mehrere Arten gemessen oder kalkuliert werden kann, beschreibt sie einen allgemeinen Aspekt des bereitgestellten Service und ist daher die richtige Servicedomäne.
>
> Die Verfügbarkeit lässt sich beispielsweise auf verschiedene Arten messen, eine davon ist der Prozentsatz der verfügbaren Zeit. Andere Möglichkeiten können der Prozentsatz der Verfügbarkeit während oder außerhalb der Geschäftszeiten, die Ausfallhäufigkeit, die mittlere Zeit zwischen Ausfällen (MTBF), die mittlere Reparaturzeit, die maximale Ausfallzeit, die durchschnittliche Ausfallzeit, die Gesamt-Ausfallzeit usw. sein. Dies alles sind verschiedene Arten, um die Verfügbarkeit eines bestimmten Systems auszuwerten.

Beim Modellierungsprozess zu berücksichtigende Fälle

Nachfolgend sind einige Beispiele für häufige und auch speziellere Fälle aufgeführt, die Konzepte beschreiben, die im Modellierungsprozess beachtet werden sollten. Diese Konzepte können eine präzisere Definition der Metriken und ein stabileres Framework ergeben.

Metriken ohne definiertes Ziel

Da das Zielfeld innerhalb der Metrik-Definition nicht obligatorisch ist, sind Service Level-Berichte in Fällen, in denen das Zielfeld nicht definiert ist, für die Metrik verfügbar. Es sind jedoch keine Berichte über den Service Level im Vergleich zum Ziel und zur Abweichung möglich (da es kein Ziel gibt, mit dem der tatsächlich berechnete Service Level verglichen werden könnte). Diese Metriktypen werden für die Fälle definiert, in denen Berichte für Informationen erforderlich sind, die nicht Teil der eigentlichen vertraglichen Verpflichtungen sind.

Durch die Definition dieses Metrik-Typs erhält der Benutzer alle möglichen Drilldown-Berichterstellungsoptionen. Zugleich erhält der Service Level-Manager die Möglichkeit, die Messungen in jeder künftigen Phase auf ein Ziel anzuwenden.

***Zum Beispiel*:**

Die Vertragsgarantie lautet, 99 % der Netzwerkverfügbarkeit bereitzustellen und die Ausfallhäufigkeit pro Monat zu dokumentieren.

Es sind zwei Metriken definiert, eine Metrik wird mit dem Ziel 99 % Verfügbarkeit definiert, die andere Metrik wird für das Zählen der monatlichen Ausfallhäufigkeit ohne Ziel definiert. Die Berichterstellung ist für beide Metriken möglich, nur die erste verfügt jedoch aufgrund der vertraglichen Verpflichtung über Abweichungskalkulationen.

Hinweis: Eine andere mögliche Methode für das Bewältigen dieser Art von Situation besteht darin, Business-Logik-Ausgaben und Freitext-Berichterstellung für diese Daten zu verwenden. Allerdings verliert der Bericht dadurch die Möglichkeit zur Verfeinerung der Daten, wie auch die Möglichkeit, den einfachen Berichtsassistenten zu verwenden. Der Vorteil, Business-Logik-Ausgaben verwenden zu können, besteht jedoch darin, Engine-Leistung zu sparen, da weniger Metriken vorhanden sind.

Weitere Informationen zu dieser Methode finden Sie in der Fallstudie Ausgaben - Anwendertabellen (siehe Seite 160).

Metriken mit Zielen

In den Fällen, in denen ein Ziel für eine Metrik definiert ist, gibt es zwei Möglichkeiten zur Festlegung des Ziels: Es kann entweder als statisches oder als dynamisches Ziel festgelegt werden. Ein statisches Ziel ist das gängigste Szenario, wobei das Ziel ein Wert sein kann, auf den man sich geeinigt hat und der für die Dauer des Vertrages gültig ist.

Zum Beispiel:

Die Netzwerk-Verfügbarkeit sollte monatlich nicht weniger als 98 % betragen.

Das Ziel ist in diesem Fall 98 %.

Alternativ kann das Ziel von der Leistung der Vormonate abhängen, oder seinen Wert im Laufe des Jahres verändern. Es gibt viele mögliche Situationen, die im Allgemeinen jedoch alle über eine Formel implementiert sind. CA Business Service Insight unterstützt diese Funktion durch einen zusätzlichen Funktionsabruf von der Business-Logik-Standardvorlage. Die Zielfunktion kann im Rahmen der Business-Logik auf andere Parameter zugreifen und kann jedes erforderliche Szenario unterstützen.

Beispiel: Die Bearbeitungszeit von Anfragen am Helpdesk, die von der Helpdesk-Auslastung abhängt: Die durchschnittliche Bearbeitungszeit für Anfragen hoher Priorität ist 1 Tag, wenn es nicht mehr als 1000 Anfragetickets pro Monat gibt. Wenn innerhalb eines Monats mehr als 1000 Anfragetickets am Helpdesk ausgestellt werden, beträgt die durchschnittliche Lösungszeit für Anfragen hoher Priorität 2 Tage.

In diesem Fall ist die Metrik als dynamisches Ziel definiert, das innerhalb des Business-Logik-Skripts gemäß der Anzahl der in diesem Monat ausgestellten Anfragetickets ausgewertet wird.

Hinweis Weitere Einzelheiten zur Implementierungsmethode von dynamischen Zielen finden Sie in der Fallstudie Implementieren dynamischer Ziele.

Metrikparameter

Ein Metrikparameter ist ein Wert, der innerhalb der Business-Logik der Metrik angesprochen werden kann und der durch die Metrikdefinition auf einfache Weise geändert werden kann, ohne dass der eigentliche Code verändert werden muss. Er wird verwendet, um den hartcodierten Wert zu ersetzen, und er lässt sich leicht ändern.

Es ist wichtig, Metrikparameter zu identifizieren, um so die Business-Logik-Module auf einfache Weise identifizieren und wiederverwendbare Inhalte erstellen zu können. Darüber hinaus kann über den Vertragsassistenten auf Metrikparameter zugegriffen werden, wodurch der Endbenutzer leicht Änderungen vornehmen kann.

***Zum Beispiel*:**

- Vorfälle mit dem Schweregrad 1 sollten innerhalb von 24 Stunden geklärt werden.

  In der oben aufgeführten Verpflichtung besteht das Ziel in einem Klärungszeitraum von 24 Stunden, und der Schweregrad (Severity1) kann als Parameter festgelegt werden.

- Die Ausfallhäufigkeit während eines Monats sollte 3 Ausfälle nicht überschreiten, wobei Ausfallzeiten die Zeiten sind, in denen das System länger als 5 Minuten nicht verfügbar ist.

  In der oben aufgeführten Verpflichtung kann die Zeit, die als Ausfallzeit gilt, als Parameter definiert werden.

Vertragsparameter

Ein Vertragsparameter ist ein Wert, der von jeder Metrik innerhalb eines Vertrags angesprochen werden kann. Ein Vertragsparameter kann innerhalb einer Metrik verwendet werden, die dieselbe Methode wie ein Metrikparameter verwendet, wobei dieser stattdessen als dynamischer Parameter definiert ist.

Die Verwendung eines Vertragsparameters wird empfohlen, wenn mehr als eine Metrik denselben Wert benötigt. Ein anderer wichtiger Grund zur Verwendung eines Vertragsparameters besteht in der Vereinfachung der Vertragspflege. Da sich Parameter in der Regel häufig ändern und dadurch Aktualisierungen innerhalb des Systems erforderlich machen, ist es einfacher, auf einen einzigen Speicherort im Vertrag zuzugreifen und alle Parameterwerte gleichzeitig zu ändern, als auf jede Metrik im Vertrag separat zuzugreifen und die Werte der Parameter auf der Metrik-Ebene zu ändern.

Daher besteht die am meisten empfohlene Modellierung darin, die Parameter auf Vertragsebene als Vertragsparameter zu definieren und auf ihre Werte über die dynamischen Parameter auf der Metrik-Ebene zuzugreifen.

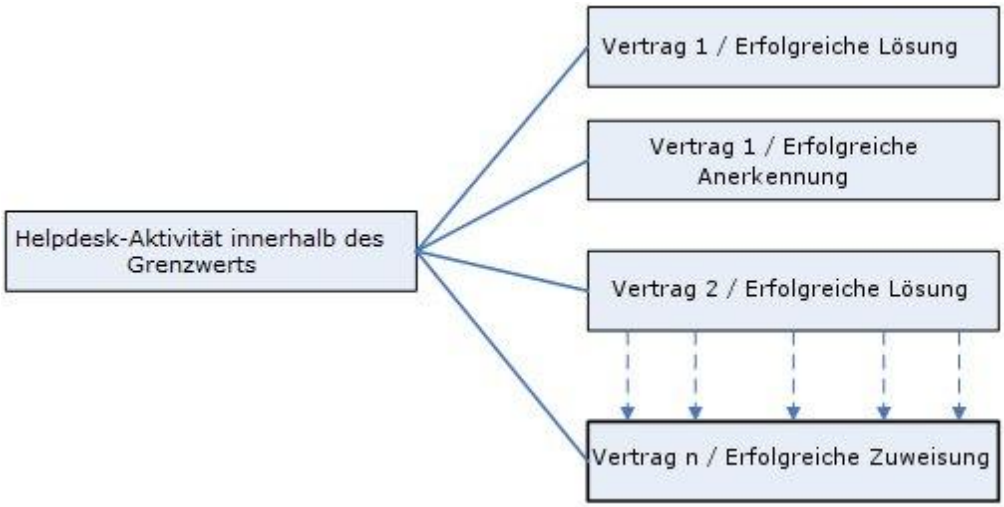
Ein Beispiel finden Sie in der Fallstudie Helpdesk-Leistung (siehe Seite 229).

Business-Logik-Vorlagen und -Module

Business-Logik-Vorlagen sind eine einfache Methode zum Speichern einer Berechnungsmethode für eine Metrik. Sie sind eine vollständige Business-Logik-Komponente und stellen eine praktische Art zur Erstellung einer Nulllinie für andere Business-Logik-Komponenten dar. Neue Business-Logik-Komponenten, erstellt von einer Vorlage durch Kopieren des Codes und Erstellung einer neuen Instanz. Im Allgemeinen ist die Flexibilität bei Verwendung von Vorlagen jedoch sehr gering, und wo immer dies möglich ist, sollten Business-Logik-Module verwendet werden.

Business-Logik-Module sind unabhängige Codekomponenten, die die Wiederverwendung derselben Codes basierend auf anderer Business-Logik ermöglichen. Module können auch andere Module einschließen, daher können die Hierarchieebenen mehrfach verzweigt sein. Bei der Verwendung von Modulen ist der Code an einem Speicherort abgelegt. Er wird von allen anderen Komponenten wiederverwendet, die mit dem Code verknüpft sind. Diese Wiederverwendung von Codeabschnitten vereinfacht die Wartung, da eine Codeduplizierung vermieden wird, und es möglich ist, systemweite Logikänderungen schnell und mühelos zu übernehmen.

In der Design-Phase müssen die Business-Logik-Hauptmodule samt ihrer zugehörigen Parameter identifiziert werden. Sobald die Vertragsmodellierung abgeschlossen ist und der Vertragsmanager eine klare Sicht der zu verwendeten Logik hat, lassen sich die gemeinsamen Kalkulationen identifizieren und in separaten Modulen festlegen.



Das oben aufgeführte Diagramm stellt ein Modul dar, welches die Erfolgsrate der Helpdesk-Aktivität berechnet, um ein Ziel innerhalb der festgelegten Schwellenwerte zu erfüllen. Um es, wie beschrieben, zu implementieren, müssen zwei Parameter - die so genannten Metrikparameter - definiert werden: Ein Metrikparameter, der die Art der Helpdesk-Aktivität definiert, und der andere Metrikparameter für den Schwellenwert, mit dem verglichen wird (siehe die Definition von *Metrikparameter* unter Während des Modellierungsprozesses zu berücksichtigende Fälle (siehe Seite 33)).

Durch sorgfältige Berücksichtigung der Berechnungsarten, die im System implementiert sind, werden Sie möglicherweise feststellen, dass sich eine Reihe ähnlicher Typen durch die Änderung nur eines kleinen Codeabschnitts ausführen lassen, und dass der Parameter als "Schalter" zwischen den Typen fungieren kann. Auf diese Weise können Sie die Menge der Codes minimieren, die Sie erstellen müssen, und die Häufigkeit der Wiederverwendung der Codes maximieren.

Service Level-Vorlagen

Eine Service Level-Vorlage ist eine Zusammenstellung von Servicekomponenten sowie deren zugewiesenen Metriken, die zu deren Messung definiert wurden. Diese Service Level-Vorlagen können als erforderlich erstellt werden, und sie werden oft gemäß den gängigsten gemessenen Serviceaspekten definiert.

Das Hauptproblem beim Festlegen von Service Level-Vorlagen ist, dass alle möglichen Parameter, die zur Verhaltensänderung der Metrik dienen könnten, identifiziert und dargelegt werden sollten. Dadurch ergibt sich die größte Flexibilität für das System und es erleichtert die Konfiguration für die Systembenutzer. Wird die Service Level-Vorlage verwendet, um neue Metriken zu erstellen, wird jeder der Konfigurationsparameter auf der Assistentenoberfläche dargelegt, sodass nur eine geringe oder überhaupt keine weitere Benutzeranpassung nötigt ist, um den Vertrag zu aktivieren. Die Parameter, die dem Benutzer durch die Verwendung des Assistenten angezeigt werden, sind in der Zielvorgabe aufgeführt. Berücksichtigen Sie daher in der Zielvorgabe der Metrik alle Parameter, die der Endbenutzer ändern können soll.

Um bei Service Level-Vorlagen für maximale Effizienz zu sorgen, sollten Sie versuchen, die gesamte Business-Logik über die Modulfunktion, wie zuvor beschrieben, abzuwickeln.

Nachfolgend finden Sie ein Beispielszenario für die Verwendung von Service Level-Vorlagen, wobei dem Kunden die Applikationshostingservice-Komponenten, je nach Summe, die der Kunde zu zahlen gewillt ist, auf den drei Ebenen Bronze, Silber und Gold bereitgestellt werden. Zusätzliche Metriken können auf allen höheren Host-Ebenen - mit strengeren Zielen - bereitgestellt werden. Jede dieser Ebenen kann ein guter Kandidat für die Erstellung einer Service Level-Vorlage sein, wie im folgenden Beispielszenario dargestellt:

- Applikationshosting - Bronze (5 Metriken)
- Applikationshosting - Silber (8 Metriken)
- Applikationshosting - Gold (12 Metriken)

Wenn dem System jetzt ein neuer Kunde hinzugefügt wird, ist es einfach, die erforderliche Definition gemäß den Kundenwünschen auszuwählen. Durch die Verwendung der Assistentenoberfläche kann jede der darin enthaltenen Metriken für den jeweiligen Kunden angepasst werden. Beachten Sie, dass es auch möglich ist, nur einige der Metriken von den Definitionen auszuwählen, oder sogar Metriken von zwei verschiedenen Definitionen zum neuen Kundenvertrag hinzuzufügen, wenn einige besondere Benutzeranpassungen erforderlich sind.

Die Bedeutung eines soliden Servicekatalogs

Wie zuvor beschrieben, ist der Servicekatalog eine Hauptkomponente des Systems, die auf strukturierte Weise konfiguriert werden sollte. Dadurch kann das System von allen Benutzern effektiv genutzt werden und Verwirrung wird vermieden. Dadurch kann das System zudem in der Organisation wachsen und künftige Erweiterungen und Zusätze mit minimalen Auswirkungen auf das bisher schon Geschaffene bewältigen.

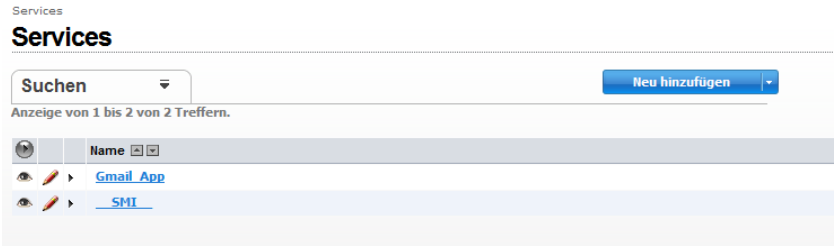
Das System bietet bei der Erstellung und Verwaltung des Servicekomponentenkatalogs und der SLAs ein hohes Maß an Flexibilität. Da es jedoch nur so gut sein kann wie der Entwurf, ist der Zeitaufwand für die Zukunftsplanung in den frühen Entwurfsphasen unverzichtbar.

Die Definition des CA Business Service Insight-Servicekatalogs auf effiziente und strukturierte Weise bietet Ihrem System die Flexibilität, dem Servicekatalog zu einem späteren Zeitpunkt Serviceleistungen und Domänen hinzuzufügen. Es ermöglicht auch, zukünftig Verträge, Metriken, Warnungen und Berichte hinzuzufügen. Außerdem führt es zu einer strukturierteren Herangehensweise an die Business-Logik und ebnet den Weg für die Möglichkeit zur Nutzung standardisierter Module und Vorlagen, um so Geschäftsdaten und zugehörige Kalkulationen abwickeln zu können.

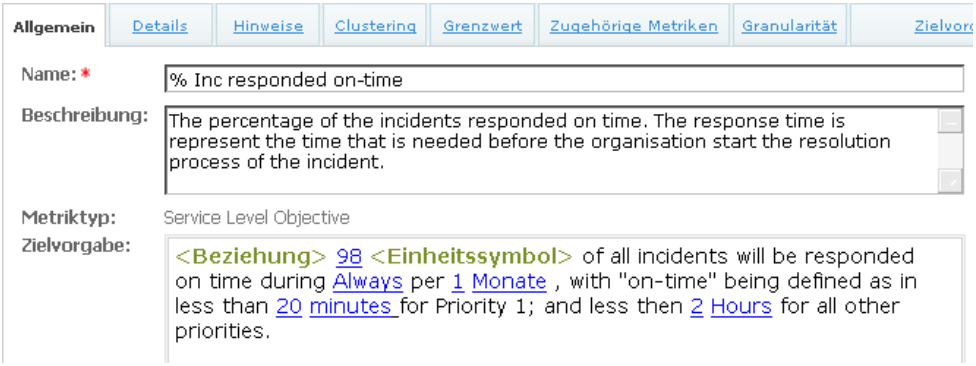
Zusammen mit dem Katalog stellen die im System angegebenen Service Level-Vorlagen eine ausgezeichnete Methode für den Vertragsmanager dar, mühelos neue Verträge zu erstellen und mit wenigem oder ohne Vorwissen bezüglich der zugrunde liegenden Ebenen der benötigten Daten. Nach der effektiven Einrichtung sollte es möglich sein, einen neuen Vertrag für einen Kunden nur durch das Ändern der Parameter in der Service Level-Vorlage zu konfigurieren. Dafür müssen jedoch der Katalog und die Definitionen möglichst effektiv angelegt sein. Diese Parameter werden alle über die Zielvorgabe jeder Metrik in der Service Level-Vorlage dargelegt, und können entweder dort oder vom Assistenten unter Verwendung der Definition geändert werden.

Beispiel:

Wenn eine Service Level-Vorlage verwendet wird, um eine Kundenberatungsmetrik auf einen Vertrag anzuwenden, lassen sich die erforderlichen Metriken aus einer vorhandenen Definition auswählen.



In dieser Service Level-Vorlage gibt es eine Metrik, die so genannte "*% der pünktlichen Lösungen*". Sie können sehen, dass es hier ein gewisses Maß an Subjektivität gibt, da die Komponente "*pünktlich*" fraglich sein kann. Das folgende Beispiel erklärt die in dieser Metrik erstellte Messung:



Die Zielvorgabe unten auf der Registerkarte Allgemein (oder alternativ auf der Registerkarte Zielvorgabe) enthält alle Parameter dieser Metrik. In der vorherigen Abbildung ist die Definition von"*Betriebszeit*" als 20 Minuten angegeben. Dies ist ein benutzerdefinierbarer Parameter, um so unsere eigene Neuinterpretation dieser vordefinierten Metrik zu ermöglichen. Um diesen Wert zu verändern, können Sie auf den Parameterlink 20 Minuten klicken.

Auf diese Weise kann die neue, anhand der Service Level-Vorlage erstellte Metrik individuell gestaltet werden, ohne dass dazu die zugrunde liegende Business-Logik der Metrik geändert werden muss. Beachten Sie, dass dabei zudem davon ausgegangen wird, dass die Business-Logik so geschrieben wurde, dass diese Parameter in die Service Level-Berechnung integriert werden können.

Dieses einfache Beispiel zeigt deutlich, wie wichtig es ist, einen leistungsstarken und flexiblen Satz Service Level-Vorlagen für den Systemkatalog zu erstellen, um bei zukünftigen Verträgen auf diese zurückgreifen zu können.

Finanzmanagement (Vertragsstrafen, Bonus und Kosten)

In früheren CA Business Service Insight-Versionen gab es Vertragsentitäten - so genannte Vertragsstrafen - die mithilfe von Excel-ähnlichen Formeln implementiert wurden. Vertragsstrafen basieren ihre Ergebnisse allein auf die Eingabe von den Vertragsmetriken und stützen die Berechnung des resultierenden Vertragsstrafmaßes auf die Basisfunktionen. Ab Version 4.0 sind diese durch vom Benutzer generierbare, komplette Finanzmetriken ersetzt worden, die eine größere Flexibilität ermöglichen. Diese Finanzmetriken können verwendet werden, um Bonus- oder Kosten-Informationen zum Vertrag zu liefern.

Hinweis: Ein Bonus ersetzt die alte Vertragsstrafen-Terminologie von CA Business Service Insight ab Version 3.0 und kann, je nach Leistung, positiv oder negativ sein. Ein negativer Bonus entspricht jedoch im Wesentlichen einer Vertragsstrafe. Ferner muss beachtet werden, dass Sie, wenn Sie eine Vertragsstrafe mit dem Bonus-Metriktyp implementieren, daran denken *müssen,* die Funktion "Result()" so zu konfigurieren, dass ein negativer Wert ausgegeben wird. Dadurch können Additionsfunktionen, die die verschiedenen Metrikergebnisse kombinieren, die Summen in die richtige Richtung (positiv/negativ) zusammenzufassen. Ein Bonus erhöht also den Wert, während eine Vertragsstrafe ihn vermindert.

CA Business Service Insight Version 4.0 bietet zudem die Möglichkeit zur Erstellung einer Verbrauchsmetrik, um die Nutzung von Servicekomponenten und -ressourcen zu messen, sowie um sie zur Ermittlung der Kosten dieses Services oder dieser Ressource mit einer Preisposten-Metrik zu kombinieren. Durch die Kombination mit der optimierten Prognosefunktion können jetzt einige umfassende Finanzmanagement-Metriken erstellt werden.

Finanzmetriken können auch die Ausgabe von anderen Vertragsmetriken akquirieren, und die zugehörigen Vertragsstrafen- oder Bonuswerte basierend auf der Leistung dieser Vertragsmetriken ermitteln. Sie können auch mit anderen Informationsarten arbeiten, um deren Ergebnis zu bestimmen, wie Stückpreis-Daten und Prognosemodelle, die Berichterstellungsfunktionalitäten, wie Sollkosten verglichen mit Istkosten, ermöglichen.

***Kostenposten-Beispiel*:**

Eine spezielle Risikoanwendung verfügt über zugewiesene Kosten basierend auf der Anzahl gleichzeitiger Benutzer des Systems. Die Kosten werden monatlich kalkuliert, und es gibt einen Prognosewert für diese Anwendung. Der Stückpreis dieser Anwendung (Kosten pro gleichzeitigem Benutzer) ist in der Tabelle unten aufgeführt (bedenken Sie, dass diese Anwendung unter den Index 1 fällt):

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

Die prognostizierte Anzahl gleichzeitiger Benutzer für diesen Zeitraum ist ebenfalls verfügbar (wieder Index 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

Beim Modellieren dieser Kostenmetrik unter Verwendung dieser Kalkulationstabellen lassen sich die monatlichen Anwendungskosten basierend auf der tatsächlichen Anzahl gleichzeitiger Benutzer ermitteln. Diese Informationen stammen von der Datenquelle und können mit den oben aufgeführten Zahlen für den Preis-pro-Einheit multipliziert werden, um so die Kostenzahlen zu liefern. Sie können auch mit den prognostizierten Werten verglichen werden, um eine Analyse der Sollkosten im Vergleich mit den Istkosten zu liefern. Die Business-Logik wird in diesem Fall zur Ermittlung der Istanzahl gleichzeitiger Benutzer im entsprechenden Zeitraum zu ermitteln und sie mit den Werten für die Stückkosten zu multiplizieren. Außerdem bezieht sich die Prognosefunktion in der Business-Logik auf die Prognosedaten. Dies ist ein Beispiel für die Anwendung einer Kostenposten-Metrik.

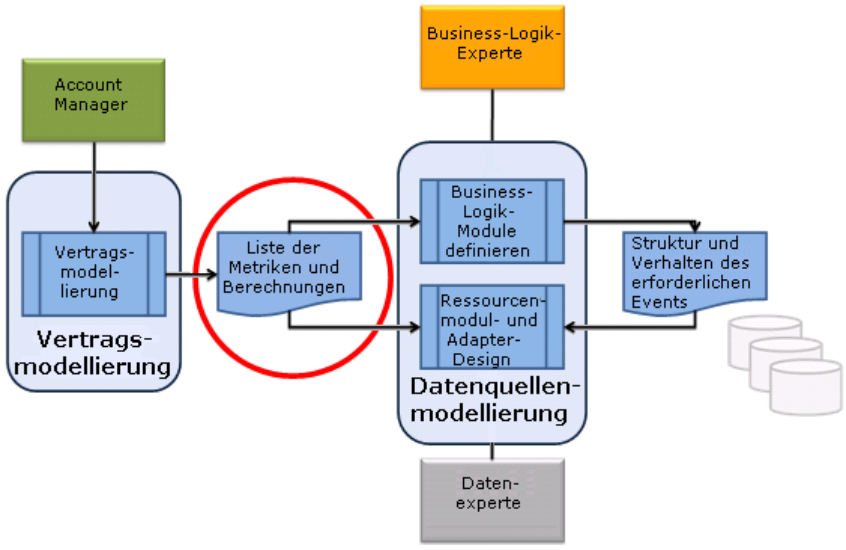
***Vertragsstrafen-Szenario-Beispiel*:**

Der Kunden-SLA beinhaltet eine Nichterfüllungsklausel, um sicherzustellen, dass das Netzwerk 98 % der Zeit eines bestimmten Monats während der Geschäftszeiten verfügbar ist. Ein monatlicher Service Level darunter geht Strafzahlungen basierend auf einer Formel (Vertragsstrafe = $1000 pro jeden ganzen Prozentsatz unter dem Ziel (d. h. 96, 5 % = (98-Gerundet (96,5)) * 1000 = (98-97) * 1000 = -$1000)) ein.

Um diese Vertragsstrafenbedingung zu implementieren, können Sie eine Finanzbonus-Metrik erstellen, die seine Eingabe von einer vorhandenen Metrik (*Netzwerkverfügbarkeit->=-98 %)* übernimmt. Der Registrierungsprozess für diese Metrik nutzt den RegisterByMetric()-Prozess, um die Service Level-Werte von dieser Metrik zu erhalten und die Vergleiche durchführen zu können. Dies sendet die Service Level-Werte für den Kontrollzeitraum dieser Metrik in die Finanzmetrik, wie Ereignisse, die dann als Teil der Kalkulation verwendet werden, um die Höhe der Vertragsstrafe für denselben Zeitraum mithilfe der Formel aus dem Szenario zu ermitteln.

Hinweis: Weitere Fallstudien finden Sie unter Finanzmetrikmodellierungs-Studie (siehe Seite 232).

Vertragsmodellierungsphasen-Ausgaben (Vertragsmanager, Datenexperte)



Wie im vorherigen Diagramm dargestellt, muss der Vertragsmanager am Ende der Vertragsmodellierungsphase die Liste der Metriken und deren erforderliche Berechnungen bereitstellen, um mit der Datenmodellierungsphase der Lösung fortzufahren. Die Liste umfasst die folgenden Informationen:

- Eine abgeschlossene Servicekatalogdefinition umfasst:
  - Vollständige Liste der angebotenen Servicekomponenten
  - Eine Aufgliederung der Servicedomänen und Domänenkategorien
  - Definition aller benötigten Zeitfenster für jede der Metriken
  - Eine Liste von Business-Logik-Modulen zur Abwicklung jeder Kalkulationsart (einschließlich der Parameter, die zu deren Implementierung benötigt werden)
- Eine vollständige Liste der zu implementierenden Metriken einschließlich:
  - Gut definierte Namensgebungsstandards für Metriken
  - Kategorisierung jeder Metrik gemäß vordefinierter Servicekatalogkomponenten

Dieses Dokument ist ein nützliches Werkzeug, um sicherzustellen, dass alle Hauptdefinitionen für einen Vertrag generiert und dass alle seine Metriken vollständig definiert wurden.

Die Informationen in dieser Tabellenkalkulation sind die Eingabe, die der Datenexperte benötigt, um mit der nächsten Phase - der Datenmodellierung - fortzufahren.

Nachdem diese Elemente fertig gestellt wurden, kann mit der nächsten Phase fortgefahren werden, in der Sie mit der Modellierung mit tatsächlichen Daten aus den Quellen beginnen können. Ohne ein abgeschlossenes Vertragsmodell ist es sehr schwierig, genau zu wissen, was von der Datenquelle zur Erfüllung der vertraglichen Verpflichtungen benötigt wird.

Daten-Modellierung (Datenexperte, Business-Logik-Experte)

Der Abschnitt über die Datenmodellierung beschreibt den zweiten Teil des Entwurfsprozesses. Die Datenmodellierungsphase ist der Prozess, bei dem Daten von Kundenquellen übernommen werden, spezifische Komponenten der erforderlichen Daten identifiziert werden, sich entschieden wird, wie diese Daten standardisiert werden müssen, und beurteilt wird, wie die relevanten Daten den entsprechenden Metriken (über den Registrierungsprozess) zugewiesen werden.

Diese Phase umfasst die folgenden Aufgaben:

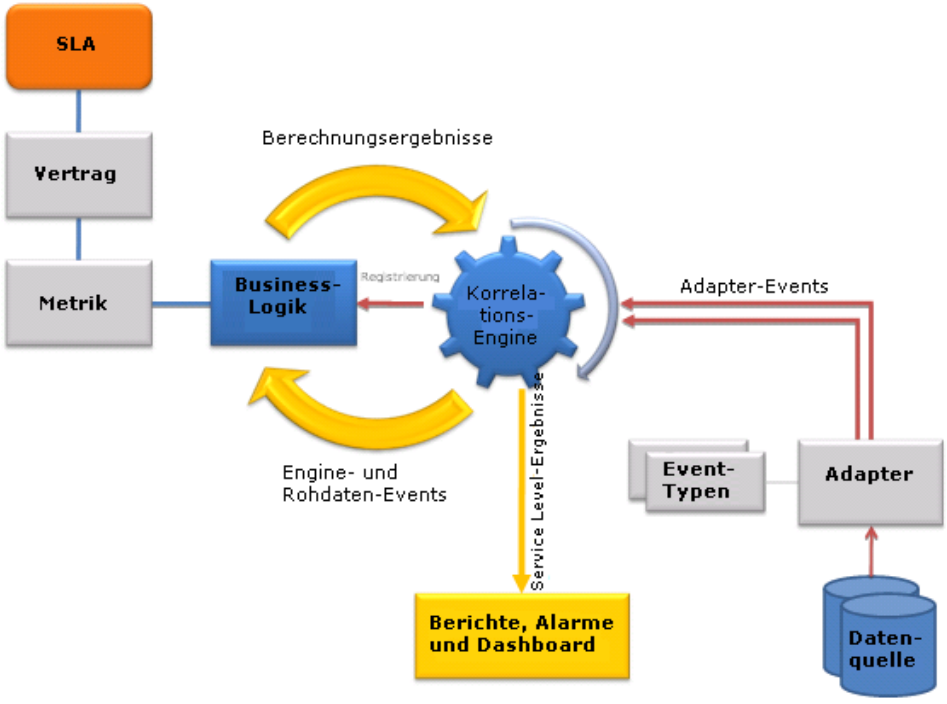
- **Business-Logik-Experte**:
  - Identifizieren und Definieren der für die Kalkulation erforderlichen Eingabe-Event-Struktur, damit sie später als Event-Typ festgelegt werden kann
  - Einbinden der Pflichtfelder in den Event-Typ, um so alle erforderlichen Daten für die Berechnungen, die diese nutzt, zu liefern
  - Definieren des Metrikerfassungsprozesses zur Maximierung des Event-Flusses
  - Entscheiden, wie das Ressourcenmodell von verfügbaren Daten aufgebaut werden soll, um alle Berichterstellungs- und Logikanforderungen zu erfüllen
- **Datenexperte**:
  - Identifizieren des Datenquellentyps und Entscheiden, wie er vom Adapter in definierte Event-Typen standardisiert werden und in die CA Business Service Insight-Datenbank eingelesen werden soll
  - Bestimmen von Event-Typ-Kennungen, die den Daten angefügt werden sollen

Ereignisse und ihr Flow

Der Datenfluss innerhalb des Systems erfolgt in Form von Ereignissen. Wobei ein Ereignis eine Informationsmeldung ist, die von dem Adapter basierend auf Quellendaten erstellt wird ist und die in einem Format vorliegt, das von CA Business Service Insight für seine Service Level-Berechnungen verwendet werden kann. Rohdaten bestehen immer aus Ereignissen.

Der Schwerpunkt des Entwurfs muss daher auf diesem Ereignisfluss innerhalb des Systems liegen.

Vor der Modellierung der Datenanforderungen müssen sowohl der Business-Logik-Experte als auch der Datenexperte ein gutes Verständnis für die Ereignisse und ihren Fluss innerhalb des CA Business Service Insight-Systems haben. Das folgende Diagramm veranschaulicht in hohem Maß diesen grundlegenden Ereignisfluss.



Das vorherige Diagramm stellt dar, wie Ereignisse von der Datenquelle durch die Adaptern abgerufen und in eine Standard-Ereignisstruktur, die als Event-Typ definiert ist, standardisiert werden. Diese Ereignisse werden von den Adaptern an CA Business Service Insight gesendet. Diese Ereignisse werden als Rohdatenereignisse bezeichnet.

Die Business-Logik-Kalkulationen in jeder Metrik basieren auf einer Teilmenge der Rohdatenereignisse. Die Business-Logik fordert diese Teilmenge daher mit der Durchführung der *Registrierung* an.

Basierend auf die Registrierungsbeschreibung sendet die Korrelations-Engine nur die Rohdatenereignisse, die für die Business-Logik-Berechnungen relevant sind.

Engine-Ereignisse sind weitere Event-Typen, die an die Business-Logik gesendet werden. Alle in diesen Prozess einbezogenen Begriffe werden in diesem Kapitel ausführlich behandelt.

Dieser Abschnitt konzentriert sich auf die folgenden Teile des Diagramms:

- Datenquelle
- Adapter
- Event-Typ(en)
- Metrikerfassung

Das CA Business Service Insight-Datenmodell wurde entwickelt, um die Leistungsfähigkeit dieses Datenstroms innerhalb des Systems zu maximieren.

Im Allgemeinen funktioniert CA Business Service Insight auf zwei Ebenen: Der Infrastrukturebene und der Geschäftsmodellebene. Als eine simple HW-Störung schließt die Infrastrukturebene Adapter, Ressourcen und Event-Typ-Objekte ein, während die Geschäftsebene Verträge, Metriken und Service-Objekte umfasst. Zwischen den beiden Ebenen gibt es eine virtuelle Shim-Ebene, die so genannte Korrelationsebene.

Eine Ereigniskennung ist das Event-Typ-Objekt. Der Event-Typ legt fest, wie Ereignisse definiert werden und wie sie CA Business Service Insight gemeldet werden. Er legt zudem die Struktur des Feldes für Ereignisdaten fest, sodass dieses während der Verarbeitung durch die Business-Logik interpretiert werden kann.

Eine andere Ereigniskennung ist die Ressource, die kleinste in Kalkulationen verwendete Entität. Beispiel: Bei der Berechnung der Serververfügbarkeit kann die logische Definition der kleinsten Entität, über die eine Berichterstellung erforderlich ist, ein spezieller Server sein, oder es kann ein Kunde sein, wenn über die Handhabung des Anfragetickets dieses Kunden berichtet wird. Die Ressource ist eine Definition einer CA Business Service Insight-Entität, die sowohl von der Datenquelle als auch von der Berechnungsanforderung abgeleitet wird. Jeder Ressource wird ein Ressourcentyp zugewiesen, eine Ressourcenkennung, die genau festlegt, welcher "Typ" Ressource definiert wird. Jeder Ressource muss ein Ressourcentyp zugewiesen sein, was zudem das Hinzufügen von individuellen Attributen zu jeder Ressource ermöglicht. Weitere Informationen zu diesen Attributen finden Sie unter Ressourcen und ihre Verwaltung (siehe Seite 53).

Die Korrelation tritt zwischen eingehenden Adapterereignissen und Vertragsmetriken auf. Das Kernstück dieses Korrelationsprozesses sind die Ressourcenzuordnung und die Metrikerfassung.

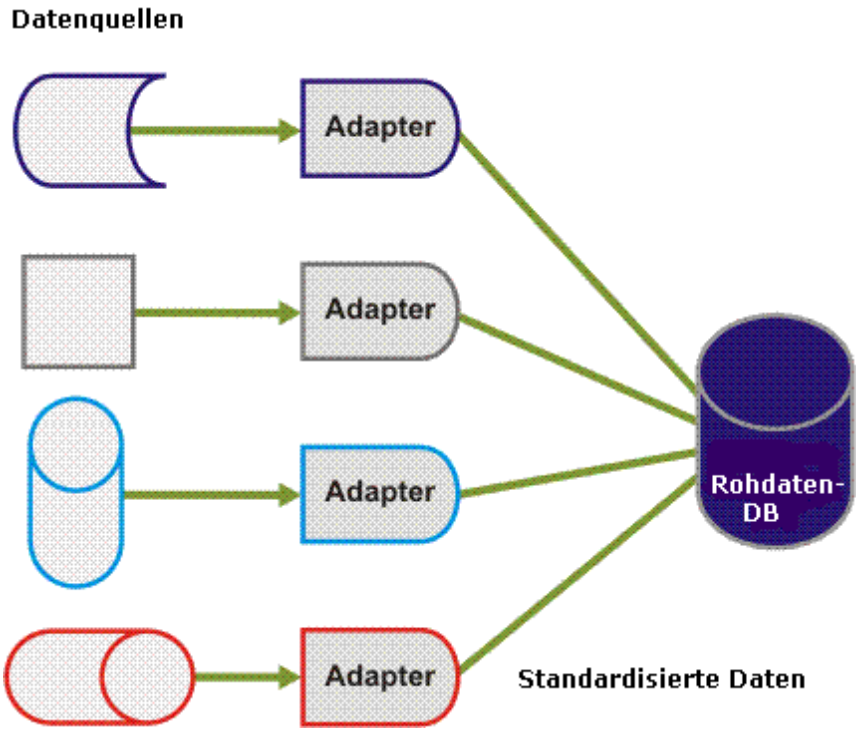
Ressourcenzuordnung und Metrik-Registrierung legen fest, welche Ressourcenereignis-Streams gemessen werden, und von welcher Metrik.

Zu beachten ist, dass es durch die Metrikerfassung zu einem gewissen Grad von Wiederverwendung und Coabhängigkeit mit anderen Metriken kommen kann, da es möglich ist, die Ausgabe einer Metrik als Eingabe bei einer anderen zu verwenden. Entsprechend gibt es Übergangsereignisse, die nicht als Ausgabe einer Metrik zur Service Level-Messung verwendet werden, sondern eher als Berechnungszwischenschritt, der dann von anderen Metriken verwendet werden kann.

Datenmodell - Übersicht

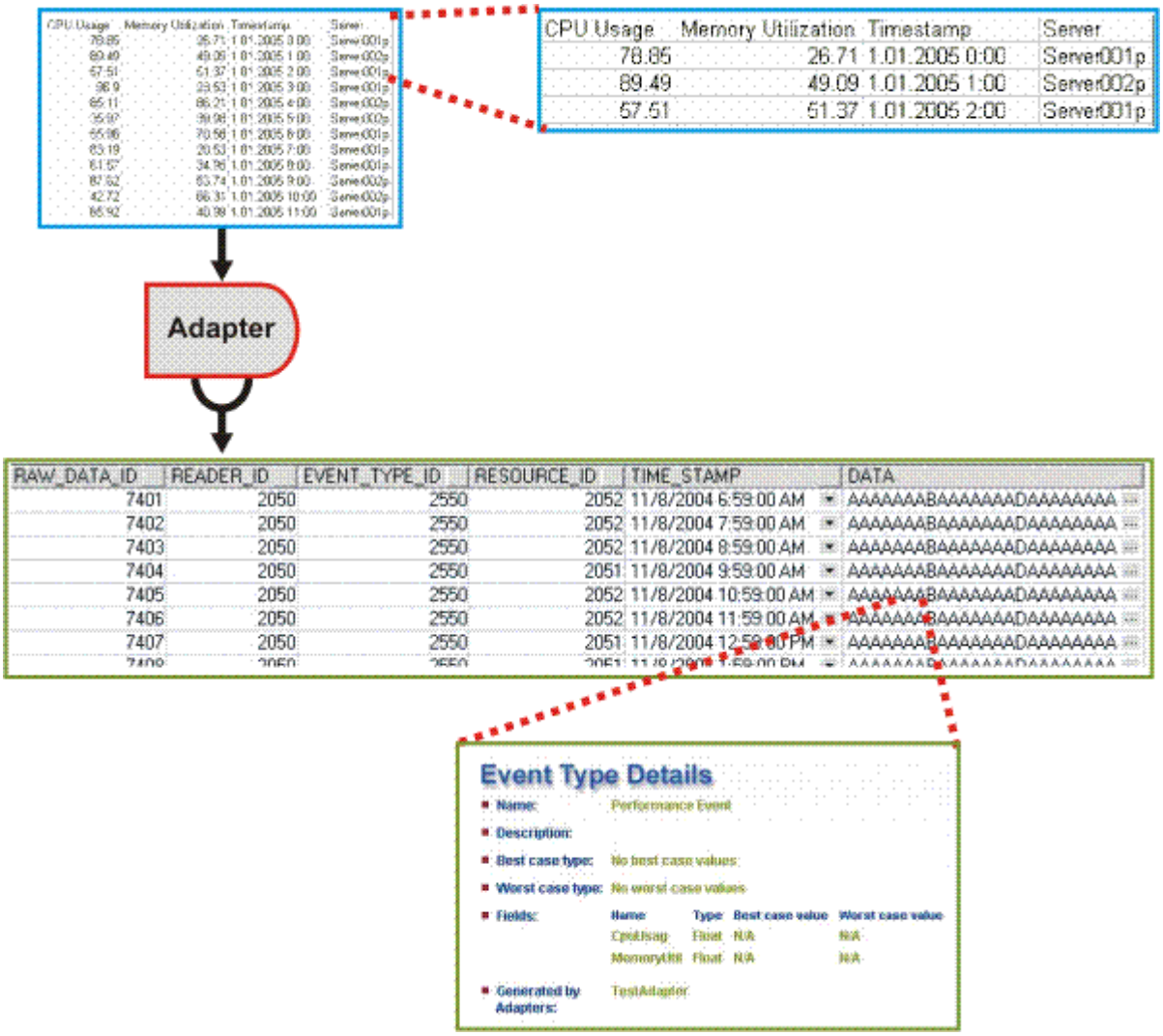
Das CA Business Service Insight-Datenmodell wurde konzipiert, um die folgende Herausforderung anzunehmen und sie zu überwinden.

Rohdaten werden von den Adaptern von verschiedenen, disparaten Datenquellen abgerufen und in einer Vielzahl von Formaten gehalten. Diese vielfältigen Daten müssen in eine einzelne Datenbanktabelle abgerufen und vereinheitlicht werden. Daher müssen Adapter die Daten in ein vereinheitlichtes Datenmodell einlesen und standardisieren, wie in der folgenden Abbildung dargestellt.



Als Teil dieses Prozesses werden alle Datenfelder in dasselbe Datenbanktabellenfeld eingefügt, jedoch in verschlüsselter Form. Jeder Zeile, die in die CA Business Service Insight-Datenbank eingefügt wird, ist eine Event-Typ-Kennung zugewiesen. Die Event-Typ-Definition enthält Beschreibungen der Datenfelder. Sie ermöglicht zudem der Korrelations-Engine, die Datenfelder korrekt zu deuten und zu erkennen, wann sie von der Business-Logik für Kalkulationen benötigt werden.

Die folgende Abbildung zeigt eine grafische Darstellung des Datenabruf- und Datenbankpopulations-Abschnittes dieses Prozesses. Auch dargestellt ist ein erweiterter Abschnitt, der zeigt, was die Daten unter Real-Life-Bedingungen darstellen, und nicht, wie die Rohdaten angezeigt werden.



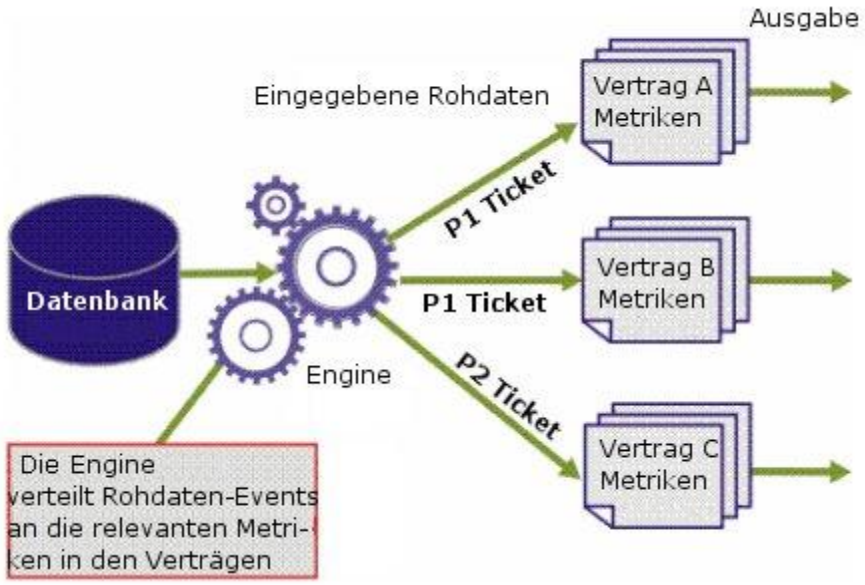
Das CA Business Service Insight-System umfasst zudem alle Verträge und Metriken, die gegen die Rohdaten evaluiert werden müssen, um Service Level-Leistungsdaten zu liefern. Jede Metrik darf nur die Teilmenge der Daten erhalten, die für seine Berechnung relevant sind. Die Rohdaten beinhalten eine potenziell riesige Anzahl von Datensätzen verschiedener Arten. Die Metrik zum Filtern der relevanten Ereignisse nach ihren Werten zu verwenden, ist äußerst ineffizient. Daher verteilt die CA Business Service Insight-Engine die relevanten Rohdaten an jede spezifische Metrik.

***Beispiel*:**

Für die folgenden zwei Metriken in einem Vertrag:

- Durchschnittliche Problemlösungszeit von Tickets der Priorität 1 (P1)
- Durchschnittliche Problemlösungszeit von Tickets der Priorität 2 (P2)

Die erste Metrik muss nur Tickets mit der Priorität 1 auswerten, die zweite Metrik nur Tickets mit der Priorität 2. Daher muss die Engine die Datensätze entsprechend verteilen. Darüber hinaus wird die Problemlösungszeit innerhalb eines Vertrags für P1-Tickets kalkuliert, die für die Vertragspartei A geöffnet sind, während im zweiten Vertrag P1-Tickets für die Vertragspartei B und im dritten Vertrag P2-Tickets für die Vertragspartei C geöffnet werden. Daher muss die Engine den Ticket-Typ und den Kunden auswählen, für den die Berichterstellung erfolgte, wie in der nachstehenden Abbildung dargestellt.



Wie zuvor erklärt, sind den Rohdatendatensätzen Kennungen angefügt, mit denen die Engine die relevanten Datensätze und Ereignisse zur Business-Logik einer jeden Metrik identifizieren kann. Die beiden Kennungen sind der Event-Typ und die Ressource.

Adapterübersetzung und Standardisierung

Der Adapter hat die Aufgabe, Daten von der Datenquelle auszulesen und sie in das Format eines Ereignisses zu standardisieren. Jedes Ereignis in CA Business Service Insight besteht aus den folgenden Feldern:

- Ressourcen-ID
- Event-Typ-ID
- Timestamp
- Wertfeld(er) gemäß dem Event-Typ

Der Adapter muss die Originalfelder in der Datenquelle mit den entsprechenden CA Business Service Insight-Ressourcenfelder verknüpfen. Dazu verwendet er eine Übersetzungstabelle, die den in der Datenquelle gefundenen Wert und die entsprechende CA Business Service Insight-Ressourcenkennung (Ressourcen-ID) enthält.

Der Prozess des Anfügens der Ressourcen-ID und der Event-Typ-ID an den relevanten Datenquellenwert wird als *Adapterübersetzung* bezeichnet. Während dieses Prozesses wird die Übersetzungstabelle mit den übereinstimmenden Werten erstellt. Diese Tabelle wird vom Adapter verwendet, um die relevante Event-Typ-ID und die Ressourcen-IDs in den Event-Datensatz einzugeben, den der Adapter gerade erstellt. Für das Übersetzen von Ressourcen und Event-Typen werden unabhängige Tabellen erstellt.

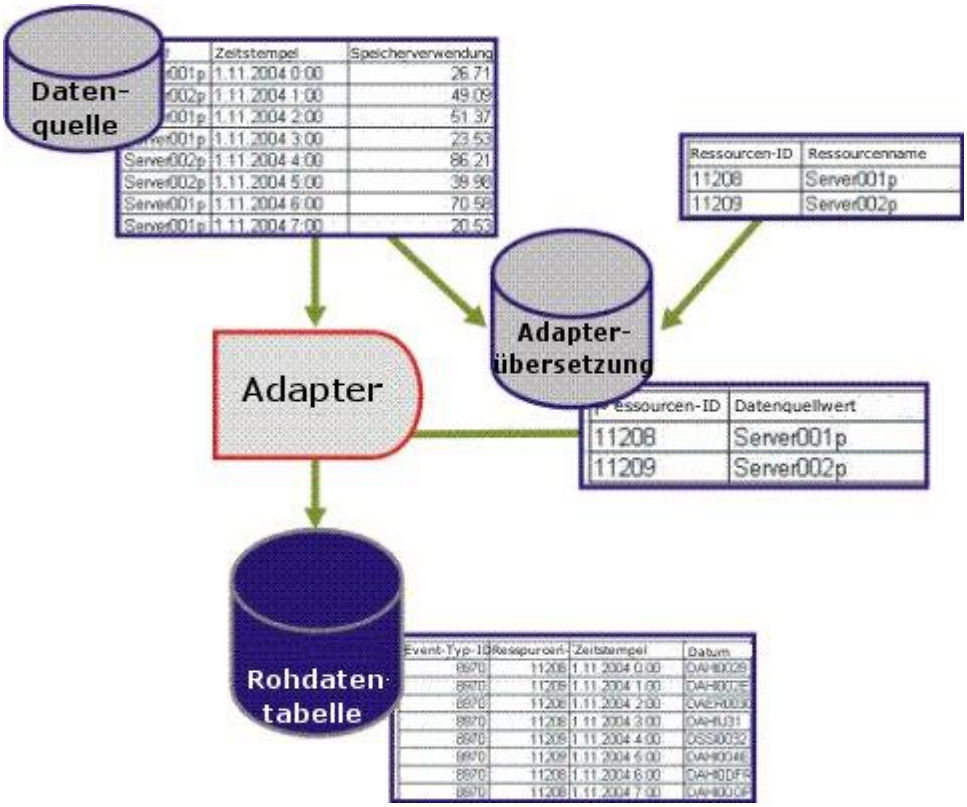
Wie zuvor erwähnt, werden die Ressourcen-ID und die Event-Typ-ID zu Registrierungszwecken verwendet, während die Datenfeld- und Zeitstempel-Werte für tatsächliche Kalkulationen verwendet werden.

Das Feld Zeitstempel wird von der Engine auch verwendet, um Reihenfolge und Timing der Ereignisse zu bestimmen, die für Berechnungen versendet wurden.

Die Event-Typ-Definition wird basierend auf der Eingabedatenquelle und der erforderlichen Ausgabe manuell in CA Business Service Insight vorgenommen.

Hinweis: Die Ressourcendefinition kann entweder manuell oder automatisch mit Übersetzungsskripten erfolgen (weitere Details finden Sie unter Ressourcen und ihre Verwaltung (siehe Seite 53)).

Die folgende Abbildung zeigt die Interaktion zwischen der Datenquelle, der Adapterübersetzungstabelle, dem Adapter und der CA Business Service Insight-Rohdatentabelle.


Daten-
quelle
Zeitstempel
Speicherverwendung
1.11.2004 0:00
26,71
1.11.2004 1:00
49,09
1.11.2004 2:00
51,37
1.11.2004 3:00
23,53
Server002p
1.11.2004 4:00
86,21
Server002p
1.11.2004 5:00
39,96
Server001p
1.11.2004 6:00
70,58
Server001p
1.11.2004 7:00
20,53
Ressourcen-ID
Ressourcenname
11208
Server001p
11209
Server002p
Adapter-
übersetzung
Adapter
Ressourcen-ID
Datenquellwert
11208
Server001p
11209
Server002p
Rohdaten-
tabelle
Event-Typ-ID
Ressourcen-
Zeitstempel
Datum
8970
11208
1.11.2004 0:00
8970
11208
1.11.2004 1:00
8970
11208
1.11.2004 2:00
8970
11208
1.11.2004 3:00
8970
11209
1.11.2004 4:00
8970
11209
1.11.2004 5:00
8970
11208
1.11.2004 6:00
8970
11208
1.11.2004 7:00

Metrikerfassung

Damit die Korrelations-Engine weiß, welche Daten möglicherweise angefordert werden, muss die Metrik ihr Vorhandensein und ihre Anforderungen bei der Korrelations-Engine registrieren lassen.

Die Metrikerfassung ist die Anfrage einer Metrik, Ereignisse zu empfangen, und zwar nur solche, die in die Berechnung übernommen werden müssen. Diese Anfrage wird vom Status des Event-Typs der Ereigniskennung und der Ressource gestellt.

Die Registrierung kann basierend auf einer einzelnen Ressource oder auf einer Gruppe von Ressourcen erfolgen.

***Beispiel*:**

Für die folgende informative Metrik: "Gesamt-Ausfallhäufigkeit des Servers X" und unter der Voraussetzung, dass die Datenquelle eine Benachrichtigung ausgibt, wenn ein Server herauf- oder herunterfährt, und dass die Benachrichtigung angibt, ob er zu einer bestimmten Zeit in herausgefahren oder heruntergefahren ist, ist die Registrierung:

Event-Typ: Server-An/Aus-Ereignis

Ressource: Server X

Basierend auf der oben aufgeführten Registrierung filtert die Engine alle Ereignisse mit Ein-/Aus-Definitionen im Feld Event-Typ sowie Server X im Feld Ressource heraus.

Sobald ein Vertrag im CA Business Service Insight-System aktiviert wird, registrieren alle Metriken die relevanten Ereignisse, die für deren Kalkulationen erforderlich sind. Basierend auf diesen Anfragen kennzeichnet die Korrelations-Engine die Ereignisse, die für die jeweilige Business-Logik relevant sind. Sobald die Berechnungen starten, werden die relevanten Ereignisse an die jeweilige Metrik zur Berechnung gesendet.

Ressourcen und ihre Verwaltung

Damit eine Registrierung dynamisch sein kann, können Ressourcen auch individuell zugewiesen werden, entweder gemäß ihrem eigenen eindeutigen Namen/ihrer eindeutigen Kennung oder gemäß ihrer Beziehung zu logischen Gruppen.

***Beispiel*:**

Für die Metrik: "Gesamt-Ausfallhäufigkeit der Datencenter-Server" ist die Registrierung:

Event-Typ: Server-An/Aus-Ereignis

Ressource: Alle Ressourcen, die von Servern des Datencenters gekennzeichnet werden. (Dies wäre wahrscheinlich eine Ressourcengruppe.)

Verstehen des Ressourcen-Lebenszyklus

Eine Ressource ist eine physische oder logische Entität, die ihre Merkmale mit der Zeit ändern kann. Sie kann gewissen Servicekomponenten oder Vertragsparteien usw. zugewiesen werden, und dann zu einem bestimmten Zeitpunkt in der Zukunft wieder neu zugewiesen werden. Jede dieser Änderungen oder Wiederzuordnungen wird von CA Business Service Insight erfasst, um so zu jedem beliebigen Zeitpunkt anhand der Ressourcenkonfiguration und -zuweisung zu genau diesem Zeitpunkt Berechnungen durchführen zu können.

Änderungen einer Ressource oder ihren Zuordnungen können jeder Zeit vorgenommen werden, dazu muss jedoch eine neue Version dieser Ressource erstellt werden. Für jede neue Version muss ein effektiver Stichtag festgelegt sein, wann die Änderungen in Kraft treten werden. Die Änderungen werden dann in die Zukunft übertragen, sofern nicht weitere Änderungen in einer späteren Version derselben Ressource gefunden werden. Alle Änderungen sind nur dann sichtbar und für die Berechnungs-Engine verfügbar, wenn diese neue Version aktiviert wurde und in Kraft ist. Dieser Prozess wird als "Inkraftsetzen" der Ressource bezeichnet.

Innerhalb von CA Business Service Insight gibt es ebenfalls eine Möglichkeit zur Handhabung mehrerer Ressourcenzuweisungen auf einmal. Für diese Methode wird ein "Änderungssatz" verwendet. Änderungssätze ermöglichen das Vornehmen von Änderungen an großen Ressourcenvolumen in einer einzigen "Transaktion", ähnlich wie eine transaktionale Datenbank arbeitet. Alle Änderungen können an allen Ressourcen vorgenommen werden, die einem Änderungssatz zugewiesen sind. Dazu wird die Operation am Änderungssatz als Ganzem vorgenommen und der Änderungssatz wird dann in einem Durchgang in Kraft gesetzt.

Im Umgang mit Ressourcen und ihren Änderungen lohnt es sich, die folgenden Punkte bezüglich der Berechnungs-Engine zu berücksichtigen:

- Wenn Änderungen einer spezifischen Ressource oder Gruppe von Ressourcen (Änderungssatz) aktiviert werden (Inkraftsetzung), überlegen Sie, was von den Änderungen im System betroffen sein kann. Da das Ressourcenmodul beim Ändern möglicherweise eine Neuberechnung auslöst, ist es wichtig, das Aktivierungsdatum der Ressource(n) sowie die Anzahl an Änderungen, die in einem Durchgang aktiviert werden, zu optimieren.
- **Massenaktualisierung**:- Es ist möglich, dieselbe Änderung auf viele Ressourcen anzuwenden (Massenaktualisierung). Da das Ressourcenmodul beim Ändern möglicherweise Neuberechnungen durchführt, ist es wichtig, diese Operation zu optimieren.

Das oben ausgeführte Beispiel geht eine Ressource nicht direkt, sondern über deren logische Zuweisung zu einer Funktion oder einem Speicherort (in diesem Fall zu seiner Funktion, einem Datencenter-Server).

Die Registrierungsanfrage könnte sich sehr aufwändig gestalten, wenn Ereignisse für jeden einzelnen Server angefordert werden, der im Datencenter gehalten wird. Ein Problem ist die Anzahl von Ressourcen, auf die sich bezogen wird. Das andere ist, dass sich die Infrastruktur des Datencenters in regelmäßigen Abständen ändert, sodass ein Server, der ein Teil des Datencenters war, dort möglicherweise überhaupt nicht mehr vorhanden ist, oder dass möglicherweise ein neuer Server hinzugefügt wurde. Daher muss die Liste dynamisch sein.

Das vorherige Beispiel zeigt deutlich, dass die Ressourcen an eine logische Gruppe angehängt werden müssen, sodass sie über diese logische Entität angesprochen werden können. Darüber hinaus kann die logische Gruppe selbst verwaltet werden müssen, wenn sie sich immer wieder ändert.

Ressourcenzuweisung ist die CA Business Service Insight-Methode der Kennzeichnung von Ressourcen. Eine Ressource kann einer oder mehreren Gruppen, Ressourcentypen, Vertragsparteien oder Services zugewiesen sein. Ressourcenzuweisungen werden unter Verwendung der CA Business Service Insight-Versionskontrolle verwaltet.

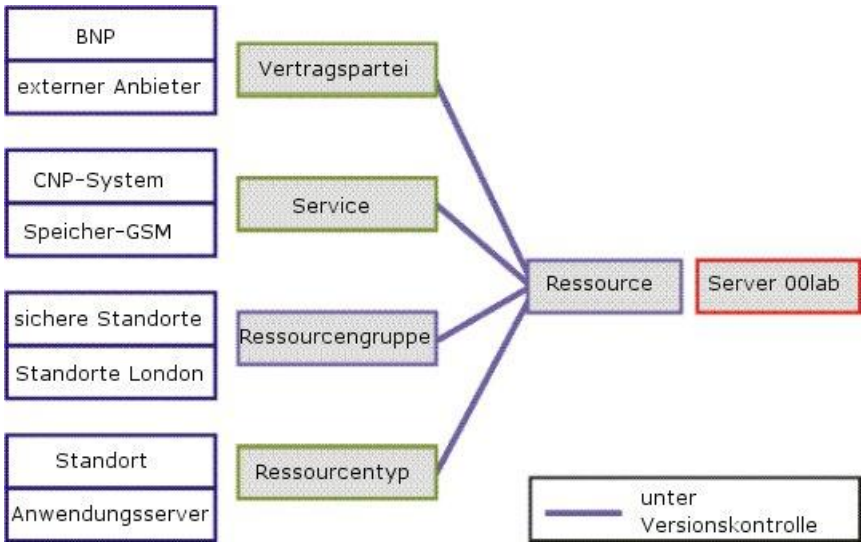
Die für die Aufnahme in die Kalkulationen verfügbaren Ressourcen werden von den Ressourcen, die gegenwärtig im System in Kraft sind, ermittelt (in Relation zum Zeitbereich, der zu diesem Zeitpunkt berechnet wird).

Jetzt wieder zurück zum vorherigen Beispiel:

Gesamt-Ausfallhäufigkeit auf Datencenter-Servern

Das Datencenter kann im System als ein Service dargestellt werden, der dann allen Servern innerhalb der Datenzentrale zugewiesen wird. Es kann auch als eine Ressourcengruppe - dem so genannten "Datencenter-Server" - definiert werden. Dies sind zwei alternative Methoden zur Auswahl der Ressourcenzuweisung in diesem speziellen Fall, es sind jedoch weitere Optionen verfügbar.

Das folgende Diagramm demonstriert, an welche Entitäten eine Ressource angehängt werden kann, sowie deren logische Verwendungen.



Eine Ressourcengruppe kann einen Aspekt der Ressource widerspiegeln, der für Kalkulationen erforderlich ist, wie beispielsweise ihren Speicherort oder die Technologie, die sie enthält.

Der Hauptzweck der Zuweisung von Ressourcen zu diesen Entitäten besteht darin sicherzustellen, dass die Übereinstimmung mit den Berechnungsanforderungen gewährleistet ist, als auch, dass das Modell so dynamisch wie möglich bleibt.

Anwenderspezifische Attribute

Eine andere für eine Ressource verfügbare Funktion ist die Möglichkeit zur Bereitstellung von anwenderspezifischen Attributen. Jeder Ressourcentyp kann über diese anwenderspezifischen, ihm hinzugefügten Attribute verfügen, und wiederum jede Ressource, die als zu diesem bestimmten Ressourcentyp zugehörig definiert wurde, übernimmt ("erbt") diese anwenderspezifischen Attribute.

Unter Verwendung des vorherigen Beispiels für jeden der Datencenter-Server wird auch deren IP-Adresse zugewiesen. Wenn jeder der Datencenter-Server mit dem Ressourcentyp Data Centre Server definiert ist, stellen Sie sicher, dass ein anwenderspezifisches Attribut "IP_Address" dem Ressourcentyp des Datencenter-Servers hinzugefügt wird. Auf diese Art kann dann jede Ressource (Server) mit seiner IP-Adresse über das anwenderspezifische Attribut zugewiesen werden.

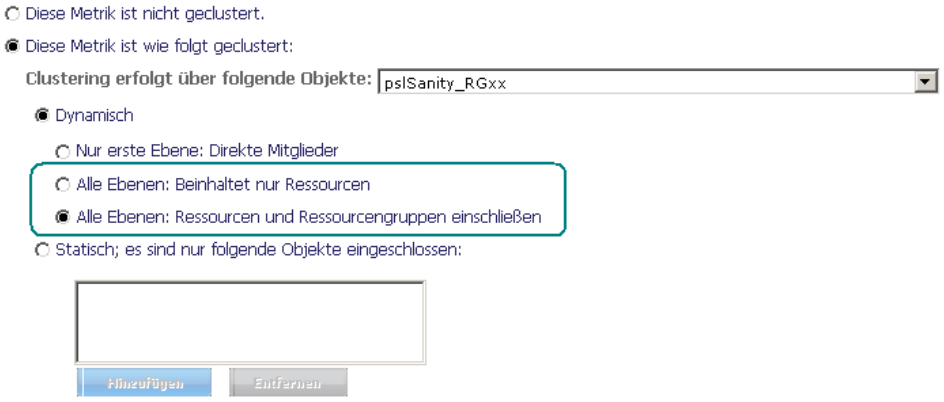
Hinweis: Weitere Details sowie eine Muster-Fallstudie finden Sie unter Anwenderspezifische Fallstudie unter Verwendung von individuellen Attributen. (siehe Seite 250)

Was ist eine geclusterte Metrik?

Eine geclusterte Metrik ist eine Metrik, die zur Berechnung von Service Leveln für eine Gruppe von Ressourcen definiert wurde. Die Berechnungen werden für jede der Ressourcen individuell vorgenommen, ohne dass die Metrik bei jeder Registrierung einer anderen Ressource dupliziert werden muss.

Das Clustern kann nur auf einer Gruppe von Ressourcen ausgeführt werden, denen entweder kein Ziel-Service Level zugeordnet ist oder die das gleiche Ziel-Service Level haben. Beispielsweise sollte die Verfügbarkeit aller Anwendungsserver mindestens 99,9% pro Server betragen. In diesem Fall wird eine Metrik über eine Ressourcengruppe geclustert, die alle Anwendungsserver enthält. Die Engine kalkuliert ein gesondertes Service Level-Ergebnis für jeden Server, bei dem das Ziel 99,9% beträgt.

Zusätzlich zu dieser Art der Clusterung unterstützt CA Business Service Insight eine "Rollup"-Art des Clusterns, die mehrere Ebenen der Ressourcen- und Gruppenberichterstattung mit einer einzigen Metrik zur Verfügung stellt. Dies ermöglicht die Berechnung des Service Levels auf mehreren Ebenen der Ressourcenhierarchie und eine Verfeinerungsfunktion zwischen den verwandten Elementen dieser Ressourcenstruktur. Auf der Registerkarte "Metrik-Clustering" können Sie diese Option durch Auswählen der dazugehörigen Optionen auf dieser Seite aktivieren.



Das Datenmodell des Systems aufbauen

Als Teil des Datenmodellbildungsverfahrens werden erforderliche Komponenten anhand der Datenquelle und der Berechnungsanforderungen identifiziert.

Nachfolgend ist eine Liste der Komponenten aufgeführt, die im Datenmodellbildungsprozess und deren Definitionen identifiziert werden sollten.

Ereignisname Der Name des Ereignisses, wie er in CA Business Service Insight erscheint. Er sollte so aussagefähig wie möglich sein.

Ereignisverhalten Das Verhalten eines entsprechenden Ereignisses: wann von der Datenquelle empfangen, welche Bedingungen, und so weiter.

Zeitstempelfeld Feld in der Datenquelle, welches als Ereignis-Zeitstempel verwendet wird.

Event-Typ-Feld Feld einer in einen Event-Typ zu übersetzende Datenquelle, beschreibt die Art der Berichterstattung. Es ist wichtig, dass die Anzahl der verschiedenen Event-Typen so weit wie möglich minimiert wird, da der Event-Typ manuell definiert wird und der Vorgang im Idealfall nur einmal durchgeführt werden sollte.

Datenfelder Felder einer als Datenfelder abzufragenden Datenquelle.

Ressourcenfeld Feld einer in einer Ressource zu übersetzenden Datenquelle. Enthält ein Element für eine erforderliche Berichterstattung mit einer relativ festen Lebensdauer. Eine Ressource ist ein Element mit einem definierten Lebenszyklus, bei dem Änderungen dynamisch innerhalb des Systems verwaltet werden können. Verweise auf einen Ressourcenlebenszyklus zeigen, wie häufig neue Ressourcen in der Zuordnung zu unterschiedlichen Services oder anderen Elementen der Zuordnung, wie oben erwähnt, hinzugefügt und geändert werden. Das als eine Ressource in CA Business Service Insight zu übersetzende Feld sollte wenige Zuordnungsänderungen und andere Beschränkungen aufweisen.

Und schließlich auf allen oben genannten Definitionen basierend:

Registrierung über Definiert Registrierungskriterium. Das Registrierungskriterium definiert den Event-Typ und die Ressource die Metrikregister. Eine Anfrage für Ressourcen kann direkt durch die Registrierung nach Ressource oder nach Zuordnungselement, wie Service, Vertragspartei, Ressourcengruppe, Ressourcentyp, und so weiter, ausgeführt werden. Diese Definition wird von den Registrierungsfunktionen bestimmt.

Eine andere Methode ist die Registrierung durch eine Metrik, wo die aktuelle Metrik Ausgaben (Service Level-Ergebnis) einer anderen Metrik enthält und sie als Eingabe verwendet. Möglich ist auch die Verwendung der Ergebnisse aus mehreren Metriken als Eingabe.

Richtlinien für das Definieren von Registrierungen

Verwenden Sie die folgenden Leitlinien für das Definieren von Registrierungen:

- Legen Sie niemals die Registrierung nur durch den Event-Typ fest. Auch wenn die Berechnungsanforderung kein Ressourcenfiltern erfordert, fügen Sie die Ressourcenfilterung mindestens auf dem Ressourcentyp hinzu.

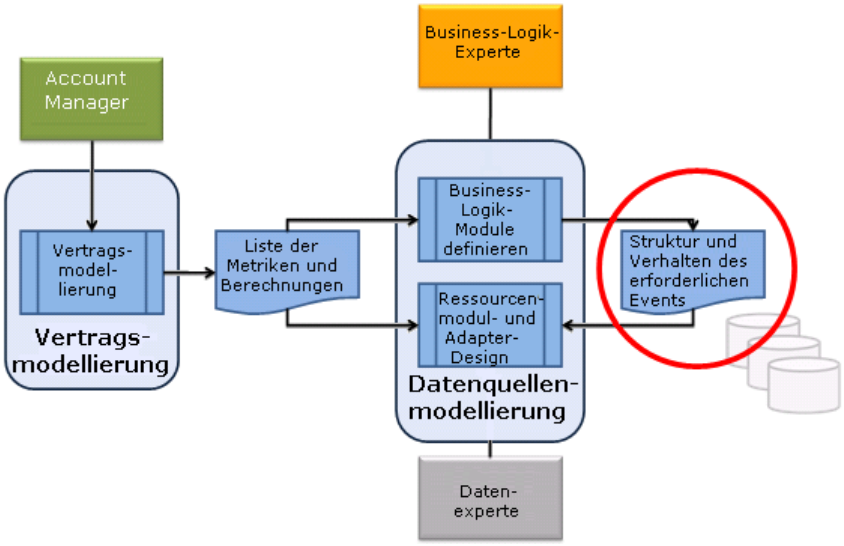
  Beispiel: Bei der Berechnung der gesamten durchschnittlichen Antwortzeiten eines Anwendungsservers muss das Antwortzeit-Event nur dann gemeldet werden, wenn es einem dieser Anwendungsserver zugeordnet ist (d. h., wenn der Ressourcentyp der dem Event zugeordneten Ressource ein "Anwendungsserver" ist). In solch einem Fall kann das System andere Arten von Ressourcen, wie Standorte oder Router haben, die den gleichen Event-Typ zum Senden von Daten verwenden, um so zwischen ihnen zu unterscheiden. Der Ressourcentyp, über den berichtet wird ("Anwendungsserver"), sollte dem Registrierungsbefehl als 3. Parameter hinzugefügt werden.

  Der Grund dafür ist, dass wenn eine Ressourcenänderung eintritt, die Engine die dieser Ressource zugeordnete Metrik als "Erfordert eine Neuberechnung" markiert. In einem Fall, wo die Metrik sich nur durch den Event-Typ registriert, zeigt die Engine die Metrik als zu allen Ressourcen registriert an und markiert sie deshalb für die Neuberechnung bei Aktivierung jeder Ressource. Um dies zu vermeiden, müssen Sie den "optionalen" 3. Parameter des Ressourcentyps verwenden.

- Die effizienteste Methode für eine Registrierung ist über Vertragspartei und Service. Die Ressourcen in dieser Weise anzuordnen aktiviert das Ausdrücken der logischen Beziehung zwischen der Datenschicht und der Businessschicht im System. Um Ressourcen durch diese Elemente zu registrieren, sind keine Änderungen an den Formeln erforderlich, wenn sie in unterschiedlichen Verträgen oder für unterschiedliche Services verwendet werden. Der Metrik-Kontextvertrag und Service definiert die dazugehörige Vertragspartei und den Service. Die in diesem Typ von Registrierung definierten Business-Logik-Formeln sind leicht wiederverwendbar, weil sie keinerlei Änderungen in der Registrierung benötigen.

Hinweis: Registrierungen können auch über die Registerkarte *Registrierung* innerhalb jeder Metrik bearbeitet werden. Diese Schnittstelle stellt für die ausgewählte Metrik einen Assistenten zur Verfügung, der Sie durch den Prozess führt.

Datenmodellierungsphasen-Ausgaben (Datenexperte und Business-Logik-Experte)



Um die Implementierungsstufe der Lösung durchzuführen, ist ein Datenexperte erforderlich, der die folgenden Bedingungen vorweisen kann:

- Ein gründliches Verständnis für die zu verbindenden Datenquellen, einschließlich:
  - Ausgewählte historische Daten
  - Kommunikations-Methoden
  - Verständnis von Schlüsselfeldern jeder für wesentliche Komponenten zu verwendenden Datenquelle des Ressourcenmodells
- Für jede Datenquelle gibt es eine inhärente Liste von Event-Typen, die eine Struktur und das Verhalten beschreiben, einschließlich der folgenden Definitionen:
  - Ereignisname
  - Ereignisverhalten
  - Zeitstempelfeld
  - Event-Typ-Feld
  - Datenfelder
  - Ressourcenfeld

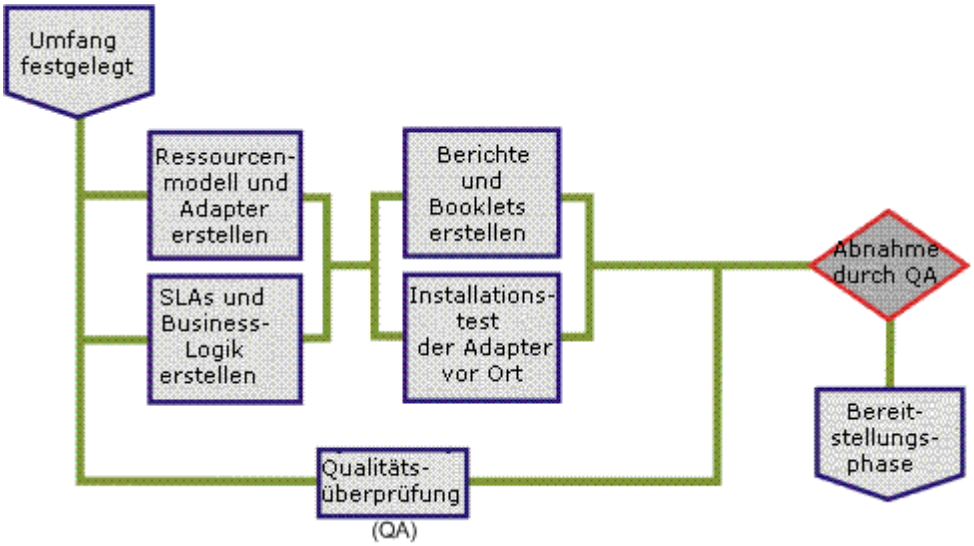
Hinweis: Sie sollten eine Zusammenfassung der Informationen erstellen, die Sie erhalten.

Kapitel 3: Implementierung

Dieses Kapitel enthält folgende Themen:

Implementierung - Einführung (siehe Seite 61)
Den Rahmen aufstellen (Vertragsmanager) (siehe Seite 64)
Einrichten der Vorlagenbibliothek (Vertragsmanager) (siehe Seite 64)
Wie erstellt man Verträge (Vertragsmanager) (siehe Seite 65)
Datenerfassung (Datenexperte) (siehe Seite 71)
Business-Logik-Skripting (Business-Logik-Experte) (siehe Seite 138)
Aktivieren der Verträge (Vertragsmanager) (siehe Seite 183)
Erstellen von lieferbaren Ergebnissen (Vertragsmanager) (siehe Seite 186)

Implementierung - Einführung



Dieses Kapitel erläutert den Prozess und den Gedankengang, der sich hinter der Implementierungsphase des Projekts verbirgt. Wie in der vorhergehenden Abbildung angezeigt, folgt die Implementierungsphase der Designphase, die anschließend der Installations- und Bereitstellungs-Phase folgt.

Das Ziel der Implementierungsphase ist es, dass der Vertragsmanager in der Lage ist, die eigentliche Erstellung aller Elemente und Objekte im CA Business Service Insight-System abzuschließen, die vorher während der Designstufe definiert wurden. Während der Implementierungsphase ist das Team daran beteiligt, wenn das System für die volle Bereitstellung und Installation vorbereitet wird.

Diese Phase sollte nicht gestartet werden, bevor die komplette Designphase abgeschlossen ist und alle notwendigen Ziele sorgfältig abgearbeitet und korrekt definiert wurden. Wenn die Designphase nicht richtig abgeschlossen, oder nicht alle Verträge, Metriken, Adapter und so weiter klar definiert wurden, wird es Probleme mit dem System geben oder Elemente fehlen, die während der Implementierungsphase hätten implementiert werden müssen. Die Implementierungsphase sollte nur nach der abgeschlossenen Design-Prüfung starten.

Es ist auch wichtig, dass die Implementierungsphase vollständig abgeschlossen wird, bevor zur Installation und Bereitstellung des Systems übergegangen wird. Dies sollte nur ausgeführt werden, nachdem eine Qualitätsprüfung ausgeführt wurde.

Der Konfigurationsprozess umfasst die Ausführung der folgenden Schritte durch den Vertragsmanager:

- Einrichtung des Servicekatalogs
- Verträge erstellen
- Verträge aktivieren
- Konfigurieren der Sicherheitseinstellungen
- Erstellen von Berichten/Alarmen/Dashboards

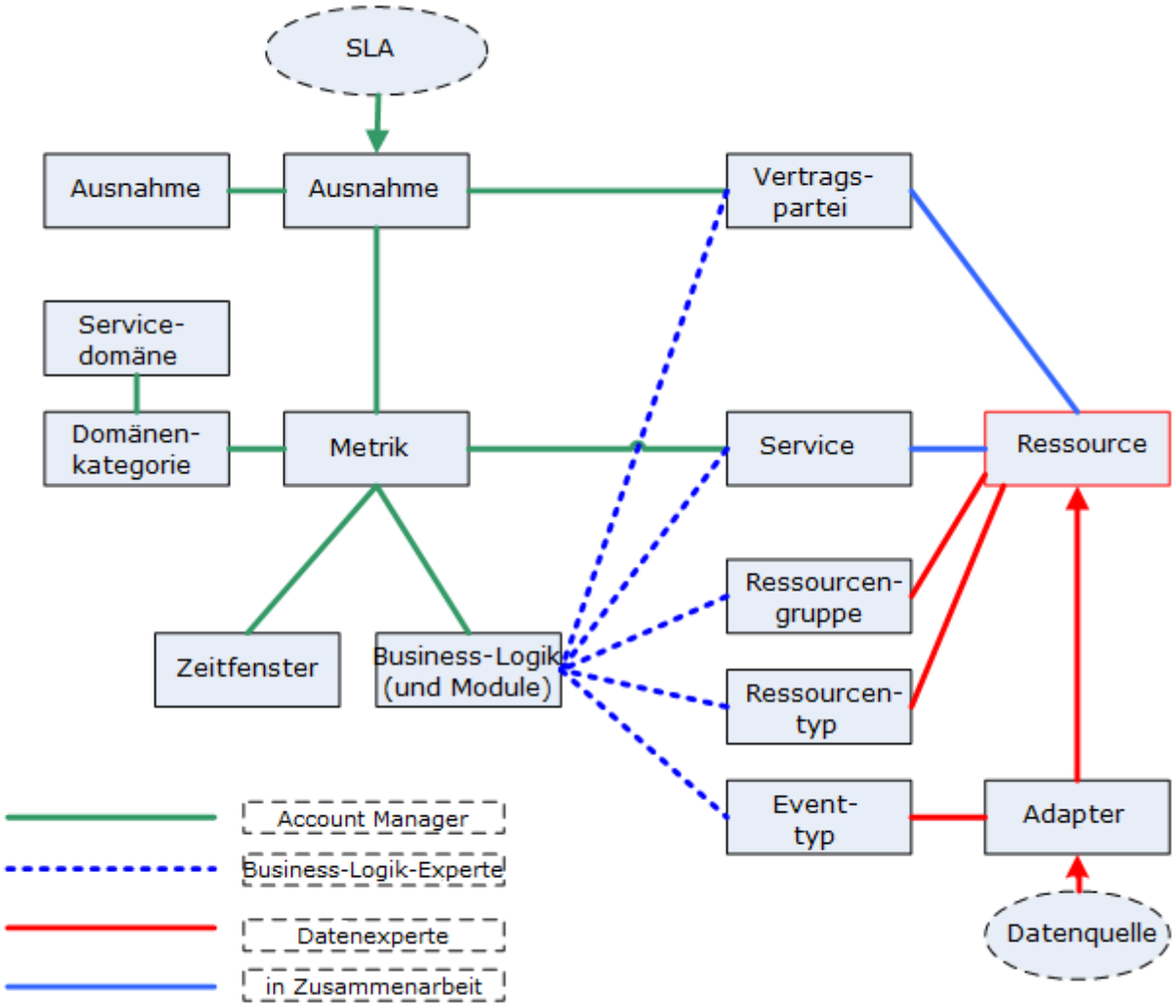
Der Datenexperte sollte die Adapterkonfiguration und -integration mit den Datenquellen ausführen. Der Datenexperte muss auch den Ressourcenübersetzungsschritt ausführen, um die Datenquellenstrukturen mit den innerhalb des CA-Systems angegebenen Elementen zu verbinden. Dieser Schritt ist für den ganzen Prozess entscheidend und kann auch etwas Unterstützung vom Vertragsmanager benötigen.

