

CA Business Service Insight

Guia de Implementação

8.2.5



A presente documentação, que inclui os sistemas de ajuda incorporados e os materiais distribuídos eletronicamente (doravante denominada Documentação), destina-se apenas a fins informativos e está sujeita a alterações ou revogação por parte da CA a qualquer momento.

A Documentação não pode ser copiada, transferida, reproduzida, divulgada, modificada ou duplicada, no todo ou em parte, sem o prévio consentimento por escrito da CA. A presente Documentação contém informações confidenciais e de propriedade da CA, não podendo ser divulgadas ou usadas para quaisquer outros fins que não aqueles permitidos por (i) um outro contrato celebrado entre o cliente e a CA que rege o uso do software da CA ao qual a Documentação está relacionada; ou (ii) um outro contrato de confidencialidade celebrado entre o cliente e a CA.

Não obstante o supracitado, se o Cliente for um usuário licenciado do(s) produto(s) de software constante(s) na Documentação, é permitido que ele imprima ou, de outro modo, disponibilize uma quantidade razoável de cópias da Documentação para uso interno seu e de seus funcionários referente ao software em questão, contanto que todos os avisos de direitos autorais e legendas da CA estejam presentes em cada cópia reproduzida.

O direito à impressão ou, de outro modo, à disponibilidade de cópias da Documentação está limitado ao período em que a licença aplicável ao referido software permanecer em pleno vigor e efeito. Em caso de término da licença, por qualquer motivo, fica o usuário responsável por garantir à CA, por escrito, que todas as cópias, parciais ou integrais, da Documentação sejam devolvidas à CA ou destruídas.

NA MEDIDA EM QUE PERMITIDO PELA LEI APLICÁVEL, A CA FORNECE ESTA DOCUMENTAÇÃO "NO ESTADO EM QUE SE ENCONTRA", SEM NENHUM TIPO DE GARANTIA, INCLUINDO, ENTRE OUTROS, QUAISQUER GARANTIAS IMPLÍCITAS DE COMERCIALIZABILIDADE, ADEQUAÇÃO A UM DETERMINADO FIM OU NÃO VIOLAÇÃO. EM NENHUMA OCASIÃO, A CA SERÁ RESPONSÁVEL PERANTE O USUÁRIO OU TERCEIROS POR QUAISQUER PERDAS OU DANOS, DIRETOS OU INDIRETOS, RESULTANTES DO USO DA DOCUMENTAÇÃO, INCLUINDO, ENTRE OUTROS, LUCROS CESSANTES, PERDA DE INVESTIMENTO, INTERRUPÇÃO DOS NEGÓCIOS, FUNDO DE COMÉRCIO OU PERDA DE DADOS, MESMO QUE A CA TENHA SIDO EXPRESSAMENTE ADVERTIDA SOBRE A POSSIBILIDADE DE TAIS PERDAS E DANOS.

O uso de qualquer produto de software mencionado na Documentação é regido pelo contrato de licença aplicável, sendo que tal contrato de licença não é modificado de nenhum modo pelos termos deste aviso.

O fabricante desta Documentação é a CA.

Fornecida com "Direitos restritos". O uso, duplicação ou divulgação pelo governo dos Estados Unidos está sujeita às restrições descritas no FAR, seções 12.212, 52.227-14 e 52.227-19(c)(1) - (2) e DFARS, seção 252.227-7014(b)(3), conforme aplicável, ou sucessores.

Copyright © 2013 CA. Todos os direitos reservados. Todas as marcas comerciais, nomes de marcas, marcas de serviço e logotipos aqui mencionados pertencem às suas respectivas empresas.

Entrar em contato com o Suporte técnico

Para assistência técnica online e uma lista completa dos locais, principais horários de atendimento e números de telefone, entre em contato com o Suporte técnico pelo endereço <http://www.ca.com/worldwide>.

Índice

Capítulo 1: Introdução 9

Funções	9
Gerenciador do catálogo de serviços	10
Gerente de contratos	10
Especialista em lógica de negócios	11
Especialista em fonte de dados	11
Administrador do sistema	12
Processo da solução	12
Planejamento	13
Design	14
Implementação	15
Instalação e implantação	15

Capítulo 2: Planejamento e Design 17

Sessão de coleta de requisitos (Todas as funções)	18
Coletando dados de amostra (Especialista em fonte de dados)	20
Design	21
Introdução ao design (Gerente de contratos, Especialista em lógica de negócios, Especialista em fonte de dados)	22
Modelagem de contratos (Gerente de contratos)	23
Saídas da etapa de modelagem de contratos (Gerente de contratos e Especialista em fonte de dados)	42
Modelagem de dados (Especialista em fonte de dados e Especialista em lógica de negócios)	44
Saídas da etapa de modelagem de dados (Especialista em fonte de dados e Especialista em lógica de negócios)	59

Capítulo 3: Implementação 61

Implementação - Introdução	61
Configurando a estrutura (Gerente de contratos)	64
Configurando a biblioteca de modelos (Gerente de contratos)	64
Como criar contratos (Gerente de contratos)	65
Criar contratos de um serviço	66
Criar modelos de nível de serviço	68
Ciclo de vida do contrato e controle de versão	68
Coleta de dados (Especialista em fonte de dados)	70
Funcionalidade do conector	71
Ambiente do conector	72

Arquivos principais	75
Arquivos de trabalho	75
Comunicação do conector	82
Configurações de registro do conector	85
Modos de execução do conector	87
Ferramentas de configuração e manutenção	89
Como configurar conectores e conversões	89
Conversões automáticas com scripts de conversão	127
Tópicos avançados para o recurso do conector	128
Criação de script de lógica de negócios (Especialista em lógica de negócios)	136
Fluxo de trabalho da criação de script de lógica de negócios	138
Módulos de lógica de negócios	139
Dentro da lógica de negócios	140
Como ativar os contratos (Gerente de contratos)	177
Recálculo completo das métricas do contrato	179
Criar produtos a serem entregues (Gerente de contratos)	180
Definir configurações de segurança (administrador)	180
Como criar relatórios	181
Configurar páginas do painel	194
Adicionar perfis de alerta de nível de serviço	197

Capítulo 4: Instalação e implantação **201**

Introdução	202
Preparações	204
Instalação	206
Como importar ou exportar entre ambientes (especialista em fontes de dados)	207
Integração - Instalação do servidor de email (Administrador do sistema)	209
Definir preferências do sistema (Administrador do sistema)	210
Configurações de segurança (Administrador do sistema)	210
Especificar configurações de sincronização do SSA	211

Apêndice A: Domínios de serviço e categorias de domínio da amostra **213**

Apêndice B: Exemplos de estudo de caso **215**

Exemplos de modelagem do contrato	215
Estudo de caso 1: Disponibilidade do sistema	216
Estudo de caso 2: Disponibilidade do sistema 2	217
Estudo de caso 3: Tempo de resposta do sistema	219
Estudo de caso 4: Desempenho da Central de atendimento	223
Estudo de caso 5: Backup do sistema	225

Exemplo de modelagem de métricas financeiras	226
Estudo de caso 6: Condições de modelagem financeira de um contrato/serviço	226
Exemplos de modelagem de dados.....	234
Estudo de caso 7: Desempenho do Servidor.....	234
Estudo de caso 8: Desempenho da Central de atendimento.....	237
Usando exemplo de atributos personalizados	244
Estudo de caso 9: Vários objetivos dinâmicos	244
Exemplos de script de conversão	247
Estudo de caso 10: Conversão automática básica	247
Estudo de caso 11: Atualizações do modelo de recurso	250
Exemplos de script de lógica de negócios.....	253
Estudo de caso 12: Usando a lógica do contador para calcular o número de falhas.....	254
Estudo de caso 13: Tratamento do grupo dinâmico do componente.....	259
Estudo de caso 14: Controle do relógio de acumulação de tempo	264
Criando exemplos de lógica de negócios efetivas.....	270
Estudo de caso 15: Métricas agrupadas.....	273
Estudo de caso 16: padrões de design da lógica de negócios.....	277
Estudo de caso 17: Funcionalidade integrada.....	282
Estudo de caso 18: Registro	285
Caso de estudo 19: assistente de conector para fonte de dados com base em arquivo	288
Criação do conector	290
Implantação de conectores.....	300
Programação de conectores	301
Estudo de caso 21: exemplo de integração LDAP	303
Estudo de caso 22: gerando relatórios com o PERL	310

Apêndice C: Especificações de configuração do conector **313**

Estrutura geral.....	313
Seção Geral.....	314
Seção Interface do CA Business Service Insight	320
Seção DataSourceInterface	323
Seção da interface SQL.....	326
Seção InputFormatCollection	330
Seção TranslationTableCollection	335
Seção TranslatorCollection.....	337

Apêndice D: Definindo fórmulas de lógica de negócios (especialista em lógica de negócios) **341**

O que evitar ao criar fórmulas de lógica de negócios	341
Vigência de métricas agrupadas e recursos	341

Capítulo 1: Introdução

Este guia aborda o processo de implementação e as funções, responsabilidades e interações relacionadas e envolvidas no planejamento, design, implementação, instalação e implantação do CA Business Service Insight.

Esta seção contém os seguintes tópicos:

[Funções](#) (na página 9)

[Processo da solução](#) (na página 12)

Funções

Uma função é uma pessoa que executa um conjunto comum de tarefas que são executadas durante uma implementação típica. A palavra função também pode se referir ao próprio conjunto de tarefas. O CA Business Service Insight tem uma série de funções predefinidas que podem ser editadas pelo administrador do sistema. Além disso, funções e permissões específicas podem ser criadas.

As funções a seguir são necessárias para a implementação:

- Gerenciador do catálogo de serviços
- Gerente de contratos
- Especialista em lógica de negócios
- Especialista em fonte de dados
- Administrador do sistema

As responsabilidades e qualificações de cada função são descritas abaixo.

Observação: a mesma pessoa pode executar diversas funções, como definido pelo administrador do sistema.

Gerenciador do catálogo de serviços

Responsabilidades:

- Gerenciar o catálogo de serviços da organização.
- Definir o catálogo de ofertas de serviços da organização.
- Gerenciar os requisitos de definições e geração de relatórios de contrato.