Zusätzlich ist der Business-Logik-Experte erforderlich, um für alle Metriken, entsprechend den Plänen der Designphase, die Business-Logik zu schreiben. Dies kann die Erstellung aller erforderlichen Module und die Konfiguration zugeordneter Parameter beinhalten, um die erwarteten Kalkulationsfunktionen zur Verfügung zu stellen.

Alle obigen Punkte werden in den Abschnitten innerhalb dieses Kapitels tief greifender erklärt.

Hinweis: Es ist für den Vertragsmanager wichtig sich bewusst zu sein, dass falsche Entscheidungen in diesem Stadium sich ungünstig auf die Funktion von CA Business Service Insight auswirken können und es schwierig oder unmöglich sein kann, dies zu einem späteren Zeitpunkt rückgängig zu machen.

Die folgende Abbildung beschreibt im Großen und Ganzen den logischen Workflow.



Den Rahmen aufstellen (Vertragsmanager)

Der Rahmen ermöglicht:

- Definieren von Services, Servicegruppen, Servicedomänen und Einheiten
- Erstellen und Pflegen der Vorlagen, einschließlich Business-Logik und Zeitfenstervorlagen sowie Business-Logik-Module
- Verwalten benutzerspezifischer Attribute für die Ressourcentypen

Zu diesem Zeitpunkt werden alle systemübergreifenden, während der Designphase identifizierten Elemente, im Rahmen-Abschnitt der Anwendung erstellt. Nur wenn das System diese Rahmenelemente enthält ist es möglich, mit der Erstellung von Verträgen und ihren zugehörigen Metriken fortzufahren.

Die Bildung der Rahmen schließt das Hinzufügen der folgenden neuen Elemente mit ein:

- Services
- Servicegruppe
- Servicedomänen und Domänenkategorien
- Maßeinheiten
- Zeitfenstervorlagen
- Vertragsparteien
- Anwenderspezifische Attribute

Weitere Informationen über jedes der oben genannten Elemente finden Sie in der Onlinehilfe.

Einrichten der Vorlagenbibliothek (Vertragsmanager)

Vorlagenbibliotheken ermöglichen das Definieren und Verwalten von:

- Vorlagenbibliotheken
- Vorlagenordner
- Service Level-Vorlagen
- Vertragsvorlagen
- Sicherheitseinstellungen für Benutzerzugriffsrechte

Weitere Informationen über jedes der oben genannten Elemente finden Sie in der Onlinehilfe.

Wie erstellt man Verträge (Vertragsmanager)

Zu diesem Zeitpunkt werden die Verträge und deren verbundenen Elemente, die in der Designphase definiert wurden, im System erstellt.

Befolgen Sie diese Schritte:

1. Fügen Sie einen neuen Vertrag hinzu, und legen Sie die allgemeinen Details des Vertrags fest.
2. Geben Sie für jeden Vertrag seine Metrik an, und legen Sie ihre allgemeinen Details fest.

Nur die allgemeinen Details des Inhalts eines Vertrags werden zu diesem Zeitpunkt festgelegt, ohne Einschluss der Business-Logik und Clusterung der Metriken des Vertrags.

Die folgende Beschreibung dieser Schritte hebt einige wichtige Punkte hervor, die zu diesem Zeitpunkt berücksichtigt werden sollten. Diese Schritte werden ausführlich in der Onlinehilfe beschrieben.

Schritt 1: Fügen Sie einen neuen Vertrag hinzu, und legen Sie die allgemeinen Details des Vertrags fest.

Die Vertrags-Definition sollte folgende Punkte einschließen:

- Festlegen des Vertragsnamens.
- Auswahl der zugehörenden Vertragspartei(en).
- Einfügen der zugehörenden Services.
- Festlegen der Wirksamkeitsdaten für den Vertrag. Die Wirksamkeitsdaten der Verträge stellen den Datumsbereich dar, für den die Korrelations-Engine den Service Level für diesen Vertrag berechnen wird; Ergebnisse für die Berichterstattung sind nur für diese Daten verfügbar. Bei der Festlegung der Daten ist es wichtig, die Anforderungen in Bezug auf die Verfügbarkeit der vertragsbezogenen Berichte und die verfügbaren Rohdaten zu berücksichtigen.
- Festlegen der Vertrags-Zeitzone und Währung. Diese Definition ist für die Berichterstellung gedacht und sorgt dafür, dass die Berichte zu diesem Vertrag die entsprechende Zeitzone berücksichtigen. Die Währungsdefinition ermöglicht der Berichterstellungs-Engine zu bestimmen, in welcher Währung die Vertragsstrafenergebnisse für die Vertragsstrafen-Formeln angezeigt werden sollen.

Schritt 2: Fügen Sie allgemeine Details der Metriken hinzu.

Sobald der Vertrag fertiggestellt ist, sollten Sie die darin enthaltenen Metriken erstellen. Im Verfahren zur Definition der Metriken, müssen Sie die folgenden Schritte ausführen:

- Festlegen des Namens der Metrik.

- Anwenden der Metrikdetails (Service, Domäne, Einheit, Zeitfenster, Zeitzone, und so weiter).
- Festlegen der Dashboard-Schwellenwerte (siehe Konfigurieren von Dashboard-Seiten (siehe Seite 200)).
- Einfügen zugehöriger Metriken (wenn anwendbar) und die Beziehung zwischen ihnen.
- Definieren der Granularität, bei der die Metrik von der Engine berechnet wird.
- Die Zielvorgabe und Parameter festlegen.

Zu diesem Zeitpunkt der Metrikdefinition fehlen noch die folgenden Elemente:

- Business-Logik-Formel/Modul und Registrierung
- Clusterungsdefinition

Diese Elemente werden nur in die Metrik des Vertrags eingeschlossen, nachdem die Systeminfrastruktur gebildet worden ist und die Business-Logik-Formeln entwickelt und getestet wurden.

Hinweis: Eine alternative Methode zu der obigen Vorgehensweise ist, zuerst die Service Level-Vorlagen innerhalb des Systems zu entwickeln, anstelle der sofortigen Festlegung der Verträge. Dies ermöglicht die Erstellung der Vorlage die dann verwendet werden kann, um weitere Verträge zu erstellen. In einigen Fällen können schon vorhandene Service Level-Vorlagen von einer anderen CA-Instanz in das System importiert werden. Dies kann ein Vorsprung im Vergleich dazu sein, die Verträge ganz neu zu erstellen. Weitere Details zum Erstellen eigener Service Level-Vorlagen finden Sie unter Erstellen von Service Level-Vorlagen (siehe Seite 68).

In einigen Fällen ist es empfehlenswert, zunächst einen Mustervertrag zu erstellen, um zu testen, ob alles richtig und erwartungsgemäß funktioniert. Es ist dann möglich, die Service Level-Vorlage(n) aus diesem Vertrag zu erstellen und im Servicekatalog zu speichern, als eine solide Ausgangsbasis für alle Verträge im System.

Erstellen von Verträgen aus einem Service

Das Erstellen von Verträgen im System durch einen Servicekatalog, entweder aus einer Vorlage heraus oder ohne Vorlage (auf mehreren Service Level-Vorlagen basierend), ermöglicht eine wesentlich höhere Effizienz und hohe Wiederverwendbarkeit für viele unterschiedliche Verträge. Im Allgemeinen enthalten die Service Level-Vorlagen eine Sammlung von vordefinierten und auf bestimmte Servicekomponenten anwendbare Metriken. Falls erforderlich können Sie auch mehr als einen Service (und zugeordnete Metriken) in einer einzelnen Service Level-Vorlage haben. Im Allgemeinen werden die Inhalte einer Service Level-Vorlage in Abhängigkeit davon definiert, wie sie verwendet wird, und sie kann entsprechend den Anforderungen variieren.

Als Beispiel betrachten wir ein Applikationshostingservice, der von einer Organisation angeboten wird. Die Organisation kann seinen Kunden drei unterschiedliche Servicepakete anbieten, wie Bronze, Silber und Gold, je nach dem, was im jeweiligen Paket enthalten ist. Ein gutes Beispiel für die Verwendung von Service Level-Vorlagen könnte sein, eine für jedes der Pakete zu erstellen.

Einmal definiert, können diese Definitionen verwendet werden, um neue Kundenverträge sehr effizient zu erstellen. Zum Beispiel entscheidet sich Kunde ABC für ein Gold-Applikationshostingpaket. Sie können diesen neuen Vertrag im System direkt über die Service Level-Vorlage wie folgt erstellen:

Klicken Sie auf der Vertragsseite auf "Neu hinzufügen > Servicekatalog benutzen", und wählen Sie entweder "Basierend auf Vorlage" oder "Ohne Vorlage". Folgen Sie dann dem Vertragsassistenten, um die Vertragserstellung abzuschließen. Wenn Sie Basierend auf Vorlage auswählen, müssen Sie die Vorlageeinstellungen angeben.

Der Vertragsassistent zeigt eine Liste von Service Level-Vorlagen an und Sie können angeben, welche Service Level-Vorlage Sie in den Vertrag aufnehmen möchten. Beachten Sie, dass es möglich ist, bestimmte Metriken aus mehreren Service Level-Vorlagen auszuwählen oder einfach die ganze Definition für die Aufnahme auszuwählen. In diesem Beispiel alle Metriken aus dem Gold-Hostingpaket. Beachten Sie, dass die Auswahl der obersten Ebene automatisch alle untergeordneten Knoten mit auswählt. Beachten Sie auch, dass es möglich ist, die Metriken einem anderen Service zuzuweisen, falls dies erforderlich ist. Allerdings ist die Standardeinstellung sie den gleichen Servicekomponenten zuzuweisen wie in der Definition zugeordnet.

Sobald alle der erforderlichen Metriken ausgewählt wurden, klicken Sie auf die Schaltfläche Weiter, um diese Metriken auf die neue Vertragsvorlage zu übertragen und zur Eingabe eines Vertragsnamens und allgemeinen Details aufgefordert werden. Klicken Sie auf Speichern und Weiter, um den Vertrag zu erstellen.

Sobald dies abgeschlossen ist, haben Sie die folgenden Optionen:

- Fahren Sie mit dem Vertragsassistenten fort, definieren Sie die Parameter, und starten Sie den Metrik-Assistenten, der die Assistenten-Schnittstelle öffnet und die Benutzeranpassung der Vertragsmetriken ermöglicht. Der Assistent ermöglicht eine Überprüfung und Änderung jeder der Metriken, indem die verfügbaren Felder, wie die Metrikparameter in der Zielvorgabe den Metriknamen, das Zeitfenster und die Beschreibung verändert werden. Sobald dies für jede Metrik abgeschlossen ist, leitet Sie der Assistent zur Seite "Vertragsmetriken" zurück, wo Sie den neuen Vertrag schließen und speichern können.
- Öffnen Sie die Vertragsseite, um den Vertrag anzuzeigen oder zu bearbeiten.

Erstellen von Service Level-Vorlagen

Die Erstellung einer Service Level-Vorlage ist ein ziemlich einfacher Prozess. Aus einem vorhandenen Vertrag (auf der Seite mit Vertragsdetails) wählen Sie alle Metriken aus, die Sie einschließen möchten, und klicken auf "Als Service Level-Vorlage speichern".

Das nächste Fenster fordert Sie zum Benennen der Service Level-Vorlage auf. Sie können dann die Service Level-Vorlage speichern. Wenn gespeichert, werden alle mit den ausgewählten Metriken verbundenen Zeitfenster auf der Registerkarte Zeitfenster eingeschlossen. Von hier aus kann es notwendig sein, die Service Level-Vorlage weiter anzupassen, um sie vollständig flexibel zu machen. Dies könnte das Hinzufügen von Parametern zu den Metriken und das Anzeigen dieser Parameter über die Zielvorgabe für jede Metrik einschließen. Vielleicht auch die Erstellung von Parametern für die Service Level-Vorlagen (ähnlich der Vertragsparametern), auf die sich einige oder alle Metriken dann beziehen können. Nach Abschluss ist die Service Level-Vorlage für den Einsatz in anderen Verträge verfügbar.

Vertragslebens-Zyklus und Versionierung

Sobald ein Vertrag komplett konfiguriert wurde, muss er übernommen werden. Diese Aktion ermöglicht der Berechnungs-Engine, mit der Berechnung der Service Level für alle Metriken im Vertrag zu beginnen. Das Inkraftsetzen des Vertrags verändert den Vertragsstatus von "Vorläufig" (wo er editierbar ist) zu "Wirksam" (wo er nicht mehr editierbar ist). Jede weitere Änderung dieses Vertrages erfordert es, eine neue Version des Vertrages zu erstellen. Werden die gleichen wirksamen Daten für die neue Version gewählt, wird dann, nachdem die Änderungen vollständig abgeschlossen wurden und die neue Version in Kraft gesetzt wurde, die Vorgängerversion komplett überschrieben. Dies löst auch die Engine aus, um die Metriken neu zu berechnen, die sich von der Vorgängerversion unterscheiden. Wirksame Versionen können sich auch teilweise überschneiden, zum Beispiel wenn eine Änderung an den Zielen einiger Metriken im Vertrag teilweise durch die Wirksamkeitsdaten gemacht wird. In diesem Fall wird die alte Version noch verwendet, bis die Wirksamkeitsdaten der zweiten Version aktiv werden. Zu diesem Zeitpunkt nimmt die zweite Version den wirksamen Status für die Berechnung an.

Die folgende Tabelle zeigt Änderungen durch den Benutzer gegenüber der Neuberechnung von Auswirkung, Umfang und Zeitrahmen. Eine Änderung kann sich auf eine bestimmte Metrik innerhalb einer vertragsspezifischen Version auswirken, oder sie kann sich auf Metriken auswirken, die vertragsübergreifend und Vertragsversionen sind.

Änderung	Auswirkung auf Umfang	Auswirkung auf Zeitrahmen
Gemachte Änderungen in Metriken		
Änderung von Metrikdetails - Business-Logik-Formel	Auswirkungen auf jede Metrik innerhalb einer vertragsspezifischen Version	Vom Beginn einer vertragswirksamen Version an
Änderung von Metrikdetails - Zielwert	Auswirkungen auf jede Metrik innerhalb einer vertragsspezifischen Version	Vom Beginn einer vertragswirksamen Version an
Änderung von Metrikdetails - Kontrollzeitraum	Auswirkungen auf jede Metrik innerhalb einer vertragsspezifischen Version	Vom Beginn einer vertragswirksamen Version an
Änderung von Metrikdetails - Metrikparameter	Auswirkungen auf jede Metrik innerhalb einer vertragsspezifischen Version	Vom Beginn einer vertragswirksamen Version an
Änderung von Metrikdetails - Zeitzone	Auswirkungen auf jede Metrik innerhalb einer vertragsspezifischen Version	Vom Beginn einer vertragswirksamen Version an
Änderung von Metrikdetails - Clusterung	Auswirkungen auf jede Metrik innerhalb einer vertragsspezifischen Version	Vom Beginn einer vertragswirksamen Version an

Änderung von Vertragsdetails - Wirksamkeitsdaten	Auswirkungen auf jede Metrik innerhalb einer vertragsspezifischen Version	Vom Beginn einer vertragswirksamen Version an
Änderung von Vertragsdetails - Zeitfenster	Auswirkungen auf jede Metrik innerhalb einer vertragsspezifischen Version	Vom Beginn einer vertragswirksamen Version an
Änderung der Vertragsebenenparameter	Auswirkungen auf jede Metrik innerhalb einer vertragsspezifischen Version	Vom Beginn einer vertragswirksamen Version an
Änderung der SLALOM-Module	Auswirkungen auf jede angehängte Metrik an veränderten Slalommodulen in allen Verträgen und Vertragsversionen	Vom Beginn einer vertragswirksamen Version an
Allgemeine Vorgänge		
Änderung des Metrik-Ressourcenmodells / Änderungssatz (CA Business Service Insight 4.0)	Auswirkungen auf jede Metrik, die Ressource in allen Verträgen und Vertragsversionen registriert	Vom nächsten Status vor dem Datum, in dem Änderung in der Ressource auftraten
Erhalten verzögerter Events für die Metrik Events mit zurückliegendem Zeitstempel (Rohdaten- oder Wiederverwendbarkeits-Events)	Auswirkungen auf jede Metrik, die Ressource in allen Verträgen und Vertragsversionen verwendet	Vom nächstliegenden Stand vor dem Datum, in dem eine Änderung im Event auftrat
Metrik-Korrekturendaten hinzufügen	Auswirkungen auf jede Metrik, die Ressource in allen Verträgen und Vertragsversionen verwendet	Vom nächstliegenden Stand vor dem Datum, in dem eine Änderung in der Korrektur auftrat
Verändern, aktualisieren oder löschen Sie die Metrik-Ausnahmenzeit (Aktivierung & Deaktivierung)	Je nach Ausnahmeeinstellungen und bestimmter Implementierung wirkt sich dies auf eine bestimmte Metrik innerhalb vertragsspezifischer Version aus oder kann sich auf Metriken auswirken, die vertrags- und vertragsversionenübergreifend sind	So nah wie möglich zur Ausnahmezeit
Benutzerspezifische Attributaktualisierung	Auswirkungen auf jede Metrik, die Ressource in allen Verträgen und Vertragsversionen registriert	Vom nächsten Status vor dem Datum, in dem Änderung in der Ressource auftraten

Schließlich einige wesentliche Punkte, um an Vertragsversionen zu erinnern:

- Wenn die neue Version das gleiche Wirksamkeitsdatum hat, werden nur die Metriken neu berechnet, die verändert wurden und die ab Beginn der Version neu berechnet werden.
- Wenn die neue Version unterschiedliche Daten hat, werden all diese Metriken in der neuen Version vom Beginn dieser Version an berechnet und alle Metriken in der Vorgängerversion werden von bestimmten Punkten in dieser Version neu berechnet, bis die neue Version gültig wird. Der genaue Betrag der Neuberechnung hängt von der Statuskonfiguration ab.
- Es wird empfohlen, Verträge mit Wirksamkeitsdaten für 1 Jahr zu erstellen und sie zu verlängern, wenn sie ablaufen. Dies verhindert Neuberechnungszeiträume von mehr als einem Jahr.
- Nicht wirksame Vertragsversionen (das aktuelle Datum liegt nach dem Ende der wirksamen Vertragsdaten) werden berechnet (und somit immer noch als aktive Metriken im System gezählt, da sie für die Berichterstattung mit ihnen verbundene berechnete Service Level-Daten aufweisen).

Die sich mit globalen Variablen innerhalb der Metriken verbindenden Werte werden nicht zwischen Vertragsversionen übertragen (das heißt, die OnLoad-Routine in der Business-Logik wird am Anfang jeder Vertragsversion aufgerufen).

Hinweis: Für eine Reihe von Walk-Through-Szenarien und Falluntersuchungen siehe Contract Modelling Case Studies (siehe Seite 221).

Datenerfassung (Datenexperte)

In der Datenerfassungsphase des Implementierungsprozesses arbeiten Sie mit Adaptern. Die folgenden Themen umreißen diesen Prozess.

Adapter-Funktionalität

Adapter sind Module für die Erhebung von Daten aus Datenquellen und dafür verantwortlich, diese an das CA Business Service Insight-System zu übergeben. Adapter filtern eingehende Daten von der Datenquelle und verändern sie in einer Weise, sodass sie, wenn sie das System erreichen, nur die für die Berechnungen der Service Level in der richtigen Struktur benötigten Informationen enthalten.

Die Adapter-Plattform bietet die Flexibilität um:

- Daten online oder offline mit jeder gewünschten Häufigkeit zu empfangen
- Daten auf verschiedenen Ebenen, roh, berechnet, oder aggregiert zu empfangen
- Daten von einer Vielzahl von Tooltypen zu empfangen

Hauptsächlich besteht jeder Adapter aus zwei Komponenten:

- **Generische Adapter-Komponente**:

  Es gibt zwei Typen der generischen Adapter-Komponente, eine ASCII-Dateiadapterkomponente und eine ODBC-basierte SQL-Adapterkomponente. Diese Komponenten ermöglichen eine Verbindung zu einer Datenquelle und sie als eine ASCII-Datei zu parsen oder eine SQL-Abfrage darauf auszuführen.

- **Adapterkonfigurationsdatei**:

  Jeder Adapter erfordert eine Konfigurationsdatei um zu wissen, wo und wie sie verbunden wird, was empfangen wird, und wie transformiert und die Daten in generische CA-Transaktionen und Events übersetzt werden. CA Business Service Insight liefert jeden generischen Adaptertyp mit einer Standard-XML-Konfigurationsvorlagendatei, die fein abgestimmt werden kann, um Kundeneinzelheiten hinsichtlich der Datenquelle darzustellen, mit der sie eine Verbindung herstellen muss. Die XML-Konfigurationsdatei definiert, welche Felder abgefragt werden sollten, wie sie identifiziert werden sollten, wie sie in die standardisierte Systemdatenbank konvertiert werden sollten, und so weiter.

  Hinweis: Ein Adapterassistent wurde der Benutzeroberfläche hinzugefügt, der eine grundlegende Anpassung dieser XML-Vorlage online ermöglicht. Dies dient dem gleichen Zweck, wie die XML-Konfigurationsdatei für den Adapter zu erstellen. Weitere Informationen über diese Funktion finden Sie in diesem Kapitel.

Die Adapterplattform enthält eine Neustart-/Wiederherstellungsfunktion, die Probleme mit Daten fremder Systeme, wie Abstürze, Netzwerkprobleme, fehlende Daten, doppelte Daten, Datenmüll, Datenlücken, Datenvalidierung etc. bewältigen kann. Jeder Adapter sorgt für volle Datenintegrität und eine lückenlose Verfolgung und Protokollierung aller Adapter-Meldungen und wird später in diesem Handbuch genauer beschrieben.

CA Business Service Insight-Adapter können als ein Service oder als eine Anwendung ausgeführt werden (sichtbar oder nicht sichtbar). CA Business Service Insight-Adapter-Technologie unterstützt fortschrittliche Sicherheitsmechanismen, wie Verschlüsselung, Handshake und Authentifizierungsprozesse.

Es ist wichtig, zu diesem Zeitpunkt zu beachten, dass der Adapterassistent ein Mechanismus ist, um die folgenden Prozesse und Aufgaben zu automatisieren. Während bestimmte Elemente erwähnt werden, können sie bei der Verwendung des Assistenten nicht immer sichtbar sein, sie sind aber "hinter den Kulissen" der Assistenten-Schnittstelle immer noch vorhanden.

Adapter-Umgebung

Die folgenden Elemente beziehen sich auf den Adapter und seine Konfigurations- und Ausführungsparameter.

Datenquelle**:**

Datenquelle, mit der der Adapter eine Verbindung herstellt und aus der er Daten in seinem ursprünglichen Format abruft.

Arbeitsdateien**:**

Vom Adapter erzeugte Ausgabedateien und innerhalb der Prozesse geschriebene Dateien (weitere Informationen finden Sie unter Arbeitsdateien (siehe Seite 77)).

CA Business Service Insight-Adapter-Listener**:**

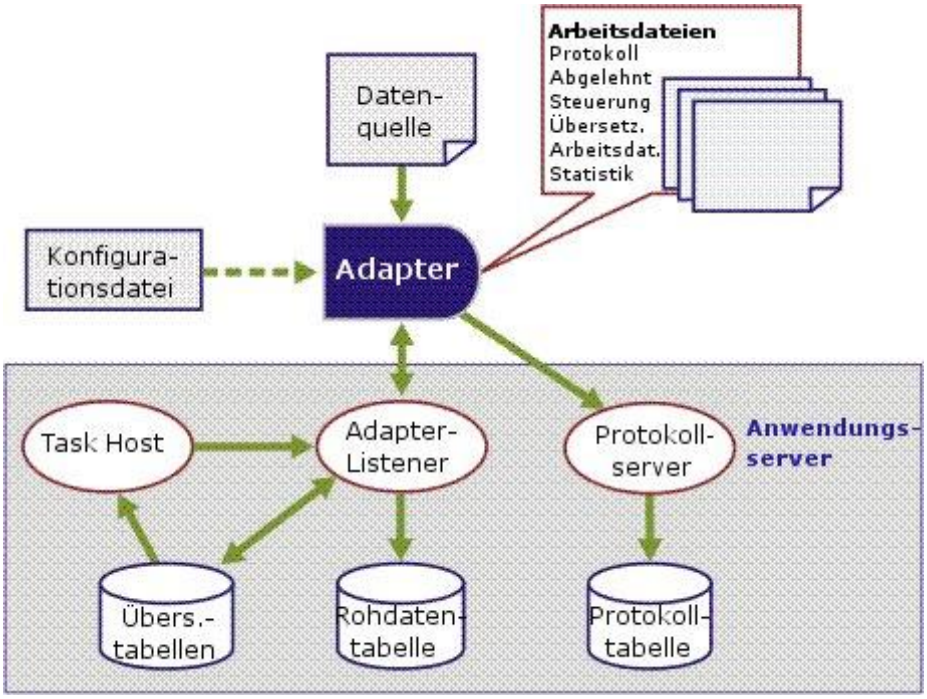
Drei Meldungstypen werden zwischen dem Adapter und dem Adapter-Listener übertragen:

- **Kontrolle**: Start\Stopp\Pause-Meldungen, vom Listener zum Adapter und wieder zurück gesandt, wenn der Adapter seinen Status verändert.
- **Übersetzung**: Adapter sendet Anfragen für den Übersetzungstabelleninhalt und Anfragen für bestimmte Übersetzungswerte. Der Listener gibt die Tabellen und den übersetzten Wert zurück. Der Adapter-Listener erhält die Anzeige vom Task-Host, dass ein Übersetzungseintrag übersetzt wurde. Er sendet dann die Meldung an den Adapter.
- **Rohdaten**: Einheitliche vom Adapter gesandte Rohdaten-Events. Diese Events werden in Paketen gesandt und schließen Bestätigungen ein.

CA Business Service Insight-Protokollserver

Adapter kann konfiguriert sein, um dem Systemprotokoll Protokollmeldungen zu senden und sie in einer lokalen Datei abzuspeichern. Wenn der Port und die IP-Adresse des Protokollservers angegeben und in den Registrierungseinstellungen des Adapters festgelegt werden, sendet der Adapter dann auch Meldungen an den Protokollserver.

Das folgende Diagramm beschreibt den Adapter-Prozess in Beziehung zu jedem der Elemente mit denen er interagiert.



Im Folgenden finden Sie eine Beschreibung der Adapter-Prozessinteraktion mit diesen Elementen:

Konfigurationsdatei**:**

Enthält Einstellungen für einige oder alle Konfigurationsparameter des Adapters. Der Adapter verwendet die Konfigurationsdatei, um die Verbindungsmethode zu bestimmen, die vom Adapter und der Metrik verwendet wird, um die Event-Ausgabe zu erstellen. Dies ist eine XML-Datei, und das Format enthält sechs Grundelemente:

Allgemein:

Verschiedene Adapter-Attribute (Arbeitsverzeichnis, Ausgabedateien, Merker-Fehlersuche).

OblicoreInterface:

Attribute für die Verbindung mit dem CA Business Service Insight-Server.

DataSourceInterface:

Attribute für die Verbindung mit der Datenquelle (Dateipfad und Muster, Verbindungszeichenfolgen, SQL-Abfragen usw.)

InputFormatCollection:

Parsen der Metriken für das Analysieren und für die Bearbeitung der Quelldaten.

TranslatorCollection:

Metriken für die Erstellung des einheitlichen Events, kombiniert mit den geparsten und bearbeiteten Datenfeldern.

TranslationTableCollection:

Metriken für Zuordnung von Daten zwischen ursprünglichen Daten und CA Business Service Insight-Elementen.

Jeder dieser sechs Abschnitte enthält alle dazugehörige Informationen, die es dem Adapter ermöglichen, eine Verbindung mit der Datenquelle herzustellen, die erforderlichen Informationen zu erhalten, sie in einheitliche CA Business Service Insight-Event-Strukturen zu parsen und sie in der CA Business Service Insight-Rohdaten-Tabelle zu speichern.

Hauptdateien

Der Adapter besteht aus zwei Hauptdateien: die ausführbare Datei und die Konfigurationsdateien. Die ausführbare Datei ist eine generische Datei. Es gibt zwei solche ausführbare Dateien: SQL-Adapter und Dateiadapter.

Eine XML-Konfigurationsdatei wird für jeden Adapter zugeschnitten, um die bestimmten Datenquellenanforderungen zu speichern. Die Konfigurationsdatei gibt an, dass sich die Informationen auf die Datenquelle (Name, Speicherort, Verbindungsmethode und Struktur) und die Struktur der Ausgabe-Events beziehen, die vom Adapter generiert worden sind.

Die Konfigurationsdatei schließt innerhalb einer vordefinierten strukturierten XML-Datei jene Parameter und Werte ein, die für Attribute festgelegt werden.

Wenn Sie einen neuen Adapter erstellen, ist es notwendig, die vorhandene dazugehörige ausführbare Datei zu verwenden (entsprechend dem Zieldatenquellentyp, Datei für flache Dateidatenquellen, SQL für Datenbankdatenquellen) und dann die Konfigurationsdatei, wie benötigt, zu ändern. Die zwei Strukturen enthalten leicht unterschiedliche Konfigurationselemente für einen Text- oder SQL-Adapter. Dies wird im Allgemeinen automatisch eingerichtet, wenn der Adapter mit dem Dienstprogramm "Adaptermanager" erstellt wird.

Andere auf Adapter bezogene Dateien sind Arbeitsdateien, die vom Adapter im Prozess des Lesens der Datenquelle und des Schreibens der Events zum CA Business Service Insight-System erstellt wurden.

Hinweis: Für weitere Informationen über die Anpassung der Konfigurationsdatei siehe Eigenschaften der Adapterkonfiguration (siehe Seite 321).

Arbeitsdateien

Die Adapter-Arbeitsdateien werden erstellt, wenn der Adapter das erste Mal ausführt wird und werden bei jedem Adapterstart immer wieder aktualisiert.

Jeder Adapter hat seine eigenen Arbeitsdateien. Die Namen der Arbeitsdateien können in der Adapterkonfigurationsdatei festgelegt werden (optional), oder sie können ihre Standardnamen beibehalten. Der Speicherort der Arbeitsdatei wird vom "Arbeitsordner" festgelegt und kann auch in der Konfigurationsdatei festgelegt werden. Beachten Sie, dass der angegebene Pfad relativ zum aktuellen Verzeichnis sein kann, innerhalb dessen sich der Adapter befindet. Der angegebene Pfad muss schon vorhanden sein (oder Sie müssen es erstellen), damit der Adapter richtig ausgeführt werden kann.

Hinweis: Der Ordner wird nicht automatisch erstellt, falls er nicht vorhanden ist.

Alle dazugehörigen Parameter in der Konfigurationsdatei sind im Abschnitt "Allgemein" enthalten. Nur der Speicherort der Protokolldatei wird in der Registrierung festgelegt oder wird über der Befehlszeile weitergegeben.

AdapterStatistics.txt

Der Adapter speichert statistische Informationen in dieser Datei in Minutenabständen ab. Die Schlusszeile wird geschrieben, wenn der Adapter stoppt. Jede Zeile in der Datei enthält Zahlen aus:

- Aufgenommenen Events
- Ignorierten Events
- Events mit Fehlern
- Gesendeten Meldungen
- Gesendeten Paketen

Jedes Mal, wenn der Adapter startet, initiiert er die Statistik.

RejectedEvents.txt

Diese Datei enthält alle Events, die der Adapter fehlerhaft an CA Business Service Insight sendet, weil der Event-Wert, definiert als für die Übersetzung benötigt, keine passende ID in der Übersetzungstabelle hat. Dies heißt, dass die dazugehörige Übersetzung nicht ausgeführt wurde. Jedes Event das mindestens einen auf die Übersetzung wartenden Wert hat, wird in der Datei rejectedEvents abgespeichert.

Zu Beginn eines jeden Starts versucht der Adapter zuerst, die Events der Datei rejectedEvents an CA Business Service Insight zu senden, während er außerdem versucht, die passende ID für den dazugehörigen Wert in der Übersetzungstabelle zu finden. Wenn der Wert festgestellt wird, sendet der Adapter das Event und löscht es aus der Datei. Wenn ein passender Wert nicht gefunden wird, bleibt das Event in der Datei rejectedEvents stehen.

Sie können die Obergrenze für die Anzahl abgelehnter Events konfigurieren, indem Sie den Parameter RejectedEventsUpperLimit in der Adapterkonfigurationsdatei festlegen. Wenn das Limit erreicht wird, hört der Adapter auf, neue Datensätze zu lesen und gibt einen "gesperrten" Status ein. Dies können Sie sehen, wenn die Debugausgabe während der Adapter-Ausführung auf dem Bildschirm angezeigt wird. Wenn Sie eine kontinuierliche Zeichenfolge vom groß geschriebenen Buchstaben "B" sehen, wird der Adapter gesperrt und erfordert eine Übersetzung einiger ausstehender Einträge, bevor er weitere Daten lädt.

Die ausstehenden Events werden in der Datei in einem XML-Format abgespeichert. Nachfolgend ist ein Beispiel eines Events aus der Datei:

```
<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>
```

Adapterprotokoll

Das Adapterprotokoll ist die Datei, in die der Adapter Protokollmeldungen abspeichert.

Es wird empfohlen, sich die Adapter-Protokolldatei anzuschauen, die das Protokoll-Browser-Hilfsprogramm verwendet.

Es ist möglich in der Adapterkonfigurationsdatei LogDebugMode, die Ebene der Berichterstattung dieser Protokolldatei durch das Ändern eines Parameters festzulegen. Wenn auf "ja" gesetzt ist, schreibt der Adapter normale Anzeigenmeldungen in das Protokoll, sowie den ursprünglichen Datensatz, das Parsingergebnis und das bestimmte Event.

Dieser Parameter wird üblicherweise auf "ja" festgelegt, während der Adapter testet und überwacht.

Standardmäßig ist die Dateigröße auf 1 MB beschränkt. Wenn dieses Limit erreicht wird, verändert der Adapter seinen Namen, hängt die Endung "_old" an und erstellt eine neue Protokolldatei. Der Adapter kann potenziell bis zu 2 MB an Protokollmeldungen speichern; 1 MB für die alte Datei und 1 MB für die aktuelle Datei.

Das Größenlimit der Protokolldatei kann als Eintrag in der Registrierung für jeden Adapter mit einem Maximum von 10 MB konfiguriert werden. Dieser Eintrag in der Registrierung wird LogFileMaxSize genannt und wird unter dem betreffenden Adapter mit seinem Wert, der ein Vielfaches von KB definiert, angegeben.

DataSourceControl.xml

Die Datei DataSourceControl.xml wird vom Adapter verwendet, um seinen Zugriff zur Datenquelle zu steuern und um sicherzustellen, dass er, wann auch immer er ausführt wird, immer weiter vom letzten Punkt an fortfährt, zu dem er liest.

Die Datei "Adapter" behält den Namen der letzten gelesenen Datei, der letzten gelesenen Zeile und die Position in der Datei die er erreichte bei. Das nächste Mal wenn der Adapter läuft, greift er auf die Datei von dem Speicherort zu, welcher in den Informationen in der Datei DataSourceControl.xml gefunden wird. Bei der Verwendung dieses Mechanismus, kann der Adapter nur neue Datensätze bei jedem Start lesen.

Der Adapter funktioniert nicht direkt auf den Quelldateien, sondern kopiert zuerst die aktuelle Datei zu einer Arbeitsdatei. Daher wird die gleiche Information für die Arbeitsdatei und für die Quelldatei beibehalten. Wenn die Quelldatei angehängt wird, werden nur neue Datensätze zur Arbeitsdatei kopiert.

Wenn der Adapter so konfiguriert wird, die Quelldatei zu löschen sobald sie verarbeitet wurde, indem der Parameter in der Konfigurationsdatei DeleteAfterProcessing auf "ja" festgelegt ist, speichert er die Information nicht in der Quelldatei. Nach Abschluss liest er eine neue Datei, die in dem Arbeitsordner existiert, der mit dem in der Konfigurationsdatei angegebenen Dateimuster übereinstimmt.

Nur wenn DeleteAfterProcessing auf "nein" festgelegt wird, überprüft er auf neue Datensätze in der letzten Datei. Wenn es keine gibt, geht er zur nächsten Datei in lexikografischer Reihenfolge. Damit die Quelldatendateien in der richtigen Reihenfolge gelesen werden können, sollten Sie konsequent auf eine aufsteigende, sequenzielle Namensgebung achten. Hängen Sie zum Beispiel den Dateien einen umgekehrten Datumswert (yyyymmdd-hhmmss) an, um dies zu unterstützen. Zum Beispiel:

- DataSourceABC20070517-14:00:00.csv
- DataSourceABC20070517-15:30:00.csv
- DataSourceABC20070517-17:00:00.csv, und so weiter.

Nachfolgend finden Sie ein Beispiel einer DataSourceControl.xml für einen Dateiadapter.

```
<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>
```

Ein SQL-Adapter behält den letzten Wert der Abfrageschlüsselfelder für jede ausgeführte Abfrage bei. Die Schlüsselfelder sind einmalige Bezeichner für Datensätze in der Zieldatenbanktabelle. Der Adapter verwendet diese Werte beim Erstellen der Abfrage für den nächsten Start. Dies ermöglicht dem Adapter, nur neue Datensätze zu lesen.

Betrachten wir zum Beispiel die folgende SQL-Anweisung für das Abrufen einiger Trouble-Ticket-Daten.

Wählen Sie ticket_id, status, organization, open_date, respond_date, resolved_date, record_modified_date from t_ticket_data;

In diesem Beispiel wurde das Abfrageschlüsselfeld ausgewählt, um alle neuesten Datensätze von der Datenquelle zu erfassen, um sicherzustellen, dass die neueste Information record_modified_date ist, da sie neue Tickets erzeugt, die seit der letzten Adapterausführung angesprochen wurden, sowie die Aktualisierung vorhandener Tickets durchführt. Durch das Auswählen dieses Feldes statt des Abfrageschlüsselfelds, hängt der Adapter automatisch den folgenden Abschnitt ans Ende der Abfrage während Ausführung an:

wo record_modified_date > :previous_value-Reihenfolge von record_modified_date asc

Folglich ruft er nur die neueren Aufzeichnungen ab. Beachten Sie, dass es eine Reihe von Überlegungen bei der Wahl des Abfrageschlüsselfelds gibt und dies beim Auswählen immer vom Verhalten der Datenquelle und dem abhängt, was Sie versuchen mit den gefundenen Daten zu erreichen. Beachten Sie auch, dass die im vorherigen Beispiel gewählten Felder nicht immer die beste Wahl für *jede* Situation sind.

Ein Beispiel einer DataSourceControl.xml-Datei für einen SQL-Adapter wird unten angezeigt.

```
<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

Alle Events die erstellt werden und zur Sendung an CA Business Service Insight bereit sind, werden zuerst in die Datei send geschrieben.

SendControl.xml

Die Datei sendcontrol.xml enthält alle Zeilen, die gesandt und von CA quittiert wurden.

Die Datei ermöglicht dem Adapter, dem Schiebefenster-Quittierungsprotokoll zu folgen, um die Daten nach CA zu übertragen. Für weitere Informationen über diesen Mechanismus siehe Adapter-Kommunikation (siehe Seite 83).

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

IllegalEvents output file (.txt)

Der Adapter schreibt in die Datei IllegalEvents alle Datensätze die gelesen wurden, aber ein Fehlerparsing hatten. Dies wird üblicherweise durch die in der Adapterkonfigurationsdatei eingegebenen Validationslogik verursacht. Der Adapter speichert diese Datensätze wenn der Parameter SaveIllegalEvents in der Konfigurationsdatei auf "ja" gesetzt wird. Beachten Sie, dass der Pfad für diese Option auch durch die Verwendung des "IllegalEventsDirectoryName" festgelegt werden muss. Dieser Ordner muss schon vorhanden sein, da er nicht automatisch erstellt wird. Wenn der Ordner nicht vorhanden ist, meldet der Adapter einen Fehler sobald er gestartet wird.

Innerhalb eines Dateiadapters hat die Datei mit den fehlerhaften Aufzeichnungen den gleichen Namen wie die Quelldatei, während sie im SQL-Adapter den Namen der Abfrage trägt.

Translations.xml

Die Datei translation.xml beinhaltet die Adapterübersetzungs-Tabellen.

Wenn der Adapter in Onlinemodus ausführt wird, enthält die Datei eine Kopie der Übersetzungstabelle von der Datenbank. Wenn die Übersetzungstabelle als Remote konfiguriert wird, lädt der Adapter die Übersetzungstabelle von der Datenbank in diese Datei und ersetzt sie. Wenn eigenständig, liest er die lokale Datei.

Wenn der Adapter im Offlinemodus läuft, benutzt er die Datei nur als seine Übersetzungstabelle (weitere Informationen über Online-/Offlinemodi finden Sie unter Ausführungsmodi des Adapters (siehe Seite 88))

```
<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>
```

Adapter-Kommunikation

Adapter interagieren auf der einen Seite mit der Datenquelle und dem CA Business Service Insight-Adapter-Listener und Protokollserver auf der anderen, wie im folgenden Diagramm dargestellt.

Der Adapter kommuniziert mit der Datenquelle zum Abrufen der Daten über eine ODBC-Verbindung und kann sich entweder lokal oder entfernt von der Datenquelle befinden, solange der Adapter die ODBC-Verbindung feststellen kann.

Der Adapter kommuniziert mit dem CA Business Service Insight-Anwendungsserver über das TCP/IP-Protokoll und kann sich deswegen lokal oder entfernt davon befinden, solange er die TCP/IP-Verbindung herstellen kann.

Der Adapter muss zwei Ports geöffnet haben, einen für den Adapter-Listener und einen für den Protokollserver. Die Adapter-Listener-Ports müssen einmalig pro Adapter sein und dürfen nicht mit anderen Netzwerkoperationen oder Anwendungen im Konflikt stehen, die auch diese Ports verwenden können. Zum Beispiel sollten Sie den Port 1521 nicht verwenden, da dieser Port im Allgemeinen vom Oracle-TNS-Protokoll für die Kommunikation zur Datenbank, und so weiter, verwendet wird. Vielleicht müssen Sie auch lokale Firewalls berücksichtigen, die diesen Datenverkehr sperren können.

Hinweis: Erkundigen Sie sich bei Ihrem lokalen Administrator, wenn Sie nicht sicher sind, welche Ports für Verwendung verfügbar sind, oder falls Sie eine Öffnung von Ports benötigen, um diese Kommunikation zu ermöglichen.

Der Port und die Adresse des Adapter-Listeners wird in der Adapterkonfigurationsdatei festgelegt. Der Port und die IP-Adresse des Protokollservers wird über die Einträge des Adapters in der Registrierung festgelegt.

Die Client/Serveroperation hinsichtlich des Adapter-Listeners ist konfigurierbar, was es ermöglicht den Adapter so zu konfigurieren, um entweder als ein Client oder als ein Server zu funktionieren. Die Konfiguration der Client/Serveroperation wird auf der Adapter-Seite in den Parametern der Konfigurationsdatei ausgeführt. Dazu müssen der Port, die Adresse und die ConnectionInitiator-Variablen dementsprechend festgelegt werden.

Wenn der ConnectionInitiator festgelegt wurde, der Adapter zu sein, wird nur ein Zielport benötigt. Wenn er festgelegt wird, CA Business Service Insight zu sein, dann ist ein Port und eine IP-Adresse des Adapter-Listeners auf CA Business Service Insight erforderlich. Standardmäßig wird der Server festgelegt der Adapter zu sein. Dies ist manchmal eine wichtiges Merkmal, um eine Firewallregel zu aktivieren, um ausgelöst zu werden (eine als Port Triggering bekannte Funktion). Manchmal erlaubt eine Firewall nur eine innere Anfrage auf einem Port, wenn eine Meldung aus dem "Inneren" der Firewall auf dem gleichen Port gesendet wurde. Sie löst dann die Firewall aus, um die Kommunikation zu ermöglichen.

Hinweis: Konsultieren Sie Ihren Netzwerkadministrator für weitere Informationen über die lokalen Bedingungen, die sich auf die Adapter-Kommunikation auswirken können.

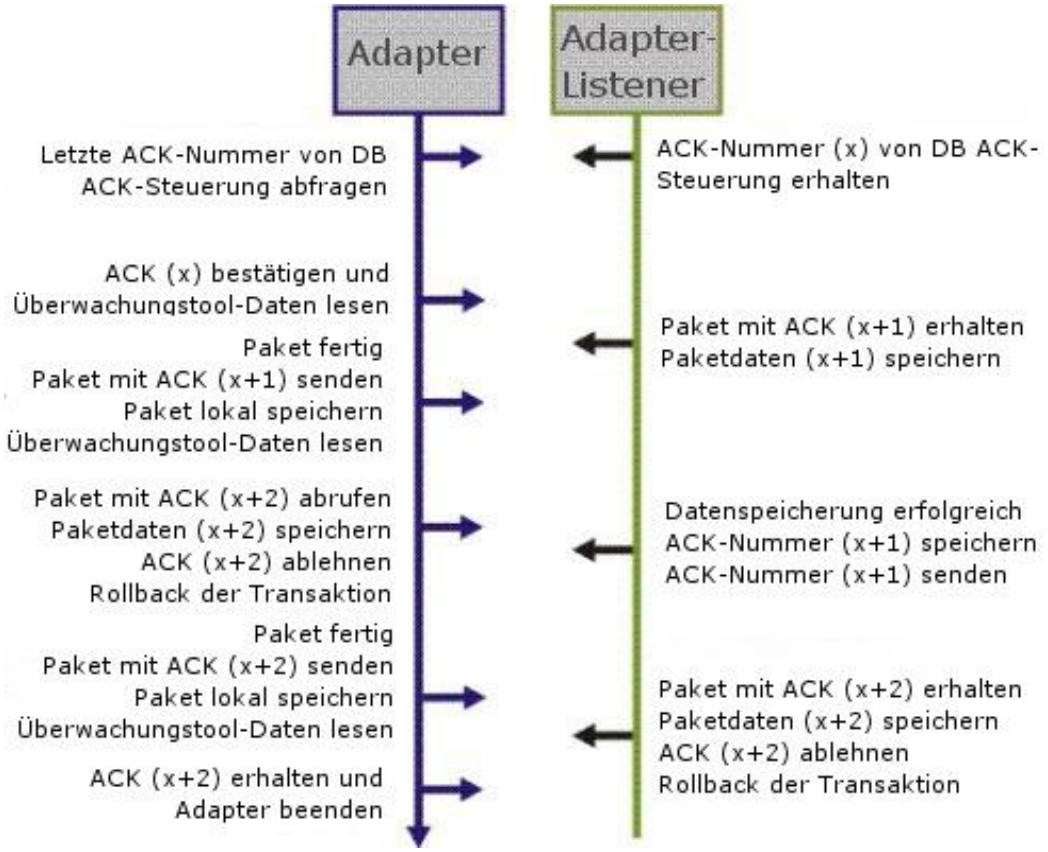
Aus sicherheitstechnischer Sicht ist es empfehlenswert, dass der Adapter festgelegt wird der Client zu sein, da dies das Ziel von Events sichert, wenn in einer mehrfachen Bereitstellungsumgebung für Test und Produktion gearbeitet wird.

Um den Übertragungserfolg von Datensätzen vom Adapter zum CA Business Service Insight-Adapter-Listener zu überprüfen, integriert der Adapter einen ACKs/Schiebefensteralgorithmus über die TCP/IP-Schicht. Dieser Algorithmus sendet hauptsächlich die Daten in Paketen und wartet dann auf eine Bestätigung vom Adapter-Listener, bevor er das nächste Paket bewegt. Jedes Paket enthält einige Rohdatenmeldungen. Die Anzahl von Meldungen in einem Paket kann durch das Festlegen des Paketgrößen-Parameters konfiguriert werden. Jedes Paket hat eine Sequenz, die in der Bestätigungsmeldung enthalten ist. Alle dazugehörigen Parameter, die den Prozess kontrollieren, sind im CA Business Service Insight-Schnittstellenabschnitt der Konfigurationsdatei enthalten. Im Allgemeinen jedoch ist es nicht erforderlich, diese Parameter zu ändern.

Der Listener des Adapters schreibt die Rohdaten im Paket in einer einzelnen Transaktion.

Hinweis: Die ACK-Operation kann nur auf den an CA Business Service Insight gesandten Rohdatenmeldungen ausgeführt werden.

Die folgende Abbildung zeigt den Kommunikationsprozess des Adapters.



Adapter-Registrierungseinstellungen

In Fällen, in denen die Information in der Befehlszeile fehlt, verwendet der Adapter einige Definitionen, die in der Systemregistrierung auf dem Server gespeichert sind, auf dem der Adapter installiert wurde.

Die Registrierungseinträge werden vom Adaptermanager-Dienstprogramm geschrieben, wenn das Dienstprogramm für die Adapter-Installation verwendet wurde. Wenn es nicht für die Adapter-Installation verwendet wurde, können diese Eingaben von Hand der Registrierung hinzugefügt werden.

Hinweis: Wenn ein Adapter in einer UNIX-Umgebung installiert wird, müssen diese Einträge von Hand hinzugefügt werden, da es keinen Adaptermanager für diese Umgebung gibt.

Unten sind die vom Adapter und dem Adaptermanager verwendete Registrierungseinträge aufgelistet.

Allgemeine Server-Einträge

Die folgenden Einträge werden in die \HKEY_LOCAL_MACHINE\SOFTWARE\Oblicore\Adapters registry geschrieben:

Mögliche Eigenschaften:

Name	Typ	Beschreibung
Adapterverzeichnis	String	Stammverzeichnis für alle Adapter.*
FileAdapterConfTemplate	String	Dateiadapter-Konfigurationsvorlagepfad.* Der Adaptermanager verwendet diese Information, um eine Konfigurationsvorlage zum Ordner des neuen Adapters zu kopieren, wo angegeben ist, wie Teile der Erstellungsvorgänge des Adapters zu verarbeiten sind.
GenericFileAdapter	String	Dateiadapter, ausführbare Datei.* Entweder erstellt der Adaptermanager eine Verknüpfung zu einer ausführbaren Datei, oder kopiert sie zum Ordner des neuen Adapters, um anzugeben, wie Teile der Erstellungsvorgänge des Adapters zu verarbeiten sind.
GenericSqlAdapter	String	SQL-Adapter, ausführbare Datei.* Entweder erstellt der Adaptermanager eine Verknüpfung zu einer ausführbaren Datei, oder kopiert sie zum Ordner des neuen Adapters, um anzugeben, wie Teile der Erstellungsvorgänge des Adapters zu verarbeiten sind.

Name	Typ	Beschreibung
LogServerAddress	String	Protokoll-Server-Netzwerkadresse. (Optional)** Protokoll-Serverport, üblicherweise 4040. (Optional)** In Fällen, in denen diese Parameter festgelegt sind, meldet der Adapter dem CA Business Service Insight-Protokollserver Protokollmeldungen.
LogServerPort	String	
SqlAdapterConfTemplate	String	SQL-Adapter-Konfigurationsvorlagepfad.* Der Adaptermanager verwendet diese Information, um eine Konfigurationsvorlage zum Ordner des neuen Adapters zu kopieren, wo angegeben ist, wie Teile der Erstellungsvorgänge des Adapters zu verarbeiten sind.

* Nur vom Adaptermanager-Hilfsprogramm verwendet

** Vom Adapter verwendet

Individuelle Adapter-Einträge

Die folgenden Einträge werden in der Registrierung \HKEY_LOCAL_MACHINE\SOFTWARE\Oblicore\Adapters\AdapterName geschrieben:

Mögliche Eigenschaften:

Name	Typ	Beschreibung
ConfigurationFileName	String	Adapterkonfigurationsdateiname.**
Verzeichnis	String	Adapterverzeichnis*
LogFileName	String	Adapter-Protokolldateiname.**
Pfad	String	Adapter, ausführbarer Dateipfad*
Starten als	Zahl	Betriebsmodus.* Service/Konsolenanwendung/Windows-Anwendung
Typ	Zahl	Adaptertyp.* Datei/SQL
LogFileMaxSize	Zahl	Wert ist eine Zahl in KB.** Erlaubter Bereich ist 1.000-100.000 und der Standardwert ist 1.000.

* Nur vom Adaptermanager-Hilfsprogramm verwendet

** Vom Adapter verwendet

Ausführungsmodi des Adapters

Der Adapter kann ausgeführt werden durch:

Service**:**

Der Adapter kann als ein regulärer Windows-Service installiert werden. Dies ermöglicht dem System, seinen Status zu steuern (Starten, Pause, Stopp, Automatik) als wäre es ein normaler Windows-Service.

Die Installation des Adapters als ein Windows-Service, der durch die ausführbare Adapterdatei von der Befehlszeile aus durch die Verwendung des **-i,** als Service installiert wird und durch **-u** deinstalliert wird.

Anwendung**:**

Ausführen der ausführbaren Adapterdatei über die Befehlszeile. Die Adapter-Befehlszeile kann in der folgenden Weise gestartet werden:

Befehlszeilen-Optionen:

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

Parameter	Funktion
-i	Installiert den Service
-u	Deinstalliert den Service
-v	Zeigt die Version
-d	Führt den Adapter als Konsolenanwendung aus
-t	Nur überprüfen -Überprüft die Konfigurationsdatei und stoppt
-f	Legt den Konfigurationsdateinamen fest
-l	Legt den Protokolldateinamen fest
-n	Legt den Servicenamen fest
-a	Legt die Anwendungsserveradresse fest
-p	Legt die Anwendungsserver-Portnummer (1024-49151) fest
-la	Legt die Protokollserveradresse fest
-lp	Legt die Protokollserver-Portnummer (1024-49151) fest

Diese Art der Ausführung wird üblicherweise in Projekten verwendet. Dies ermöglicht die Ausführung des Adapters über eine **.bat** -Datei und ermöglicht auch, den Windows-Planer zu verwenden, um das Timing der Adapter-Ausführung zu steuern. Um den Adapter über den Windows-Planer zu planen, ist es für einen einmaligen Start notwendig, seinen ausführenden Modus zu konfigurieren.

RunOnce: (optional [ja/nein]). Wenn in der Konfigurationsdatei auf "nein" gesetzt wird, läuft der einmal ausgeführte Adapter kontinuierlich. Wenn er auf "ja" gesetzt ist, startet der Dateiadapter, liest Datensätze und stoppt automatisch, wenn keine neuen Datensätze angezeigt werden. Ein Dateiadapter liest die ganze Datei, wartet dann ein paar Sekunden und versucht die neuen Datensätzen zu lesen (je nach SleepTime-Einstellungen). Wenn es keine neuen Datensätze gibt, stoppt er. Ein SQL-Adapter durchläuft jede Abfrage nur einmal. Wenn RepeatUntilDry auf "nein" festgelegt wird, hört er sofort auf. Wenn RepeatUntilDry auf "ja" festgelegt wird, wartet er (abhängig von SleepTime). Er versucht, die Abfrage nochmals zu durchlaufen (je nach Ruhezeit der Abfrage). Wenn es keine neuen Datensätze gibt, wird er gestoppt.

Weitere Informationen über die Attribute SleepTime und RepeatUntilDry finden Sie unter Adapterkonfigurations-Spezifikationen (siehe Seite 321).

Der CA Business Service Insight-Schnittstellen-Abschnitt der Konfigurationsdatei besteht aus Attributen, die zwei Verbindungsmodi in CA Business Service Insight angeben: online und offline.

Im Onlinemodus verbindet sich der Adapter mit CA Business Service Insight, ruft die Übersetzungstabellen ab und steuert die Befehle von CA Business Service Insight. Dann sendet er Events, Status, und Übersetzungsanfragen an CA Business Service Insight zurück. Im Offlinemodus arbeitet der Adapter mit einer lokalen Übersetzungstabellendatei und schreibt die Events in eine Ausgabe-Datei.

Der Offlinemodus wird üblicherweise bei der ersten Entwicklung eines Adapters und für Tests verwendet.

Wenn Sie ConsoleDebugMode mit der Einstellung "ja" verwenden, können Debug-Meldungen auf der Konsole angezeigt werden

Weitere Informationen zu den verschiedenen Indikatoren, speziell zu dem Attribut **ConsoleDebugMode**, finden Sie unter Adapterkonfigurations-Spezifikationen (siehe Seite 321).

Konfigurations- und Wartungstools

Die als Teil des Prozesses der Konfiguration und Wartung der Adapter notwendigen Tools sind hauptsächlich eigenständige Dienstprogramme, wie beispielsweise die CA Business Service Insight-Hilfsprogramme, welches einfache unter Windows ausführbare Dateien sind.

Hinweis: Wenn Adapter in einer UNIX-Umgebung konfiguriert werden, sind diese Dienstprogramme nicht verfügbar und die Konfiguration muss von Hand ausgeführt werden.

Wenn auf einem CA Business Service Insight-Anwendungsserver gearbeitet wird, werden die Dienstprogramme als Teil der Installation der Anwendung mit installiert und befinden sich hier: Program Files\CA\Cloud Insight\Utilities.

Als Teil dieser Installation wird eine Verknüpfung im Start-Menü erstellt. Es wird empfohlen diese Verknüpfung zu verwenden, um die Dienstprogramme zu starten.

In einer Situation, in der nicht auf dem CA Business Service Insight-Anwendungsserver gearbeitet wird, können diese Dienstprogramme auf einen Windows-Computer als standardmäßige Dateien kopiert oder über das mitgelieferte CA Business Service Insight-Paket und durch eine benutzerspezifische Installation installiert werden. Die Installationsroutine ist allerdings nicht zwingend erforderlich und ein Kopieren der .exe-Dateien auf einen geeigneten Speicherort auf der Arbeitsstation ist üblicherweise ausreichend. Mit dieser Option können Sie allerdings feststellen, dass einige .dll-Dateien fehlen und auch vom Server in den lokalen Ordner auf den Computer kopiert werden sollten, sofern sie vorhanden sind.

Konfigurieren von Adaptern und Übersetzungen

Dieser Schritt schließt die Ausführung der folgenden Schritte mit ein:

1. Konfigurieren des Adapters mit dem Adapterassistenten oder manuelles Bearbeiten der XML-Konfigurationsdatei (im folgenden Kapitel beschrieben).
2. Einsetzen des Adapters.
3. Testen des Adapters.
4. Ausführen der Übersetzungen.
5. Schreiben Sie die Übersetzungsskripte, um einen automatischen Übersetzungs-Prozess zu unterstützen (optional).

Hinweis: Die Bereitstellung des Adapters kann automatisch ausgeführt werden, wenn der Adapterassistent verwendet wird, so wie der neue Adapter-Bereitstellungsservice als ein Hintergrundservice auf dem Anwendungsserver ausgeführt wird, um diese Aufgabe zu bewältigen.

Einsatz eines neuen Adapters (Adapterassistent)

Wenn Sie einen neuen Adapter mithilfe des Adapterassistenten erstellen, stellen Sie sicher, dass die Services "Adapter-Listener" und "Adapter-Bereitstellung" ausgeführt werden.

Einsatz eines neuen Adapters (manuell)

Voraussetzungen für das Erstellen eines neuen Adapters

Für den Beginn der Erstellung eines neuen Adapters, müssen folgende Bedingungen erfüllt sein:

- **Stammordner des Adapters**: Wenn CA Business Service Insight auf dem Server installiert wird, existiert dieser Ordner unter dem Stammordner Program Files\CA\Cloud Insight, andernfalls sollte er erstellt werden.
- **Individueller Adapter-Ordner**: Erstellen Sie einen Ordner unter dem Stammordner des Adapters für den bestimmten Adapter.

  Hinweis: Wenn Sie den Adaptermanager verwenden, erstellt das Hilfsprogramm automatisch den Ordner, wenn der neue Adapter hinzugefügt wird.
- **Ausfürbare Dateien des Adapters**: TextFileAdapter.exe, die ausführbare Datei des Textdateiadapters für den Textdateiadapter; SQLAdapter.exe, SQL die ausführbare Datei des Dateiadapters für den SQL-Adapter. Diese finden Sie auf dem Anwendungsserver im Ordner "Programme\CA\Cloud-Insight\Adapters".

  Hinweis: Während der Adapter-Erstellungsvorgänge über den Adaptermanager sollten Sie möglichst häufig die Option zum Erstellen einer Verknüpfung zu den ausführbaren Dateien wählen, anstatt eine Kopie zu erstellen. Dies stellt sicher, dass alle binären Komponenten richtig aktualisiert werden, wenn ein Upgrade oder Service Release (SR) auf CA Business Service Insight angewendet wird.
- **Konfigurationsvorlagen**: Vorlagen der Adapterkonfigurationsdatei. Ordnen Sie die Dateien in den Stammordner des Adapters ein. Diese finden Sie auf dem Anwendungsserver im Ordner "Programme\CA\Cloud-Insight\Adapters". Die Konfigurationsvorlagen werden verwendet, um die Konfigurationsdatei zu erstellen, was verhindern soll sie von vorne erstellen zu müssen. Es ist auch üblich, eine vorhandene Adapterkonfigurationsdatei hierfür zu verwenden.
- **Adaptermanager**: Eine eigenständige, ausführbare Datei. Es ist ausreichend, eine Kopie der AdaptersManager.exe vom Hilfsprogramm-Ordner im Ordner Program Files\CA\Cloud Insight\Utilities auf dem Anwendungsserver zu erstellen. Es ist nicht notwendig, bei der Erstellung des Adapters dieses Hilfsprogramm zu verwenden. Dieses Hilfsprogramm kann nur auf einem Windows-Server angewandt werden.

- **Two open TCP\IP ports**: Ein Port für den AdapterListener auf dem Anwendungsserver und einen anderen für den LogServer. Der LogServer-Port ist üblicherweise 4040.
- Überprüfen Sie, welche CA Business Service Insight-Anwendungsservicekomponenten ausgeführt werden. Für die Zwecke, den Adapter auszuführen, müssen die folgenden Servicekomponenten ausgeführt werden:
  - **AdapterListener**
  - **TaskHost**
  - **LogServer**

Befolgen Sie diese Schritte:

1. Starten Sie den Adaptermanager (siehe Abschnitt "AdapterManager-Hilfsprogramm", oben).
2. Bereiten Sie alle obigen Voraussetzungen vor und erstellen Sie eine Batch-Datei mit der ausführbaren Befehlszeile (siehe Ausführungsmodi des Adapters (siehe Seite 88)).

Ändern der Adapterkonfigurationsdatei

Bei der Erstellung eines Adapters beträgt der Großteil der Arbeit die Bearbeitung der Konfigurationsdatei.

Diese Arbeit bedeutet, die Attribute in der XML-Datei festzulegen, um das Verhalten des Adapters zu steuern, sodass er wie erforderlich funktioniert. Die Konfigurationsdatei ist eine XML-Datei mit je einem Abschnitt, der einem Schritt in seinem internen Workflow entspricht.

Die Abschnitte sind:

- **Abschnitt General**: Verschiedene Adapter-Attribute, Arbeitsdateiverzeichnis, Ausgabedateieigenschaften; Merker-Fehlersuche, Standards, und so weiter.

Oblicore:

- **Abschnitt OblicoreInterface**: Attribute der Verbindung mit dem CA Business Service Insight-Server (TCP/IP-Port, Sicherheitsmodi usw.).
- **Abschnitt DatasourceInterface**: Attribute der Verbindung mit der Datenquelle (Dateipfad und -muster, Verbindungsstrings, SQL-Abfragen, und so weiter).
- **Abschnitt InputFormatCollection**: Parsing-Regeln für das Analysieren und das Bearbeiten des ursprünglichen Datenformats (Trennzeichen, Feldtypen, Reihenfolge der Daten, reguläre Ausdrücke, und so weiter).
- **Abschnitt TranslatorCollection**: Regeln für das einheitliche CA Business Service Insight-Event, zusammengesetzt aus geparsten und bearbeiteten Datenfeldern.
- **Abschnitt TranslationTableCollection**: Regeln der Datenzuordnung zwischen ursprünglicher Datenterminologie und CA Business Service Insight-Datenelementen.

Diese Abschnitte werden ausführlich unter Adapterkonfigurations-Spezifikationen (siehe Seite 321) beschrieben.

Hinweis: Die Reihenfolge der XML-Knoten innerhalb jeden Abschnitts ist nicht wichtig.

Dateiadapter

Dateiadapter verwenden die FileAdapter-Generische Komponente (ausführbare Datei) und die Konfigurationsdatei mit den zu parsenden ASCII-Dateien.

Dateiadapter-Workflow:

- Kopieren/Umbenennen der Quelldatei zu einer Arbeitsdatei.
- Liest den logischen Datensatz.
- Parsen des Datensatzes entsprechend der Trennzeichen oder regulärer Ausdrücke.
- Sucht das richtige Eingabeformat.
- Erstellen des Event-Datensatzes.
- Übersetzen des Event-Datensatzes.
- Aktualisieren der Steuerdatei.

Konfiguration einer Beispieldatei

Das folgende Beispiel verwendet eine einfache ASCII-Dateidatenquelle mit seinen Event-Ausgabeanforderungen und überprüft die erforderlichen Einstellungen für die Adapterkonfigurationsdatei.

Die Konfigurationsdatei kann angezeigt und mit dem XMLPad-Hilfsprogramm bearbeitet werden.

Für eine Übersicht über die Struktur und den Inhalt der Konfigurationsdatei siehe die dazugehörigen Abschnitte.

Hinweis: Die überprüften Einstellungen sind nur die Haupt- und wichtigsten Attributeinstellungen. Vollständige Attributeigenschaften finden Sie unter Eigenschaften der Adapterkonfiguration (siehe Seite 321).

Dateiadapter-Falluntersuchung

Betrachten Sie das folgende Serverüberwachungssystem, welches die Protokolldateien mit Informationen entsprechend den folgenden Strukturen erzeugt:

Server	Zeitstempel	Speicher-nutzung	CPU-Auslastung
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

Die Dateien werden dem CA Business Service Insight-Adapter-Ordner in einen Unterordner namens "Daten" geliefert. Den Dateinamen wird allen "ServerData" vorangestellt und das Datum und die Zeit angehängt. Die Dateien sind auch CSV-Dateien mit der Endung ".csv".

Das folgende Ausgabe-Event wird benötigt:

- **Feld "Ressourcenübersetzung"**: Server
- **Timestamp field**: Zeitstempel
- **Data fields**: Speicher-Ausnutzung, CPU-Auslastung

Darüber hinaus ist davon auszugehen, dass sich die Datenquelle in Belgien befindet (MEZ-Zeitzone, die eine +1-Zeitzonenabweichung aufweist, und während der Sommerzeit eine Ergänzungsstunde einschließt).

Konfigurationsdatei - Allgemeiner Abschnitt

- **MajorVersion And MinorVersion**: Sollte standardmäßig auf Anwendungsversion festgelegt sein.
- **WorkingDirectoryName (optional)**: Gibt den Standardpfad für die Adapter-Ausgabedateien an (Datenquellensteuerung, Übersetzung, Senden). In diesem Fall ist er auf "Ausgabe" eingestellt, und als solche in diesem Ordner dann unter dem Adapter-Hauptordner erstellt.

Die folgenden Indikatoren steuern die Art und Weise, wie der Adapter die Datensätze liest und übersetzt:

- **RunOnce**: Wenn auf "Ja" festgelegt, liest der Adapter einmal alle verfügbaren Daten und stoppt dann.
- **DataSourceDifferenceFromUTC**: Anzeige des Zeitunterschieds zwischen UTC (Standardzeit-Konstante, immer mit Nullpunktkorrektur. im Vergleich zu GMT) und die Zeitzone der Datenquelle. Die Zeitzone der Datenquelle ist die Zeitzone, mit der die Zeitfelder darin angezeigt werden. Der Grund für dieses Attribut ist der, weil der Adapter alle Daten in UTC standardisiert. Wenn alle Daten in UTC vorhanden sind, besitzt die Anwendung die Flexibilität, um dann die Zeiten entsprechend den Anforderungen des Benutzers anzuzeigen. Die folgenden Attribute geben dem Adapter die Einzelheiten, wie die Zeitfelder von der Eingabe nach UTC transformiert werden müssen:
  - **DefaultOffset**: Verschiebung zu UTC wenn nicht in Sommerzeit. In diesem Fall auf "1" gesetzt, für Mitteleuropäische Zeit (MEZ).
  - **TimeFormat**: Das Format, von dem die Sommerzeitdetails (als Nächstes beschrieben) geparst werden sollten. Das europäische Zeitformat ist "dd/mm/yyyy-hh:mi:ss", während CA Business Service Insight-Formatspezifikationen als "%d/%m/%Y %H:%M:%S" festgelegt werden.
  - **DayLightSaving**: Eine Sommerzeitperiode der Datenquellenzeitzone. Dieses Element ist optional (bedeutet, dass es, wenn nicht ausgewählt, keine Sommerzeiten gibt) und es kann mehr als einmal vorhanden sein; einmal für jeden eingegebenen Sommerzeit-Zeitraum. Wenn einige Elemente vorhanden sind, müssen sie nach Zeit geordnet werden und die Zeiträume dürfen sich nicht überschneiden. Üblicherweise wird dieses Element bei der Konfiguration der Adapter angegeben, um einige Jahre voraus zu sein. In diesem Fall wird nur ein Jahr als Beispiel konfiguriert.
- **Von**: Anfangsdatum des Zeitraums. Die vorgesehene Verwendung des oben genannten Zeitformats ist "25/03/2007 02:00:00".
- **Bis**: Enddatum des Zeitraums. Die vorgesehene Verwendung des oben genannten Zeitformats ist "28/10/2007 03:00:00".
- **Unterschied**: Zeitverschiebung zum DefaultOffset innerhalb der Sommerzeit hinzugefügt. Die Eingabe von "1" als Zeitverschiebung verschiebt die Zeit um eine Stunde während der Sommerzeit zwischen den 2 angegebenen Daten.

Konfigurationsdatei - OblicoreInterface-Abschnitt

Dieser Abschnitt definiert die Verbindung zum CA Business Service Insight-Server.

- Modus im Onlinemodus: Der Adapter verbindet sich mit CA Business Service Insight, ruft die Übersetzungstabellen ab und steuert die Befehle von CA Business Service Insight. Dann sendet er Events, Status und Übersetzungsanfragen an CA Business Service Insight. Konfigurieren Sie den Adapter in diesem Modus, und legen Sie den Wert auf "online" fest.
- Port: Die TCP/IP-Portnummer, die der Adapter verwendet, um mit dem CA Business Service Insight-Server zu kommunizieren (auf dem sich der AdapterListener befindet).

  Vorausgesetzt, dass es damit keine Probleme gibt, konfigurieren Sie den Adapter zur Verwendung des Ports "5555" (willkürlich gewählt). Dies muss auch auf dem Server im GUI für den Adapter angegeben werden, um die Kommunikation zu aktivieren.

Konfigurationsdatei - DataSourceInterface-Abschnitt

Der DatasourceInterface-Abschnitt besteht aus Attributen, die die Verbindung und den Verbindungstyp zwischen dem Adapter und der Datenquelle angeben. Es gibt zwei Arten von Schnittstellen, Datei und SQL. Der Hauptunterschied zwischen den beiden ist, dass die Datei-Sammlung für Dateien, und die Abfragensammlung für SQL benötigt wird.

Der DataSourceInterface-Abschnitt gibt auch an, wie der Adapter die Quelldatei verwaltet (ob er die ursprüngliche Datei löscht, wenn sie nur für den Adapter erstellt wurde, oder ob er die Daten beibehält, wenn sie für andere Verwendungen benötigt werden, und so weiter).

Für Dateiadapter: um ASCII-Dateien zu lesen und zu parsen, wird die Dateischnittstelle verwendet, wie in der folgenden Abbildung gezeigt. Wählen Sie die folgenden Werte für die Einstellungen wie folgt beschreiben aus:

Der Dateien-Abschnitt unter dem Datenquellen-Schnittstellenknoten bezieht sich auf die Verbindung zur Datenquelle. Konfigurieren Sie die folgenden Attribute:

Hinweis: Dieser Abschnitt sieht vollkommen anders für einen SQL-Adapter aus.

- **DeleteFileAfterProcessing**: Legt die Art und Weise fest, in die der Adapter die Quelldatei verarbeitet und entscheidet, wie der Adapter das Lesen steuert, um nur neue Datensätzen zu prüfen. In diesem Fall werden die Quelldateien an Ort und Stelle auf dem Server belassen und dieser Wert ist auf "nein" festgelegt.

  In Fällen, bei denen eine Datei nur für den Adapter erstellt wird und sie gelöscht werden kann, nachdem sie bearbeitet wurde, sollte der Wert auf "ja" gesetzt werden. Die Datei wird dann umbenannt, bearbeitet und gelöscht.

  Wenn der Wert auf "nein" gesetzt ist, wird die Datei kopiert und die Verarbeitung findet in der kopierten Datei statt. Wenn neue Datensätze ans Ende dieser Datei angehängt werden, kopiert der Adapter diese neuen Datensätze während des nächsten Zyklus zur Arbeitsdatei. Wenn neue Datensätze nicht an die Datei angehängt werden, sucht der Adapter nach der nächsten Datei mit dem gleichen Muster und Namen (in lexikografischer Reihenfolge) wie die aktuelle Datei. Wenn der Adapter solch eine Datei findet, fährt er mit der Arbeit auf dieser Datei fort. Der Adapter kehrt nicht zur vorherigen Datei zurück, auch wenn neue Datensätze angehängt werden.

  Setzen Sie den Wert auf "nein", wenn Sie die Integrität der Quelldatei behalten müssen und wenn die Datei angehängt werden soll.

- **InputFormat**: Bezieht sich auf den Namen des InputFormat-Elements in d. nächsten InputFormatCollection, das die Daten von dieser Datenquellenverbindung verarbeitet. Das Eingabeformat ist die Felderstruktur von den Eingangsdaten, die von der Datenquelle kommen, nachdem sie vom Adapter geparst worden sind. Die Parsingmethode wird im Trennzeichenattribut, wie unten erklärt, angegeben. Beim Umgang mit mehr als einer Verbindung über unterschiedliche Schnittstellenformate, wird dieses Feld immer wichtiger und entscheidet, welche Eingabeformatstruktur alle Datenquellendaten verarbeiten.

- **Pfad**: Physikalischer Speicherort der Datenquellendateien. Beispiel: C:\Program Files\CA\Cloud Insight\Adapters\ServersAdapter\data\.
- **NamePattern**: Gibt den Datenquellendateinamen an. Platzhalter können verwendet werden, wenn mehr als eine Datei das gleiche Eingabeformat verwendet. Wenn ein Dateiname ohne Platzhalter angegeben wird, sucht der Adapter nur nach einer Datei mit diesem Namen. Bei der Verwendung von Platzhalter sucht der Adapter nach allen Dateien, die dem Muster entsprechen, sortiert sie lexikografisch und liest sie dann einer nach der anderen ein. Während des weiteren Ablaufs sucht er nach neuen Datensätzen in der letzten Datei, bevor die nächste verarbeitet wird.

  In diesem Beispiel, wenn das Platzhalterzeichen "*" verwendet wird, sind die Attributwerte "ServerData*.csv". (Der Adapter liest alle Dateien mit Namen, die mit ServerData anfangen und die Erweiterung ".csv" aufweisen.)

  Wichtig! Es wird empfohlen, dass ein Datum und die Zeit am Ende der Dateinamen hinzugefügt werden, die das folgende Format "YYYYMMDD-HHMISS" verwenden, um sicherzustellen, dass die Dateien richtig sortiert, in der richtigen Reihenfolge eingelesen werden und dass keine Datei verloren geht. Der Uhrzeit-Teil kann auch hinzugefügt werden, wenn jeden Tag mehrere Dateien erstellt werden.
- **Trennzeichen**: Eine Methode, mit der festgelegt wird, wie die Datei geparst wird. Es können eine oder mehrere Zeichen entsprechend den Datenzeilen, die in die Felder geparst werden, festgelegt werden. Wenn keine Angabe erfolgt, ist der Standardwert "/t".

Die Datenquellendatei in diesem Beispiel ist eine CSV-Datei (durch Kommata getrennt). Die einfachste Möglichkeit, solche Dateien zu parsen, ist, das Komma als Trennzeichen festzulegen.

Andere verfügbare Methoden zum Parsen sind:

- **RegExForParser**: Verwendet einen regulären Ausdruck, um die Parsingregel festzulegen.
- **LogicLineDefinition**: Wird verwendet, wenn eine Zeile in der Datei aus mehreren Zeilen zusammengesetzt wird.
- **TitleExists** (optional): Gibt an, ob die erste nicht leere Zeile der Datenquellendatei eine Titelzeile ist. Der Titel kann vom Adapter beim Parsen der Daten verwendet werden. In diesem Beispiel enthält jede Datei eine Titelzeile, d. h. Sie sollten "Ja" für dieses Attribut angeben.

Abschnitt "InputFormatCollection" der Konfigurationsdatei

In diesem Abschnitt werden die Struktur der Daten, die von der Datenquelle abgerufenen wurden, die Art und Weise, wie eine Datenzeile in Felder aufgeteilt wird, und die Feldtypen und Formate festgelegt. In diesem Abschnitt erfolgen in den kombinierten Feldern die Eingangsdatenfilterung und Datenbearbeitung.

In diesem Abschnitt ist es möglich, die Validierungs-Metriken für die Eingabedatensätze, die "InputFormatValidation" und "ValidationCase" verwenden, festzulegen. Sie bestimmen, ob ein Datensatz gültig ist oder nicht.

Der InputFormatCollection-Knoten kann einen oder mehrere InputFormat-Knoten enthalten.

Der allgemeine Arbeitsablauf dieses Abschnitts, dem der Adapter folgt, ist wie folgt:

- Die Datenzeile wird mit InputFormat unter DataSourceInterface abgeglichen.
- Die Daten werden nach der Abstimmungs-InputFormat-Spezifikation in Felder zerlegt. Die Reihenfolge der InputFormatFields sollte der Reihenfolge der Felder, die von der Datenquelle geparst wurden, entsprechen.
- Den Kombifeldern werden Werte zugewiesen, indem die zuvor festgelegten Datenfelder verknüpft und dann aufgebrochen werden. Kombifelddefinitionen sollten nach der normalen Felddefinition kommen.
- Bearbeitete Daten werden mit den TranslatorSwitch-Bedingungen abgeglichen, um zu bestimmen, welcher Übersetzer verwendet wird, um das Ausgabe-Event aus dem Eingabedatensatz zu erstellen.
- Entweder werden bearbeitete Daten an den Abstimmungs-Übersetzer gesandt oder ignoriert.

Arbeiten Sie mit den folgenden Parametern:

- **InputFormatName**: Name für dieses Format, auf den im DatasourceInterface-Abschnitt Bezug genommen wird.
- **InputFormatFields**: InputFormatFields können einen oder mehrere Feldknoten entsprechend der Datenquellenanzahl von Eingabefeldern enthalten.
- **InputFormatField**: Gibt ein Datenfeld der ursprünglichen Datenzeile oder ein Kombifeld an.

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

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

  - **Name**: Name für dieses Feld, auf den andere Elemente Bezug nehmen, wie z. B. das Kombielement oder die TranslatorFields als Quellenfeld.
  - **Type**: Der Datentyp des Feldes - String/Integer/Real/Time.
  - **Source**: Der Quellenwert für dieses Feld, der verwendete Standardwert ist "Event", mögliche Werte:

- **Event**: Der Feldwert wird dem von der Datenquelle eingehenden Event entnommen. Die Werte von Feldern werden in derselben Reihenfolge entnommen, in der sie von der Datenquelle eingehen.
- **compound**: Der Feldwert wird aus anderen Feldern erstellt - basierend auf der Manipulation von anderen Feldwerten oder -konstanten. Die manipulierten Felder sollten bereits vorher definiert worden sein.
- **title**: Der Feldwert wird den Titelfeldnamen entnommen. Das angegebene Feld sollte bereits vorher definiert worden sein.
- **filename**: Der Feldwert wird dem Datenquellen-Dateinamen entnommen; nur für Textdateiadapter.
- **constant**: Der Feldwert ist konstant und wird dem ConstantValue entnommen, dessen Eigenschaft als Nächstes angezeigt werden sollte.

- **TranslatorSwitch**: Bestimmt, welcher Übersetzer verwendet wird, um die Datenzeile in ein einheitliches Event zu übersetzen.
  - **DefaultTranslator**: Übersetzer, der in Fällen verwendet wird, in denen keine Kriterien abgeglichen werden können. Wenn der Wert auf "Ignore" eingestellt ist, wird kein Übersetzer verwendet, und die Zeile wird ignoriert und nicht an CA Business Service Insight gesendet.

Abschnitt "TranslatorCollection" der Konfigurationsdatei

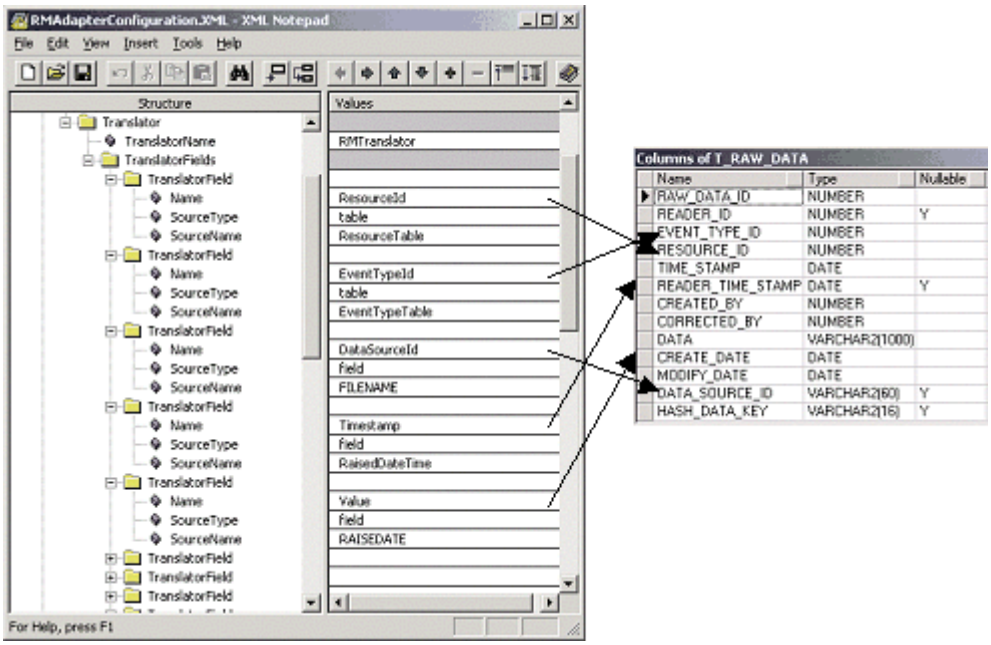
Der Abschnitt "Translator Collection" definiert, wie der geparste und verarbeitete Datenquellen-Datensatz, der in vorherigen Abschnitten extrahiert wurde, in ein CA Business Service Insight-Event übersetzt wird.

Dieser Abschnitt gibt auch an, wie doppelte Events gehandhabt werden und der Mechanismus der Event-Besonderheit verwendet wird (weitere Informationen finden Sie unter Event-Besonderheit (siehe Seite 131)).

Wenn der Schnittstellenmodus auf "Online" gesetzt wird, hat das CA Business Service Insight-Event eine einheitliche Struktur, die folgende Felder enthält:

- **Timestamp**: Die Zeit des Eintretens des Events.
- **ResourceId**: Die Ressourcen-ID, die mit dem Event verknüpft ist (die Ressource, die innerhalb dieses Events gemessen wurde).
- **EventTypeId**: Der Event-Typ, der mit dem Event verknüpft ist, beschreibt den Event-Typ (Typ der Ressourcenmessung, Ticketaktion usw.).
- **DataSourceId** (optional): Jeder Wert. Liefert zusätzliche Filterkriterien für Rohdaten-Events.
- **Value** (mehrere): Wert(e) des Events (Messevent, Ticketnummer usw.). Dieses Feld ist häufig mehr als einmal vorhanden.

Die Struktur des Übersetzers entspricht der Struktur des Event-Typs in CA Business Service Insight und auch der Datenbanktabelle T_RAW_DATA, die das Event, wie in der folgenden Abbildung angezeigt, speichert:



- **Translator**: Beschreibt, wie der Satz von Feldern, die er erhält, in das Ausgabe-Event übersetzt wird.
- **TranslatorName**: Der Name, der von InputFormat verwendet wird, um Feldsätze an diesen Übersetzer zu senden. Dieses Beispiel verwendet den Namen "Translator1". Beziehen Sie sich auf die vorherige Abbildung für Werte, die für dieses Szenario verwendet werden können.
- **TranslatorFields**: Enthält eine Liste von TranslatorField-Elementen, die jeweils die folgenden Attribute enthalten:
  - **Name**: Feldname. Bei der Onlineschnittstelle muss er Timestamp, ResourceId, EventTypeId, ResourceId oder Value lauten.
  - **SourceType**: Gibt die Quelle des Feldwerts an. Hierbei kann es sich um einen der folgenden Optionen handeln:
    - **Field**: Der Wert dieses Feldes wird vom Feld im Eingabeformat übernommen. Das SourceName-Attribut enthält den InputFormat-Feldnamen.
    - **Table**: Der Wert des Feldes wird von der Übersetzungstabelle übernommen. Das SourceName-Attribut enthält den Namen der Übersetzungstabelle, der sich auf einen Namen bezieht, der im nächsten Abschnitt von "TranslationTableCollection" angegeben wird. Dieser Typ wird für Werte verwendet, die ausgewählt werden, um in Ressourcen und Event-Typen des Events übersetzt zu werden.
    - **Lookup**: Der Wert dieses Feldes wird von der Übersetzungstabelle übernommen. Das SourceName-Attribut enthält den Tabellennamen. Der Wert soll vom Attribut "LookupValue" und nicht vom Eingabeformat übersetzt werden. Er wird üblicherweise verwendet, wenn ein konstanter Wert zum Übersetzen benötigt wird.
    - **Constant**: Der Wert des Feldes ist konstant, und sein Wert befindet sich im Attribut "ConstantValue". Wird ein konstanter Wert verwendet, ist es notwendig, den Feldtyp durch die Verwendung des Typattributs anzugeben.
  - **SourceName**: Enthält den Feldnamen oder den Namen der Übersetzungstabelle.
  - **LookupValue**: Enthält den Suchwert, wenn SourceType="lookup".
  - **ConstantValue**: Enthält den konstanten Wert, wenn SourceType=constant. Wenn der Typ des Feldes "Time" ist, wird der konstante Wert entsprechend dem "TimeFormat" (festgelegt im Abschnitt "Allgemein" des Adapters), "Now" oder "NowUtc" formatiert, wobei "Now" die aktuelle Zeit im Datenquellen-Gebietsschema und "NowUtc" die aktuelle Zeit in UTC-Format ist.

Dieser Abschnitt enthält auch die Zuordnungstabellen, die die Zuordnung von Datenquellenwerten zu CA Business Service Insight-Event-Feldern festlegen, und die Tabellendefinition mit dem zugehörigen Datenquellenwert, der übersetzt werden soll.

- **LoadingMode**: Der Standardwert für die Onlineschnittstelle ist "Remote" und für die Offlineschnittstelle "Eigenständig".

  Festgelegte Lademethode der Übersetzungstabellen ist wie folgt:

  - **Standalone**: Der Adapter lädt die Übersetzungstabellen lokal. Es gibt in Bezug auf die Übersetzung keine Verbindung zum CA Business Service Insight-Server. Änderungen in den Übersetzungstabellen werden nur in der lokalen Datei gespeichert.
  - **Remote**: Der Adapter sendet eine Anfrage, um alle Tabellen des CA Business Service Insight-Servers zu laden. In den Übersetzungstabellen vorgenommene Änderungen werden auch lokal gespeichert.

- **TranslationTable**: Verlinkt den Ereigniswert zur Zuordnungstabelle.Name: Field name.

  - **Name**: Name der Übersetzungstabelle, den der Übersetzer verwendet und auf den er sich bezieht.
  - **DestinationType**: [resource, event_type, contract_party, service, time_zone, value]. Enthält den Typ der Übersetzungstabelle. In diesem Beispiel wird die Spalte **Server** in der Datenquellendatei in eine CA Business Service Insight-Ressource übersetzt. Daher ist der Typ der Übersetzungstabelle "Ressource", und die Tabelle enthält die Übersetzungen der Werte in Ressourcen-IDs in CA Business Service Insight.
  - **TranslationField**: Der Feldname, von dem übersetzt werden soll und der von den Eingabeformatfeldern übernommen wird. Kann maximal fünf Felder enthalten.

Jede in der Konfigurationsdatei festgelegte Übersetzungstabelle muss eine entsprechende Definition in der CA Business Service Insight-Benutzeroberfläche aufweisen.

Die XML-Darstellung einer Beispielkonfigurationsdatei sieht wie folgt aus:

```
<?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"/>
                         <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>
```

SQL-Adapter

SQL-Adapter werden im Wesentlichen mit den entsprechenden Einstellungen in der Konfigurationsdatei von der generischen SQL-Adapterkomponente verwendet. Der SQL-Adapter kann sich mit allen Datenquellen verbinden, die ODBC und OLEDB unterstützen. Die Verbindung wird über einen Verbindungsstring hergestellt. Hierzu muss der entsprechende Treiber auf dem Server, auf dem der Adapter installiert ist, installiert sein.

Datenquellen-Beispiele:

- Oracle
- SQL Server
- Zugriff
- Excel
- Textdateien, CSV-Dateien (diese könnten auch über den TEXT-Adapter angeschlossen werden, aber eine ODBC-Verbindung bietet häufig zusätzliche Filter-/Abfrage-Optionen).

SQL-Adapter-Workflow:

- Verbindung herstellen.
- Lokale Variablen durch die Werte des letzten Schlüsselfelds ersetzen.
- Eine automatische Vervollständigung durchführen, Wo-Klauseln für die Abfragen erstellen und die Enden der Abfragen verknüpfen.

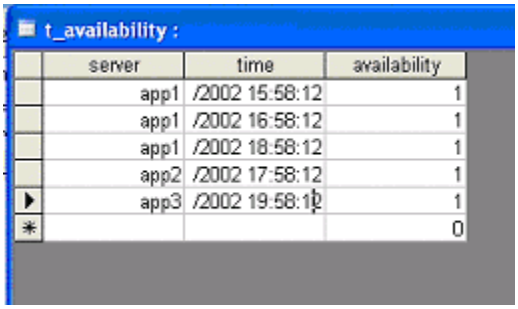
  Abfrage ausführen, um eine Datensatzgruppe zu erhalten.

  Für jeden Datensatz der Datensatzgruppe, die bei der Abfrage ausgegeben wird, gilt Folgendes:

  - Das richtige "InputFormat" finden.
  - Erstellen des Event-Datensatzes.
  - Den Datensatz übersetzen.
  - Den Wert des Schlüsselfeldes in der Kontrolldatei aktualisieren.
- Die Verbindung schließen.
- In den Standby-Modus wechseln.

Beispiel einer SQL-Adapter-Konfigurationsdatei

Gezeigt wird eine MS Access-Datenbank (.mdb) mit folgender Tabelle:



Die SQL-Adapter-Konfigurationsdatei und die Dateiadapter-Konfigurationsdatei unterscheiden sich nur im Abschnitt DatasourceInterface.

Im Abschnitt DatasourceInterface des Dateiadapters wird die Dateien-Sammlung gespeichert, wohingegen bei einer SQL-Adapter-Datei die Optionen "ConnectionString" und "QueryCollection" verwendet werden.

Der Hauptunterschied zwischen den beiden Konfigurationsdateien liegt in der Abfrage- und Parsing-Methode. Der Rest der Datei ist identisch.

Die SQL-Schnittstelle legt die Verbindung zur Datenbank und die Abfragen, die zum Abrufen der Daten verwendet werden, fest.

Details:

Hinweis: Der Abschnitt wird basierend auf der obigen Datenquellen-Datenbank festgelegt.

Verbindungsstring-Element

ConnectionString: Verbindungsstring, mit der auf die Quellendatenbank zugegriffen werden kann.

Der ConnectionString kann im DataSourceInterface-Element und/oder in den Abfrage-Elementen festgelegt werden. Die ConnectionString-Definition wird standardmäßig im DataSourceInterface-Element festgelegt und wird nur bei einer Abfrage ohne eine ConnectionString-Definition verwendet. Dies ermöglicht dem Adapter, eine Verbindung zu mehreren Datenbanken herzustellen, wobei jede Abfrage ihren eigenen Verbindungsstring haben kann. Weitere Details zum Mechanismus der Verbindungsstrings finden Sie im folgenden Abschnitt.

Abschnitt "Abfragesammlung" der Konfigurationsdatei

Query: Gibt die Abfrageattribute an.

- **QueryName**: Name für die Abfrage.
- **InputFormat**: InputFormat, das mit dieser Abfrage verknüpft ist. Der Adapter verwendet das InputFormat, um die Daten von der Quelle zu extrahieren.

  Hinweis: Die Reihenfolge der InputFormat-Felder muss der Reihenfolge der für die Abfrage ausgewählten Felder entsprechen.
- **SleepTime**: Zeit in Sekunden, die sich der Adapter im Standby-Modus befindet, um auf den Empfang neuer Daten zu warten.
- **SelectStatement**: Enthält eine ausgewählte Anweisung, die auf der Quelldatenbank ausgeführt werden muss.

  Hinweis: Sie müssen die Abfrageschlüsselfelder als erste ausgewählte Werte in der Anweisung eingeben.

  - **AutoCompleteQuery**: Wenn diese Option auf "Ja" gesetzt wurde, verknüpft der Adapter automatisch wie folgt eine Wo-Anweisung mit der gewählten Abfrage:
    - Erstellt eine Wo-Anweisung, die nur neue Werte basierend auf den Schlüsselfeldern findet.
    - Fordert die auf den Schlüsselfeldern basierte Ergebnisanweisung an.
- **QueryKeyFields**: Definiert die Felder, um den nächsten Datenabruf zu starten.
  - **KeyField**:
    - **Name**: Name des Feldes, der von den Feldern der Abfrage übernommen wird.
    - **Sort**: Daten-Sortierreihenfolge (aufsteigend/absteigend).

- **ValueLeftWrapper**: Verknüpft die Zeichen vor dem Wert des Feldes. Der Standard ist ' (Apostroph).
- **ValueRightWrapper**: Verknüpft die Zeichen nach dem Wert des Feldes. Der Standard ist ' (Apostroph).
- **Critical**: Stoppt die Ausführung anderer Abfragen, wenn diese bestimmte Abfrage fehlschlägt.
- **SleepTime**: Die Standby-Zeit zwischen erforderlichen Operationen. Der Standard ist ' (Apostroph).

Hinweis: Für die Felder "Datum" sind häufig Sonderzeichen notwendig, um das Datum selbst einzuschließen. Die Zeichen, die zur Identifizierung des Felds als ein Datumsfeld erforderlich sind, hängen von der Datenquelle ab. Das Zeichen # (siehe Abbildung am Anfang des Abschnitts) kann verwendet werden, um das Wertfeld in Excel zu umschließen. Andere Datenquellen können jedoch andere Methoden erfordern. Weitere Informationen finden Sie unter Adapterkonfigurations-Spezifikationen (siehe Seite 321).

- **SelectInitialValues**: Eine ausgewählte Anweisung, die die Anfangswerte für die Schlüsselfelder für die erste Wo-Anweisung bereitstellt (wenn die Kontrolldatei leer ist).

  Beispiel einer Abfrage (Excel-basierter ODBC), bei der AutoCompleteQuery="yes" ist:

  SELECT INC_CLOSE_DATE, INCIDENT_REF, Severity, `Resolve Time`, `Date Logged`, `Date Resolved` FROM `AllCalls$` WHERE INC_CLOSE_DATE>CDate('03.07.05 13:06:21') order by INC_CLOSE_DATE

  Diese ausgewählte Anweisung muss in der Zieldatenbank ausgeführt werden können, in der die Abfrage durchgeführt wird. Dies kann zwischen den Quellen und den ODBC-Treibern, die zur Herstellung einer Verbindung zu den Quellen verwendet werden, unterschiedlich sein. In Oracle können Sie beispielsweise die Werte aus einer speziellen "Dual"-Tabelle auswählen (select 'aaa', '1-jan-1970' from dual), in Excel hingegen können Sie die Werte nur direkt - ohne Tabelle - auswählen. (select 'aaa')

Nachfolgend ist eine vollständige Konfigurationsdatei im XML-Format dargestellt:

```
<?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>
      <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','01.01.70'

               </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>
```

SQL-Adapter-Verbindungsstring

Der Mechanismus des MSQL-Adapter-Verbindungsstrings ist so aufgebaut, dass er die folgenden Funktionalitäten bereitstellt:

- Arbeiten mit mehreren Verbindungsstrings im gleichen Adapter
- Arbeiten mit einer Verbindungsstringvorlage, bei der der Dateiname geändert werden kann, ohne die Konfigurationsdatei zu verändern.
- Löschen der Dateidatenquelle, sobald der Adapter mit dem Lesen fertig ist.
- Verschieben der Dateidatenquelle an einen anderen Speicherort, sobald der Adapter mit dem Lesen fertig ist.

Zusätzlich zur einfachen Definition des Verbindungsstrings als eine Zeichenfolge im ConnectionString-Element kann der Verbindungsstring von Segmenten definiert werden, wobei jedes Segment die bestimmten Werte enthält, die mit dem Verbindungsstring verknüpft sind. Dies ermöglicht dem Adapter, den Verbindungsstring dynamisch zu erstellen.

Es gibt zwei Segmenttypen. Der eine Typ ist Text und enthält Text, der als solches mit dem Verbindungsstring verknüpft ist. Der zweite Typ ist eine Datei und enthält einen Dateinamen mit oder ohne Platzhalter. Das Dateisegment kann nur einmal angezeigt werden. Es enthält andere Attribute, die festlegen, wie mit der Lesedatei weiter verfahren wird.

Der ConnectionString kann im QueryCollection-Element und/oder in den Abfrage-Elementen festgelegt werden. Die ConnectionString-Definition wird standardmäßig im DataCollection-Element festgelegt und wird nur bei einer Abfrage ohne eine ConnectionString-Definition verwendet.

Elemente und Attribute

ConnectionString-Element

Dieses Element kann untergeordnete Segmentelemente enthalten. Wenn es mindestens ein Segmentelement enthält, wird der Verbindungsstring damit verknüpft. Andernfalls wird der Verbindungsstring aus dem ConnectionString-Elementtext übernommen (wie bei der aktuellen Version).

Segment-Element

Dieses Element enthält ein obligatorisches Attribut namens *Type*, das über zwei gültige Werte, Text und Datei, verfügt. Wenn Type="file" ist, muss das Segment-Element mindestens ein Datei-Element enthalten. Jedes Datei-Element wird als eine andere Abfrage angesehen.

Datei-Element

Dieses Element enthält Attribute, die festlegen, welche Dateien im Verbindungsstring verwendet werden sollten und wie mit der Datei zu verfahren ist, wenn der Adapter mit dem Lesen der Datei fertig ist.

- **Path**: Definiert den Dateipfad (Verzeichnis).
- **NamePattern**: Definiert den Dateinamen anhand des angegebenen Pfads. Er kann Platzhalter enthalten.
- **UsePath**: Gültige Werte: ja/nein. Der Standardwert ist "ja". Wird der Wert auf "Ja" gesetzt, wird der Dateipfad mit dem Verbindungsstring verknüpft.
- **UseFileName**: Gültige Werte: ja/nein. Der Standardwert ist "ja". Wird der Wert auf "Ja" gesetzt, wird der Dateiname mit dem Verbindungsstring verknüpft (z. B. erforderlich für Excel-Dateien).
- **UpdateSchemaFile**: Gültige Werte: ja/nein. Der Standardwert ist "nein". Wird der Wert auf "Ja" gesetzt, aktualisiert der Adapter die Datei "schema.ini" mit dem aktuellen Dateinamen.

  Hinweis: Verwenden Sie dieses Attribut nur, wenn Sie wollen, dass der Adapter die Datei "schema.ini" ändert.

Interne Variablen

Zwei zusätzliche interne Variablen sind hinzugefügt worden und können in den SelectStatement- und SelectInitialValues-Elementen verwendet werden. Diese sind:

- :file_name: wird durch den aktuellen Dateinamen und die Erweiterung ersetzt.
- *:file_name_no_ext*: wird durch den aktuellen Dateinamen ohne die Erweiterung ersetzt.

Beispiele

Beispiel 1: Einfacher Verbindungsstring für Oracle:

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

oder

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

Beispiel 2: Einfacher Verbindungsstring für Excel mit einer einzigen Datei:

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

oder

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

Beispiel 3: Einfacher Verbindungsstring für die Verwendung von mehreren Excel-Dateien:

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

Beispiel 4: Einen standardmäßigen ODBC-DSN-Eintrag verwenden:

Bei der Verwendung eines standardmäßigen ODBC-DSN-Eintrags können Sie mit einer Quelle Verbindung aufnehmen, die eine DSN-Eingabe im ODBC-Manager auf dem Anwendungsserver erstellt hat. Der standardmäßige ODBC-DSN-Eintrag befindet sich im Abschnitt Administrative Tools" auf dem Server.

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

Lesen von Datensätzen von einer Datei-Datenquelle

Solange es eine ODBC-Schnittstelle zur Datenquelle gibt, ist es möglich, einen SQL-Adapter zu konfigurieren, um Dateien abzufragen. Um den Adapter so zu konfigurieren, dass er Daten von mehreren Dateien einliest, ist es notwendig, die Segment-Elemente im ConnectionString-Element zu verwenden. Ein Beispiel finden Sie im vorherigen Abschnitt, der sich mit den Verbindungsstring befasst.

Der SQL-Adapter arbeitet folgendermaßen mit den Dateien:

1. Bei jeder Abfrage versucht der Adapter die Daten von der Datei einzulesen, bis er keine weiteren Datensätze mehr erhält.
2. Der Adapter geht dann zur nächsten Datei und versucht, sie zu lesen.
3. Wenn es keine weiteren Dateien gibt, wird der Adapter für diese Abfrage entsprechend dem SleepTime-Attribut in den Standby-Modus versetzt.

Die Datei Schema.ini

Wenn ein Text-ODBC-Treiber installiert ist, wird das Format der Textdatei von der Schemainformationsdatei bestimmt (schema.ini). Diese Schemainformationsdatei sollte in dasselbe Verzeichnis wie die Textdatenquelle gespeichert werden.

Schema.ini-Dateien werden anhand von Eingaben erstellt und jede Eingabe beschreibt die Struktur einer einzelnen Textdatei. Die Anfangszeile bei jeder Eingabe ist der Name der Textquelldatei (eingeschlossen in eckigen Klammern).

Wenn der Adapter mit Dateien arbeitet, die mithilfe von Platzhaltern definiert wurden, ändert sich der Dateiname immer wieder. Da der Name in der schema.ini-Datei keine Platzhalter enthalten kann, muss der Adapter die schema.ini-Datei ändern, bevor er eine Verbindung zur Datenbank herstellt.

Sie müssen daher eine Anzeigezeile vor der Dateieingabezeile hinzufügen. Diese Anzeigezeile enthält das Namensmuster vom Verbindungsstring-Element in der Adapter-Konfigurationsdatei (eingeschlossen in eckigen Klammern). Der Adapter ersetzt die nächste Zeile durch den neuen Dateinamen, der in eckige Klammern eingeschlossen ist.

Hinweis: Die schema.ini-Datei kann mehrere Eingaben enthalten. Es ist Ihre Verantwortung, die Zeilen an der richtigen Stelle hinzuzufügen.

Beispiel einer schema.ini-Datei

```
[sqltes*.txt]
[sqltest2.txt]
ColNameHeader = False
CharacterSet = ANSI
Format = TabDelimited
Col1=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
----------------------------------------
```

Datei "DataSourceControl"

Bei jeder Abfrage enthält die Datenquellen-Kontrolldatei das Dateinamensmuster und den Namen der aktuellen Eingabedatei, um mit derselben Datei fortzufahren, wenn der Adapter neu startet.

Nachfolgend ist ein Beispiel einer Kontrolldatei im XML-Format dargestellt:

```
<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>03.07.05 13:06:21</LastValue>
            </KeyField>
            <LastFileName>IncidentsMarch.xls</LastFileName>
         </Query>
      </QueryCollection>
   </Data>
```

Beibehalten von Eingabe-Dateien

Wenn der Adapter die aktuelle Datei zu Ende gelesen hat, sucht er nach der nächsten Datei. Die nächste zu lesende Datei ist die erste Datei, deren Namensmuster passt und deren Name größer (in lexikografischer Reihenfolge) ist als der vorherige Dateiname. Der Adapter kehrt nicht zu vorherigen Dateien zurück, auch dann nicht, wenn ihnen neue Datensätze hinzugefügt werden.

Der Adapter verwendet das InitialFileName-Attribut nur, wenn diese beiden Attribute auf "nein" gesetzt sind und die Kontrolldatei den letzten Dateinamen nicht enthält.

Abfragen-Überprüfung

Der Adapter überprüft den Verbindungsstring und die Abfrage nur, wenn die angegebene Datei existiert. Wenn sie mithilfe von Platzhaltern definiert wurde, prüft der Adapter nur die erste Datei.

Fehler können auftreten, wenn der Adapter versucht, von einer neuen Datei zu lesen. In diesem Fall bleibt der Adapter sofort stehen, wenn das Attribut "Critical" auf "ja" gesetzt wurde. Wurde es auf "nein" gesetzt, fährt der Adapter nicht mit dieser Abfrage, aber mit den anderen Abfragen fort. Alternativ verlässt der Adapter die aktuelle Datei und geht weiter zur nächsten Datei.

Interne Variablen des Adapters

Es gibt zwei interne Variable, die in der Konfigurationsdatei verwendet werden können, um sich auf den aktuellen Zusammenhang des Dateinamens zu beziehen. Diese Variablen sind :file_name und :file_name_no_ext. Sie beziehen sich entsprechend auf den aktuellen Dateinamen bzw. den aktuellen Dateinamen abzüglich der Dateierweiterung.

Diese Variablen können im SelectStatement-Element, SelectInitialValues-Element und im QueryKeyField\Function-Attribut verwendet werden.

Der Adapter ersetzt die Variable durch den Dateinamen in den Abfragen.

Zum Beispiel:

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

Erstellen eines Adapters mit der CA Business Service Insight-Benutzeroberfläche

Jeder Adapter, der für den Betrieb in einer CA Business Service Insight-Umgebung konfiguriert ist, muss in der Benutzeroberfläche registriert und in der Registrierung festgelegt sein. Der Hauptgrund hinter diesem Schritt ist, die Einstellungen der Adapter-Listener-Seite einzurichten, sodass der Adapter-Listener bereit ist, Events vom Adapter aufzunehmen. Während dieses Schrittes werden alle Adaptereinstellungen, wie Übersetzungstabellen und Event-Typen, festgelegt.

Befolgen Sie diese Schritte:

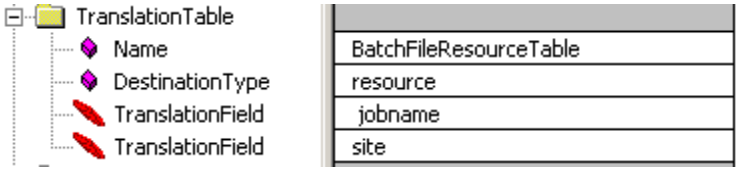
1. Erstellen Sie den Adapter.
2. Wählen Sie Ressourcen/Adapter.
3. Überprüfen Sie die vorhandenen Adapter in der Liste, um sicherzustellen, dass keiner auf denselben Port eingestellt ist wie Ihr Adapter. Dies ist derselbe Port der in der Konfigurationsdatei des Adapters im Verzeichnis OblicoreInterface\OnlineInterface\Port angegeben ist.
4. Klicken Sie auf Neu hinzufügen, und wählen Sie dann die Methode aus, die Sie verwenden wollen, um den Adapter zu erstellen. Es gibt eine Reihe von Optionen:
    a. **Create manually**: Hiermit können Sie den Adapter-Listener so einrichten, dass die Verbindung zum Adapter, den Sie festgelegt haben (oder werden), manuell hergestellt wird.
    b. **Using the Wizard**: Hiermit können Sie den Adapter mithilfe der Screen-By-Screen-Assistentenschnittstelle erstellen. Weitere Details hierzu finden Sie im nächsten Abschnitt.
    c. **From a template**
    d. **From an existing configuration file**: Hiermit können Sie eine vorkonfigurierte Adaptervorlage hochladen, die automatisch die Felder des Adapterassistenten mit den Optionen füllt, die in der ausgewählten Konfigurationsdatei festgelegt sind.
    e. Wie der angezeigte Bildschirm aussieht, hängt von der gewählten Option ab.
5. Füllen Sie die Felder aus:

    Netzwerkadresse: Geben Sie die IP-Adresse des Adapters ein. Handelt es sich um einen lokalen Host, geben Sie Localhost ein, andernfalls geben Sie den festgelegten Port ein.
6. Klicken Sie auf Speichern.

So erstellen Sie die notwendigen Übersetzungstabellen:

Hinweis: Diese Schritte sollten für jede in der Konfigurationsdatei festgelegte Übersetzungstabelle ausgeführt werden:

1. Klicken Sie im Menü "Design" auf "Übersetzung", "Übersetzungstabellen", und klicken Sie auf die Schaltfläche "Neu hinzufügen".
2. Alle Einstellungen der Übersetzungstabelle sollten der äquivalenten Tabellendefinition in der Adapter-Konfigurationsdatei entsprechen:
   - *Name*: Sollte mit dem Attribut Name unter Übersetzungstabelle, Name in der Konfigurationsdatei übereinstimmen.
   - Quellfelder: Sollte alle Felder vom TranslationField der Übersetzungstabelle enthalten. Fügen Sie alle Felder hinzu. Es kann eine Kombination zweier oder mehrerer Felder geben, die den Übersetzungswert bilden. Nachfolgend sehen Sie ein Beispiel dazu, wie dies aussehen könnte:



3. Zieltyp: Sollte auch mit dem DestinationType-Attribut der Definition der Übersetzungstabelle in der Konfigurationsdatei übereinstimmen. (resource, event_type usw.).
4. Registrierte Adapter: Fügen Sie die Adapter hinzu, die diese Übersetzungstabelle verwenden sollen. Eine einzelne Übersetzungstabelle kann von mehr als einem Adapter verwenden werden.
5. Klicken Sie auf Speichern.
6. Importieren Sie die Definition der Event-Typ-Felder für jeden Event-Typ.

   Um die Felddefinition aus einer bestimmten Adapter-Konfigurationsdatei importieren zu können, muss dieser Adapter mindestens einmal ausgeführt worden sein und mit CA Business Service Insight verbunden gewesen sein. Wenn der Adapter mit CA Business Service Insight verbunden ist, sendet er die Konfigurationsdatei an CA Business Service Insight. Dies ermöglicht dem System, die Felddefinition aus dieser Datei zu verwenden.

   Hinweis: Alternativ können Sie die Felddefinitionen manuell für den Event-Typ festlegen.
7. Aktivieren Sie den Adapter:
   a. Klicken Sie im Menü "Design" auf "Datenerfassung", "Adapter".
   b. Klicken Sie die Schaltfläche Start für den Adapter.

Die folgende Tabelle enthält die unterschiedlichen Status der Adapter:

Status	Beschreibung
Inaktiv	Anfangsstatus. Adapter ist inaktiv und ist noch nicht gestartet worden.
Listener inaktiv	Adapter-Listener-Dienst (Zuteiler) wird nicht gestartet.
Starten	Adapter startet.

Gestartet	Adapter wurde gestartet.
Wird angehalten	Wird gestoppt.
Pausing	Wird angehalten.
Paused	Angehalten.
Wird nicht ausgeführt	Der Adapter wird nicht auf dem Computer des Adapters ausgeführt.
Fehler	Es ist ein Fehler in der Konfigurationsdatei des Adapters aufgetreten; der Adapter kann nicht starten.
Verbindungsfehler	Es ist ein Fehler bei der Herstellung einer Verbindung zum Adapter (falscher Host/Port) oder vor dem ersten Start des Adapters aufgetreten und es wurde kein Signal empfangen. Dies ist der Status, wenn der Adapter erstmals gestartet wird.
Gesperrt	Die maximale Anzahl zurückgewiesener Events wurde erreicht.

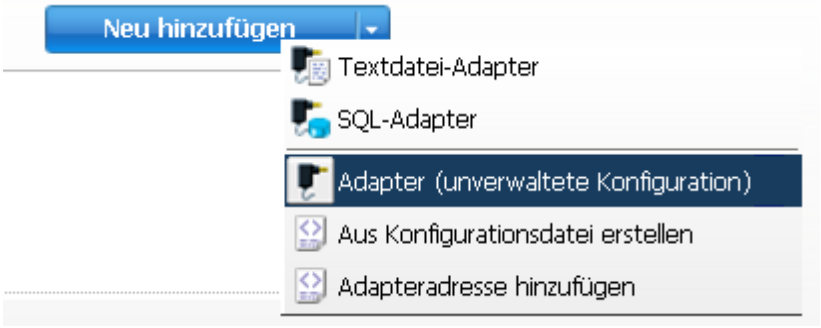
Erstellen eines Adapters mit dem Adapterassistenten

Der Adapterassistent ist eine neuere Funktion der CA Business Service Insight-Benutzeroberfläche, hat eine intuitivere Oberfläche als der XML-Editor und bietet eine Anleitung zum Erstellen eines neuen Adapters. Der Assistent führt Sie durch eine Reihe von Registerkarten, die alle erforderlichen Informationen enthalten, um einen Adapter zu erstellen. Zum Schluss können Sie eine Kopie der kompilierten XML-Konfigurationsdatei herunterladen. Es gibt jedoch einige Beschränkungen bezüglich dessen, was der Assistent ausführen kann.

Der Assistent ermöglicht Ihnen derzeit nicht:

- von unterschiedlichen Datenquellen-Schnittstellen aus auf dieselbe Eingangsstruktur (Eingangsformat) zu verweisen
- von unterschiedlichen Eingängen aus auf dieselbe Ausgangsstruktur (Übersetzer) zu verweisen
- einen Eingabeformat-Schalter zu konfigurieren und diesen zu verwenden, um zu entscheiden, welches Eingabeformat von der Schnittstelle der Datenquelle verwenden werden soll
- eine Definition eines Kennungsfeldes für die Datenquelle in der Ausgangsstruktur anzugeben
- eine Definition von mehr als einer Datei im Verbindungsstring für die Abfrage von Text oder Excel-Dateien mit derselben Abfrage anzugeben
- UTC- oder UtcNow-Zeitbeschränkungen zu verwenden
- Werte festzulegen, die mit einem "Leerzeichen" beginnen oder enden

Wenn man einen neuen Adapter mithilfe des Assistenten erstellt, gibt es vier Auswahloptionen, wie in der folgenden Abbildung dargestellt:



Mit den ersten beiden Optionen können Sie die Assistenten-Schnittstelle verwenden, um einen Textdatei-Adapter oder einen SQL-Adapter zu erstellen. Die nächste Option Adapter (unverwaltete Konfiguration) ist zu wählen, wenn Sie einen vorkonfigurierten Adapter hinzufügen wollen, den Sie mit dem XML-Editor erstellt haben. Diese Option erlaubt Ihnen jedoch nicht, die Konfiguration dieses Adapters zu einem späteren Zeitpunkt mit dem Assistenten weiter zu bearbeiten. Die letzte Option "Aus Konfigurationsdatei erstellen" ermöglicht es Ihnen, eine vorkonfigurierte Adapterkonfiguration hochzuladen und das System so einzustellen, dass es die Konfiguration zur weiteren Bearbeitung in die Assistentenschnittstelle importiert. Hierfür dürfen jedoch keine der oben erwähnten Assistentenbeschränkungen in dieser bestimmten Konfiguration implementiert sein.

Darüber hinaus bieten die Konfigurationsoptionen des Adapterassistenten dieselbe Funktionalität wie die manuelle Adapterkonfiguration. Der Assistent soll eine einfachere und freundlichere Schnittstelle für das Bearbeiten der Konfigurationseinstellungen bereitstellen. Da dieselben Funktionen und Prinzipien wie bei der manuellen Alternative gelten, ziehen Sie die relevanten Abschnitte für weitere Informationen zu Rate.

Ausführen und Testen des Adapters

Die Einstellung der Adapter-Konfigurationsdatei kann üblicherweise nicht in einem einzelnen Zyklus abgeschlossen werden. Es kann eine Reihe von Iterationen notwendig sein, während denen der Adapter ausgeführt wird. Die Ergebnisse werden überprüft, um sicherzustellen, dass die Adapterkonfiguration korrekt ist.

Nachfolgend sind einige der gängigen Probleme aufgeführt, die behoben werden müssen:

- Verbindungsprobleme (entweder zwischen der Datenquelle und dem Adapter oder dem Adapter und dem Listener)
- Fehler in der Konfigurationsdatei, wie z. B.:
  - Falsche Struktur
  - Falsche Verwendung von Attributen
  - Falscher Schreibweise verwendet (Adapter unterscheiden zwischen Groß- und Kleinschreibung - "Ja" und "ja" werden unterschiedlich behandelt usw.)
  - Falsche Wertzuweisung
- Daten-Manipulationsfehler, wie z. B. Ausgabe-Event-Struktur, falsche Event-Werte, Fehler in Abfragen.

Befolgen Sie diese Schritte:

1. Setzen Sie den Adapter auf folgende Werte: RunOnce = "ja" und LogDebugMode = "ja". Stellen Sie den Wert für RejectedEventsUpperLimit auf eine sinnvolle Zahl ein (siehe Ausführungsmodi des Adapters (siehe Seite 88)).

   Die folgende Abbildung zeigt die Konfigurationseinstellungen, die für den Test erforderlich sind.

2. Es ist auch möglich, den Offlinemodus zu verwenden, um die Einstellungen der Konfigurationsdatei vorzunehmen.
3. Sobald die Konfigurationsdatei erfolgreich geladen wurde, setzen Sie die Einstellung zurück auf Online (siehe Ausführungsmodi des Adapters).
4. Jede Iteration umfasst die folgenden Schritte:
   a. Aktualisieren Sie die Adapter-Konfigurationsdatei und beheben Sie etwaige Probleme.
   b. Löschen Sie alle Ausgabe-Dateien des Adapters.
   c. Doppelklicken Sie auf die Verknüpfung zur *Adapter*-Ausführungsdatei oder auf die *.bat* Datei, die erstellt wurde, um den Adapter zu starten.
   d. Öffnen Sie die Protokolldatei des Adapters mit dem Protokoll-Browser (das Hilfsprogramm Log Browser.exe befindet sich unter CA Business Service Insight Utilities), und vergewissern Sie sich, dass es keine Fehlermeldungen gibt.

5. Der erste Schritt dient dazu, die richtige Konfigurationsdatei zu erstellen und ein Status zu erreichen, wo der Adapter die Konfigurationsdatei erfolgreich lädt. Wiederholen Sie diesen Schritt, bis Sie erfolgreich mit CA Business Service Insight und der Datenquelle verbunden sind und abgelehnte Events und Anfragen für eine Übersetzung haben.
6. Prüfen Sie Folgendes, um diese Phase abzuschließen:
   - Es gibt keine Fehlermeldungen in der Protokolldatei des Adapters.
   - Der Adapter ist erfolgreich mit CA Business Service Insight und der Datenquelle verbunden.
   - Der Adapter sendet Anfragen für Übersetzungen und abgelehnte Events.

   Sie sollten erwartungsgemäß den Buchstaben "R" für jeden Datensatz auf der Konsole sehen, den der Adapter ablehnt. Denken Sie daran, dass abgelehnte Events erwartet werden, bis alle notwendigen Übersetzungen fertig gestellt worden sind.
7. Vergewissern Sie sich, dass die Datei rejectedEvents Datensätze enthält und nicht leer ist.

YGWorkflowAdapter
Adapter YGWorkflowAdapter
Running
<Connecting Oblicore>
LTT<Connecting DataSource (Commit) >
(Commit) RRRR

Ressourcen- und Event-Übersetzungen

Im vorherigen Schritt gab es eine Reihe von abgelehnten Events, die während der Ausführung des Adapters erstellt wurden. Diese abgelehnten Events wurden in der Datei RejectedEvents.txt, aber auch als ausstehende Übersetzungseinträge in der CA Business Service Insight-Datenbank gespeichert. Der nächste Schritt beim Laden von Rohdaten in das System ist, eine Übersetzung von dem, was erfasst wurde, bereitzustellen, sodass CA Business Service Insight diese Daten nach Bedarf verwenden kann.

Rohdaten-Events könnten abgelehnt werden, wenn entweder der Event-Typ ODER die Ressource nicht im System definiert wurde. Die ausstehenden Events werden darüber definiert, von welchem Übersetzungstabellentyp sie stammen. Bei den üblichen Beispielen stammen die Event-Typen aus einer Übersetzungstabelle und die Ressourcen aus einer anderen.

Wenn eine neue Ressource vom Adapter gefunden wird (z. B. wenn ein neuer Server zum Netzwerk-Überwachungstool hinzugefügt wird und als neuer Eintrag in der Datenquelle dieses Überwachungstools erscheint), kann sie zum Ressourcenmodell des Systems hinzugefügt werden. Es gibt zwei Schritte, die Sie für diese neue Ressource ausführen müssen, damit sie in die Berichte von CA Business Service Insight aufgenommen wird.

Sie müssen zuerst die Ressource als eine CA Business Service Insight-Entität erstellen (eine Ressource). Dann muss eine Übersetzung eingegeben werden. Dadurch werden die in der Datenquelle gefundene Stringangabe und die als Ressource in CA Business Service Insight definierte Entität verknüpft. Dieses zweistufige Verfahren kann mit einer Aktion und zwar über die Benutzeroberfläche mit der Funktion "Hinzufügen und Übersetzen" durchgeführt werden, d. h. die neue Ressource und der notwendige Übersetzungseintrag werden in einem Schritt erstellt. Mit der Option zum Hinzufügen und Übersetzen können Sie mehrere Einträge auswählen, vorausgesetzt die Zuordnungseinstellungen sind dieselben. Die Durchführung einer Übersetzung bezeichnet einen Prozess, bei dem ein Abgleich zwischen dem Datenquellenwert und der CA Business Service Insight-Entität erfolgt. Bei der Erstellung von Übersetzungen wird ein Eintrag mit den abgeglichenen Werten in die Übersetzungstabelle eingefügt. Dadurch wird der Adapter bei nachfolgenden Abfragen automatisch wissen, wie er diesen neuen Wert verarbeiten soll.

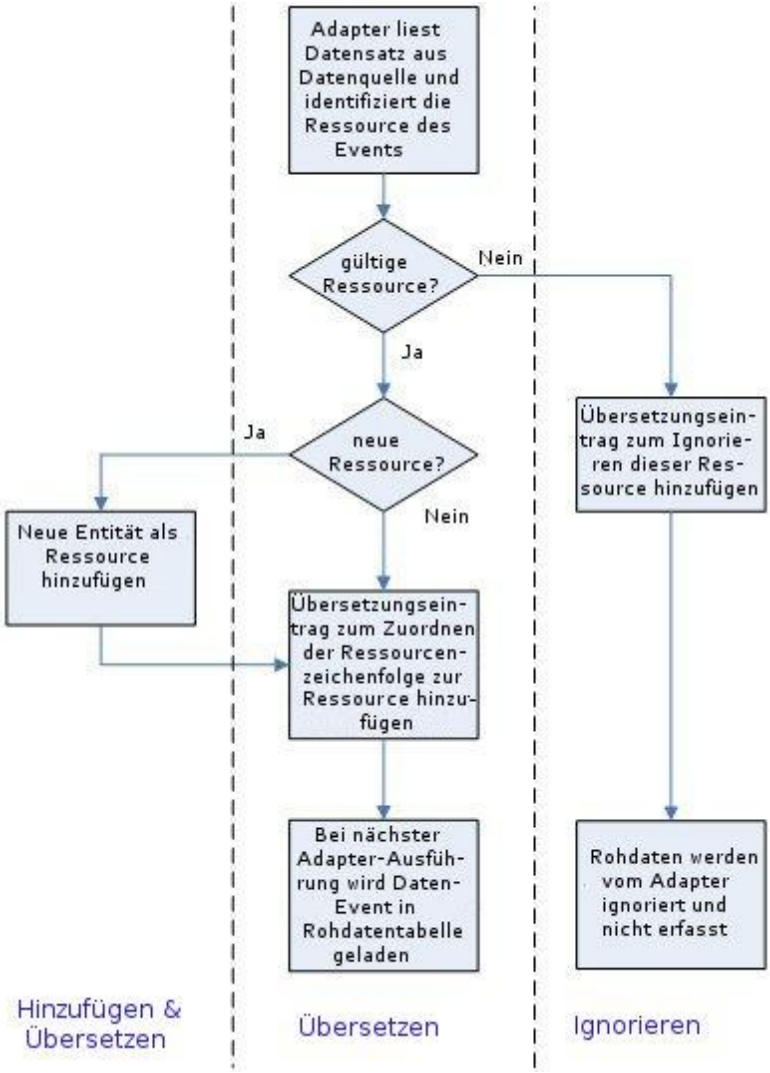
In dieser Phase ist der Adapter bereits ausgeführt worden und er hat Übersetzungsanfragen für jeden Wert in der Übersetzungstabelle gesendet. Die mit diesen Werten verknüpften Events wurden abgelehnt und warten darauf, an CA Business Service Insight gesendet zu werden, sobald die Übersetzung fertig gestellt worden ist. Übersetzungen können manuell oder mit einem Übersetzungsskript automatisch ausgeführt werden.

Die folgenden Übersetzungsaktionen können für ausstehende Übersetzungseinträge ausgeführt werden:

Übersetzen: Hiermit können Sie einen Eintrag in der Übersetzungstabelle erstellen und einen Datenquellenwert und die dazugehörige CA Business Service Insight-Entität abgleichen. Die CA Business Service Insight-Entität, für die Übersetzungen erstellt werden, muss schon existieren. (Ein gutes Beispiel hierfür ist eine falsch geschriebene Ressource von einer Datenquelle. Es können verschiedene Namen in der Datenquelle vorhanden sein, die sich tatsächlich auf eine einzige logische Entität beziehen. Beispiel: Server001 und SERVER001. Bei CA Business Service Insight-Ressourcen ist die Groß- und Kleinschreibung zu beachten.)

- **Hinzufügen und Übersetzen**: Hiermit können Sie gleichzeitig eine neue Entität in CA Business Service Insight erstellen *und* einen Übersetzungseintrag für diese Entität in der Übersetzungstabelle hinzufügen. Dies ist die gängigste Aktion für ausstehende Übersetzungseinträge, da der Übersetzungs-Mechanismus verwendet wird, um die Infrastruktur in CA Business Service Insight zu erstellen.
- **Ignorieren**: Wird ein Eintrag ignoriert, werden alle zugehörigen Events ebenfalls ignoriert und nicht an die CA Business Service Insight-Rohdatentabelle gesendet. Die ignorierten Events werden verloren gehen. Wenn die Datenquelle beispielsweise Informationen von allen Servern in einer Datenzentrale enthält, aber nur die Daten der Anwendungsserver für die Berechnung des Service Levels benötigt werden, dann werden zwar alle Server für eine Übersetzung auf den Adapter zugreifen, aber nur die Anwendungsserver werden übersetzt. Alle anderen werden ignoriert, da CA Business Service Insight sicherstellt, dass das unnötige Event ebenso ignoriert wird. Ein ignorierter Eintrag kann bei Bedarf zu einem späteren Zeitpunkt übersetzt werden, es werden aber nur die Daten ab diesem Punkt erfasst.
- **Löschen**: Der Übersetzungseintrag wird entfernt und das zugehörige abgelehnte Event wird auch gelöscht. Wenn derselbe Wert später noch einmal von der Datenquelle gesendet wird, wird ein neuer ausstehender Eintrag erstellt.

Der folgende Ablaufplan fasst die Fälle für die Verwendung dieser Optionen zusammen:


Adapter liest Datensatz aus Datenquelle und identifiziert die Ressource des Events
gültige Ressource?
Nein
Ja
neue Ressource?
Ja
Nein
Neue Entität als Ressource hinzufügen
Übersetzungseintrag zum Zuordnen der Ressourcenzeichenfolge zur Ressource hinzufügen
Bei nächster Adapter-Ausführung wird Daten-Event in Rohdatentabelle geladen
Übersetzungseintrag zum Ignorieren dieser Ressource hinzufügen
Rohdaten werden vom Adapter ignoriert und nicht erfasst
Hinzufügen & Übersetzen
Übersetzen
Ignorieren

Manuelle Übersetzungen

Manuelle Übersetzungen werden benötigt, wenn die Entität schon in CA Business Service Insight vorhanden ist. Dies kann in einer Reihe von Situationen auftreten. Zum Beispiel, wenn ein Übersetzungsskript, mit dem die Ressourcen einer externen Quelle automatisch erstellt werden sollen, ausgeführt wurde (aber die Übersetzungen nicht automatisiert werden konnten). Oder wenn eine Datenquelle zweifelhafte Einträge hat und eine Ressource zweimal in unterschiedlicher Weise definiert wurde (z. B. Server001p und server001P). Es könnte sogar aufgrund von Ressourcen sein, die manuell erstellt wurden.

So führen Sie eine manuelle Übersetzung durch, wenn die Ressource bereits existiert:

1. Klicken Sie im Menü "Design" auf "Übersetzungen", "Übersetzungseinträge".
2. Standardmäßig werden alle ausstehenden Einträge angezeigt.
3. Aktivieren Sie das Kontrollkästchen neben dem ausstehenden Übersetzungseintrag, den Sie übersetzen wollen, um ihn auszuwählen.
4. Klicken Sie auf "Übersetzen".
5. Wählen Sie die dazugehörige Entität (Ressource, Event-Typ usw.) aus der Liste aus. (Wenn keine Elemente in der Liste angezeigt werden, müssen Sie möglicherweise die Standardsuchkriterien in dem Feld ändern.)
6. Klicken Sie auf die Zeile, die das Element enthält, um den Ressourcen-/Event-Typ aus der angezeigten Liste der Entitäten auszuwählen. Es bleibt hervorgehoben, wenn es ausgewählt ist.
7. Klicken Sie auf "Übersetzen". Der Übersetzungseintrag ist nun im System gespeichert.

So führen Sie eine manuelle Übersetzung durch, wenn die Ressource noch NICHT existiert:

1. Klicken Sie im Menü "Design" auf "Übersetzungen", "Übersetzungseinträge".
2. Standardmäßig werden alle ausstehenden Einträge angezeigt.
3. Aktivieren Sie das Kontrollkästchen neben dem ausstehenden Übersetzungseintrag, den Sie in eine CA Business Service Insight-Ressource verwandeln und übersetzen möchten, um ihn auszuwählen.
4. Klicken Sie auf Hinzufügen & Übersetzen.
5. Stellen Sie sicher, dass der Ressourcennamen so wie von Ihnen gewünscht festgelegt wurde. Sie können das Feld *Anzeigename* auch anpassen, um die Art und Weise zu verändern, wie die Ressource in einem Bericht angezeigt wird. Wenn diese Ressource als Teil eines "Änderungssatzes" verwaltet werden soll, sollte hier auch der bestimmte Änderungssatz festgelegt werden.
6. Das Wirksamkeitsdatum der Ressource sollte als das Datum konfiguriert werden, ab dem diese Ressource in einen Bericht innerhalb des Systems aufgenommen wird. Beachten Sie, dass die Ressource ERST in Berichten, die nach dem hier festgelegten Datum erstellt werden, angezeigt wird.
7. Klicken Sie auf die Registerkarte Details und wählen Sie die folgenden Optionen für die Ressourcenzuordnung aus: Effective (true/false), Ressourcentyp, Resource Group membership, Service and Contract association.
8. Klicken Sie auf Speichern & Übersetzen. Die Ressource wird dem Ressourcen-Servicekatalog hinzugefügt, und der Übersetzungseintrag ist nun auch im System gespeichert. Sobald alle ausstehenden Einträge verarbeitet wurden, können Sie überprüfen, ob die Daten in das System geladen werden:
9. Navigieren Sie zu den Ressourcen in CA Business Service Insight, und vergewissern Sie sich, dass die neue Ressource erstellt worden ist.
10. Führen Sie den Adapter erneut aus.
11. Überprüfen Sie, ob die Datei für die abgelehnten Events leer ist.
12. Überprüfen Sie, ob die Datei für die gesendeten Events leer ist.
13. Überprüfen Sie mit dem Rohdatentool, ob die Events zu den Rohdaten hinzugefügt worden sind.

Einrichtung der Infrastruktur

Die Einrichtung der Infrastruktur umfasst die Definition der Daten-Modellierungsentitäten, so wie sie während der Design-Phase festgelegt wurde.

Diese Phase schließt die Definition aller Infrastruktur-Entitäten nicht ein und ist abgeschlossen, wenn die Adapterkonfiguration abgeschlossen worden ist.

Während dieser Phase werden die folgenden Entitäten ins CA Business Service Insight-System geladen:

- Event-Typen
- Ressourcentypen
- Ressourcen & Ressourcen-Zuordnungen
- Ressourcengruppen

Hinweis: Sobald der Adapter erfolgreich ausgeführt worden ist, können Sie die automatische Importfunktion verwenden, um Event-Typ-Felder zu definieren.

Automatische Übersetzungen mit Übersetzungsskripten

Eine automatische Übersetzung bezeichnet die Automatisierung der Erstellungs- und Übersetzungsprozesse/-infrastruktur. Diese Automatisierung basiert auf einer externen Datenquelle, die Skripte verwendet, um die Übersetzungsaktionen auszuführen.

Eine automatische Übersetzung erfolgt über Übersetzungsskripte. Übersetzungsskripte beschleunigen den Prozess der Zuweisung neuer IT- und Unternehmensressourcen in CA Business Service Insight. Ein Übersetzungsskript ermittelt, wenn ein neuer Übersetzungseintrag empfangen wird, und übersetzt die Ressourcen automatisch, sodass die Ressourcen rasch und effizient zugeordnet werden. Die Automatisierung unterstützt die Schnittstelle zu CMDBs, Dies ermöglicht dem System, die Ressourcen basierend auf ihrer konfigurierten Definition zu identifizieren. Die automatische Übersetzung hat mehrere Vorteile; unter anderem erleichtert sie die Pflege der Übersetzungen und verhindert das Auftreten von Fehlern. Übersetzungsskripte können verwendet werden, um neue Ressourcen zu erstellen und Änderungen zuzuordnen.

Zusätzlich können Übersetzungsskripte verwendet werden, um:

- Einträge in vorhandene CA Business Service Insight-Objekte zu übersetzen.
- neue CA Business Service Insight-Objekte hinzuzufügen und sie entsprechend den vorhandenen Übersetzungseinträgen zu übersetzen.
- Objekte anzulegen und Einträge basierend auf Tabellen außerhalb von CA Business Service Insight, wie z. B. Ressourcentabellen von einem anderen externen CMS, zu übersetzen.
- zu prüfen, ob ein Objekt existiert.
- Ressourcen zu erstellen und Ressourcenzuordnungen durchzuführen, wie z. B. Ressourcentypen, Ressourcengruppen, Vertragsparteien und Servicekomponenten.
- Ressourcen zuzuordnen bzw. die Zuordnung aufzuheben, um Sätze zu ändern.

Da der Übersetzungsprozess auch manuell in der Benutzeroberfläche ausgeführt werden kann, muss eine Entscheidung getroffen werden, welcher Übersetzungsprozess zu wählen ist. Wenn Sie dies tun, berücksichtigen Sie die folgenden Vor- und Nachteile einer automatischen Übersetzung:

- Erhöhte Projektkomplexität - Übersetzungsskripte erfordern entsprechende Fähigkeiten und einen Mehraufwand für die Skriptentwicklung.
- Mehr Entwicklungs- und Qualitätssicherungszeit für das Testen.
- Muss nicht in Fällen implementiert werden, wenn es sich bei dem Übersetzungsprozess nur um einen einmaligen ersten Aufwand handelt.
- Kann für eine Implementierung als Teil einer zweiten Phase geplant werden.
- Weniger laufende Wartungsarbeiten.
- Weniger durch Menschen verursachte Übersetzungsfehler.
- Weitere Informationen zum Erstellen und Ausführen von Übersetzungsskripten finden Sie im *SDK*-Handbuch im Kapitel zur CMDB-Integration.

Erweiterte Themen der Adapter-Funktion

Die folgende Themen behandeln erweiterte Themen der Adapter-Funktion.

Event-Besonderheit

Event-Besonderheit ist ein Adapter-Mechanismus, der einen Prozess in Gang setzt, mit dem verhindert werden soll, dass Rohdaten doppelt in die T_Raw_Data-Tabelle eingefügt werden.

Ohne die Funktion der Event-Besonderheit wird keine Validierung durchgeführt, ob ein identisches Event vorliegt, wenn der Adapter gegen eine Datenquelle läuft und Events in die Datenbank lädt. Die Funktion der Event-Besonderheit löst dieses Problem, indem sie die Möglichkeit bietet, festzulegen, ob Events vor dem Einfügen auf ihre Besonderheit geprüft werden sollen und was zu tun ist, wenn diese Situation eintreten sollte. Dieser Prüfungsprozess kann sich allerdings ungünstig auf die Leistung des Adapters auswirken.

Die Lösung ermöglicht dem Benutzer, einen Schlüssel festzulegen, der auf den Feldern des Events basieren kann. Solch ein Schlüssel repräsentiert die Einmaligkeit des Events, d. h., wenn zwei Events denselben Schlüssel haben, handelt es sich um dieselben Events.

Es ist auch möglich, zu bestimmen, welche Aktion in dem Fall auszuführen ist, wenn ein Duplikat eines Schlüssels bereits in der Datenbank vorhanden ist. Diese Aktionen werden nachfolgend näher betrachtet.

Hinweis: Der Schlüssel kann als eine Kombination aus mehreren Feldern vom Übersetzer definiert werden.

Adapter-Konfigurationsdatei mit Event-Besonderheit

TranslatorCollection/Translator/OnDuplication

Dieses Feld bestimmt, was getan wird, wenn das Event schon in der Datenbank existiert.

Mögliche Werte sind:

- **Add** - fügt (zusätzliche) Events in die Datenbank ein.
- **Update** - löscht (in einigen Fällen) ein vorhandenes Event und fügt ein neues ein, wenn validiert wurde, dass sich das Event geändert hat.
- **updateAlways** - löscht (in einigen Fällen) ein vorhandenes Event und fügt ein neues ein, ohne dies auf Änderungen zu prüfen.
- **Ignore** - fügt kein neues Event in die Datenbank ein.

Der Standardwert ist Add.

Funktion "Ignore" vs. "Add"

Es kann zu einer geringen Leistungsminderung kommen, während der Adapter im Modus Ignore ausgeführt wird. Im Modus Add führt das System jedoch immer eine Neuberechnung durch, wenn ein doppeltes Event vorliegt.

Aktualisieren vs. Immer aktualisieren

Die Option Update vermindert die Leistung des Adapters, während die Option updateAlways die Berechnungsleistung vermindert, während in jedem Fall eine Neuberechnung ausgelöst wird.

TranslatorCollection > Translator > TranslatorFields > TranslatorField > IsKey

Dieses Attribut bestimmt, ob das Feld, zu dem es gehört, seinen Wert zum einmaligen Schlüssel für die Rohdaten beitragen sollte. Er wird eingeschlossen, wenn der Wert = "ja" lautet, und nicht eingeschlossen, wenn der Wert = "nein" lautet. Der Standardwert (wenn kein Wert angegeben ist) ist für Standardfelder (ResourceId, EventTypeId und Timestamp) "ja" und für alle anderen "nein". Der Schlüssel sollte sorgfältig ausgewählt werden, um zu garantieren, dass die Rohdaten die erforderliche Integrität bewahren und sicherstellen, dass die Berechnungen absolut präzise sind.

Adapter-Listener-Verhalten mit Event-Besonderheit

Wenn ein neues Event vom Adapter erhalten wird, überprüft der Listener den Wert des Felds OnDuplication. Wenn der Wert "add" ist, wird der normale Einfügungsprozess ausgeführt. Wenn der Wert nicht "add" lautet, prüft der Listener, ob ein Event mit demselben UniqueKey und derselben Leser-ID in der Datenbank vorhanden ist. Wenn die Datenbank bereits wie beschrieben ein Event enthält, wird das neue Event nicht in die Datenbank eingefügt, wenn der Wert für OnDuplication "Ignorieren" ist.

Wenn der OnDuplication-Wert auf "update" gesetzt wurde, wird das Event auf Änderungen geprüft. Wenn alle Felder identisch sind, wird das neue Event nicht in die Datenbank eingefügt.

Wenn der Bei Duplizierung-Wert auf "updateAlways" gesetzt wurde, wird die vorherige Prüfung verworfen und dennoch eine Aktualisierung durchgeführt.

In den Modi "update" und "updateAlways" werden die folgenden Schritte unternommen:

- Wenn ResourceId, EventTypeId und Timestamp verändert wurden, erfolgt eine Neuberechnung aller Metriken, die die alte Eventformel verwenden.
- Das Event wird mit den entsprechenden Werten aktualisiert.

Aggregieren von Transaktionsdaten

Transaktionsdaten werden häufig erfasst, um sie mit Schwellenwerten abzugleichen oder um die etwaigen periodischen Prozentsätze des Erfolgs berechnen zu können. Beispiel: Alle fünf Minuten wird eine virtuelle Transaktion gegen ein System ausgeführt und das Ergebnis (Antwortzeit in Millisekunden) wird wie unten dargestellt gespeichert:

```
[…]
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 anderen Situationen kann auf tatsächliche Transaktionen, die im System stattfinden, zugegriffen werden, anstatt virtuelle Transaktion zu verwenden. In diesen Fällen können Hunderte oder sogar Tausende von Transaktionen pro Stunde ausgeführt werden.

In den beiden oben aufgeführten Fällen sollte es möglichst vermieden werden, solch eine große Datenmenge in CA Business Service Insight zu laden.

Die Aggregation von Daten nach Zeiträumen ist die beste Möglichkeit, die Datenmengen zu reduzieren. Wenn der Schwellenwert, gegen den der Erfolg gemessen wird, festgelegt ist, kann die Aggregation ausgeführt werden. Dabei wird dem Adapter ermöglicht, die Anzahl der Transaktionen, die erfolgreich waren, innerhalb des Sammelzeitraums zu zählen. Beispiel: Wenn der Erfolgsschwellenwert im vorherigen Beispiel auf 500 Millisekunden festgelegt wurde, waren nur fünf von sieben Transaktionen innerhalb des angezeigten Zeitraums erfolgreich. Das Problematische an dieser Vorgehensweise ist der festgelegte Schwellenwert: Was ist, wenn man zu einem späteren Zeitpunkt den Schwellenwert ändern möchte? In solch einer Situation müssen Rohdaten wieder gelesen und vom Adapter gegen den neuen Schwellenwert getestet werden.

Deswegen sollte der Adapter Transaktionsdaten optimalerweise auf flexible Art und Weise ansammeln, ohne bedeutende Daten zu verlieren.

Eine beschränkte Lösung ist, dem Adapter zu ermöglichen, die Transaktionen gegen einige Schwellenwerte zu testen. Es gibt zwei Arten, dies zu tun:

- Ein Event mit mehreren Tests - Event Type = {TotalTransactions, TransBelow250, TransBelow500, TransBelow750, […]}
- Mehrere Events mit einem Test - Event Type = {TotalTransactions, Threshold, TransBelowThreshold}.

Beide Optionen weisen dasselbe Problem auf - die Schwellenwerte können später nur innerhalb eines kleinen Satzes von vordefinierten Werten geändert werden.

Empfohlene Lösung

Annahme: Alle möglichen Schwellenwerte sind das Vielfache einer bestimmten Zahl. Alle Schwellenwerte (in Millisekunden) sind beispielsweise ein Vielfaches von 250, d. h. 0, 250, 500, 1750 und 3000 Millisekunden sind mögliche Schwellenwerte.

Basierend auf dieser Annahme lautet die vorgeschlagene Lösung, Transaktionen anzusammeln, indem alle Werte auf das gemeinsame Vielfache gerundet werden, und zu zählen, wie viele Transaktionen unter jeden gerundeten Wert fallen. Event Type = {RangeFrom, RangeTo, TransactionCount}

Die folgenden Events werden zum Beispiel erstellt, um die zuvor angezeigten Daten zu aggregieren, wobei das gemeinsame Vielfache 250 Millisekunden ist:

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

Kommentare:

Der Zeitstempel dieser Events wäre der gleiche. (Alle aggregierten Events können z. B. den Zeitstempel 1/1/2007 00:00 aufweisen, und es kann einen weiteren Satz von Events für das nächste Zeitbeispiel um 1/1/2007 01:00 geben, vorausgesetzt es findet stündlich eine Aggregation statt)

RangeTo wird berechnet, indem man eine Transaktion auf das gemeinsame Vielfache rundet (siehe nachfolgend).

RangeFrom ist RangeTo abzüglich des Vielfachen plus 1. Es wird nur aus Gründen der Klarheit angegeben, und Sie können es überspringen.

Beispiel: Eine stündlich ausgeführte Aggregation würde ungefähr wie folgt aussehen (ersetzen Sie MULTIPLE durch den Wert des Vielfachen):

```
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
```

In der Business-Logik kann folgende Bedingung auf die Events angewandt werden:

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

Einige schlüssige Einblicke:

- Da die Verteilung der Transaktionen üblicherweise normal verläuft, sollte die Anzahl der aggregativen Events relativ niedrig sein.
- Durch die Auswahl von höheren gemeinsamen Vielfachen werden weniger aggregative Events abgeleitet.
- Die Menge an aggregativen Events sollte immer geringer sein als die Rohdatenmenge oder ihr entsprechen.

Ausführen eines Adapters hinter einer Firewall

Um einen Adapter hinter einer Firewall ausführen zu können, wird die folgende Lösung vorgeschlagen:

Öffnen Sie zwei Ports. Der erste Port ist der dem Adapter in CA Business Service Insight zugewiesene Port. Der zweite Port, der optional ist, aber empfohlen wird, ist der Protokollserver-Port. Der Protokollserver-Port wird standardmäßig als Port 4040 festgelegt. Durch das Öffnen des Protokollserver-Ports kann der Adapter in das CA Business Service Insight-Protokoll schreiben und auch Warnungen erzeugen. Dies ist optional, weil der Adapter immer einer lokalen Protokolldatei berichtet.

Für beide Ports wird das TCP/IP-Protokoll verwendet.

Laden von Daten aus mehreren Verzeichnissen

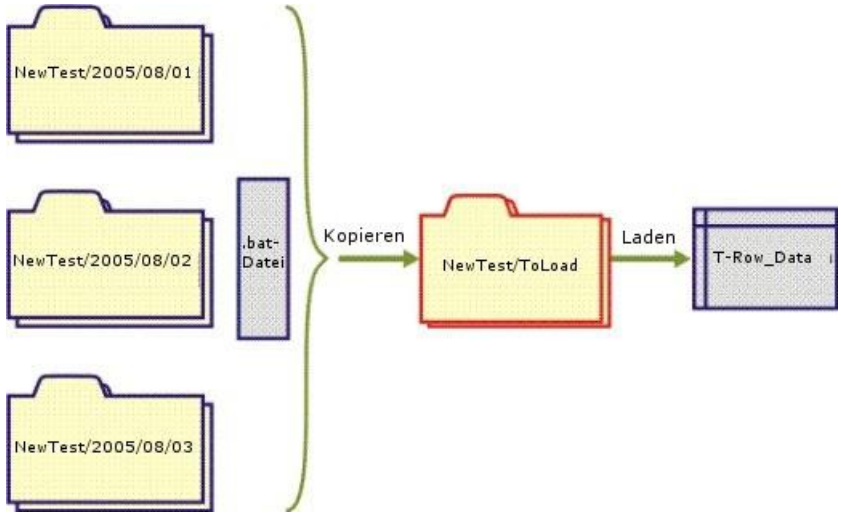
Dieser Abschnitt beschreibt eine Lösung, die implementiert werden kann, wenn sich die Datenquellendateien (Eingabe für einen CA Business Service Insight-Adapter) jeden Tag in unterschiedlichen Verzeichnissen befinden, oder für alle festgelegten Zeiträume (entsprechend einer bestimmten Namenskonvention).

So kann die vorgegebene Verzeichnisstruktur z. B. c:\NewData\YYYY\MM\DD sein. In diesem Fall wird jeden Tag ein neues Verzeichnis angelegt, in dem die relevanten Dateien für diesen Tag gespeichert werden.

Der CA Business Service Insight-Adapter muss das neueste Verzeichnis durchsuchen und die neuen Dateien laden.

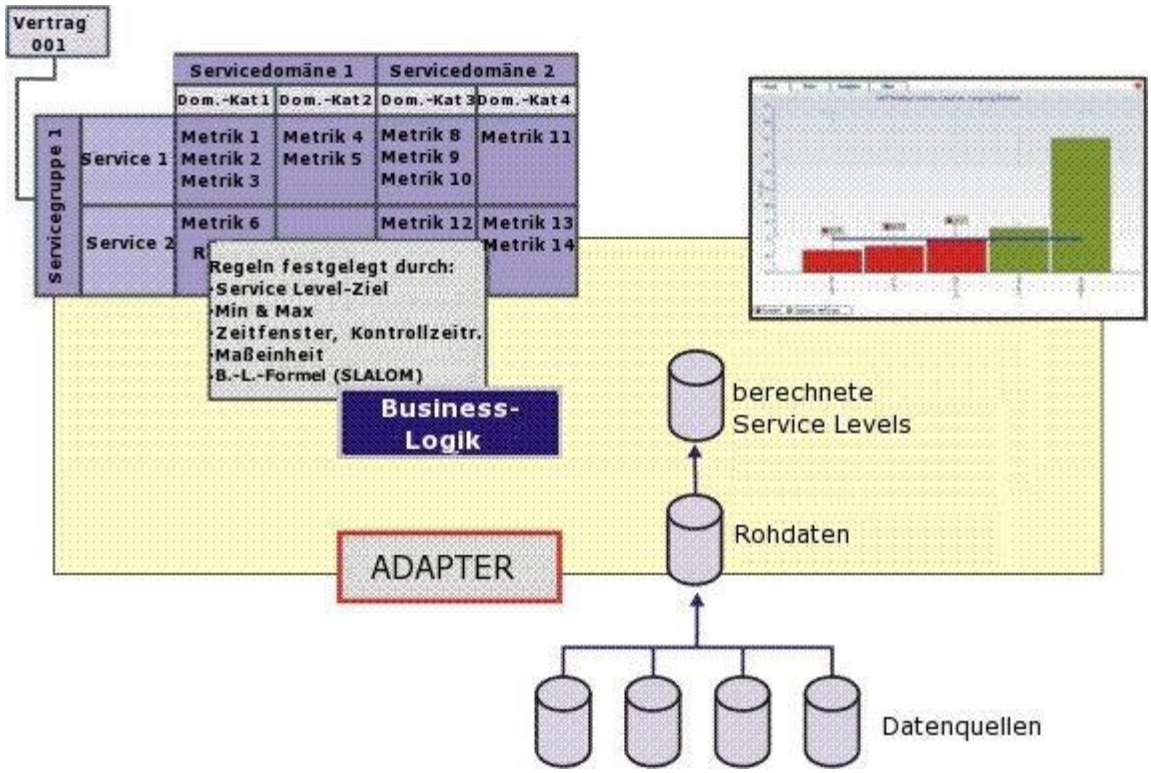
Eine verfügbare Umgehungslösung ist, einige Befehle am Anfang der bat-Datei hinzuzufügen, über die der Adapter ausgeführt wird. Diese Befehle kopieren die Datenquellen, die geladen werden müssen, aus dem bestimmten Ordner (entsprechend ihrer Konventionen) und fügen sie in einen einzelnen, dafür vorgesehenen Ordner ein, der speziell für diesen Zweck erstellt wurde. Der Adapter lädt die Informationen immer aus diesem Ordner.

Das folgende Bild zeigt diese Lösung grafisch:



Business-Logik-Skripting (Business-Logik-Experte)

Das folgende Bild stellt die Position der Business-Logik innerhalb von CA Business Service Insight dar. Sie fällt unter jede der Metriken innerhalb der Verträge.



In dieser Phase der Implementierung werden die zugehörigen Adapter konfiguriert, und die Rohdatendatensätze sollten bereits in der Tabelle T_RAW_DATA in CA Business Service Insight verfügbar sein. Es ist jetzt möglich, die Business-Logik auf die Events anzuwenden, um das eigentliche Service Level-Event für jede Metrik zu erzeugen.

Das Business-Logik-Skripting ist ein Prozess, bei dem Codes geschrieben werden, die logisch auf die Rohdaten (von Adaptern gesendete Rohdaten) angewendet werden, um die Service Levels zu berechnen.

Für jede Metrik gibt es eine eigene Business-Logik-Formel, anhand derer der aktuelle Service Level berechnet wird, obwohl viele der Metriken im System normalerweise eine gemeinsame Logik besitzen, die auf verschiedene Rohdaten-Eventsätze angewandt werden kann.

Beispielsweise werten eine Metrik, die die Bearbeitungszeit einer Anfrage mit dem Schwierigkeitsgrad 1 berechnet, und eine andere Metrik, die die Bearbeitungszeit einer Anfrage mit dem Schwierigkeitsgrad 2 berechnet, unterschiedliche Aufzeichnungssätze aus. Die eine verwendet nur die Anfragen mit dem Schwierigkeitsgrad 1 und die andere nur diejenigen mit dem Schwierigkeitsgrad 2. Beide würden jedoch höchstwahrscheinlich dieselbe Methode für die Berechnung der Bearbeitungszeit anwenden. Das Bearbeitungszeit-Skript wird einmal entwickelt und geprüft, als Business-Logik-Modul definiert und dann von beiden Metriken durch Einbeziehen dieses Moduls in die Business-Logik der Metriken angewandt.

Daher werden beim Entwickeln von Business-Logik-Skripten normalerweise die wichtigsten Module oder Vorlagen entwickelt, damit sie allen Metriken des Systems zur Verfügung stehen. Außerdem stellt jede Domänenkategorie normalerweise einen anderen Messungstyp dar und folgt daher im Allgemeinen einem anderen Business-Logik-Modul bzw. einer anderen Vorlage.

Workflow beim Business-Logik-Skripting

Die Phase des Business-Logik-Skriptings beinhaltet die Durchführung folgender Schritte:

- Formel definieren

  Erstellen Sie eine Formel basierend auf den Berechnungsanforderungen, die in der Designphase definiert wurden. Bei den definierten Formeln handelt es sich durchweg um einmalige Formeln, die in den Vertragsmetriken in ihren verschiedenen Abwandlungen jeweils als Business-Logik-Modul benutzt werden.

  Enthält ein Vertrag zum Beispiel drei Metriken für die Berechnung der durchschnittlichen Problemlösungszeit pro Anfrage und eine Metrik für jede Anfragepriorität, wird für die Berechnung der Anfrage-Problemlösungszeit eine einzige Formel mit dem Parameter Anfragepriorität entwickelt. Nach dem Testen wird diese Formel als Modul definiert und allen anderen relevanten Metriken angehängt.

- Formel testen

  Es werden Prüfungen durchgeführt, um zu gewährleisten, dass die Formel korrekt und fehlerfrei definiert wurde und die Berechnungen das erwartete Ergebnis liefern. Es ist wichtig, alle Extrem- und Grenzfälle bei den Prüfungen zu berücksichtigen. Der Business-Logik-Umfang ist der Bereich, in dem die Formel beim Prüfen ausgeführt wird. Nachdem eine Formel definiert wurde, wird sie in ihrer Gesamtheit geprüft. Wenn sie später als Modul auf alle Metriken angewandt wird, muss jede Metrik mindestens einmal im Business-Logik-Umfang ausgeführt werden, damit man sieht, ob sie Events erhält (d. h. dass die Erfassung korrekt durchgeführt wurde) und dass sie sinnvolle Events liefert.

- Das zugeordnete SLALOM-Modul definieren

  Jedes Modul ist eine einmalige Business-Logik-Berechnung und kann mit ihrer Parameterdefinition auf alle relevanten Metriken angewendet werden. Bei der Definition des Moduls ist es wichtig, das Modul gründlich zu testen und detailliert zu dokumentieren. Dies bedeutet: Was bewirkt das Modul (Beschreibung der Berechnung), welche Parameter erwartet es (Name, Bedeutung und Verwendung) usw.

- Alle Metriken an das entsprechende Business-Logik-Modul anfügen

  Für jede Metrik in den definierten Verträgen sollte eine Verknüpfung zum entsprechenden Business-Logik-Modul angelegt werden. Diese sollte dann im Business-Logik-Umfang ausgeführt werden, damit gewährleistet ist, dass die Verknüpfung korrekt implementiert wurde und die Erfassung richtig funktioniert, die entsprechenden Events empfangen und erwartete Events produziert werden.

Business-Logik-Module

Es gibt eine Reihe wichtiger Überlegungen, die beim Erstellen von Modulen für eine Business-Logik berücksichtigt werden müssen, insbesondere wenn Sie mehrere Module innerhalb einer einzelnen Metrik verwenden:

- Um sicherzustellen, dass die Verwendung eines Moduls klar ist, sollten Sie einen Kommentar am Anfang der Business-Logik für diese Metrik hinzufügen.
- Wenn Sie ein kurzen Code innerhalb des Business-Logik-Raums der Metrik verwenden und einen oder mehrere Module in den Code einbinden, sollten Sie sicherstellen, dass Sie diesen Code-Abschnitt bei allen Standardeventsroutinen oder Subroutinen aus der Business-Hauptlogik der Metrik entfernen. Sie müssen sich vergewissern, dass jede Subroutine und Eventroutine nur *einmal* in jedem der Module definiert ist, die von einer bestimmten Metrik verwendet werden. Dies soll verhindern, dass die Berechnungs-Engine durcheinander gebracht wird und unerwartete Events erzeugt.

  Hinweis: Wenn Sie zum Beispiel die Funktion OnPeriodStart() in Ihrem Modul definieren, dort einen spezifischen Code einfügen und dann die Standardfunktion OnPeriodStart() ohne Code im Business-Logik-Hauptfenster Ihrer Metrik belassen, weiß die Engine während der Laufzeit nicht, welche Subroutine sie ausführen soll. Sie führt möglicherweise einen anderen Code aus als den, den Sie beabsichtigt haben.
- Sie müssen äußerst vorsichtig sein, wenn Sie die OnRegistration-Subroutine parametrieren. In einigen Fällen kann dadurch der automatische Auslöser unterbrochen werden, der innerhalb des Systems erstellt wurde, um die Metriken basierend auf der Hinzufügung neuer Rohdaten neu zu berechnen.

Innerhalb der Business-Logik

Die gesamte Business-Logik befindet sich in einem eventgesteuerten Skript, das auf der Syntax von VBScript basiert. Jedes Event, das die Business-Logik erreicht, löst einen Event-Handler aus.

Die folgenden Typen von Events werden von der Engine gesendet und müssen von der Business-Logik evaluiert werden:

- Rohdaten-Events. Tatsächliche Rohdateneingaben, auf die die Business-Logik ihre Events stützt. Die Engine sendet nur die relevanten Rohdaten-Events, die auf der Erfassung der Formel basieren.
- Engine-Events. Von der Engine implizit gesandte Events, die die Eigenschaften der Metrik-Definition reflektieren, wie z. B. Zeitfensterdefinition und Kontrollzeiträume.
- Intermediäre Events. Events, die von anderen Business-Logik-Skripten erstellt wurden und von anderen verwendet werden können.

Die in der Metrik-Definition enthaltene Business-Logik-Formel ist für die Beurteilung der Events und Ausgabe eines Service Level-Events verantwortlich, auf dessen Grundlage die Berichte erstellt werden. In Abhängigkeit von diesen Service Level-Events und der Domänen-Definition erzeugt die Engine auch ein Abweichungsergebnis (wenn ein Service Level-Ziel auf die Metrik angewandt wurde). Die erzeugten Events werden in einer Datenbanktabelle mit dem Namen "T_PSL" gespeichert. Es ist diese Tabelle, die vom Berichtassistenten abgefragt wird, wenn Berichte erstellt werden. Daher werden alle Berichtsdaten vorberechnet, um eine maximale Leistung der Berichterstellung sicherzustellen.

Eventfluss

Wie zuvor erwähnt, sind die Eingaben für die Business-Logik die Engine-Events und die Rohdaten-Events.

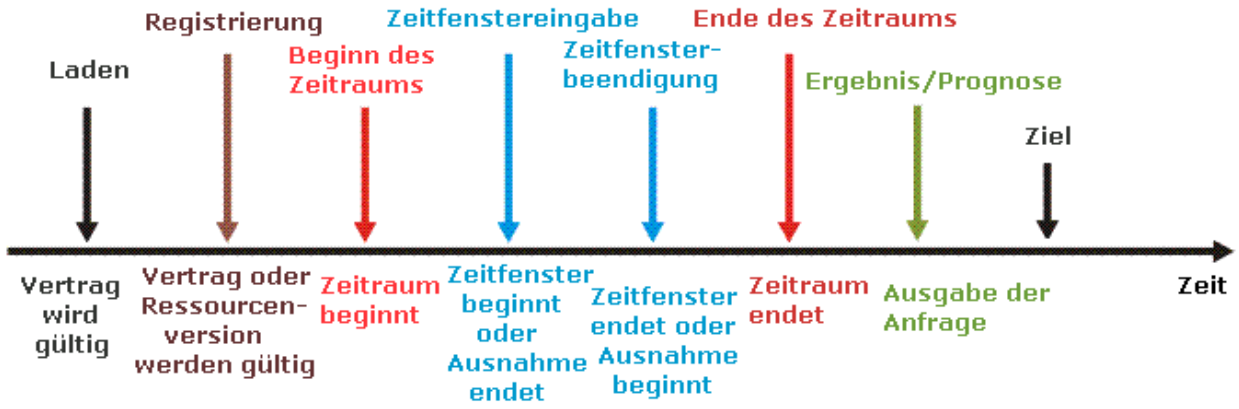
Von der Business-Logik empfangene Rohdaten-Events werden von der Erfassungsfunktion bestimmt, bei der der Code einen bestimmten Satz an Rohdaten-Events anfordert, die durch ihren Event-Typ und den Ressourcen-Bezeichnern definiert werden.

In der Business-Logik ordnet die Erfassung auch eine benutzerspezifische Subroutine zu, die ausgeführt wird, um das Rohdaten-Event zu verarbeiten, sobald ein Event erfasst wird. (Standardmäßig ist dies das OnXXXEvent, das sinnvoll umbenannt werden sollte.)

Die Engine-Events werden von der Engine entsprechend den zugeordneten Vertrags- und Metrik-Definitionen ausgelöst. Sobald das Engine-Events ausgelöst und empfangen wird, führt die Engine die angemessene Eventroutine aus. Jedes Engine-Event hat eine implizite Eventroutine. Diese Eventroutinen sind Funktionen und Verfahren, die oben im VBScript definiert sind. Die Eventroutine, die die Erfassung ausführt, und die "Ergebnis"-Funktion müssen beide zwingend in den Code implementiert werden. Alle anderen Eventroutinen sind optional. Allerdings befasst sich die Business-Logik nicht mit Engine-Event, für die keine Eventroutinen implementiert wurden. Deswegen ist es eine gute Vorgehensweise, sie an Ort und Stelle zu belassen (auch wenn nicht verwendet werden), um zukünftige Erweiterungen zu ermöglichen.

Hinweis: Beim Schreiben des Business-Logik-Skripts ist es wichtig, alle Engine-Events zu implementieren, um alle endgültigen Möglichkeiten abdecken zu können. Auch wenn zum Beispiel während der ersten Phase der Implementierung keine Zeitfenster-Definition verfügbar war, dies jedoch zukünftig der Fall sein sollte, dann müssen alle Formeln geändert werden. Es wird daher empfohlen, dass der Business-Logik-Experte das Verhalten "in timeslot" und "out of timeslot" weit im Voraus festlegt, auch wenn diese Verhaltensweisen anfänglich nicht verfügbar sind. So sind die erforderlichen Änderungen geringfügig, wenn das Verhalten eingeführt wird.

Nachfolgend finden Sie verschiedene Engine-Events und ihre Eventroutinen:



- **OnLoad (Time)** - (optional) wird einmal zu Beginn der Berechnungen aufgerufen, wenn der Vertrag aktiviert wird. Kann für die Initialisierung globaler Variablen verwendet werden.
- **OnRegistration (Dispatcher)** - (erforderlich) Methode, mit dem die relevanten Rohdaten-Events angefordert und mit den benutzerspezifischen Verfahren verknüpft werden, mit denen sie verarbeitet werden. Die Events werden angefordert und mithilfe der Methoden des Dispatcher-Objekts mit den Verfahren verknüpft.
  OnRegistration - wird von der Berechnungs-Engine einmal am Anfang der Metrik-Berechnung und dann jedes Mal, wenn eine für die Metrik registrierte Ressource wirksam wird, aufgerufen. Daraufhin beurteilt die Engine den Satz von Änderungen, die an dieser Ressource vorgenommen wurden und sich auf den Eventsatz auswirken können, der an die Formel weitergeleitet wird. Die Engine umfasst die Definition der Event-Anfrage, die vom Event-Typ und den Ressourcen-Bezeichnern definiert wird, und in den Fällen, in denen sich eine Ressource (oder ein Änderungssatz, der mehrere Ressourcen enthält) ändert, etwas, dass sich auf diesen Satz bezieht. Sobald es in Kraft tritt, wird ein Event Handler OnRegistierung ausgelöst.
- **OnPeriodStart (Time)**-(optional), am Anfang des Agentenzeit-Zeitraums aufgerufen (wird gemäß der Zeiteinheit festgelegt). Der erste OnPeriodStart wird ausgelöst, sobald der Vertrag in Kraft tritt, wobei die Bilanz der Zeiträume in ganzen Stundenzeiteinheiten anfängt. Dieser Event Handler wird üblicherweise verwendet, um in regelmäßigen Abständen Variablen zu initialisieren, wie beispielsweise einen Zähler, der die Anzahl an Vorfällen, die während des Berechnungszeitraums geöffneten wurden, feststellt, die dann zu Beginn eines jeden Zeitraums auf Null initialisiert werden sollten.
- **OnPeriodEnd (Time,IsCompleteRecord)** -(optional), wird am Ende des Agentenzeit-Zeitraums aufgerufen (gemäß der Zeiteinheit festgelegt). Es wird immer am gerundeten Ende der Zeiteinheit in runden Stunden aufgerufen (zum Beispiel 24:00). "IsCompleteRecord" ist "Wahr", wenn der Zeitraum der Metrik beendet ist (gemäß der Realitätszeit des Anwendungsservers), und "Falsch", wenn eine Zwischenberechnung durchgeführt wird. Dieser Event-Handler wird üblicherweise verwendet, um letzte Berechnungen für den Endzeitraum durchzuführen, und so die Grundlage für das von der Ergebnisfunktion gelieferte Endergebnis zu schaffen.
- **OnTimeslotEnter (Time)**-(optional), wird aufgerufen, wenn ein TimeSlot-Zeitraum gemäß der zugewiesenen metrischen Definition eingegeben wird. Beispiel: Wenn die Metrik mit einer Zeitfensterdefinition von Geschäftszeiten verbunden ist, wo jeden Tag um 08:00 ein Zeitfenster beginnt, wird der Event-Handler an jedem Tag um diese Uhrzeit ausgelöst.
- **OnTimeSlotExit (Time)**-(optional), wird aufgerufen, wenn ein TimeSlot-Zeitraum gemäß der zugewiesenen metrischen Definition eingegeben wird. Beispiel: Wenn die Metrik einer Zeitfensterdefinition von Geschäftszeiten zugewiesen ist, wo jeder Tag um 17:00 Uhr das Ende einer Zeitspanne ist, dann wird dieser Event Handler an jedem Tag um diese Uhrzeit ausgelöst.