Qualificações necessárias:

- Familiaridade com o processo de entrega de serviços E2E da organização.
- Familiaridade com os conceitos do CA Business Service Insight.

Gerente de contratos

Responsabilidades:

- Interação com o usuário final.
- Ser responsável pela criação do catálogo de serviços de acordo com os requisitos definidos.
- Criar contratos, mantendo os contratos existentes.
- (Opcional, recomendado) Ser responsável pela Implementação de SLA específica do cliente.

Qualificações:

- Noções básicas dos princípios e da lógica das métricas.
- Usuário especialista nos recursos com base em GUI (Graphical User Interface - Interface Gráfica do Usuário) do CA Business Service Insight.

Especialista em lógica de negócios

Responsabilidades:

- Implementar as métricas.
- Escrever scripts de lógica de negócios; manter scripts de lógica de negócios existentes.

Qualificações:

- Noções básicas de desenvolvimento e grande conhecimento profissional de linguagens de scripts, como VB-Script.
- Um bom entendimento do fluxo de dados do CA Business Service Insight.
- Especialista em criação de script de lógica de negócios do CA Business Service Insight.
- Um bom entendimento da arquitetura e dos componentes do CA Business Service Insight.
- Usuário especialista nos recursos com base em GUI (Graphical User Interface - Interface Gráfica do Usuário) do CA Business Service Insight.

Especialista em fonte de dados

Responsabilidades:

- Design da infraestrutura do CA Business Service Insight.
- Criar conectores; manter os conectores existentes, fazer interface com a infraestrutura operacional para obter medições.
- Construir e manter a infraestrutura do CA Business Service Insight.

Qualificações:

- Compreender o ambiente e a estrutura das fontes de dados do cliente.
- Bom conhecimento profissional de banco de dados, XML e linguagens de scripts.
- Compreender o fluxo de dados e os processos de coleta de dados do CA Business Service Insight.
- Especialista em conectores do CA Business Service Insight.
- Compreender a arquitetura e os componentes do CA Business Service Insight.
- Usuário especialista nos recursos com base em GUI (Graphical User Interface - Interface Gráfica do Usuário) do CA Business Service Insight.
- Noções de desenvolvimento é uma vantagem.

Administrador do sistema

Responsabilidades:

- Gerenciar a instalação e as atualizações.
- Cumprir os requisitos de hardware e software do sistema.
- Gerenciar as configurações de segurança: definições de usuário e de função.
- Monitorar e gerenciar o sistema.

Qualificações:

- Compreender as infraestruturas e redes do cliente.
- Compreender a arquitetura e os componentes do CA Business Service Insight.
- Conhecimento profissional de banco de dados, XML e linguagens de scripts.
- Compreender o fluxo de dados do CA Business Service Insight.
- Usuário dos recursos com base em GUI do CA Business Service Insight.

Processo da solução

O processo da solução inclui três etapas básicas:

- Planejamento e Design
- Implementação
- Instalação e implantação

As funções descritas anteriormente participam das diferentes etapas do processo de implementação, cada uma de acordo com suas especialidades. Essas funções interagem para concluir o processo inteiro, do início ao término, da definição contratual até o relatório de saída.

Este guia está estruturado com orientação para função e processo, de acordo com um fluxo geral de etapas que cria o processo de implementação. Descreve como cada função está envolvida e como as funções interagem.

Os parágrafos a seguir fornecem uma visão geral das etapas incluídas no processo de implementação e as atividades de cada função.

O restante do guia é estruturado por etapa, com uma indicação de quais funções participam da etapa. Cada capítulo indica a função responsável por cada atividade.

Veja a seguir uma breve descrição das tarefas básicas de cada fase e a função das diferentes funções. Mais detalhes podem ser encontrados em cada capítulo.

Planejamento

A finalidade da etapa de planejamento é identificar e quantificar os aspectos que devem ser considerados como parte da implementação.

Nessa etapa, o Gerente de contratos coleta os requisitos de negócios do Gerente de catálogo de serviços para compreender as expectativas do CA Business Service Insight após uma implementação bem-sucedida. É importante nessa etapa compreender os requisitos atuais e os planos futuros para verificar se a implementação proporciona crescimento e evolução fáceis e sem problemas.

De acordo com os requisitos de implementação definidos, é necessário que o Gerente de contratos examine o processo relacionado existente (se houver) verificando suas entradas e saídas. Nessa etapa, é preciso identificar todas as fontes de informações necessárias e obter amostras, contratos, relatórios de saída e fontes de entrada usadas para calcular as saídas existentes. Essas informações devem ser especificadas com atenção para que o Gerente de contratos identifique todas as entradas necessárias, de modo que a saída esperada possa ser derivada.

Nessa etapa, o Gerente de contratos examina os contratos para verificar se o catálogo de serviços e os modelos fornecidos como parte da implementação oferecem suporte para as necessidades existentes e futuras e se a implementação atual abrange todas as áreas do contrato.

O Gerente de contratos examina os relatórios e seus formatos para verificar se todas as medições definidas foram feitas para conseguir produzi-las com a implementação existente.

Em seguida, o Especialista em fonte de dados examina as amostras de dados de entrada por medições necessárias e cálculos fornecidos pelo Gerente de contratos. Isso permite que o Especialista em fonte de dados identifique quaisquer problemas de entrada que devem ser resolvidos, como formatos personalizados ou deficiências, e determine se fontes de dados adicionais são necessárias.

A finalidade dessa análise é verificar se as fontes contêm as informações necessárias para calcular as métricas exigidas e as informações iniciais sobre o processo de recuperação, por exemplo, os métodos de comunicação e a estrutura das fontes de dados.

Design

Durante a fase de design, os requisitos de entrada e saída coletados são convertidos na estrutura de modelo do CA Business Service Insight, que consiste em Contratos, Métricas e Recursos. Isso significa transformar os dados reais e conceituá-los de modo que se ajustem à estrutura do CA Business Service Insight.

O design do sistema inclui o seguinte:

Modelagem de contratos

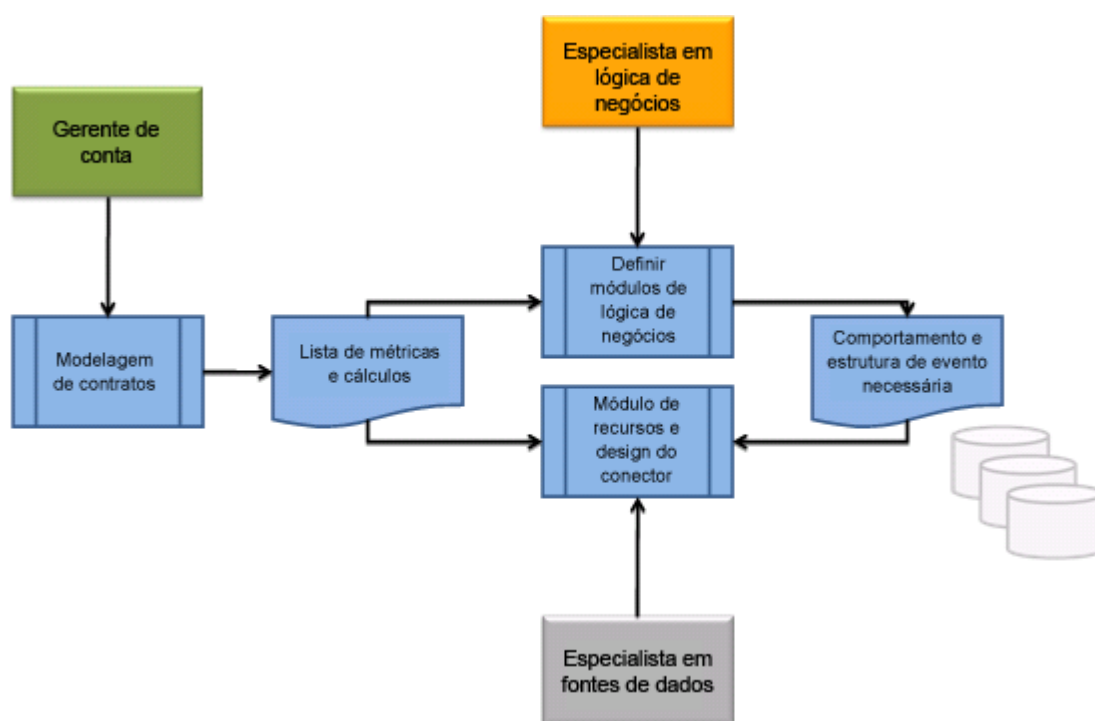
Os SLAs de clientes são interpretados nos contratos do CA Business Service Insight, e entidades do catálogo de serviços (como Pastas de modelo) são definidas. Isso é feito principalmente pelo Gerente de contratos.

Modelagem de dados

Os dados do recurso são examinados e modelados no modelo de recurso do CA Business Service Insight. Isso é feito pelo Especialista em fonte de dados e pelo Especialista em lógica de negócios.

Pode haver vários métodos disponíveis para modelagem de contratos ou de dados. No entanto, muitas vezes há uma prática ideal que não resolve apenas o problema, mas também oferece mais flexibilidade ou produtividade, fornecendo uma estrutura forte para o crescimento futuro.

Para ajudar o Especialista em fonte de dados a escolher os métodos mais adequados, estudos de caso são usados com soluções sugeridas.



Conforme mostrado no diagrama de fluxo de trabalho anterior, como resultado do processo de modelagem de contratos, o Gerente de contratos fornece como saída a lista de métricas que devem ser configuradas no sistema e a definição da estrutura de tópicos do cálculo.

Exemplo:

Cliente A, tempo médio de resposta do sistema CNP.

A lista Métricas é fornecida como uma entrada para o Especialista em lógica de negócios que define a estrutura de evento necessária e o comportamento do evento que permite definir o script de cálculo necessário.

A lista Métricas e os requisitos de estrutura e comportamento do evento são a entrada para o design do modelo de recurso e os conectores de dados pelo Especialista em fonte de dados.

Implementação

Na fase de implementação, o sistema do CA Business Service Insight é configurado com base nos resultados da fase de design. A fase de implementação envolve obter os resultados da etapa de design teórico e usá-los para criar os requisitos operacionais para o CA Business Service Insight.