- **Target()**-(optional), wird aufgerufen, wenn für die Metrik ein dynamisches Ziel festgelegt ist. Dadurch können Sie das Service Level-Ziel einer Metrik während der Business-Logik-Laufzeit bestimmen. Zu diesen Zielen können der Verbrauch von Servicekomponenten oder Infrastrukturänderungen zählen. Sie besteht aus vier Werten, wie bei der Ergebnisfunktion; einen für jeden Modus. Diese Funktion wird *nach* der Ergebnisfunktion im Rahmen der normalen Ausführung ausgeführt.
- **Forecast()**-(optional), wird ein Mal aufgerufen, wenn eine Übernahme der Vertragsversion durchgeführt wird, bei der die Berechnungs-Engine den Vertrag ein Mal in einem Prognosemodus kalkuliert. Der Prognosemodus führt einen vollen Berechnungs-Engine-Zyklus des Vertrags durch (vom Anfangs- bis zum Enddatum der Vertragsversion). In diesem Zyklus werden nur die Prognosewerte berechnet. (Es sind keine Berechnungen in der Tabelle T_RAW_DATA vorhanden). Diese Funktion wird *statt* der Ergebnisfunktion in diesem Ausführungsmodus, aufgerufen.
- **Result()**-(erforderlich), wird von der Berechnungs-Engine aufgerufen, um das Berechnungsergebnis für einen bestimmten Zeitraum zu erhalten. Die Ergebnisfunktion wird stets nach dem Event Handler OnPeriodEnd aufgerufen.

Nachfolgend sind die Schritte aufgeführt, denen der Berechnungsmechanismus (PslWriter-Service) zur Verarbeitung einer einzelnen Business-Logik-Formel folgt:

- PslWriter lädt die Formel in den Arbeitsspeicher und führt jede Erklärung aus, die im Abschnitt "Erklärungen" vorhanden ist (dieser Abschnitt "Erklärungen" umfasst jeglichen Code, der sich außerhalb einer Funktion oder Subroutine befindet). Beachten Sie zudem, dass alle angehängten Module und Bibliotheken eingeschlossen sind und zur Durchführung in dieses einzelne Codemodul kompiliert werden.
- PslWriter ruft den Event Handler OnLoad auf.
- PslWriter den Parameter OnRegistration auf.
- Bei der Berechnung des Zeitraums wird mit dem Aufrufen von OnPeriodStart begonnen.
- Für jedes während der OnRegistration protokollierte Rohdatenereignis, das in den Zeitbereich des Zeitraums fällt, ruft PslWriter den diesem Event zugewiesenen, benutzerspezifischen Event Handler auf.
- Wenn der Anfang oder das Ende des Zeitfensters in den Zeitbereich dieses Zeitraums fällt, wird OnTimeslotEnter oder OnTimeSlotExit aufgerufen.
- Wenn es eine relevante Infrastrukturänderung innerhalb des Zeitraums gibt, wird OnRegistration aufgerufen.
- Die Berechnung des Zeitraums endet mit dem Aufruf von OnPeriodEnd.
- Wenn ein dynamisches Ziel festgelegt wurde, wird diese Funktion aufgerufen.
- PslWriter ruft die Ergebnisfunktion auf, um das Endergebnis der Zeitraumberechnung zu erhalten.

Hinweis: Wenn die Vertragsversion zuerst übernommen wird, und eine Prognose ausgewählt wurde, wird die Prognosefunktion statt der Ergebnisfunktion aufgerufen. Dies trifft allerdings nur dieses eine Mal für jede Version zu.

Während jedes Berechnungszyklus bewertet die Engine, auf worauf die entsprechenden Engine-Events und Rohdatenereignisse im Berechnungszeitraum basieren. Sie sortiert sie zunächst nach der Zeit und sendet dann die relevanten Formeln zur Berechnung. Die Zeit des Rohdatenereignisses ist sein Zeitstempel; die Zeit des Engine-Events ist seine Aktivierungszeit. Beide Eventtypen werden dann in einer Zeitsequenz zusammengeführt und zur Berechnung gesendet.

Das Timing der Events entspricht der zugewiesenen lokalen Metrik, Parameter "Time" der Event-Handler (d. h. OnPeriodStart (Time)) und der Zeitstempel des Rohdatenereignisses entsprechen ihrem UTC-Wert. Die Engine vergleicht sie gemäß ihren UTC-Werten, um so einen konstanten Bezugspunkt zu verwenden.

***Beispiel*:**

Wenn der Zeitzonenunterschied einer UTC-Metrik zwei Stunden (d. h. GMT+2) beträgt, und ein Vorfallöffnungs-Event einen Zeitstempel von 10.00 Uhr hat, wird er Zeitstempel, den der Event Handler nutzt, in der Engine entsprechend verschoben und beginnt tatsächlich um 8.00 Uhr UTC. Ausgehend von der Annahme, dass der Adapter auf die Nutzung derselben Zeitzone konfiguriert ist, werden die Rohdatenereignisse auch in der Datenbank um 2 Stunden zurück auf UTC-Zeit zurückgesetzt. Wenn die Events zur Business-Logik weitergeleitet werden, nutzt der Berechnungsagent, der für die Berechnung der Events für den Zeitraum ab 10.00 Uhr zuständig ist, eigentlich die UTC-Zeit aller Events ab 8.00 Uhr. Wenn Sie die Meldung out.log im Code zum Ausdrucken der Zeitstempel verwenden, wird trotz des festgelegten Zeitraums von (gemäß der Metrik) 10.00 Uhr der auf UTC verschobene Zeitstempel - also 8.00 Uhr - angezeigt.

In der folgenden Berechnungsanforderung muss der Zeitstempel des Events konvertiert werden, bevor sie angewendet wird:

Wenn eine Metrik in der Berechnung der Häufigkeit von Vorfällen besteht, die an demselben Tag geschlossen wurden, an dem sie eröffnet wurden, muss die Öffnungs- und Schließungszeit eines jeden Vorfalls verglichen werden. Wenn die Öffnungs- und Schließungszeit des Vorfalls auf denselben Zeit fällt (und in dasselbe definierte Zeitfenster), wird der Vorfall gezählt.

Es kann passieren, dass sich der Tag bei der Verschiebung des Vorfalls von der ursprünglichen Lokalzeit zur UTC-Zeit ändert (d. h. dass wieder GMT+2 verwendet wird). (Ein Vorfall, der um 1.00 Uhr geöffnet wird, wird in UTC zurück auf 23.00 Uhr am Vortag verschoben). Er wird dann nicht gezählt, selbst wenn er eigentlich hätte gezählt werden müssen. Dies ist eine Situation, in der die Zeit zunächst zurück in die lokale Metrik-Zeit konvertiert werden muss, und erst dann verglichen werden kann. Verwenden Sie in einem solchen Fall die Methode GetLocaleTime(utcTime). Bei dieser Methode wird eine Zeit in einer UTC-Zeitzone in die entsprechende Zeitzone der lokalen Metrik konvertiert.

```
Nachfolgend finden Sie den Event Handler-Code:
Sub OnIncidentEvent(eventDetails)
```

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

Registrierung

Registrierung ist der Prozess, bei dem die Business-Logik eine Anfrage an die Berechnungs-Engine zum Festlegen von Rohdatenereignissen sendet, die in die Berechnung aufgenommen werden sollen.

Die Registrierung ist auf zwei Arten möglich: Mit dem Registrierungsassistenten oder manuell mit dem Dispatcher-Objekt in der Business-Logik.

Der Registrierungsassistent ist äußerst benutzerfreundlich - treffen Sie einfach eine Auswahl aus den verfügbaren Optionen. Sie haben dieselben Optionen zur Verfügung, wie bei der manuellen Registrierung, allerdings ohne die Möglichkeit zur Verwendung von Parametern. Wenn Sie Parameter verwenden müssen, müssen Sie die Registrierung manuell vornehmen. Die Grundstruktur des Assistenten erfordert, dass Sie zunächst bestimmten, welche Art von Registrierung Sie vornehmen wollen, dass Sie dann die Ressourcentypen und Events festlegen, für die die Registrierung durchgeführt werden soll, und dass Sie sich schließlich entscheiden, welcher Event Handler zur Verarbeitung der erfassten Events verwendet werden soll.

Sobald die Registrierungen abgeschlossen sind, werden sie auf der Registerkarte "Registrierung" der Metrik angezeigt. Beachten Sie zudem, dass es pro Metrik mehrere Registrierungsaussagen geben kann.

Tatsächlich nutzt der Registrierungsassistent dieselbe Funktionalität wie die manuelle Registrierung. Auf alle diese Optionen wird im folgenden Abschnitt näher eingegangen.

Wenn die Registrierung der Formel manuell aus der Business-Logik heraus vorgenommen wird, wird sie vom Event Handler *OnRegistration* vorgenommen. Dies muss in der Formel implementiert und bei jeder Aktivierung eines Registrierungs-Engine-Events ausgelöst werden. Das Registrierungs-Event wird ein Mal bei Aktivierung des Vertrages ausgelöst, und dann jedes Mal, wenn eine relevante Ressource oder ein Änderungssatz aktiviert werden. Es heißt, eine Änderung der betreffenden Ressource sei von Bedeutung, wenn sich diese auf Events bezieht, die die Metrik empfangen soll. Beispiel: Wenn die Registrierung von der Vertragspartei ausgeführt wird (*RegisterByContractParty*), bedeutet dies, dass alle Events des definierten Typs, deren Ressourcen der Vertragspartei der Metrik angehängt sind, Bestandteil der Berechnung sind. In diesem Fall wird die Registrierungsmethode jedes Mal, wenn eine neue Ressource der Vertragspartei angehängt wird bzw. wenn die Anhängung einer solchen Ressource aufgehoben wird, aktiviert, um die Engine über die Änderung zu informieren.

Die Registrierungsmethoden werden vom Dispatcher-Objekt bereitgestellt, das als Argument an OnRegistration übertragen wird. Die verschiedenen Methoden bietet unterschiedliche Arten zur Definition der Filterkriterien basierend auf der Event-Typ-Definition und Ressourcen-Zuweisungskriterien, wie Ressourcen einer Ressourcengruppe oder Ressourcen eines bestimmten Typs.

Die Verwendung der Registrierungsmethoden von "Vertragspartei" und "Service" wird nachdrücklich empfohlen, da es die Nutzung der Business-Logik als Modul oder Vorlage vereinfacht. Werden diese Registrierungsmethoden verwendet, werden die jeweilige Vertragspartei und der jeweilige Service der zugewiesenen Metrik-Definition entnommen. Bei der Wiederverwendung der Formel für andere Verträge und/oder Servicekomponenten muss die Registrierung dann nicht verändert werden.

Eine weitere beliebte Registrierungsmethode ist RegisterByResourceGroup, die praktisch ist für die Arbeit mit Ressourcen, die logisch zusammengefasst sind, jedoch möglicherweise nicht immer Vertragsparteien oder Services zugewiesen sind. Die Zuweisung von Ressourcen zu den Gruppen hier lässt sich dann über den Ressourcenkatalog (individuell oder über Änderungssätze) verwalten, und ließe sich sogar automatisch mithilfe von Übersetzungsskripten aktualisieren.

Im Allgemeinen wird die Registrierungsmethode während der Entwurfsphase ermittelt und direkt vom definierten Datenmodell gesteuert.

Hinweis: Zur Überprüfung, ob das Dispatcher-Objekt ordnungsgemäß angewendet wurde, wird die Funktion OnRegistration auch während der SLALOM-Syntaxüberprüfung aufgerufen. Daher sollte nicht davon ausgegangen werden, dass OnLoad vor der Funktion OnRegistration ausgeführt wurde. Ferner sollten nicht einfach einige der Eigenschaften des Kontextobjekts, wie "TimeUnit", "IntervalLength", "IsPeriod", "CorrectionsApply" und "ExceptionsApply" im Event Handler "OnRegistration" verwendet werden.

Die Registrierungsmethoden sind zudem zuständig für die Zuweisung der Events zu einem Verfahren, das gemäß dem Zeitstempel des Events aktiviert wird. Das definierte Verfahren kann beliebig benannt sein, es hat jedoch stets das Objekt eventDetails als seinen Parameter. Das Objekt eventDetails spiegelt das empfangene Rohdatenereignis wider und umfasst alle Event-Details als Datenfelder, wie in der folgenden Registrierung dargestellt:

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

Die oben aufgeführte Registrierungserklärung besagt, dass alle Rohdatenereignisse des Event-Typs "MemUse", die dem Ressourcentyp "Server" zugewiesen sind, an den Event Handler "OnMemUseEvent" in der Business-Logik gesendet werden.

Das folgende Verfahren muss ebenfalls vorab in der Business-Logik definiert werden:

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

Durch den Verweis auf das Objekt eventDetails-Objekt und durch die Anwendung des Parameters "MemoryUsage" extrahiert die Erklärung den Wert des Feldes "MemoryUsage" aus dem Event, das in die Funktion übernommen wurde. Diese Felder sind dieselben wie die, die im Event-Typ definiert sind. Bei den Feldnamen muss die Groß-/Kleinschreibung beachtet werden.

Registrieren geclusterter Metriken

Geclusterte Metriken ermöglichen die Definition einer Metrik, um für jedes einzelne Mitglied einer Ressourcengruppe dieselbe Definition und Logik für eine Reihe von Elementen anwenden zu können. Ein Clustern kann entweder statisch auf einem vordefinierten Satz an Ressourcen oder dynamisch auf die Ressourcengruppenmitglieder durchgeführt werden, während die Gruppe im Laufe der Zeit verändert werden kann und Mitglieder aus der Gruppe ein- oder ausgeschlossen werden können.

Hinweis: Eine detaillierte Beschreibung finden Sie im Anhang E - Definieren von Business-Logik-Formeln (Business-Logik-Experte).

Geclusterte Metriken werden verwendet, wenn ein Service Level-Ergebnis für jedes Element in einer Ressourcengruppe berechnet werden muss. Die Elemente in einer Ressourcengruppe können entweder Ressourcen oder andere Ressourcengruppen sein, daher muss die Erfassungsmethode in einem Business-Logik-Skript einer geclusterten Metrik RegisterByResourceGroup oder RegisterByResource sein, wobei der definierte Ressourcenname oder der definierte Ressourcengruppenname als Element im Cluster festgelegt ist. Dazu wird die Eigenschaft "ClusterItem" des Kontextobjekts verwendet, das den Namen des aktuellen Clusterelements enthält.

Beispiel:

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

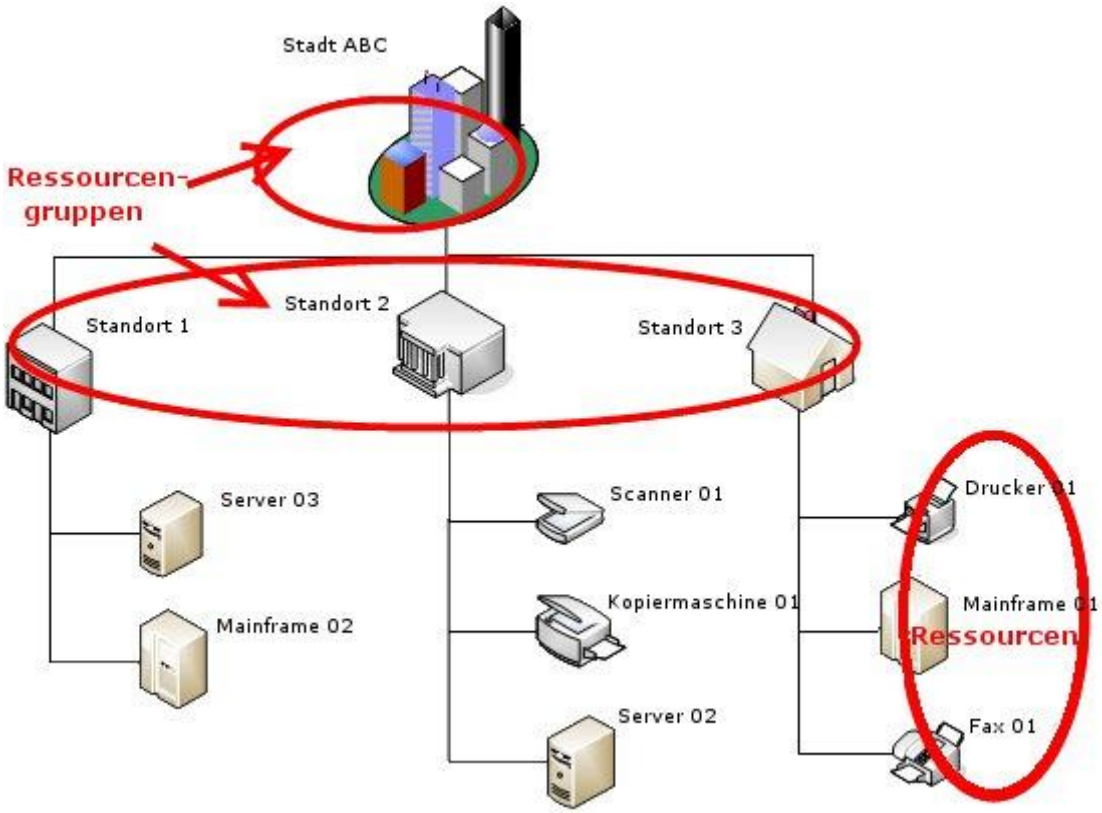
In Fällen, in denen diese Registrierungsmethode verwendet wird, berechnet die Metrik ein Ergebnis für jede der Ressourcen in der Ressourcengruppe, über die die Metrik geclustert wird,

– ODER –

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

In Fällen, in denen diese Registrierungsmethode verwendet wird, berechnet die Metrik ein Ergebnis für jede der Ressourcengruppen, die in der Ressourcengruppe, für die die Metrik geclustert ist, enthalten sind.

Das Clustering kann, je nachdem, wie das Ressourcenmodell erstellt wird, auf verschiedenen Ebenen erfolgen. Es ist oft der Fall, dass Firmen verschiedene Gruppierungsebenen haben, die sie präsentieren wollen. Beispiel: In einer bestimmten Stadt gibt es möglicherweise eine Reihe von Standorten und innerhalb eines jeden Standortes kann es eine Reihe von Infrastrukturgeräten geben. (Drucker, Scanner, Server usw.). Mit dieser Gruppierungsart können Sie eine definierte Ressourcenhierarchie strukturieren, die mehrere Ebenen (Level) und Gruppierungen dieser Hardwarekomponenten enthält. (Vorausgesetzt, ein Infrastrukturgerät ist die "Ressource".) Die hier beschriebene Struktur könnte folgendermaßen aussehen:



Wie anhand des Diagramms ersichtlich ist, gibt es mehrere Gruppenebenen. Die Gruppe "Stadt ABC" der obersten Ebene umfasst drei verschiedene Standorte (die auch Ressourcengruppen sind). Die Ressourcengruppe "Ressourcengruppe des Standortes 3" umfasst drei verschiedene Ressourcen. Um also gemäß dem vorherigen Beispiel die Metrik übergreifend über die drei verschiedenen Standorte zu clustern, benötigten Sie die folgende Registrierung:

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

In diesem Fall bezieht sich Kontext.ClusterItem auf die Ressourcengruppe "Standorte der Stadt ABC", die drei andere Ressourcengruppen - "Ressourcen von Standort 01", "Ressourcen von Standort 02" usw. enthält, und auf der Registerkarte "Clustering" der Metrik folgendermaßen aussehen kann.

Beachten Sie ferner, dass das Clustering auf "Dynamisch" festgelegt ist, da dabei automatisch alle Änderungen an der Gruppe sofort, wenn diese vorgenommen werden, übernommen werden. Statisches Clustern kann für Untergruppen von Ressourcengruppen hilfreich sein, oder wenn Sie nicht wollen, dass sich das Clustering im Laufe der Zeit ändert.

Nutzen Sie zur Erstellung einer Metrik, die über die Ressourcen der Gruppe "Standort 3) berichtet, die folgende Registrierungserklärung:

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

In diesem Fall bezieht sich das Kontext.ClusterItem auf die individuellen Ressourcen und lassen sich somit nur nach Ressource registrieren. Die Registerkarte Clustering der Metrik enthält einen Verweis auf die Gruppe "Ressourcen des Standortes 03".

Sie können das Clustern konfigurieren, um auf unterschiedlichen Ebenen der Hierarchie innerhalb einer einzelnen Metrik zu arbeiten. Beispiel: Unter Annahme der im vorherigen Beispiel beschriebenen Situation und bei nochmaliger Gruppierung (Clustering) der Metrik übergreifend über die Gruppe "Standorte der Stadt ABC". Sie können die Ressourcenmitglieder von verschiedenen Hierarchieebenen in eine einzige Metrik einschließen. In diesem Fall gibt es drei Optionen, welche Ressourcen in dieser Gruppierung enthalten sein können:

- **Nur erste Ebene: Direkte Mitglieder**: Entspricht den zuvor beschriebenen, älteren geclusterten Methoden.
- **Alle Level: Nur Ressourcen einschließen**: Schließt alle Ressourcen ein, die in den drei Standort-Ressourcengruppen auf einer einzelnen Ebene enthalten sind, und berechnet für sie individuell den Service Level.
- **Alle Level: Ressourcen und Ressourcengruppen einschließen**: Schließt alle in den Ressourcengruppen der drei Standorte enthaltenen Ressourcen sowie die drei Ressourcengruppen ein, und berechnet für sie individuell den Service Level.

Agenten

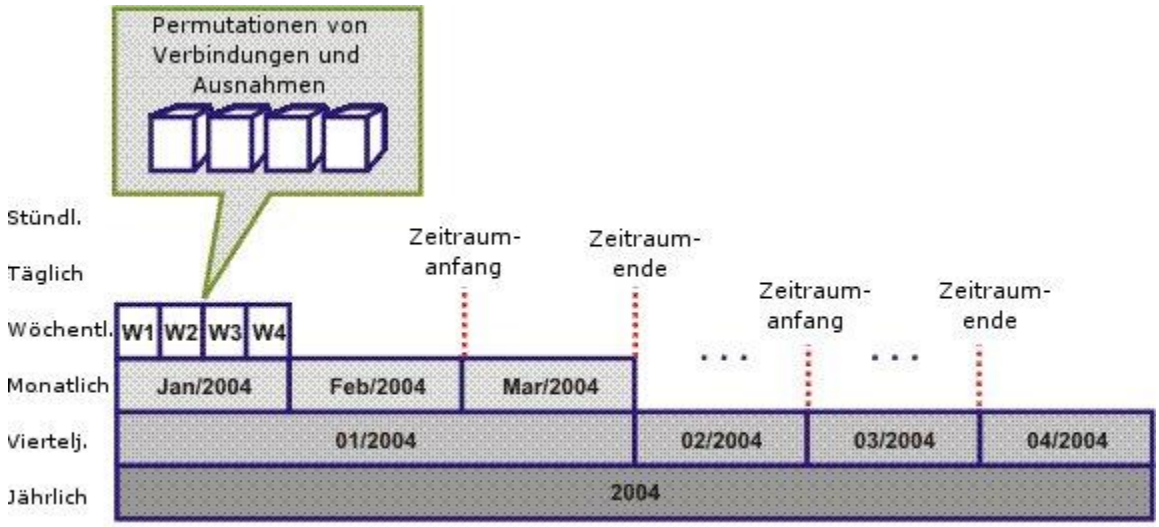
Jede Metrik verfügt über eine Definition eines Kontrollzeitraums. Der Kontrollzeitraum ist der Zeitraum, in dem die Metrik ein Service Level-Ergebnis ausgeben muss. Diese Ergebnis sollte dann mit dem definierten Ziel verglichen werden. Bei dem für den Kontrollzeitraum erzeugten Ergebnis handelt es sich um das Geschäftsergebnis, mit anderen Worten, es ist das Vertragsergebnis, für das sich der Anbieter zur Bereitstellung eines bestimmten Ziel-Service Levels verpflichtet hat. Jede Instanz der Business-Logik für jeden Zeitraum wird als Agent bezeichnet. Dieser bezieht sich direkt auf die der jeweiligen Metrik zugewiesenen Granularität.

Beispiel: Wenn die Metrik mit einem Kontrollzeitraum von einem Monat definiert ist, wird die Metrik ausgeführt, um ein monatliches Ergebnis zu liefern.

Zur Bereitstellung einer Verfeinerungsoption, mit der der Benutzer das monatliche Ergebnis so verfeinern kann, dass das tägliche Ergebnis eingesehen werden kann, muss die Metrik über eine Definition weiterer Zeiteinheiten verfügen, innerhalb der sie berechnet werden sollte. Die Zeiteinheiten werden im Detailabschnitt "Allgemein" der Metrik auf der Registerkarte "Granularität" definiert.

Für jede der auf der Registerkarte "Granularität" der Metrik und für den Kontrollzeitraum definierten Zeiteinheiten wird von der Engine eine separate Business-Logikinstanz ausgeführt. Jede dieser Instanzen führt denselben Business-Logikcode aus, OnPeriodStart und OnPeriodEnd werden jedoch für jede dieser Instanzen anders aktiviert. Für die tägliche Instanz wird er am Anfang und am Ende eines Tages aktiviert. Für jede Zeiteinheit wird er gemäß den Start- und Endpunkten der Zeiteinheit aktiviert.

Jede der Business-Logik-Instanzen wird ausgeführt, um das entsprechende Zeiteinheitsergebnis zu generieren. Darüber hinaus muss jeder Zeitraum über ein Ergebnis verfügen, bei dem alle Ausnahmen berücksichtigt werden. Jeder Zeitraum, der diese Kriterien nicht erfüllt (wenn Ausnahmen definiert sind), muss seinem Benutzer die Möglichkeit zur Wahl geben, ob das Service Level-Ergebnis mit oder ohne die berücksichtigten Ausnahmen angezeigt werden soll. Aufgrund dieser Anforderung verfügt jede Metrik potenziell über 14 Agenten (Instanzen), die von der Engine ausgeführt werden, zwei Agenten für die Geschäftsergebnisse und 12 für die sechs zusätzlichen Betriebszeiträume.



Business-Logik - Zeitskala

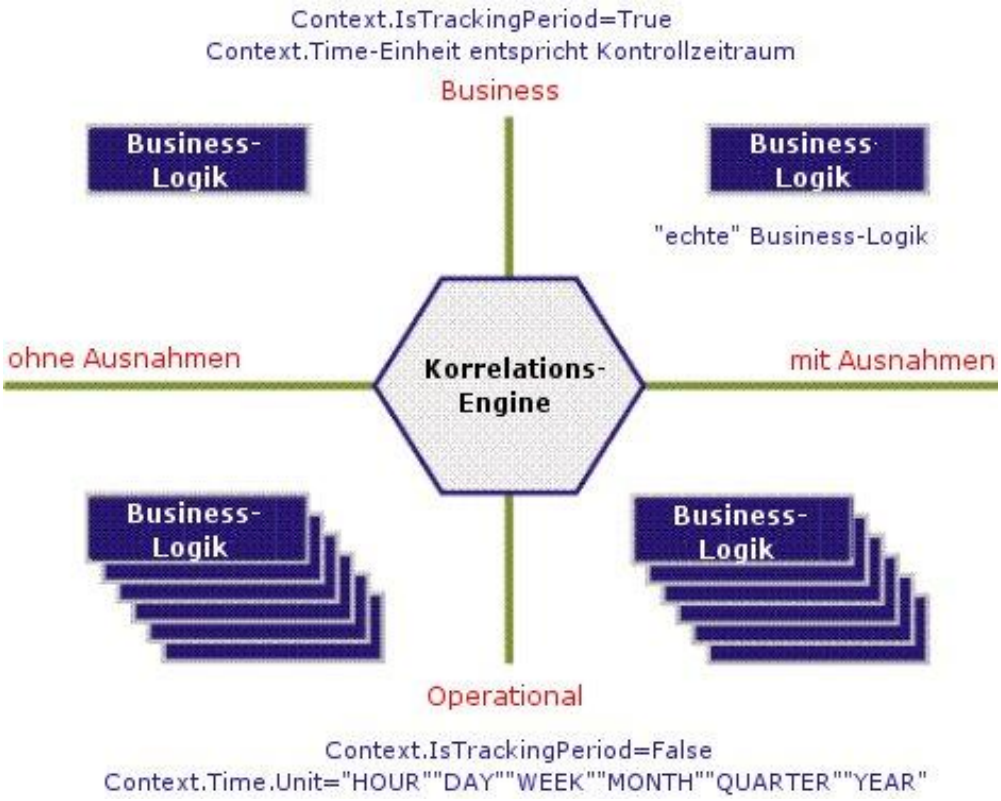
Folglich hat die Granularitätsdefinition große Auswirkungen auf die Leistung der Berechnungs-Engines, da jeder Zeitraum für einen anderen Agenten berechnet wird. Daher wird in Fällen, in denen die Verfeinerungsoption entweder nicht oder nur partiell erforderlich ist, empfohlen, einige der Agenten zu deaktivieren. Dies wirkt sich besonders nachhaltig bei geringeren Granularitäten, wie stündlichen Ergebnissen, aus. Es wirkt sich auch maßgeblich auf die geclusterte Metrik aus, da die Engine jede der oben aufgeführten Berechnungen für jedes erkannte ClusterItem vornimmt. Tatsächlich wird jedes ClusterItem wie eine neue Metrik behandelt. Beispiel: Bei der Berechnung einer Metrik übergreifend über eine Ressourcengruppe aus 50 Elementen hat die Engine im Vergleich zu derselben, nicht-geclusterten Metrik das 49-fache an Arbeit.

Beispiel: Wenn die Metrik auf die Berechnung der Problemlösungszeit in der Anzahl an Tagen festgelegt ist, ist ein stündliches Ergebnis sinnlos, und die Auswahl des entsprechenden Kontrollkästchens sollte auf der Registerkarte "Granularitäten" aufgehoben werden, damit die Engine keine unnötigen Berechnungen durchführt.

Das Attribut TimeUnit des Kontextobjekts (d. h. context.TimeUnit in der Business-Logik) gibt die Zeiteinheit des derzeit ausführenden Agenten aus, wobei die folgenden Ausgabewerte möglich sind: STUNDE, TAG, WOCHE, MONAT, QUARTAL, JAHR.

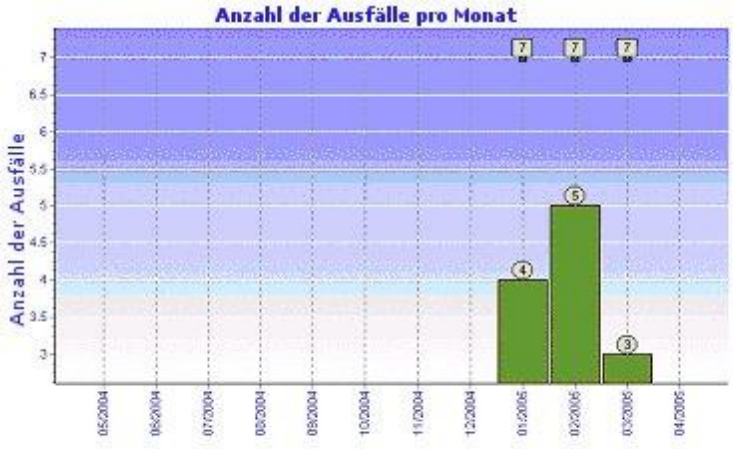
So gibt beispielsweise die Kontext.TimeUnit für den täglichen Agenten "Tag" aus.

Das Attribut IsTrackingPeriod des Kontextobjekts gibt "True" für den Agenten aus, der die Zeiteinheit für den Kontrollzeitraum berechnet. Es muss ferner berücksichtigt werden, dass die auf der Registerkarte "Granularität" der Metrik angezeigten Agenten zusätzlich zum Agenten des Kontrollzeitraums der Metrik vorhanden sind. Somit können Sie sogar für eine Metrik mit einem monatlichen Kontrollzeitraum den monatlichen Agenten deaktivieren, und er berechnet dennoch weiterhin den monatlichen Service Level, jedoch nur als "Geschäftsergebnisse" (nicht-operative Ergebnisse).

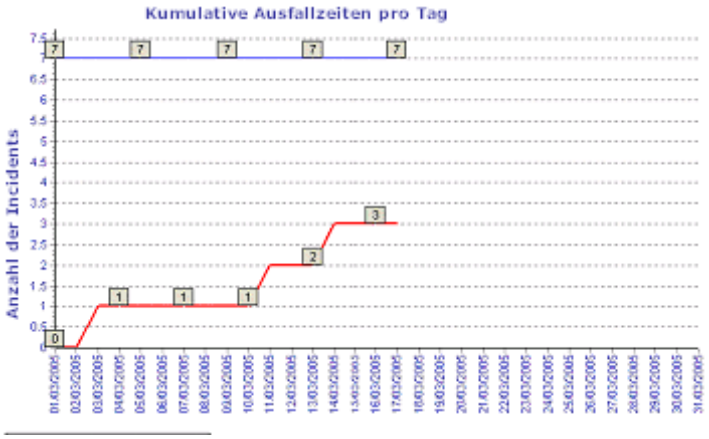


Wie oben erwähnt, führen die verschiedenen Agenten üblicherweise denselben Business-Logikcode aus, es gibt jedoch Fälle, in denen es notwendig ist, eine geringfügig andere Logik anzuwenden.

Im monatlichen Fall sollte das Ergebnis beispielsweise die Häufigkeit der Ausfallzeiten für jeden Monat separat ausweisen, wie nachfolgend dargestellt:



Für die tägliche Verfeinerung ist es möglicherweise notwendig, die kumulativen Ausfallzeiten einzusehen, wobei jeder Tag eine Summe aller Tage ab Monatsanfang ist. Diese Methode sollte auf alle Zeiteinheiten angewendet werden, die - wie unten dargestellt - kleiner sind als ein einzelner Monat:



Der Unterschied zwischen den zwei Zeiteinheiten ist, dass der Ausfallzeitzähler für den Agenten, der den Kontrollzeitraum berechnet, auf 0 initialisiert wird, für den täglichen Agenten wird der Zähler jedoch nur initialisiert, wenn es sich bei dem Tag um den ersten Tag des Monats handelt.

Nachfolgend sehen Sie den Event Handler OnPeriodStart:

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

Was ist unter Neuberechnung zu verstehen?

^Die Neuberechnung wird durchgeführt, wenn die Korrelations-Engine feststellt, dass ein periodisches Endergebnis einer Metrik nicht mehr gültig ist, und die Ergebnisse somit neu berechnet werden müssen.

Eine Neuberechnung kann durch Folgendes verursacht werden:

- Den Empfang neuer Events, die eigentlich in der Vergangenheit stattfanden. (früher als bis zu dem Zeitpunkt, bis zu dem die Engine die Berechnung für diese Metrik *derzeit* durchgeführt hat)
- Eine Ressource, die für die Metrik registriert ist, wird verändert (neues Versionsdatum und an der Ressource vorgenommene Änderungen).
- Übernehmen einer neuen Vertragsversion, die die zuvor berechnete Zeit mit Änderungen an einigen Metriken überlagert (nur veränderte Metriken werden neu berechnet)

Die effizienteste Methode für eine Registrierung ist über Vertragspartei und Service. Durch die Anordnung der Ressourcen unter Vertragspartei und Service lässt sich die logische Beziehung zwischen der Datenebene und der Geschäftsebene im System ausdrücken. Bei einer Registrierung von Ressourcen durch diese Entitäten sind keine Änderungen an den Formeln erforderlich, wenn sie in unterschiedlichen Verträgen oder für unterschiedliche Services verwendet werden. Der aktuelle Kontext der Metrik, der im Vertrag und im Service identifiziert ist, und der die relevante Vertragspartei und den zugehörigen Service festlegt. Die in diesem Typ von Registrierung definierten Business-Logik-Formeln sind leicht wiederverwendbar, weil sie keinerlei Änderungen in der Registrierung benötigen.

Ausgaben - Benutzertabellen

Das Standard-Business-Logik-Skript hat keinen Zugriff auf externe Ausgabetabellen. Es gibt nur zwei Ausgabeziele:

- Die PSL-Tabelle (T_PSL), in die die Engine automatisch die Service Level-Ergebnisse schreibt, und in die das in der Ergebnisfunktion festgelegte Ergebnis geschrieben wird. Service Level-Werte, die in diese Tabelle geschrieben werden, können nur nummerischer Natur sein. Bei den in die T_PSL-Tabelle geschriebenen Ergebnissen handelt es sich um die Ergebnisse, über die der Berichtassistent berichtet. Es gibt keine Kontrolle über die Struktur oder die Methode, in der diese Ergebnisse geschrieben werden. Dies wird automatisch von der Berechnungs-Engine ausgeführt.
- Die SLALOM-Ausgabetabelle (T_SLALOM_OUTPUTS). Das Schreiben in diese Tabelle erfolgt mithilfe der im Tool-Objekt bereitgestellten Methoden aus der Business-Logik-Formel heraus. Bei einer Ausgabe an diese Tabelle wird ein logischer Tabellenname bereitgestellt, der als Benutzertabelle bezeichnet wird. Diese Tabellen dienen bei der Service Level-Berechnung zur Ausgabe von Daten. Diese Daten können später zur Erstellung von Freiformberichten verwendet werden. Es kann viele Benutzertabellen geben.

Die Verwendung der externen Tabelle T_SLALOM_OUTPUTS ist erforderlich, wenn eine zusätzliche Ausgabe über das periodische Service Level-Ergebnis hinaus benötigt wird, diese zusätzliche Ausgabe jedoch nicht durch das Hinzufügen einer weiteren Metrik erreicht werden kann, oder wenn durch das Hinzufügen einer weiteren Metrik die Rechenleistung verringert wird, indem dieselbe Datensatzgruppe durchgegangen wird, nur um eine andere Ausgabe zu liefern.

Angenommen eine Metrik ist auf die Berechnung des Prozentsatzes an Tickets eingestellt, die in weniger als einem Tag gelöst wurden, und es soll ein Bericht mit einer Liste aller Tickets, die nicht in weniger als einem Tag gelöst wurden, erstellt werden. In diesem Fall ist muss die Formel jedes Ticket, das die Kriterien nicht erfüllt, in eine externe Tabelle ausgeben und es gleichzeitig zur Rechenstatistik hinzufügen.

Mit der oben aufgeführten Anforderung kann die reguläre Service Level-Ausgabetabelle diese Ausgabe nicht bieten, da:

- Alle Serviceergebnisse numerischer Natur sind
- Für jeden Zeitraum nur ein einzelnes Service Level-Ergebnis möglich ist.

Datensätze werden nur für den Agenten in die Benutzerausgabetabellen geschrieben, der für den Kontrollzeitraum der Metrik ausgeführt wird, und der die Ausnahmen und Korrekturen berechnet. Beispiel: Wenn die Beispielsweise so definiert wird, das sie über einen monatlichen Kontrollzeitraum verfügt, werden die Ausgabeerklärungen (Tools.SaveRecord und Tools.SaveFields) NICHT ausgeführt, wenn die Engine die Formel für die anderen Granularitäten, wie STUNDE, TAG, WOCHE, QUARTAL und JAHR, ausführt.

Ein zusätzlicher Nutzen einer externen Tabelle ist, dass sie für mehrere Berichterstellungsanforderungen verwendet werden kann. Von diesen Tabellen lassen sich zusätzlich zu den vertraglichen Anforderungen andere typische Berichterstellungsanforderungen generieren. Eine Metrik, die beispielsweise die "Anzahl der geschlossenen Tickets" in einem Monat berechnet, könnte auch die Ticket-Problemlösungszeit berechnen und all diese Daten in die Ausgabetabelle ausgeben. Dies ermöglicht eine detailliertere Analyse der einzelnen Tickets, die innerhalb des Zeitraums geschlossen wurden, sowie zusätzlicher Einzelheiten, die in Bezug auf eine separaten Berichterstellungsanforderung möglicherweise erforderlich sind.

Die Daten in diesen Tabellen werden ebenfalls über den Engine-Neuberechnungsmechanismus verwaltet. Während dieses Neuberechnungsprozesses werden Datensätze als inaktiv gekennzeichnet, und stattdessen wird eine neue Zeile geschrieben. Dies ist ein wichtiger Punkt, den es zu bedenken gilt, da alle Freibericht-SQL-Erklärungen eine Überprüfung des Feldes IS_ACTIVE-Feld in der T_SLALOM_OUTPUTS-Tabelle umfassen müssen (wobei is_active = 1), da nur diese Datensätze aktuell sind.

Hinweis: Wenn der Business-Logik-Umfang während des Fehlerbehebungsprozesses der Formeln ausgeführt wird, werden die Meldungen tatsächlich in die Tabelle T_DEBUG_SLALOM_OUTPUTS und nicht in die Tabelle T_SLALOM_OUTPUTS geschrieben.

Bei der Dokumentation von Daten mit T_SLALOM_OUTPUTS handelt es sich bei den eingegebenen Daten stets um Text (da die Felder von T_SLALOM_OUTPUTS alles varchar2 sind). Daher werden Datumswerte unter Anwendung des Betriebssystemformats in Text konvertiert, welches sich während des Lebenszyklus der Anwendung ändern kann. Bei T_SLALOM_OUTPUTS kann es daher zu Inkonsistenzen in den Datumsformaten kommen. Zudem verarbeitet die Business-Logik UTC-Daten, wobei zu erwarten ist, dass T_SLALOM_OUTPUTS lokale Zeitstempel umfasst, sodass es in einigen Fällen erforderlich sein kann, die Konvertierungsfunktion *Tools.GetLocaleDate(date)* als Umgehungslösung zu verwenden.

```
Mit der folgenden Funktion werden Datumsangaben in lokale Uhrzeiten konvertiert,
wobei die Datumsformat-Konsistenz durch die Formatierung der Datumsangaben in das
Format "dd/mm-/yyyy-hh24:mi:ss-" gewahrt bleibt:
Function FormatDate(time)
     Dim LocalTime
     LocalTime=Tools.GetLocaleTime(time)
     FormatDate=Day(LocalTime) & "/" & Month(LocalTime) & "/" &
  Year(LocalTime) & " " & _
     Hour(LocalTime) & ":" & Minute(LocalTime) & ":" & Second(LocalTime)
End Function
```

Es kann auf zwei Arten aus der Business-Logik-Formel heraus in die externe Tabelle geschrieben werden:

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

Beide Methoden des Tool-Objekts sind nachfolgend ausführlicher beschrieben:
`Tools.SaveRecord tableName, key,[val1],[val2],…`

Diese Methode speichert einen Datensatz in einer Tabelle namens T_SLALOM_OUTPUTS. Der Parameter tableName legt die (virtuelle) Tabelle in T_SLALOM_OUTPUTS fest, in die die Daten geschrieben werden sollen. Jeder Datensatz in der Benutzertabelle verfügt über einen eindeutigen Schlüssel, der angibt, in welchen Datensatz die Daten geschrieben werden sollen. Jeder Datensatz verfügt darüber hinaus über bis zu 20 Zeichenfolge-Wertfelder. Bei der Methode SaveRecord werden ein Benutzertabellenname und ein -schlüssel erhalten. Darüber hinaus werden alle Wertfelder in der Benutzertabelle akzeptiert. (Diese Wertparameter sind optional und können weggelassen werden.) Wenn bereits ein Datensatz mit demselben Schlüssel vorhanden ist, wird er aktualisiert. (Es werden nur die als Parameter übertragenen Wertfelder aktualisiert.) Existiert kein Datensatz mit diesem Schlüssel, wird einer angelegt.
`Tools.SaveFields tableName, key, [fieldName1,fieldVal1], [fieldName2,fieldVal2]`

Diese Methode ist ähnlich wie die Methode SaveRecord, anstatt jedoch alle Werte zu nummerieren, werden hier Feldnamenpaare und diesbezügliche Feldwerte bereitgestellt. Die Feldnamen können durch Feldnummern ersetzt werden. Die Feldnamen sollten zuvor in der Tabelle T_SO_FIELD_NAMES manuell definiert werden. Diese Tabelle wird verwendet, um die Struktur der Ausgabetabellen aufzuzeichnen.

Es wird empfohlen, dass der Business-Logik-Experte die Struktur der der Ausgabetabelle definiert, bevor in T_SLALOM_OUTPUTS geschrieben wird, da die Struktur und die Bedeutung des Feldes in diesem Fall bereits ausreichend definiert sind, wodurch sich die Abfragen erheblich einfacher gestalten.

Die Tabellenstruktur ist:

- **table_name** - Jede logische Tabelle verfügt über einen eindeutigen Namen
- **field_name** - Jedes Feld in einer Tabelle ist eindeutig
- **field_id** - Jedes Feld ist mit einer seriell fortgeführten Nummer versehen, angefangen bei 1

Die Verwendung der Methode SaveFields ist vorzuziehen, da die Struktur und die Bedeutung des eingegebenen Wertes dokumentiert werden.

***Beispiel*:**

Angenommen, es ist erforderlich, eine Liste aller Vorfälle mit einer Problemlösungszeit über dem definierten Schwellenwert zu erstellen, und gleichzeitig muss das Metrik-Ergebnis den Prozentsatz an Tickets berechnen, die sich unterhalb dieses Schwellenwertes lösen ließen. Das Schreiben in Ausgabetabellen erfolgt in der Event-Handler-Prozedur OnXXXEvent nach der Schwellenwertvalidation.

Dies wird vom folgenden Beispiel verdeutlicht:
`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
```

Nachfolgend finden Sie einige auf die Verwendung der Tabellen T_SLALOM_OUTPUTS bezogene Vorschläge:

- Es wird empfohlen, um nur die notwendigen Daten in die Ausgabetabellen zu schreiben (nicht mehr).
- Schreiben Sie nur die erforderlichen Daten für die Metrikgranularitäten in die Ausgabetabellen.
- Alle Wertfelder können standardmäßig nur 50 Zeichen aufnehmen (außer der Spalte VAL_1, die 512 aufnehmen kann). Möglicherweise müssen Sie bestimmte Werte für die Felder abkürzen, um sicherzustellen, dass ihre Datengröße passt, da es ansonsten zu Laufzeitfehlern in der Business-Logik kommt.
- Dort sind nur 20 Spalten für Ihre Daten verfügbar (VAL_1 ... VAL_20)

Hinweis: Das Schreiben in die Ausgabetabellen kann Auswirkungen auf die Leistung der Engine haben, da das Schreiben in eine Tabelle im Vergleich mit einer Berechnung im Arbeitsspeicher rechenintensiv ist. Überlegen Sie sich daher genau, ob diese Funktionalität benötigt wird, und was dafür erforderlich ist. Minimieren Sie dann die Zugriffshäufigkeit auf die Tabellen.

Berichterstellung über Daten aus den benutzerspezifischen Tabellen

Zur Berichterstellung über Daten, die in die Ausgabetabellen geschrieben werden, kann der standardmäßige CA Business Service Insight-Berichtassistent nicht verwendet werden. Zur Berichterstellung über diese Daten muss zunächst ein Freiformbericht erstellt werden. Im Wesentlichen bedeutet dies die Generierung einer SQL-Abfrage auf der Grundlage dieser Tabelle. Da diese Tabelle viele logische Tabellen enthält, die von einer Vielzahl von Formeln in diese Tabellen geschrieben werden, wird empfohlen, dass jemand, der sich mit SQL auskennt (Datenexperte) eine Maske über T_SLALOM_OUTPUTS erstellt, um die Differenzierung zwischen den darin enthaltenen logischen Tabellen zu vereinfachen, und um sicherzustellen, dass die Daten plangemäß abgerufen werden.

Die Definition der Ansicht entspricht in ihrer Form bereits ganz den Zeichenfolge-Feldtypen, und sie umfasst auch bereits die gesamte erforderliche Filterung (logische Tabelle, aktive Datensätze, relevante Metrik usw.).

Nachfolgend finden Sie ein Beispiel für die Ansichterstellung:

```
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'
```

Erstellen von Business-Logik-Modulen

Die Benutzer können eigenständige Business-Logik-Module definieren, die von mehreren Service Level-Zielen (Metriken) genutzt werden können. Zur Anwendung systemweiter Änderungen auf die Business-Logik ändert der Benutzer die "Basis"-Logikkomponente, die dann per einfachem Mausklick auf alle relevanten SLAs angewendet werden kann.

Ein Business-Logik-Modul ist eine Code-Komponente zur Definition und übersichtlichen Verwaltung der Business-Logik unter weitgehender Reduzierung der Redundanz. Ein einzelnes Modul kann von mehreren Metriken verwendet werden.

In der Konfigurationsphase können die Formeln auf die Definition der Business-Logik-Hauptmodule definiert konfiguriert werden. (Sehen Sie im Kapitel Design: Business-Logik-Vorlagen und Module.)

Es gibt drei Typen von Business-Logik-Modulen:

- **Volle Business-Logik** - wird verwendet, wenn eine vollständige Formel als Modul verwendet werden muss. Das Ergebnis und die OnRegistration-Methoden müssen in das Modulskript implementiert werden.
- **Klasse** - Modul mit einer Definition von einer einzelnen VB-Klasse.
- **Bibliothek** - Modul mit einer beliebigen Zusammenfassung von Prozeduren, Funktionen und Klassen.

Module können aus einer der folgenden Optionen heraus verwendet werden:

- Metriken
- Andere Module
- Business-Logik-Vorlagen
- Metriken in Vertragsvorlagen und Service Level-Vorlagen.

Module können Parameter verwenden, die vom Metrik-Kontext gesteuert werden Parameters("ParamName").

Hinweis: Legen Sie zur Vermeidung von Laufzeitfehlern stets einen Standardwert in den Business-Logik-Modulen fest. Die Formel gibt eine Protokollfehlermeldung für nicht vorhandene Parameter aus.

```
If Parameters.IsExist("ReportedDowntimesNum") Then
    maxBufferSize = Parameters("ReportedDowntimesNum")
Else
    maxBufferSize = 3
    out.log "ReportedDowntimesNum parameter not set", "E"
End If
```

Beispiel für Business-Logik-Module

Es gibt ein als "Helpdeskaktivität innerhalb Schwellenwerts" bezeichnetes Service Level-Objekt. Das folgende Helpdesk-System hat einen Ticket-Lebenszyklus mit den folgenden Status:

- Offen
- Zugewiesen
- Geschlossen.

Zwei der Metriken, die zur Beschreibung der Leistung des Helpdesks definiert werden können, sind:

- **Metrik-Name** - **Erfolgreiche und pünktliche Problemlösung**.

  Zielvorgabe - Nicht weniger als 99 % der Anfragen sollten innerhalb von 4 Stunden gelöst sein.

  Business-Logik - Die Behebung sollte von Offen zu Geschlossen berechnet werden.

- **Metrik-Name** - **Erfolgreiche und pünktliche Ticket-Zuweisung**.

  Zielvorgabe - Nicht weniger als 99 % der Anfragen sollten innerhalb von 30 Minuten zugewiesen sein.

  Business-Logik - Die Zuweisung sollte von Offen zu Geschlossen berechnet werden.

Da dieselbe Logik für beide Metriken identifiziert werden kann, lässt sich ein Modul erstellen, das beiden Metriken gerecht wird.

Das Modul erfordert folgende Parameter im Zusammenhang mit der Metrik:

- Erster Status
- Zweiter Status
- "Threshold"

Sobald das Business-Logik-Modul erstellt wurde, kann die definierte Metrik ein Modul als Teil der Definition verwenden.

Im nächsten Schritt kann die Logik geändert werden. Beispiel: Ein neuer Status "Warten auf Kunden" soll berücksichtigt werden. "Warten auf Kunden" wird für eine Anfrage gesetzt, wenn der Helpdesk auf weitere Informationen bezüglich der Anfrage seitens des Kunden wartet. Innerhalb der Business-Logik sollte das Ticket für den Zeitraum, in dem es auf "Warten auf Kunden" gesetzt ist, nicht in die Berechnung einbezogen werden. Daher muss nur das Business-Logik-Modul entsprechend geändert werden, um den neuen Status und die neue Logik zu berücksichtigen. Es wird eine neue Modulversion mit der neuen Logik erstellt.

Wenn die Änderung übernommen wird, wird Ihnen eine Liste der Metriken angezeigt, die dieses Modul nutzen. Sie können die Änderung entweder kollektiv auf alle Metriken anwenden oder sich auf bestimmte Metriken auf der Liste beschränken.

Beschränken Sie sich bei der Auswahl auf bestimmte Metriken aus der Liste, werden Sie aufgefordert, ein neues Business-Logik-Modul für die ausgewählten Metriken zu erstellen. Das alte Modul, das bisher von den ausgewählten Metriken verwendet wurde, wird durch das neue Business-Logik-Modul ersetzt, und die Neuberechnung erfolgt mithilfe der neuen Logik.

Erstellen von Parametern

Parameter bieten Geschäftsbenutzern eine schnelle und intuitive Möglichkeit zur Darstellung und Änderung ihrer Werte über die grafische Benutzeroberfläche (GUI), ohne dass dazu das Formelskript bearbeitet werden muss.

Durch die Verwendung von Parametern innerhalb der Business-Logik wird die Erstellung allgemeiner Formeln, die systemübergreifend eine breite Anwendung finden, ermöglicht, und die Nutzung der Module wird optimiert.

Parameter können auf Vertrags- oder Metrik-Ebene definiert werden. Parameter der Metrik-Ebene werden auf der Registerkarte "Zielvorgabe" in den Metrik-Details angezeigt und konfiguriert. Die Business-Logik hat nur Zugriff auf die Parameter auf Metrik-Ebene; um also einen Vertragsparameter aus einer Metrik heraus aufzurufen, wird lokal innerhalb der Metrik ein anderer Parametertyp angelegt. Dieser wird als dynamischer Parameter bezeichnet. Dieser bezieht seinen Wert als Referenz aus den Parametern auf Vertragsebene. Im dynamischen Parameter sind als Referenzwerte nur solche zugelassen, die im übergeordneten Vertrag der Metrik definiert sind.

Parametertypen:

- Date-in Date Modal (Date + Time).
- Zahl - maximale Größe von 15 Ziffern mit einer maximalen Genauigkeit von 5 Ziffern.
- Text - maximale Größe von 100 Zeichen.
- Tabelle - maximale Größe von 120 Einträgen.

m auf die Parameterwerte des Formelcodes zugreifen zu können, muss das Parameterobjekt verwendet werden, und es muss sich auf den Parameternamen verwiesen werden.

Beispiel:

```
Parameter ("Threshold")
(Beachten Sie, dass es sich hierbei um eine Abkürzung zum Abrufen des Wertes
handelt - normalerweise erfolgt dies mit Parameters.Item("Threshold").)
```

Oder für einen Tabellentyp-Parameter:

Parameter ("Table")("Word")("Price")

(wobei die Werte "Word" und "Price" jeweils die in der tabellarisierten Zeilen- und Spaltennamen bezeichnen)

Tabellen-Parameter sollten nur nach einer Reihe von Hauptpunkten verwendet werden:

1. Definieren Sie eine globale Variable (z. B. dim myTableParam)
2. Geben Sie in der Funktion OnLoad die Variable des Parameterobjekts ein (z. B."Set myTableParam = Parameters("MyParametersTable"))

3. Verwenden Sie danach nur das erstellte Objekt myTableParam. Der Parameter selbst sollte nicht außerhalb der Funktion "OnLoad" verwendet werden, und Sie sollten nur auf das globale variable Objekt verweisen, das Sie anhand des Parameters erstellt haben.
   (Beispiel: "dim myVal: myVal = myTableParam ("myRow")("myColumn")).

Dies setzt Speicherplatz frei, den der Parameter zuvor belegt hat, und verhindert, dass die Engine die Zuweisung des Parameters für jeden Parameterabruf sowie für jeden "OnXXXEvent" (dieser kann tausende von Malen pro Metrik abgerufen werden) erstellt.

Der folgende Code verdeutlicht die richtige Anwendung eines Tabellenparameters:

```
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
```

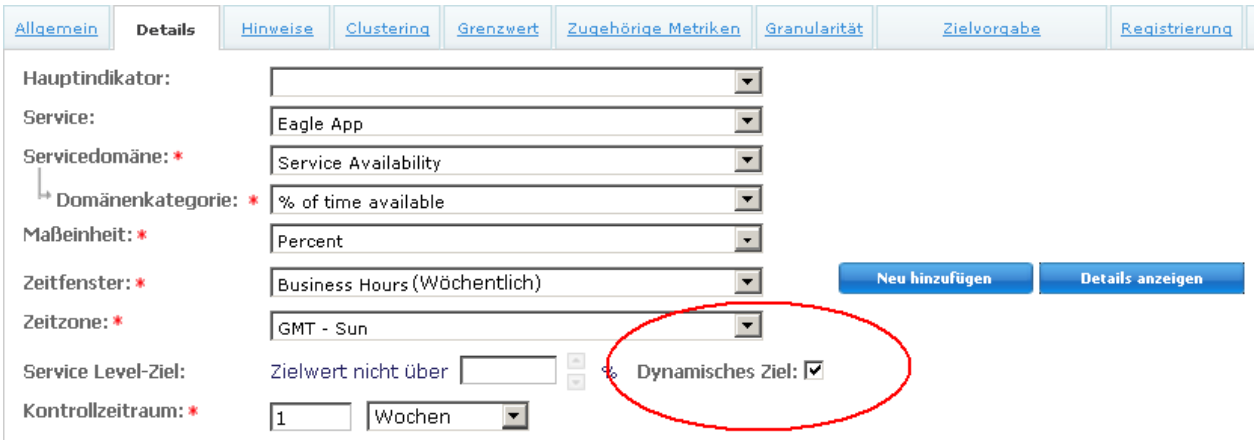
Die folgenden Methoden sind für jedes, innerhalb des Codes erstelltes Parameterobjekt verfügbar.

Parameter	Description
IsExist	Der Parameter ist vorhanden. Funktioniert nicht für Vertragsparameter.
IsNumeric	Ist ein Parameter vom Typ Zahl.
IsText	Ist ein Parameter vom Typ Text.

IsDate	Ist ein Parameter vom Typ Datum.
IsTable	Ist ein Parameter vom Typ Tabelle.
IsEntryExist	Ist ein vorhandener Eintrag in der Tabelle.
IsEntryNumeric	Ist ein Eintrag in einer Tabelle vom Typ Nummerisch.
IsEntryText	Ist ein Eintrag in einer Tabelle vom Typ Text.
IsEntryDate	Ist ein Eintrag in einer Tabelle vom Typ Datum.
Dump	Gibt eine Liste aller Parameter zurück.
Item	Verweist auf den Parameterwert.

Implementieren dynamischer Ziele

Dynamische Ziele werden von der Business-Logik mithilfe eines Event Handlers im Standard-Business-Logik-Skript verwaltet, ähnlich der Funktion "Result()", die zur Ausgabe des Service Level-Wertes von der Metrik verwendet wird. Das dynamische Ziel muss, wie nachfolgend dargestellt, auf der Metrik-Registerkarte "Details" festgelegt werden.



Wenn ein dynamisches Ziel festgelegt wurde, wird anstelle des statischen Wertes, der auf der Registerkarte "Details" der Metrik festgelegt ist, das Ziel aus der Funktion "Target()" in der Business-Logik genommen. Die Zielfunktion sieht folgendermaßen aus:

```
Function Target
  'TODO: ADD code here TO handle dynamic target calculation
  Target = Null
End Function
```

Zur Ausgabe des gewünschten Zielwerts für einen speziellen Zeitraum sollte diese Funktion gemäß den Anforderungen an die Metrik implementiert werden. Die Funktion kann eine beliebige Zahl zurückgeben, die ihr von der Business-Logik zugewiesen werden kann.

Ein reales Beispiel für dynamische Ziele

Für ein Call Centre ist das Ziel für eine Metrik, die die "*Durchschn. Anrufzeit*" misst, möglicherweise abhängig vom Anrufvolumen. Wenn zwischen 0 und 800 Anrufe eingehen, sollte das Ziel < 15 Sekunden liegen. Gehen zwischen 801 und 1500 Anrufe ein, sollte das Ziel bei < 20 Sekunden liegen. Bei allem über 1500 Anrufen sollte das Ziel unter 25 Sekunden liegen. Dies könnte folgendermaßen implementiert werden: (ausgehend davon, dass TotalCalls ein Zähler ist, der für jeden eingehenden Anruf-Event hochgezählt wird, und dass TotalCalls nicht weniger als 0 sein kann)

```
Function 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
```

Ein weiteres Beispiel dafür, wie ein dynamisches Ziel angewendet werden kann

Stellen Sie sich eine Situation vor, in der sich das Ziel für eine Metrik je nach Granularität der Berechnung ändern kann. Es kann ein Tagesziel einer Verfügbarkeit für eine Servergruppe von 98 % vorliegen, das monatliche Ziel ist jedoch eine Verfügbarkeit von 99,5 %. Für die diesbezügliche Lösung sind eine dynamische Zielfunktion sowie der Funktionsabruf nach *Kontext.TimeUnit* erforderlich, um den derzeit berechneten, aktuellen Agent zu ermitteln. Daher können Sie das Ziel entsprechend anpassen.

```
Function Target
  If Kontext.TimeUnit = "DAY" Then
     Target = 98
  ElseIf
     Target = 99.5
  End If
End Function
```

Sicherung von Status

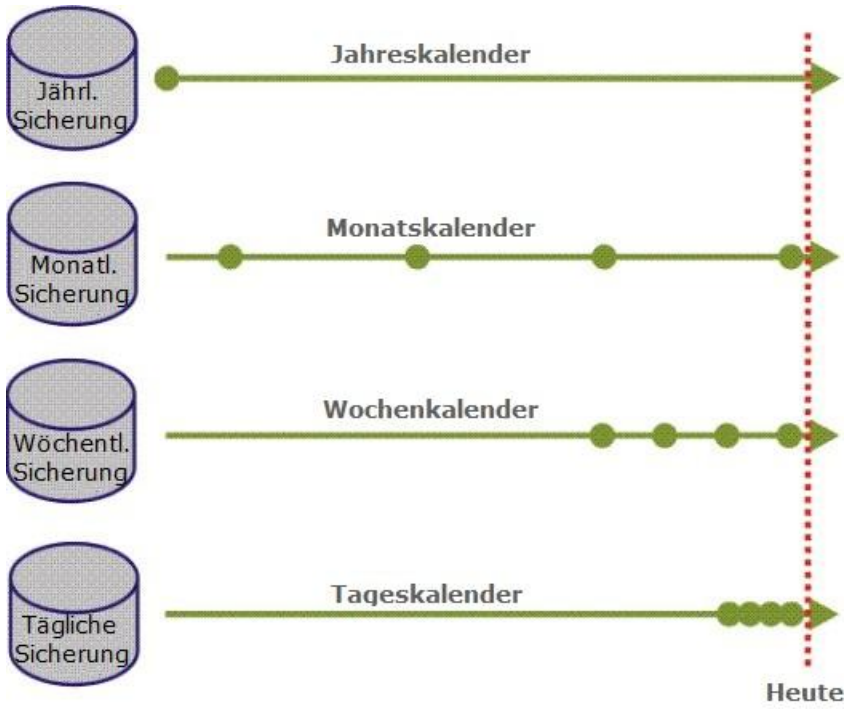
Während des kontinuierlichen Prozesses zur Berechnung der Service Level für jede der Metriken, ist die Engine oft gezwungen, eine partielle Berechnung für einen noch nicht abgeschlossenen Zeitraum durchzuführen. Um sicherzustellen, dass sie beim Eintreffen neuer Daten mit der Zeit nicht direkt an den Berechnungsanfang zurückspringen muss, führt sie eine Art von Sicherung ihres aktuellen "Status" durch, bevor sie mit ihrer nächsten Rechenaufgabe fortfährt. An diesem Punkt wird eine Momentaufnahme der aktuellen Variablen und Werte an diesem Berechnungspunkt gemacht, und dieser "Status" wird in der Datenbank gespeichert.

Der Business-Logik-Sicherungsvorgang ist ein Mechanismus, durch den der Business-Logikcode, einschließlich der Werte der Variablen, in einen binären Datenstrom verschlüsselt und in der Datenbank gespeichert wird. Dieser Mechanismus wird zudem benötigt, um die Berechnungs-Engines-Leistung bei Neuberechnungen zu beschleunigen. Der Status wird von Zeit zu Zeit gesichert, und er dient zur Neuberechnung und als Leistungsfähigkeitsmaß für die Fortsetzung der Berechnungen.

Beispiel: Wenn eine Neuberechnung rückwirkend für einen Zeitpunkt benötigt wird, der einen Monat zurückliegt, dann wird statt der Neuberechnung der Ergebnisse von Vertragsbeginn an der neueste gesicherte Status vor dem Neuberechnungsdatum verwendet, und die Berechnungen werden ab diesem Status durchgeführt.

Die Berechnungs-Engine verwendet eine vordefinierte Heuristik zur Feststellung, wann eine Sicherung erforderlich ist. Darüber hinaus nutzt sie die Sicherungsmöglichkeiten zum Speichern des verschlüsselten Status in der Datenbank.

In der nachfolgenden Abbildung stehen die roten Punkte für eine Statussicherung. Je weiter in der Vergangenheit die Berücksichtigung liegt, desto weniger gesicherte Status werden berücksichtigt. Die Logik hinter diesem Mechanismus geht davon aus, dass die Neuberechnung üblicherweise für einen Zeitraum benötigt wird, der weniger als einen Monat zurückliegt.


Jährl. Sicherung
Jahreskalender
Monatl. Sicherung
Monatskalender
Wöchentl. Sicherung
Wochenkalender
Tägliche Sicherung
Tageskalender
Heute

Optimieren des Systems zur Neuberechnung

Für die Service Level-Berechnung sind beträchtliche CPU-Ressourcen sowie viel Arbeitsspeicher und Speicherplatz erforderlich. Nachfolgend finden Sie eine Liste mit Empfehlungen, durch die sich die Systembelastung verringern und die Rechengeschwindigkeit optimieren lassen.

Hinweis: Für einige dieser Empfehlungen sind interne Einstellungen erforderlich, die über die Benutzeroberfläche nicht verfügbar sind. In solchen Fällen wenden Sie sich bitte für weitere Einzelheiten und Anweisungen an den technischen Kundendienst von CA.

- **Verändern der Konfiguration zum Speichen von Status**

  Je nach den bekannten Verzögerungen des Adapters können Sie die Statusparameter so konfigurieren, dass sie besser an Ihre Anforderungen angepasst sind. Dies bedeutet, Sie können die Anzahl und die Granularität der Speicherpunkte der Status ändern.

- **Konfigurieren Sie das System so, dass nur die wirklich benötigten Zeiteinheiten berechnet werden.**

  Metriken können bis zu sieben unterschiedliche Granularitätszeiträume aufweisen: *Jahr*, *Quartal*, *Monat*, *Woche*, *Tag*, *Stunde* und "Kontrollzeiträume". Bei einigen Implementierungen sind nicht alle Granularitäten erforderlich. Sie können unnötige Granularitäten für nicht übernommene Verträge über die Benutzeroberfläche deaktivieren. Verweisen Sie auf jede Metrik-Registerkarte "Granularitäten". Für das Ändern von Metrik-Granularitäten eines übernommenen Vertrages wenden Sie sich bitte an den technischen Kundendienst von CA.

  Hinweis: Vermeiden Sie Berechnungen von Betriebszeiträumen, die den Kontrollzeiträumen entsprechen.

- **Führen Sie keine Berechnungen der Werte "ohne Korrektur" und "ohne Ausnahmen" durch**

  Standardmäßig berechnet die Berechnungs-Engine vier verschiedene Werte: *Bereitgestellt*, *Mit Korrekturen bereitgestellt*, *Mit Ausnahmen bereitgestellt* und *mit Korrekturen und Ausnahmen bereitgestellt*. Sie können diese Einstellung ändern, um nur den Wert "Bereitgestellt" zu berechnen.

  Hinweis: Weitere Informationen erhalten Sie beim Technischen Kundendienst von CA.

- **Ändern Sie die Berechnungsreihenfolge**

  Die PSL-Berechnungsreihenfolge von Metriken kann so priorisiert werden, dass sie mit Ihren wichtigeren Metriken startet und dann mit den anderen Metriken fortfährt.

  Hinweis: Weitere Informationen erhalten Sie beim Technischen Kundendienst von CA.

- **Erstellen Sie mehr als eine PslWriter-Instanz**

Durch das Erstellen mehrerer PSL-Instanzen (der Berechnungs-Engine) können Sie die Arbeitslast aufteilen und die Berechnungsgeschwindigkeit steigern.

Hinweis: Weitere Informationen erhalten Sie beim Technischen Kundendienst von CA.

- **Planen Sie Implementierungen, um die Neuberechnungszeit zu verkürzen**

  Zur Optimierung der Neuberechnungszeit können Sie folgendermaßen vorgehen:

  - Führen Sie häufiger den Adapter aus, um verzögerte Events zu verringern und große Arbeitsrückstände von Daten zu vermeiden, die die Engine noch verarbeiten muss.
  - Deaktivieren Sie unbenutzte Agenten auf der Registerkarte "Granularität" der Metrik.
  - Kopieren Sie die Metrik und berechnen Sie unterschiedliche Agents mit derselben Metrik, um die Berechnungslast abzugleichen
  - Verwenden Sie intermediäre Metriken zur Durchführung allgemeiner Berechnungen, und nutzen Sie die Ergebnisse mit allen anderen Metriken, die dieselben Daten benötigen.

- **Planen Sie Implementierungen, um die Datenmenge zu reduzieren**

  Verwenden Sie zur Optimierung der Datenmenge den Adapter, um nur aggregierte (verarbeitete) Daten zu laden. Durch die Aggregierung von Datenquellen-Daten, bevor diese an die CA Business Service Insight-Rohdatentabelle gesendet werden, wird die Effizienz beim Lesen der PLS-Eingabe erhöht.

- **Folgen Sie den CA Business Service Insight-PSL-Konfigurationsempfehlungen**

  Die PSL-Engine kann für eine bessere Leistung gemäß der speziellen Anwendungsumgebung und den speziellen Anforderungen neu konfiguriert werden. Einige Parameter können von der Benutzeroberfläche aus festgelegt werden, andere nur über die Konfigurationstabelle des Datenbanksystems.

  Hinweis: Lesen Sie die Empfehlungen des Kundendienstes zu Ihren PSL-Konfigurationen.

Protokolle und Alarme

Es gibt Fälle, wo die Business-Logik Meldungen an das Protokoll senden oder eine Alarmmeldung auslösen muss. Des ist notwendig, wenn die Meldungen basierend auf der Event-Verarbeitung versendet werden sollen. Alle Informationen, die im Verlauf der Berechnung erfasst werden, und die sich als wertvoll erweisen können, können als Alarmmeldung versendet werden. Beispiel: Eine Alarmmeldung kann versendet werden, wenn sich ein spezielles Event unterhalb des festgelegten Auflösungszeit-Schwellenwertes befindet. Oder in der Trendanalyse: Wenn eine bestimmte Häufigkeit nacheinander auftretender Fehlfunktionen erreicht ist.

"Out" ist ein globales Business-Logik-Objekt, mit dem die Formel Alarm- und Protokollmeldungen versenden kann. Ihm zugewiesen sind zwei Methoden, die folgende Form haben:

```
Alert(<Event type>, <Resource name>, <value1, value2>, …<value16>)
```

Dieser Befehl sendet den Alarm eines festgelegten Event-Typs. Allerdings muss dieser Event-Typ für diesen Alarm manuell erstellt werden. Die Anzahl von Werten und deren Typ müssen der Definition des Event-Typs entsprechen.

```
Log(<Message>,<Level>)
```

Dieser Befehl sendet eine Meldung an das Systemprotokoll. Der erste Parameter ist die dokumentierte Informationsmeldung, bei der es sich um Freitext handeln kann. Sie können dieser Zeichenfolge auch die Werte von Variablen anhängen, um der Meldung einen Kontext zu verleihen. Der Parameter "Level" kann einen der folgenden Werte annehmen:

	Beschreibung
	Eine Warnmeldung wird gemeldet.
	Eine Fehlermeldung wird gemeldet.
	Eine Informationsmeldung wird nur bei Betrieb im Business-Logik-Umfang dokumentiert. Bei der Ausführung in PslWriter wird keine Meldung dokumentiert. Dies ist die Standardeinstellung. Sie wird zumeist für zur Fehlersuche verwendet.

***Beispiel*:**

Das folgende Beispiel stammt von einem Fall, bei dem die Informationen zur Infrastruktur des Events vor den tatsächlichen Vorfalldaten erwartet wurden. Ein Alarmmechanismus wurde eingerichtet, der den Administrator über diese Bedingung informieren sollte, damit jener das Problem unverzüglich beheben kann.