Alguns desses itens a serem criados ou considerados incluem:

Gerente de contratos

- Instalação de serviço
- Criação de contrato
- Instalação de relatórios e painel

Especialista em fonte de dados

- Configuração do conector
- Instalação da infraestrutura

Especialista em lógica de negócios

- Módulos de lógica de negócios

Instalação e implantação

A fase de instalação e implantação envolve instalar e integrar o sistema de produção, testar e monitorar o desempenho, bem como treinar os usuários. O administrador do sistema e o Especialista em fonte de dados realizam a maior parte das atividades nessa etapa.

Capítulo 2: Planejamento e Design

Esta seção contém os seguintes tópicos:

[Sessão de coleta de requisitos \(Todas as funções\)](#) (na página 18)

[Coletando dados de amostra \(Especialista em fonte de dados\)](#) (na página 20)

[Design](#) (na página 21)

Sessão de coleta de requisitos (Todas as funções)

A sessão de coleta de requisitos é a primeira etapa crítica na implementação do CA Business Service Insight e deve incluir todos os funcionários-chave envolvidos na implementação.

As informações necessárias para essa sessão devem ser fornecidas por pessoas que estão muito familiarizadas com os SLAs selecionados, incluindo a lógica de negócios do SLA, como os SLAs estão sendo gerenciados no momento, que relatórios estão sendo gerados para eles e quais são os requisitos de geração de relatórios completos esperados no futuro.

Os funcionários a seguir devem participar:

- Gerente(s) de catálogo de serviços.
- Gerente(s) de contratos.
- Especialista(s) em fonte de dados.
- Especialista(s) em lógica de negócios/criação de script.
- Qualquer outro integrante relevante da equipe familiarizado com os SLAs selecionados.
- Gerente de projetos (se indicado).

A CA recomenda o seguinte:

- É importante garantir que há funcionários que possam tomar decisões relativas a quaisquer definições confusas que podem surgir durante a verificação dos SLAs selecionados.
- É recomendável que a equipe de implementação receba, antes dessa reunião, uma cópia dos SLAs selecionados para esse projeto. Os participantes devem estar familiarizados com todas as fontes de dados relevantes que estiverem relacionadas aos SLAs selecionados, conseguir fornecer despejo de dados delas e explicar suas estruturas internas, quando aplicável.

Os principais objetivos da sessão de planejamento são definir e especificar:

- Escopo do trabalho planejado.
- Critérios de êxito.
- Funções e responsabilidades de todas as partes envolvidas.
- Processo e programação da implementação.
- Requisitos necessários de hardware/software.

Para fazer isso:

- Verificar os SLAs a serem usados na implementação.
- Verificar a lógica de negócios para cada objetivo nos SLAs selecionados.

- Verificar relatórios, alertas e requisitos do painel relacionados aos SLAs selecionados.
- Verificar as fontes de dados relevantes a serem usadas na implementação.
- Verificar a topologia de infraestrutura relevante.
- Coletar despejos de dados relevantes de fontes de dados relevantes.
- Identificar as principais pessoas e suas responsabilidades.
- Definir os caminhos de comunicação para duração da implementação, reuniões de verificação, processo de perguntas e respostas, etc.
- Revisar o processo e a programação da implementação.
- Revisar o estado atual dos assuntos, isto é, como esses SLAs são gerenciados e o que está faltando.
- Definir os critérios de êxito da implementação.
- Definir os requisitos necessários de hardware/software.
- Definir a linha do tempo da implementação.

Coletando dados de amostra (Especialista em fonte de dados)

Para executar a configuração inicial de conectores externos, é importante para fins de coleta de dados obter amostras de dados antigos. Essas amostras devem ter a mesma estrutura que os dados que serão recebidos do sistema real a partir do qual o CA Business Service Insight precisa recuperar os dados. Além das amostras, o Especialista em fonte de dados deve aprender sobre a fonte de dados com o proprietário da fonte de dados para conseguir esclarecer as seguintes questões:

Tipo:

Banco de dados, arquivo, fluxo ou outros.

Local:

Onde estão? Como chegar lá (detalhes de conexão)? Obstáculos de segurança? Plataforma?

Convenção de nomenclatura:

Como esses arquivos são nomeados? (É uma fonte de dados com base em arquivos?) Como as tabelas são nomeadas? Nomes dos campos da tabela?

Disponibilidade:

Quando pode ser acessado? Quando deve ser acessado (considerações de carregamento)? Atualização contínua ou uma vez por X tempo? Por quanto tempo é persistente?

Comportamento:

Dados são acumulados? Dados são substituídos? Um arquivo é mantido?

Alcance:

Quantos dados históricos estão disponíveis?

Volumes:

Volume de dados atual? Taxa de crescimento? Previsões?

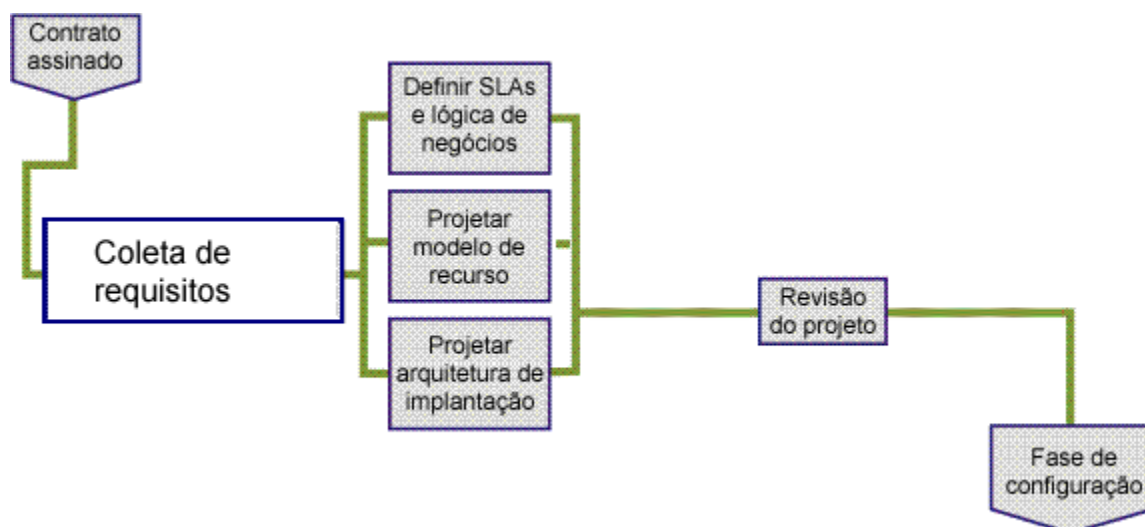
Estrutura e formato:

Como os dados são organizados na fonte de dados? Quais são os campos de dados? Quais são os nomes das tabelas? O que separa os campos de dados?

Fluxo:

O conector recebe dados ou os dados são coletados pelo conector?

Design



Introdução ao design (Gerente de contratos, Especialista em lógica de negócios, Especialista em fonte de dados)

Esta seção explica o processo e o raciocínio por trás da fase de design do processo da solução. A fase de design segue a fase de planejamento e, subsequentemente, é seguida pela fase de implementação no próximo capítulo.

O objetivo da fase de design é permitir que a equipe de implementação possa concluir o mapeamento dos contratos reais e de suas cláusulas, além dos dados de desempenho existentes no sistema do CA Business Service Insight.

Antes de iniciar esse processo, a equipe de implementação precisa estar ciente dos vários métodos e opções disponíveis, de forma a garantir que todos os requisitos são considerados e também que o design resultante será o mais otimizado possível e, ao mesmo tempo, permitir o crescimento ou alterações futuras.

O processo de design envolve a equipe de implementação que executa as seguintes etapas:

- Examinar os contratos e convertê-los nos objetos necessários do CA Business Service Insight, o que é chamado de "Modelagem de contratos" neste guia; responsabilidade do Gerente de contratos.
- Obter os conjuntos de dados existentes e decidir como e de quais extrair os itens relevantes em virtude dos resultados desejados, o que é chamado de "Modelagem de dados" neste guia; responsabilidade do Especialista em fonte de dados e do Especialista em lógica de negócios.

A seção de Modelagem de contratos explica:

- Terminologia usada (essencial para a implementação correta)
- Modelos e módulos de lógica de negócios
- Modelo de nível de serviço e como usá-lo
- Importância da criação de um catálogo de serviços forte
- Métricas de gerenciamento financeiro (Penalidades, Incentivos e custos e exemplos) aplicadas nos contratos de clientes e outras métricas contratuais

A seção de Modelagem de dados explica:

- Eventos aplicados ao CA Business Service Insight e seu fluxo pelo sistema do CA Business Service Insight
- Métricas e como elas são registradas
- Recursos do CA Business Service Insight e como identificá-los
- Sugestões de como automatizar a coleta e a definição desses recursos

Todos os pontos acima são explicadas com mais profundidade nas seções a seguir.

É importante estar ciente de que escolhas erradas feitas nesta etapa podem causar um impacto negativo na operação do CA Business Service Insight, e pode ser difícil ou impossível revertê-lo em uma etapa posterior.

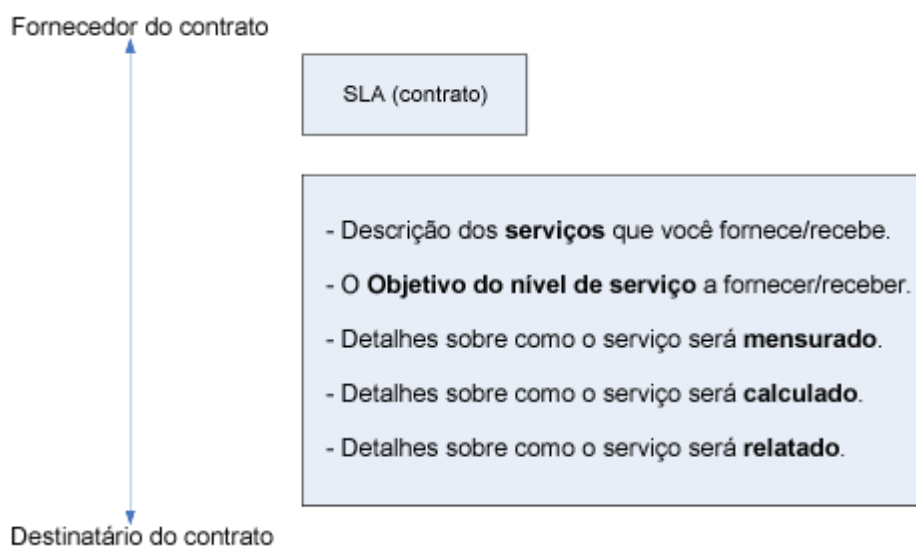
Modelagem de contratos (Gerente de contratos)

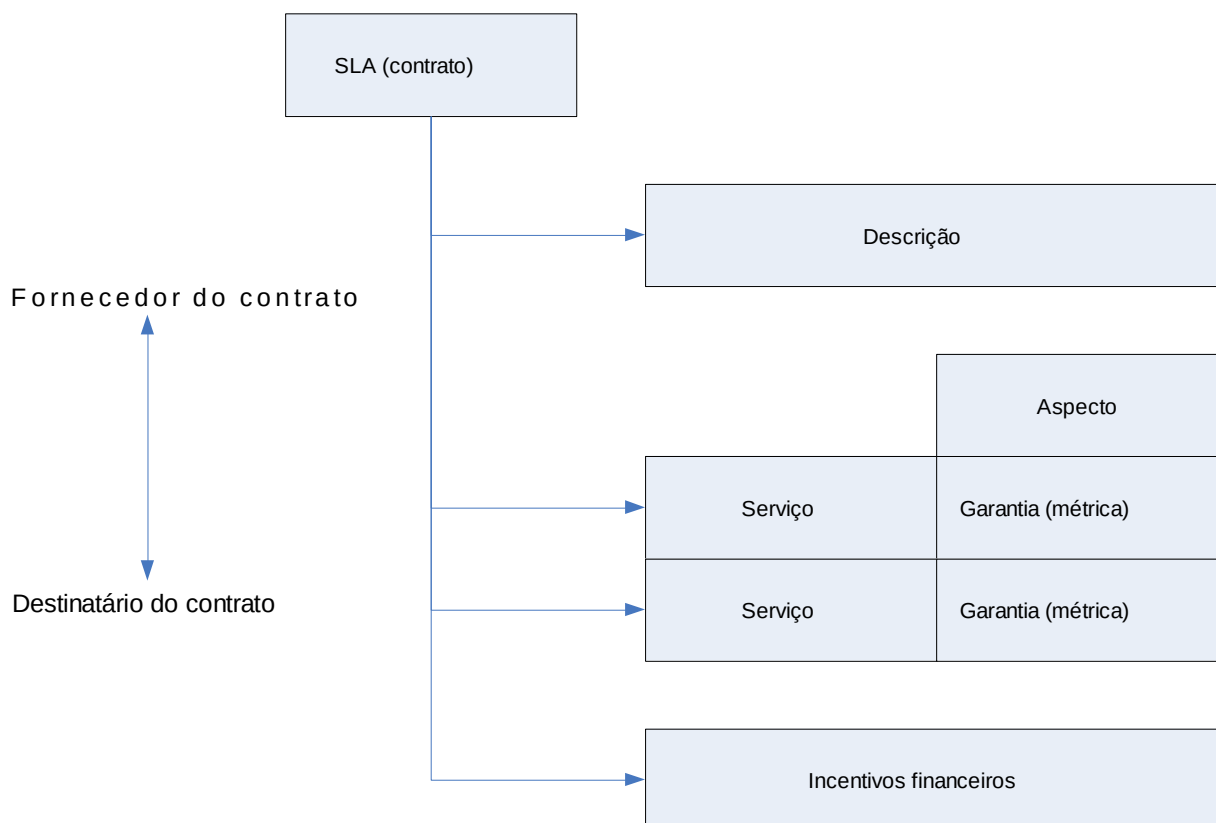
As seguintes tarefas para modelagem de contratos são executadas pelo gerente de contratos.

Terminologia do contrato

Um contrato é um conjunto de objetivos. Um objetivo pode ser contratual ou operacional (Métrica) ou financeiro (Incentivo). O processo de Modelagem de contratos envolve obter todo o conteúdo do contrato que estiver dentro do escopo da implementação e convertê-lo no modelo do CA Business Service Insight.

Os diagramas a seguir descrevem esse processo.

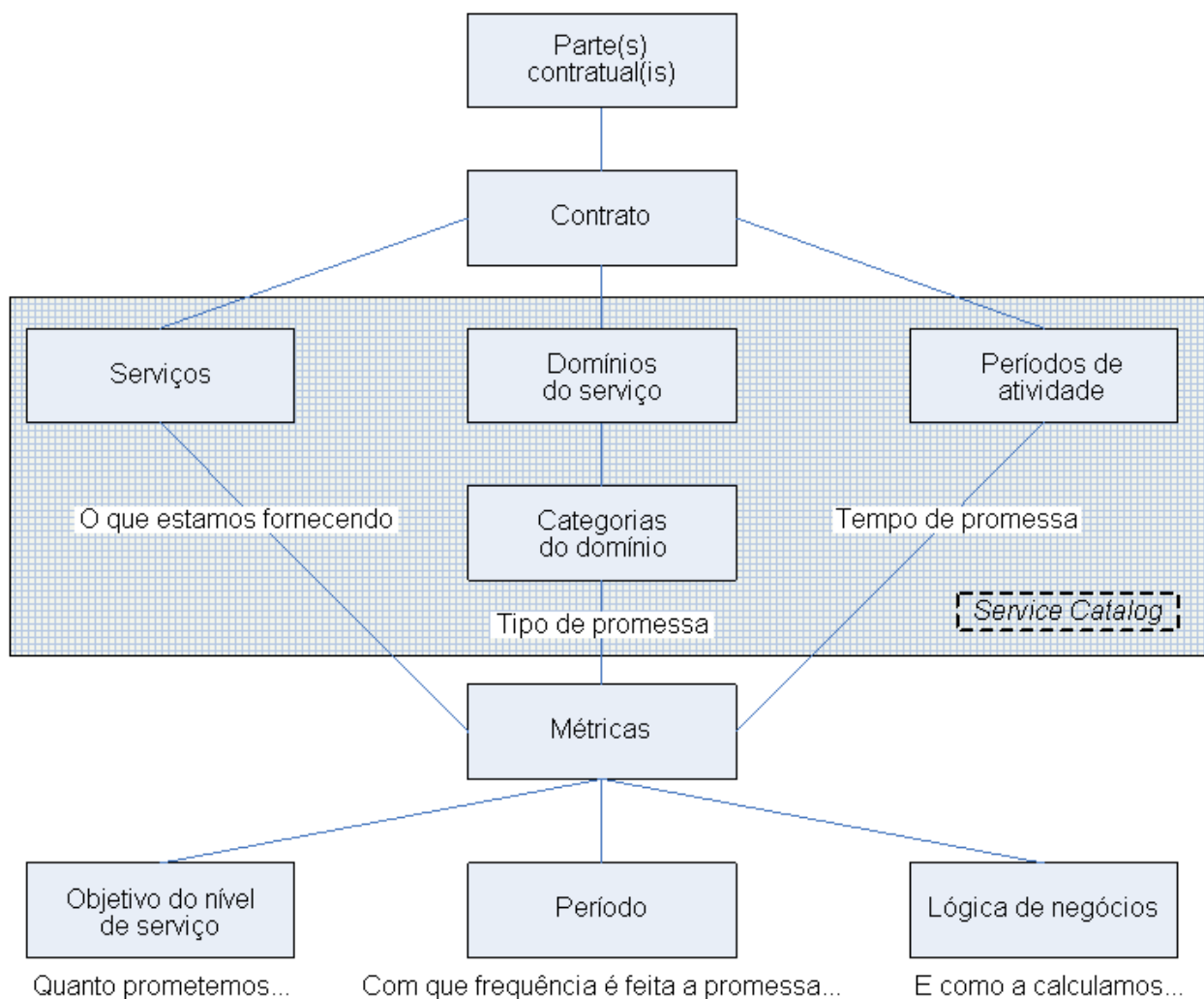




Observação: os possíveis planos futuros para o sistema devem ser considerados, embora apenas aspectos que estiverem dentro do escopo da implementação sejam modelados. O escopo deve incluir requisitos gerais de gerenciamento de contratos da organização para criar um modelo amplo o suficiente para abranger os futuros desenvolvimentos. Esse procedimento minimiza e simplifica quaisquer alterações que possam ser necessárias no futuro.

Converter contratos de clientes no modelo do CA Business Service Insight é um processo no qual o Gerente de contratos agrupa as garantias (Métricas) oferecidas dentro de limites comuns em grupos lógicos, componentes de serviço comuns, aspectos do contrato, etc. Esse agrupamento lógico permite um gerenciamento de nível de serviço flexível.

O modelo de contrato do CA Business Service Insight é composto das entidades mostradas na figura a seguir:



Contrato

Define o contrato e o conjunto de métricas. Há três tipos diferentes de contrato de acordo com o relacionamento entre as partes contratuais envolvidas. Pode ser um SLA (Service Level Agreement - Contrato de Nível de Serviço) entre a organização e seus clientes, OLA (Operational Level Agreement - Contrato de Nível Operacional) entre as divisões da organização ou um UC (Underpinning Contract - Contrato de Apoio) entre a organização e seus fornecedores, sendo que geralmente o UC é de um serviço que a organização está fornecendo a outro cliente por meio de um SLA.

Parte contratual

Define o cliente dos serviços fornecidos e com quem o contrato foi assinado. Uma única parte contratual pode estar vinculada a mais de um contrato. Observe que no contrato é possível definir o provedor e o destinatário dos serviços relacionados ao contrato.

Métricas

Define um objetivo ou garantia de um único nível de serviço. Cada métrica é uma solicitação de medição que gera o resultado real de nível de serviço no qual a geração de relatórios é executada.