```
Out.Alert "Site Unknown Alert", Kontext.ClusterItem, Kontext.Rule
```

```
Out.Log("Fault Event Received for a Site with no infrastructure details: " &
Kontext.ClusterItem)
```

Kommentare zur Ursache bei Regelbruch und Event-Kommentare

Der Kommentar zur Ursache bei Regelbruch kann in der Business-Logik zur Erklärung der Service Level-Ergebnisse festgelegt werden. Die Kommentare zur Ursache bei Regelbruch sind Metriken zugewiesen.

Es können auch Event-Anmerkungen erstellt werden. Dabei handelt es sich um Anmerkungen, die Rohdatenereignisse - und nicht der Metrik - zugewiesen werden. Diese beiden Kommentartypen lassen sich über die Berichtdaten anzeigen.

Zwei Methoden im Business-Logik-Objekt "Tools" ermöglichen die Erstellung eines Kommentars zur Ursache bei Regelbruch sowie von Event-Anmerkungsdatensätzen:

- **Tools.AddRootCauseComment** (Text, Zeitstempel [resourceId])
- Speichert einen Kommentar zur Ursache bei Regelbruch. Diese Informationen können später in generierten Berichten nützlich sein. Der gespeicherte Kommentar zur Ursache bei Regelbruch erläutert eine spezielle Situation bei Ausführung der Business-Logik-Formel in einem speziellen Augenblick. Der Informationsparameter, der den Kommentar festlegt, sollte festgeschrieben werden. Bei dem Verfahren wird ein Zeitstempel empfangen, der zusammen mit dem Kommentar zu speichern ist. Darüber hinaus wird der Parameter ResourceId akzeptiert, der die auf den Methodenkontext bezogene Ressource festlegt. (Dieser Parameter ist optional und kann weggelassen werden.)
- **Tools.AddEventAnnotation** (EventId, Text)
- Diese Methoden können überall innerhalb der Business-Logik verwendet werden, und es muss der Kontext dessen, wo sie angewendet werden, berücksichtigt werden. Das Hinzufügen eines Kommentars zur Ursache bei Regelbruch kann am Ende eines Berechnungszeitraums von größerer Bedeutung sein, wenn der Grund dafür bekannt ist, warum der Service Level hinter dem für diesen Zeitraum erwarteten Service Level zurückbleibt. Angenommen, es hätte beispielsweise während des Zeitraums von einem Monat vier Ausfälle gegeben, die alle von einem einzigen Gerät verursacht wurden. Der Kommentar zur Ursache bei Regelbruch könnte dann aus einigen der Informationen zu den Ausfällen kompiliert werden, sodass bei Darstellung der Berichte für diesen Zeitraum ersichtlich ist, was zu einer möglichen Service Level-Vertragsverletzung zu dieser Zeit beigetragen hat. Der Befehl AddRootCauseComment ist für die OnPeriodEnd-Event-Handler-Routine oder eine ähnliche Funktion, die am Ende des zu berechnenden Zeitraums ausgeführt wird, am besten geeignet.
- Das Hinzufügen einer Anmerkung zum Event eignet sich jedoch besser für die tatsächliche Verarbeitung von Rohdatenereignissen, dabei wird die Nutzung des OnXXXEvent (des benutzerspezifischen Event Handlers, der in der Registrierungserklärung festgelegt ist), bevorzugt. In diesem Event Handler sind alle Felder für das tatsächliche Event über das Objekt eventDetails verfügbar.
- Nachfolgend sehen Sie ein Beispiel dafür, wie dies innerhalb der Business-Logik aussehen könnte:

```
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
```

Trennen der Ausnahmen von den Zeitfenstern

Die CA Business Service Insight-Business-Logik erhält die Ausnahme-Events nicht. Empfangen werden hingegen der Parameter OnTimeslotExit, wenn ein Ausnahmezeitraum beginnt, sowie der Parameter OnTimeslotEnter, wenn ein Ausnahmezeitraum endet. Die Business-Logik kann deswegen nicht zwischen Ausnahmezeiten und außerhalb des Zeitfensters liegenden Zeiten unterscheiden. Überdies kann sie nicht zwischen Ausnahmetypen unterscheiden. Dadurch kann keine differenzierte Logik für das Ausnahmezeitverhalten sowie für das Verhalten "außerhalb des Zeitfensters" implementieren.

Eine Möglichkeit zur Implementierung besonderer Ausnahmen (d. h. einer Ausnahme, die sich nicht als Zeitraum "außerhalb des Zeitfensters" verhält, ist die Definition von dedizierten Event-Typen statt der Verwendung des in CA Business Service Insight integrierten Mechanismus für den Umgang mit Ausnahmen. Diese Ereignisse werden generiert, indem sie mithilfe eines Adapters von einer dedizierten Datenquelle ausliest.

Diese Ausnahmen können in einer Excel-Tabelle (oder in jeder anderen Datenquelle) gespeichert werden, und ein Adapter kann die Daten dann laden und eine Antwort generieren: Ausnahme-Eingabe- und Ausnahme-Ausgabe-Events. Alternativ können die Ausnahmen durch die Verwendung von Korrekturen hinzugefügt werden. Zusätzlich zur Korrektur sollte eine Dummy-Ressource definiert und diesen Events zu Registrierungszwecken zugewiesen werden. Diese Ressource dient nur als Platzhalter, wie vom Befehl angefordert.

Um die von diesen dedizierten Ereignissen gemeldeten Ausnahmezeiten verarbeiten zu können, sollte die Business-Logik-Formel mit diesen Ausnahme-Ereignissen zusätzlich zu der normalerweise erforderlichen Registrierung für die in der Berechnung verwendeten Rohdatenereignisse, registriert werden.

Es wird empfohlen, dass der Business-Logik-Experte ein Feld für den Ausnahmetyp im Event-Typ enthält, um so eine differenzierte Handhabung der verschiedenen Arten von speziellen Ausnahmen zu ermöglichen.

Diese Vorgehensweise weist die folgenden Eigenschaften auf:

- Sie trennt zwei Ausnahmesätze - den Standard-Ausnahmesatz und den speziellen Ausnahmesatz.
- Spezielle Ausnahmen verfügen nicht über eine Benutzeroberfläche zu Wartungszwecken.
- Berichte, die auf den als Events von einem Adapter erstellten Ausnahmen basieren, kommentieren deren Vorhandensein nicht. In Fällen, in denen der Korrekturmechanismus verwendet wurde, liegt ein Kommentar vor. Die Verwendung der Korrekturmethode wird zur Wahrung der Integrität der vom System generierten Ergebnisse empfohlen.
- Keine Spezifikation "Mit/Ohne Ausnahmen" berücksichtigt diese Ausnahmen.
- Wo der Korrekturmechanismus verwendet wird, gilt die "Mit/Ohne"-Korrektur.

Nach der Implementierung wird empfohlen, den Business-Logik-Experten die Logik auf alle Systemmetriken anwenden zu lassen.

Es gibt noch eine weitere Methode, eine Ausnahme nach Bedarf auf eine einzelne Ressource anzuwenden. Bei dieser Methode wird der Ressourcenstatus "Wirksam" verwendet. Indem der Status einer Ressource auf "Unwirksam" festgelegt wird, ignoriert die Berechnungs-Engine während dieses Zeitraums alle Rohdatenereignisse, die für diese Ressource versendet werden. Durch das Festlegen eines Zeitraums, in dem die Ressource unwirksam ist, durch die Erstellung neuer Ressourcenversionen - eine zu Beginn des Ausnahmezeitraums, eine andere am Ende des Ausnahmezeitraums.

Wenn jedoch die Ressource Teil einer geclusterten Metrik ist, und die Ressource in demselben Berechnungszeitraum wirksam bzw. nicht wirksam wird, wird, wie oben aufgeführt, nur der letzte Zeitraum berücksichtigt, in dem die Ressource wirksam war. In diesem Fall wird empfohlen, die benutzerspezifische Attributfunktion zu verwenden. Es lässt sich ein zusätzliches Attribut für die Ressource verwalten, das den Status der Ressource angibt, und die Business-Logik-Formel wird an jeder relevanten Stelle im Skript für den Status der Ressource abgefragt.

Überlegungen zum Arbeitsspeicher und zur Leistung

Nachfolgend finden Sie eine Reihe von Situationen, die bei der Entwicklung von Business-Logik-Lösungen berücksichtigt werden sollten. Die beschriebenen Situationen sind alles Fälle, in denen die Leistung der Berechnungs-Engine negativ beeinflusst werden kann:

- **Parameter**

  Wenn ein Parameterwert im Code benötigt wird, wird die Erstellung einer globalen Variablen empfohlen, der der Parameterwert dann zugewiesen wird. Verwenden Sie ferner stattdessen die globale Variable, wenn der Wert des Parameter benötigt wird. Dies verhindert eine Situation, in der die Engine die Parameterzuordnungsgrafik für jeden Parameterabruf erstellt.

- **Verwenden von Zuordnungen in geclusterten Metriken**

  Große globale Zuordnungsobjekte innerhalb der Business-Logik von geclusterten Metriken sollten nur nach sorgfältiger Überlegung verwendet werden. Während die Engine eine geclusterte Metrik berechnet, ist sie mit dem separaten Laden der globalen Variablen der vorherigen Status für jedes Element im Cluster beschäftigt.

- **Business-Logik-Registrierung**

  Es wird empfohlen, die Rohdatenereignisse ausschließlich gemäß den Registrierungsmethoden zu filtern. Durch das Hinzufügen einer internen Filterung mit einer "if"-Erklärung erhöht sich die Verarbeitungszeit. Wichtiger noch: Von der Engine wird zusätzliche Overhead-Zeit für das Abrufen und Verarbeiten von nicht benötigten Rohdaten-Dateneinträgen benötigt.

- **Vermeiden der Verwendung von Dispatcher.RegisterByEventType**

  Verbessert die Leistung. Die Verwendung dieser Registrierungsmethode bedeutet, dass Sie sich für alle Ressourcen im System registrieren lassen, und nicht nur für Ressourcen mit Events dieses speziellen Typs. Damit wirkt sich jede Änderung in der Ressource auf die Metrikberechnungen aus. Ein anderer Nachteil bei der Verwendung dieser Registrierungsmethode entsteht in Bezug auf die Metrik-Laufzeit, wenn diese auf die Rohdaten zugreift. Sie muss dann aus den Rohdaten nur die Events mit dem speziellen Event-Typ herausfiltern; die anderen Ereignisse muss sie ignorieren.

- **Dispatcher.Register**

  Wenn Sie Dispatcher.Register verwenden, prüfen Sie *stets*, dass Sie den 3. Parameter festgelegt haben. Eine Registrierung ohne den 3. Parameter entspricht einer Registrierung nach Event-Typ (Dispatcher.RegisterByEventType). Mit anderen Worten: "Stellen Sie sicher, dass Sie neben den ersten beiden Parametern wenigstens einen weiteren verwenden."

- **Berechnungsgranularität**

  Es ist wichtig, dass nur die für die Berechnung und die Verfeinerung erforderlichen Agenten aktiviert werden. Die Berechnung aller Zeiteinheiten des Agenten nimmt sehr viel Prozessorleistung in Anspruch.

Aktivieren der Verträge (Vertragsmanager)

Ein Vertrag wird durch Übernehmen aktiviert. (Weitere Details finden Sie in der Onlinehilfe.)

Durch die Aktivierung des Vertrags wird die Engine aktiviert, die die Berechnungen für die Vertragsmetrik einleitet und beginnt, Ergebnisse für den Vertrag zu erzeugen.

Überprüfen Sie vor der Vertragsaktivierung, ob alle nachfolgenden Bedingungen erfüllt sind:

- Alle Metriken müssen die für sie definierte Business-Logik aufweisen (in der Metrik oder als verknüpftes Modul), die entsprechend getestet wurde und fehlerfrei ist
- Alle Metriken haben einen definierten Dashboard-Schwellenwert. (Weitere Einzelheiten dazu finden Sie in der Onlinehilfe.) Es ist wichtig, dass die Schwellenwerte schon definiert sind, sodass das Dashboard die Grenzen für die Metrik bereits während der Berechnung auswerten kann.
- Die Datumsangaben zum Inkrafttreten entsprechen dem richtigen Zeitraum, und es müssen Rohdaten-Datensätze verfügbar sein. Überprüfen Sie ferner anhand des Business-Logik-Umfangs, dass die Ergebnisse erwartungsgemäß ausfallen.

Überprüfen Sie, sobald der Vertrag übernommen wurde, ob ordnungsgemäß mit der Aktivierung begonnen wurde und ob die Berechnungen erwartungsgemäß vorgenommen werden.

Befolgen Sie diese Schritte:

1. Stellen Sie sicher, dass alle CA Business Service Insight-Servicekomponenten gestartet wurden (besonders die Berechnungs-Engine, die sowohl PSLWriter als auch PenaltyWriter umfasst). Es wird empfohlen, stets alle Servicekomponenten auszuführen, wenn eine Berechnung erforderlich ist.
2. Vertrag diagnostizieren - die Vertragsdiagnose-Funktion zeigt die Ergebnisse einer Reihe von Tests an, die an allen Vertragsmetriken vorgenommen wurden (sowie die entsprechenden Vertragsstrafenformeln, sofern diese angewendet wurden). Der Schweregrad des Prüfergebnisses wird zusammen mit einem Behebungsvorschlag angegeben. Es wird empfohlen, bei jeder Aktivierung eines Vertrages sowie nach Abschluss der Berechnung für diesen Vertrag eine Diagnose durchzuführen.
3. Generieren Sie den "Berechnungsstatus"-Freiformbericht. Dieser Bericht wird als Teil der CA Business Service Insight-Anfangsinstallation gebündelt und im Paketordner für Admin-Berichte gespeichert. Er liefert Informationen zum Berechnungsfortschritt und lässt sich an diesem Punkt nutzen, um zu verifizieren, ob die Verarbeitung durch die PSL-Engine erfolgt und ob die Berechnung abgeschlossen wurde. Prüfen Sie diesen Bericht, um zu bewerten, ob potenzielle Probleme in den Berechnungen vorliegen.

Der Bericht enthält die folgenden Spaltenfelder:

Feld	Beschreibung
Vertrag	Enthält den Namen des Vertrags. Die Liste enthält die Namen der wirksamen und der unwirksamen Verträge.
Metrik	Enthält den Namen der Metrik innerhalb des Vertrags. Die Liste enthält alle im jeweiligen Vertrag enthaltene Metriken.
Kontrollzeitraum	Enthält den Metrik-Berechnungszeitraum. Die Liste enthält gemäß den aktiven Agenten sowie basierend auf der Granularitätsdefinition der Metrik einen Eintrag für jede Berechnungszeit der Metrik. In Fällen, in denen der Berechnungszeitraum gleich dem Kontrollzeitraum ist, wird dies nicht dokumentiert.
Aktualisiert bis	Enthält die Aktualisierungszeit für das letzte Ergebnis. Dies zeigt an, dass das Ergebnis dieser speziellen Metrik bis zu diesem Datum verfügbar ist. Beispiel: Wenn 01.01.06 angezeigt ist, bedeutet dies, dass alle Ergebnisse für diese Metrik in dieser Zeiteinheit bis zu diesem Datum aktualisiert sind, und dass die Berichte dafür bis zu diesem Datum zur Verfügung stehen.
Letzter Zyklusbeginn	Enthält die Zykluszeit, zu der die Berechnung, bei der das Ergebnis der Metrik aktualisiert wurde, begann.
Erfordert Neuberechnung von	In Fällen, in denen die Engine eine bestimmte Metrik zur Neuberechnung markiert, diese jedoch noch immer nicht aktualisiert wurde, wird das resultierende Datum mit dem Zeitpunkt angezeigt, für den die Ergebnisse nicht mehr relevant sind und daher aktualisiert werden müssen. Dies kann in Fällen von Neuberechnungen auftreten.
Letzte Aktualisierung	Die Zeit, zu der die Engine den Datensatz mit dem letzten Ergebnis aktualisiert hat.
Verarbeitet von Engine	Enthält die Nummer der Engine, die zugewiesen wird, um die Berechnung der betreffenden Metrik zu verarbeiten.

Dieser Bericht kann auch die folgende, auf die verfügbaren Rohdaten basierte, Information angeben:

- Die Zeit, welche die Engine dafür benötigte, eine einzelne Metrik zu berechnen. Durch Sortieren aller Metriken, die in einem einzelnen Zyklus durch deren aktualisierter Zeit berechnet wurden, ist es möglich, sich anzeigen zu lassen wie lange die Engine benötigte, um jede Metrik zu aktualisieren. Alle Datensätze mit dem gleichen Wert "Letzter Zyklusbeginn" werden während des gleichen Zyklus berechnet und die Aktualisierungszeit ist die Zeit "Letzte Aktualisierung". Es ist möglich die Zeit auszuwerten, die die Engine benötigte, um die ganze Metrik mit den ihr zugrunde liegenden Agenten-Zeiteinheiten sowie jede der Zeiteinheiten zu berechnen.
- Die Zeit, die die Engine benötigte, um einen vollständigen Vertrag zu berechnen. Dies wird durchgeführt, indem die Aktualisierungszeit der ersten Metrik des Vertrags und die letzte Aktualisierungszeit der letzten Metrik dieses Vertrags geprüft, und die Zeit dazwischen berechnet wird.

Volle Neuberechnung von Vertragsmetriken

Der Prozess der Neuberechnung im aktuellen Zusammenhang ist die Einleitung einer vollständigen Systemneuberechnung, entweder vom Datenexperten oder Business-Logik-Experte. Es ist nicht die während eines normalen Berechnungsprozesses ausgeführte Engine-Neuberechnung. Diese Art von Aktion wird üblicherweise während oder nach dem Prozess der Vertrags-Konfiguration ausgeführt, wenn verschiedene Störungen identifiziert sein können.

Es wird empfohlen, dass mit einer vollständigen Neuberechnung erst angefangen wird, sobald das System in einem stabilen Zustand ist (das heißt, nicht während der Einrichtung vom System) bevor es auf der Seite aktiviert wird.

Die Neuberechnung wird gegenwärtig durch die Ausführung eines SQL-Skripts auf der Datenbank ausgeführt. Dieses Skript leert die PSL-Tabelle und alle mit ihr zugeordneten begleitenden Tabellen, die ein Teil des Berechnungsprozesses sind. Dieses Skript sollte, bevor es ausgeführt wird, genehmigt und vom CA Support-Team bestätigt werden, da gelegentliche Strukturänderungen im System Änderungen im Datenbankschema und/oder den davon abhängigen Objekten verursacht haben könnten.

Hinweis: Bevor Sie das Skript ausführen, müssen alle Servicekomponenten gestoppt werden und können nach der Ausführung neu gestartet werden, um die Berechnungen neu zu starten.

Die folgenden Situationen können die Notwendigkeit einer kompletten Neuberechnung bedeuten:

- Probleme mit den Definitionen im Vertrag - falls zu diesem Zeitpunkt festgestellt wird, dass ein Metrikziel falsch angegeben wurde, oder ein Metrik-Schwellenwert falsch ist.
- Fehler im Vertrag.
- Eine Anforderung zur Neubewertung der Dashboard-Schwellenwerte oder SLA-Alarme zu regenerieren.

Hinweis: Kontaktieren Sie den CA Support, um diese Option zu erörtern, falls erforderlich, und auch um Kopien der dazugehörigen Skripte für die Version von CA Business Service Insight zu erhalten, die installiert ist.

Erstellen von lieferbaren Ergebnissen (Vertragsmanager)

Zu diesem Zeitpunkt werden alle Ausgaben vom System so konfiguriert, dass sie den Anforderungen, einschließlich der Erstellung und Formatierung aller Berichte, entsprechen.

Die Konfiguration der lieferbaren Ergebnisse schließt das Folgende ein:

- Die Sicherheitseinstellungen definieren (Benutzerberechtigungen und Gruppen).
- Gespeicherte Berichte erstellen.
- Booklets erstellen.
- Vertrags-Exportvorlagen erstellen.
- Ansichten von Service Delivery Navigator (SDN) erstellen
- Dashboard-Zuordnungen konfigurieren
- Service Level-Alarmprofile erstellen.

Definieren von Sicherheitseinstellungen (Administrator)

Bei der Definition der Sicherheitseinstellungen, müssen die CA Benutzer und ihre zugeordneten Berechtigungen konfiguriert werden. Diese Einstellungen umfassen die Festlegung, welche Information einem Benutzer zugänglich sein sollte (welche Elemente innerhalb des Systems die Benutzer anzeigen oder bearbeiten können). Diese Berechtigungen können auf verschiedenen Ebenen von Benutzergruppen, Rollen oder sogar individuell pro Benutzer festgelegt werden. Zugreifbare Informationen werden in Beziehung zu Vertragsparteien angegeben und können direkt an den Benutzer oder aus der Benutzergruppe vererbt werden, zu der der Benutzer gehört.

Zu diesem Zeitpunkt wurden die Hauptrollen konfiguriert und die Benutzergruppen mit ihnen verbunden, sodass sie, wenn ein neuer Benutzer hinzugefügt wird, nur an einer Gruppe angefügt werden müssen, um die dazugehörigen Einstellungen zu erben.

Erlaubte Aktionen werden in den Rollen konfiguriert und werden an den Benutzer durch direktes Verknüpfen mit einer Rolle ausgegeben, oder von der Benutzergruppe ausgegeben, zu der der Benutzer gehört. Zulässige Dashboard-bezogene Aktionen sind auch in der anhängenden Rolle angegeben.

Es wird empfohlen, dass der Administrator entscheidet, welche Benutzergruppen und Rollen definiert werden sollten und ihre erforderliche Berechtigung, um in der Lage zu sein, eine Struktur festzulegen, die das einfache Hinzufügen von Benutzern ermöglicht.

Erstellen von Berichten

Verwenden Sie den folgenden Prozess, um einen Bericht zu erstellen:

1. Erstellen Sie alle für den Bericht erforderlich Ordner - alle Ordner sollten im Voraus eingerichtet werden, um beim Speichern jedes neuen gespeicherten Berichtes verfügbar zu sein. Üblicherweise hat jeder Vertrag seinen eigenen Ordner, einschließlich eines Management-Ordners für die High-Level-Berichte.
2. Definieren Sie die Kriterien für die Berichtsfilterung mithilfe des Berichtassistenten - jede gespeicherte Berichterstellung beginnt mit der Generierung des Berichtes durch den Berichtassistenten. Im Berichtassistenten, werden die erforderlichen Filterkriterien ausgewählt und der Bericht wurde generiert. IT-Administratoren können für die Berichtsparameter benutzerdefinierbare Felder angeben, die durch den Berichtsbenutzer/-betrachter ausgefüllt werden können, wodurch das Generieren von bestimmten Berichten möglich ist, die für diesen Benutzer von Interesse sind.

   Beim Generieren eines Berichtes für diese Zwecke, der Einstellung als ein gespeicherter Bericht, ist es sehr wichtig, dass die Filterkriterien so flexibel wie möglich definiert werden. Dies verhindert unnötige Arbeit, während sich das System entwickelt und größer wird. Das Ziel sollte sein, sicherzustellen, dass jeder Bericht aktuelle und aktualisierte Informationen anzeigt. Wenn ein Bericht beispielsweise gegenwärtig drei Servicekomponenten zeigt, ist es zukünftig beim Hinzufügen neuer Servicekomponenten wichtig, dass dieser Service automatisch dem Bericht hinzugefügt wird und keine neue Berichtserstellung erfordert. Oder wenn die Berichterstellung nach Monat erfolgt und der Bericht gegenwärtig drei Monate zeigt, dann sollte er im nächsten Monat vier Monate, einschließlich des letzten Monats, anzeigen. In vielen Fällen sollten die Berichte immer die letzten 6 Monatswerte der Daten zeigen. Diese Arten von "Schiebefenster"-Berichte sind äußerst nützlich im Gegensatz zu Berichten mit einer festen Dauer, da sie keine zusätzliche Aufmerksamkeit benötigen, sobald sie einmal erstellt wurden.

Im Folgenden finden Sie ein paar Tipps für das Festlegen der Filter-Kriterien für gespeicherte Berichte:

Registerkarte Kriterien

- Die Verwendung der Option "Nur die Geschäftsdaten" beschränkt die angezeigten Daten nur auf das, was auf die Kontrollzeiträume der Metrik bezogen wird.
  - Geben Sie immer der *group by option* oder der *report by option* den Vorzug, anstelle der Auswahl einer bestimmten Metrik über die berichtet werden soll.
  - Legen Sie einen dynamischen Zeitbereich fest. Wenn dies als ein fester Zeitbereich festgelegt wird, zeigt der Bericht immer die gleichen Ergebnisse an. (d. h. die letzten 3 Monate ist ein Dynamikbereich)

Registerkarte Sonstiges

- Überprüfen Sie, ob die "unvollständigen Zeiträume" im Bericht dargestellt werden sollten. Wenn dem so ist, dann wählen Sie diese Option aus der Liste aus. Üblicherweise, wenn Berichte konfiguriert werden, wird der unvollständige Zeitraum ausgeschlossen, weil er kein Endergebnis und deswegen kein Businessergebnis ist.
- Erstellen Sie das Berichtslayout - sobald der Bericht generiert wurde, ist es möglich, sein Layout durch Verwendung des Design Tool auf der Report Page zu erstellen. Das Design kann bestimmte Anforderungen haben, je nach zukünftigem Empfänger des Berichts. Es wird empfohlen, eine Reihe von Entwürfen zu erstellen, einen für jedes mögliche Szenario, und diese Designs auf den Rest der zu generierenden Berichte anzuwenden. Dafür wählen Sie bei der Definition der Kriterien vom Bericht in der Registerkarte *Anzeige*, *Design from*.

  Hinweis: Für Tipps wie mit dem Editiertool das Design des Layouts der Berichte erstellt werden kann, siehe nächster Abschnitt.

Registerkarte "Allgemein"

- Speichern des Berichts - Sobald der Bericht generiert und gestaltet wurde, kann er dann im dazugehörigen Ordner gespeichert werden.
- Während des Prozesses den Bericht zu speichern, verbindet er sich mit Benutzern, die Zugriff auf ihn haben. Deshalb ist es wichtig eine Benutzergruppe schon angegeben zu haben, sodass es möglich ist, Berichte mit Benutzern zu verbinden. Benutzer mit den richtigen Berechtigungen können zu einem späteren Zeitpunkt am Bericht durch die Verwendung der Ordneroptionen angefügt werden.
- Verbinden Sie die zugehörigen Berichte, sodass die Navigation zwischen Berichten die ähnlich sind, oder eine Beziehung zueinander haben, für die Benutzer jener Berichte vereinfacht wird.
- Auf der Seite "Berichtsordner" kann man Berichte, Berichtsgruppen, Kombiberichte, Freiformberichte, Booklets, Verknüpfungen und Ordner erstellen und danach suchen.

  Klicken Sie im Menü "Berichte" auf "Berichtsordner". Die Seite "Berichtsordner" wird angezeigt und zeigt eine Liste von gespeicherten Berichten an.

Abweichungsbericht

Der Abweichungswert wird automatisch von der CA Business Service Insight-Engine für Metriken, die ein Ziel haben, berechnet. Der Wert stellt den Unterschied zwischen dem eigentlichen Service Level und dem Ziel dar. Die Abweichungsberechnungsformel, automatisch von CA berechnet, ändert sich entsprechend der Service Level-Definition: ob das Service Level-Ziel ein Maximum ist (wenn der eigentliche Service Level "nicht mehr als" ist), oder ein Minimalwert (wenn der eigentliche Service Level ist "nicht weniger als" ist). Sehen Sie unten ein Beispiel davon, wie sich die Formel ändert:

Zielvereinbarung	Service Level-Schwellenwert	Abweichungsformel
Der Service sollte mindestens 99 % der geplanten Zeit verfügbar sein.	Das Ziel ist der minimale Schwellenwert	$Service_{Dev} = (-1)x \frac{Service_{SL} - Service_{SLT}}{Service_{SLT}}$
Durchschnittlicher MTTR sollte nicht 4 Stunden pro Monat überschreiten.	Das Ziel ist der maximale Schwellenwert	$Service_{Dev} = \frac{Service_{SL} - Service_{SLT}}{Service_{SLT}}$

Wo $Service_{Dev}$ = Serviceabweichung

Wo $Service_{SLT}$ = Erwartete Service-Leistung

Wo $Service_{SL}$ = Tatsächliche Service-Leistung

Das folgende Beispiel veranschaulicht eine minimale Abweichungsberechnung:

Der Service sollte für mindestens 99 % des geplanten Zeitfensters verfügbar sein. Der aktuelle Service Level ist 98 % während des geplanten Zeitfensters.

$$Service_{Dev} = (-1)x \frac{98\% - 99\%}{99\%} = 1.0101\%$$

Abweichungsberichte werden für High-Level-Ansichten für Garantien einer fremden Umgebung (unterschiedlicher Typ von Berechnung) benutzt, und um sie in einem einzelnen Balken mit gemeinsamen Bezug zu aggregieren.

Wenn zum Beispiel im Vertrag die folgende Matrix vorkommt:

Service-Helpdesk	**Servicedomäne-Ticket-Verwaltung Priorität 1**	**Servicedomäne-Ticket-Verwaltung Priorität 2**	**Servicedomäne-Ticket-Verwaltung Priorität 3**
	P1 durchschnittliche Bearbeitungszeit	P2 durchschnittliche Bearbeitungszeit	P3 durchschnittliche Bearbeitungszeit
	P1 % der gelösten Anfragen innerhalb von T1	P2 % der gelösten Anfragen innerhalb von T1	P3 % der gelösten Anfragen innerhalb von T1
	P1 % der beantworteten Anfragen innerhalb von T1	P2 % der beantworteten Anfragen innerhalb von T1	P3 % der beantworteten Anfragen innerhalb von T1
	P1 durchschnittliche Antwortzeit	P2 durchschnittliche Antwortzeit	P3 durchschnittliche Antwortzeit

Die Ergebnisse der Generierung eines Servicedomänen-Berichtes, gefiltert durch den Helpdesk, sind im folgenden Diagramm angezeigt.

Der oben genannte Bericht erlaubt es dem Service Level-Manager, zu verstehen, wie gut der Helpdesk auf verschiedenen Prioritäten funktioniert, ungeachtet des Vertrags oder der Art der Verpflichtung.

Alle Helpdesk-Metriken werden in einen einzelnen auf ihre Priorität basierten Balken aggregiert.

Zum Beispiel zeigt der Balken "Priorität 1" die drei innerhalb der Metrik angegebenen Metriken und aggregiert ihre Abweichung in einen Einzelwert. (Die Aggregationsmethode kann im Berichtassistenten gewählt werden, um durchschnittlich / Minimal / Maximal zu sein.

Mit einem solchen Bericht kann der Manager zum Beispiel schlussfolgern, dass er/sie mehr Ressourcen in Priorität 1 Anfragen investieren muss oder die Verträge im Bezug zu ihnen verändern muss.

Dieses Beispiel zeigt, dass die Modellierung sowohl den Bericht einer einzelnen Verpflichtung, die zeigt, ob ein Vertrag erfüllt oder verletzt wurde, als auch einem breiteren Verwaltungsbericht zur Verfügung stellt, was dem Service Level-Manager erlaubt, seine Ressourcen effizienter zu verwalten und er dadurch seine Servicekomponenten verbessern kann.

Freiformberichte

Freiformberichte ermöglichen es Benutzern, Berichte zu generieren, die auf SQL-Abfragen der CA Business Service Insight-Datenbank oder von einer anderen externen Datenquelle, auf die über einer Verbindung vom CA Business Service Insight-Server zugegriffen werden kann, basieren. Dies schließt auch andere Arten von Datenquellen ein, auf die über ODBC zugegriffen werden kann, wie Excel, Access, Lotus Notes, Textdateien etc. Freiformberichte werden üblicherweise benutzt, um statistische Berichte zu konfigurieren, die auf Daten, die mit den Business-Logik-Befehlen "Tools.SaveRecord" und "Tools.SaveFields" erstellt wurden, basieren.

Freiformberichte schließen über einem Verbindungsstring an einer ausgewählten Datenbank an, und führen eine SQL-Abfrage auf der eine Abfragezeichenfolge verwendenden Datenbank aus. Sie können beiden Strings Parameter hinzufügen, um dynamische Berichte zu erstellen, die es dem Benutzer ermöglichen, bestimmte Werte in die Abfrage durch Eingeben oder Auswählen aufzunehmen, etwa den Benutzernamen und das Passwort für die Verbindung mit der Datenbank.

Freiformberichte werden in den Registerkarten "Diagramm", "Daten" und "Filter" angezeigt, ähnlich den Berichten, die bei der Verwendung des Berichtassistenten generiert werden.

Hinweis: Freiformberichte können nur Diagramme einschließen, wenn alle Spalten, außer der ersten Spalte, numerisch sind. Die Daten in der ersten Spalte dienen als Überschriften in der X-Achse. Die Spaltennamen werden für andere Überschriften verwendet.

Aufgrund der Tatsache, dass Freiformberichte direkten Zugriff auf eine Datenbank und eine offene SQL-Abfrage verwenden, ist die Wartung problematisch. Große Sorgfalt sollte dafür aufgebracht werden, nicht die zugrunde liegenden Daten zu beeinträchtigen, die als eine Quelle für die Freiformberichte dienen. Wenn Berichte von einer externen Datenquelle generiert werden, wird empfohlen, dass ein Benachrichtigungsprozess eingerichtet wird, um sicherzustellen, dass jene Datenquellen nicht verändert werden, ohne zuerst den für die Freitextdatenberichte verantwortlichen Vertragsmanager zu konsultieren.

Beim Erstellen von Freiformberichten sind allgemeine Informationen zu beachten.

- Aufgrund internationaler Datumsformatierungsprobleme ist es oft nützlich, das genaue Format für das Datum durch das Erstellen einer benutzerspezifischen Formel in der Business-Logik anzugeben. Dies konvertiert zu jeder Zeit das Datum in eine Zeichenfolge mit dem gleichen Format und vermeidet Probleme bei der US- und EU-Datumsformatierung. Sie sollte für Daten verwendet werden, die aus der Tabelle T_SLALOM_OUTPUTS über die Befehle "Tools.SaveFields" oder "Tools.SaveRecord" entnommen werden. Die Beispielformel wird unten angegeben:

Funktion "FormatDate(DateField)"

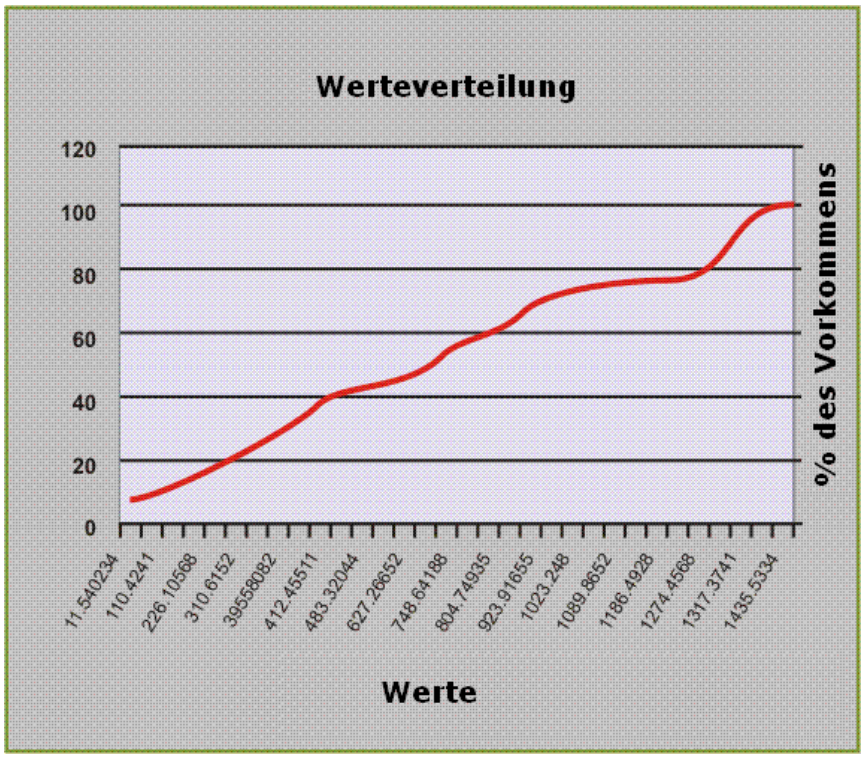
```
If DateField = "" Then
  FormatDate = ""
```

```
    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
```

- Bei der Verwendung der "Uhrzeit"-Parameter vom Business-Logik-Ereignis-Handler, oder eines Zeitstempels vom eventDetails-Objekt, sollten Sie sich immer über die Auswirkungen auf die Zeitzonenverschiebung bewusst sein. Bedenken Sie, dass die Daten beim Schreiben in die Tabelle eher in UTC-Zeit als Ortszeit sein können. Vielleicht müssen Sie die Funktion "Tools.GetLocaleTime()" verwenden, um dies zu korrigieren.
- Sie können weiterhin das Berichts-Designerhilfsprogramm (wenn ein Freiformbericht ein Diagramm erzeugt) verwenden, um die Anzeige anzupassen.
- Die PDF-Exportoptionen für den Freiformbericht sind über die Verwendung des Abschnitts "Berichtsparameter" des Freiform-Setupfensters benutzerdefinierbar. Dinge wie das Seitenlayout (Landschaft, Porträt)
- HTML kann in die Berichte eingebettet werden, um unterschiedliche Funktionen auszuführen, wie Hyperlinks hinzuzufügen oder Spalten- und Linienfarben, Schriften oder Anzeigeneinstellungen zu verändern.
- Datenbankansichten (Oracle-Funktionalität) können in der T_SLALOM_OUTPUTS-Tabelle erstellt werden, um das benötigte SQL für die Berichte zu vereinfachen.
- Bei der Angabe der Parameter für die Berichte ist es möglich, dass eine Reihe an Art und Weisen festgelegt werden, einschließlich: Freitext, Numerisch (gültig), Kennwort (ausgeblendete Zeichen - nützlich für Kennwörter zu DB-Verbindungen), Datum (bei Verwendung des mitgelieferten Datenabgriffshilfsprogramms) und Listen (erstellt durch Angeben einer SQL-Anweisung, um die Liste auszufüllen)
- Die im SQL-Abschnitt des Freiformberichts angegebenen Parameterwerte müssen durch den Parameternamen ersetzt werden, gefolgt von dem "@"-Symbol. Z. B. @PARAM1. Die Verwendung von Anführungszeichen um diesen Parameterwert kann dann erforderlich werden, wenn die Werte Zeichenfolgen sind.

Generische Histogramm-Freiformberichte

Die folgende Abfrage kann in einem Freiformbericht verwendet werden, um die Verteilung von Werten in einer Tabelle nach dem Prozentsatz anzugeben, wie im folgenden Diagramm angezeigt:



Im obigen Diagramm können Sie sehen, welches Verhältnis (Prozentsatzweise) bei den Werten unter 11,5 (0 %), unter 804,74 (~50 %), und unter 1.435,53 sind (100 %) vorliegt.

Wenn das SLA Ziele angibt wie; "x % der Werte sollten unter y sein", helfen die Ergebnisse dieser Freiform bei der Suche der x-y-Werte, die die Einhaltung mit des SLA sichern.

Die folgenden Parameter werden in der Abfrage verwendet:

- **@Query** - eine Auswahlanweisung mit der folgenden Form; wählen Sie *some_value* **val** aus *some_table*. Das Alias "Val" ist verbindlich. Diese Abfrage gibt die Werte an, für die die Verteilung analysiert wird.
- **@Buckets** - die Anzahl der Werte auf der X-Achse. Quellenwerte werden auf diese Zahlen gerundet. Zum Beispiel, wenn Sie @Buckets = 100 angeben, werden die Werte der Quelldaten in 100 Gruppen geteilt und die Abfrage wird berechnen, wie viele Werte innerhalb jeder Gruppe gefallen sind. Je höher @Buckets ist, desto genauer das Ergebnis, das Diagramm ist jedoch optisch nicht ansprechend.
- **@Relation** - der Richtung der Akkumulation. Mögliche Werte; "<=" (wenn Ziel nicht weniger als) oder ">=" (andernfalls).

Die Abfrage kann gegen die Datenquelle oder T_SLALOM_OUTPUTS für die besten Ergebnisse ausgeführt werden.

Die folgende Abfrage erzeugt das Diagramm, wie oben angezeigt:

```
select val,100*records/(select count(*) from (@Query))
von
(
  wählen Sie x.bucket_val val,
    sum(y.records) records
  von
  (
    wählen Sie round(val/bucket_size,0)*bucket_size bucket_val,
      count(*) records
    von
    (
      wählen Sie (max(val)-min(val))/@Buckets bucket_size
      von
      (
        @Query
      )
    ) Parameter,
    (
           @Query
       ) Quelle
    gruppieren nach round(val/bucket_size,0)*bucket_size
    sortieren nach round(val/bucket_size,0)*bucket_size
  ) x,
  (
    wählen Sie round(val/bucket_size,0)*bucket_size bucket_val,
```

```
      count(*) records
    von
    (
      wählen Sie (max(val)-min(val))/@Buckets bucket_size
      von
      (
        @Query
         )
    ) Parameter,
    (
           @Query
       ) Quelle
    gruppieren nach round(val/bucket_size,0)*bucket_size
    sortieren nach round(val/bucket_size,0)*bucket_size
  ) y
  wenn y.bucket_val @Relation x.bucket_val
  gruppieren nach x.bucket_val
  sortieren nach x.bucket_val
)
```

Nachfolgend ist eine Beispiel-Parameterliste (als XML) aufgeführt, die verwendet werden könnte:

```
<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_Kontext_Activation_Success.CSV</value>
                              <text>PDP Context Activation Success</text>
                         </item>
                         <item>
                              <value> wählt den Durchsatz als val aus [gprs
throughput volume by apn.csv]</value>
                              <text> Durchsatz eines einzelnen APN</text>
                         </item>
                         <item>
                              <value> wählt den Durchsatz als val aus [Generic
GPRS Throughput.CSV]</value>
                              <text>Generischer Durchsatz</text>
                         </item>
                    </list>
               </param>
               <param name="@Buckets" disp_name="X Axis Values" type="LIST">
```

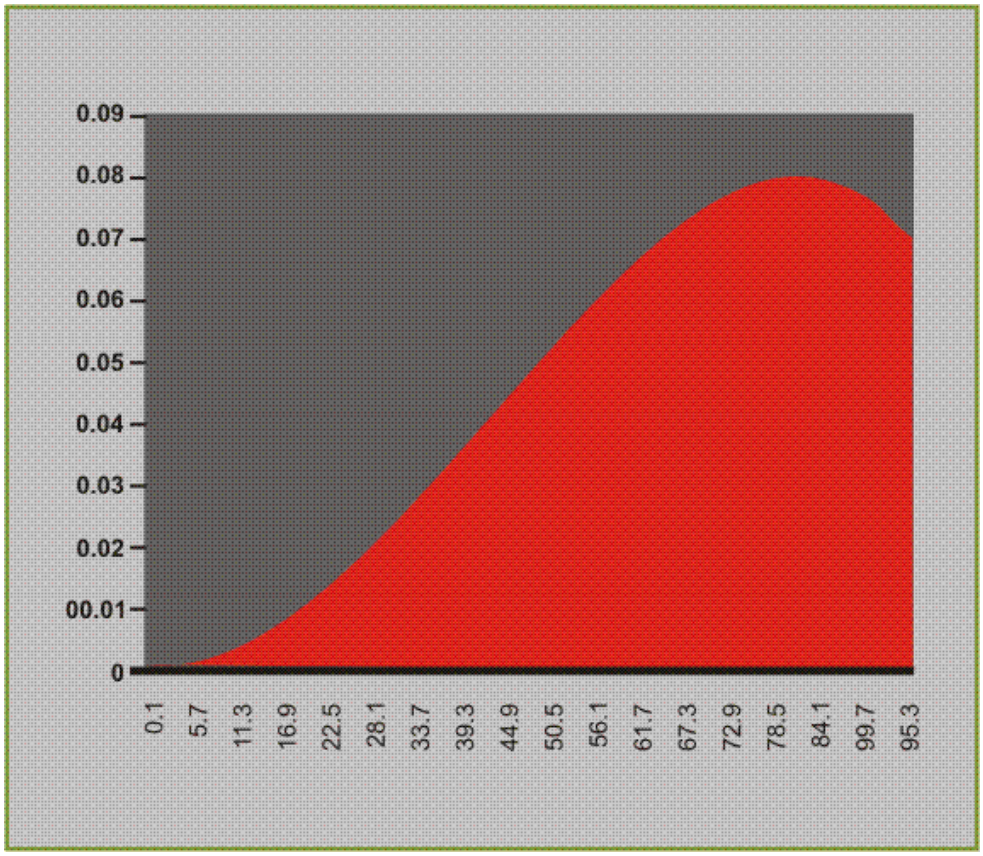
```
                    <value>100</value>
                    <list>
                         <item>
                              <value>25</value>
                              <text>25</text>
                         </item>
                         <item>
                              <value>50</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>liefert zu wenig</value>
                    <list>
                         <item>
                              <value>&gt;=</value>
                              <text>liefert zu wenig</text>
                         </item>
                         <item>
                              <value>&lt;=</value>
                              <text>liefert zu viel</text>
                         </item>
                    </list>
               </param>
          </params>
     </query>
</custom>
```

Kommentare

- Die Abfrage wurde für Oracle- und SQL-Server optimiert. Für andere ODBC-Datenquellen kann es erforderlich sein, dass man "als" vor Spaltenpseudonyme hinzufügen und andere Modifikationen anwenden muss.
- Beim Exportieren der Ergebnisse zu Excel wird empfohlen, dass der Benutzer einen XY-Bericht (Streuung) generiert, um sie darzustellen. Wenn das Diagramm punktiert wird, ist es auch möglich, die Streuung von den Werten anzuzeigen.

Generische Normalverteilung (Gaußglocke) Freiformbericht

Die unten ausführlich behandelte Abfrage kann in einem Freiformbericht verwendet werden, um die normale Verteilung von Werten in einer Tabelle, wie im folgenden Diagramm demonstriert, darzustellen:



Die folgenden Parameter werden in der Abfrage verwendet:

- @Query - eine Auswahlanweisung der folgenden Form:

```
wählen Sie min. (irgendwas) min_val,
          max. (irgendwas) max_val,
          mittel (irgendwas) avg_val,
          stddev (irgendwas) stddev_val
aus table_name
```

  x_val wird benötigt. Diese Abfrage gibt die Werte an, für die die Verteilung analysiert wird.

- @Buckets - die Anzahl der Werte auf der X-Achse. Quellenwerte werden auf diese Zahlen gerundet. Zum Beispiel, wenn Sie @Buckets = 100 angeben, werden die Werte der Quelldaten in 100 Gruppen geteilt und die Abfrage wird den normalen Verteilungswert für jede Gruppe zeigen. Je höher @Buckets ist, desto genauer ist das Ergebnis, die Berechnung ist jedoch schwerer. 100 ist eine gute Wahl für @Buckets.

Zur Wiederholung: Die Abfrage kann gegen die Datenquelle oder T_SLALOM_OUTPUTS für die besten Ergebnisse ausgeführt werden.

Die folgende Abfrage wird das Diagramm, wie oben angezeigt, erzeugen:

```
wählen Sie (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))
aus     (
          @Query
     ),
     (
          wählen Sie     rownum serial
          aus t_psl
          wo     rownum < @Buckets
     )
sortiert nach Serie
```

Seine entsprechenden XML-Parameter:

```
<custom>
     <connection>
          <params/>
          </connection>
     <query>
          <params>
               <param name="@Query" disp_name="Data Type" type="LIST">
                    <value>PDP Context Activation Success</value>
                    <list>
                         <item>
<value>
wählen Sie min(success_rate) min_val,max(success_rate) max_val,avg(success_rate)
avg_val,stddev(success_rate) stddev_val
       aus PDP_Kontext_Activation_Success.CSV
```

```
</value>
                              <text>PDP Context Activation Success</text>
                         </item>
                         <item>
                              <value>
wählen Sie min(throughput) min_val,max(throughput) max_val,avg(throughput)
avg_val,stddev(throughput) stddev_val
       aus [gprs throughput volume by apn.csv]
</value>
                              <text>Durchsatz 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>
```

Kommentare

- Die Abfrage wurde für Oracle- und SQL-Server optimiert. Für andere ODBC-Datenquellen kann es erforderlich sein, dass Sie "als" vor Spaltenpseudonyme hinzufügen und andere Modifikationen anwenden muss.
- Beim Exportieren der Ergebnisse zu Excel wird empfohlen, dass der Benutzer einen XY-Bericht (Streuung) oder einen Bereichsbericht generiert, um sie darzustellen.

Konfigurieren von Dashboard-Seiten

Der folgende Abschnitt enthält Empfehlungen über den Inhalt, der für die CA Business Service Insight-Benutzer konfiguriert werden sollte. Die Empfehlungen sind allgemein und sollten kontospezifischen Kundenanforderungen entsprechen. Die Seiten werden im Folgenden im Zusammenhang eines bestimmten Benutzers beschrieben, der über seine Rolle in der Organisation beschrieben wird.

Geschäftsleitungsmanager-Seiten

Die Annahme ist, dass ein Geschäftsleitungsmanager an abstrakten Ansichten über Abteilungen, Länder, Konten usw. interessiert sein könnte. Ein Geschäftsleitungsmanager arbeitet üblicherweise nicht im Betrieb und benötigt Ansichten, die ihm die Informationen liefern, um Entscheidungen in Bezug auf Strategie zu treffen. Daher könnte ein Vertragsstatus eher dafür relevant sein, in Karten und Aggregative-Dashviews für die Geschäftsleitungsmanager angezeigt zu werden, als ein aktueller Status.

Zum Beispiel können die folgenden Dashviews in der Übersicht enthalten sein:

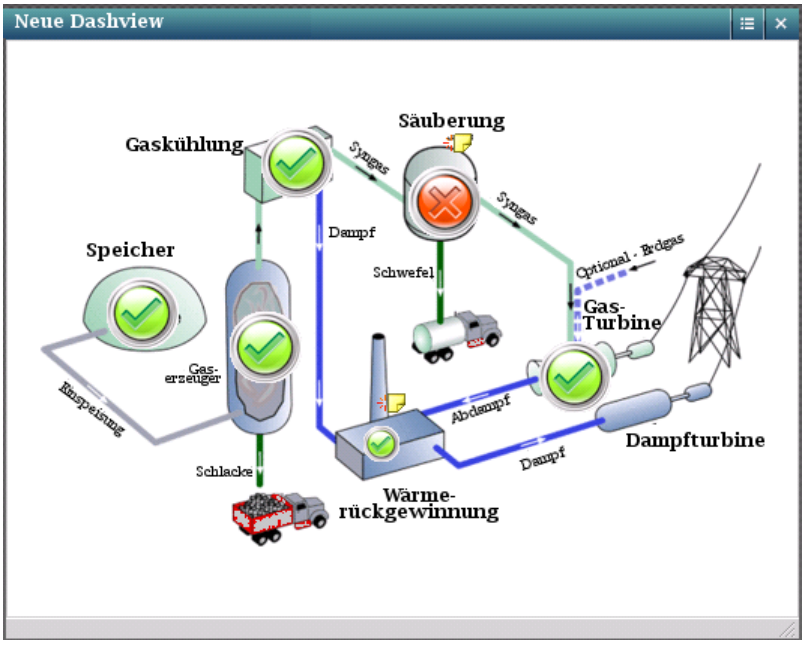
Dashview	Beschreibung
Critical accounts	Enthält alle Vertragsparteien, die als sensitiv markiert werden. Der Geschäftsleitungsmanager wählt die Verträge oder Vertragsparteien aus, die von seinem Standpunkt als sensitiv zu berücksichtigten sind.
Overall performance	Enthält benutzerspezifische Widgets zur Gesamtqualität, Widgets zu aggregativen Ansichten, die alle KQIs der Konten einschließen.
Departments	Verwendet Hintergrundbilder mit Organigrammen und platziert die Widgets in den relevanten Abteilungen. Üblicherweise ist die Servicekomponentengruppe hier nützlich, je nachdem welche Abteilung in der Organisation dargestellt wird.
Geographies	Verwendet Hintergrundbilder mit einer geografischen Zuordnung und lokalisiert Vertragsparteiengruppen-Widgets der relevanten Orte.
Finanzleistung	Enthält Widgets, die aggregative Informationen über Finanzmetrik einschließen.
URL	Enthält Seiten aus dem Unternehmensportal als Beispieltrends, beispielsweise im Außendienst

Empfohlene Berichte für die Seite:

Bericht	Beschreibung
Abweichungsbericht	Die zehn schlechtesten Verträge für einen bestimmten Zeitraum, stellt Informationen über die Bereiche zur Verfügung, die die schlechteste Leistung in Bezug auf die Serviceerbringung liefern.
Finanzbericht für den aktuellen Monat	Fasst den Finanzstatus zusammen, aggregiert im Zeitverlauf

Prozess-Seite:

Diese Seite sollte einen Dashview, der ein Prozessdiagramm mit Widgets darstellt, enthalten, der jede Kette im Prozess wie im Beispiel unten angibt:



Vertragsmanager-Seiten

Seiten die für eine Vertragsmanageransicht konfiguriert wurden, wie gut der Service für jedes der Konten bereitgestellt wird, die er verwaltet. Dashview der entsprechenden Verträge, um den Manager über den aktuellen Service Level der Verpflichtungen von Verträgen auf dem Laufenden zu halten, die unter seiner Verantwortung sind. Außerdem zeigt eine solche Seite die verfügbaren Berichte für jede der Servicekomponenten, die in den Vertrag eingeschlossen werden.

Die Übersichtsseite umfasst die folgenden Dashviews:

Dashview	Beschreibung
My accounts	Widgets von allen Verträgen oder Vertragsparteien, die in der Verantwortung des Vertragsmanagers stehen.
My services	Servicekomponenten über die vom Vertragsmanager verwalteten Konten.
Finanzleistung	Widgets, die aggregative Informationen über Finanzmetriken für die verwalteten Konten enthalten.
URL	Portal wie das CRM-System

Erstellen Sie eine Konto-Seite für jedes der verwalteten Konten, die eine Ansicht der bestimmten Kontoverpflichtungen bereitstellen. Es wird empfohlen, Ansichten der aktuellen Statusberechnung mit dem Vertragsstatus zu kombinieren, um dem Kontenverwalter die Möglichkeit zu geben, proaktiv und schnell zu sein. Zum Beispiel:

Konto-Seite	Beschreibung
Account obligations	Enthält in den Vertrag eingeschlossene Metriken
Finanzleistung	Widgets, die aggregative Informationen über Finanzmetriken für die verwalteten Konten enthalten.

Service-Manager-Seiten

Für einen bestimmten Service oder Domäne verantwortlichen Manager zu konfigurierende Seiten, die eine detaillierte Ansicht der Serviceziele in den unterschiedlichen Verträgen und den Status seiner Ziele anzeigen. Darüber hinaus enthält eine derartige Seite auch Berichte, welche die servicekritischen Service Level-Parameter hervorheben, die gemessen werden.

Die in die Seiten des Service-Managers eingeschlossenen Dashviews sind den Account-Manager-Seiten ähnlich, wo nur die Informationen in einem Service Level angezeigt und aggregiert werden, anstelle im Vertrag oder der Vertragsparteiebene.

Startseiten

Eine Dashboard-Seite kann als Startseite zugewiesen werden, um einen benutzerdefinierbaren Gateway für Interaktion mit dem System zu aktivieren, um einen schnellen Zugriff zu Informationen und Aktionen zu erhalten.

Hinzufügen von Service Level-Alarmprofilen

Das Alarmprofil ermöglicht die Definition von Bedingungen, unter die eine Alarm-Benachrichtigung mit einem bestimmten Gerät an einen vordefinierten Empfänger gesandt wird.

Vor der Erstellung von Profilen ist es eine Überlegung wert, welche Bedingungen wichtig genug sind, eine Benachrichtigung zu erfordern, und wer die zukünftigen Empfänger sein sollten. Es ist wichtig, Profile anzugeben, die dem Vertragsmanager und dem Systemadministrator dabei helfen können, Probleme und gefundene Serviceprobleme effizient zu verwalten. Alle Benachrichtigungen sollten auf die erforderlichen Ressourcen der dringendsten Probleme ausgerichtet sein, um eine Situation von Vertragsbruch zu verhindern.

Beim Hinzufügen von Service Level-Alarmen gibt es eine Reihe von Feldern, die dafür verwendet werden können, den Status der Metrik zu bestimmen, und um zu beschließen, ob eine Alarmbedingung ausgelöst werden soll oder nicht. Jede der standardmäßigen Metrikdetails kann für das Filtern und die Bedingungskriterien (wie Vertrag, Service, Metriknamen, Ziele, Zeitfenster, Ergebnisse Service Level etc.) verwendet werden. Über Vorhersagen für Service Level kann auch alarmiert werden, in welchen einige zusätzliche Werte, wie beste/schlechteste Fallabweichung angegeben und Service Levels auch geschätzt werden können. Diese sind äußerst nützlich für eine Entscheidung, ob eine Service Level-Vertragsverletzung innerhalb eines bestimmten Zeitraums behoben werden kann oder nicht.

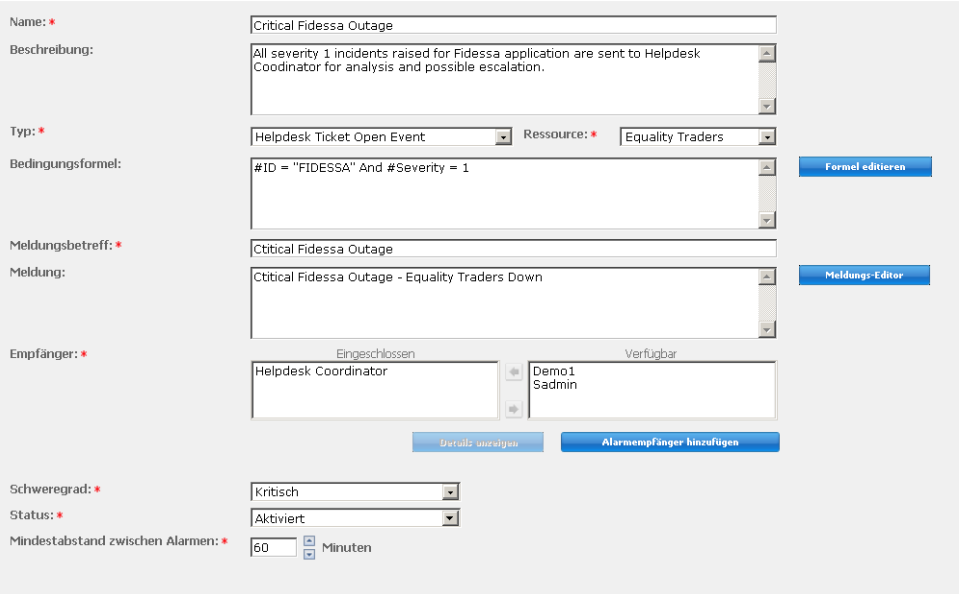
Auf das Systemprotokoll und den allgemeinen Abschnitten von CA Business Service Insight basierende Alarme sind andere äußerst nützliche Funktionen und ermöglichen eine auf eine Aktion, die innerhalb des Systems auftrat (die ins Systemprotokoll geschrieben wurde), basierende Alarmmeldung zu senden. Da fast alle Aktionen protokolliert werden, stellt dies einen sehr breiten Anwendungsbereich für die Alarmprofile dar. Allgemeine Systemprotokoll-Alarmprofile bestehen aus:

- Benachrichtigung von einem Adapter mit laufendem Alarm für den Datenexperten.
- Im Protokoll eingetragene "Fehler"-Bedingungen, die per E-Mail an den Systemadministrator zu senden sind.
- Angepasste Fehler, die von der Business-Logik erstellt wurden und die von den Datenquellen bestimmte relevante Bedingungen erfüllen, können eine E-Mail an die für den Service innerhalb der Organisation verantwortliche Gruppe und an den Vertragsmanager generieren. (Z. B. Vorfälle mit hoher Priorität wurden nicht innerhalb des Zeitrahmens gelöst > an den Helpdesk einen Alarm senden, um das Problem zu eskalieren)

- Wiederholte ungültige Anmeldungsversuche können per E-Mail an den Systemadministrator gesandt werden, um zu ermitteln, ob jemand versucht in das System einzudringen.
- Neue Übersetzungseinträge kamen vom Adapter (z. B. Neue Ressourceneinträge wurden in einer Datenquelle festgestellt, die eine Übersetzung im System benötigen können, um das SLA zu ermöglichen, richtig berechnet zu werden).

Das folgende Alarmprofil-Detailfenster zeigt die Konfiguration eines benutzerspezifischen Alarmprofils, das den benutzerspezifischen Event-Typ "Helpdeskereignis Ticket offen" meldet. Wenn die Kriterien übereinstimmen, wird ein kritischer Alarm dem Helpdesk-Koordinator zugesandt, um sicherzustellen, dass dieses Ticket effizient verwaltet wird. Dies könnte für Situationen nützlich sein, wo eine Anwendung zu irgendeinem Zeitpunkt unter Untersuchung steht und einen besonderen Grad an Sorgfalt benötigt.

Details des Alarmprofils



Im Folgenden ist ein Beispiel für die Umsetzung eines Alarms über eine Service Level-Vertragsverletzung angegeben, basierend darauf, ob das SLA die Bedingungen der aktuellen Service Level und der verbleibenden Zeit in den Businesskontrollzeiträumen der Metrik noch erfüllen kann.

Beispiel: Reversibler SLA-Vertragsverletzungsalarm

Dieser Alarm wird ausgelöst, wenn ein Ziel (Metrik) gegenwärtig verletzt wird, aber da das beste Fallszenario die Einhaltung des Ziels anzeigt, kann die Vertragsverletzung vermieden werden, wenn künftig ein besser geeigneter Service Level zur Verfügung gestellt wird.

Typ: Service Level-Vorhersage

Bedingungsformel:

#Deviation>0 And #BestCaseDeviation<=0 And
#BestCaseServiceLevel<>#WorstCaseServiceLevel And #ClusterItem = ""

Meldung:

Beachten Sie, dass das Ziel "#Metric" des Vertrags "#Contract" gegenwärtig verletzt wird, aber die Verletzung reversibel ist.

Während das Service Levelziel #Target ist, werden die folgenden Service Levels vorausgesagt:

- Service Level inzwischen: #ServiceLevel (Abweichung: #Deviation%).
- Schlechtester Szenario-Service Level: #WorstCaseServiceLevel (Abweichung: #WorstCaseDeviation%).
- Bester Szenario-Service Level: #BestCaseServiceLevel (Abweichung: #BestCaseDeviation%).

Kapitel 4: Installation und Bereitstellung

Dieses Kapitel enthält folgende Themen:

Einführung (siehe Seite 208)
Vorbereitungen (siehe Seite 210)
Installation (siehe Seite 212)

Einführung

Sobald die Konfigurationsphase abgeschlossen wurde und die Systemberechnungen richtig überprüft wurden, ist das System bereit für den Einsatz in der Produktionsumgebung. Dieses Kapitel deckt alle Schritte ab, die während dieser Phase benötigt werden können. Diese Schritte könnten zwischen den Installationen variieren; allerdings sollten die folgenden Elemente überprüft werden, ob alle Vorbereitungen für die Integration des Systems in einer Produktionsumgebung ausgeführt wurden.

- Produktionshardware und -software ist verfügbar und bereit für die Installation.
- Mail-Server ist konfiguriert, um externen/internen Mailversand zu ermöglichen.
- FTP-Agent-Existenz (für Dateiübertragungen erforderlich).
- Datenquelleneingaben sind verfügbar und funktionieren

Es gibt zwei Phasen innerhalb der Installation. Während der ersten Phase, der Installationsphase, sollte der Systemadministrator die CA Business Service Insight-Komponenten in der Produktionsumgebung installieren und integrieren. Während der zweiten Phase wird die Systemfunktionalität überprüft und das System sollte in einem überwachten Status gesetzt werden, um sicherzustellen, dass Systemprozesse und Benutzeroberflächen-Funktionen (UIs) alle richtig funktionieren.

Während des Installationsprozesses kann es erforderlich sein, dass auf die Produktionsserver über Remote-Verbindungstools zugegriffen werden muss, die in der Anwendung und auf Webserver installiert werden sollten, um eine vollständige und sichere Installation zu ermöglichen.

Die Installation der Oracle-Datenbank sollte wenn möglich von einem Oracle-DBA-Fachmann ausgeführt werden, um sicherzustellen, dass die Konfiguration richtig ausgeführt wird, und dass sie den an Ort und Stelle möglicherweise vorhandenen Unternehmensanforderungen entsprechen. Alternativ kann die Datenbankerstellung dem Systemadministrator überlassen werden, der das mit der CA Business Service Insight-Installation mitgelieferte Datenerstellungshilfsprogramm verwendet. In diesem Fall sollte die Installation vom DBA überprüft werden, um sicherzustellen, dass sie erfolgreich abgeschlossen wurde. Wenn das System schon auf einem Entwicklungssystem konfiguriert wurde und zugeordnet werden muss, ist es notwendig, den Datenbankinhalt zu dieser neuen Produktionsdatenbank zu verschieben.

Manchmal, sobald das vollständige System auf die neue Produktionsumgebung übertragen wurde, ist es empfehlenswert, die gesamte Berechnung und die im System gespeicherten Rohdaten zu verwerfen und das System von Grund auf (Servicekatalog, Ressourcen und Übersetzungen natürlich vor Ort belassend) neu zu laden. Dies ist eine ausgezeichnete Art und Weise, die Funktion des ganzen Systems innerhalb der Produktionsumgebung zu testen.

Zusammenfassend sind folgende Schritte in dieser Phase enthalten:

- Installation und Verbindung des Systems.

- Laden der konfigurierten Systemdatenbank, wenn benötigt.
- Integrieren der Verbindungen zu E-Mail, Exchange, SMS etc.
- Testen der Funktionalitäts- und Leistungsperformance.
- Installieren, Konfigurieren und Testen der Remote-Verbindungen.
- Reinitialisieren des Systems für die Bereitschaft zum "Echtzeit"-Betrieb.
- Aktivieren der Daten-Adapter, um damit zu beginnen, die Datenquellen abzufragen und die Rohdaten zu laden.
- Einschalten der CA Business Service Insight-Engine und Ermöglichen, mit den Berechnungen zu beginnen.
- Abschließen aller notwendigen Dokumentationen, wie Systemverwaltung, Datenbank und andere Wartungsverfahren.
- Sicherstellen, dass die Benutzer alle Schulungen erhalten haben.

Vorbereitungen

Um eine effiziente Installation zu sichern, ist es wichtig, dass ein paar Dinge im Voraus vorbereitet werden:

1. **Der Datenbankexport der Entwicklungsumgebung**. Dieser Datenbankexport sollte mit CA-anerkannten Skripten ausgeführt werden, die eine Extract (Ausgabe)-Datei erstellen, um sie zurück in die Software auf dem Produktionssystem importieren zu können.

   Hinweis: Es ist wichtig, dass der Export von einem Datenbankbenutzer ausgeführt wird, der hinreichende Berechtigungen hat und dass dieser gleiche Benutzer erreichbar ist, wenn die Datenbank importiert wird. Zu diesem Zweck kann entweder das "**oblicore**"-Konto verwendet werden, da dies immer auf jedem System erstellt wird, oder alternativ das "**sys**"-Konto. Stellen Sie allerdings sicher, dass das "**sys**"-Kennwort auf der Zieldatenbank verfügbar ist, um den möglicherweise auftretenden Importprozess zu ermöglichen.

2. **Database scripts** - wenn der DBA die Datenbankerstellungsskripte ändern will, sollten sie im Voraus von einem CA-DBA überprüft werden. Diese Skripte sollten im Voraus überprüft werden und bereit sein, um eine reibungslose Installation zu ermöglichen. Es ist üblich, dass verschiedene Organisationen Kommentare und Änderungen haben, die vom CA-Datenbankadministrator genehmigt werden müssen, um sicherzustellen, dass die Datenbankkonfiguration mit lokalen Richtlinien übereinstimmt.

   - **Server connectivity and accessibility** - sowohl Anwendungs- als auch Webserver sollten einen Remote-Verbindungsclient installiert haben, um externe Zugänge zu ermöglichen (wenn ein physikalischer Zugriff am Einsatzzeitort oder in der Zukunft für Problemunterstützung nicht möglich ist). Die externe Zugangsmöglichkeit ist auch wichtig, um während einer Bereinigungsperiode das System zu überwachen. Wenn ein externer Remoteanschluss erforderlich ist, könnte es aufgrund der zusätzlichen Sicherheitsimplikationen länger dauern, zu organisieren; daher sollte er vorher adressiert werden.
     Remote-Verbindungen können mit der Verwendung eines der folgenden Tools hergestellt werden:
   - PcAnyWhere
   - Proxy Host / Master
   - Microsoft Terminal Server
   - Remote Desktop
   - VNC
   - Jedes andere von der Organisation unterstützte Remote-Verbindungstool.

   Hinweise:

1. Microsoft Terminal Server nimmt mit dem Server durch das Simulieren einer Serveranmeldung Verbindung auf, im Gegensatz zu PcAnyWhere, VNC, Proxy und anderen Tools.
2. Manche Oracle-Software kann nicht über Microsoft Terminal Server installiert und muss lokal installiert werden.

3. **Data source accessibility** - Die Datenquelle sollte zugreifbar für manuelle Bearbeitung sein und die ODBC-Verbindung sollte so konfiguriert werden, dass es dem Adapter möglich ist, sich mit der Datenquelle zu verbinden.
4. **Server security** - sowohl auf Anwendungs- als auch auf Webservern wird ein Benutzerprofil mit lokalen Administratorrechten benötigt. Wenn die Datenbankplattform ein Standard-Windows-Betriebssystem ist, wird ebenfalls ein Benutzerprofil mit lokalen Administratorrechten benötigt. Für Unix-Plattformen sollte der Datenbankadministrator die entsprechenden Berechtigungen haben, um die Datenbank auf dem Server zu erstellen.
5. **Access to Internet** - dies ermöglicht dem **Systemadministrator** mit dem Internet für Betriebssystem- oder Anwendungsaktualisierungen eine Verbindung aufzunehmen, wenn erforderlich.
6. **Ein Standard-Desktop-PC des Benutzers zum Testen der Anwendungsfunktionalität**. Beachten Sie, dass die Anwendung die Verwendung von einigen zu installierenden ActiveX-Steuerelementen benötigt, was oft von organisatorischen Sicherheitsrichtlinien gesperrt wird.
7. **Training schedule with relevant staff** - Einführende Schulungen, die es den Mitarbeitern ermöglichen, Akzeptanzschulungen durchzuführen und beginnen können, mit dem System zu arbeiten.

Installation

Der Installationsprozess wird ausführlich im Installation Guide beschrieben und beinhaltet die folgenden Aktivitäten:

1. **Datenbank-Installation**:

   Die Datenbank-Installation liegt in der Verantwortung des Datenexperten oder Oracle DBA und sollte in besonderen Situationen unter der Leitung von einem CA-Ingenieur stehen. Die Datenbankinstallation beinhaltet die folgenden Schritte:

   - Die Oracle-Installation auf dem Datenbankserver.
   - Die Oracle-Patchinstallation auf dem Datenbankserver (wenn benötigt - Es sollten immer die neuesten Support-Serviceversionen der Oracle-Datenbanksoftware verwendet werden).
   - Die Oracle-Clientinstallation auf dem Anwendungs-/Webserver.
   - Die Oracle-Client-Patchinstallation auf Anwendungs-/Webserver (wenn benötigt - Stellen Sie immer sicher, dass diese Version mit dem Server übereinstimmt).

2. **Installation der CA Business Service Insight-Anwendung**:

   Die Installation der Anwendung wird vom Systemadministrator durchgeführt. In Fällen, wo die Installation bereits in der Testumgebung (während der Konfigurationsphase) bei der Integration und den Adaptertests ausgeführt wurde, ist nur ein Initialisierungsstatus oder Datenbankimport erforderlich. Eine Installation in der Produktionsumgebung kann dann erforderlich sein.

   Die Installation der Anwendung beinhaltet die folgenden Schritte:

   - CA Business Service Insight-Datenbankerstellung.
   - Anwendungsserverinstallation.
   - Webserverinstallation.
   - Adapterinstallation.

Installieren Sie immer das neueste CA Service-Release auf den Anwendungs-/Webserver. Während der Installation der Anwendung und der Service Releases werden SQL-Skripte eingeschlossen, um sicherzustellen, dass die Datenbank auf der richtigen Struktur für das Release aktualisiert wurde. Dieses ist alles im Installationsverzeichnis der Anwendung für den Fall gespeichert, das die Datenbank zu einem späteren Zeitpunkt aktualisiert werden muss. (Zum Beispiel wenn Sie die Datenbank aus einer Sicherung importieren müssen, die auf einem früheren Service Release erstellt wurde. In diesem Fall müssen Sie das neueste Service Release-Aktualisierungsskript ausführen, um sicherzustellen, dass die Struktur von der Datenbank im Einklang mit den binären Komponenten installiert wurde, als Teil des gleichen Service Release)

Optional können Sie auch zusätzliche Supporttools, wie PLSQL-Entwickler oder alternative SQL-Hilfsprogramme installieren, um in einer niedrigeren Ebene Datenmanipulation zu unterstützen, die erforderlich werden kann.

Importieren oder Exportieren zwischen Umgebungen (Datenexperte)

Diese Situation deckt die Migration von Adaptern ab, die konfiguriert und in einer Entwicklungs-/Testumgebung auf einer Echtzeit- oder Produktionsumgebung getestet wurden. Es wird davon ausgegangen, dass die Produktionsumgebung schon mit einer standardmäßigen CA Business Service Insight-Installation installiert wurde und die Datenbank vorhanden ist.

Der Prozess besteht aus dem Folgenden:

So exportieren Sie die Datenbank und importieren sie in die neue Umgebung

1. Stoppen Sie jede Arbeit auf dem Entwicklungssystem, und wählen Sie einen logischen Punkt im System, um einen Export auszuführen, der für das Produktionssystem verwendet werden kann. Fahren Sie alle CA Business Service Insight-Servicekomponenten und CA Business Service Insight-COM+-Komponenten herunter. Verwenden Sie für den Export die standardmäßige CA Business Service Insight-Systemexportskripte. (Kontaktieren Sie bei Bedarf den CA Support)
2. Nehmen Sie die Kopie des Datenbankextrakts, und platzieren Sie sie auf dem Betriebsserver, der zum Importieren bereit ist. Beachten Sie, dass die Oracle-Versionen der Entwicklungs- und Produktionsdatenbanken übereinstimmen müssen. Importieren Sie die Datenbank, die die standardmäßigen Systemimportskripte verwendet (mit den Exportskripten geliefert)
3. Sobald dies abgeschlossen ist, prüfen Sie auf Importprobleme. Wenn keine vorhanden sind, stellen Sie sicher, dass Sie die neuesten Service-Release SQL-Skripte (wenn erforderlich) ausführen.
4. Starten Sie die "Kleinen Assistenten"-Verknüpfung, um die Datenbank für die Einstellungen des neuen Produktionssystems zu konfigurieren.
5. Starten Sie alle CA Business Service Insight-Servicekomponenten, und melden Sie sich am Produktionssystem an, um zu bestätigen, dass das System richtig importiert wurde.

So migrieren Sie die Adapter

1. Installieren Sie den Adapter mit dem AdapterManager-Hilfsprogramm mit den Einstellungen ähnlich dem Adapter den Sie importieren (stellen Sie sicher, dass die Benennung für den weiteren Schritt genau gleich ist, um korrekt zu arbeiten)
2. Kopieren Sie die Adapterkonfigurationsdatei vom Entwicklungssystem zum neuen Adapter-Ordner auf dem Produktionssystem. Überschreiben Sie die angegebene Standardkonfiguration. (Sie sollten sicherstellen, dass die Datei überschrieben wird. Andernfalls kann es sein, dass die Benennung nicht gleich ist, was dann während der Ausführung Probleme verursacht.)
3. Aktualisieren Sie die Adapterkonfigurationsdatei - die Verbindung mit dem CA Business Service Insight-Server und der Datenquelle sollte aktualisiert werden, um in die neue Umgebung zu passen. Der OblicoreInterface-Abschnitt sollte mit dem richtigen Adapter-Port aktualisiert werden. Der DataSourceInterface-Abschnitt sollte mit dem richtigen ConnectionString oder Dateinamensmuster oder -pfad aktualisiert werden, wenn erforderlich.
4. Stellen Sie sicher, dass alle benötigten ODBC, DSNs (Datenquellennamen) eingerichtet sind und auf dem neuen Rechner arbeiten.
5. Testen Sie die Adapter-Konnektivität.
6. Testen Sie die Adapter-Ausführung.
7. **Übersetzung** - in Fällen, wo es Übersetzungsskripte gibt, müssen sie aktiviert sein. Überprüfen Sie, dass sie synchron mit dem Adapter sind und dass die Übersetzungen erwartungsgemäß funktionieren. Wo eine manuelle Übersetzung verwendet und nicht vorher abgeschlossen wurde, sollten weitere Übersetzungen zu diesem Zeitpunkt ausgeführt werden.
8. **Adapterplanung** - Planen der Adapter, um erwartungsgemäß auszuführen. Wenn der Adapter als eine Anwendung in einem einmaligen Ausführungsmodus angegeben wurde, kann er über den Windows-Planer geplant werden. Zu sehen unter Control Panel > Scheduled Tasks. Siehe Ausführungsmodi des Adapters (siehe Seite 88) für weitere Informationen.

Integration - Mail-Servereinrichtung (Systemadministrator)

Um die Benachrichtigungsfunktionen vom System zu aktivieren, ist es notwendig zu wissen, welcher Mail-Server und welches Postfach verwendet wird, um CA Business Service Insight-E-Mails zu senden. Mit dem Mail-Server werden Mails übertragen, indem sie als SMTP von den CA Business Service Insight-Servern zu diesem Mail-Server unter Verwendung des angegebenen Kontos gesendet werden. Nachdem das Setup des Mailservers abgeschlossen wurde, können Sie E-Mail-Funktionen in den Funktionen für Alarme und Vertragsgenehmigungen in CA Business Service Insight verwenden.

Klicken Sie auf das Menü "Administration", und wählen Sie "Site-Einstellungen" und "Alarme" aus. Konfigurieren Sie die E-Mail-Definitionen im Alarm-Einstellungsabschnitt einschließlich des E-Mail-Servers, der Senderadresse und Absendername mit Angaben der SMS-Anbieterinformationen für die Verwendung von SMS-Gateways.

Ein Teil des Integrationstests ist es zu überprüfen, ob der Anwendungsserver den Mail-Server der Organisation erreichen und diesen verwenden kann, um CA Business Service Insight-Alarme zu senden.

Um die Verbindung zwischen dem Mailserver und dem Anwendungsserver zu testen, führen Sie den folgenden Befehl auf dem Anwendungsserver aus:

```
C:\Documents and Settings>telnet ORGANIZATION-MAIL 25
```

ORGANIZATION-MAIL

Definiert den Mailserver, den Sie auf Ihrem E-Mail-Client definiert haben. Setzen Sie sich mit Ihrem Systemadministrator in Verbindung, um diese Einstellung zu erhalten, wenn Sie sich nicht sicher sind.

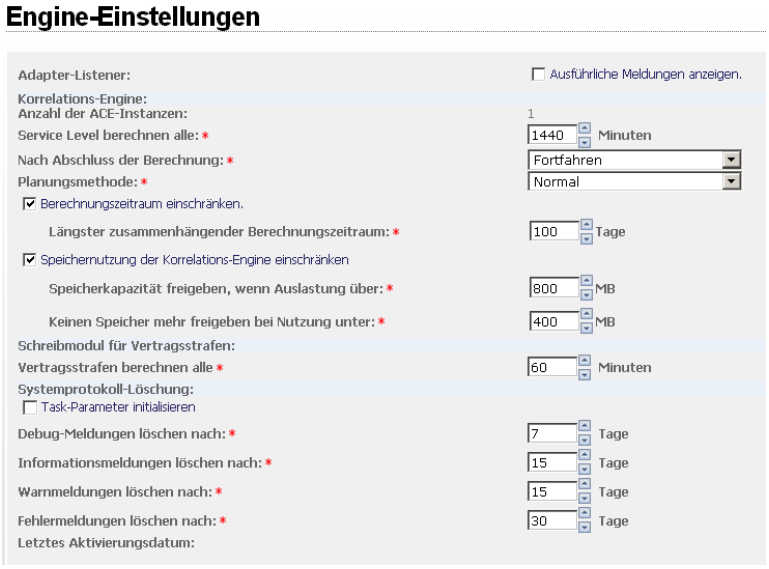
So prüfen Sie den Outlook-Client-Mail-Server:

1. Gehen sie in Microsoft Outlook auf "Extras" > "E-Mail-Konten", und wählen Sie "Ansicht" aus oder ändern Sie die vorhandenen E-Mail-Konten.
2. Klicken Sie auf "Ändern".
3. Kopieren Sie den Microsoft Exchange-Server. Dieser Server ist der Mailserver Ihrer Organisation**.**

   Wenn es dann eine Antwort vom Server mit diesem Befehl gibt, ist die Verbindung erfolgreich gewesen. Die Antwort sollte dem Folgenden ähnlich sein:

   Wenn Sie eine andere Meldung empfangen, bedeutet dies, dass keine Verbindung zwischen den beiden Servern besteht. Wenden Sie sich an den Sicherheitsadministrator.

Festlegen von Systemvoreinstellungen (Systemadministrator)



Der Prozess, die Systemvoreinstellungen festzulegen, beinhaltet, die dazugehörigen Werte auf die Systemvariablen anzuwenden. Klicken Sie im Menü "Verwaltung" auf "Site-Einstellungen", "Engines", um das Dialogfeld mit den Engine-Voreinstellungen zu öffnen.

Weitere Informationen in Bezug auf die verschiedenen Parameterempfehlungen finden Sie im Administratorhandbuch.

Sicherheitseinstellungen (Systemadministrator)

Die Sicherheitseinstellungen beinhalten die Erstellung von Benutzer, Benutzergruppen und Rollen. Standardmäßig verbinden sich alle Benutzer mit der während der Installation der Anwendung angegebenen Organisation. Allerdings ist es auch möglich, bei Bedarf zusätzliche Organisationen zu erstellen.

Die meisten der erforderlichen Definitionen sind während der Konfigurationsphase schon abgeschlossen, deshalb ist es normalerweise nur erforderlich, etwas Feinabstimmung der zusätzlichen Einstellungen durchzuführen, die seit dieser Zeit identifiziert sein könnten.

Weitere Informationen über Sicherheitseinstellung finden Sie in der Onlinehilfe oder Administratorhandbuch.

Angeben von Einstellungen für SSA-Synchronisation

Wenn Sie CA Spectrum Service Assurance (SSA) verwenden, um Services für CA Business Service Insight zu finden, können Sie Einstellungen konfigurieren, um die automatische Synchronisation zu aktivieren. Mit Automatisierungsfunktionen können Sie die Liste der Services und die Servicedaten aktuell halten.

Hinweis: Sie müssen Zugriff auf die RESTful API in SSA haben, um diese Einstellungen zu bearbeiten.

Führen Sie folgende Schritte aus:

1. Klicken Sie im Menü "Verwaltung" auf "Site-Einstellungen", "SSA-Einstellungen".

   Das Dialogfeld "SSA-Einstellungen" wird geöffnet.

2. Geben Sie folgende Informationen in das Feld "SSA-Anwenderauthentifizierung" ein:

   Server-URL

   Gibt die URL für den SSA-Zielserver an.

   Anwender-ID

   Gibt die Anwender-ID für den SSA-Server an.

   Kennwort

   Gibt das Kennwort für die Anwender-ID des SSA-Servers an.

3. Geben Sie folgende Informationen in das Feld "Synchronisationsoptionen" ein:

   Automatische Synchronisation

   Gibt an, dass die Synchronisation automatisch entsprechend der Synchronisationshäufigkeit (siehe nächsten Abschnitt) erfolgt.

   Synchronisationshäufigkeit festlegen

   Gibt die Häufigkeit an, zu der nach neuen Services gesucht wird. Sie können einen Wert in Stunden oder Tagen angeben.

   Manuelle Synchronisation

   Ermöglicht eine manuelle Synchronisation aus dem Dialogfeld.

4. Geben Sie folgende Informationen in das Feld "Standardmäßige Datenoptionen" ein:

   Standardservice auf:

   Ermöglicht für erkannte Services das Festlegen des Standards auf "Verwaltet" oder "Unverwaltet".

   Berechnung von Vergleichsmetriken aktivieren

   Ermöglicht die Angabe der Standardaktion für das Festlegen von Vergleichsmetriken für SSA-Services.

5. Klicken Sie auf "OK".

   Ihre SSA-Einstellungen werden aktiviert.

Anhang A: Beispiele für Servicedomänen und Domänenkategorien

Die folgende Tabelle zeigt eine Aufstellung von üblichen Servicedomänen und Domänenkategorien, die üblicherweise verwendet werden.

Servicedomäne	Domänenkategorie	Kommentare
Systemverfügbarkeit	% der Zeit verfügbar	
	Anzahl von Ausfällen/Ausfallzeiten	
	Mittlere Reparaturzeit	
	Mittlere Zeit zwischen Fehler	
Service-Verfügbarkeit	Minuten an Ausfallzeit	
	Anzahl der Störtage	
Finanzmanagement	Service-Kosten	
Verarbeitungsleistung	% der Verarbeitung pünktlich abgeschlossen	
	Anzahl abgeschlossener Verarbeitungszyklen	
Incident Management	% Incidents gelöst < T1	
	% Incidents reagierten auf < T1	
	Anzahl von Incidents	
	% Incidents konvertierten in Probleme	
Kundenzufriedenheit	% Kundenzufriedenheit	
	Durchschnittliche CSI-Bewertung	
Helpdesk-Leistung	% unerledigte Anrufe	
	Durchschnittliche Anrufzeit	
	% FLLR	(Erste Ebene-Lösungsrate)
Datenqualität	% Genauigkeit	
	% Pünktlichkeit	
	Anzahl der Fehler/Defekte	

Anhang B: Beispiele für Fallstudien

Dieses Kapitel enthält folgende Themen:

Vertragsbildungsbeispiele (siehe Seite 221)
Beispiel für eine Finanzmetrikmodellierung (siehe Seite 232)
Beispiele für die Datenmodellierung (siehe Seite 240)
Beispiel für die Verwendung von anwenderspezifischen Attributen (siehe Seite 250)
Beispiele eines Übersetzungsskripts (siehe Seite 254)
Beispiele für das Business-Logik-Skripting (siehe Seite 260)
Beispiele für das Schreiben einer effektiven Business-Logik (siehe Seite 277)
Fallstudie 19: Adapterassistent für dateibasierte Datenquelle (siehe Seite 295)
Fallstudie 21: LDAP-Integration – Beispiel (siehe Seite 311)
Fallstudie 22: Mit PERL generierte Berichte (siehe Seite 318)

Vertragsbildungsbeispiele

Verwenden Sie für jede der folgenden Fallstudien die folgenden Kategorisierungselemente für die beschriebenen Ziele:

- Service
- Servicedomäne
- Domänenkategorie
- Zeitfenster
- Ziel
- Kontrollzeitraum
- Maßeinheit
- Berechnungsentwurf
- Vertrags-\Metrik-Parameter

Fallstudie 1: Systemverfügbarkeit

Betrachten Sie die folgende Vertragsgarantie:

"System X, muss mindestens 99,6 % im Monat während der Geschäftszeiten verfügbar sein".

Dies könnte durch Verwendung der folgenden CA Business Service Insight-Entitäten beschrieben werden:

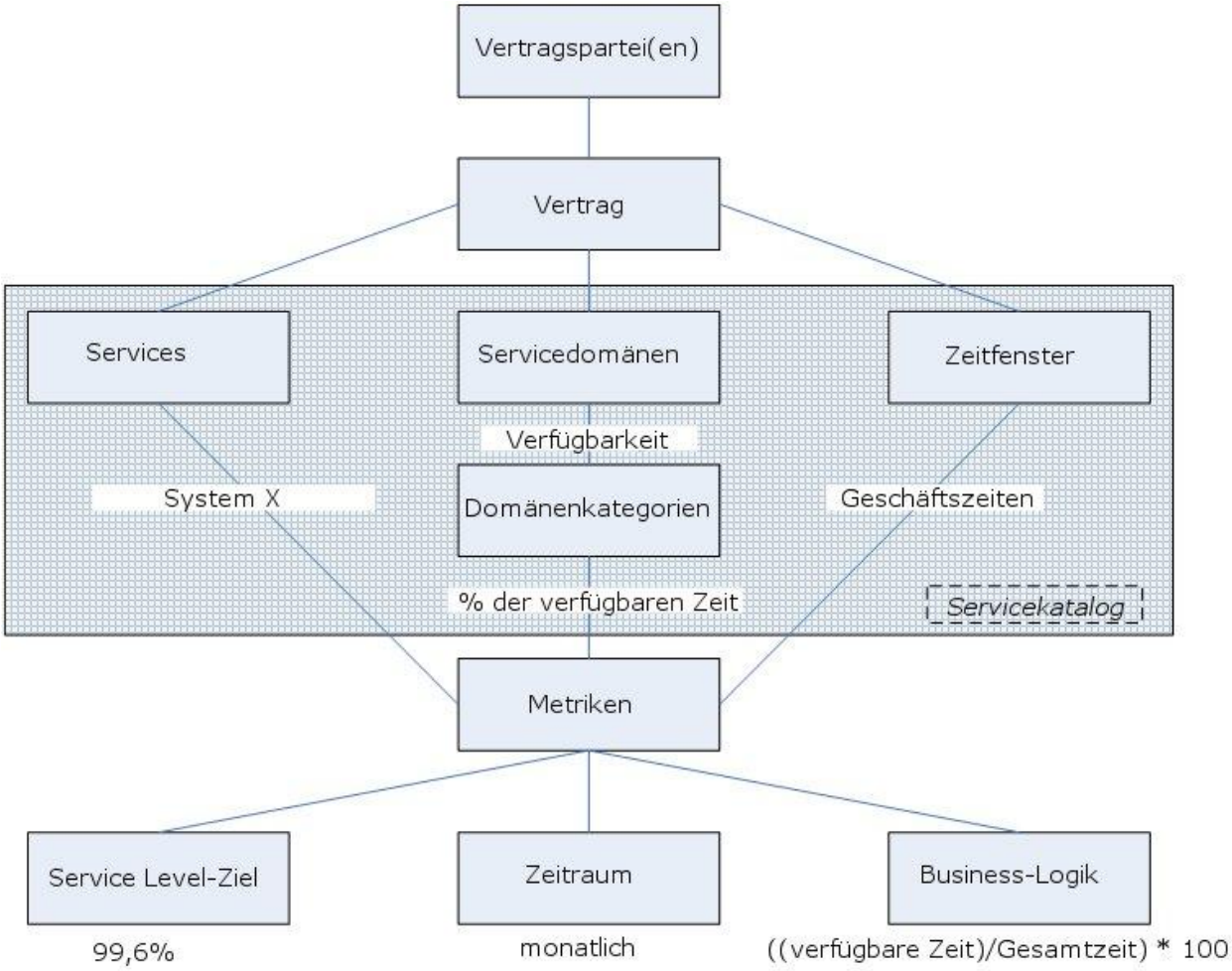
- **Metrikname**: System X % der Zeit verfügbar
- **Kontrollzeitraum**: 1 Monat
- **Zeitfenster**: Geschäftszeiten (spätere Definition notwendig)
- **Business-Logik**: ((Zeitverfügbarkeit)/(komplette Zeit))*100

  Hinweis: Diese Fallstudie ist nur notwendig, wenn die Überwachung innerhalb der Geschäftszeiten liegt (im Zeitraum der Metrik)
- **Ziel**: 99,6 %

Zusätzlich zu den vorhergehenden Metrikinformationen können die folgenden Elemente vom Systemservicekatalog auch aus der obigen Beschreibung abgeleitet werden:

- **Service**: System X.
- **Servicedomäne**: Verfügbarkeit.
- **Domänenkategorie**: % der Zeit verfügbar.
- **Servicegruppe**: Eine Gruppe, die mehr als ein System identifiziert, welche sie überwachen kann. Zu diesem Zeitpunkt ist es schwierig, festzustellen, ob eine geeignete Gruppe erstellt werden kann.

Nun können Sie das Diagramm vom Vertragsbildungsabschnitt dieses Dokuments reproduzieren, das diese Elemente in einem Diagramm anzeigt.



Fallstudie 2: Systemverfügbarkeit 2

Die CNP-Systemverfügbarkeit sollte die folgenden Ebenen verwalten:

Umgebung	Wochentage	Wochenenden
Produktion	99,9 %	98,9 %
Entwicklung	90 %	NV
Testen/Qualitätssicherung	Keine Garantie	NV
Netzwerk	99,9 %	NV

Vorgeschlagene Lösungen:

Metrik	**CNP-System-Produktionsdurchschnitt an Wochentagen**
Ziel	99,9
Kontrollzeitraum	1 Monat
Maßeinheit	%
Service	CNP-System-Produktion
Servicedomäne	Verfügbarkeit
Domänenkategorie	Anwendungsverfügbarkeit
Zeitfenster	Wochentage
Metrik	**CNP-System-Produktionsdurchschnitt an Wochenenden**
Ziel	98,9
Kontrollzeitraum	1 Monat
Maßeinheit	%
Service	CNP-System-Produktion
Servicedomäne	Verfügbarkeit
Domänenkategorie	Anwendungsverfügbarkeit
Zeitfenster	Wochenenden
Metrik	**CNP-System-Entwicklungsdurchschnitt an Wochentagen**
Ziel	90
Kontrollzeitraum	1 Monat
Maßeinheit	%
Service	CNP-Systementwicklung
Servicedomäne	Verfügbarkeit
Domänenkategorie	Anwendungsverfügbarkeit
Zeitfenster	Wochentage
Metrik	**CNP-System-Test-/Qualitätssicherungsdurchschnitt an Wochentagen**
Ziel	Kein(e)
Kontrollzeitraum	1 Monat

Maßeinheit	%
Service	CNP-Systemtests Q/A
Servicedomäne	Verfügbarkeit
Domänenkategorie	Anwendungsverfügbarkeit
Zeitfenster	Wochentage
Metrik	**CNP-System-Netzwerkverfügbarkeit**
Ziel	99,9
Kontrollzeitraum	1 Monat
Maßeinheit	%
Service	CNP-Systemnetzwerk
Servicedomäne	Verfügbarkeit
Domänenkategorie	Netzwerkverfügbarkeit
Zeitfenster	Immer

Fallstudie 3: System-Reaktionszeit

Folgende Fallstudie stellt die Antwortzeit von Metriken dar. Ein Vertrag kann auf unterschiedliche Art und Weisen gebildet werden, jede mit seinen Vorteilen.

Das folgende Beispiel prüft verschiedene Bildungsmethoden:

Vorgeschlagene Modellierung, Lösung A	**Maximalwert**
CNP-System-Transaktionsantwortzeit	Kann 750 Millisekunden pro Monat nicht überschreiten
Name der Metrik	Maximale Transaktionsantwortzeit
Ziel	750
Kontrollzeitraum	1 Monat
Maßeinheit	Millisekunden
Zeitfenster	Immer
Service	CNP-System
Servicedomäne	Leistung
Domänenkategorie	Maximale Transaktionsantwortzeit

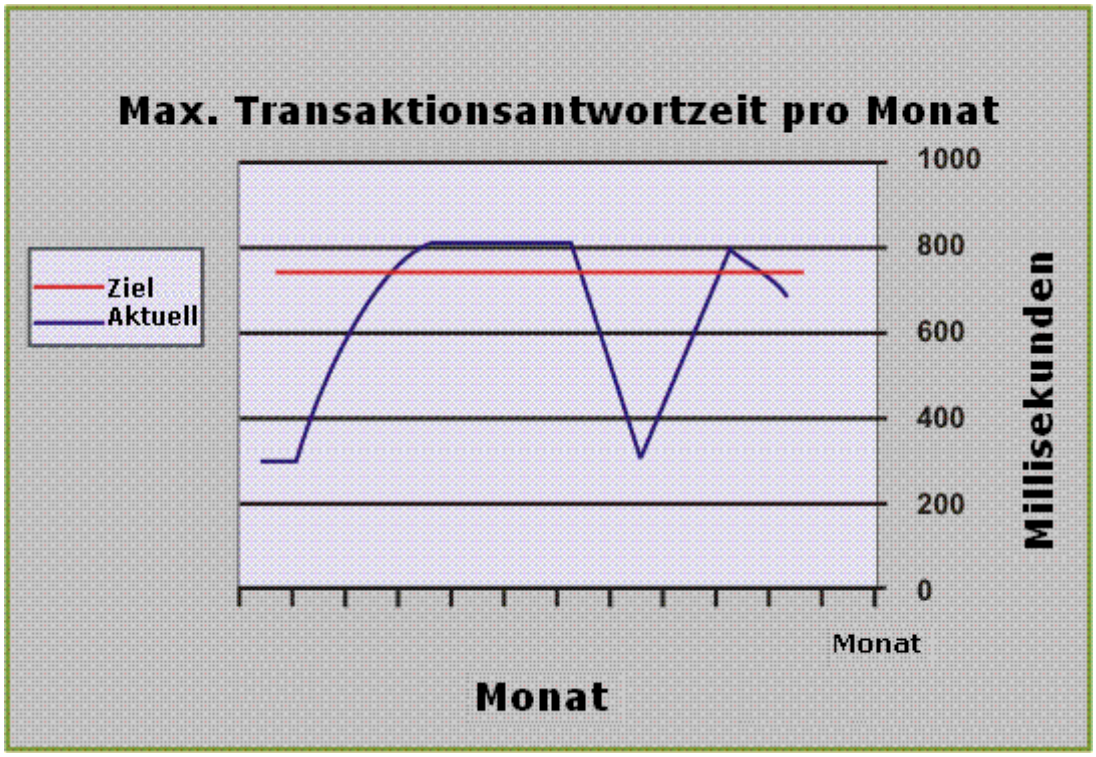
Wie wird das aktuelle Service Level basierend auf die obige Matrix berechnet?

Basierend auf die Definition der Domänenkategorie, scheint es, dass das eigentliche Service Level als Maximalwert berechnet wird. Dies bedeutet, dass die Transaktion mit dem Maximalwert für alle während eines Monats ausgeführten Transaktionen aufgezeichnet und dieser Wert mit dem Ziel verglichen wird.

Hinweis: Die Berechnung des Service Levels basiert auf eine Aggregation von Rohdaten über einen bestimmten Zeitraum. Für jeden Zeitraum gibt die Metrik ein einzelnes Ergebnis aus. Das Ziel einer Metrik wird mit nicht mit einer einzelnen Transaktion, sondern mit einem monatlichen Ergebnis verglichen, das eine periodische Aggregation aller Transaktionen innerhalb dieses Zeitraums ist. Der Vertragsmanager muss sicherstellen, dass dieses Ergebnis den Vertrag auf der einen Seite und die Servicequalität auf der anderen widerspiegelt.

Beachten Sie, dass die gemessene Antwortzeit, wie ein Maximalwert, eine sehr strikte Verpflichtung ist und sehr schwierig in der Praxis zu erreichen sein wird. Bemessung an einem Höchstwert bedeutet, dass eine einzelne Transaktion von 751 ms aus einer Million, während des Ablaufs von einem Monat, durchgeführten Transaktionen lang genug ist, einen Vertragsbruch zu verursachen. Alle Balken in den Berichten werden deshalb rot sein und nicht den wirklichen Stand des Service wiedergeben, der angegeben wurde.

Die folgende Abbildung stellt einen typischen Bericht unter diesen Umständen dar.



Eine Transaktion, die das Ziel überschreitet, wird als ein Vertragsbruch betrachtet, aber als Basis dafür, zu verstehen, dass die eigentliche Servicequalität besorgniserregend ist, da es nur eine einzelne Transaktion widerspiegelt und nichts im Bezug auf den Rest der Transaktionen bekannt ist, so wie: War es ein einzelner Fehler oder ein Trend? Wenn es kein Einzelfall ist, wie viele Fehler waren dann vorhanden, oder wie ist das Verhältnis von fehlerhaften Transaktionen zur Gesamtzahl der innerhalb des Monats ausgeführten Transaktionen? Es kann eine Reihe von Monaten mit solchen Vorkommnissen geben und daher ein Vertragsbruch sein, aber wie ist der Trend? Verbessert er sich oder wird er schlechter? All das sind Fragen, die der Service Level-Manager fragen könnte und die der Bericht beantworten können sollte.

Hinweis: Bei der Definition der Metrik und ihren zugeordneten Berechnungsentwurf, ist es sehr wichtig, sich vorzustellen, wie die Ergebnisse in einem Bericht angezeigt werden. Dieser Bericht muss zwei entscheidende Elemente angeben:

- Hat es hat einen Vertragsbruch gegeben?
- Er muss Managern genug Datentransparenz zur Verfügung stellen und ihnen die Fähigkeit verschaffen, die Ursache des Fehlers zu untersuchen und auch Managern die notwendigen Tools bereitstellen, um dessen Servicekomponenten vollständig verstehen zu können.

Vorgeschlagene Modellierung, Lösung B	**Durchschnittliche Antwortzeit**
CNP-System-Transaktionsantwortzeit	Darf nicht mehr als 750 Millisekunden pro Monat sein
Metrik	Durchschnittliche Transaktionsantwortzeit
Ziel	750
Kontrollzeitraum	1 Monat
Maßeinheit	Millisekunden
Domänenkategorie	Durchschnittliche Transaktionsantwortzeit

Die Berechnung der durchschnittlichen Antwortzeit gibt eine bessere Vorstellung der monatlichen Servicequalität, und dennoch können gleichzeitig noch jene Monate mit extremen oder Außerhalb-von-Vertrag-Antwortzeiten wiedergeben.

Vorgeschlagene Modellierung, Lösung C	**Prozentsatz der Transaktionen, die erfolgreich unter dem Grenzwert abgeschlossen wurden.**
CNP-System-Transaktionsantwortzeit	Darf nicht mehr als 750 Millisekunden pro Monat sein
Metrik	Erfolgreiche Transaktionsantwortzeit
Ziel	100
Kontrollzeitraum	1 Monat
Maßeinheit	% Erfolgreich
Metrikparameter	750 ms

Service	CNP-System
Servicedomäne	Leistung
Domänenkategorie	Erfolgreiche Transaktionsantwortzeit
Zeitfenster	Immer

Bei der Verwendung dieser Methode wird bei der Berechnung der Prozentsatz der Transaktionen, die erfolgreich unter dem Grenzwert von 750 ms während des angegebenen Zeitraums abgeschlossen wurden, von dieser Formel errechnet:

*((Anzahl von Transaktionen unter 750 ms.)/(Gesamtanzahl der Transaktionen))*100*

Das Ausdrücken einer Garantie als Erfolgsrate ermöglicht es, eine strikte Garantie beizubehalten (Ziel 100 %), wobei der tatsächliche Istwert darstellt, wie gut oder schlecht der Service war.

Mit dieser Methode ist das Ziel zwar nicht die Obergrenze von 750 ms, aber das beizubehaltende Verhältnis. In Fällen, wo die Garantie streng ausgelegt werden muss, sollte das Ziel dann 100 % sein, welches dann keinen Platz für auch nur einen einzelnen Fehler zulässt. Beachten Sie, dass eine zusätzliche Variable in diesem Fall eingeführt wurde, der Metrikparameter. Dieser Parameter sollte als ein Metrikparameter implementiert werden, um einfache Anpassungen bei Bedarf zu aktivieren.

Ein alternatives Modell zu dieser Methode kann die Erzwingung eines Eskalationstypenmodells sein:

Die folgenden Lösungen definieren drei Metriken anstelle einer einzelnen Metrik, wie in den bisherigen Lösungen.

Metrik	Erfolgreiche Transaktionsantwortzeit
Ziel	95
Kontrollzeitraum	1 Monat
Maßeinheit	% Erfolgreich
Metrikparameter	750 ms
Metrik	Erfolgreiche Transaktionsantwortzeit
Ziel	99
Kontrollzeitraum	1 Monat
Maßeinheit	% Erfolgreich
Metrikparameter	850 ms
Metrik	Erfolgreiche Transaktionsantwortzeit
Ziel	100

Kontrollzeitraum	1 Monat
Maßeinheit	% Erfolgreich
Metrikparameter	1000 ms

In einem Fall, wo es notwendig ist, über die vertragliche Verpflichtung ebenso wie über die Anzahl an Transaktionen zu berichten, die den Grenzwert von 750 ms überschreiten, müssen Sie eine zusätzliche Metrik angeben, um die Anzahl von fehlgeschlagenen Transaktionen zu zählen.

Hinweis: Jede Metrik generiert ein einzelnes Ergebnis über einen bestimmten Zeitraum. Wenn sie festgelegt wird, um den Prozentsatz von Transaktionen zu berechnen, kann sie nicht auch die Anzahl der Transaktionen angeben.

Sie können zusätzliche Berichte nur aus einer Metrik erzeugen, indem die Ausgaben der Business-Logik verwendet werden. (Siehe Outputs - User Tables (siehe Seite 160), welche die Ausgabeergebnisse der Business-Logik behandelt).

Metrik	**Anzahl fehlgeschlagener Transaktionen**
Ziel	Kein Ziel
Kontrollzeitraum	1 Monat
Maßeinheit	Anzahl an Transaktionen
Metrikparameter	750 ms
Service	CNP-System
Servicedomäne	Leistung
Domänenkategorie	Anzahl an Transaktionen
Zeitfenster	Immer

Fallstudie 4: Helpdesk-Leistung

Eine Helpdesk-Situation veranschaulichende Fallstudie

Der Helpdesk muss 100 % Erfolg beim Erreichen der folgenden Ziele haben:

Ticket-Typ	**Lösungszeit**
Priorität 1	1 Stunde
Priorität 2	2 Stunden
Priorität 3	4 Stunden

Vorgeschlagene Modellierung, Lösung A:

Metrik	Lösungszeit für Priorität 1
Ziel	100
Kontrollzeitraum	1 Monat
Maßeinheit	% Erfolgreich
Vertragsparameter	Lösungszeit-Matrix
Service	Helpdesk
Servicedomäne	Helpdesk-Leistung
Domänenkategorie	Problemlösungszeit
Zeitfenster	Immer

Die obige Matrix bezieht sich auf drei Metriken. Für jede Priorität wird eine gesonderte Metrik mit allen Prioritäten innerhalb der gleichen Kategorien angegeben.

Vorgeschlagene Modellierung, Lösung B:

Die Metrik-Definition bleibt die gleiche wie in Lösung A angezeigt.

Option 1:

Service	Helpdesk
Servicedomäne	Ticket-Verwaltungspriorität 3
Domänenkategorie	Problemlösungszeit
Domänenkategorie	Problemreaktionszeit
Zeitfenster	Immer

Option 2:

Service	Helpdesk
Servicedomäne	Ticket-Verwaltung
Domänenkategorie	Priorität 3, Ticket-Lösungszeit
Domänenkategorie	Problemreaktionszeit
Zeitfenster	Immer

Fallstudie 5: Systemsicherung

Eine Sicherung wird folgendermaßen ausgeführt:

Zeitraum	Anzahl an Sicherungen
Wöchentlich	6
Monatlich	27
Jährlich	350

Vorgeschlagene Lösungen:

Metrik	Wöchentliche Sicherungsanzahl
Ziel	6
Kontrollzeitraum	1 Woche
Maßeinheit	Sicherungen
Service	Sicherung
Servicedomäne	Sicherung
Domänenkategorie	Anzahl von Sicherungen pro Woche
Zeitfenster	Immer
Metrik	**Monatliche Sicherungsanzahl**
Ziel	27
Kontrollzeitraum	1 Monat
Maßeinheit	Sicherungen
Service	Sicherung
Servicedomäne	Sicherung
Domänenkategorie	Anzahl von Sicherungen pro Woche
Zeitfenster	Immer
Metrik	**Jährliche Sicherungsanzahl**
Ziel	350
Kontrollzeitraum	1 Jahr
Maßeinheit	Sicherungen
Service	Sicherung
Servicedomäne	Sicherung

Domänenkategorie	Anzahl von Sicherungen pro Woche
Zeitfenster	Immer

Beispiel für eine Finanzmetrikmodellierung

Folgende Fallstudie präsentiert ein Beispiel von Finanzmodellierung.

Fallstudie 6: Modellieren von Finanzbedingungen eines Vertrags/Services

Es gibt drei allgemeine Metriktypen, die verwendet werden, um die Finanzbedingungen eines Service oder Vertrags zu modellieren. Diese sind:

- Festpreis-Kosten
- Verbrauchskosten
- Vertragsstrafe/Bonus-Gebühren

Berücksichtigen Sie das folgende Beispiel:

Ein neues Unternehmen wird gegründet und benötigt einen E-Mail-Service, der ihm neben der Einrichtung und Verwaltung seiner Postfächer bereitgestellt wird. Mit jedem neuen Mitarbeiter steigt offenkundig auch die Anzahl der Postfächer. Für die Bereitstellung eines E-Mail-Services für einen Vertrag fallen Festpreiskosten von $1 000 an. Darüber hinaus entstehen weitere Kosten für jedes Postfach, die monatlich abgerechnet werden. Diese Kosten pro Postfach berechnen sich nach einem gestuften Preismodell wie folgt:

Anzahl von Postfächern	**Kosten pro Postfach**
1-1 000	$1,00
1 001 - 5 000	$0,80
5 001+	$0,50

Je mehr Postfächer hinzugefügt werden, desto niedriger sind die zusätzlichen Kosten. (Zum Beispiel: 1 500 Postfächer kosten (1 000 x $1) + (500 x $0,80) = $1 400.) Bei der Verwendung dieses Modells können zwei Metriken erzeugt werden, um dies im Vertrag zu reflektieren.

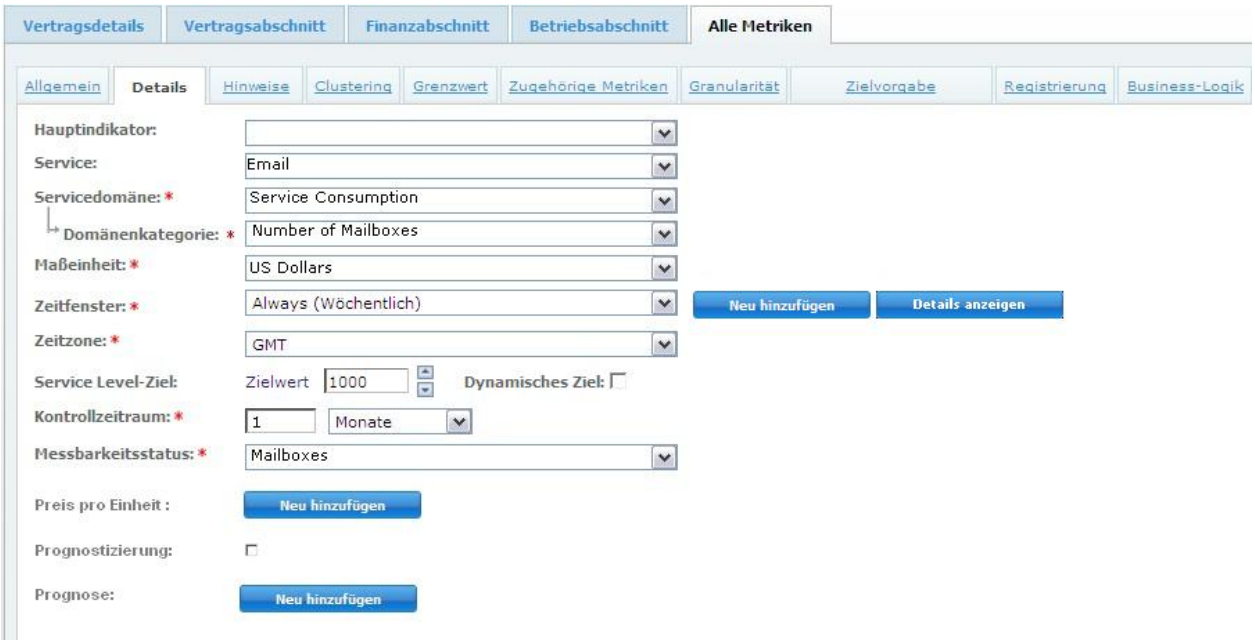
- E-Mail-Service-Kosten (fest)
- Kosten für die Postfachverwendung (variable, basierend auf dem Verbrauch)

Darüber hinaus schätzt das Verwaltungsteam die Anzahl der Mitarbeiter über das Jahr gesehen (2007) wie nachfolgend angezeigt. Der Trend ergibt sich aus dem Anfangswachstum des Unternehmens infolge von steigender Beschäftigung und dann durch die Eröffnung neuer Büros in anderen Regionen:

Jan.	Feb.	Mär.	Apr.	Mai	Jun.	Jul.	Aug.	Sept.	Okt.	Nov.	Dez.
50	100	500	900	1600	1700	1800	2500	2600	3500	3600	5800

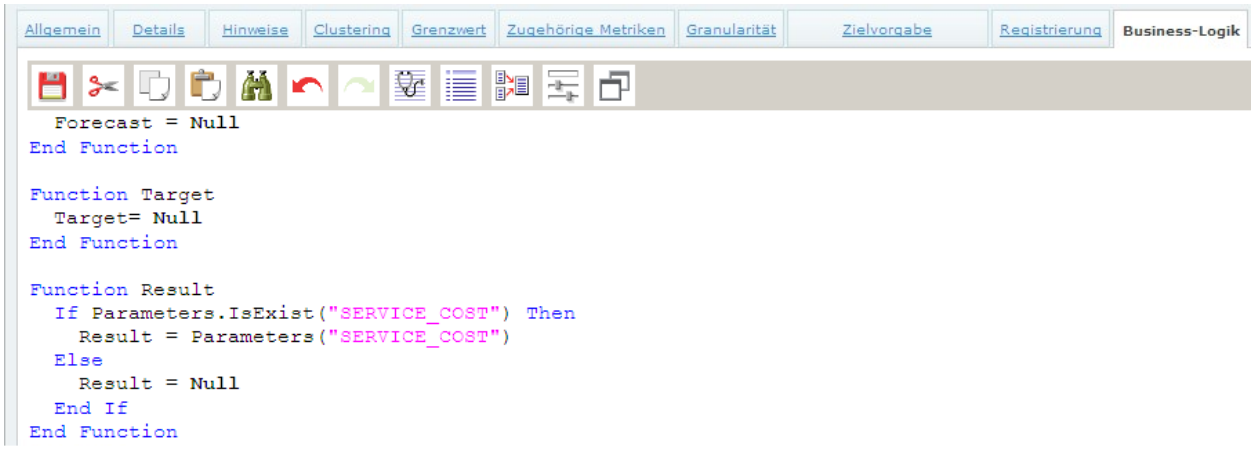
Um diese Metriken zu modellieren, gehen Sie wie folgt vor:

Erstellen Sie die Fixkosten-Metrik (verwenden Sie den Preiselement-Typ) im Vertrag, indem Sie folgende Details verwenden:

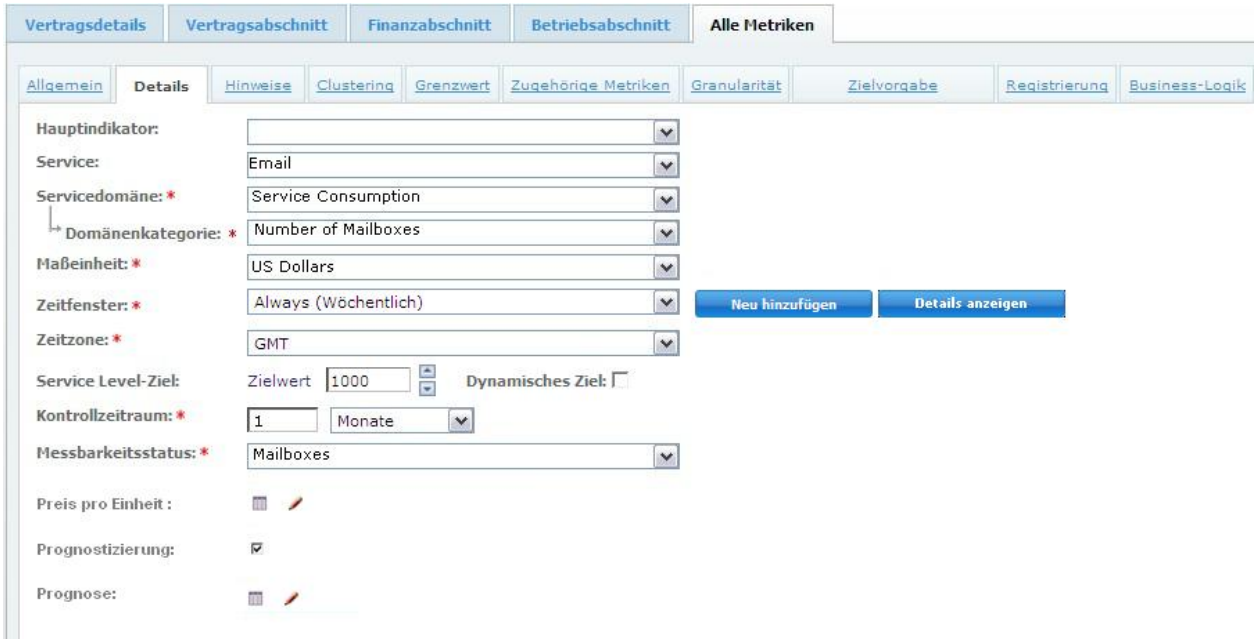


Um die Fixkosten für diesen Service im Vertrag festzulegen, implementieren Sie dies als Parameter in der Business-Logik (wo die Fixkosten von der Ergebnisfunktion zurückgegeben werden müssen). Dieser Parameter kann dann über die Zielvorgabe der Metrik, wie unten dargestellt, angezeigt werden:

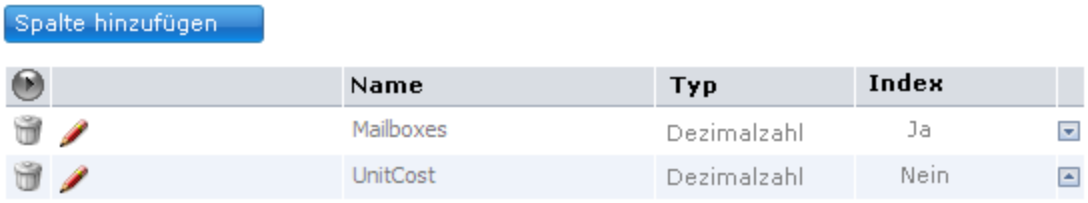
Die Rückgabe des Parameterwerts für diese Metrik erfolgt einfach über die Rückgabe des Werts der Servicekosten über die Ergebnisfunktion.



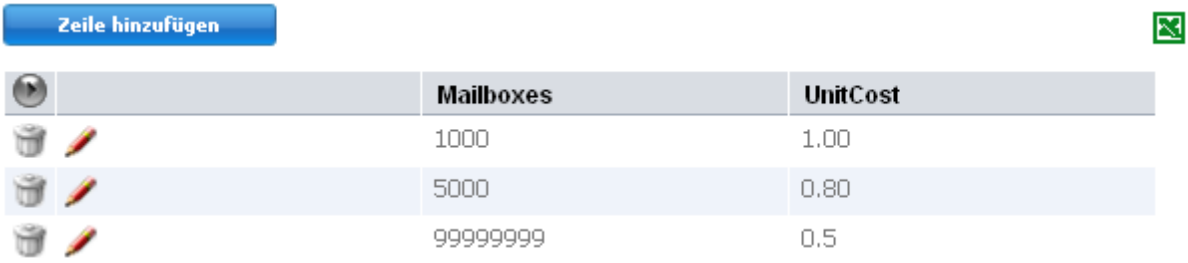
Erstellen Sie als Nächstes die variable Preisgestaltungs-Metrik (verwenden Sie erneut den Preiselement-Typ), um die Verbrauchskosten für die Anzahl der verwendeten Postfächer zu bestimmen. Nennen Sie diese Metrik '*Postfach-Verbrauchskosten*' und erstellen Sie sie mit den folgenden Details:



In diesem Fall müssen Sie die Verbrauchsparameter in die Metrik-Details eingeben. Diese werden in die Einheitspreis-Tabelle übertragen. Um die obige Tabelle für die Anzahl von Postfächern in Bezug auf die Kosten zu modellieren, erstellen Sie eine Spalte für den oberen Grenzwert für die Postfächer und eine Spalte für die Einheitspreise:

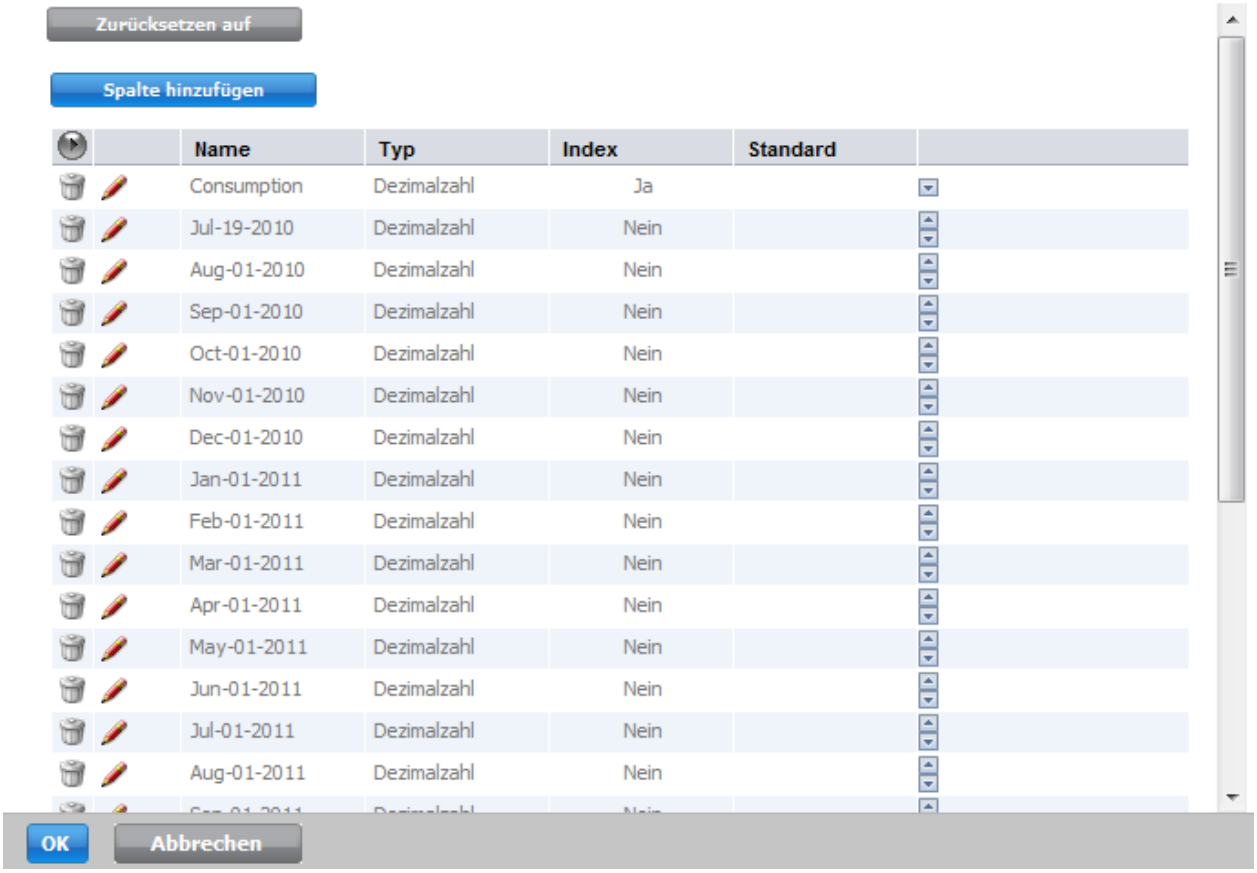


Geben Sie dann die Werte für jede Stufe ein. In diesem Fall bestimmt der obere Grenzwert für die Postfächer die mit ihm verknüpfte Kostenklammer. Da es 3 Stufen gibt, werden sie wie folgt zur Tabelle hinzugefügt:



	Mailboxes	UnitCost
	1000	1.00
	5000	0.80
	99999999	0.5

Implementieren Sie zusätzlich dazu die Prognosefunktion für den Verbrauch von Postfächern. Führen Sie dies aus, indem Sie die Tabelle "Prognose" mit dem voreingestellten monatlichen Layout erstellen.



	Name	Typ	Index	Standard
	Consumption	Dezimalzahl	Ja	
	Jul-19-2010	Dezimalzahl	Nein	
	Aug-01-2010	Dezimalzahl	Nein	
	Sep-01-2010	Dezimalzahl	Nein	
	Oct-01-2010	Dezimalzahl	Nein	
	Nov-01-2010	Dezimalzahl	Nein	
	Dec-01-2010	Dezimalzahl	Nein	
	Jan-01-2011	Dezimalzahl	Nein	
	Feb-01-2011	Dezimalzahl	Nein	
	Mar-01-2011	Dezimalzahl	Nein	
	Apr-01-2011	Dezimalzahl	Nein	
	May-01-2011	Dezimalzahl	Nein	
	Jun-01-2011	Dezimalzahl	Nein	
	Jul-01-2011	Dezimalzahl	Nein	
	Aug-01-2011	Dezimalzahl	Nein	

Diese wird dann mit den Werten der Tabellen gefüllt, die in der Szenariobeschreibung angegeben sind.

Verbrauch	Jan-01-2007	Feb-01-2007	Mär-01-2007	Apr-01-2007	Mai-01-2007	Jun-01-2007	Jul-01-2007	Aug-01-2007	Sep-01-2007	Okt-01-2007	Nov-01-2007	Dez-01-2007
1	50	100	500	900	1600	1700	1800	2500	2600	3500	3600	5800

Jetzt können Sie die Zielvorgabe für die Metrik hinzufügen. In diesem Fall werden keine Parameterwerte benötigt, da sie von den Einheitspreis- und Prognose-Tabellen abgeleitet werden.

Beenden Sie schließlich die Business-Logik folgendermaßen:

```
Option Explicit

Dim PPUmap1, PPUmap2, PPUmap3, PPUkey, FCmap, periodFC, TierPPU
Dim currentMonth, TotalMailboxes, MailboxesThisPeriod, TotalPrice

Sub OnRegistration(dispatcher)
   'sample registration only
   dispatcher.RegisterByMetric "OnMailboxAddedEvent", "NewMailboxEventType", _
      "MailboxResource", "MONTH", "MetricName", "MetricContract", _
      "MetricContractParty"
End Sub

Sub OnLoad(TIME)
   'Initialise the price tier maps and forecast maps
   Set PPUmap1 = Kontext.Field ("Price Per Unit")(1)
   Set PPUmap2 = Kontext.Field ("Price Per Unit")(2)
   Set PPUmap3 = Kontext.Field ("Price Per Unit")(3)
   Set FCmap = Kontext.Field ("Forecast")(1)
End Sub

Sub OnPeriodStart(TIME)
   'TODO: ADD code here TO handle period START event
   currentMonth = GetMonth (time)
   If Context.IsInForecast Then
      periodFC = getForecastValue (currentMonth)
   End If
   MailboxesThisPeriod = 0
   TotalPrice = 0
End Sub

Sub OnPeriodEnd(TIME, isComplete)
   ' Calculate the current price of all the mailboxes using the tiered
   ' pricing model
   ' This uses a cumulative approach as it goes through each tier to
   ' determine total cost.
   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
```

```
Function Forecast
   Forecast = getMailboxCost (periodFC)
End Function

Function Target
   Target = Null
End Function

Function Result
   result = TotalPrice
End Function

Function getforecastvalue(q)
   getforecastvalue = FCmap (q)
End Function

Function getmonth(time)
   'this function retrieves the month
   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)
   'Function calculates the cost of the mailboxes using the tiered pricing model
   Dim returnValue
   If num_boxes <= PPUmap1 ("Mailboxes") Then
      'First tier
      returnValue = num_boxes * PPUmap1 ("UnitCost")
      'Out.Log "Tier1: " & num_boxes
   Else If num_boxes > PPUmap1 ("Mailboxes") And num_boxes <= PPUmap2
("Mailboxes") Then
      'second tier only
      returnValue = (PPUmap1 ("Mailboxes") * PPUmap1 ("UnitCost")) + _
         ((num_boxes - PPUmap1 ("Mailboxes")) * PPUmap2 ("UnitCost"))
      'Out.Log "Tier2: " & num_boxes
   Else If num_boxes > PPUmap2 ("Mailboxes") Then
      'third tier
      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
```

Hinweis: Dieses Business-Logik-Skript handhabt sowohl die Prognoseberechnung (mithilfe der "Prognose"-Tabelle) als auch die Ergebnisse für die Finanz-Verbrauchskosten. Für beides wird dieselbe Formel getMailboxCost() verwendet. Die Formel berechnet die gestuften Preise basierend auf der "Einheitspreis"-Tabelle, die für diese Metrik festgelegt wurde.

Beispiele für die Datenmodellierung

Die folgenden zwei Fälle veranschaulichen den grundlegenden Prozess der Datenmodellierung sowie einige Feinheiten, die möglicherweise auftreten können.

Da der Prozess der Datenmodellierung nach dem Prozess der Vertragsmodellierung erfolgt, sind die von den Garantien des Vertrags abgeleiteten Berechnungsanforderungen bereits bekannt, identifiziert und als Teil des Prozesses der Vertragsmodellierung festgelegt worden.

Das Datenmodell muss alle Berechnungsanforderungen berücksichtigen.

In den folgenden zwei Fallstudien wird eine einzelne Anforderung bzw. werden einige bestimmte Anforderungen ausgewählt, um den Prozess der Datenmodellierung zu veranschaulichen.

Fallstudie 7: Serverleistung

Dies ist eine typische Fallstudie zur Serverleistung.

Die folgende Datenquellenstruktur ist vorgegeben:

Anzeige	Server	Maß	Timestamp
Verfügbarkeit	Appserv01	1	03.01.2001 07:44
Antwortzeit	Appserv01	354,6	03.01.2001 09:58
CPU-Auslastung	Dbserv02	83 %	03.01.2001 12:12
Verfügbarkeit	Appserv01	0	03.01.2001 14:26
CPU-Auslastung	Dbserv02	94,30 %	03.01.2001 16:40
Kapazität	Firewall01	10 %	03.01.2001 18:54
Antwortzeit	Dbserv02	476,89	03.01.2001 21:08
Verfügbarkeit	Appserv02	1	03.01.2001 21:24
Antwortzeit	Appserv01	774,4	03.01.2001 21:48
CPU-Auslastung	Dbserv01	83 %	03.01.2001 21:52

Neben dem oben genannten gibt es folgende Berechnungsanforderungen:

Berechnen Sie die %-Verfügbarkeit von jedem Anwendungsserver.

Die Verfügbarkeit von jedem Server sollte gesondert berechnet werden. Um die Verfügbarkeit eines einzelnen Servers zu berechnen, ist es notwendig, nur Events für diesen bestimmten Server zu empfangen. Darüber hinaus enthalten die Datenquellen andere Leistungsindikatoren, die nicht relevant für die Verfügbarkeitsberechnungen sind (Antwortzeit, Kapazität usw.). Daher müssen die Verfügbarkeitsindikatoren sowie der entsprechende Server gefiltert werden.

Da es sich bei den Filterkriterien in CA Business Service Insight um den Event-Typ und die Ressourcen handeln, müssen Sie die Filterkriterien von den Datenquellwerten in eine Ressourcen- und Event-Typ-Definition übersetzen.

In diesem Fall ist der Indikator ein idealer Wert, um ihn in CA Business Service Insight als Event-Typ zu übersetzen, da er den Typ des Events logisch beschreibt. Es gibt eine begrenzte Anzahl von Typen, wie z. B. Verfügbarkeit, Antwortzeit, Kapazität und CPU-Auslastung. Dies bedeutet, dass die Registrierung bei den Metriken, die die Verfügbarkeit des Servers berechnen, für den Verfügbarkeits-Event-Typ erfolgt.

Wenn es in diesem Fall eine Vielzahl von Servern gibt und es notwendig ist, die Verfügbarkeit von jedem Server zu berechnen, folgt daraus, dass jeder Server als eine Ressource definiert werden muss. Sie muss dann in einer Ressourcengruppe zusammengefasst werden und die Metrik wird über diese Ressourcengruppe geclustert.

Vorgeschlagene Modellierung:

Event-Name	Verfügbarkeits-Event.
Verhalten	Als Änderung des Status von 0 oder 1 angegeben.
Zeitstempelfeld	Zeitstempel (das einzige Zeitfeld in der Datenquelle).
Ressourcenfeld	Server (jeder Server, der in der Datenquelle vorhanden ist, wird in eine CA Business Service Insight-Ressource übersetzt).
Event-Typ-Feld	Indikator (jeder der Werte in diesem Feld wird in CA Business Service Insight in ein Event-Typ übersetzt. Es gibt vier Event-Typen).
Datenfelder	Messwert ist 0 oder 1 (nur für Verfügbarkeitsdatensätze).

Die folgenden Ressourcenzuordnungen sollten festgelegt werden:

Ressourcentyp-Attribut	Anwendungsserver
Zuordnung zu Vertragspartei	Jeder Anwendungsserver wird der Vertragspartei zugewiesen, bei der der entsprechende Server seine Anwendung ausführt. Dies ermöglicht eine Registrierung über die Vertragspartei, die dementsprechend alle Server abruft.
Zuordnung zu Service	Siehe oben.
Zuordnung zu Ressourcengruppe	Optional. Es ist üblicherweise notwendig, Ressourcen zu gruppieren, für die ein Clustering erforderlich ist.

Und schließlich basierend auf allen oben genannten Definitionen:

Registrierung über

Bei der geclusterten Metrik, die die Verfügbarkeit von jedem Server einzeln berechnet, erfolgt die Registrierung über die Ressource.

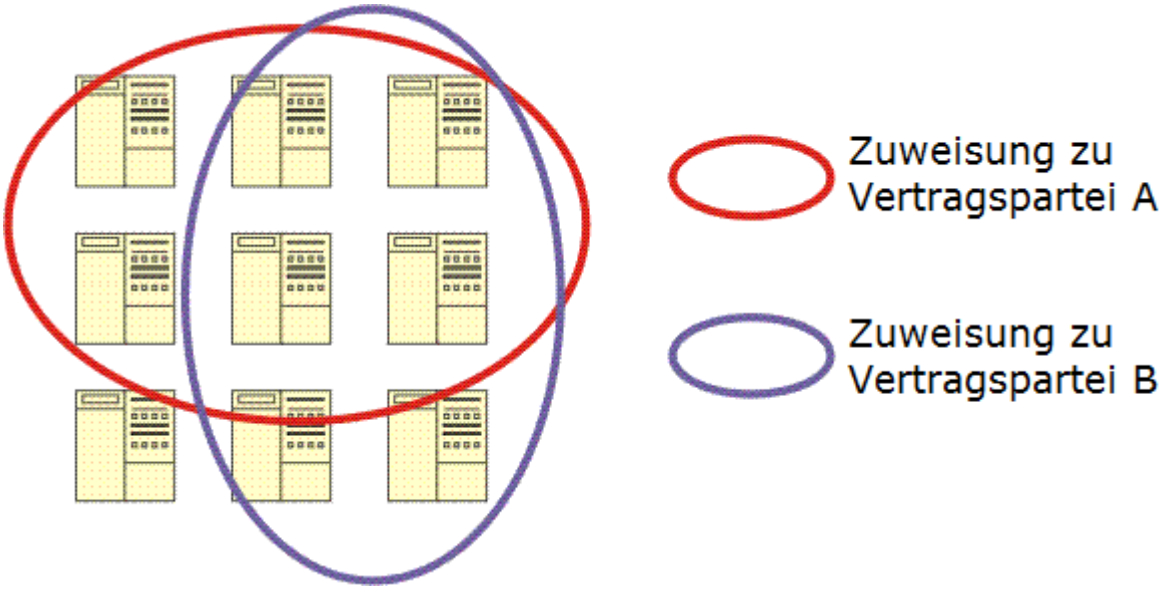
Um den obigen Anforderung entsprechen zu können, wird das folgende Kriterium hinzugefügt:

Berechnen Sie die durchschnittliche Antwortzeit der Anwendungsserver für jede Vertragspartei gesondert.

Für diese Anforderung ist es notwendig, Antwortzeiten-Events für alle Anwendungsserver zu empfangen, die Teil der Gruppe von Servern sind, die Anwendungen für die bestimmte Vertragspartei ausführen. Die Antwortzeit-Events werden empfangen, indem der Event-Typ registriert wird, der vom Angabefeld mit dem Wert "Antwortzeit" erstellt wurde. So wird gewährleistet, dass nur Events empfangen werden, die die Antwortzeiten der Server melden.

Um nur die Events zu empfangen, die über die relevanten Server einer bestimmten Vertragspartei berichten, müssen die Ressourcen durch die Vertragspartei-Zuordnung folgendermaßen registriert werden:

Eine Ressource kann mehr als einer einzelnen Vertragspartei, einem Service oder einer Gruppe zugewiesen werden. Daher kann es vorkommen, dass ein Event, das zur Berechnung der Antwortzeit als Teil der Vertrags mit Vertragspartei A gesendet wurde, auch Teil der Berechnungen für Vertragspartei B ist.



Fallstudie 8: Helpdesk-Leistung

Dies ist eine typische Fallstudie zur Helpdesk-Leistung.

Die nachfolgend gezeigte Datenquelle ist vorgegeben:

TK Nr.	TK-Priorität	Erzeugt am	Geschlossen am	Gelöst am	Kundenreferenz	Standort
3800968	1	19.12.2003 07:51	01.05.04 11:31	22.12.2003 12:00	CM3	London
38000509	1	18.12.2003 09:21	01.05.04 11:33	22.12.2003 12:00	CM1	Ipswitch
38084199	2	01.07.04 11:20	14.01.2004 09:10	01.09.04 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	12.12.03 14:04	01.05.04 11:35	19.12.2003 12:00	CM286	Birmingham
3796448	1	12.12.03 14:18	01.05.04 11:39	19.12.2003 12:00	CM263	Luton
37965039	2	12.12.03 14:57	14.01.2004 15:05	18.12.2003 12:00	CM264	Leeds
37970699	2	15.12.2003 09:26	01.05.04 11:22	22.12.2003 12:00	CM288	London
37997259	1	17.12.2003 15:58	01.05.04 11:27	23.12.2003 12:00	CM302	Ipswitch
38000259	1	18.12.2003 09:11	01.06.04 14:44	29.12.2003 12:00	CM340	London
38021049	1	22.12.2003 09:32	01.06.04 14:28	31.12.2003 12:00	CM341	London

Die nachfolgend gezeigte Datenquelle enthält Details für die Helpdesk-Tickets, die für jeden Kunden an zahlreichen von den Kunden betreuten Standorten bearbeitet werden. Ein einzelner Standort kann auch von mehreren Kunden gemeinsam genutzt werden.

Die folgenden drei Anforderungen werden dafür benötigt, zu berichten, wann diese Datenquelle verwendet wird:

- % der Tickets der Priorität 1, die innerhalb von vier Stunden für den Kunden CM3 gelöst werden.
- % der Tickets der Priorität 1, die innerhalb von vier Stunden für den Kunden CM3 an jedem Standort gelöst werden.
- % der Tickets der Priorität 1, die innerhalb eines Tages für den Kunden CM3 an jedem Standort gelöst werden.

Die obigen Anforderungen zeigen an, dass die Events folgendermaßen gefiltert werden sollten:

- Priorität
- Kunde
- Standort

Sie müssen angeben, welche dieser Filterkriterien als Event-Typ übersetzt wird und welches die dazugehörige Ressource sein soll.

Wie wähle ich einen Event-Typ aus?

Von den drei benötigten Filterkriterien ist die Ticketpriorität aus folgenden Gründen am besten geeignet, um in ein Event-Typ übersetzt zu werden:

- Sie beschreibt die Art des Events, das bearbeitet wird (Tickets der Priorität 1).
- Die Anzahl verschiedener Prioritäten, die vorhanden sind, ist relativ klein (Priorität 1, 2, 3).
- Der Event-Typ selbst ist relativ konstant (der Helpdesk verändert selten die Prioritäten, mit der er Tickets verarbeitet).

Wie wird eine Ressource ausgewählt?

Die zwei anderen erforderlichen Filterkriterien sind "Kunde" und "Standort". Daher ist die Kombination aus Kunde und Standort die Ressource.

Kunde und Standort sind relativ feste Entitäten und haben einen bestimmten Lebenszyklus, wobei neue Kunden oder neue Standorte hinzugefügt werden können. Darüber hinaus kann sich die Beziehung zwischen einem Standort und einem Kunden ändern.

Für Übersetzungszwecke ist es möglich, mehr als ein Feld von der Datenquelle zu verwenden. Während das Serverfeld in der vorherigen Fallstudie in eine CA Business Service Insight-Ressource übersetzt wurde, wird die Ressource in diesem Fall hingegen durch die Kombination von zwei Feldern gebildet. Jede Permutation erzeugt daher eine neue Ressource.

Die Ressourcen-Liste wird unten angezeigt:

Kundenfeld	Standortfeld	Ausgabe-Ressource
CM3	London	CM3@London
CM1	Ipswitch	CM1@Ipswitch
CM2	Ipswitch	CM2@Ipswitch
CM3	Leeds	CM3@Leeds
CM286	Birmingham	CM286@Birmingham
CM263	Luton	CM263@Luton
CM264	Leeds	CM264@Leeds
CM288	London	CM288@London
CM302	Ipswitch	CM302@Ipswitch
CM340	London	CM340@London
CM341	London	CM341@London

Die Ausgabe-Ressource ist eine Kombination aus Kunden- und Standortfeld und das Symbol "+" wird verwenden, um diese beiden Felder zu verbinden. Es ist wichtig, den Namen der Ressource zu kennen, da er aus der Datenquelle extrahiert und später in den Berichten angezeigt wird. Die Berichterstellungsfähigkeiten müssen den Erwartungen entsprechen.

Hinweis: Einer der häufigsten Fehler beim Modellieren eines Helpdesk-, Ticket- oder Incident-Systems ist, einen einzelnen Incident als Ressource zu definieren.

Die folgenden falschen Annahmen führen häufig zu Fehlern:

1. Der einzelne Incident ist derjenige, der im Bericht angezeigt wird. Legen Sie nicht die Entität, die für die Berichterstellung verwendet wird, als die Entität fest, für die die Berechnungen erstellt werden, d. h. der einzelne Incident sollte nicht die Grundlage für die Berechnungen des Kundenstandorts bilden. Im Allgemeinen basiert der SLA auf der Gesamtleistung und nicht auf der Bearbeitungsleistung einzelner Tickets.
2. Die Garantie erfolgt nach Ticketlevel. Dies ist ein Fehler, weil die Garantien periodisch sind. Es wird eine Aggregation über die Zeit berechnet.

Einstellen von Zuordnungen für die Ressourcen

Für die erste Berechnungsanforderung:

1. % der Tickets der Priorität 1, die innerhalb von vier Stunden für den Kunden CM3 gelöst werden.
   - Ticket-Events, die einem bestimmten Kunden zuzuschreiben sind, müssen empfangen werden. Der Kunde ist Teil dieser Ressource, die auch den Kundenstandort anzeigt. Es gibt daher zwei Optionen, um alle Events, die sich auf die Ressourcen beziehen und diesem Kunden zuzuschreiben sind, zu erfassen:
   - In Fällen, in denen der Kunde in der Datenquelle eine Vertragspartei repräsentiert, können die Ressourcen der zugehörigen Vertragspartei angehängt werden. Dies erlaubt eine Registrierung über die Vertragspartei. Dies ist immer, wo es möglich ist, vorzuziehen.
   - Erstellen Sie eine Ressourcengruppe für jeden der Kunden in der Datenquelle und gruppieren Sie alle zugehörigen Ressourcen innerhalb dieser Gruppe. Bei der Verwendung dieser Methode erfolgt die Registrierung über die Ressourcengruppe.

   Für die nächsten beiden Berechnungsanforderungen:
2. % der Tickets der Priorität 1, die innerhalb von vier Stunden für den Kunden CM3 an jedem Standort gelöst werden.
3. % der Tickets der Priorität 1, die innerhalb eines Tages für den Kunden CM3 an jedem Standort gelöst werden.

Sammeln Sie die Ressourcen für diese Anforderungen in Ressourcengruppen, da die Metriken geclustert werden müssen - vorausgesetzt, dass die Berechnung für jeden Standort individuell ausgeführt werden muss.

Hinweis: Auch wenn die Zuordnungen der Ressourcen für das aktuelle Modell und die aktuellen Anforderungen nicht notwendig sind, ist es wichtig, sie zu erstellen, um künftige Anforderungen zu berücksichtigen. Dies verhindert einen Overhead bei der zukünftigen Systementwicklung.

Auswählen des Zeitstempelfelds

Wie zuvor erwähnt ist der Zeitstempel sehr wichtig für die Art und Weise, wie die Korrelations-Engine die Events verarbeitet. Metriken berechnen das Service Level immer für einen Zeitraum, und die Events, die in diesem Zeitraum verarbeitet werden, sind diejenigen, deren Zeitstempel in diesen Zeitraum fallen.

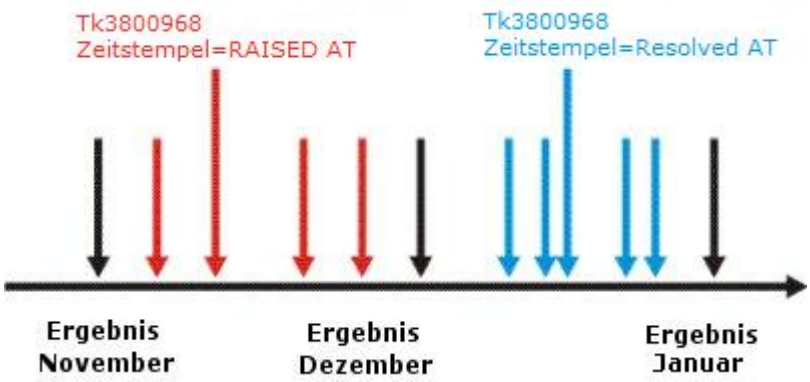
In der ersten Fallstudie hat die Datenquelle nur ein Zeitfeld. In diesem Fall jedoch gibt es drei mögliche Felder, die als Zeitstempel festgelegt werden können. Prüfen Sie die beiden ersten Datensätze:

TK Nr.	TK-Priorität	Erzeugt am	Geschlossen am	Gelöst am	Kundenreferenz	Standort
3800968	1	19.12.2003 07:51	01.05.04 11:31	22.12.2003 12:00	CM3	London
38000509	1	18.12.2003 09:21	01.05.04 11:33	22.12.2003 12:00	CM1	Ipswitch

Um die Lösungszeit berechnen zu können, ist sowohl die Zeit in "Erzeugt am" als auch die Zeit in "Gelöst am" erforderlich. Für Event-Zwecke ist es möglich, nur einen Zeitstempel anzuhängen. Dann kann der andere als ein Wert innerhalb der Wertfelder verwendet werden.

Wenn der Wert in "Erzeugt am" als Zeitstempel verwendet wird, dann wird das Ticket in die Dezember-Ergebnisse aufgenommen. Wenn der Wert in "Gelöst am" als Zeitstempel verwendet wird, dann wird das Ticket in die Januar-Ergebnisse aufgenommen. Beide Optionen können verwendet werden. Die Auswahl muss nur den Erwartungen bezüglich des Zeitpunkts, wann die Tickets in Berichten angezeigt werden sollen, entsprechen.

Dies ist ein sehr wichtiger Punkt, der während der Implementierung zu berücksichtigen ist, da er bestimmt, wann die Events für Berechnungen verwendet werden. Wenn ein Ticket beispielsweise im November erstellt und bis Dezember noch nicht geschlossen wurde, wann sollte es in das SLA-Ergebnis aufgenommen werden? Wird es in den November-Daten oder doch in den Dezember-Daten erfasst?



In diesem Fall können die Daten nur erfasst werden, nachdem das Ticket geschlossen ist, da das Ticket erst an die Datenquelle übermittelt wird, wenn es geschlossen ist. Normalerweise liegt das "Geschlossen"-Datum nach dem "Erzeugt"- und "Gelöst"-Datum. Wenn das Datum in "Erzeugt" als Zeitstempel gewählt wurde, dann sollte es als Teil der Dezember-Ergebnisse verarbeitet werden. Das "Geschlossen"-Datum war im Januar, d. h., dass der Dezember bereits vorbei ist, als dieses Ticket übermittelt wurde. Demzufolge waren die Ergebnisse für Dezember bereits veröffentlicht worden. Die Korrelations-Engine erfasst das Event als ein in der Vergangenheit liegendes Event, das entsprechend dem Zeitstempel in den Dezember gehört, und löst eine Neuberechnung aus. Daher ändern sich die Dezember-Ergebnisse rückwirkend.

Diese Konsequenzen müssen vollkommen verstanden werden, um festlegen zu können, welches Zeitfeld als Zeitstempel gewählt werden muss. Normalerweise wird das "Geschlossen"-Datum gewählt, um Berichte zu haben, die sich nicht rückwirkend verändern.

Auf der anderen Seite verzögert die Verwendung des "Geschlossen"-Datums als Zeitstempel, dass die Tickets in die Berechnungen einbezogen werden. Ein Ticket, das gelöst wurde, kann möglicherweise nur eine beträchtliche Zeit später geschlossen werden.

Beachten Sie allerdings, dass durch die Verwendung des "Geschlossen"-Datums ein Prozess in der Organisation ausgelöst werden könnte, durch den die Zeit bis zum Schließen der Tickets verringert wird.

Der letzte Vorschlag:

Event-Name	Ticket der Priorität 1 (es können bei Bedarf auch andere Prioritäten festgelegt werden)
Verhalten	Übermittelt, wenn das Ticket geschlossen ist
Zeitstempelfeld	Geschlossen am
Ressourcenfeld	Kunden-Feld und Standort-Feld
Event-Typ-Feld	Priorität
Datenfelder	Alle
Ressourcentyp-Attribut	Vertragspartei-Standort
Zuordnung zu Vertragspartei	Jeder Standort wird der zugehörigen Vertragspartei zugewiesen
Zuordnung zu Service	Siehe oben
Zuordnung zu Ressourcengruppe	Ressourcengruppe wird für jede Vertragspartei erstellt, um ein Clustering zu ermöglichen
Registrierung über	Bei geclusterten Metriken erfolgt die Registrierung über die Ressource, und die Metrik wird der Ressourcengruppe mit dem Namen "Standorte der Kunden CM3" angehängt Bei Metriken mit ausgewähltem "Geschlossen"-Datum erfolgt die Registrierung über die Vertragspartei und den Service

Beispiel für die Verwendung von anwenderspezifischen Attributen

Die folgende Fallstudie präsentiert ein Beispiel für mehrere dynamische Ziele.

Fallstudie 9: Mehrere dynamische Ziele

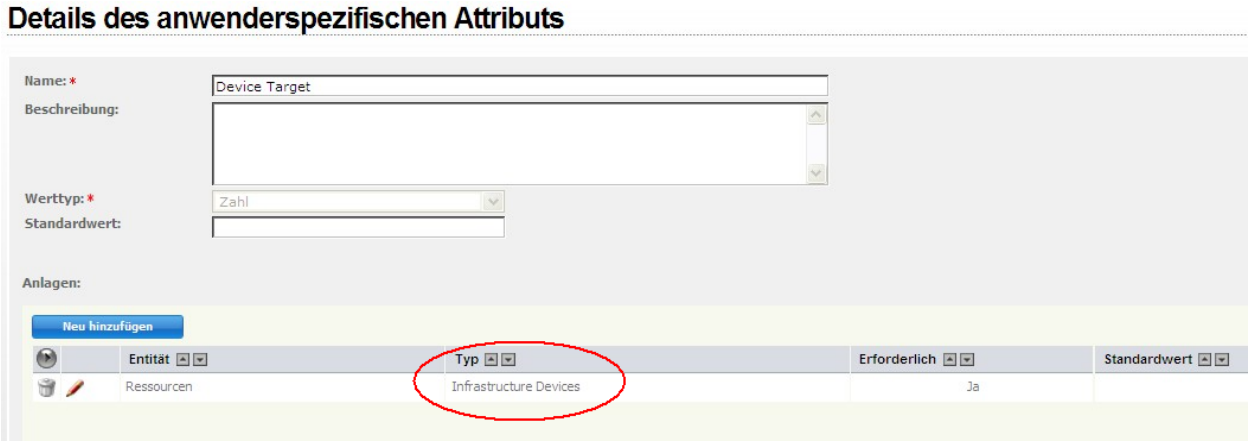
Stellen Sie sich ein Szenario vor, bei dem alle Hardware-Infrastrukturgeräte in einer Kundenumgebung individuell eingestellte Ziele für ihre Verfügbarkeitsanforderungen haben. Bei der Verwendung des Standard-Modellierungsansatzes würde dies eine wirklich schwer erfüllbare Aufgabe sein und zahlreiche logische Gruppierungen für die Geräte und das Management beinhalten, die das Ressourcenmodell verwenden. Und was das Ganze noch komplexer macht, ist die Tatsache, dass sich die Ziele dieser Geräte mit der Zeit ändern können. Diese Zielwerte werden in CA Business Service Insight von einem Übersetzungsskript aktualisiert, sobald die Details in einer CMDB gespeichert sind (in Beispiele für die Best Practices bei Übersetzungsskripten (siehe Seite 254) finden Sie ein Übersetzungsskript-Beispiel)

In diesem Fall könnte die Metrik folgendermaßen aussehen:

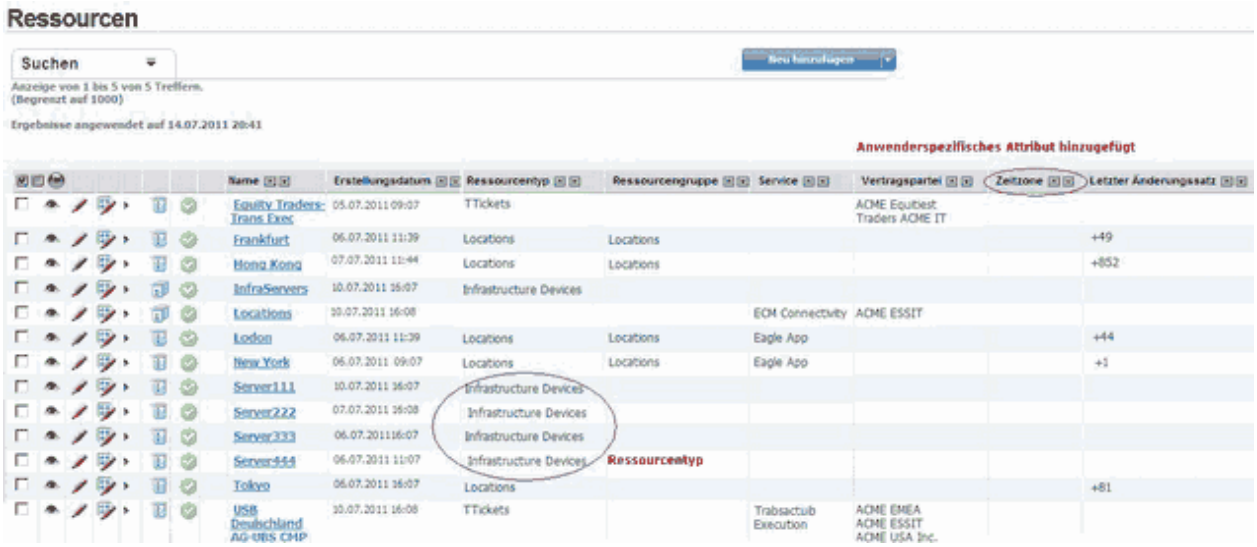
% der Verfügbarkeit pro Hardware-Gerät.

Eine Möglichkeit, dies richtig zu modellieren, ist, neben der Funktion "Anwenderspezifische Attribute" eine der weiteren Hauptfunktion, die Funktion "Dynamischer Ziele", zu verwenden. Beide Funktionen können mit einer geclusterten Metrik verwendet werden, um die gewünschten Ergebnisse zu erreichen. Wird das Service Level-Ziel direkt zur Ressource hinzugefügt, kann die Business-Logik das Service Level von jeder Ressource (Hardware-Gerät) mit ihrem eigenen Ziel vergleichen. Eine geclusterte Metrik gibt die individuelle Service-Compliance für jede Hardware an, die eine einzelne Metrik verwendet.

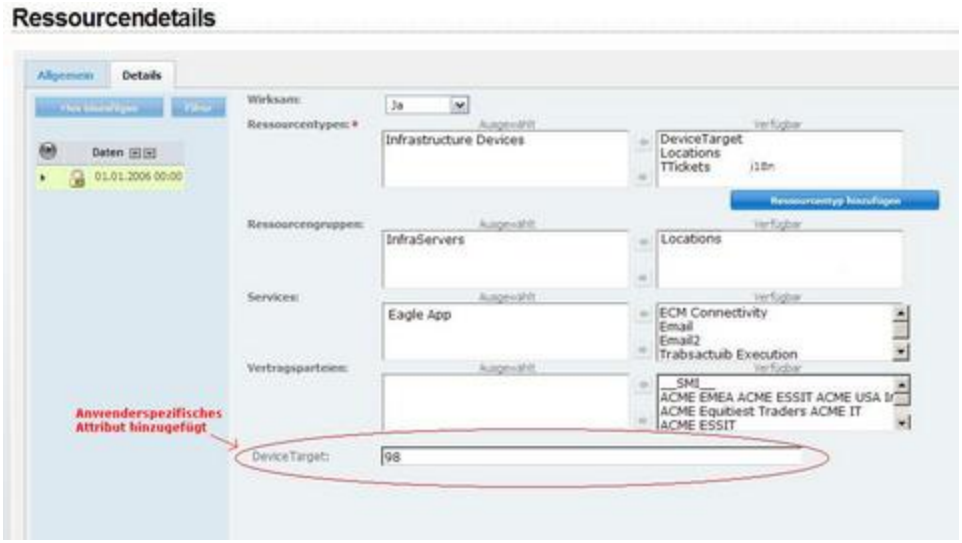
Daher ist es notwendig, zuerst das anwenderspezifische Attribut zu erstellen, indem es zum Ressourcentyp dieser Geräte hinzugefügt wird (wobei alle Geräte den Ressourcentyp "Infrastrukturgerät" aufweisen). Das erstellte anwenderspezifische Attribut wird als "DeviceTarget" bezeichnet und kann über das Menü unter *Servicekatalog->-Anwenderspezifische Attribute* hinzugefügt werden. Hinweis: Wenn Sie das anwenderspezifische Attribut erstellen, müssen Sie es mit den Ressourcentypen verknüpfen, die es benötigen.



Wenn Sie jetzt die Ressourcen im System anzeigen, können Sie sehen, dass das neue anwenderspezifische Attribut für den Ressourcentyp verfügbar ist, mit dem es verknüpft wurde.



Und die individuellen Ressourcen haben ein neues Feld, das aktualisiert werden kann.



In diesem Beispiel würde dieses Feld normalerweise vom Übersetzungsskript eingefügt bzw. aktualisiert werden.

Da jede der Ressourcen ein für sie festgelegtes Ziel hat, können Sie die Logik entwickeln, um die erforderliche Berechnung durchzuführen (nachdem die Ressourcenänderungen vorgenommen wurden). Der folgende Beispielcode zeigt, wie der anwenderspezifische Attributwert aus der Ressource extrahiert wird (in Fettschrift).

```
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

Function Target
   Target = eventDetails.CustomAttribute ("DeviceTarget")
End Function

Function Result
   If TotalTime > 0 Then
      Result = (TotalTime - OutageTime) / TotalTime
   Else
      Result = Null
   End If
End Function
```

Beispiele eines Übersetzungsskripts

Die folgenden Fallstudien schließen Beispiele zu grundlegenden automatischen Übersetzungen und Aktualisierungen von Ressourcenmodellen ein.

Fallstudie 10: Grundlegende automatische Übersetzung

Das hier angegebene Beispiel eines Übersetzungsskripts ist ein ziemlich einfaches Beispiel für eine Verarbeitung von ausstehenden Einträgen im Bildschirm "Übersetzungseinträge".

Der OnTranslation-Event-Handler führt eine einfache Überprüfung der ersten Zeichen in der Ressource und dann entsprechend des Werts eine Aktion aus: Wenn der Wert "a" ist, wird der Übersetzungseintrag auf "ignorieren" gesetzt. Ist der Wert "b", wird der Eintrag gelöscht und ist der Wert "c", wird er übersetzt. Bei einem anderen Wert wird der Eintrag unverändert gelassen, um ihn manuell zu übersetzen. Hinweis: Anhand der Zählungen innerhalb des Codes kann nachverfolgt werden, welche Aktionen während der Skriptausführung durchgeführt werden. Dies ist jedes Mal, wenn das Skript ausgeführt wird, sehr nützlich für das Debugging oder die Dokumentation der Skriptausführungen, insbesondere wenn dies automatisch geschieht. Es ist sehr wichtig, an den Befehl *Tools.Commit* am Ende der Funktion zu denken, da ohne diesen Befehl keine der durch das Skript vorgenommenen Änderungen in der Datenbank gespeichert werden.

Die so genannte Funktion "TranslateResource()" führt einfach eine Prüfung durch, um zu sehen, ob bereits eine Ressource mit demselben Namen wie der, der vom ausstehenden Übersetzungseintrag (zusammen mit dem Präfix E2E) übermittelt wurde, im System existiert. Ist dies nicht der Fall, fügt das Skript diese Ressource hinzu und führt anschließend die Übersetzung aus. Existiert der Name bereits, dann erzeugt das System einen Übersetzungseintrag aus dem Ressourcenstring für die vorhandene CA Business Service Insight-Ressource.

Die Ergebnisfunktion am Ende des Skripts gibt einfach eine Beschreibung der vom Skript ausgeführten Aufgaben aus. Der Code sieht wie folgt aus:

```
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
```

Fallstudie 11: Aktualisieren der Ressourcenmodelle

Eine weitere Verwendung von Übersetzungsskripten ist, das CA Business Service Insight-Ressourcenmodell mit Werten, die von einer externen Datenquelle übernommenen wurden, zu aktualisieren. Dieses Beispiel bezieht sich streng genommen nicht mehr auf die Übersetzung an sich, sondern ist eine sehr nützliche Möglichkeit, das System automatisch zu aktualisieren.

Im vorherigen Abschnitt zu anwenderspezifischen Attributen wurde ein Szenario beschrieben, in dem die Hardware-Infrastrukturgeräte einer Organisation zusammen mit einem zu erwartenden Verfügbarkeitsziel für jedes Gerät in einer externen CMDB gespeichert wurden. Diese Informationen müssen im CA Business Service Insight-Ressourcenmodell repliziert werden, um die Infrastrukturzuordnung (und die Geräteziele) auf dem neuesten Stand zu halten.

In diesem Fall muss das Skript die folgenden Aufgaben ausführen:

- Fügen Sie neue Hardware-Geräte hinzu, die gegenwärtig noch nicht im System vorhanden sind.
- Aktualisieren Sie das zu erwartende Verfügbarkeitsziel von Geräten, die schon vorhanden sind (wenn das Ziel verändert wird).

Das Skript sieht wie folgt aus:

```
Option Explicit

'*********************************************************************
'Global Variables and constants
'*********************************************************************

Dim added
Dim updated
Dim ChangeSetName

added = 0
updated = 0

Const RESOURCE_TYPE = "Infrastructure Devices"
Const RESOURCE_GROUP = "InfraServers"
Const CHANGESET_NAME = "Infrastructure Devices"
Const CHANGESET_EFFECTIVE_DATE = "01.01.07"

'*********************************************************************
'Sub OnLoad :
'Preparing the foundation infrustructure enteties
'*********************************************************************

Sub OnLoad()
   Tools.log "Translation Script : In OnLoad procedure", "D"
```

```
   'Search for existing preliminary resource version
   Dim ChangeSetMap
   Set ChangeSetMap=Tools.SearchChangeSets(CHANGESET_EFFECTIVE_DATE,
CHANGESET_NAME)

   'If no existing version create a new version
   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")
      'Add resources to latest version if it doesnt exist already
      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
            updated = updated + 1
         End If
```

```
      End If

      Tools.Commit
      rs.MoveNext
   Loop

   rs.Close
   conn.Close
   Set rs = Nothing
   Set conn = Nothing
End Sub

Function Result
   ' Commit the transaction
   Tools.CommitChangeSets CHANGESET_NAME
   Result = added & " resources added and " & updated & " resources updated."
End Function

'*******************************************************************************
```

Beispiele für das Business-Logik-Skripting

Hier finden Sie einige allgemeine Richtlinien für das Business-Logik-Skripting:

Globale Variablen

- Achten Sie darauf, die globalen Werte, die Sie angeben, zu unterzeichnen. Der PSL-Statusmechanismus kann keine Variablen mit einem Wert von Null speichern.

Allgemeine Codierung

- Bevor Sie eines der unten aufgelisteten Objekte verwenden, stellen Sie sich, dass es existiert (mit einer geeignete IsExist-Methode):
  - Alle Parametertypen (z. B. Tabelle, Liste usw.)
  - Anwenderspezifisches Attribut
  - Resource
- Stellen Sie sicher, dass Sie ein Business-Logik-Modul zusammen mit all den erforderlichen Parametern bereitstellen.
- Überprüfen Sie, welche Metriken den Namen verwenden, bevor Sie den Namen einer Ressource verändern, und aktualisieren Sie sie entsprechend.

Zuordnungsgrafik

- Die Verwendung von Zuordnungsgrafiken in geclusterten Metriken: Zuordnungsgrafiken benötigen mehr Berechnungsaufwand seitens der Engine. Denken Sie daran, dass sich der Aufwand um die Anzahl der Cluster-Objekte multipliziert, wenn Sie eine Metrik clustern.
- Große globale Zuordnungsgrafiken innerhalb der Business-Logik von geclusterten Metriken sollte nur nach sorgfältiger Überlegung verwendet werden. Während die Engine eine geclusterte Metrik berechnet, ist sie damit beschäftigt, die globalen Variablen für den Status eines jeden Objekts im Cluster gesondert zu laden.
- Achten Sie darauf, die Zuordnungsgrafiken und Vektoren zu löschen, wenn Sie mit ihnen fertig sind
- Wenn es erforderlich ist, größere Zuordnungsgrafiken zu verwenden, stellen Sie sicher, dass die Zuordnungskarte effizient verarbeitet werden kann, indem Sie sie in logische Bereiche einteilen.

Fallstudie 12: Verwenden der Zähler-Logik, um die Anzahl der Fehler zu berechnen

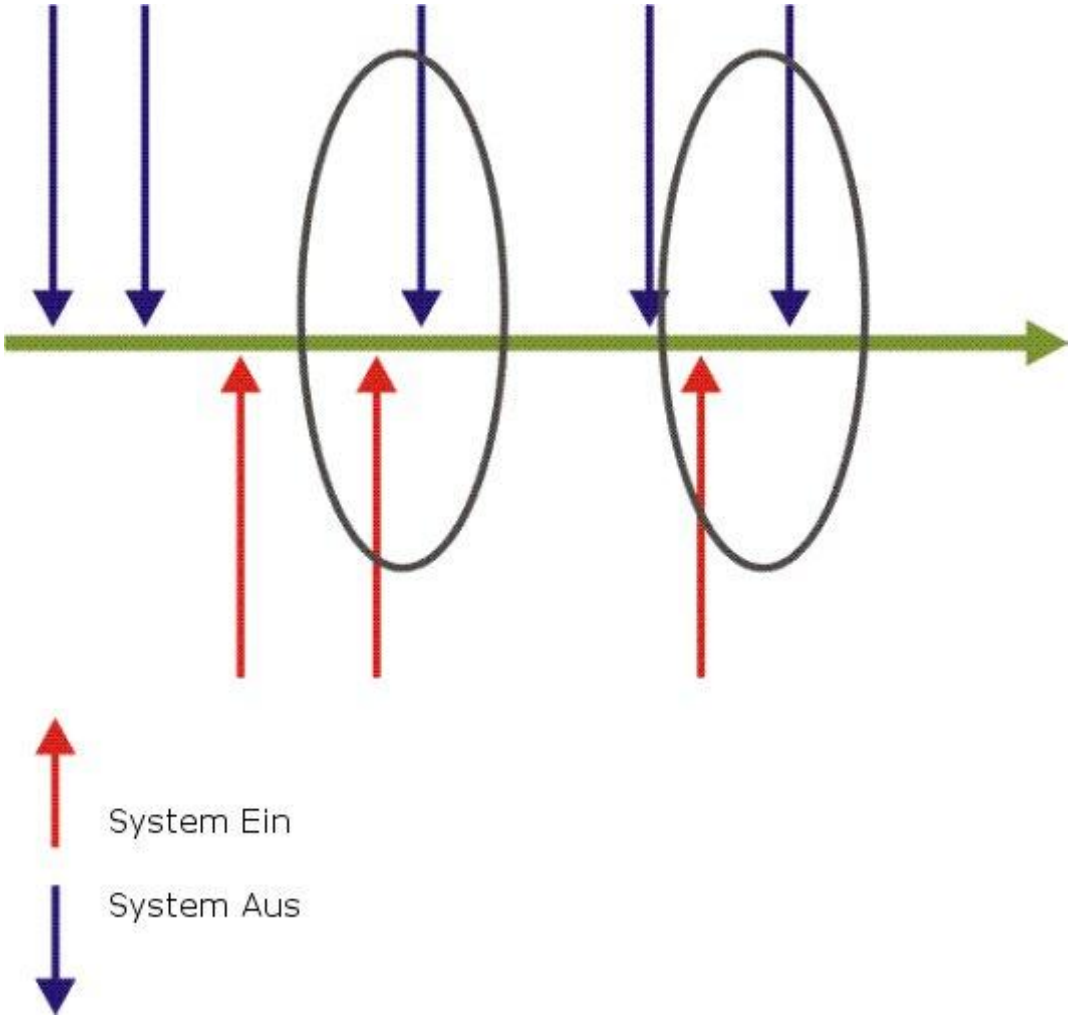
Das folgende Beispiel berechnet die Anzahl der Fehler, die innerhalb eines bestimmten Berechnungszeitraums auftraten. Diese Formel und die Methoden, mit denen sie implementiert wurde, können als Beispiel einer Formel verwendet werden, die immer erforderlich ist, wenn etwas berechnet werden soll.

Es wurden folgende Annahmen für die Berechnung getroffen:

- Eingabe-Events:
  - Verfügbarkeits-Events, Status (0/1)
  - Das Verfügbarkeits-Event trifft alle paar Minuten ein. Es ist nicht möglich, die Häufigkeit von Events zu schätzen (das Event kann alle fünf Minuten oder einmal pro Stunde auftreten), und es gibt möglicherweise auch redundante Events (zum Beispiel: 0, 0, 0, 1, 1, 0, usw.).
- Zeitfenster - Business-Stunden, Fehler, die außerhalb eines Zeitfenster auftreten, werden nicht gezählt.
- "Erforderliches Ergebnis" zeigt an, wie viele Fehler während des Zeitraums aufgetreten sind.

Es ist notwendig, eine periodische Zähler-Variable und auch eine Variable zur Speicherung des Systemzustands zu speichern, um die Fehler, die während des Berechnungszeitraums auftreten, zu zählen. Da redundante Event-Informationen aufgenommen werden können (d. h. ein "Aufwärts"-Event gefolgt von einem anderen "Aufwärts"-Event), ist es auch notwendig, die Anzahl von Orten, an denen eine Änderung des Systemstatus von "Aufwärts" zu "Abwärts" vorkommt, und nicht nur den Empfang eines "Abwärts"-Events zu zählen. Dies könne ein redundantes Event sein, dass einen bereits gezählten Fehler repräsentiert.

Die folgende Abbildung zeigt grafisch, wie die "Aufwärts"- und "Abwärts"-Zeiten des Systems gezählt werden.


System Ein
System Aus

Wichtige Punkte, die berücksichtigt werden sollten:

- **Konstanten** - Es wird empfohlen, eine Konstantendefinition anstatt eines Konstantenwerts in einem Code zu verwenden. Auf diese Weise ist es einfacher, wenn ein Wert geändert werden muss. Sie müssen einfach den Wert in der Konstantendefinition ändern und nicht den ganzen Code nach diesem Wert durchsuchen. Außerdem ist es so leicht, den Wert bei Bedarf in einen Parameter zu verwandeln. Im obigen Beispiel werden die Werte, die den im Event empfangenen Systemstatus repräsentieren, als Konstante festgelegt, um den Code verständlicher zu machen und auch um zu differenzieren, wann die Null als Zahl (für Zählzwecke) verwendet wird und wann sie einen Systemstatus wiedergibt.
- **Globale Variablen:**
  - **Zähler** - Die Definition der Zählervariablen ist eine globale Definition. In der Formel wird sie oben im Code und außerhalb von Subroutinen oder Funktionen angegeben. Es ist wichtig, die Zählervariable in diesem Fall als eine globale Variable festzulegen. So kann sie in mehreren Subroutinen/Funktionen innerhalb des Codes verwendet werden. Ferner kann dadurch das System seinen Wert über den Berechnungszeitraum behalten und sein Ergebnis am Ende des Zeitraums bereitstellen.

    In diesem Beispiel muss die Zählervariable an drei Stellen im Code verwendet werden:
    - Sie muss am Anfang eines Zeitraums initialisiert werden.
    - Sie muss im Falle eines fehlgeschlagenen Events im Event-Handler erhöht werden.
    - Sie muss von der Ergebnisfunktion am Ende des Zeitraums ausgegeben werden.
  - **Vorheriger Systemstatus** - Diese Variable repräsentiert den Status des Systems und wird jedes Mal aktualisiert, wenn ein neues Event mit dem Systemstatus empfangen wird. Diese Variable muss auch global sein, weil es in mehreren Subroutinen/Funktionen im Code verwenden wird, wie z. B.:
    - Sie muss am Anfang der Berechnung initialisiert werden.
    - Sie muss aktualisiert werden, wenn ein neues Event empfangen wird.
- **Überlegungen zum Zeitfenster** - Es ist möglich den Wert der Eigenschaft *context.IsWithinTimeSlot* zu verwenden, um zu verifizieren, ob ein Fehler in einem Zeitfenster aufgetreten ist oder nicht. Der Kontext ist ein globales Objekt, das überall im Code verwendet werden kann. In diesem Fall wird es verwendet, um anzuzeigen, wann ein Fehler empfangen wurde und ob dies innerhalb oder außerhalb des Zeitfensters war. Wenn das Kennzeichen zum Zeitpunkt des Event-Zeitstempels "WAHR" zurückgibt, dann tritt das Event innerhalb eines Zeitfensters auf. Wenn er jedoch "FALSCH" ausgibt, findet das Event außerhalb des Zeitfensters statt. Entsprechend der erforderlichen Logik wird jeder Fehler, der außerhalb des Zeitfensters auftritt, ignoriert und somit nicht gezählt.
- **Variable initialisieren**:

- **Zählervariable** - Die Variable repräsentiert einen periodischen Wert und wird daher in der *OnPeriodstart*-Routine initialisiert, um sicherstellen, dass jeder Berechnungszeitraum seine Fehler gesondert zählt. Jeder Zeitraum startet mit Null und gibt nur die Anzahl der Fehler in diesem Zeitraum aus.

  Wenn es notwendig ist, die akkumulierten Fehler innerhalb jedes Zeitraums zu berechnen (d. h. das Ergebnis von jedem Zeitraum umfasst alle bis zum Ende dieses Zeitraums aufgetretenen Fehler, einschließlich aller vorherigen Zeiträume), dann muss der Zähler nur in der *OnLoad*-Routine initialisiert und aus der *OnPeriodStart*-Routine gelöscht werden. Auf diese Weise fährt der Zähler fort, zu zählen und die Ergebnisse zwischen den Zeiträumen wie nachfolgend gezeigt zu akkumulieren:

```
Sub OnLoad(time)
    FingerprInted=0
End Sub
```

- **Systemstatus-Variable** - Die Variable sollte einmal am Anfang der Berechnung initialisiert und dann bei jedem Status-Event aktualisiert werden. Diese Variable behält seinen Wert während der Berechnung und ist nicht auf einen bestimmten Berechnungszeitraum beschränkt. Sie muss ihren Wert auch zwischen den Berechnungszeiträumen beibehalten. Da der Status des Systems vor dem Empfang des ersten Events unbekannt ist, muss eine Annahme bezüglich des Systemstatus getroffen werden. Die übliche Annahme ist, dass das System den Status "Aufwärts" hat. Diese Annahme sollte zuvor vereinbart und geprüft werden, da sie zu Folgendem führen kann:

  Wenn die Berechnung gestartet wurde und das System in Wirklichkeit den Status "Abwärts" hat und dann das erste Event den Status "Abwärts"-Status anzeigt, wird dies als Fehler gezählt, da der angenommene Status "Aufwärts" war. Dieser Fehler wird für den ersten Berechnungszeitraum gezählt, auch wenn er nicht notwendigerweise während dieses Zeitraums aufgetreten.