Uma métrica é composta dos seguintes itens:

Tipo

Há oito tipos diferentes de métrica:

SLO	Objetivo do nível de serviço.
Informativo	Objetivo do nível de serviço sem um destino.
KPI	Principal indicador de desempenho, usado para oferecer suporte aos vários conceitos industriais. Para as empresas de telecomunicações, KPI=SLO é uma obrigação do cliente e em TI, é uma obrigação técnica.
KQI	Indicador de qualidade chave. Uma métrica geral para medir o resultado agregado de vários indicadores.
Provisório	Medição provisória para usar em outras métricas. Esses resultados de métrica não podem ser relatados.
Consumo	Usado pelas métricas financeiras para calcular o consumo de serviços/recursos. Usado em conjunto com as métricas de item de preço.
Incentivo	Métrica financeira, usada para incentivo (termo positivo para penalidade).
Item de preço	Métrica financeira, usada para calcular os resultados com base nos consumos. Preços diferentes por período ou por número de unidades.

Período de monitoramento

Período de geração de relatórios contratuais ou período no qual o resultado calculado real de nível de serviço é comparado com o destino. O período de monitoramento no CA Business Service Insight pode ser definido com uma granularidade de Hora, Dia, Semana, Mês, Trimestre ou Ano. Para fins de análise de causa raiz, além do período de monitoramento, a métrica também é calculada nas outras seis granularidades, mas os resultados desses períodos são marcados apenas como resultados operacionais.

Observação: esta opção só será executada se essas granularidades de cálculo estiverem especificadas para a métrica.

Período de atividade

Tempo no período de monitoramento durante o qual a garantia ou obrigação específica é aplicável, incluindo definições excepcionais de período de atividade, como janelas de manutenção previstas, feriados públicos padrão, etc.

Lógica de negócios

Fórmula que define como avaliar os dados brutos para calcular o valor do nível de serviço real para determinado aspecto do serviço e as premissas específicas que o cálculo deve considerar. Durante a fase de design, apenas a estrutura de tópicos do cálculo é identificada. É definida com mais detalhes durante a fase de configuração.

Destino

Obrigação de nível de serviço contratual. O destino pode ou não ser obrigatório, dependendo do tipo de métrica em questão. No caso em que um destino não estiver definido, a métrica fornecerá resultados apenas para fins de geração de relatórios, e não será possível compará-los com uma obrigação contratual ou garantia. Os destinos podem ser estáticos ou dinâmicos.

Observação: para seguir alguns dos conceitos apresentados até o momento, consulte o [Apêndice B - Estudos de caso](#) (na página 215).

Cada métrica está relacionada com as seguintes entidades de todo o sistema contidas na pasta de modelos do sistema:

Componente de serviço

Descreve o que é uma obrigação fornecer.

Grupo de componentes de serviço

Conjunto de componentes de serviço. Um único componente de serviço pode fazer parte de mais de um grupo. Um grupo de componentes de serviço é uma entidade opcional e, se usado, pode fornecer mais flexibilidade para a geração de relatórios.

Domínio do serviço

Aspecto específico dos componentes de serviço que precisam ser medidos para fins de monitoramento de nível de serviço (por exemplo, desempenho ou disponibilidade).

Categoria do domínio

Unidade de medida específica. Define a unidade de medida padrão e se o objetivo de destino é um valor mínimo ou máximo. Basicamente, é a dimensão específica do domínio do serviço que está sendo medido (isto é, % de tempo disponível, número de interrupções, taxa de transferência média, etc.).

Todas as entidades acima, bem como seus relacionamentos com todos os objetivos do nível de serviço que devem ser monitorados no CA Business Service Insight, devem ser identificados durante a etapa de Modelagem de contratos.

Como funciona o processo de modelagem

Durante o processo de modelagem, todas as métricas que precisam ser configuradas no sistema devem ser definidas em conjunto com suas entidades relacionadas, com base no contrato em análise e nas exigências de geração de relatórios.

Antes ou durante essa etapa, é uma boa prática escolher um padrão de nomenclatura a ser usado para os nomes das métricas, de modo que o sistema fique limpo, organizado e fácil de navegar. Além disso, considere a descrição da métrica que pode ser usada na seção de declaração de objetivo da métrica.

O processo de análise do contrato inclui as seguintes etapas:

1. Identificar os objetivos contratuais.

Para cada objetivo, identificar:

- Um nome adequado (lembre-se dos padrões de nomenclatura)
- Destino (opcional)
- Período de monitoramento
- Unidade de medida
- Componente de serviço
- Domínio do serviço
- Categoria do domínio (e unidade de medida)
- Período de atividade (quando: 24 x 7, apenas no horário comercial?)
- Estrutura de tópicos do cálculo (que variáveis são necessárias e como o nível de serviço é calculado)

Observação: alguns deles podem não ficar claros na primeira análise, mas podem ser ordenados mais tarde, ao refinar o catálogo de serviços.

2. Identificar, no contrato, os objetivos financeiros e determinar para cada um deles:
 - É um objetivo de penalidade, incentivo ou custo?
 - Quais são as condições que o acionam?
 - Para quais componentes de serviço ocorre?
 - Domínio do serviço
 - Categoria do domínio (a unidade aqui pode ser uma moeda ou quantidade de custo, ou uma porcentagem de pagamento, etc.)

Uma vez que todos os objetivos tiverem sido apontados e definidos, é recomendável verificar a lista completa de métricas e tentar otimizar/normalizar o catálogo.

Durante essa etapa, é importante garantir que cada um dos Componentes de serviço, Domínios do serviço e Categorias do domínio será definido com lógica e que eles poderão formar um catálogo claro e conciso do que é oferecido em todo o sistema. Isso inclui TODAS as métricas e contratos no sistema. Isso garante a criação de um catálogo de serviços bem forte no sistema para possibilitar o futuro crescimento do sistema.

Domínios do serviço e Categorias do domínio NÃO devem ser definidos para um nível granular muito baixo. Eles devem ser descritivos, mas em um nível mais alto do que a métrica.

Por exemplo, se você tiver as três métricas a seguir:

- % de relatórios de interrupções entregues no prazo.
- % de relatórios de exceções entregues no prazo.
- % de relatórios de gerenciamento entregues no prazo.

A melhor categoria em que todas as três métricas provavelmente ficariam seria no domínio do serviço de Desempenho e na categoria do domínio de % de relatórios entregues no prazo. A Categoria do domínio não deve mencionar o tipo de relatório. Essas três métricas provavelmente teriam o mesmo método de cálculo e usariam a mesma lógica de negócios (com exceção de talvez um único parâmetro para distinguir entre os diferentes tipos de relatório).

Domínios do serviço e Categorias do domínio adequados podem ser obtidos no padrão ITIL (Information Technology Infrastructure Library - Biblioteca de Infraestrutura de Tecnologia da Informação). No entanto, esses são apenas exemplos sugeridos, e cada organização específica pode ter seus próprios padrões definidos para eles. Consulte o [Apêndice A - Amostras de Domínios do serviço e Categorias do domínio](#) (na página 213) para encontrar algumas sugestões de domínios do serviço e categorias do domínio.

Observação: pode ser útil neste ponto fazer uma reunião com todos os principais interessados, a fim de obter a confirmação de cada um deles de que o catálogo oferece suporte às suas necessidades atuais e futuras.

Outros exemplos que demonstram pontos importantes são apresentados nos apêndices. Nesses exemplos, um único objetivo do contrato é descrito com a sua modelagem. Ao modelar situações reais, é necessário ter conhecimento de todos os objetivos, de modo que as entidades do catálogo representem todos os objetivos de uma maneira ampla e que abrange todos os aspectos.

Após a conclusão do processo de identificação de todas as métricas e entidades relacionadas, o Gerente de contratos terá uma matriz que descreve todas as métricas, conforme mostrado no diagrama a seguir.





Outras questões que devem ser consideradas no processo de modelagem são descritas nas seções a seguir.

Perguntas para o Gerente de contratos

A seguir, estão as perguntas que o Gerente de contratos deve considerar para garantir que todos os aspectos foram levados em consideração:

Como sei que escolhi o Componente de serviço correto?

Se o componente de serviço puder ser aplicado em mais de um contrato e ser medido em mais de um aspecto, foi definido corretamente.

Por exemplo, o Sistema X é fornecido para mais de um cliente e pode ser medido por sua disponibilidade, confiabilidade, desempenho, etc.

Como sei que escolhi o Domínio do serviço correto?

Se o domínio do serviço puder ser medido ou calculado em mais de uma maneira, estará descrevendo um aspecto geral do serviço fornecido e, portanto, será o domínio do serviço correto.

Por exemplo, a disponibilidade pode ser medida de várias maneiras, sendo que uma delas é a porcentagem de tempo disponível. Outras pode ser a porcentagem de disponibilidade durante ou fora do horário comercial, o número de falhas, o MTBF (Mean Time Between Failure - Tempo Médio entre Falhas), o tempo médio de reparo, o downtime máximo, a média de downtime, o downtime total, etc. Todas essas são diferentes maneiras de avaliar a disponibilidade de um determinado sistema.

Casos a serem considerados durante o processo de modelagem

A seguir, estão vários exemplos de casos, alguns mais comuns e outros mais exclusivos, que descrevem conceitos que devem ser considerados no processo de modelagem. Esses conceitos podem resultar em uma definição mais precisa das métricas e em uma estrutura estável.

Métricas sem destino

Como o campo de destino na definição da métrica não é obrigatório, em casos em que não estiver definido, os relatórios de nível de serviço ficarão disponíveis para a métrica. No entanto, nenhum relatório de Nível de serviço versus objetivo e Desvio é possível (porque não há nenhum destino com o qual comparar o resultado calculado real de nível de serviço). Esses tipos de métrica são definidos em casos onde os relatórios são necessários para as informações que não fazem parte das obrigações contratuais reais.

Definir esse tipo de métrica fornece ao usuário todos os possíveis recursos de detalhamento de geração de relatórios, além de fornecer ao Gerenciador de nível de serviço a opção de aplicar as medições em um destino em qualquer etapa no futuro.

Por exemplo:

A garantia contratual é fornecer 99% da disponibilidade de rede e relatar o número de downtime por mês.

Duas métricas são definidas: uma com um destino de 99% para disponibilidade, e outra para contar o número de downtime por mês sem um destino. As duas métricas podem ser relatadas, mas apenas a primeira tem cálculos de desvio devido à obrigação contratual.

Observação: outro método possível para esse tipo de situação é usar saídas de lógica de negócios e geração de relatórios de formato livre desses dados. No entanto, isso faz com que a capacidade de detalhamento do relatório nos dados seja perdida, bem como a opção de usar o assistente de relatório simples. A vantagem de usar saídas de lógica de negócios, por outro lado, é a questão de economia de energia do mecanismo ao ter um número menor de métricas.

Para obter mais informações sobre esse método, consulte o estudo de caso [Saídas - Tabelas de usuário](#) (na página 157).

Métricas com destinos

Nos casos em que houver um destino definido para uma métrica, há duas maneiras possíveis de especificá-lo. Pode ser especificado como um destino estático ou dinâmico. Um destino estático é o tipo mais comum de cenário, em que o destino pode ser um valor acordado válido por toda a duração do contrato.

Por exemplo:

A disponibilidade de rede não deve ficar abaixo de 98% por mês.

O destino, neste caso, é 98%.

Como alternativa, o destino poderá depender do desempenho dos meses anteriores, ou simplesmente alterar seu valor ao longo do ano. Há muitas situações alternativas que podem ser encontradas aqui, mas, em geral, elas são implementadas por meio de uma fórmula. O CA Business Service Insight oferece suporte a esse recurso por uma chamada de função do modelo padrão de lógica de negócios. A função de destino pode acessar outros parâmetros do contexto da lógica de negócios e pode oferecer suporte a qualquer possível cenário que seja necessário.

Por exemplo, o tempo de resolução de tickets na central de ajuda que depende da carga da central de ajuda: o tempo de resolução médio para tickets de alta prioridade é 1 dia, se não houver mais de 1000 tickets durante o mês. Se houver mais de 1000 tickets emitidos na central de ajuda no mês, o tempo de resolução médio para tickets de alta prioridade será de 2 dias.

Nesse caso, a métrica é definida como tendo um destino dinâmico que é avaliado no script da lógica de negócios de acordo com o número de tickets emitidos no mês.

Observação: para obter detalhes sobre o método de implementação do destino dinâmico, consulte o estudo de caso Implementando destinos dinâmicos.

Parâmetro da métrica

Um Parâmetro da métrica é um valor que pode ser obtido na lógica de negócios da métrica e que pode ser facilmente alterado na definição da métrica sem que seja necessário alterar o código real. É usado no lugar do valor codificado e pode ser facilmente alterado.

É importante identificar os parâmetros de métrica para identificar facilmente os módulos de lógica de negócios e criar um conteúdo reutilizável. Além disso, os parâmetros de métrica são acessíveis por meio do Assistente de contrato, que permite que um usuário final execute as alterações facilmente.

Por exemplo:

- Os incidentes de gravidade 1 devem ser resolvidos dentro de 24 horas.
Na obrigação acima, o destino é uma taxa de resolução de 24 horas, e o nível de gravidade (Severity1) pode ser definido como um parâmetro.
- O número de downtime durante um mês não deve ultrapassar três, sendo que downtime é o tempo em que o sistema não está disponível por mais de 5 minutos.
Na obrigação acima, o tempo que é considerado como downtime pode ser definido como um parâmetro.

Parâmetro do contrato

Um Parâmetro do contrato é um valor que pode ser obtido por todas as métricas em um contrato. Um Parâmetro do contrato pode ser usado na métrica com o mesmo método de um Parâmetro da métrica, mas, em vez disso, é definido como um parâmetro dinâmico.

É recomendável usar um Parâmetro do contrato quando mais de uma métrica exigir o uso do mesmo valor. Outro incentivo importante para usar um Parâmetro do contrato é para facilitar a manutenção do contrato. Como os parâmetros tendem a mudar com frequência e exigir atualização no sistema, é mais fácil acessar um único local no contrato e alterar todos os valores dos parâmetros ao mesmo tempo do que acessar cada métrica no contrato e alterar os valores dos parâmetros no nível da métrica.

Portanto, a modelagem mais recomendada é definir os parâmetros no nível do contrato como parâmetros do contrato e acessar os valores por meio dos parâmetros dinâmicos no nível da métrica.

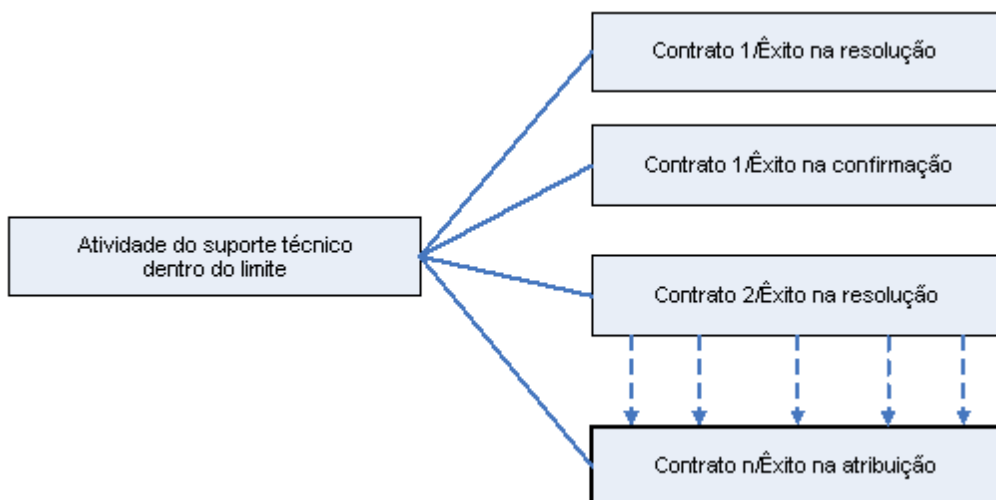
Para obter um exemplo, consulte o estudo de caso [Desempenho da central de ajuda](#) (na página 223).

Modelos e módulos de lógica de negócios

Os Modelos de lógica de negócios são uma maneira simples de armazenar um método de cálculo para uma métrica. Eles são um componente de lógica de negócios completa e uma maneira útil de criar uma linha de base para outros componentes de lógica de negócios. Componentes de lógica de negócios criados a partir de um modelo copiam o código e criam uma nova instância dele. No entanto, em geral, a flexibilidade ao usar modelos é muito baixa, e os Modelos de lógica de negócios devem ser usados sempre que possível.

Os Modelos de lógica de negócios são componentes de código independentes que permitem a reutilização da mesma base de código por outras lógicas de negócios. Módulos também podem incluir outros módulos, de modo que pode haver vários níveis de hierarquia. Ao usar os módulos, o código é armazenado em um único local e reutilizado por todos os outros componentes que estiverem vinculados a ele. Essa reutilização de seções de código facilita a manutenção, eliminando a duplicação de código e tornando possível aplicar alterações de lógica em todo o sistema de forma rápida e fácil.

Durante a fase de design, é necessário identificar os principais modelos de lógica de negócios e seus parâmetros relacionados. Após a conclusão da modelagem de contratos e quando o Gerente de contratos tiver uma visão clara da lógica a ser usada, se torna possível identificar os cálculos que eles têm em comum e que podem ser definidos em módulos separados.



O diagrama acima mostra um módulo que calcula a taxa de êxito da atividade da central de ajuda para atender a um destino dentro dos limites determinados. Para implementar como descrito, é necessário definir dois parâmetros, conhecidos como Parâmetros de métrica: um que define o tipo de atividade da central de ajuda e outro para o limite usado como comparação (consulte a definição de *Parâmetro da métrica* em [Casos a serem considerados durante o processo de modelagem](#) (na página 32)).

Ao considerar com atenção os tipos de cálculos implementados no sistema, você provavelmente descobrirá que uma série de tipos semelhantes podem ser realizados ao alterar uma pequena seção de código e usar um parâmetro para atuar como "comutador" entre eles. Dessa maneira, você pode minimizar a quantidade de código que precisa ser criado e maximizar a quantidade de código reutilizado.

Modelos de nível de serviço

Um Modelo de nível de serviço é um conjunto de pacotes de componentes de serviço e métricas associadas definidas para medi-los. Esses modelos de nível de serviço podem ser criados quando necessário e geralmente são definidos de acordo com os aspectos do serviço medidos com mais frequência.

O principal problema ao definir os modelos de nível de serviço é que todos os parâmetros que podem ser usados para alterar o comportamento da métrica devem ser identificados e expostos. Isso possibilita mais flexibilidade para o sistema e facilita a configuração para os usuários do sistema. Ao usar o modelo de nível de serviço para criar métricas, todos os parâmetros de configuração são expostos na interface do assistente, o que significa que pouca ou nenhuma personalização é necessária para ativar o contrato. Os parâmetros expostos ao usuário usando o assistente estão na declaração de objetivo. Portanto, considere na declaração de objetivo da métrica todos os parâmetros que você deseja disponibilizar para alteração por parte do usuário final.

Para obter o máximo de eficácia com modelos de nível de serviço, você deve ter a meta de executar todas as lógicas de negócios por meio da funcionalidade de módulo, conforme descrito anteriormente.

A seguir, está o exemplo de um cenário para uso dos modelos de nível de serviço, em que os componentes de serviço de hospedagem de aplicativos são fornecidos aos clientes em três níveis - Ouro, Prata e Bronze -, dependendo da quantidade que o cliente deseja pagar. Métricas adicionais podem ser fornecidas em cada nível de hospedagem mais elevado, juntamente com destinos mais rígidos. Cada um desses níveis pode ser um bom candidato para criar um modelo de nível de serviço, como mostrado no seguinte exemplo de cenário:

- Hospedagem de aplicativos - Bronze (5 métricas)
- Hospedagem de aplicativos - Prata (8 métricas)
- Hospedagem de aplicativos - Ouro (12 métricas)

Agora, quando um novo cliente é adicionado ao sistema, é fácil escolher a definição necessária com base no que o cliente deseja. Usar a interface do assistente permite que cada uma das métricas possa ser personalizada para esse cliente. Observe que também é possível selecionar apenas algumas das métricas das definições, ou ainda adicionar as métricas de duas definições diferentes ao contrato do novo cliente, se alguma personalização especial for necessária.