```
Option Explicit

'Konstantendefinitionen
Const UP=1
Const DOWN=0

'Global variable for counting failures
Dim FingerprInted
Dim SystemStatus

Sub OnRegistration(dispatcher)
   ' The parameters of the method are: <procedure name>, <Event Type>
   dispatcher.RegisterByContractPartyAndService
"OnAvailabilityEvent","AvailabilityEvent"
End Sub

Sub OnLoad(time)
   SystemStatus = UP 'assume the first system status
End Sub
```

```
Sub OnPeriodStart(time)
   FingerprInted = 0
End Sub

Sub OnAvailabilityEvent(eventDetails)
   ' In case it's a failure and within the timeslot then it is counted
   If context.IsWithinTimeSlot and SystemStatus=UP and _
         eventDetails("Status")=DOWN Then
      FingerprInted = FingerprInted + 1
   End If
   ' update the system status for next comparison
   SystemStatus = eventDetails("Status")
End Sub

Function Result
   Result = FingerprInted
End Function
```

Fallstudie 13: Umgang mit dynamischen Komponentengruppen

Es ist häufig notwendig, Werte für eine Gruppe von Ressourcen zu speichern, wobei die Mitglieder dieser Gruppe dynamisch sein und sich während des Berechnungszeitraums ändern können. In der folgenden beispielhaften Anforderungsberechnung ist es notwendig, eine intermediäre Berechnung für jede Ressource durchzuführen, um am Ende das aggregative Ergebnis zu erhalten.

Nachfolgend sind einige solche Beispiele angegeben:

- **Incidents am Standort** - Berechnen Sie den Prozentwert für die Standorte mit Incidents, deren Lösung länger als X dauerten, oder zählen Sie die Standorte, die Incidents mit durchschnittlichen Lösungszeiten hatten, die höher waren als X.

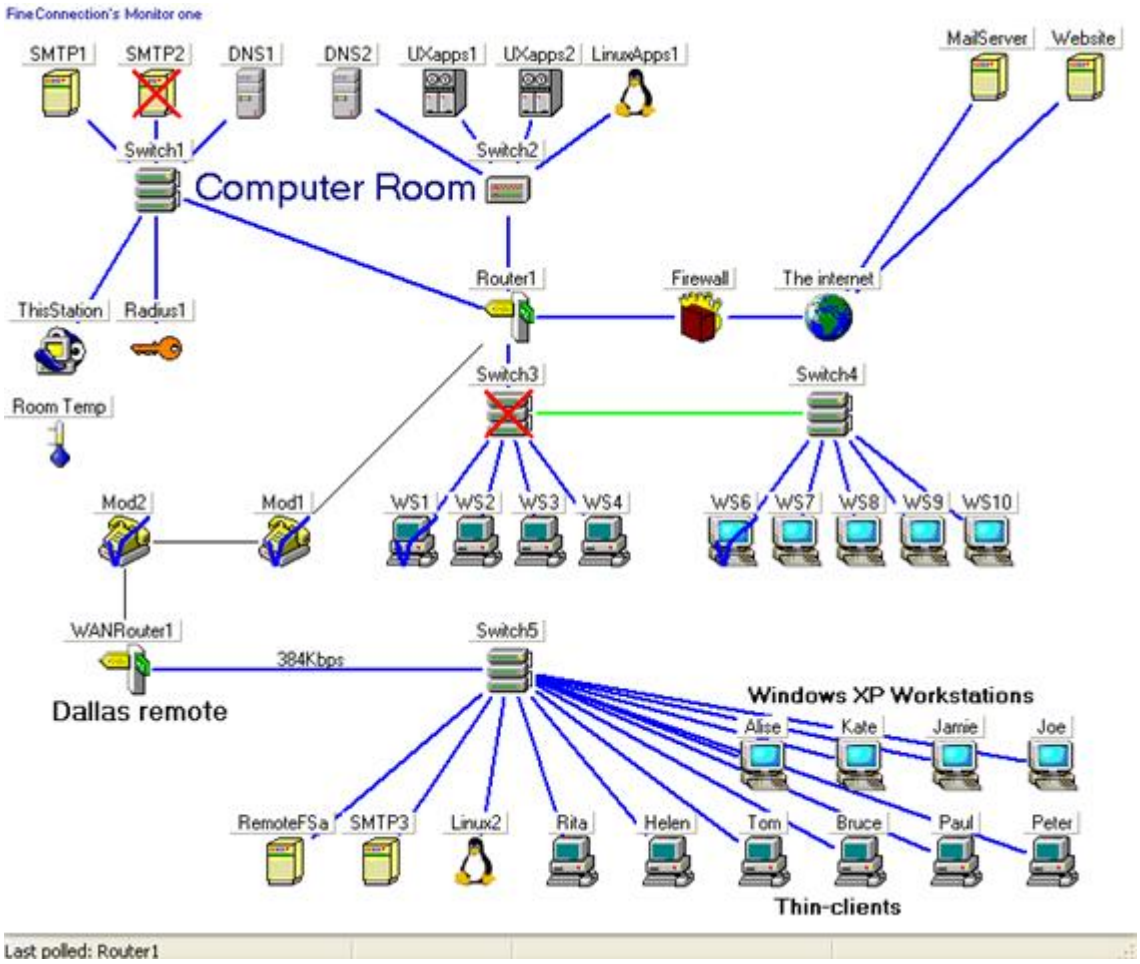
  In diesen Beispielen sind Standorte Ressourcen, die mit ihnen verknüpfte Incidents haben.

- **Serververfügbarkeit** - Zählen Sie die Anzahl der Server, deren Verfügbarkeitszeit höher war als X %.

  Ein Server ist eine Ressource, für die der Verfügbarkeitsprozentsatz bewertet werden muss.

- **Transaktionstypen** - Berechnen Sie den Prozentwert der Transaktionstypen, die mehr als X Mal fehlgeschlagen sind.

  In diesem Fall ist ein Transaktionstyp eine Ressource, die mit ihren fehlgeschlagenen Events verknüpft sind. Für jeden Transaktionstyp wird ein Fehlerzähler als intermediäres Ergebnis gespeichert und die Anzahl von unterschiedlichen Transaktionstypen, die mehr als X Fehler hatten, wird gezählt.


FineConnection's Monitor one
SMTP1
SMTP2
DNS1
DNS2
UXapps1
UXapps2
LinuxApps1
MailServer
Website
Switch1
Switch2
Computer Room
Router1
Firewall
The internet
ThisStation
Radius1
Switch3
Switch4
Room Temp
Mod2
Mod1
WS1
WS2
WS3
WS4
WS6
WS7
WS8
WS9
WS10
WANRouter1
384Kbps
Switch5
Dallas remote
Windows XP Workstations
Alise
Kate
Jamie
Joe
RemoteFSa
SMTP3
Linux2
Rita
Helen
Tom
Bruce
Paul
Peter
Thin-clients
Last polled: Router1

***Beispiel*:**

Für die folgende Berechnungsanforderung:

Berechnen Sie den Prozentsatz der Verfügbarkeit für ein System, das aus einem Cluster von Servern besteht. Das System wird nur dann als verfügbar angesehen, wenn alle zu Grunde liegenden Server verfügbar sind.

Das Lösungsdesign wird folgendermaßen implementiert:

Die Systemverfügbarkeit wird anhand der Verfügbarkeit der zu Grunde liegenden Cluster-Ressourcen beurteilt. In diesem Fall ist es erforderlich, den Status aller geclusterten Ressourcen zu bearbeiten und zu speichern, um den Systemstatus zu jedem Zeitpunkt zu beurteilen.

Dazu muss die Formel eine Zuordnungsgrafik (die Zuordnungsgrafik "ServerAvailabilityIndicators" im nachfolgenden Beispielcode) umfassen, die einen Eintrag für jede der überwachten Ressourcen enthält. Da alle Zuordnungsgrafikobjekte ein Schlüsselfeld erfordern, um auf den zugeordneten Wert hinzuweisen, wird der Ressourcenindikator der Schlüssel (dies ist die Ressourcen-ID) dieser Zuordnungsgrafik und der Wert der Komponentenstatus sein. Wann immer sich der Status einer Komponente ändert, sollte auch der zugehörige Eintrag in dieser Zuordnungsgrafik geändert werden. Die Formel wird den Status der zugehörigen Ressource in der Zuordnungsgrafik nach jedem angezeigten Verfügbarkeits-Event aktualisieren und den Systemstatus entsprechend neu beurteilen.

Da sich die überwachten Ressourcen ändern können (einige könnten während des Berechnungszeitraums entfernt und andere hinzugefügt werden), muss dies innerhalb der Formel verwaltet werden. Dazu wird eine Funktion hinzugefügt, die die Änderung identifiziert und die überwachte Zuordnungsgrafik der Komponenten aktualisiert, indem ein neuer Eintrag für eine neue Komponente hinzugefügt oder ein Eintrag einer entfernten Komponente gelöscht wird.

OnRegistration ist der Event-Handler, der ein Registrierungsevent verarbeitet, das durch eine Änderung in den überwachten Ressourcen ausgelöst wird. Solch eine Änderung kann das Ergebnis der Übernahme einer Ressource (oder einem Änderungssatz) sein, die entsprechend der Registrierungsmethode der Formel zu Änderungen in den Ressourcen führen kann, die in die Berechnung eingeschlossen sind.

Während jeder Registrierung ist es notwendig, die Zuordnungsgrafik zu aktualisieren, die den Ressourcenstatus mit allen erforderlichen Änderungen umfasst. Dies bedeutet, dass die Zuordnungsgrafik, die die Ressourcenstatus enthält, mit der Zuordnungsgrafik, die die Ressourcen zum Zeitpunkt der Registrierungsausführung (basierend auf der Registrierungsmethode) enthält, verglichen wird. Schließen Sie dazu alle Ressourcen ein, die hinzugefügt wurden, oder löschen Sie die Ressourcen, die entfernt wurden.

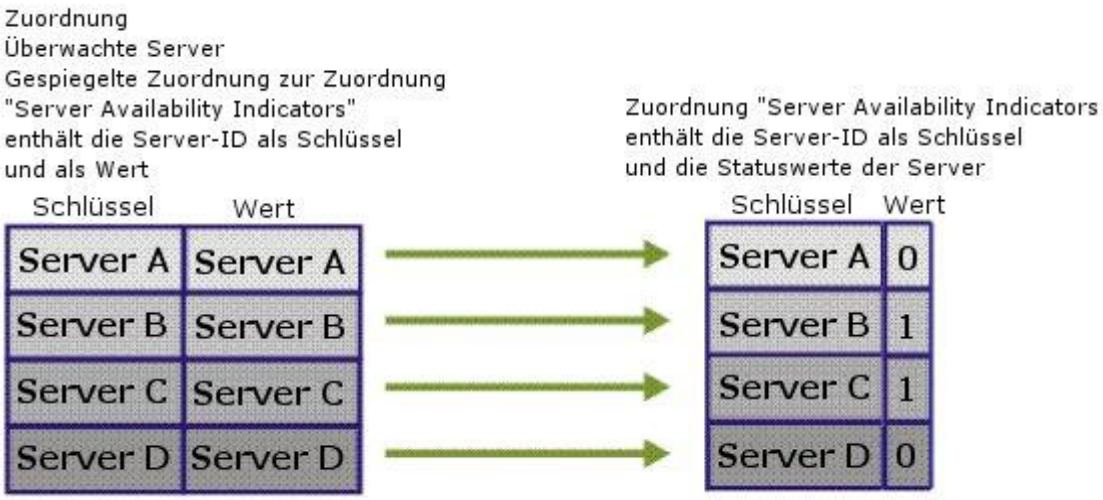
Das OnRegistration-Verfahren muss daher eine Funktion ausführen, die die überwachten Ressourcen mit den neu zugewiesenen Ressourcen vergleicht, um die überwachten Ressourcen dementsprechend zu strukturieren.

Das Kontextobjekt verfügt über einen Satz an Methoden, die mit den Registrierungsmethoden vergleichbar sind. Diese geben eine Zuordnungsgrafik mit all den Ressourcen aus, die entsprechend der Registrierungsmethode Teil der Ressourcen sind.

Im folgenden Beispiel ist die Registrierung der Formel "ByContractParty" und die gleiche Methode wird daher von "Context.ResourcesOfContractParty" verwendet. Dies gibt eine Zuordnungsgrafik mit all den Ressourcen aus, die zum Zeitpunkt der Registrierung Teil dieses Satzes sind.

Um die beiden Zuordnungsgrafiken zu vergleichen, müssen die Zuordnungsgrafiken parallel iteriert werden. Die Iteration der Zuordnungsgrafik erfolgt mithilfe der Erklärung "For Each". Diese Erklärung ermöglicht eine Iteration der Werte einer Zuordnungsgrafik und daher wird eine andere Zuordnungsgrafik als Spiegelbild der Statuszuordnungsgrafik benötigt, um die Ressourcen und nicht ihre Statuswerte zu iterieren. Diese Erklärung iteriert die Werte einer Zuordnungsgrafik und nicht deren Schlüssel. Aus diesem Grund wird eine weitere Zuordnungsgrafik benötigt, die die IDs sowohl als Schlüssel als auch als Wert enthält. Darüber hinaus muss die Spiegelzuordnungsgrafik gepflegt werden, damit sie die Statuszuordnungsgrafik laufend spiegelt und so immer denselben Satz an Ressourcen beinhaltet. Dies bedeutet, dass bei einer Aktualisierung der Statuszuordnungsgrafik auch die Spiegelzuordnungsgrafik aktualisiert werden muss.

Die folgende Abbildung zeigt das Konzept der Spiegelzuordnungsgrafiken.



***Beispiel*:**

```
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
   Next

   For Each Server In MonitoredServers
      If Not allocatedServers.Exist(Server) Then
         MonitoredServers.Erase Server
         ServerAvailabilityIndicators.Erase Server
      End If
   Next
End Sub
```

***Beispiel*:**

Die folgende Funktion zeigt auf, wie die Spiegelzuordnungsgrafik zum Iterieren der Statuszuordnungsgrafik verwendet wird, wenn es erforderlich ist, den Status des ganzen Systems basierend auf dem Status von jeder überwachten Ressource zu beurteilen.

In diesem Beispiel wird das System als verfügbar angesehen, wenn alle Ressourcen verfügbar sind. Sobald nur eine einzige Komponente ausgeschaltet ist, wird das System als nicht verfügbar angesehen:

```
Function SystemAvailability
   Dim Server
   For Each Server In MonitoredServers
      If ServerAvailabilityIndicators(Server) = DOWN then
         SystemAvailability=DOWN
         Exit Function
      End if
   Next
```

```
End Function
```

Ein komplettes Beispiel der Business-Logik mit dynamischer Ressourcenhandhabung wird im folgenden Designmusterbeispiel beschrieben.

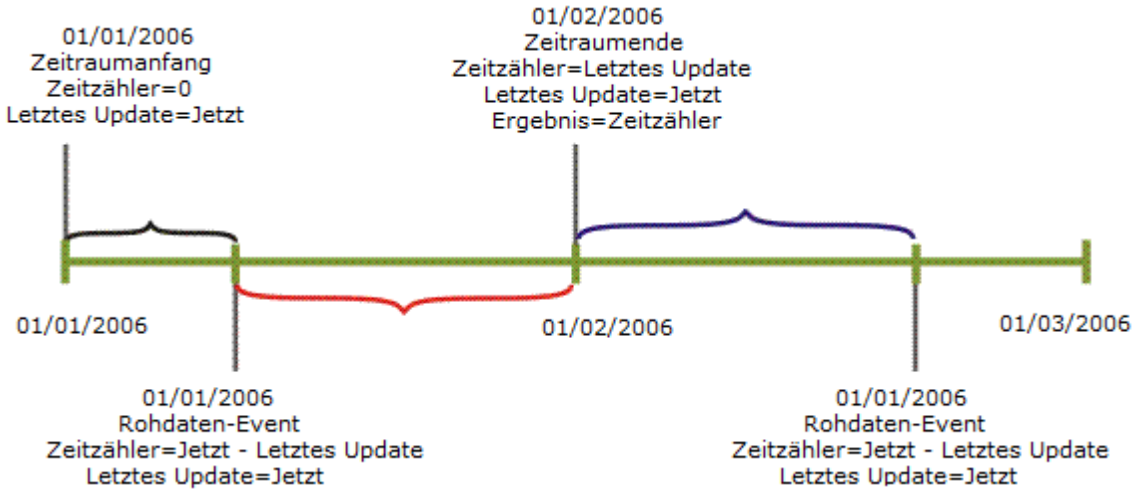
Fallstudie 14: Umgang mit der Zeitakkumulationsuhr

Das in diesem Abschnitt beschriebene Designmuster ist immer dann geeignet, wenn das erforderliche Ergebnis eine Funktion eines Zeitraums ist, der zwischen Events verstrichen ist, wie z. B.:

- Die verfügbare Zeit, berechnet als Zeit, die zwischen einem Fehler und einem anderen verstrichen ist.
- Die Lösungszeit, berechnet als Zeit zwischen der Öffnungs- und Schließzeit eines Incidents.

Bei der Akkumulationszeit muss eine Variable, in der die Zeit akkumuliert werden kann (in Sekunden), zugewiesen und eine Funktion implementiert werden, die sowohl die Bedingungen als auch die seit der letzten Aktualisierungszeit akkumulierte Zeit überprüfen. Diese Funktion wird dann für jedes Event, das von der Formel empfangen wird, ausgeführt.

Die folgende Abbildung stellt die Akkumulation der Uhr-Betriebszeit dar.



Die Variable "LastUpdateTime" speichert die letzte Zeit, zu der die Aktualisierung ausgeführt wurde, unabhängig davon, ob der Zeitzähler aktualisiert wurde oder nicht. Die Funktion enthält die Bedingung, die bestimmt, ob die Zeit aktualisiert und akkumuliert werden sollte. Die Zeit sollte zum Beispiel nicht berücksichtigt werden, wenn sie das Zeitfenster überschreitet, der Systemstatus "Abwärts" lautet oder der Incident einen ausstehenden Status hat.

Obwohl in der hier ausführlich behandelten Situation häufig die Funktion "Tools.NetTime()" verwendet wird, um die Dauer zu berechnen, kann es in einigen Fällen sinnvoller sein, die VB-Funktion "DateDiff()" zu verwenden.

Die Funktion "Tools.NetTime" bedingt jedes Mal, wenn sie verwendet wird, einen zusätzlichen Aufwand für das Prüfen des Zeitfensters. Es wird empfohlen, zu vermeiden, die Funktion "NetTime" im Daten-Event zu verwenden, da diese Verfahren für jedes neue eingehende Event abgerufen werden und daher den "NetTime" -Abruf aktivieren. Wenn Sie ein Zeitfenster von "24/7" haben, wird empfohlen, dass Sie die Funktion "DateDiff" einsetzen, um einen zusätzlichen Aufwand für das Prüfen des Zeitfensters zu vermeiden.

Beispiel 1:

Die folgende Routine "Zähler aktualisieren" akkumuliert den Gesamtzeitraum in der Variablen "PeriodNetTime". Bei "AvailabilityTime" akkumuliert die Routine die Zeit, in der das System den Status "Aufwärts" hatte, d. h. in der das System verfügbar war.

Das Tools-Objekt enthält die "NetTime"-Methode - NetTime(beginTime, endTime)0. Diese Methode gibt die Anzahl von Sekunden zwischen "beginTime" und "endTime" aus, die sich innerhalb des Zeitfensters der aktuellen Metrik befinden. Jede Zeit zwischen diesen beiden Variablen ist Teil eines ausgeschlossenen Zeitfensters, daher wird sie häufig für die Berechnung der Dauer, bei der ein Zeitfenster verwendet wird, eingesetzt. (Zum Beispiel: bei Tickets der Priorität 1, die innerhalb von vier Geschäftsstunden gelöst werden sollen, wurde ein Ticket am Ende eines Geschäftstages erstellt, das möglicherweise erst am nächsten Morgen gelöst wird. Dennoch wird die SLA eingehalten, da die Stunden des Zeitfensters ausgeschlossen wurden.)

```
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
```

Beispiel 2:

Im folgenden Beispiel wird die Anwendungsverfügbarkeit berechnet, in dem die "Aufwärts"- und "Abwärts"-Events von verschiedenen wichtigen Komponenten sowie die Wartungsevents dieser Komponenten verarbeitet werden. Wenn alle Komponenten gewartet werden, wird die Zeit nicht als erwartete Verfügbarkeitszeit angesehen.

Die Subroutine "UpdateCounters" erhöht die Zeitzähler, wenn dies notwendig sein sollte, und wird bei jedem Event, das von der Formel empfangen wird, ausgeführt (Rohdaten-Event/Engine-Event). Sie aktualisiert auch die erwartete Verfügbarkeitszeit in Fällen, in denen die Zeit im Zeitfenster liegt und sich die Komponenten nicht in einem Zeitraum für einen geplanten Ausfall befinden. Die Formel aktualisiert die tatsächliche Verfügbarkeitszeit nur, wenn sie auch einen Systemstatus "Aufwärts" erfasst.

"DateDiff" ist eine standardmäßige VB-Funktion, die die Zeit zwischen zwei Daten ausgibt, jedoch keine Zeitfenster-Informationen ausschließt.

```
'Force variable declaration
Option Explicit

'Global variables
Dim ExpectedAvailabilityTime
Dim ActualAvailabilityTime
Dim LastUpdateTime
Dim AvailabilityIndicators
Dim MonitoredComponents
Dim DowntimeStatuses

'Map objects creation
Set AvailabilityIndicators=CreateObject("SlalomMap.Map")
Set MonitoredComponents=CreateObject("SlalomMap.Map")
Set DowntimeStatuses=CreateObject("SlalomMap.Map")

'After loading and whenever an infrastructure change occurs
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")

   'make sure formula is monitoring only relevant and all the relevant resources
   UpdateMonitoredComponents AllocatedComponents
End Sub

Sub OnLoad(time)
   'When system goes up for the first time - assume availability OK
   LastUpdateTime = time
End Sub
```

```
Sub OnPeriodStart(time)
   'initializing counters to renew periodic calculation
   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
   'write availability status of reported-on resource
   AvailabilityIndicators(eventDetails.ResourceId) = _
      AvailabilityIndicators(eventDetails.ResourceId)+1
End Sub

Sub OnComponentUpEvent(eventDetails)
   UpdateCounters eventDetails.Time
   'write availability status of reported-on resource
   AvailabilityIndicators(eventDetails.ResourceId)= _
      AvailabilityIndicators(eventDetails.ResourceId)-1
End Sub

Sub OnMaintenanceStartEvent(eventDetails)
   UpdateCounters eventDetails.Time
   'write availability status of reported-on resource
   DowntimeStatuses(eventDetails.ResourceId)= _
      DowntimeStatuses(eventDetails.ResourceId)+1
End Sub

Sub OnMaintenanceEndEvent(eventDetails)
   UpdateCounters eventDetails.Time
   'write availability status of reported-on resource
   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
      'update counter of seconds in period (when availability is expected)
      ExpectedAvailabilityTime = ExpectedAvailabilityTime +
DateDiff("s",LastUpdateTime,time)
      If SystemIsAvailable Then
         'update seconds-of-availability counter
         ActualAvailabilityTime = ActualAvailabilityTime +
DateDiff("s",LastUpdateTime,time)
      End If
   End If
   LastUpdateTime=time
End Sub

Sub UpdateMonitoredComponents(allocatedComponents)
   Dim Component
   'add to monitored Components map all new Components to be monitored
   For Each Component In allocatedComponents
      If Not MonitoredComponents.Exist(Component) Then
         MonitoredComponents(Component) = Component
         AvailabilityIndicators(Component) = 0
         DowntimeStatuses(Component) = 0
      End If
   Next

   'Aus der Zuordnungsgrafik für überwachte Komponenten alle nicht länger
zugehörigen Komponenten entfernen
   For Each Component In MonitoredComponents
      If Not allocatedComponents.Exist(Component) Then
         MonitoredComponents.Erase Component
         AvailabilityIndicators.Erase Component
         DowntimeStatuses.Erase Component
      End If
   Next
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
      'system availability is evaluated with availability
      SystemAvailability = SystemAvailability And ComponentAvailability
   Next
   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
   Next

   If ComponentsInPlannedDowntime = MonitoredComponents.Count Then
      AllComponentsAreInPlannedDowntime = True
   Else
      AllComponentsAreInPlannedDowntime = False
   End If
End Function
```

Beispiele für das Schreiben einer effektiven Business-Logik

Die "Best Practice"-Empfehlungen zur Art und Weise, wie eine effektive Business-Logik geschrieben wird, werden anhand der folgenden Themen aufgezeigt:

- **Geclusterte Metriken**
  - Wenn Sie die Größe des Systems beurteilen, zählen Sie eine geclusterte Metrik als Anzahl der Objekte in der Metrik. Denken Sie daran, dass alles multipliziert wird.
  - Eine Neuberechnung eines Cluster-Objekts bedeutet, die Neuberechnung des ganzen Clusters. Beachten Sie dies also, wenn Sie das Clustering und die Art und Weise, wie Adapter konfiguriert werden, planen und die Ressourcenstruktur ändern.
  - Dieselben Rohdaten-Events, die an viele Cluster-Objekten übermittelt werden, haben hohe Leistungskosten (Kontextschaltung)
- **Globale Variablen**
  - Fügen Sie Parameter und Attributwerte an jede Stelle im Code ein, wo sie erforderlich sind. Vermeiden Sie, sie als globale Variable einzufügen, besonders in geclusterten Metriken (dies würde die Größe des "Status" erhöhen)
  - Vermeiden Sie eine Logik, die große Zuordnungsgrafiken verarbeitet. Verarbeiten Sie stattdessen jedes Event in der OnXXEvent-Methode
  - Entfernen Sie so früh wie möglich Objekte aus den Zuordnungsgrafiken. Zum Beispiel, wenn ein Ticket geschlossen wird und nicht erst am Ende des Zeitraums
- **Designmuster**

  Das vordefinierte Inhaltspaket enthält mehrere Designmuster für übliche Szenarien. Die Verwendung dieser Designmuster kann die Leistung verbessern
- **Integrierte Funktionalität**

  ACE verfügt über ein integrierte Funktionalität und Tools für verschiedene Zwecke, wie z. B.:
  - Zeitfenster-Funktionalität
    - NetTime
    - IsWithinTimeslot
  - Zeit der Events
    - TimeOfLastEvent
    - TimeOfLastEventHandler
  - Kontextobjekt
    - Enthält viele umgebungssensitive Methoden
    - Verwenden Sie diese Methoden und vermeiden Sie "Sichere ODBC"

- **Ausgaben der Business-Logik**

  Behalten Sie die Struktur in "T_SLALOM_OUTPUTS" bei. Dies bedeutet, dass es hilfreich ist, ähnliche logische Felder im selben physikalischen Feld zu platzieren, wenn Sie mehrere logische Tabellen mit einer ähnlichen Struktur in "T_SLALOM_OUTPUTS" haben. Dies ermöglicht eine einfache Indexierung zur Verbesserung der Berichtsleistung

- **Event-Wiederverwendbarkeit**

  Tun Sie dies, wenn Folgendes vorliegt:

  - Wenn mehrere Metriken die erste Phasenberechnung durchführen, die selbst für den Vertrag benötigt wird, und Events für eine Übersichtsmetrik senden, die das Ergebnis berechnet (z. B. Finanzkalkulation, KPI mit hohem Level)
  - Wenn eine einzelne Metrik eine vorläufige Aggregation von Rohdaten ausführt und Events an mehrere verschiedene Metriken sendet. Sinnvoll, wenn die Metrik erheblich weniger Events sendet, als sie empfängt, oder bedeutende Berechnungen ausführt, die andernfalls viele Male ausgeführt werden würden

  Vermeiden Sie dies, wenn Folgendes vorliegt:

  - Wenn Sie die Anzahl der Metriken wesentlich erhöhen
  - Wenn Sie mehr als drei Level implementieren
  - Wenn Sie die Komplexität der Implementierung erhöhen und die Verwaltung schwieriger wird

- **Neuberechnungen**

  - Vermeiden Sie massive Neuberechnungen als Teil des Normalbetriebs des Systems
  - Die Gründe für Neuberechnungen sind:
    - Rohdaten mit einem Zeitstempel in der Vergangenheit
    - Event-Besonderheit, die in der Vergangenheit liegende Rohdaten verändert
    - Korrekturen
    - Ausnahmen
    - Änderungen in den Business-Logik-Modulen
    - Änderungen in einem Vertrag
    - Wiederverwendbarkeit von Events - Events mit einem Zeitstempel in der Vergangenheit
    - Änderungen in der Ressourcenstruktur
  - Berücksichtigen Sie die Funktion zur Sperrung von Datenberechnung

- **Business-Logik-Module**

- Die Business-Logik-Module sollten so geschrieben werden, dass sie nach einer kompletten Prüfung nicht mehr geändert werden müssen
- Die Richtlinie ist: ein Modul entspricht einer generischen Logik
- Die Business-Logik-Module, die sehr spezifisch sind, können weder für viele Metriken verwendet werden, noch fördern sie die Wiederverwendbarkeit von Code und Logik
- Die Business-Logik-Module, die zu generisch sind, sind schwierig zu handhaben. Wenn ein Business-Logik-Modul viele komplexe Logiken implementiert, verursacht ein fester Wert in einem Durchgang (der von einem Teil der Metriken verwendet wird) zudem eine Neuberechnung aller Metriken

- **Registrierung**
  - Filtern Sie alle Events mithilfe der Registrierung. Wird das Filtern über den Code ausgeführt, hat dies einen großen Einfluss auf die Leistung
  - Einfache Gestaltung
  - Verwenden Sie bei einem Code, bei dem es sich nicht um die Registrierung selbst handelt, die Methoden "dispatcher.IsRunTimeMode" und "OnResourceStructureChange", insbesondere wenn es viele Ressourcenänderungen gibt
  - Vermeiden Sie es, die Methode "RegisterByEventType" zu verwenden
  - Verwenden Sie in Business-Logik-Modulen eine generische Form (von Vertragspartei, Service, Ressourcentyp) oder Parameter oder lassen Sie die Registrierung über die Benutzeroberfläche ausführen (bevorzugte Option für Event-Wiederverwendbarkeit)

Fallstudie 15: Geclusterte Metriken

Wenn üblicherweise ein bestimmter Teil einer Software beschrieben wird, kann die Beschreibung in zwei Abschnitte gegliedert werden - dem Abschnitt WAS und dem Abschnitt WIE. Im Abschnitt WAS wird beschrieben, was dieser Teil des Codes tut. Der Abschnitt WIE gibt an, wie er das tut. Die meisten neigen dazu, sich auf den WAS-Abschnitt zu konzentrieren und den WIE-Abschnitt zu ignorieren. Der Grund dafür ist einfach und in vielen Fällen gerechtfertigt. Indem Sie so vorgehen, reduzieren Sie die Kopplung zwischen Ihren Komponenten und füllen Ihren Kopf nicht mit Informationen, die in vielen Fällen irrelevant sind. In vielen Fällen jedoch kann das Ignorieren des WIE-Abschnitts zu Lasten der Leistung gehen.

Diese Fallstudie untersucht die Art und Weise, wie die Engine geclusterte Metriken berechnet (d. h. wie sie auf den WIE-Abschnitt reagiert) und beschreibt die Leistungskosten für bestimmte Implementierungen. Sie geht auch näher auf einige Möglichkeiten ein, wie diese Kosten durch die Änderung der Implementierung geändert werden kann.

Was sind geclusterte Metriken?

Geclusterte Metriken sind Metriken, die eine bestimmte Gruppe von Ressourcen in ihrer Definition eingebettet haben. Diese Gruppe wird als Cluster der Metrik bezeichnet, und jede der Ressourcen in dieser Gruppe wird als Cluster-Objekt bezeichnet. Wenn man eine geclusterte Metrik berechnet, wird eine gesonderte Berechnung für jedes dieser Cluster-Objekte ausgeführt. Die Berechnungen für jedes dieser Cluster-Objekte entsprechend einander, außer:

- Context.ClusterItem - Der Wert der Eigenschaft "Context.ClusterItem", der dem Cluster-Objekt entspricht, das berechnet wird.
- Registrierung über Cluster-Objekt - Wenn die Metrik über ein Cluster-Objekt für Events registriert ist, empfängt jedes Cluster-Objekt Rohdaten und Wiederverwendbarkeits-Events, die an dieses Cluster-Objekt gesendet werden.

So werden geclusterte Metriken berechnet

Das wichtigste, was man bei der Berechnung einer geclusterten Metrik verstehen muss, ist, dass alle Cluster-Objekte parallel berechnet werden. Parallel bedeutet in diesem Fall nicht, dass sie von verschiedenen Threads berechnet werden, sondern, dass die Events während der Verarbeitung zahlreicher Events, die von verschiedenen Cluster-Objekten verarbeitet werden sollten, sequenziell verarbeitet werden. Für jedes Event werden die zugehörigen Cluster-Objekte aufgerufen, die dann dieses Event verarbeiten. Zum Beispiel gibt es viele Events, die von vielen Cluster-Objekten verarbeitet werden sollten. Es gibt zwei Optionen, dies zu tun:

Beispiel: Option 1

```
Für jedes Cluster-Objekt C
     Für jedes Event E, das von C verarbeitet werden soll
          Lassen Sie E von C verarbeiten
```

Beispiel: Option 2

```
Für jedes Event E
     Für jedes Cluster-Objekt C, das E verarbeiten soll
          Lassen Sie E von C verarbeiten
```

Die Engine verarbeitet geclusterte Metriken nach der Option 2.

Ein weiterer wichtiger Punkt, den Sie verstehen sollten, ist, dass die Ausführung des VBScript im PslWriter von einer Komponente mit dem Namen Skript-Steuerelement ausgeführt wird. Es gibt nur eine einzelne Instanz dieser Komponente pro Metrik und diese Instanz wird für die Berechnung zahlreicher Cluster-Objekte wieder verwendet. Da die Cluster-Objekte wie zuvor erwähnt parallel berechnet werden und die Komponente Skript-Steuerelement zu jedem Zeitpunkt nur die Daten eines einzelnen Cluster-Objekts enthalten kann, müssen Sie die Daten in der Komponente Skript-Steuerelement häufig umschalten.

Um diesen Vorgang zu erklären, ist nachfolgend ein detaillierter Pseudocode dargestellt, der die Berechnungen durchführt.

```
1-     Für jede Metrik M
2-          Lassen Sie X das früheste Event sein, dass noch nicht von M
verarbeitet wurde
3-          Lassen Sie T den Zeitstempel des letzten Status vor X sein
4-          Lassen Sie L die Liste aller Events sein, die von M registriert
wurden (alle Cluster-Objekte), beginnend mit dem Zeitstempel T bis hin zur
aktuellen Zeit
5-          Für jedes Event E in L
6-               Für jedes Cluster-Objekt C, das E verarbeiten sollte
7-                    Lassen Sie C' das Cluster-Objekt sein, das gegenwärtig ins
Skript-Steuerelement geladen wird
8-                    Nehmen Sie die Werte der globalen Variablen vom Skript-
Steuerelement und speichern Sie sie für C'
9-                    Nehmen Sie die Werte der globalen Variablen, die für C
gespeichert wurden, und laden Sie sie in das Skript-Steuerelement
10-                    Verarbeiten Sie das Event E
```

Dieser vollständige Prozess, um einen Zeitpunkt eines Events zu finden, der noch nicht berücksichtigt wurde, und dann die Berechnung von diesem Zeitpunkt an durchzuführen, wird Neuberechnung genannt. Der Prozess, bei dem die Werte der globalen Variablen ersetzt werden (Schritte 8 und 9 im obigen Code), wird als Kontextumschaltung bezeichnet.

Wie Sie anhand des obigen Codes leicht sehen können, ergeben sich die folgenden beiden Hauptprobleme:

- Neuberechnungen werden für alle Cluster-Objekte zusammen ausgeführt. Der Zeitpunkt T (Schritt 3 im obigen Code) kommt nur ein Mal vor und alle Cluster-Objekte führen dann eine Neuberechnung basierend auf diesem Zeitpunkt durch. Dies bedeutet, dass jedes Mal, wenn ein einzelnes Cluster-Objekt über neue Events verfügt, alle Cluster-Objekte neu berechnet werden.

- Eine Kontextumschaltung kommt sehr häufig vor. Dies ist leicht zu erkennen, da die Kontextumschaltung (Schritte 8 und 9 im obigen Code) innerhalb der inneren Schleife erfolgt.

Neuberechnung von geclusterten Metriken

- Problembeschreibung

  Wie bereits erläutert, werden alle Cluster-Objekte in einer geclusterten Metrik als Ganzes neu berechnet. Dies bedeutet Folgendes: Wenn Sie eine Metrik haben, die in über 1000 Cluster-Objekte geclustert ist, und eines dieser Objekte einer Neuberechnung des letzten Jahres bedarf, da einige neue Events empfangen wurden, dann werden alle 1000 Cluster-Objekte für das letzte Jahr neu berechnet.

- Mögliche Lösungen

  Die folgenden Lösungsvorschläge können die Auswirkungen dieses Problems reduzieren. Sie sind jedoch nicht immer anwendbar und jeder Vorschlag hat auch seine eigenen Nachteile. Es ist wichtig, das Problem und seine geschätzten Kosten zu verstehen, und diese Kosten mit den Kosten der vorgeschlagenen Lösungen zu vergleichen.

  - Wenn die Anzahl der Cluster-Objekte klein ist, können Sie die Option in Betracht ziehen, bei der Sie jedes Objekt als gesonderte Metrik definieren. Ein Nachteil dieser Vorgehensweise sind natürlich die Wartungskosten für die Wartung mehrere Metriken. Darüber hinaus ist auch die Tatsache nachteilig, dass Sie nicht einen Bericht für die ganze Metrik erstellen und diesen in bestimmte Cluster-Objekte aufgliedern können
  - Wenn die Anzahl der Cluster-Objekte groß ist, aber nur eines (oder wenige) von ihnen häufig neu berechnet wird (werden), können Sie möglicherweise dieses Cluster-Objekt in eine gesonderten Metrik einfügen und alle anderen Cluster-Objekte in der anderen Metrik belassen
  - Verwenden Sie häufig einen Berechnungsstopp für den zugehörigen Vertrag bzw. die zugehörige Metrik, sodass diese Metrik nie lange Neuberechnungen ausführt
  - Führen Sie einige Änderungen in den Adaptern und den Datenquellen aus, sodass es keine langen Neuberechnungen gibt (d. h. senden Sie keine Events, deren Zeitstempel älter als einen Monat ist)

Kontextumschaltung

- Problembeschreibung

  Wie bereits erläutert, erfolgt die Kontextumschaltung in der innersten Schleife. Mit anderen Worten, für jedes Event, das von jedem Cluster-Objekt verarbeitet werden sollte. Wenn eine Metrik viele Events empfängt und jedes Event von vielen Cluster-Objekten verarbeitet wird, kann dieser Wert sehr hoch sein. Hinzu kommt noch, dass der Kontextumschaltungsvorgang relativ kostenintensiv ist (im Vergleich zur Verarbeitung des Events selbst in der Business-Logik), und schon haben Sie ein Problem.

Die Kosten des Kontextumschaltungsvorgangs sind proportional zur Größe der Daten, die "umgeschaltet werden". Die Daten, die Sie während der Kontextumschaltung umschalten, sind die Werte der globalen Variablen in Ihrer Business-Logik (auch "der Status" genannt). Je mehr globale Variablen Sie also haben und je größer die Größe dieser globalen Variablen ist, desto kostenintensiver ist der Kontextumschaltungsvorgang.

Es wird vor allem nicht empfohlen, Zuordnungsgrafiken für die Business-Logik in geclusterten Metriken zu verwenden, insbesondere dann, wenn die Größe dieser Zuordnungsgrafiken sehr groß sein kann.

- Mögliche Lösungen
  - Reduzieren Sie die Zeit der Kontextumschaltungen

    Die Idee dahinter ist, die Statusgröße zu reduzieren (globale Variable). Dies kann ausgeführt werden, indem die Business-Logik neu geschrieben wird, sodass es keine Zuordnungsgrafiken enthält. Natürlich ist dies nicht immer möglich, aber sollte das der Fall sein, wird diese Vorgehensweise empfohlen.

  - Reduzieren Sie die Anzahl der Kontextumschaltungen

    Wenn der Cluster klein ist, ist es möglich, eine gesonderte Metrik für jedes Cluster-Objekt zu erstellen.

    Vermeiden Sie geclusterte Metriken mit vielen geclusterten Objekten, die für dasselbe Event registriert werden. Die Idee dahinter ist folgende:

    Wenn jedes Event von einem einzelnen Cluster-Objekt verarbeitet wird, dann steigt die Anzahl der Kontextumschaltungen proportional zur Anzahl der Events

    Wenn jedes Event von allen Cluster-Objekten verarbeitet wird, dann steigt die Anzahl der Kontextumschaltungen proportional zur Anzahl der Events mal die Anzahl der Cluster-Objekte

    Erstellen Sie eine nicht geclusterte Metrik, die die Ergebnisse für alle ursprünglichen Cluster-Objekte (die jetzt einfache Ressourcen und nicht Cluster-Objekte sind) berechnet. Gestalten Sie diese Metrik so, dass sie das Ergebnis der Cluster-Objekte in Form eines Events sendet. Erstellen Sie eine weitere Metrik, die geclustert ist, die Events von der ersten Metrik empfängt und den Wert überträgt, der in diesen Events als Ergebnis empfangen wurde. Die Idee dahinter ist, dass die große Anzahl an Rohdaten-Events von einer nicht geclusterten Metrik verarbeitet wird, während die geclusterte Metrik ein einzelnes Event pro Zeitraum und Cluster-Objekt verarbeitet.

Fallstudie 16: Designmuster der Business-Logik

Es gibt mehrere "Designmuster", die in vielen Business-Logiken verwendet werden können. Diese Designmuster sind geprüft und ihre Verwendung, dort wo es möglich ist, kann viel Zeit sparen und in vielen Fällen zu einer effizienteren Business-Logik führen. Diese Fallstudie befasst sich schwerpunktmäßig mit solch einem Designmuster.

Designmuster "Zähler aktualisieren"

Dieses Designmuster ist in beinahe allen Business-Logiken hilfreich, die dafür vorgesehen sind, die Zeit zwischen bestimmten Events zu messen. Beispiele für derartige Business-Logik sind: Verfügbarkeitsmessung, Ausfallzeit, mittlere Zeit zwischen Ausfällen, mittlere Reparaturzeit, durchschnittliche Antwortzeit, durchschnittliche Bearbeitungszeit, Prozentsatz von Komponenten mit einer Verfügbarkeit kleiner als X, Anzahl von nicht rechtzeitig gelösten Fällen usw.

Diese Arten von Business-Logik haben gemeinsam, dass ihr Ergebnis vom Zeitstempel verschiedener Events abhängt, die sie empfangen.

Eine Faustregel dafür, ob Ihre Business-Logik von diesem Designmuster profitieren kann, ist: Wenn Ihre Business-Logik vom Zeitstempel der verschiedenen Events abhängt, die sie empfängt, sollte sie dieses Designmuster verwenden.

Aufbau dieses Designmusters

Der Code einer Business-Logik, die dieses Muster nutzt, besteht aus zwei Teilen: einem Rahmen und einer Implementierung. Der Rahmen enthält Code, der in den meisten Fällen vorgegeben und für die verschiedenen Arten von Business-Logik gleich ist. Dieser Teil ist für die Berechnung der Verfügbarkeit und der Anzahl von nicht rechtzeitig gelösten Tickets gleich. Die Implementierung enthält Code, der spezifisch für jede Business-Logik ist.

Es wird empfohlen, diese beiden Codeteile in separaten Business-Logik-Modulen zu nutzen und das Modul des Rahmens in unterschiedlichen Metriken wiederzuverwenden.

Code des Rahmens:

```
Dim g_PrevEventTimestamp

Sub OnLoad(time)
     g_PrevEventTimestamp = time
     InitializeStatusVariables
End Sub

Sub OnRegistration(Dispatcher)
     ' If there is a separate copy of status variables
     ' for each registered resource depend on the registered
     ' resources you should set initial values to the
     ' status variables of the newly added resources here
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
```

Aufbau des Implementierungsmoduls:

```
' Define your status variables here. This can be one
' simple global variable or many complex global variables
' depending on the business logic
Dim g_StatusVar_1, g_StatusVar_2, ... ,g_StatusVar_n

' Define your counters here.
' This can be one simple global variable or many
' complex global variables depending on the business logic
Dim g_Counter_1, g_Counter_2, ... , g_Counter_n

Sub InitializeStatusVariables ()
     ' Set initial values to the various status variables
End Sub

Sub InitializeCounters ()
     ' Set initial values to the various counters
     g_Counter_1 = 0
```

```
     g_Counter_2 = 0
     '…
     g_Counter_n = 0
End Sub

Function CalculateResult ()
     ' Calculate the result. The result should depend on
     ' the values of the counters. It should not depend on
     ' the value of the status variables. It should not
     ' change the values of the counters or the status
     ' variables
End Function

Sub UpdateStatus(method, eventDetails)
     ' Update the value of the status variables based on
     ' the parameters (and posibly on the old value of the
     ' status variables)
End Sub

Sub UpdateCounters(diff)
     ' Update the values of the counters based on their
     ' previous value, on the value of the status variables
     ' and on the value of the diff parameter.
     ' In many cases this calculation is also based on the
     ' value of Kontext.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
```

Um dieses Designmuster besser zu erklären, finden Sie hier ein Beispiel für die Implementierung dieses Musters für die Berechnung der Verfügbarkeit. In diesem Beispiel wird davon ausgegangen, dass Events für UP und DOWN vorhanden sind, die von zwei separaten Event-Routinen in der Business-Logik verarbeitet werden. Die Verfügbarkeit wird als Prozentsatz der Zeit angegeben, in der das System während der Gesamtzeit im Zeitfenster betriebsbereit ist. Der Status des Systems ist der Status des letzten empfangenen Events (UP oder DOWN).

Code der Implementierung (Code des Rahmens ist gleich):

```
' Status variable
Dim g_Status
' Counters.
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
```

Es gibt einige Variationen dieses Musters. Eine der gängigsten Variationen ist, dass ein separater Zeitzähler für unterschiedliche Entitäten verwaltet werden soll. Beispiel: Wenn Sie die Bearbeitungszeit messen möchten, sollte ein separater Zähler für jedes offene Ticket verwaltet werden. Wenn ein nur für ein Ticket relevantes Event verarbeitet wird, ist es in diesem Fall effizienter, nur den Zähler dieses Tickets zu aktualisieren. Wenn ein gängiges Event verarbeitet wird (wie "OnPeriodEnd" oder "OnTimeslotEnter"), sollten die Zähler aller Tickets aktualisiert werden.

Hinweis: Für diese Mustervariation muss eine separate Kopie der globalen Variable "g_PrevEventTimestamp" für jedes Ticket verwaltet werden.

Einige gute Beispiele für die Verwendung dieses Musters finden Sie in unseren vordefinierten Inhalten. Beachten Sie, dass dieses Muster in den vordefinierten Inhalten anders verwendet wird und die Trennung zwischen Rahmen und Implementierung nicht so deutlich ist.

Fallstudie 17: Integrierte Funktionalität

ACE bietet integrierte Funktionalität und Tools für verschiedene Zwecke. Diese integrierte Funktionalität sollte in VBS geschrieben werden. Da VBS eine interpretierte Sprache ist, führt die Reproduzierung in VBS zu einer Verschlechterung der Leistung.

Im Folgenden finden Sie eine Liste der integrierten Funktionen und der entsprechenden Verwendungsweise:

IsWithinTimeslot

Dies ist die einfachste der integrierten Funktionen. Sie aktiviert die Business-Logik, die angibt, ob sich das System gegenwärtig in einem Zeitfenster befindet. Das heißt, es muss keine Variable in den Funktionen für den Eintritt in das Zeitfenster und das Verlassen des Zeitfensters verwaltet werden. Beispiel: Statt folgenden Code auszuführen:

```
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
```

können Sie diesen viel einfacheren Code ausführen:

```
Sub OnEvent(eventDetails)
     If context.IsWithinTimeslot Then
          count = count + 1
     End if
End sub
```

Wenn Sie Informationen zum Zeitstempel von Zeitfenstereintritt und Verlassen des Zeitfensters verwenden oder beibehalten möchten, ist diese Funktionalität nicht geeignet. Normalerweise ist dies aber nicht erforderlich, und dieser Code ist ausreichend.

TimeOfLastEvent

Diese Funktion gibt den Zeitstempel der letzten verarbeiteten Rohdaten oder des letzten verarbeiteten Zwischendaten-Events aus. Das heißt, dass Sie diese Informationen nicht im Event-Handler speichern müssen, da sie direkt über diese Funktion verfügbar sind. Zum Beispiel:

```
Function result
     Dim LastEventTimestamp
```

```
    LastEventTimestamp = Kontext.TimeOfLastEvent
End function
```

TimeOfLastEventHandler

Diese Funktion gibt den Zeitstempel der letzten von ACE aufgerufenen Event-Handler zurück. Dies schließt nicht nur Event-Handler für Roh- und Zwischendaten-, sondern auch System-Events ein, die ebenfalls aufgerufen wurden. Dies ist besonders nützlich bei Event-Handler, die die Zeit z. B. für die Ergebnisfunktion nicht empfangen. Zum Beispiel:

```
Function result
    Dim LastEventHandlerTimestamp
    LastEventHandlerTimestamp= Context.TimeOfLastEventHandler
End function
```

NetTime

Diese Funktion erlaubt es Ihnen, zwei Zeitstempel und die Nettozeit (in Sekunden) anzugeben, in der sich das System innerhalb des Zeitfensters für die aktuelle Regel zwischen diesen beiden Zeitstempeln befunden hat. Dies ist eine besonders aufwändige Funktionalität, die nicht in VBS implementiert werden sollte. Die Implementierung in VBS würde eine Liste aller Zeitfenstereintritte und -austritte beinhalten oder die direkte Berechnung des Unterschieds zwischen den Zeiten der Zeitfenstereintritte und -austritte, um die Zeitspanne dazwischen zu ermitteln. Unter extremen Bedingungen könnte dies oft geschehen und die Berechnungsleistung negativ beeinflussen. Mit der internen Funktion erreichen Sie nach umfassender Optimierung dasselbe, sind nur effizienter. Zum Beispiel:

```
Function result
    Dim MyNetTime
    MyNetTime = Tools.NetTime(MyBeginTimestamp, MyEndTimestamp)
End function
```

Das Kontextobjekt

Das Kontextobjekt umfasst verschiedene Parameter, die Informationen bieten zu:

- der aktuellen Metrik
- dem entsprechenden Vertrag
- dem aktuellen Berechnungsstatus
- Servicedomäne, Domänenkategorie und ihren Werten (z. B. Grenzwert)
- Cluster-Informationen, d. h. dem Cluster im Allgemeinen und dem spezifischen Cluster-Objekt, das verarbeitet wird
- Funktionen, die Listen von Ressourcen basierend auf Anwenderanforderungen zurückgeben
- Funktionen, die es Ihnen erlauben, Ressourcennamen in Ressourcen-IDs zu konvertieren und umgekehrt

Der Zugriff auf diese Informationen direkt über die Datenbank mit Safe ODBC ist nicht sehr effizient und nicht sinnvoll, da die Informationen bereits über das Kontextobjekt verfügbar sind. Verwenden Sie nach Möglichkeit immer die integrierte Funktionalität, um Informationen abzurufen.

Fallstudie 18: Registrierung

Business-Logik wird oft so geschrieben, dass eine Übersicht der Ressourcenstruktur der Metrik für Berechnungen beibehalten wird. Da die Ressourcenstruktur sich im Laufe der Zeit ändert, muss die Business-Logik die Struktur in der Übersicht aktualisieren.

Die Methode "OnRegistration" wird aufgerufen, wenn die Ressourcenstruktur sich ändert, da sie dafür verantwortlich ist, das Engine-Verhalten zu verwalten, das mit den Änderungen in den Registrierungen und dem Clustering aufgrund von Ressourcenstrukturänderungen verknüpft ist. Dadurch, dass diese Methode für jede Ressourcenstrukturänderung aufgerufen wird, ist sie geeignet, um die oben erwähnte Übersicht zu aktualisieren. Das Befüllen der Übersicht ist jedoch nicht für den Registrierungsprozess relevant. Das bedeutet, dass beim Befüllen der Übersicht die Leistung der Funktion "OnRegistration" beeinträchtigt wird. Dies spielt während der Laufzeit keine Rolle, da es normalerweise nicht sehr oft geschieht. Allerdings wird die Methode "OnRegistration" auch während des Infrastrukturverarbeitungsprozesses der Engine aufgerufen. Während des Prozesses ermittelt das System, ob Ressourcenstrukturänderungen für die Registrierung jeder spezifischen Metrik relevant sind, für die die Instanz verantwortlich ist. Während dieses Prozesses wird die Methode "OnRegistration" für jede Änderung in der Ressourcenstruktur aufgerufen, auch wenn die Strukturänderung für die aktuelle Metrik nicht relevant ist. Das heißt, dass die Methode pro Metrik oft aufgerufen wird.

Wenn eine derartige Logik in der Methode "OnRegistration" implementiert wird, kann aus einer geringen Leistungsverschlechterung während der Laufzeit eine erhebliche Leistungsverschlechterung während der Infrastrukturverarbeitung werden.

Um dieses Problem zu lösen, haben Sie zwei Möglichkeiten, Übersichten zu befüllen oder andere Initialisierungen durchzuführen (bei einer Änderung in der Ressourcenstruktur, die nicht für die Registrierung relevant ist):

Verwenden der Eigenschaft "IsRunTimeMode" im Dispatcher-Objekt

Diese Eigenschaft erlaubt es dem Anwender, zu ermitteln, ob es sich bei der aktuellen Ausführung um eine Berechnung handelt, und eine Logik, die nicht relevant für die Registrierung ist, in einer IF-Anweisung einzuschließen. Dies stellt sicher, dass sie nur während der Laufzeit ausgeführt wird.

Im unteren Beispiel ist der blau markierte Teil der Business-Logik der Teil, der relevant für die Registrierung ist und immer ausgeführt werden muss. Der grün markierte Teil ist für die Registrierung nicht relevant und kann in der neuen IF-Anweisung eingeschlossen werden.

```
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)
     For Each resource In ThisResourceMap
          GlobalResourceVector.Add-Ressource
     Next
End Sub
```

Dieser Code kann verbessert werden, indem er folgendermaßen geändert wird:

```
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 = Kontext.ResourcesOfResourceGroup(Kontext.ClusterItem)
          For Each resource In ThisResourceMap
               GlobalResourceVector.Add-Ressource
          Next
     End If
End Sub
```

Verwenden der Methode "OnResourceStructureChanged"

Diese Methode wird direkt nach der Methode "OnRegistration" ausgeführt und weist dieselbe Funktionalität wie die ursprüngliche Methodik auf. Sie wird jedoch nur während der Laufzeit ausgeführt. Diese Methode wird nicht während der Infrastrukturverarbeitung aufgerufen, um die Leistung nicht zu beeinträchtigen.

Im unteren Beispiel ist der blau markierte Teil der Business-Logik der Teil, der relevant für die Registrierung ist und in der Methode "OnRegistration" beibehalten werden muss. Der grün markierte Teil ist für die Registrierung nicht relevant und kann in der neuen Funktion platziert werden.

```
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)
     For Each resource In ThisResourceMap
          GlobalResourceVector.Add-Ressource
     Next
End Sub
```

Dieser Code kann verbessert werden, indem er folgendermaßen geändert wird:

```
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)
     For Each resource In ThisResourceMap
          GlobalResourceVector.Add-Ressource
     Next
End Sub
```

Fallstudie 19: Adapterassistent für dateibasierte Datenquelle

Diese Fallstudie prüft die Best Practice-Methoden für die Integration mit einer dateibasierten Datenquelle. Das Beispielszenario verarbeitet eine CSV-Datendatei, die vom Quellsystem erstellt wird. Für die meisten dateibasierten Integrationen empfiehlt CA die Einhaltung einiger grundlegender Richtlinien, um das Risiko bei der Integration zu minimieren. Zu diesen Richtlinien zählen folgende:

- Wenn die Option verfügbar ist, sollten die Daten dem Dateisystem des CA Business Service Insight-Applikationsservers bereitgestellt werden. Dies stellt sicher, dass der Bereitstellungsmechanismus nicht vom Adapter abhängig ist, der versucht, Daten von einem Remote-Speicher abzurufen (minimiert Berechtigungsprobleme mit Anwenderkonten und Synchronisationsprobleme usw.)
- Die Dateinamenskonvention ist wichtig, da der Adapter die Dateien entsprechend der alphanumerischen Namensgebung ordnet. Wenn Sie dies steuern können, empfiehlt CA Folgendes:
  - Eine sinnvolle Namenskonvention, die auf dem Quelldateiinhalt basiert (besonders, wenn mehr als eine Datei von der Quelle stammt)
  - Ein Zeitstempel in umgekehrter Reihenfolge, um sicherzustellen, dass die Dateien so geordnet werden, dass die aktuellste Datei zuletzt aufgelistet wird (d. h. *<file_name>_yyyymmdd_hhmiss.csv*). Die Tiefe des verwendeten Zeitstempels ist von der Häufigkeit der bereitgestellten Daten abhängig.

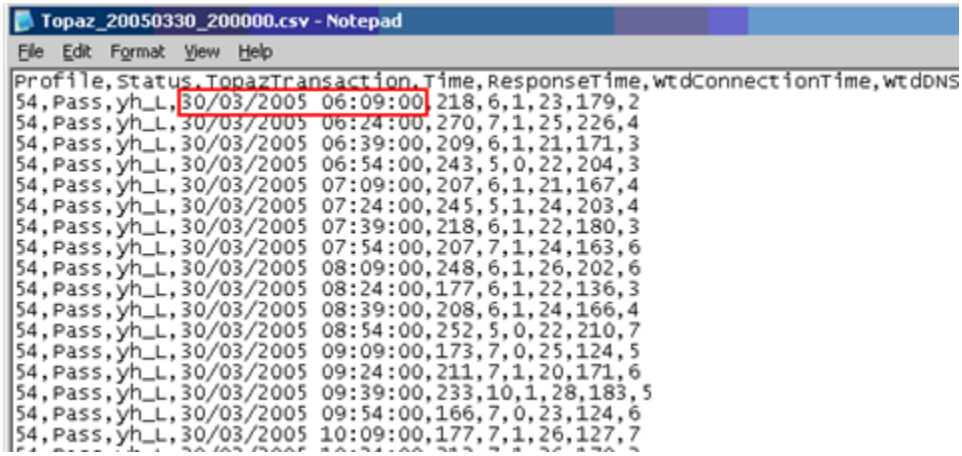
In diesem Szenario stammen die Quelldateien von einer Topaz-Datenquelle (jetzt Mercury Global Monitor, HP). Dies wurde mit einer API des Produkts erstellt, um die erforderlichen Dateien für die benötigten spezifischen KPIs einzuschließen. Die Dateien werden dem CA Business Service Insight-Applikationsserver direkt von einem externen automatisierten Prozess bereitgestellt. Die Quelldateien werden wie folgt benannt: *Topaz_yyyymmdd_hhmiss.csv*.

Hinweis: Der Zeitstempel der Datei ist der Zeitpunkt ihrer Erstellung, das heißt, alle Eingaben in der Datei werden berücksichtigt.

Ein Beispiel für die Daten in der Datei finden Sie unten.

	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	

Hinweis: CA empfiehlt, CSV-Dateien in Notepad zu überprüfen (statt nur in Excel), um sicherzustellen, dass das Datumsformat Ihren Erwartungen entspricht. In Excel werden Datumsangaben gemäß den regionalen Einstellungen des Computers formatiert, sodass sie dem tatsächlichen, dem Adapter bereitgestellten Format eventuell nicht entsprechen.



Bevor Sie Adapter erstellen, ist es auch wichtig, dass Sie die Datenquelle und die verknüpften KPIs ausreichend analysieren und prüfen, um Folgendes zu ermitteln:

- Welche Felder werden von der Business-Logik benötigt?
- Welches Datumsformat weist die Datei auf?
- Welche Zeitzone für die Datums-/Uhrzeitwerte liegt in der Quelldatei vor? (Diese Zeitzonen sollten vor der Adaptererstellung im System erstellt werden.)
- Sind Datumsfelder mit leeren Einträgen/NULL-Einträgen vorhanden?
- Wie verhält sich die Datenquelle? (Werden alle Datensätze hinzugefügt, oder sind einige Datensätze eine Aktualisierung eines früheren Events?)
- Welche Verfeinerungen werden für die KPIs benötigt? (Dies kann sich auf die Wahl der "Ressource" auswirken.)
- Welche "Event-Typen" sind für das effektive Filtern der Events in der Business-Logik erforderlich?

Wenn Sie dies wissen, können Sie mit der Erstellung beginnen.

Nun kann auf der Basis dieses Szenarios die Adaptererstellung mit dem Adapterassistent durchgeführt werden.

In unserem Szenario nutzen wir "TopazTransaction" als Ressource, "Profil" als Event-Typ und das Feld "Zeit" als Zeitstempel. Wir schließen auch die Felder "Status", "ResponseTime" und "WtdDownloadTime" in die Event-Struktur für die Business-Logik ein.

Adaptererstellung

Stellen Sie zunächst sicher, dass das System bereit ist, den neuen Adapter zu erstellen und ihn auf dem Server ordnungsgemäß bereitzustellen, indem Sie prüfen, ob folgende Services auf dem Applikationsserver gestartet werden.

- Adapter-Bereitstellungsservice
- Adapter-Listener-Service

Navigieren Sie anschließend zur Seite "Adapter", und erstellen Sie einen neuen Adapter. Wählen Sie auf der Seite "Adapter" die Option "Neu hinzufügen" > "Textdatei-Adapter".

Schritt "Allgemein"

Geben Sie im Schritt "Allgemein" des Adapterassistenten folgende Felder ein:

- *Name*: Geben Sie einen geeigneten Namen für den Adapter an.
- *Adapteradresse*: LOCALHOST ist die Standardoption (für die Bereitstellung des Anwendungsservers). Sie können mithilfe der Schaltfläche jedoch auch andere Adressen eingeben, falls erforderlich.
- *Zeitformat*: Dies ist das in Datums-/Uhrzeitfeldern in der Datenquelle verwendete Standardzeitformat. Bei einer ordnungsgemäßen Auswahl wird sichergestellt, dass die Felder später vom Assistenten automatisch korrekt erkannt werden. Neue Zeitformate können jetzt mithilfe der Schaltfläche neben diesem Feld eingegeben werden, falls erforderlich.
- *Zeitzone*: Dies ist die Zeitzone, in der die Datensätze aufgezeichnet werden. Sie wird benötigt, damit der Adapter im Feld "event_timestamp" (und anderen Datums-/Uhrzeitfeldern) korrekt auf UTC umstellen kann (korrekter interner Zeitunterschied). Sie hätte bereits vorher gemäß der Datenquellcheckliste im vorherigen Abschnitt eingegeben werden sollen.

Hinweise:

Über die Schaltfläche "Erweitert" auf dieser Seite haben Sie Zugriff auf verschiedene Konfigurationsparameter für den Adapter. Für die meisten dieser Parameter können die Standardwerte beibehalten werden, es sei denn, Sie möchten Änderungen vornehmen.

Der Adapter-"Port" wird dem Adapter ab CA Business Service Insight automatisch zugewiesen, kann hier jedoch überschrieben werden (falls erforderlich). Andere wichtige Parameter, die geändert werden können, sind beispielsweise: Regionale Einstellungen, Online/Offline-Modus, Verbindungsdetails, Überwachungs- und Protokollierungsoptionen, Ausführungseinstellungen (einmal/immer), Fehlergrenzen, Dateinamen und Kommentare.

Diese Einstellungen werden nicht in dieser Fallstudie behandelt. Sie finden sie unter Eigenschaften der Adapterkonfiguration (siehe Seite 321).

Klicken Sie auf "Weiter", um mit dem nächsten Schritt des Assistenten fortzufahren. Der nächste Schritt ermöglicht den Zugriff auf die Schnittstelle der Datenquelle des Adapters.

Schritt "Schnittstelle der Datenquelle"

Geben Sie im Schritt "Schnittstelle der Datenquelle" des Adapterassistenten folgende Felder ein:

- *Datenquellenname*: Ein Name für diese bestimmte Quelldatei. (Ein Adapter kann mehrere Quelldateien aufweisen.)
- *Dateipfad*: Der Pfad auf dem Anwendungsserver (oder einem anderen Server), der die Quelldatendateien enthält. Verwenden Sie für andere Server als den Anwendungsserver eine UNC-Notation (d. h. //server01/sourcefolder)
- *Namensmuster*: Verwenden Sie das Namensmuster mit einem Platzhalterzeichen, um die Dateien zu filtern, die sich unter dem "Dateipfad" befinden und vom Adapter geladen werden.

Über die Registerkarte "Parsing-Definition" können Sie die Struktur der importierten Datei definieren. Die Felder können folgendermaßen verwendet werden:

- *Titel*: Kontrollkästchen (boolesche Werte), mit dem Sie angeben können, ob die Datendatei eine "Titel"-Zeile aufweist (d. h. die erste Zeile in der CSV ist ein Titelname, gefolgt von den Werten in den nachfolgenden Zeilen)
- *Trennzeichen*: Gibt das Trennzeichen in der Datei an, das einzelne Felder trennt.

Note: Es sind zwei weitere "Erweitert"-Schaltflächen vorhanden, über die Sie auf zusätzliche Konfigurationsoptionen zugreifen können. Eine auf der Registerkarte "Details" und eine auf der Registerkarte "Parsing-Definition". Über die Schaltfläche "Erweitert" auf der Registerkarte "Details" haben Sie Zugriff auf folgende Parameter:

Ist aktive Datenquelle: Boolesches Kontrollkästchen, über das Sie diesen bestimmten Quellbereich des Adapters aktivieren/deaktivieren können.

Nach Verarbeitung: Erlaubt es Ihnen, anzugeben, ob die Datei nach der Verarbeitung gelöscht oder gespeichert werden soll.

Anfänglicher Dateiname: Legt den anfänglichen Dateinamen fest, ab dem die Verarbeitung gestartet werden soll (wenn Sie nicht alle Dateien in einem bestimmten Verzeichnis laden möchten)

Neue Daten prüfen alle: Definiert das Zeitintervall, in dem auf neue Dateien geprüft wird, wenn der Adapter kontinuierlich ausgeführt wird

Über die Schaltfläche "Erweitert" auf der Registerkarte "Parsing-Definition" können Sie die Endoptionen des Datensatzes definieren, wie mehrzeilige Datensätze usw.

Nach dem Festlegen der Details und Parsing-Definitionen können Sie eine Beispiel-Quelldatei in den Assistenten laden, um die Konfigurationsoptionen zu testen und eine Vorschau der Dateiinhalte anzeigen.

Wenn Sie auf die Schaltfläche "Durchsuchen" klicken, kann die Beispieldatei in das Vorschaufenster unten geladen werden. Navigieren Sie zur Beispieldatei, und klicken Sie dann auf die Schaltfläche "Datei testen". Dies ist ein optionaler Schritt.

Hinweis: Auf dem lokalen Computer, auf dem Sie arbeiten, wird die Funktion "Durchsuchen" geöffnet. Wenn dies nicht der Anwendungsserver ist, muss eine Kopie der Quelldateien auf dem Server vorhanden sein. Sie können mit dieser Funktion auf dem Server nicht direkt durchsuchen.

Nach dem Laden der Daten sollten die Inhalte im Vorschaufenster, wie unten dargestellt, angezeigt werden.

Hinweis: Der "Feldname" wird zusammen mit dem identifizierten Feldtyp (Ganzzahl, String, Zeit usw.) im Header angezeigt. Sie sollten prüfen, ob diese korrekt und gemäß Ihren Anforderungen ermittelt wurden, bevor Sie fortfahren. Stellen Sie Folgendes sicher:

- Ihr Zeitstempel wurde als "Zeit" identifiziert. Dies erfolgt gemäß dem "Zeitformat", das im ersten Schritt des Assistenten angegeben wurde.
- Ihre Ressource wurde als "String" identifiziert.

Wenn Sie mit der Dateivorschau zufrieden sind, klicken Sie auf "Weiter". Der Schritt "Zuordnen" wird angezeigt.

Schritt "Zuordnen"

Im nächsten (und letzten) Schritt im Adapterassistenten werden die Übersetzungsaufgaben ausgeführt. Sie können die Eingabefelder aus der Datenquelle den Ausgabefeldern zuordnen (CA Business Service Insight-"Event"). Sie haben dann zwei Möglichkeiten, je nachdem, ob der "Event-Typ" bereits im System erstellt wurde. Ferner sind weitere Optionen vorhanden, die Sie konfigurieren können. Und Sie können sicherstellen, dass die Benennungsstandards eingehalten werden. Die meisten dieser Optionen sind jedoch optional, um den Prozess zu vereinfachen und die erforderlichen Schritte zu reduzieren. Obligatorische Schritte sind unten aufgeführt.

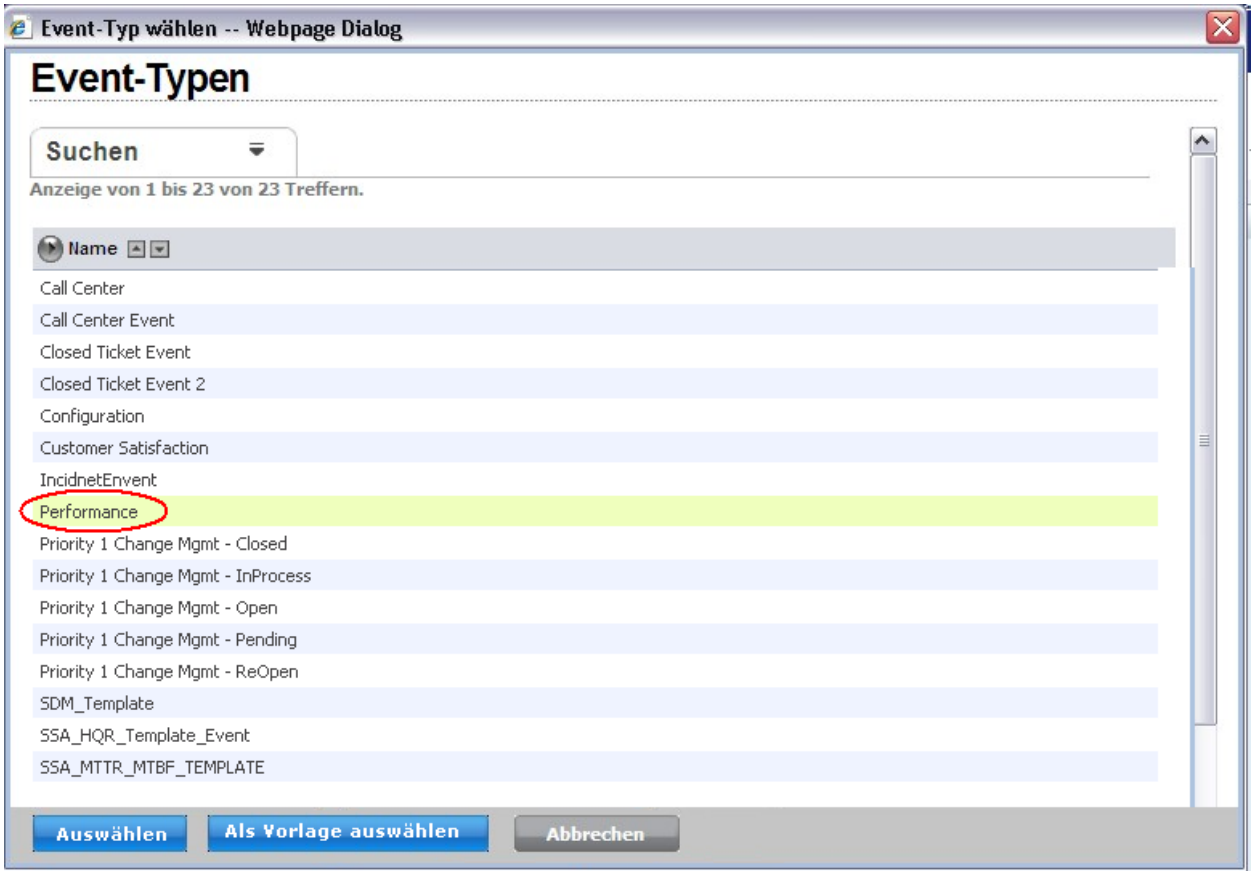
Konfiguration der Zuordnung (einschließlich der optionalen Schritte):

1. Geben Sie einen Namen für das Eingabeformat an (nützlich für Adapter mit mehreren Eingaben).
2. Fügen Sie zusätzliche erforderliche Felder hinzu (wie konstante Werte, Datenquelle, Titel, Dateiname oder Kombifeldtypen)
3. Erstellen Sie erforderliche Eingabeprüfkriterien.
4. Wählen Sie einen vorhandenen Event-Typ aus, oder erstellen Sie einen neuen Event-Typ (obligatorisch).
5. Führen Sie die Zuordnung der Felder von der Eingabe zur Ausgabe durch (obligatorisch).
6. Geben Sie einen Namen für das Ausgabeformat an.
7. Führen Sie die Zuordnung für "ResourceId", "Zeitstempel" und "Event-Typ" durch.
8. Erstellen Sie erforderliche Ausgabeprüfkriterien.
9. Geben Sie die Einstellung "OnDuplication" für die Events an.

Wenn Sie einen neuen Event-Typ erstellen möchten, wird ein neues Fenster angezeigt und basierend auf dem Eingabeformat im Hauptbildschirm vorab befüllt. Sie müssen noch einen Namen für den Event-Typ eingeben und außerdem dem Event-Typ einen Ressourcentypen zuweisen.

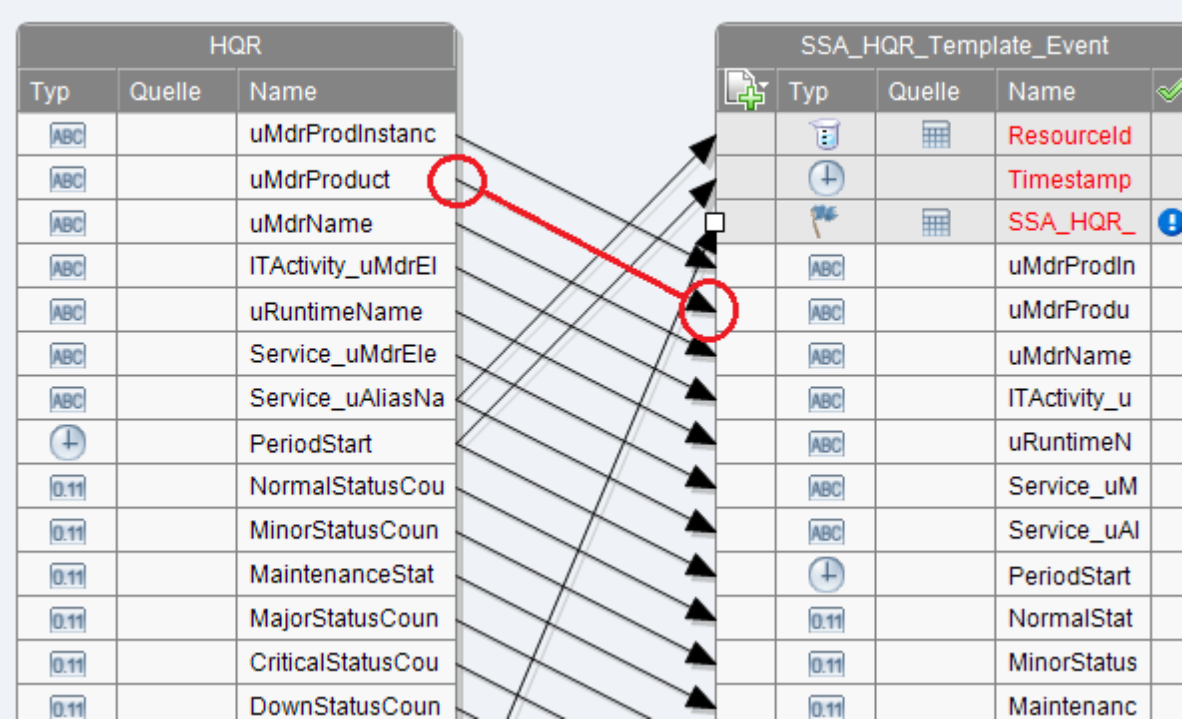
Anschließend können Sie mit "Speichern und Beenden" speichern und den Vorgang beenden. Die Zuordnung wird automatisch abgeschlossen.

Wenn Sie die Option "Neuen Event-Typ wählen" wählen, wird eine Liste der vorhandenen Event-Typen im System angezeigt, aus der Sie wählen können. Es werden jedoch nur Felder aus der Eingabe mit übereinstimmendem Namen und Datentyp des Event-Typs automatisch verknüpft. Die übrigen Felder müssen manuell zugeordnet werden.

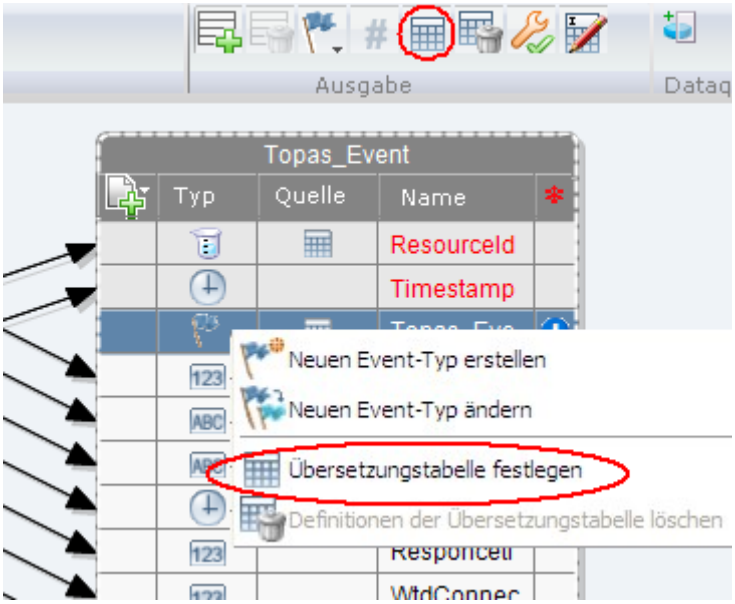

Event-Typ wählen -- Webpage Dialog
Event-Typen
Suchen
Anzeige von 1 bis 23 von 23 Treffern.
Name
Call Center
Call Center Event
Closed Ticket Event
Closed Ticket Event 2
Configuration
Customer Satisfaction
IncidnetEnvent
Performance
Priority 1 Change Mgmt - Closed
Priority 1 Change Mgmt - InProcess
Priority 1 Change Mgmt - Open
Priority 1 Change Mgmt - Pending
Priority 1 Change Mgmt - ReOpen
SDM_Template
SSA_HQR_Template_Event
SSA_MTTR_MTBF_TEMPLATE
Auswählen
Als Vorlage auswählen
Abbrechen

Der nächste Schritt umfasst die Konfiguration von "ResourceId", "Zeitstempel" und "Event-Typ". Wenn die Felder bereits vorhanden sind (falls nicht, sollten sie in den vorherigen Schritten erstellt werden), können Sie bei Bedarf mit diesen Feldern verknüpft werden.

Die Oberfläche unterstützt Drag-and-drop für die Verknüpfung. Wenn die Verknüpfung nach der Einstellung nicht bestehen bleibt, stellen Sie sicher, dass der Typ übereinstimmt (d. h. Zeit/String/Ganzzahl usw.).



- "ResourceId" sollte gemäß den (bei der Analyse identifizierten) Anforderungen von einem "String"-Wert in der Eingabe zugeordnet werden.
- Als Zeitstempel sollte die Zeitvariable festgelegt werden, die die Uhrzeit angibt, zu der das Event stattgefunden hat. Diese sollte auch für die Berechnung verwendet werden.
- Der Event-Typ ist standardmäßig ein konstanter Wert (gemäß dem Feld "Datenquellenname" im vorherigen Bildschirm). Dies kann überschrieben werden, falls erforderlich. Dies ist erforderlich, wenn mehrere Events von diesem Adapter gesendet werden sollen, gemäß dem Wert eines spezifischen Feldes (d. h., Sie möchten ein anderes Event abhängig vom Wert eines Eingabefeldes senden). Klicken Sie dazu mit der rechten Maustaste auf die Event-Typ-Zeile (oder klicken Sie auf die obere Schaltfläche wie im unteren Screenshot dargestellt), und wählen Sie "Übersetzungstabelle festlegen".



Es wird ein Popup-Fenster angezeigt.

Wenn die Übersetzungstabelle für dieses Wertfeld bereits vorhanden ist, kann sie hier ausgewählt werden. Alternativ kann eine neue Übersetzungstabelle erstellt werden. Verwenden Sie dazu die Schaltfläche auf dem obigen Screenshot.

Nach der Erstellung der Tabelle müssen Sie angeben, welches der Eingabefelder den oben angegebenen "Quellfeldern" zugeordnet werden soll.

Wenn Sie für die Ressourcen des Adapters eine andere Übersetzungstabelle angeben möchten, ist dies der geeignete Zeitpunkt. Dies erfolgt mithilfe eines Prozesses, der dem Prozess für den Event-Typ ähnelt.

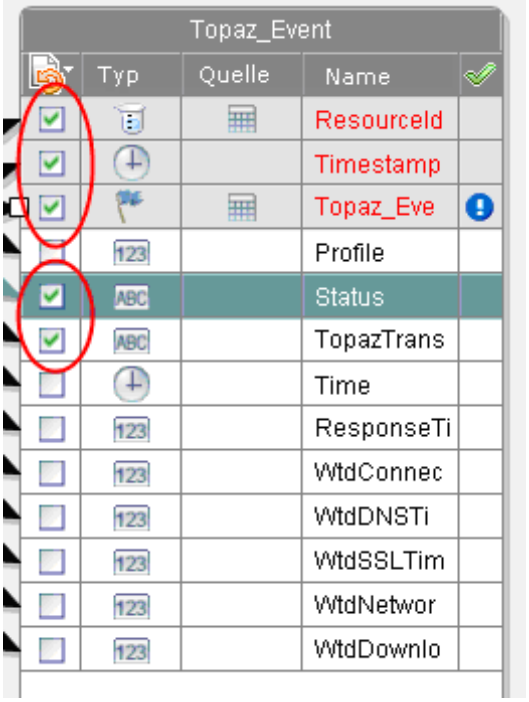
Hinweis: Standardmäßig weist der Adapterassistent alle Ressourcen einer vorhandenen Übersetzungstabelle namens "Default_Translation_Table" zu, wenn nicht anders angegeben. Dies ist für einfache Implementierungen sinnvoll. Für komplexere Implementierungen (und für die Datentrennung) empfiehlt CA die Verwendung einer anderen Tabelle. Dies ist auch obligatorisch, wenn sich die "Quellfelder" im Bereich der Adapterzuordnung unterscheiden oder mehr als einen Wert enthalten.

Der letzte Schritt im Schritt "Zuordnung" ist die Konfiguration der Einstellung "OnDuplication" für den Adapter. Diese Einstellung beschreibt die Aktion, die durchgeführt wird, wenn ein zweites Event, mit übereinstimmenden Werten für alle "Schlüsselfelder", vorhanden ist. Dieser eindeutige "Schlüssel" kann für jede Ausgabe des Adapters definiert werden. (Weitere Informationen dazu finden Sie unten). Standardmäßig ist für "OnDuplication" der Wert "Add" eingestellt. Das heißt, dies muss nur geändert werden, wenn eine andere Aktion erforderlich ist. Die verfügbaren Werte sind:

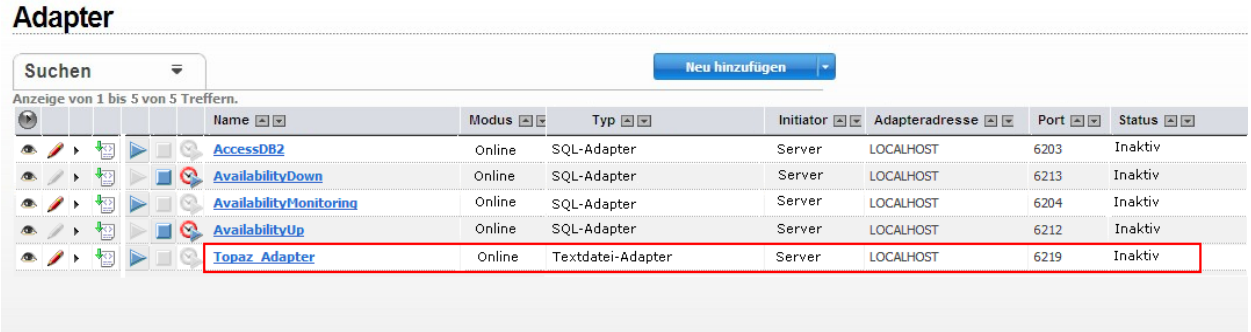
- *Add*: Das neue Event wird nur als separates neues Event hinzugefügt.
- *Ignore*: Das neue Event wird vom Adapter ignoriert (gelöscht).
- *Update*: Das neue Event ersetzt das vorher geladene Event nur, wenn sich die "Wert"-Felder des Events unterscheiden.
- *Update Always*: Das neue Event ersetzt das vorher geladene Event.

Wenn Sie andere Optionen als "Add" verwenden, kann sich dies auf die Leistung des Adapters auswirken. Dies sollten Sie vor der Implementierung berücksichtigen, besonders bei sehr großen Datenmengen.

Wenn Sie einen anderen Wert als "Add" verwenden, zeigt die Ausgabestruktur verschiedene Kontrollkästchen an, mit denen Sie den eindeutigen "Schlüssel" definieren. Der "Schlüssel" besteht aus allen Elementen und einer Prüfung. Die Wahl des Schlüssels sollte auf den Anforderungen basieren, die in einer umfassenden Analyse ermittelt wurden.



Klicken Sie nach der Konfiguration der Zuordnung unten rechts im Bildschirm auf die Schaltfläche "Fertig stellen". Sie kehren zur Liste der Adapter im System zurück. Der von Ihnen erstellte Adapter hat den Status "Inaktiv".



Einsatz von Adaptern

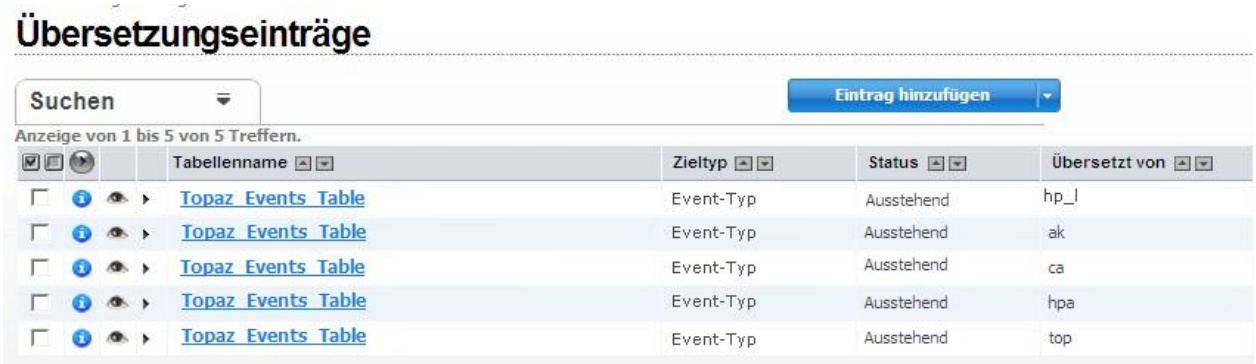
Nach der Konfiguration des Adapters müssen Sie den Adapter auf dem Anwendungsserver bereitstellen. Wenn Sie auf die Schaltfläche "Adapter starten" klicken, stellt der Adapter-Bereitstellungsservice den Adapter für das Dateisystem bereit. Dies umfasst Folgendes:

- Erstellt die Ordner, die für die Ausführung des Adapters erforderlich sind
- Kopiert die XML-Konfiguration in den Ordner
- Erstellt eine Verknüpfung für die Ausführung des Adapters

Sobald Sie dies durchgeführt haben, werden Ihre Änderungen auf dem Server dargestellt.

Jetzt können Sie den Adapter über die Benutzeroberfläche ausführen, indem Sie auf die Schaltfläche "Jetzt ausführen" klicken, die nun aktiv ist. Der Adapter-Bereitstellungsservice führt den Adapter auf dem Server aus und beginnt mit der Datenerfassung.

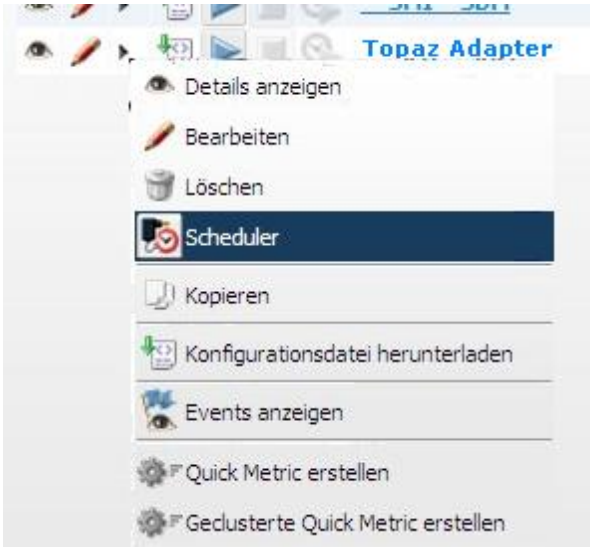
Nach der Ausführung des Adapters sollten nun die ausstehenden Ressourcen und Events im Bereich "Ausstehende Übersetzungen" des Systems angezeigt werden.



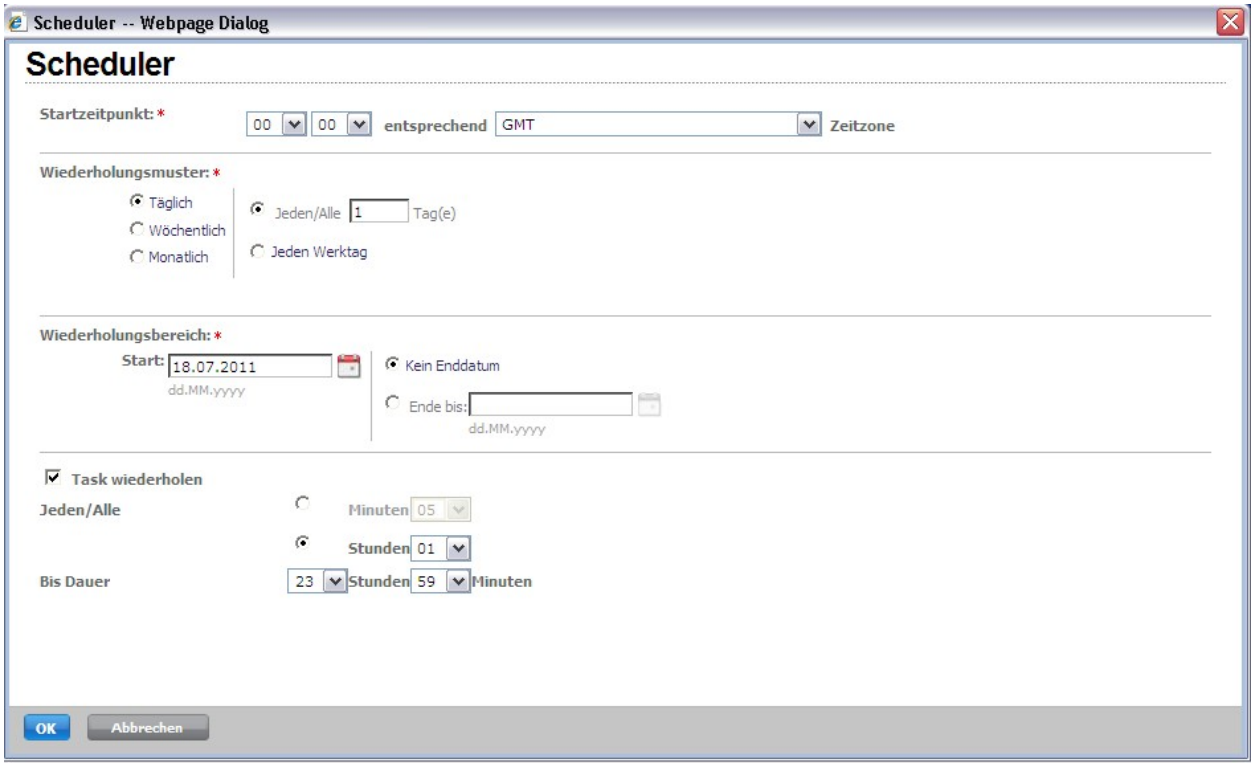
Die ausstehenden Einträge können dann gemäß der normalen Systemkonfiguration übersetzt werden. Nach der Übersetzung sollte der Adapter erneut ausgeführt werden, damit die Rohdaten in das System geladen werden.

Adapterplanung

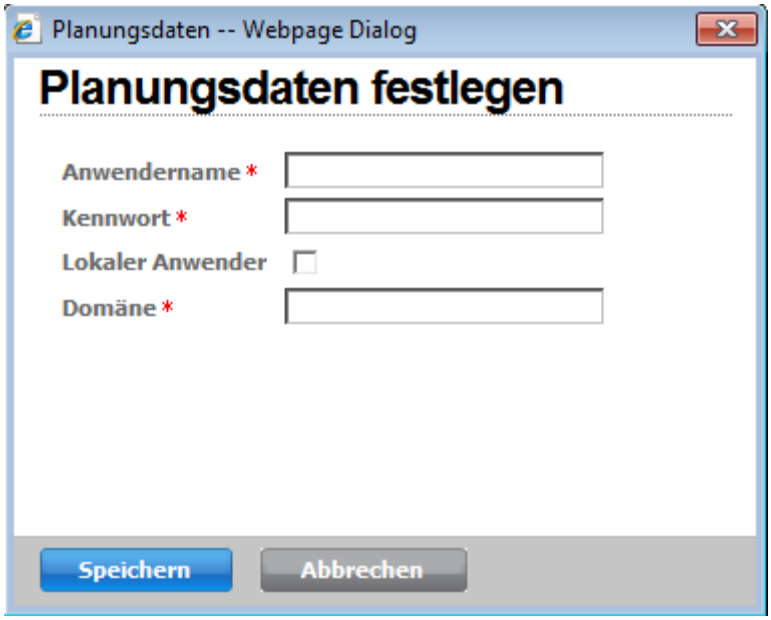
Sie können den Adapter nicht nur ausführen, sondern auch einen Plan für den Adapter in der Benutzeroberfläche festlegen. Der Adapter muss sich dazu jedoch im Status "Angehalten" befinden. Wenn der Adapter angehalten wurde, können Sie den Plan festlegen, indem Sie im Kontextmenü für den Adapter "Scheduler" wählen.



Die Planoptionen entsprechen den Optionen des Windows Task-Scheduler. Der Adapter-Bereitstellungsservice erstellt diesen Plan eigentlich im Hintergrund als Element im Task-Scheduler.



Nachdem ein Plan hinzugefügt wurde und wenn der Adapter das nächste Mal bereitgestellt wird, wird der Anwender aufgefordert, die Anwenderdaten des Kontos auf dem Server einzugeben, der die Task ausführt. Dies sollte im Servicekonto für die Ausführung des CA Business Service Insight-Systems (Standard-OblicoreSrv) oder einem anderen Verwaltungskonto eingegeben werden, falls erforderlich.



Dieser integrierte Scheduler ist sehr nützlich, da der Anwender den Plan des Adapters steuern kann, ohne direkt auf den Desktop des Anwendungsservers zuzugreifen (mit den entsprechenden Anwenderberechtigungen).

Fallstudie 21: LDAP-Integration – Beispiel

Unternehmensanforderung

Verwenden Sie die bereits vorhandenen Anwender, die für den Unternehmens-LDAP-Server definiert wurden. Ferner wird das Unternehmensportal für die CA Business Service Insight-Anmeldung und den Zugriff verwendet. Es werden die CA Business Service Insight-Funktionen für eine "lautlose Anmeldung" für Single Sign-On (SSO)-Portale genutzt.

Definieren Sie ein Visual Basic (VB)-Übersetzungsskript für die automatische Anwendererstellung im CA Business Service Insight-System (LDAP-Synchronisation). Das Übersetzungsskript wird verwendet, um den Unternehmens-LDAP-Server zu verbinden und die Anwenderliste zu extrahieren. CA Business Service Insight-Toolpaket-Methoden werden für die Erstellung von Anwendern, Gruppen und Rollen verwendet.

VB-Code für LDAP-Verbindung – Beispiel