Importância de um catálogo de serviços forte

Conforme descrito anteriormente, o catálogo de serviços é um dos principais componentes do sistema e deve ser configurado de maneira estruturada. Isso permite que o sistema seja usado de maneira eficiente por todos os usuários e evita confusão. Além disso, permite que o sistema cresça com a organização e processe futuros aprimoramentos e adições com o mínimo de impacto sobre o que já foi criado.

O sistema oferece um alto nível de flexibilidade para criar e gerenciar o catálogo de componentes de serviço e SLAs. No entanto, uma vez que pode ser apenas tão bom quanto o design, investir tempo para planejar o futuro é essencial nas etapas iniciais de design.

Definir o catálogo de serviços do CA Business Service Insight de uma maneira estruturada e eficiente fornece ao seu sistema a flexibilidade necessária para adicionar serviços e domínios ao catálogo de serviços mais tarde. Também permite a adição de contratos, métricas, alertas e relatórios no futuro. Além disso, resulta em uma abordagem mais estruturada para a lógica de negócios e prepara o caminho para usar módulos e modelos padronizados para manipular dados de negócios e cálculos associados.

Em conjunto com o catálogo, os modelos de nível de serviço definidos no sistema oferecem uma excelente maneira para o Gerente de contratos criar contratos muito facilmente e com pouco ou nenhum conhecimento dos níveis básicos de dados necessários. Depois de ter sido configurado de maneira eficiente, deverá ser possível configurar um novo contrato para um cliente ao modificar apenas os parâmetros no modelo de nível de serviço. No entanto, isso tudo depende de o catálogo e as definições serem configurados da maneira mais eficiente. Esses parâmetros são expostos por meio da declaração de objetivo para cada métrica no modelo de nível de serviço e podem ser modificados nesse local ou no assistente ao usar a definição.

Exemplo:

Ao usar um modelo de nível de serviço para implantar algumas métricas de atendimento ao cliente em um contrato, é possível selecionar as métricas necessárias de uma definição existente.

Serviços

Localizar

Adicionar

Exibindo 1 - 15 de 15 resultados.

	Nome
	Apps Hosting Bronze
	Apps Hosting Gold
	Apps Hosting Silver
	Customer Support
	Eagle App
	Equities Trading Process

Dentro desse modelo de nível de serviço, há uma métrica chamada *% de incidentes respondidos no prazo*. Você pode ver que há um nível de subjetividade aqui, em que o componente "no prazo" pode ser questionado. O exemplo a seguir explica a medição feita nessa métrica:

Geral	Detalhes	Notas	Agrupamento	Limites	Métricas relacionadas	Granularidade	Declaração de
Nome: *	b% Inc responded on-time						
Descrição:	The percentage of the incidents responded on time. The response time is represent the time that is needed before the organisation start the resolution process of the icident.						
Tipo de métrica:	Objetivo do nível de serviço						
Declaração de objetivo:	não menos que 98 # of all Incidents will be responded on time during Sempre por 20 mês/meses , with "on-time" being defined as in less then 2 Hours for Priority 1; and less then 2 Hours for all other priorities						

A declaração de objetivo exibida na parte inferior da guia Geral da métrica (ou, como alternativa, na guia Declaração de objetivo) mostra todos os parâmetros expostos nessa métrica. Na figura anterior, a definição de "no prazo" está especificada como 20 minutos. Esse é um parâmetro que é personalizável para permitir a nossa própria interpretação dessa métrica predefinida. Para alterar esse valor, você pode clicar no link de 20 minutos do parâmetro.

Parâmetro -- Webpage Dialog

Prio1_threshold: 20

Valor mín.: | Valor máx.: | Precisão:

OK Cancelar

Dessa forma, a nova métrica criada a partir do modelo de nível de serviço pode ser personalizada sem a necessidade de modificar a lógica de negócios subjacente da métrica. Observe que isso também presume que a lógica de negócios está escrita de modo a incorporar esses parâmetros no cálculo de nível de serviço.

Esse exemplo simples mostra claramente a importância de criar um conjunto de modelos de nível de serviço eficientes e flexíveis para o catálogo do sistema, a fim de aproveitá-los para criar os futuros contratos.

Gerenciamento financeiro (Penalidades, Incentivos e Custos)

Em versões anteriores do CA Business Service Insight, existiam entidades do contrato conhecidas como Penalidades, que foram implementadas usando fórmulas parecidas com as do Excel. Os resultados das Penalidades tinham como base apenas a entrada das métricas do contrato e usavam funções básicas para calcular a quantidade de penalidade resultante. A partir da versão 4.0, elas foram substituídas por métricas financeiras completas que podem ser criadas pelo usuário, permitindo mais flexibilidade. Essas métricas financeiras podem ser usadas para fornecer informações de Incentivo ou Custo sobre o contrato.

Observação: um Incentivo substitui a antiga terminologia de Penalidade das versões 3.0 e anteriores do CA Business Service Insight e pode ser positivo ou negativo, dependendo do desempenho. Um Incentivo negativo, no entanto, é basicamente igual a uma penalidade. Também é importante observar que, se estiver implementando uma Penalidade com o tipo de métrica de Incentivo, você *deve* se lembrar de fazer com que a função Result() retorne um valor negativo. Isso permite qualquer função de resumo que possa combinar diferentes resultados de métrica para ajustar os totais na direção correta. Isto é, um incentivo aumenta o valor, ao passo que uma penalidade diminui.

A versão 4.0 do CA Business Service Insight também fornece a capacidade de criar uma métrica de consumo que mede o uso dos componentes de serviço e dos recursos, e também de combiná-la com uma métrica de item de preço para determinar o custo desse serviço ou recurso. Agora, a combinação dessas métricas com a funcionalidade aprimorada de previsão torna possível criar algumas métricas de gerenciamento financeiro muito abrangentes.

Métricas financeiras também podem obter a saída de outras métricas contratuais e determinar os valores de Penalidade ou Incentivo relacionados com base no desempenho dessas métricas contratuais. Elas também podem funcionar com outros tipos de informações para determinar o resultado, como informações de Preço por unidade e modelos de previsão que ativam a funcionalidade de geração de relatórios, como Custos esperados versus custos reais.

Exemplo de item de custo:

Um determinado aplicativo de risco tem um custo associado com base no número de usuários simultâneos do sistema. É calculado mensalmente e há um valor previsto fornecido para esse aplicativo. O preço por unidade desse aplicativo (custo por usuário simultâneo) é fornecido na tabela abaixo (suponha que esse aplicativo está no índice 1):

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

O número previsto de usuários simultâneos para esse período também está disponível (novamente, índice 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

Ao modelar essa métrica de custo usando essas tabelas de custo, é possível determinar o custo mensal do aplicativo, com base no número real de usuários simultâneos. Essas informações são provenientes da fonte de dados e podem ser multiplicadas pelos valores de preço por unidade acima para obter os valores de custo. Também podem ser comparadas aos valores previstos para fornecer uma análise do custo esperado versus custo real. A lógica de negócios nesse caso é necessária para determinar o número real de usuários simultâneos durante o período e multiplicá-lo pelos valores de custo por unidade. Além disso, a função de previsão na lógica de negócios se refere às informações de previsão. Este é um exemplo de aplicação de uma métrica de item de custo.

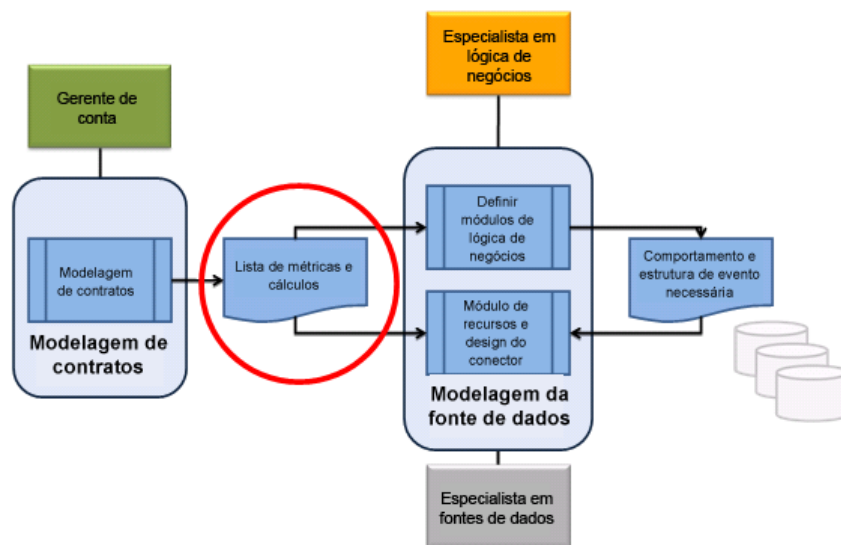
Exemplo de cenário com penalidade:

O SLA do cliente tem uma cláusula de falta de desempenho incluída para garantir que a rede estará disponível 98% do tempo em qualquer mês durante o horário comercial. Qualquer nível de serviço mensal abaixo desse número resultará no pagamento de uma penalidade com base em uma fórmula (Penalidade = US\$ 1000 por cada porcentagem completa abaixo do destino (isto é, $96,5\% = (98 - \text{Round}(96.5)) * 1000 = (98 - 97) * 1000 = - \text{US\$ } 1000$)).

Para implementar essa condição de penalidade, você pode criar uma métrica de incentivo financeiro que utiliza a entrada de uma métrica existente (*Disponibilidade de rede* $\geq 98\%$). O processo de registro para essa métrica usa o processo RegisterByMetric() para receber os valores de nível de serviço dessa métrica para executar as comparações. Envia os valores de nível de serviço do período de monitoramento dessa métrica para essa métrica financeira como eventos que depois são usados como parte do cálculo para determinar a quantidade de penalidade para esse mesmo período usando a fórmula do cenário.

Observação: para encontrar outros estudos de caso, consulte o [Estudo de caso de modelagem de métrica financeira](#) (na página 226).