```
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")
'Check user already exists
'Tools.AddUserByMap map
'Check with duplicate
map("UserName") = "acme2"
map("UserPassword") = "acme2"
map("UserPasswordExpirationInterval") = "50"
map("UserDescription") = "New description"
map("UserStatus") = "INACTIVE"
Tools.AddUserByMap map
Tools.Commit
End Sub
```

CA Business Service Insight – VB-Übersetzungsskript-Methoden

- Methoden für Organisationen

  AddOrganization/IsOrganizationExists

- Methoden für Rollen

  IsRoleExists/SearchRoles

- Methoden für Anwender

  AddUserByMap/GetUserName

  GetOrganizationName/IsUserExists

  GetUserDetails/SearchUsers

  GetUserFullName/UpdateUserByMap

- Methoden für Anwendergruppen

  AddUserGroupByMap/IsUserGroupExists

  DeleteUserGroup/SearchUserGroups

  GetUserGroupDetails/UpdateUserGroupByMap

Erstellen Sie Code für die "lautlose Anmeldung", und integrieren Sie ihn in das Unternehmensportal als CA Business Service Insight-Anmeldung.

CA Business Service Insight – Gatway C# Code – Beispiel (wird in das Unternehmensportal integriert)

```
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 WebServce
                        //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 encrupted
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
```

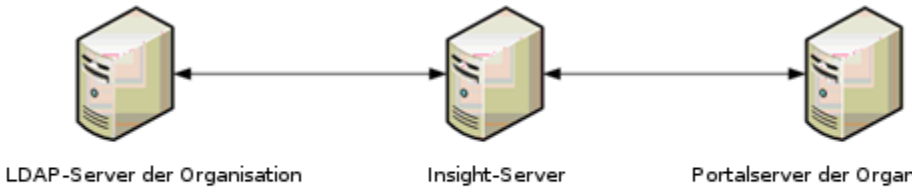
```
                        //After logging in, SilentLogin.asp page will redirect us
back to the current page along with passing OGSESSIONID parameter

//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
        }
}
```

- Flussdiagramm

  Das folgende Diagramm zeigt die Anwendersynchronisation und den Portalzugriffsfluss. Das Übersetzungsskript wird für die periodische Ausführung konfiguriert. Das Skript aktualisiert die LDAP-Anwenderliste und fügt Anwender hinzu/entfernt sie bei Bedarf.

  Die Anwender melden sich beim Unternehmensportal an. Das Portal kann so konfiguriert werden, dass sie zum CA Business Service Insight-Server weitergeleitet werden oder dass eine Liste anderer verfügbarer Anwendungen angezeigt wird. Der CA Business Service Insight-Server verwendet die Daten, die bei der ursprünglichen Portalanmeldung bereitgestellt wurden.



LDAP-Anwender
John
Albert
Sandy
Allison

Insight-Anwender einschl. mit LDAP synchronisierte Anwender
Sadmin
John
Albert
Sandy
Allison

Fallstudie 22: Mit PERL generierte Berichte

Das folgende Beispiel zeigt, wie Sie mit einem PERL-Skript eine Verbindung zum CA Business Service Insight-Bericht Webservice herstellen und die XML-Kriterienparameter mit einem HTTP-Stream in einen SOAP-Envelope übertragen.

Hinweis: Der XML-Code, der in den SOAP-Envelope übertragen wird, darf nicht die Symbole "<" oder ">" enthalten, sondern nur ihren HTML-Code (d. h. <=> >=<)

```
#!/usr/bin/perl

#use strikt;

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 Kontext - 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;
```

Anhang C: Eigenschaften der Adapterkonfiguration

Dieses Kapitel enthält folgende Themen:

Gesamtstruktur (siehe Seite 321)
Abschnitt "General" (siehe Seite 322)
CA Business Service Insight-Schnittstellenabschnitt (siehe Seite 328)
Abschnitt "DataSourceInterface" (siehe Seite 331)
Abschnitt zur SQL-Schnittstelle (siehe Seite 334)
InputFormatCollection-Abschnitt (siehe Seite 338)
Abschnitt "TranslationTableCollection" (siehe Seite 343)
Abschnitt "TranslatorCollection" (siehe Seite 345)

Gesamtstruktur

Die allgemeine Struktur für die Konfigurations-XML-Datei lautet wie folgt:

```
< AdapterConfiguration>
    <General…>
    <OblicoreInterface…>
    <DataSourceInterface…>
    <InputFormatCollection…>
    <TranslatorCollection…>
    <TranslationTableCollection…>
</ AdapterConfiguration>
```

Jeder der Knoten wird in den folgenden Abschnitten beschrieben.

Abschnitt "General"

Der Abschnitt "General" besteht aus Attributen und Unterknoten:

XML-Struktur:

```
<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="04.09.01-12:00:00"
                To="09.01.01-12:00:00"
                Shift="1"/>
</ DataSourceDifferenceFromUTC >
</General>
```

- MajorVersion: Gibt die höchste Versionsnummer an.
- MinorVersion: Gibt die niedrigste Versionsnummer an.
- WorkingDirectoryName: optional.

  Gibt den Standardpfad für die Adapter-Ausgabedateien an (Datenquellensteuerung, Übersetzung, Senden).

Standardwert = Ausgabeverzeichnis (Ordner) im Verzeichnis der Konfigurationsdatei.

- DataSourceControlFileName: optional.

  Name der Datei, die der Adapter verwendet, um die letzten Datensätze nachzuverfolgen, die aus der Datenquelle abgerufen wurden.

  Standardwert = DataSourcecontrol.xml. (Wenn nicht anders angegeben, wird der Standardwert verwendet.)

- SendFileName: optional

  Name der Datei, die die Events enthält, bevor sie an CA Business Service Insight gesendet werden.

  Standardwert = Send.txt. (Wenn nicht anders angegeben, wird der Standardwert verwendet.)

- SendControlFileName: optional

  Name der Datei, die der Adapter für das Nachverfolgen des Sendevorgangs nutzt.

  Standardwert = SendControl.xml (Wenn nicht anders angegeben, wird der Standardwert verwendet.)

- DataSourceIdMaxLength:

  Maximale Länge für das Feld "DataSourceId" in der Tabelle "t_raw_data". Wenn der Anwender einen längeren String einfügt, gibt der Adapter einen Fehler zurück.

  Dieser Wert muss kleiner als die tatsächliche Größe der Tabelle oder gleich sein.

  Standardwert = 60

- SaveIllegalEvents: Dieses Attribut gibt an, dass ungültige Events in die Datei geschrieben werden müssen.

  Standardwert = no

- WriteEventErrorToLog: Legt fest, ob Datenfehler in die Tabelle "T_Log" geschrieben werden

  Erlaubte Werte = [yes/no]

  Standardwert = yes

- IllegalEventsDirectoryName: kein Standardwert

- SendFileName: optional

  Name der Datei, die die Datensätze enthält, bevor sie an CA Business Service Insight gesendet werden.

  Standardwert = Send.txt. (Wenn nicht anders angegeben, wird der Standardwert verwendet.)

- §SendControlFileName: optional

  Name der Datei, die der Adapter für das Nachverfolgen des Sendevorgangs nutzt.

Standardwert = SendControl.xml (Wenn nicht anders angegeben, wird der Standardwert verwendet.)

- LogDebugMode: optional

  Erlaubte Werte = [yes/no]

  "yes": Die ursprüngliche Zeile aus der Datenquelle, die Parsing-Ergebnisse und das einheitliche CA Business Service Insight-Event werden protokolliert.

- ConsoleDebugMode: optional

  Erlaubte Werte = [yes/no]

  "yes": Auf der Konsole werden Debug-Meldungen angezeigt.

  - **Indikatoren für das Lesen und Übersetzen von Datensätzen**:
    - **.**: Für einen Datensatz, der gelesen und in ein Ausgabe-Event übersetzt wurde.
    - **i**: Für einen Datensatz, der vom Parser für reguläre Ausdrücke ignoriert wurde.
    - **I**: Für einen Datensatz, der gelesen und von den Übersetzungstabellen ignoriert wurde.
    - **R**: Für einen Datensatz, der gelesen, aber von der Übersetzungstabelle abgelehnt wurde (kann nicht von den Übersetzungstabellen übersetzt werden).
    - **X**: Für einen Datensatz, bei dem beim Parsen ein Problem aufgetreten ist. Er wird ignoriert und geht verloren oder wird im Verzeichnis für unzulässige Events gespeichert.

  Hinweis: Wenn der Lesedatensatz mehr als einen Übersetzer durchläuft, wird die Angabe des Datensatzes in Klammern gesetzt. Zum Beispiel: [...] oder [RRI].

  - **Statusindikatoren des Adapters**:
    - **0**: Der Adapter ist aktiv und liest in der letzten Sekunde keine Datensätze.
    - **E**: Fehlerstatus.
    - **P**: Pausenstatus.
    - **S**: Stoppbefehl von CA Business Service Insight.
    - **B**: Gesperrter Status, die Tabelle mit abgelehnten Events ist voll.
    - **Übersetzungstabellen-Indikatoren**:
    - **L**: Auf Übersetzungstabellen warten.
    - **T**: Von CA Business Service Insight gesendete/empfangene Übersetzungstabelle.
    - **t**: Von CA Business Service Insight gesendete/empfangene Änderungen in der Übersetzungstabelle.
    - **Verbindungs-Indikatoren**:

- <Connecting CA Business Service Insight: Versuch, eine Verbindung zu CA Business Service Insight herzustellen.
- <Connecting CA Business Service Insight: Verbindung hergestellt.
- <Connecting DataSource: Versuch, eine Verbindung zur Datenquelle herzustellen.
- <Connecting DataSource>: Verbindung hergestellt.

- RunOnce: optional

  Erlaubte Werte = [yes/no]

  "no": Adapter wird kontinuierlich ausgeführt.

  "yes": Der Textdatei-Adapter wird ausgeführt, liest Datensätze und wird automatisch angehalten. Der Dateiadapter liest ganze Dateien, wartet wenige Sekunden, versucht neue Datensätze zu lesen (je nach SleepTime). Wenn keine entsprechenden Datensätze vorhanden sind, wird er angehalten.

  Ein SQL-Adapter führt jede Abfrage nur einmal aus. Wenn "RepeatUntilDry" auf "no" festgelegt wird, wird er sofort angehalten. Wenn "RepeatUntilDry" auf "yes" festgelegt wird, wartet er (abhängig von "SleepTime"), versucht die Abfragen erneut auszuführen (abhängig von der "SleepTime" der Abfrage) und wird angehalten, wenn keine neue Datensätze vorhanden sind.

- RepeatUntilDry: optional

  Erlaubte Werte = [yes/no], Standardwert = yes

  Siehe Attribut "RunOnce" oben.

- RuntimeErrorsLimit: optional

  Bestimmt die Fehlergrenze (beispielsweise für Fehler in den Eingabe-Events), bevor der Adapter gesperrt wird. Bei 0 wird der Adapter nicht gesperrt.

  Standardwert = 1 (Der Adapter wird nach dem ersten Fehler gesperrt.)

- RetryRejectedEvents: optional

  Erlaubte Werte = [yes/no], Standardwert = yes

  "yes": Zu übersetzende Datensätze werden in der Datei mit abgelehnten Events beibehalten, wo sie auf die Übersetzung warten.

  Es wird nicht empfohlen, für dieses Attribut "no" zu wählen, da wichtige Daten verloren gehen können.

- RejectedEventsFileName: optional

  Name der Datei, die der Adapter verwendet, um Datensätze nachzuverfolgen, die aus der Datenquelle abgerufen wurden und auf die Übersetzung warten.

  Standardwert = Value-rejected.txt. (Wenn nicht anders angegeben, wird der Standardwert verwendet.)

- RejectedEventsUpperLimit: optional

Bestimmt den Grenzwert für abgelehnte Datensätze, die in der rejected.txt-Datei gespeichert werden. Hat der Adapter diesen oberen Grenzwert erreicht, wird er angehalten und gesperrt, bis diese Daten übersetzt werden.

Standardwert = 1000

- RegExUnlimitedSearch: Legt fest, dass der Adapter eine vollständige Suche gemäß dem regulären Ausdruck durchführt.

  Erlaubte Werte = [yes/no]

  Standardwert = no

- HoldRejectedEventsSpan: optional

  Legt fest, wie lange die Datei mit abgelehnten Events gespeichert wird (in Stunden). Wenn dieses Attribut angegeben wird, löscht der Adapter alle Events, die verarbeitet werden müssen, bevor der Adapter erneut gestartet werden kann.

  Es wird nicht empfohlen, dieses Attribut zu verwenden, da wichtige Daten verloren gehen können.

- GenerateStatistics: optional

  Erlaubte Werte = [yes/no], Standardwert = yes

  "yes": Der Adapter erstellt minütlich eine Statistikdatei mit Statistikinformationen.

- StatisticsFileName: optional

  Name der Statistikdatei

  Standardwert = Value-statistics.txt. (Wenn nicht anders angegeben, wird der Standardwert verwendet.)

- KeepHistoricState: optional

  Erlaubte Werte = [yes/no], Standardwert = no

  "yes": Der Adapter speichert alle Dateien, außer der Protokolldatei, in einem neuen Verzeichnis mit dem Namen "Historic state yyyymmdd-hhmmss", wobei "yyyymmdd" und "hhmmss" Datum und Uhrzeit der Erstellung angeben.

- DefaultTimeFormat: optional

  Standardzeitformat Wenn dieses Attribut angegeben wird, wird es als Zeitformat verwendet, wenn das TimeFormat-Attribut nicht angegeben wird. Wenn es nicht angegeben wird, ist das TimeFormat-Attribut in den anderen Elementen obligatorisch.

- DefaultDecimalSymbol: optional

  Das Standard-Dezimaltrennzeichen für reale Felder.

  Standardwert = (Wenn nicht anders angegeben, wird der Standardwert verwendet.)

- DefaultDigitGroupingSymbol: optional

  Standardsymbol für Zifferngruppierung für Felder mit Ganzzahl- und Realwerten.

  Standardwert = (Wenn nicht anders angegeben, wird der Standardwert verwendet.)

- DataSourceDifferenceFromUTC: Gibt den Zeitunterschied zwischen UTC und der Zeitzone der Datenquellenmaschine an.
  - DefaultOffset: Abweichung zu UTC, wenn nicht in Sommerzeit.
  - TimeFormat: Gibt das Format an, in dem die Sommerzeitdetails (als Nächstes beschrieben) geparst werden sollen.
  - DayLightSaving: Gibt einen Sommerzeit-Zeitraum der Datenquellenzeitzone an. Dieses Element ist optional (das heißt, wenn diese Option nicht gewählt wird, wird keine Sommerzeit angegeben) und kann mehr als einmal vorhanden sein. Wenn mehrere Elemente vorhanden sind, müssen sie nach Zeit geordnet werden. Die Zeiträume dürfen sich nicht überschneiden.
    - Von: Anfangsdatum des Zeitraums.
    - Bis: Enddatum des Zeitraums.
    - Unterschied: Zeitverschiebung, die zu "DefaultOffset" innerhalb des Sommerzeit-Zeitraums hinzugefügt wird.

CA Business Service Insight-Schnittstellenabschnitt

Der CA Business Service Insight-Schnittstellenabschnitt besteht aus Attributen, die die Verbindung mit CA Business Service Insight angeben. Es gibt zwei Modi: online und offline. Im Onlinemodus stellt der Adapter eine Verbindung mit CA Business Service Insight her, ruft die Übersetzungstabellen und den Steuerbefehl von CA Business Service Insight ab und sendet Events, Statusmeldungen und Übersetzungsanfragen an CA Business Service Insight. Im Offlinemodus arbeitet der Adapter mit einer lokalen Übersetzungstabellendatei und schreibt die Events in eine Ausgabedatei.

XML-Struktur für Onlinemodus:

```
<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: TCP/IP-Portnummer, die der Adapter verwendet, um mit dem CA Business Service Insight-Server zu kommunizieren.
- ConnectionInitiator: optional

  Erlaubte Werte = server/adapter Standardwert = server

  Definiert den Verbindungsinitiator, die Adapterseite oder den Adapter-Listener auf CA Business Service Insight-Seite.

  Der Initiator fungiert als "Slave" oder Client, die andere Seite als "Master" oder "Server".
- Adresse: Netzwerkadresse des Servers; obligatorisch, wenn der Initiator der Adapter ist.
- SecurityLevel: Definiert die Authentifizierungsstufe zwischen dem Adapter und dem CA Business Service Insight-Server ("Handshake"). Optionen:
  - none: Keine Authentifizierung.
  - high: Authentifizierung. SecurityKey muss bereitgestellt werden.
- SecurityKey: String aus 32 Zeichen. Der String muss dem für den Adapter in der CA Business Service Insight-Datenbank definierten String entsprechen.

  Der "Handshake"-Prozess umfasst Folgendes:

- Der CA Business Service Insight-Server stellt eine Verbindung mit dem Adapter her.
- Ein Zufalls-String wird vom Adapter an den CA Business Service Insight-Server gesendet.
- Der CA Business Service Insight-Server verschlüsselt den String mit einem vorkonfigurierten Schlüssel und sendet das Ergebnis zurück an den Adapter.
- Der Adapter verschlüsselt denselben Zufalls-String, der an den CA Business Service Insight-Server gesendet wurde, mit dem SecurityKey-String und vergleicht die Ergebnisse.
- Der CA Business Service Insight-Server wählt willkürlich einen String aus und sendet den String an den Adapter.
- Der Adapter verschlüsselt den String und sendet ihn zurück an den CA Business Service Insight-Server.
- Der CA Business Service Insight-Server verschlüsselt denselben String und vergleicht die Ergebnisse mit den Ergebnissen des Adapters.
- Die Verbindung wird hergestellt.
- Wenn ein Fehler auftritt, wird keine Verbindung hergestellt.

- UseAcknowledgeProtocol: optional

  Erlaubte Werte = [yes/no], Standardwert = yes

  "yes": Der Adapter verwendet das Quittierungsprotokoll. "no": Der Adapter sendet die Meldungen/Pakete und wartet nicht auf die Bestätigung durch CA Business Service Insight.

  Es wird nicht empfohlen, für dieses Attribut "no" zu wählen, da Events verloren gehen können.

- PacketSize: optional

  Erlaubte Werte = 1 bis 1 000

  Standardwert = 50

  Maximale Anzahl von Events in einem Paket

- PacketDeadline: optional

  Erlaubte Werte = 1 bis 3 600

  Standardwert = 60

  Anzahl der Sekunden, bevor nicht volle Pakete gesendet werden.

- PacketResendTimeout: optional

  Erlaubte Werte = 1 bis 3 600

  Standardwert = 60

Zeit in Sekunden, die auf eine Bestätigungsmeldung gewartet werden soll, bevor das Paket erneut gesendet wird. Dieses Attribut ist nur anwendbar, wenn Sie für das Attribut "UseAcknowledgeProtocol" die Option "yes" wählen.

- WindowSize: optional

  Erlaubte Werte = 1 bis 100

  Standardwert = 10

  Anzahl von Paketen im Fenster Dieses Attribut ist nur anwendbar, wenn Sie für das Attribut "UseAcknowledgeProtocol" die Option "yes" wählen.

XML-Struktur für Offlinemodus:

```
<OblicoreInterface
        Mode="offline">
        <OfflineInterface
                OutputFileName="outputEvents.txt"
                OutputFileType="tsv"
                OutputTimeFormat="%Y/%m/%d %H:%M:%S"
                OutputDecimalSymbol="."
        />
</OblicoreInterface>
```

- OutputFileName: optional

  Name der Datei, in der der Adapter die Ausgabe-Events speichert

  Standardwert = AdapterOutput.txt. (Wenn nicht anders angegeben, wird der Standardwert verwendet.)

- OutputFileType: optional

  Erlaubte Werte =csv/tsv/xml

  Definiert das Event-Format

- OutputTimeFormat: Definiert das Format der Datumsfelder Dieses Attribut kann ignoriert werden, wenn das Attribut "DefaultTimeFormat" im Abschnitt "General" angegeben wurde.

- OutputDecimalSymbol: optional

  Definiert das Dezimaltrennzeichen für Lesefelder

  Standardwert = . (Punkt)

Abschnitt "DataSourceInterface"

Der Abschnitt "DataSourceInterface" besteht aus Attributen, die die Verbindung und den Verbindungstyp zwischen dem Adapter und der Datenquelle angeben (Messtool, CRM, Systemprotokoll usw.), und ist in zwei Haupttypen unterteilt: Dateischnittstelle und SQL-Schnittstelle.

Dateischnittstelle

Mit dem Dateiadapter können Daten aus Protokolldateien, geplanten Berichten oder anderen textbasierten Dateien abgerufen werden. "DataSourceInterface" definiert Regeln für das Parsen der Informationen aus der Dateidatenquelle und das Extrahieren in Felder.

Der Abschnitt "DataSourceInterface" gibt auch an, wie der Adapter die Quelldatei verwaltet (ob er die ursprüngliche Datei löscht, wenn sie nur für den Adapter erstellt wurde, oder, ob er die Daten beibehält, wenn sie für andere Verwendungen benötigt werden, und so weiter).

XML-Struktur:

```
<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: optional. "DeleteFileAfterProcessing" = "no": Es wird eine Kopie der Datei mit diesem Namen erstellt. "yes": Die Datei wird in diesen Namen umbenannt. Wenn nichts angegeben wird, wird der Standardwert übernommen ('WorkFile.txt').
  - Files: Sammlung von Dateielementen. (Es können mehrere Dateien in einem Adapter angegeben werden.)
  - File: Gibt die Dateiattribute an.
    - IsActive: optional [yes/no] Definiert, ob diese Datei aktiv ist. ("no": Diese Datei wird nicht gelesen.)

- InputFormat: Das mit dieser Datei verknüpfte Eingabeformat. Der Adapter verwendet "InputFormat", um die Daten aus der Datenquelle zu extrahieren.
- Path: Pfad zum Speicherort der Quelldatendateien.
- NamePattern: Gibt den Datenquelldateinamen an. Verwendung von Platzhalter möglich, wenn mehrere Dateien dasselbe Eingabeformat nutzen.
- DeleteFileAfterProcessing [yes|no] – Art und Weise, wie der Adapter die Quelldatei verarbeitet. Wenn die Datei nur für den Adapter erstellt wurde und nach der Bearbeitung gelöscht werden kann, wählen Sie "yes". Die Datei wird umbenannt, verarbeitet und dann gelöscht. Wenn Sie "no" wählen, wird die Datei kopiert und die Verarbeitung findet in der kopierten Datei statt. Wenn neue Datensätze am Ende dieser Datei eingefügt werden, kopiert der Adapter diese neuen Datensätze während des nächsten Zyklus in die Arbeitsdatei. Wenn keine neuen Datensätze in die Datei eingefügt werden, sucht der Adapter nach der ersten Datei mit dem gleichen Muster und einem anderen Namen (in lexikografischer Reihenfolge) wie die aktuelle Datei. Wenn der Adapter eine derartige Datei identifiziert, arbeitet er mit dieser Datei weiter. Der Adapter kehrt nicht zur vorherigen Datei zurück, auch wenn neue Datensätze in diese Datei eingefügt werden. Wählen Sie "no", wenn Sie die Integrität der Quelldatei beibehalten müssen.
- InitialFileName: Name der ersten Datei, von der an der Adapter eine Datei mit dem angegebenen Muster sucht. Verwenden Sie dieses Attribut, wenn "NamePattern" Platzhalter enthält und wenn Sie nicht möchten, dass der Adapter alte Dateien liest.

- Delimiters: optional. Ein oder mehrere Zeichen dienen als Trennzeichen, entsprechend denen Datenzeilen in Felder geparst werden (falls der Standardwert nicht "\t" ist).
- IgnoreRedundantDelimiters: optional [yes/no]. Wenn Sie "yes" wählen, werden mehrere aufeinander folgende Trennzeichen als ein einziges behandelt. Andernfalls wird ein leeres Feld zwischen den Trennzeichen eingefügt.
- RegExForParser: (optional). Regulärer Ausdruck zum Extrahieren der Felder aus der Datenquelle als Ersatz für die obigen Trennzeichen. Zum Beispiel:

```
<File
    ….
    RegExForParser="^(.{8}) (.{6}) (?:[ ])*([0-9]+) "
/>
```

  In der Dokumentation zu regulären Ausdrücken finden Sie weitere Informationen.

- TitleExists: optional [yes/no]. Gibt an, ob die erste nicht leere Zeile der Datenquelldatei eine Titelzeile ist. Der Titel kann vom Adapter beim Parsen der Daten verwendet werden.

- SleepTime: Gibt das Datenabrufintervall (in Sekunden) an, d. h. die Anzahl von Sekunden zwischen dem Abrufen von Daten von der Quelldatendatei durch den Adapter.
- LogicLineDefinition: optional

```
<File
    ….
    <LogicLineDefinition
                    FirstLine="Job server:.*"
                    NumberOfLines="5"
    />
/>
```

Wenn der Datensatz aus verschiedenen Zeilen besteht, definieren folgende Attribute den Anfangs- und Endpunkt der Extraktion und die Anzahl der Zeilen mit Daten:

- AllFile: optional [yes/no]. Wenn Sie "yes" wählen, gilt die gesamte Datei als ein Datensatz und eine logische Zeile.
- FirstLine: optional. Ein regulärer Ausdruck, der die erste Zeile der logischen Zeile definiert. Diese Definition kann mit oder ohne "LastLine" und/oder "NumberOfLines" vorgegeben werden.
- LastLine: optional. Ein regulärer Ausdruck, der die letzte Zeile der logischen Zeile definiert. Diese Definition kann mit oder ohne "FirstLine" und/oder "NumberOfLines" vorgegeben werden.
- NumberOfLines: optional. Zeilenanzahl in einer logischen Zeile. Diese Definition kann mit oder ohne "FirstLine" und/oder "LastLine" vorgegeben werden.
- MatchCase: optional [yes/no]. Definiert, ob bei den regulären Ausdrücken Groß-/Kleinschreibung zu beachten ist.

Abschnitt zur SQL-Schnittstelle

Mit SQL-Adaptern können Sie Daten mittels SQL-Anweisung von einer Datenbank abrufen.

Die SQL-Schnittstelle legt die Verbindung zur Datenbank und die Abfragen fest, die zum Abrufen der Daten verwendet werden:

XML-Struktur:

```
        < DataSourceInterface >
                <ConnectionString ConnectionTimeout="60" QueryTimeout="30">
                        <![CDATA[ Driver={Microsoft Access Driver
(*.mDataBase)};DataBaseq=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;
DataBaseq="/>
                <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: Verbindungsstring, mit dem auf die Quelldatenbank zugegriffen werden kann.

"ConnectionString" kann im Element "DataSourceInterface" und/oder in den Query-Elementen definiert werden. Die "ConnectionString"-Definition wird standardmäßig im Element "DataSourceInterface" festgelegt und nur bei einer Abfrage ohne eine "ConnectionString"-Definition verwendet.

Er kann in einem String oder segmentweise angelegt werden. Wenn das Element "ConnectionString" keine Segmentelemente enthält, wird der Verbindungsstring aus dem "ConnectionString"-Elementtext entnommen. Enthält es wenigstens ein Segmentelement, wird der Verbindungsstring daraus genommen.

Es gibt zwei Segmenttypen. Der eine Typ ist Text und enthält Text, der als solches mit dem Verbindungsstring verknüpft ist. Der zweite Typ ist eine Datei und enthält einen Dateinamen mit oder ohne Platzhalter. Das Dateisegment kann nur einmal verwendet werden. Es enthält andere Attribute, die die Vorgehensweise bezüglich der Lesedatei definieren.

- ConnectionTimeout: optional. Positiver ganzzahliger Wert, der angibt, wie lange in Sekunden bis zum Öffnen der Verbindung gewartet werden soll. Der Wert 0 bedeutet, unendlich lange zu warten, bis die Verbindung geöffnet wird. Der Standardwert ist 15 Sekunden.

Hinweis: Einige Anbieter unterstützen diese Funktionalität nicht.

- QueryTimeout: optional. Positiver ganzzahliger Wert, der angibt, wie lange in Sekunden bis zum Ausführen der Abfrage gewartet werden soll. Der Wert 0 bedeutet, dass bis zum Abschluss der Ausführung unendlich lange gewartet werden soll. Der Standardwert ist 30 Sekunden.

Hinweis: Einige Anbieter unterstützen diese Funktionalität nicht.

- Segment: Gibt die Segmentattribute an.
- Type: optional (text, file), Segmenttyp
- Text: Text des Textsegments.
- File: Gibt die Dateiattribute an.
- Path: Pfad zum Speicherort der Quelldatendateien.
- NamePattern: Gibt den Datenquelldateinamen an, Verwendung von Platzhaltern möglich.
- InitialFileName: optional. Gibt dem Anbieter an, bei welcher Datei die Suche gestartet werden soll und nach welchem Muster gesucht werden soll.
- UsePath: optional [yes, no]. Wenn Sie "yes" wählen, wird der Dateipfad an den Verbindungsstring angehängt.
- UseFileName: optional [yes/no]. Wenn Sie "yes" wählen, wird der Dateiname an den Verbindungsstring angehängt (erforderlich für Excel-Dateien).
- UpdateSchemaFile: optional [yes/no]. Wird der Wert auf "yes" gesetzt, aktualisiert der Adapter die Datei "schema.ini" mit dem aktuellen Dateinamen.

Hinweis: Verwenden Sie dieses Attribut nur, wenn der Adapter die Datei "schema.ini" ändern soll (Datenbank für Textdateien).

- QueryCollection: Abfragesammlung (In einem Adapter kann mehr als eine Abfrage angegeben werden.)
  - PreliminaryCheck-: (optional [ja/nein]). Der Adapter überprüft die Abfragen, wenn er eine Verbindung zur Datenbank herstellt. Wenn für dieses Attribut "no" gewählt wird, prüft der Adapter die Abfragen, indem er sie ohne Änderung ausführt. Der Adapter wird später für die Rückgabe-Datensätze ausgeführt, die er nicht mehr liest. Wenn "yes" gewählt wird, ersetzt der Adapter jeden "where"-String in der Select-Anweisung durch "where 1=2 and". Erst dann werden die Abfragen ausgeführt. Verwenden Sie diese Option, um alle Abfragen zu prüfen, bevor der Adapter die Datensätze durchläuft und wenn dieser Zusatz die Abfragezeit erheblich verkürzt. Hinweis: Einige Anbieter optimieren den Abfrageprozess nicht. Bei einer zulässigen Abfrage erfordert die Ausführung genau so viel Zeit wie ohne den Zusatz.
  - ExternalCommand: optional. Externer Befehl, der ausgeführt wird, bevor der Adapter einen neuen Lesezyklus startet.
  - Command: Name der Batch-Datei, die ausgeführt werden soll.
  - SleepTime: Positiver ganzzahliger Wert, der angibt, wie lange in Sekunden bis zur Ausführung der Abfrage gewartet werden soll.
- Query: Gibt die Abfrageattribute an.
  - QueryName: Name für die Abfrage.
  - IsActive: optional [yes/no] "no": Der Adapter liest keine Datensätze aus dieser Datei.
  - InputFormat: Das mit dieser Abfrage verknüpfte Eingabeformat. Der Adapter verwendet "InputFormat", um die Daten aus dem Quelldatensatz zu extrahieren.
  - SleepTime: Zeit in Sekunden, in der der Adapter herunterfährt, um auf neue Daten zu warten.
  - Critical: optional [yes/no]. "yes": Der Adapter wird angehalten, wenn er einen Fehler in der Abfrage identifiziert. "no": Wählen Sie diese Einstellung, wenn der Fehler bekannt ist und ohne Ändern der Konfigurationsdatei gelöst wird.
  - TransactionMode: optional [implicit/explicit]. "explicit": Der Adapter initiiert vor der Abfrage eine Datenbanktransaktion. Diese Option löst verschiedene Probleme im Zusammenhang mit umfangreichen und komplizierten Abfragen.
  - RollbackSegementName: optional. Definiert, welches Rollback-Segment der Adapter verwendet. Andernfalls wählt die Datenbank das Rollback-Segment.
- SelectStatement: Enthält eine ausgewählte Anweisung, die auf der Quelldatenbank ausgeführt werden muss.
  - AutoCompleteQuery: optional [yes/no]. Wenn diese Option auf "yes" gesetzt wurde, verknüpft der Adapter automatisch wie folgt eine Where-Anweisung mit der gewählten Abfrage:

- Erstellt eine Where-Anweisung, die nur neue Werte basierend auf den Schlüsselfeldern abruft.
- Fordert die auf den Schlüsselfeldern basierte Ergebnisanweisung an.

- Abfrageschlüsselfelder - Definiert die Felder, um den weitern Datenabruf zu starten:
  - KeyField:
  - Name - Name des Feldes, von den Abfragefeldern übernommenen.
  - Sort - (optional) [auf/ab]. Daten-Sortierverfahren (aufsteigend/absteigend).
  - Funktion - (optional). Verwenden Sie dieses Attribut, wenn eine Sonderfunktion auf dieses Feld angewendet werden soll, zum Kombinieren vom Feldwert in der Funktion mit (:fieldname). Zum Beispiel die Verwendung der Oracle-Datumsfunktion mit einem Feldnamen: Ts-Function="to_date(':ts','dd/mm/yyy')"
  - ValueLeftWrapper - (optional). Verwenden Sie dieses Attribut, um Zeichen vor dem Wert des Feldes zu verketten. Der Standard für die Zeichenfolge und Datumsfelder ist"'" (Apostroph).
  - ValueRightWrapper - (optional). Verwenden Sie dieses Attribut, um nach dem Wert des Feldes Zeichen zu verketten. Der Standard für die Zeichenfolge und Datumsfelder ist"'" (Apostroph).
  - NameLeftWrapper - (optional). Verwenden Sie dieses Attribut, um Zeichen vor dem Namen des Feldes zu verketten. Der Standard ist Null-String.
  - NameRightWrapper - (optional). Verwenden Sie dieses Attribut, um nach dem Namen des Feldes Zeichen zu verketten. Der Standard ist Null-String.
  - IsPrimaryKey - (optional) [yes/no]. Wenn auf "no" festgelegt, wird dieser Schlüssel vom Adapter nicht in der automatischen WHERE-Anweisung in der Abfrage verwendet.
  - DefaultInitialValue - (optional). Wenn die SelectInitialValues-Abfrage Datensätze nicht zurückgibt, übernimmt der Adapter den Standard von hier. Wenn es einige Schlüsselfelder gibt, müssen alle Schlüsselfelder dieses Attribut haben.
  - SelectInitialValues - Eine Select-Anweisung, die die Initialwerte für die erste Where-Anweisung liefert (bei leerer Steuerungsdatei).

  Hinweis: Die Reihenfolge der Werte muss dieselbe wie diejenige der Feld-Elemente für dieses QueryKeyFields-Element sein und für jedes Feld ein Ergebnis zurückgeben.

InputFormatCollection-Abschnitt

Dieser Abschnitt beschreibt die Struktur der aus der Datenquelle abgerufenen Daten (wie eine Datenzeile in Felder aufgeteilt werden soll, die Feldtypen und die Formate). In diesem Abschnitt erfolgen in den kombinierten Feldern die Eingangsdatenfilterung und Datenbearbeitung.

Der allgemeine Workflow dieses Abschnitts ist folgendermaßen:

- Die Datenzeile stimmt mit einem oder mehreren InputFormat-Elementen überein.
- Die Daten werden nach der Übereinstimmung der Spezifikation "InputFormat" in Felder zerlegt.
- Kombifeldern werden Werte zugewiesen, indem Datenfelder verbunden und geteilt werden.
- Verarbeitete Daten werden gegen TranslatorSwitch-Bedingungen überprüft.
- Verarbeitete Daten werden zum passenden Übersetzer gesandt oder ignoriert.

Der Knoten "InputFormatCollection" kann einen oder mehrere InputFormat-Knoten enthalten.

XML-Struktur:

```
<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="cnano"/>
        </TranslatorCase>
</TranslatorSwitch>
</InputFormat>
</InputFormatCollection>
```

- InputFormat:

- InputFormatName - Ein beliebiger Name für dieses Format, zum Verweis durch den DataSourceInterface-Abschnitt.

- RequiredFields - (optional) Minimale Anzahl von Feldern, die voraussichtlich in einer Datenzeile gefunden werden. Eine Reihe, die weniger Felder enthält, wird ignoriert, und ein Fehler wird protokolliert.
- InputFormatFields - InputFormatFields kann entsprechend der Anzahl der Eingangsdatenfelder der Datenquelle einen oder mehrere Feldknoten enthalten.
  - InputFormatField - Gibt ein Datenfeld der ursprünglichen Datenzeile oder ein Kombifeld an.
    ```
    <InputFormatField Name="timestamp" Type="time"
    TimeFormat="%d/%m/%Y %H:%M:%S"/>
    ```
  - Name - Ein Name für dieses Feld, auf das andere Elemente Bezug nehmen können.
  - Type - Datentyp vom Feld aus string/integer/real/time
  - Source - (optional) Der übernommener Standardwert ist "event", mögliche Werte:
  - event - Der Feldwert wird vom Event der Datenquelle übernommen. Die Werte werden in derselben Reihenfolge entnommen, in der sie von der Datenquelle eingehen.
  - compound - Feld ist eine Zusammensetzung. Es erhält seine Werte nach der Bearbeitung anderer Feldwerte oder Konstanten.
  - Title - Der Feldwert wird vom Titelfeldnamen übernommen. Das betreffende Feld sollte schon definiert sein
  - filename - Der Feldwert wird dem Dateinamen der Datenquelle nur für Textdateiadapter entnommen.
  - constant - Der Feldwert ist konstant und der ConstantValue-Eigenschaft entnommen, die als Nächstes angezeigt werden sollte.
  - FieldTitleName - Wenn die Quelle ein Titel ist, wird der zu verwendende Feldtitel angegeben. Das Quellenfeld muss vorher definiert werden.
  - ConstantValue - Wenn die Quelle konstant ist, wird der Wert angegeben, der angepasst werden muss. Wenn der Typ des Feldes "Time" ist, wird der konstante Wert formatiert, entsprechend dem "TimeFormat" oder "Now" oder "NowUtc", wo die aktuelle Zeit "Now" im Datenquellen-Gebietsschema ist und "NowUtc", wo die aktuelle Zeit in UTC ist.
  - TimeFormat: Format, in dem das Feld analysiert ist. Die folgenden Zeichencodes können für das Datums- und Zeitformat verwendet werden:
  - DecimalSymbol - (optional) Das dezimale Symbol für Felder. Wenn nicht angegeben, wird das DefaultDecimalSymbol vom allgemeinen Abschnitt übernommen.

- DigitGroupingSymbol - (optional) Dieses Symbol wird als Standard-Stellengruppierungssymbol für Felder mit Ganzzahl- und Realwerten verwendet. Wenn nicht angegeben, wird das DefaultDigitGroupingSymbol, vom allgemeinen Abschnitt übernommen.

- Compound - Erforderlich wenn source=compound. Gibt die in einem Kombifeld zu erfassenden Feldbearbeitungen an.
  - Segment - Gibt eine der erstellten Zusammensetzung hinzuzufügende Feldbearbeitung an. Nur das SourceField-Attribut wird benötigt.
  - SourceField - Feld, das als Basis dienen soll. Das angegebene Feld sollte bereits definiert sein.
  - RegularExpression - Ausdrucksregeln für die Bearbeitung.
  - MatchCase - (optional) [yes/no] Definiert, ob bei den Ausdrucksregeln Groß-/Kleinschreibung zu beachten ist.
  - SelectionStart - Startposition zum Extrahieren von Text, beginnend mit null.
  - SelectionLength - Größe des zu extrahierenden Texts.
  - Prefix - String als Präfix für das Bearbeitungsergebnis.
  - Suffix - String als Suffix für das Bearbeitungsergebnis.
  - XpathExpression - Der xpath-Ausdruck für die Bearbeitung.
- InputFormatSwitch - Wird verwendet, um Formatkriterien anzugeben, wenn Datenzeilen in mehr als einem Format vorliegen.

  Hinweis: Bei der Verwendung von InputFormatSwitch ist die Reihenfolge von InputFormat-Knoten wichtig - ein angegebenes Feld sollte bereits vorher definiert worden sein.

  DefaultInputFormat - Gibt den Namen des InputFormat-Elements an, an das weitergeleitet werden soll.
- InputFormatCase - Gibt ein Kriterium an, das für Datenzeilen getestet werden muss, um zu entscheiden, an welches InputFormat-Element es weitergeleitet werden soll.
- InputFormatName - Das InputFormat, wenn die Kriterien übereinstimmen.
- LogicOperator = (optional) [und/oder].
- und - Alle Bedingungen müssen angepasst werden. (der Standard).
- oder- Mindestens eine Bedingungen muss angepasst werden.
  - Condition - Bedingung, die für eine Datenzeile geprüft wird, um deren Format zu bestimmen.

    SourceField - Feld, das geprüft werden muss.

    Operator - Prüfart der folgenden Optionen:
    - **EQ** – Gleich
    - **NE** – Nicht gleich

- **GT** – Größer als
- **LT** – Kleiner als
- **GE** – Größer oder gleich
- **LE** – Kleiner oder gleich
- **MATCH** – Ausdrucksregel muss angepasst werden.
- **UNMATCH** – Ausdrucksregel muss nicht angepasst werden.

ValueType - (optional) [constant/field/previousValue]

- constant - Der Inhalt des Wert-Attributs ist konstant, unabhängig von den Quellendaten.
- field - Der Inhalt des Wert-Attributs ist der Name des Feldes aus dem gleichen Datensatz.
- previousValue - Der Inhalt des Wert-Attributs ist der Name des Feldes aus dem vorherigen Datensatz in der gleichen Abfrage und mit dem gleichen Eingabeformat.

Value - Wert, der angepasst werden muss, oder eine Ausdrucksregel.

MatchCase - (optional) [yes/no] Definiert, ob beim Prüfen Groß-/Kleinschreibung zu beachten ist. Wenn die Option auf "yes" gesetzt ist, werden die zwei Werte vor dem Test in Kleinbuchstaben übersetzt.

- TranslatorSwitch - Legt fest, welcher Übersetzer verwendet werden soll, um die Datenzeile in ein einheitliches CA Business Service Insight-Event zu übersetzen.
  - DefaultTranslator - Übersetzer, der zu verwenden ist, falls keine Kriterien übereinstimmen. Wenn der Wert "Ignorieren" ist, wird kein Übersetzer verwendet, und die Zeile wird ignoriert.
  - TranslatorCase - Gibt Kriterien für verarbeitete Daten an, um festzulegen, zu welchem Übersetzer sie weitergeleitet werden sollen.

    Break [yes|no]

    "yes" - (Standard) Wenn mit den Kriterien übereinstimmend, müssen keine weiteren Kriterien geprüft werden.

    "no" - Fahren Sie nach der Bewertung der Kriterien und der Ausführung des Übersetzers (bei Übereinstimmung) in jedem Fall mit den nächsten Kriterien fort.

    LogicOperator = (optional) [und/oder].

    - und - Alle Bedingungen müssen angepasst werden. (der Standard).
    - oder- Mindestens eine Bedingungen muss angepasst werden.

    TranslatorName - Der Übersetzer, zu dem geleitet wird, wenn die Bedingungen erfüllt werden.

Condition - Die Bedingung, die für verarbeitete Daten geprüft werden muss, um den zu verwendenden Übersetzer zu bestimmen. Es ist die gleiche wie die Bedingung im InputFormatSwitch.

Abschnitt "TranslationTableCollection"

Der Abschnitt enthält Zuordnungstabellen von den Datenquellwerten der Felder des Events aus CA Business Service Insight.

XML-Struktur:

```
<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: (optional) Gibt den Dateinamen an, in dem die Tabellen lokal gespeichert werden; wenn nicht angegeben, werden die Standardwerte übernommen ("Translation.XML").
- LoadingMode: (optional) [Eigenständig, Remote].

  Hinweis: Onlineschnittstellen sind standardmäßig remote und Offlineschnittstellen eigenständig. Die angegebene Lademethode der Übersetzungstabellen ist wie folgt:
- Eigenständig: Der Adapter lädt die Übersetzungstabellen lokal. Keine Verbindung mit dem CA Business Service Insight-Server im Bezug auf die Übersetzung. Änderungen in den Übersetzungstabellen werden nur in der lokalen Datei gespeichert.
- Remote: Der Adapter sendet eine Anfrage zum Laden aller Tabellen vom CA Business Service Insight-Server. Änderungen durch die Übersetzungstabellen werden auch lokal gespeichert.
- LoadTimeout: (optional) wenn der Lademodus remote ist, können Sie hier eine Zeitüberschreitung in Sekunden eingeben.
  - TranslationTable: Verlinkt den Event-Wert zur Zuordnungstabelle.

- Name: Ein Name, der vom Übersetzer verwendet und auf den verwiesen wird. Ein regulärer Name fängt mit Buchstaben oder Unterstrich an und enthält Buchstaben, Ziffern und Unterstriche.
- DestinationType: [resource, event_type, contract_party, service, time_zone, value].
- ValueType: (optional) [integer, real, string] Der Typ des zurückgegebenen Wertes der Tabelle. Der Stringwert und der Realwert sind in der Regel nur für Tabellen mit DestinationType="value"
- TranslationField: Der Feldname aus dem zu übersetzen ist, der Name des Feldes wird von den Eingabeformatfeldern übernommen. Sie können bis zu 5 Felder haben.

Abschnitt "TranslatorCollection"

Der Abschnitt "TranslatorCollection" beschreibt, wie die im vorherigen Abschnitt zu einem CA Business Service Insight-Event extrahierten, geparsten und bearbeiteten Datenquellendatensätze übersetzt werden sollen.

Wenn der Schnittstellenmodus "online" ist, hat das CA Business Service Insight-Event eine einheitliche Struktur, welche die folgenden Felder enthält:

- Timestamp: Die Zeit des Eintretens des Events.
- ResourceId: Die mit dem Event verbundene Ressourcen-ID von CA Business Service Insight (die gemessene Ressource etc.).
- EventTypeId: Der mit dem Event verbundene Event-Typ von CA Business Service Insight, der die Art des Events beschreibt (Art der Messung auf der Ressource, Art der Ticketaktion etc.).
- DataSourceId: (optional) Der Name der Eingabedatenquelle (Dateiname/Tabellenname ...).
- Value: Der Wert des Events (Messergebnis, Ticketnummer etc.). Diese Feld kann mehrmals vorhanden sein.

Wenn der Schnittstellenmodus "offline" ist, gibt es keine Beschränkungen auf die Anzahl der Feldern oder auf deren Namen.

XML-Struktur:

```
<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>
```

- Translator: Beschreibt, wie der Satz von Feldern, die er erhält, in das Ausgabe-Event übersetzt wird.
  - TranslatorName: Name, der von "InputFormat" verwendet wird, um diesem Übersetzer Feldeinstellungen zu senden.
  - OnDuplication: Mitglied, das "ignore"-, "add"-, "update"- oder "updateAlways"-Werte enthält, um zu entscheiden, was mit dem doppelten Event zu tun ist (siehe Event-Besonderheit)
  - TranslatorFields: Enthält eine Liste von TranslatorField-Elementen, jeweils mit den folgenden Attributen:
  - Name: Feldname. In der Onlineschnittstelle muss es "Timestamp", "ResourceId", "EventTypeId", "Value" oder "DataSourceId" sein.
  - SourceType:

    Field: Der Wert dieses Feldes wird vom Feld im Eingabeformat übernommen. Das SourceName-Attribut enthält den Feldnamen.

    Table: Der Wert des Feldes wird von der Übersetzungstabelle übernommen. Das SourceName-Attribut enthält den Tabellennamen.

    Lookup: Der Wert dieses Feldes wird von der Übersetzungstabelle übernommen. Das SourceName-Attribut enthält den Tabellennamen. Der zu übersetzende Wert wird vom LookupValue-Attribut und nicht vom Eingabeformat übernommen.

    Constant: Der Wert des Feldes ist konstant, und sein Wert befindet sich im ConstantValue-Attribut.
  - SourceName: Enthält den Feldname zum Übersetzen des Tabellennamens
  - Type: [integer/real/string/time] Nur erforderlich, wenn der Typ des Feldes nicht vordefiniert wird (durch Feldnamen oder durch SourceType). In der Onlineschnittstelle nur für Wertfeld mit SourceType=constant erforderlich. In der Offlineschnittstelle nur für jedes Feld mit SourceType=constant erforderlich.
  - IsKey: Repräsentiert den einmaligen Schlüssel des Events. Dieser Schlüssel wird von Feldern zusammengesetzt, die als TranslatorFields?IsKey = "yes" gekennzeichnet wurden.

    (Siehe Event-Besonderheit)
  - LookupValue: Enthält den Suchwert, wenn SourceType="lookup".
  - ConstantValue: Enthält den konstanten Wert, wenn SourceType=constant. Wenn der Typ des Feldes "Time" ist, wird der konstante Wert formatiert, entsprechend dem "TimeFormat" oder "Now" oder "NowUtc", wo die aktuelle Zeit "Now" im Datenquellen-Gebietsschema ist und "NowUtc", wo die aktuelle Zeit in UTC ist.

- TimeFormat: Enthält das TimeFormat, nur für Felder mit SourceType=constant und Type=time erforderlich.
- TimeShift: Definiert die Zeitverschiebung in Sekunden, nur für Zeitfelder.
- TimeShiftFieldName: (optional) Enthält den Feldnamen vom Eingabeformat, das eine Zeitverschiebung in Sekunden enthält. TimeShift und TimeShiftFieldName können zusammen sein.

Anhang D: Definieren von Business-Logik-Formeln (Business-Logik-Experte)

Dieses Kapitel enthält folgende Themen:

Was beim Erstellen von Business-Logik-Formeln zu vermeiden ist (siehe Seite 349)
Geclusterte Metriken und Ressourceneffektivität (siehe Seite 349)

Was beim Erstellen von Business-Logik-Formeln zu vermeiden ist

Beim Erstellen von Business-Logik-Formeln sollten Sie folgende Punkte berücksichtigen:

- Weisen Sie einer globalen Variablen nie einen Nullwert zu. Einen ungültigen Wert zuzuweisen kann dazu führen, dass der PSLWriter fehlschlägt, wenn die Metrik berechnet wird. Wenn ein nicht initialisierter Wert benötigt wird, verwenden Sie stattdessen "Empty".
- Vermeiden Sie es, Bild- und Vektorobjekte in geclusterten Metriken zu verwenden. Wenn Sie diese Objekte verwenden müssen, sollten sie so klein wie möglich bleiben. Geclusterte Metriken mit großen Abbildungen und Vektorgrafiken nehmen eine lange Zeit für eine Berechnung in Anspruch.

Geclusterte Metriken und Ressourceneffektivität

Was ist eine geclusterte Metrik?

Geclusterte Metriken ermöglichen die Definition einer Metrik, um für jedes einzelne Mitglied einer Ressourcengruppe die gleiche Definition und Logik für einen Satz an Elementen anwenden zu können. Ein Clustern kann entweder statisch auf einem vordefinierten Satz an Ressourcen oder dynamisch auf die Ressourcengruppenmitglieder durchgeführt werden, während die Gruppe im Laufe der Zeit verändert werden kann und Mitglieder aus der Gruppe ein- oder ausgeschlossen werden können.

Eine Ressource oder Ressourcengruppe kann im gleichen Berechnungszeitraum (Tag, Monat, Jahr etc.) jederzeit und immer wieder neu aus der Gruppe ausgeschlossen und wieder aufgenommen werden.

Was geschieht in der Business-Logik, wenn ein Cluster-Element aus der Cluster-Basisgruppe entfernt wird?

OnPeriodEnd-Methode und Ergebnisfunktion werden für das Cluster-Element ausgelöst. Wenn dies in der Mitte vom Berechnungszeitraum geschieht, wird das Ergebnis nur in die Datenbank geschrieben, wenn der ursprüngliche Berechnungszeitraum vorbei ist (z. B. Monatsende, Jahresende).

Was geschieht in der Business-Logik, wenn ein Cluster-Element der Cluster-Basisgruppe beitritt?

Globale Variablen werden initiiert, und die Methoden "OnLoad", "OnRegistration" und "OnPeriosStart" werden für das Cluster-Element ausgelöst

Was geschieht in der Business-Logik, wenn ein Cluster-Element der Cluster-Basisgruppe beitritt, nachdem es im gleichen Berechnungszeitraum aus der Gruppe gelöscht wurde?

Das Ergebnis, das für den Zeitraum festgelegt wurde, in der das Cluster-Element Teil der Gruppe war, wird mit dem neuen Ergebnis überschrieben. Mit anderen Worten: Das Ergebnis bezieht sich am Ende des Berechnungszeitraums nur auf den letzten Teil im berechneten Zeitraum, in dem das Cluster-Element Teil der Gruppe ist.

Was ist der Einfluss des effektiven Attributs einer Ressource auf die Business-Logik?

Während der Zeit, in der eine Ressource nicht effektiv ist, werden keine Rohdaten für die nichteffektive Ressource erfasst.

Was ist der Einfluss des effektiven Attributs einer Ressource auf das Clustern?

Die Veränderung einer Ressource, um nicht effektiv zu werden, hat die gleiche Auswirkung auf das Clustern wie der Ausschluss der Ressource aus der Gruppe. Das Clustern verhält sich in der gleichen Weise sowohl für die Ressourceneffektivität als auch für die Gruppenmitgliedschaft.

Wie sollten Sie Ausnahmen auf eine Ressource implementieren? Ist die Nutzung der Ressourceneffektivität die richtige Methode?

Es gibt einige Geschäftsfälle, in denen ein Ausnahmezeitraum auf einer individuellen Ressource festgelegt werden sollte. Zum Beispiel kann ein Server in die Wartung gehen und sollte aus den Berechnungen für den Wartungszeitraum nicht beachtet werden.

Da die Business-Logik Rohdaten-Events einer nichteffektiven Ressource ignoriert, können Sie zum Implementieren von Ausnahmen auf einer Ressource den Effektivitätsmechanismus auswählen. Er kann für einige Anwendungsfälle funktionieren. Wenn jedoch die Ressource Teil einer geclusterten Metrik ist und die Ressource im gleichen Berechnungszeitraum effektiv und nichteffektiv wird, wird nur der letzte Zeitraum, für den die Ressource effektiv war, im Ergebnis berücksichtigt (wie oben angegeben). In diesem Fall wird empfohlen, die anwenderspezifische Attributfunktion zu verwenden. Ein zusätzliches Attribut für die Ressource, die den Status der Ressource angibt, kann verwaltet werden. Die Business-Logik-Formel fragt dann den Status der Ressource ab, wo auch immer relevant im Skript.

Terminologieglossar

Abhängige Metrik

Eine Metrik, die aus einer Vertragsvorlage oder Service Level-Vorlage erstellt wurde.

Adapter

Die Schnittstelle zwischen CA Business Service Insight und Datenquellen von Drittanbietern. Adapter übersetzen die aus diesen Datenquellen stammenden Daten in ein Format, das sich für die Service Level-Berechnungen eignet, die den Vertragsparteien zur Verfügung gestellt werden.

Agent

Ein Objekt, das eine Metrik in einer bestimmten Zeiteinheit repräsentiert.

Aktuelle Status-Engine

Ein eigenständiger Prozess, der aktuelle Statusmetriken berechnet. Es kann eine unbegrenzte Anzahl von Instanzen auf einem oder mehreren Computern ausgeführt werden.

Alarm

Benachrichtigungen an Anwender über Systemevents. Ein Alarm ermöglicht es den Anwendern, Abhilfemaßnahmen zu ergreifen, um Vertragsverletzungen, drohende Vertragsstrafen und dergleichen abzuwenden.

Alarmempfangsgerät

Geräte im Netzwerk, die per E-Mail, Einblendfenster, SMS oder durch Programme von Drittanbietern Alarmmeldungen erhalten.

Alarmprofil

Eine Reihe von Parametern, die die Bedingungen definieren, aufgrund derer eine Alarmmeldung ausgelöst wird und die festlegen, an welche Anwender sie ausgegeben und wie sie verschickt wird.

Änderungsprotokoll

Ein Protokoll, das alle Aktivitäten eines Vertrags verzeichnet.

Änderungssatz

Veränderungen in der Ressourcenzuordnungsgrafik des Service-Providers.

Archivierter Vertrag

Ein Vertrag, dessen Daten im Archiv abgelegt wurden. Archivierte Verträge sind nicht Bestandteil der Kalkulationen und erscheinen nicht in den Berichten.

Audit-Trail

Eine chronologische Aufzeichnung der CA Business Service Insight-Anwender- und Systemaktivitäten, die im System gespeichert werden. Das Audit kann später überprüft werden, um Anwenderaktionen im System oder Prozesse, die sich im System ereignet haben, zu lokalisieren.

Ausnahme

Zeitraum, der bei der Kalkulation der Service Levels nicht zählt. Zum Beispiel Ausfallzeiten.

Berichtassistent

Die grafische Anwenderschnittstelle (GUI) für die Definition der Berichtsparameter.

Berichtsgruppen-Bericht

Ein Bericht, der mehrere reguläre Berichte nebeneinander enthält

Berichtsordner

Der Berichtsordner ist ein Behältnis für die Zusammenstellung von Berichten, die in einer bestimmten Beziehung zueinander stehen.

Beziehung

Eine Einheit, die eine bestimmter Verbindung zwischen zwei Metriken beschreibt und bestimmte Eigenschaften aufweist.

Booklet

RTF-Datei, die Vertragsdaten und zugehörige Berichte in anschaulichem Format mit vorgegebenen und konfigurierbaren Entwurfsvorlagen enthalten kann.

Business-Logik-Module

Bibliothek der Skriptfunktionen, die in der Business-Logik eingesetzt werden können.

Business-Logik-Vorlage

Vordefinierte Business-LogikFormeln, die für die Definition neuer Business-Logik-Formeln verwendet werden können.

Dashboard

Eine Benutzeroberfläche für die obere Verwaltungsebene, die Echtzeitinformationen ordnet und präsentiert und leicht lesbare Berichte erstellt.

Dashboard-Zuordnung

Das Layout des Dashboards. Die Anordnung der Widgets im Dashboard.

Daten-Event

Von CA Business Service Insight-Adaptern generierte Events

Datum des Ressourceneintrags

Das Datum, an dem die Ressource in CA Business Service Insight eingegeben wurde. Dieses Datum darf nicht mit dem Gültigkeitsdatum der Ressource verwechselt werden.

Domänenkategorie

Ein spezifischer Aspekt des Service Levels, der im Vertrag festgehalten ist. Zum Beispiel kann eine Servicedomäne namens Help Desk und eine Domänenkategorie wie Reaktionszeit des Help Desk festgelegt sein, die diesen speziellen Aspekt des Help-Desk-Service beschreibt. In der Regel definiert der Systemadministrator des Service Providers die Domänenkategorien. CA Business Service Insight stellt auch vordefinierte Domänenkategorien zur Verfügung.

Einfacher Bericht

Eine Möglichkeit, die von CA Business Service Insight berechneten Ergebnisse anhand zahlreicher Kriterien zu analysieren

Einheit

Ein Katalog messbarer Einheiten. Dies sind beispielsweise Prozentsatz und Währung.

Empfangene Events

Eine Liste der Events, die von anderen Metriken empfangenen wurden, die im Fenster "Business-Logik-Umfang" angezeigt wird, wenn Sie auf die Registerkarte "Empfangene Events" klicken.

Event-Anmerkungen

Zusätzliche Informationen über ein bestimmtes Event. Event-Anmerkungen werden automatisch oder manuell festgelegt (in den Berichten).

Event-Typen

Ein Event ist eine Messung, die von einem Messungstool eines Drittanbieters an einer Ressource vorgenommen und anschließend vom Adapter in ein Format konvertiert wird, das von CA Business Service Insight verwendet werden kann. Event-Typen legen fest, wie diese Events für CA Business Service Insight definiert und formatiert werden.

Freiformbericht

Ein Bericht, der von einer benutzerdefinierten SQL-Anweisung basierend auf einer CA Business Service Insight-Datenbank oder anderen externen Datenbanken erstellt wird.

Geclusterte Metrik

Ein Metriktyp, der für mehrere Ressourcen oder Ressourcengruppen gilt.

Granularität

Die Granularität bestimmt die zusätzlichen Zeiteinheiten, für die der Metrikautor Ergebnisse erhalten möchte (außerhalb des definierten Kontrollzeitraums der Metrik).

Gruppenbericht

Ein Bericht, der verschiedene Berichte auf einer Seite nebeneinander anzeigt.

Gültigkeitsdatum der Ressource

Das Datum, ab dem CA Business Service Insight die Ressource verwenden kann. Das Gültigkeitsdatum der Ressource wird durch die Version bestimmt, die die Ressource enthält. Die Ressource wird nicht von einer Ressourcenversion erkannt, deren Gültigkeitsdatum vor der Ressourcenversion liegt, mit der die Ressource erstellt wurde.

Informative Metrik

Eine Metrik, die informative Berechnungen nur für Berichtszwecke vornimmt.

Kombibericht

Mehrere reguläre Berichte, die als Mehrfachserie in einem Diagramm angezeigt werden

Kommentar zur Ursache

Ein im Rahmen der Business-Logik argumentierender Kommentar, der die Service Level-Resultate erläutert. Kommentare zur Ursache sind mit Metriken verknüpft.

Kontrollzeitraum

Der Zeitraum, in dem CA Business Service Insight den bereitgestellten Service Level misst und mit dem im Vertrag definierten Service Level-Ziel vergleicht. Durch die Messungen, die in diesem Zeitraum angestellt werden, wird ermittelt, ob eine Abweichung vom definierten Ziel vorliegt (nach der Definition der Business-Logik) und ob Vertragsstrafen oder Boni angefallen sind. Der Kontrollzeitraum bestimmt auch die Art, wie Geschäftsberichte erstellt werden.

KPI

Hauptleistungsindikator. Ein aussagekräftiger Messwert, der einzeln oder in Verbindung mit anderen Hauptleistungsindikatoren eingesetzt werden kann, um zu überwachen, wie gut ein Unternehmen oder ein Service seine quantitativen Zielsetzungen realisiert.

KQI

Qualitätskennzahl Ein aussagekräftiger Messwert, der einzeln oder in Verbindung mit anderen Hauptleistungsindikatoren eingesetzt werden kann, um zu überwachen, wie gut ein Unternehmen oder ein Service seine qualitativen Zielsetzungen realisiert.

Manuell eingefügter Kommentar zur Ursache

Ein Kommentar, der manuell durch den Berichtsbetrachter eingerichtet wird, um das Service Level-Resultat einer bestimmten Metrik zu erklären oder zusätzliche Informationen zu geben. Manuell eingefügte Kommentare zur Ursache können von allen Berichtsbetrachtern ein- und derselben Metrik eingesehen werden.

Maßeinheit

Eine Maßeinheit für sämtliche für eine Domänenkategorie definierten Metriken.

Metrik

Eine Kombination mehrerer Parameter, die den Ziel-Service Level für einen bestimmten Service zu einer bestimmten Zeit definieren. Jede Metrik ist mit einer bestimmten Servicedomäne verknüpft. Zu den Feldern und Attributen der Metrik gehören die Domänenkategorie, die Service Level-Formel, der Service, das Zeitfenster, das Service Level-Ziel und der Kontrollzeitraum.

Metrik für Verbrauch

Eine Metrik, die Ihnen die Möglichkeit gibt, den Verbrauch und die Preise separat im Bericht aufzuführen.

Metrik-Assistent

Eine benutzerfreundliche Oberfläche, die entwickelt wurde, um dem Benutzer das problemlose Bearbeiten von Metriken zu ermöglichen, wenn die Metrik zum ersten Mal aus einer Service Level-Vorlage in den Vertrag eingefügt wird.

Metrik-Event

Ein von einer Metrik in CA Business Service Insight generiertes Event.

Metrikparameter

Die Parameter einer Metrik.

Metriktyp

Beschreibt den Grund für die Berechnung einer bestimmten Metrik und definiert die Liste der Felder und deren Verhalten als erforderliche Felder und Standardwerte, die Metriken eines bestimmten Typs aufweisen müssen.

OLA

Operational Level Agreement. Ein OLA ist ein Service Level-Agreement, in dem der Lieferant die interne Organisation und der Kunde eine interne Geschäftseinheit ist.

Paket

Eine Sammlung von Service Level-Vorlagen und Vorlagen, die zu einem Paket zusammengefasst wurden, um in einer anderen CA Business Service Insight-Instanz installiert und entpackt zu werden.

Parameter

Ein Wert, der unabhängig von der Business-Logik gesetzt werden kann, um die Business-Logik-Formel zu beeinflussen. Parameter werden von der Business-Logik verwendet, um die Service Level-Resultate zu definieren. Ein Parameter kann zu den Typen Text, Liste, Zahl, Datum oder Tabelle gehören. Beispiele für Parameter sind Grenzwerte und Ressourcennamen.

Parameter auf Vertragsebene

Ein Parameter in einem regulären Vertrag, einer Vertragsvorlage oder einer Service Level-Vorlage, der im Rahmen der Business-Logik von Metriken verwendet wird.

Preiselement

Eine Metrik für die Preisberechnung. Die Preise können verbrauchsbasiert oder als Festpreise vorgegeben werden.

Regionale Einstellungen

Spezifische Details für die geografische Lage der Vertragspartei. Zu den Parametern gehören Sprache, Währung, Zeitunterschied von GMT, Sommerzeit und Datumsformat.

Ressource

Ein Element in der Gesamtinfrastruktur des Service-Providers. Beispiele für Ressourcen sind einzelne Server, Switches, Hubs, Router, ein Help Desk oder jedes andere messbare Element. Mehrere Ressourcen können als Teil einer Ressourcengruppe zusammengefasst werden, die dann selbst wieder als Ressource fungiert.

Ressourcengruppe

Ein Sammlung von Ressourcen, die als logische Einheit zusammengefasst werden. Eine Ressourcengruppe kann eine oder mehrere einzelne Ressourcen oder Ressourcengruppen enthalten.

Ressourcentyp

Eine integrierte Kategorie von Ressourcen. Einer Ressource muss mindestens ein Ressourcentyp zugewiesen sein.

Ressourcenzuweisung

Durch die Zuweisung einer Ressource zu einem Service wird der Ressourcen-Events-Stream zum betreffenden Service geleitet. Eine Ressource kann einem Service, einer Vertragspartei, einem Typ oder einer Gruppe zugewiesen werden.

Rohdatenereignis

Ein von Rohdaten in CA Business Service Insight generiertes Event

Rolle

Eine Reihe von Verantwortlichkeiten, Aktivitäten und Berechtigungen. Die Anwenderrolle definiert den Anzeigemodus von CA Business Service Insight. Jedem Anwender von CA Business Service Insight werden eine oder mehrere Rollen zugewiesen. Diese Rollen bestimmen, welche Aktionen der Anwender in CA Business Service Insight ausführen kann. Aktionen, die von der Rolle nicht durchgeführt werden dürfen, werden beim Zugriff des Anwenders auf die Anwendung auf der CA Business Service Insight-Anwenderschnittstelle nicht angezeigt.

Service Level-Vorlage

Eine Servicedefinition ist eine vordefinierte Gruppe von Services und Metriken, die verwendet wird, um die Erstellung und Änderung von Verträgen gemäß den gängigen Geschäftsprozesse zu erleichtern.

Servicekatalog

Eine Sammlung aller Informationen zu Services, Domänen und Einheiten, Vorlagenbibliothek und Service Level-Vorlagen in CA Business Service Insight.

SLO

Service Level-Ziel Dient als Messgröße für vertragliche Vereinbarungen.

Übersetzungen

Die Umwandlung der von Adaptern aus den Datenquellen abgefragten Daten in Entitäten, die in CA Business Service Insight definiert wurden.

Übersetzungseintrag

Stellt eine Definition von Quell- und Zielwerten dar, die von der Übersetzungstabelle verwendet werden.

Übersetzungsskript

Um die Pflege der Übersetzungen zu erleichtern und Fehler zu vermeiden, können Sie mit Übersetzungsskripts arbeiten.

Übersetzungstabelle

Eine Einrichtung für die Übersetzung von Werten aus den Rohdaten in die im System definierten Daten.

Underpinning Contract

Underpinning Contract. Ein Vertrag mit einem externen Lieferanten über die Bereitstellung von Services zur Unterstützung der Organisation bei der Service-Bereitstellung.

Verbrauch

Ein Metriktyp für die Verbrauchsberechnung. Wird hauptsächlich für das Finanzmodul eingesetzt. Wenn dieser Metriktyp verwendet wird, kann man den Verbrauch und die Preise in den Berichten separat darstellen. Der Verbrauch kann auch mit der Preiselementmetrik berechnet werden.

Verhältnis

Ein Typ einer Einheit.

Vertragsautor

Der Anwender, der für die Erstellung eines bestimmten Vertrags verantwortlich zeichnet.

Vertragspartei

Der Kunde oder Provider, mit dem der Vertrag geschlossen wird. Die Vertragspartei kann auch eine interne Einheit innerhalb einer größeren Organisation sein.

Vertragsparteiengruppe

Eine logisch definierte Gruppe von Vertragsparteien (Kunden). Die Herstellung eines Gruppenbezugs für eine einzelne Vertragspartei verkürzt die Vertragserstellung. Eine Vertragspartei kann zu mehr als einer Vertragsparteiengruppe gehören.

Vertragsstrafe

Die Entschädigung, die der Vertragspartei geschuldet wird, wenn das Service Level-Ziel innerhalb eines bestimmten Kontrollzeitraums nicht erreicht wird. Vertragsstrafen können in CA Business Service Insight ein- und ausgeschaltet werden.

Vertragsvorlage

Vordefinierter Vertrag, der für die Erstellung eines neuen Vertrags verwendet wird.

Zielvorgabe

Die logische Beschreibung einer Metrik mit den Parametern, die die Metrik definieren. Zielvorgaben werden in der grafischen Anwenderschnittstelle von CA Business Service Insight angezeigt, um den wesentlichen Inhalt der Metrik schnell erfassen zu können.

Zugehörige Metrik

Eine Metrik, auf die in einer Beziehung als Ziel verwiesen wird.

Zwischengröße

Eine Metrik, die Events für die Berechnung von Events generieren kann. Interim-Metriken können nicht berechnet werden.