Saídas da etapa de modelagem de contratos (Gerente de contratos e Especialista em fonte de dados)



Conforme mostrado no diagrama anterior, para prosseguir para a etapa de Modelagem de dados da solução, o Gerente de contratos é necessário no final da etapa de Modelagem de contratos para fornecer a lista de métricas e os cálculos necessários. A lista inclui as seguintes informações:

- Uma definição completa do catálogo de serviços, que inclui:
 - Lista completa de componentes de serviço a serem oferecidos.
 - Divisão de domínios do serviço e categorias do domínio.
 - Definição de todos os períodos de atividade necessários para cada uma das métricas.
 - Lista dos módulos de lógica de negócios para manipular cada tipo de cálculo (incluindo os parâmetros necessários para implementá-los).
- Uma lista completa de métricas a serem implementadas, que inclui:
 - Padrões de nomenclatura bem definidos para métricas.
 - Categorização de cada métrica de acordo com os componentes do catálogo de serviços predefinidos.

Este documento é uma ferramenta útil para garantir que todas as definições principais para um contrato foram feitas e que todas as suas métricas estão totalmente definidas.

As informações nesta planilha são a entrada que o Especialista em fonte de dados precisa para prosseguir para a próxima etapa, a Modelagem de dados.

Após a conclusão desses itens, é possível prosseguir para a próxima etapa, na qual você pode começar a modelar com dados reais das fontes. Sem um modelo de contrato concluído, é muito difícil saber exatamente o que é necessário na fonte de dados para cumprir as obrigações contratuais.

Modelagem de dados (Especialista em fonte de dados e Especialista em lógica de negócios)

A seção de Modelagem de dados descreve a segunda parte do processo de design. A fase de modelagem de dados é o processo de obter dados das fontes do cliente, identificar componentes específicos dos dados necessários, decidir como esses dados precisam ser normalizados e avaliar como atribuir os dados relevantes para as métricas correspondentes (por meio do processo de registro).

Essa fase inclui as seguintes tarefas:

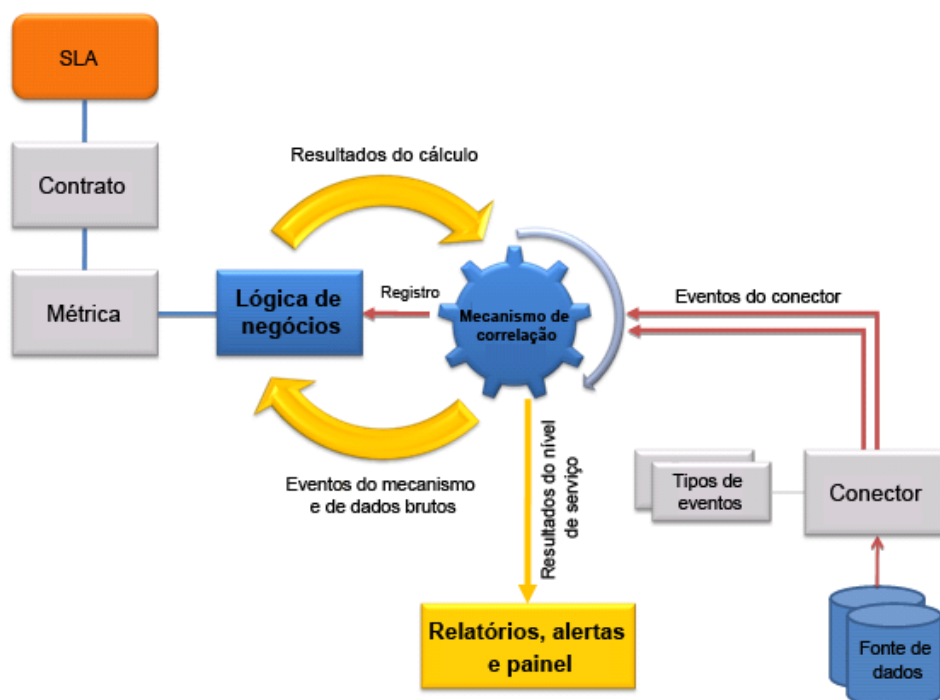
- **Especialista em lógica de negócios:**
 - Identificar e definir a estrutura de evento de entrada necessária para o cálculo a ser definido posteriormente como Eventos.
 - Criar os campos necessários para Evento para fornecer os dados necessários para todos os cálculos em que for usado.
 - Definir o processo de registro de métrica para maximizar o fluxo de eventos.
 - Determinar como criar o modelo de recurso a partir dos dados disponíveis para atender a todos os requisitos de geração de relatórios e lógica.
- **Especialista em fonte de dados:**
 - Identificar o tipo da fonte de dados e decidir como deve ser normalizado pelo conector nos eventos definidos e fornecido ao banco de dados do CA Business Service Insight.
 - Escolher os identificadores do evento que devem ser vinculados aos dados.

Eventos e seu fluxo

O fluxo de dados no sistema está na forma de eventos. Em que um evento é uma mensagem informativa criada pelo conector com base nos dados de origem e está em um formato que possa ser usado pelo CA Business Service Insight para seus cálculos de nível de serviço. Os dados brutos sempre consistem em eventos.

Portanto, o foco do design deve ser esse fluxo de eventos no sistema.

Antes de modelar os requisitos dos dados, o Especialista em lógica de negócios e o Especialista em fonte de dados devem ter uma excelente compreensão dos eventos e de seu fluxo no sistema do CA Business Service Insight. O diagrama a seguir ilustra, em alto nível, esse fluxo básico de eventos.



O diagrama anterior demonstra como eventos são recuperados da fonte de dados pelos conectores e normalizados em uma estrutura de evento padrão definida como Evento. Esses eventos são enviados pelos conectores para o CA Business Service Insight. Esses eventos são conhecidos como eventos de dados brutos.

Os cálculos da lógica de negócios em cada métrica têm como base um subconjunto dos eventos de dados brutos. Portanto, a lógica de negócios solicita esse subconjunto ao realizar um *registro*.

Com base na declaração do registro, o mecanismo de correlação envia apenas os eventos de dados brutos relevantes para os cálculos da lógica de negócios.

Outros eventos enviados para a lógica de negócios são os eventos do mecanismo. Todos os conceitos envolvidos nesse processo são abordados em detalhes neste capítulo.

Esta seção se concentra nas seguintes partes do diagrama:

- Fonte de dados
- Conector
- Evento(s)
- Registro de métrica

O modelo de dados do CA Business Service Insight foi criado para maximizar a eficácia desse fluxo de dados no sistema.

Em geral, o CA Business Service Insight funciona em duas camadas: a camada de infraestrutura e a camada de modelo de negócios. Em uma divisão simples, a camada de infraestrutura inclui os objetos Conectores, Recursos e Evento, ao passo que a camada de negócios inclui os objetos Contratos, Métricas e Serviços. Entre as duas camadas, há uma camada corretiva virtual, chamada de camada de correlação.

Um identificador de evento é o objeto Evento. O Evento determina como os eventos são definidos e como são relatados ao CA Business Service Insight. Também define a estrutura do campo de dados do evento, para que possa ser interpretado pela lógica de negócios durante o processamento.

Outro identificador de evento é o recurso, que é a menor entidade usada nos cálculos. Por exemplo, ao calcular a disponibilidade do servidor, a definição lógica da menor entidade em que a geração de relatórios é necessária pode ser um servidor específico, ou pode ser um cliente, para geração de relatórios sobre o tratamento do ticket desse cliente. O Recurso é uma definição de uma entidade do CA Business Service Insight derivada da fonte de dados e do requisito de cálculo. A cada Recurso é atribuído um Tipo de recurso, que é o identificador do recurso que determina exatamente que "tipo" de recurso está definido. Cada Recurso deve ter um Tipo de recurso associado, o que também permite a adição de atributos personalizados para serem associados a cada recurso. Para obter mais informações sobre esses atributos, consulte [Recursos e seu gerenciamento](#) (na página 52).

A correlação ocorre entre os eventos do conector e as métricas do contrato. A essência desse processo de correlação é a alocação de recursos e o registro de métricas.

A alocação de recursos e o registro de métricas especificam que fluxos de eventos do recurso são medidos e por qual métrica.

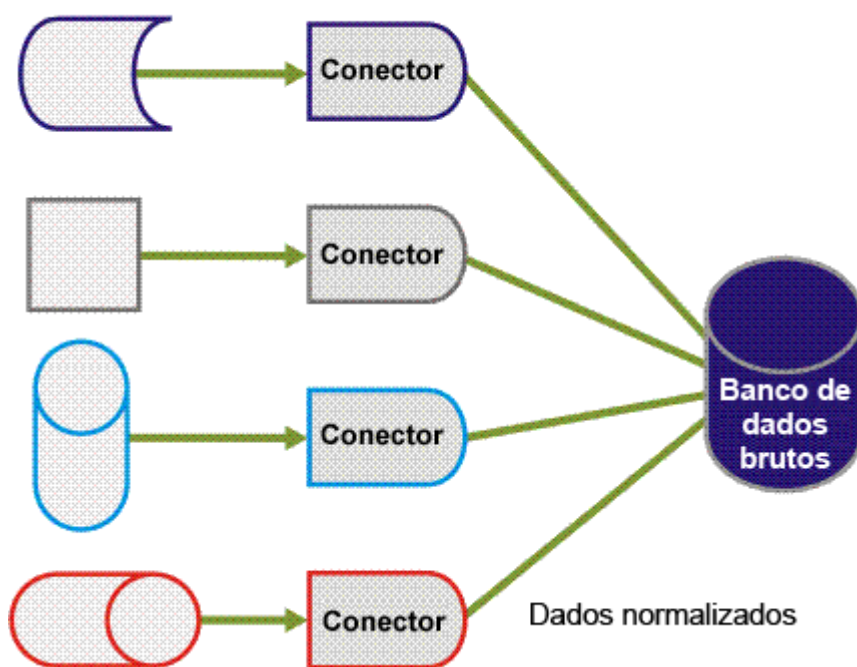
Observe que, com o registro de métrica, pode haver um grau de reutilização e codependência com outras métricas, pois é possível usar a saída de uma métrica como entrada para outra. De maneira semelhante, há eventos provisórios que não são usados como saída de uma métrica para a medição do nível de serviço, mas como uma etapa de cálculo intermediária que depois pode ser usada por outras métricas.

Modelo de dados - Visão geral

O modelo de dados do CA Business Service Insight foi criado para enfrentar e superar o desafio a seguir.

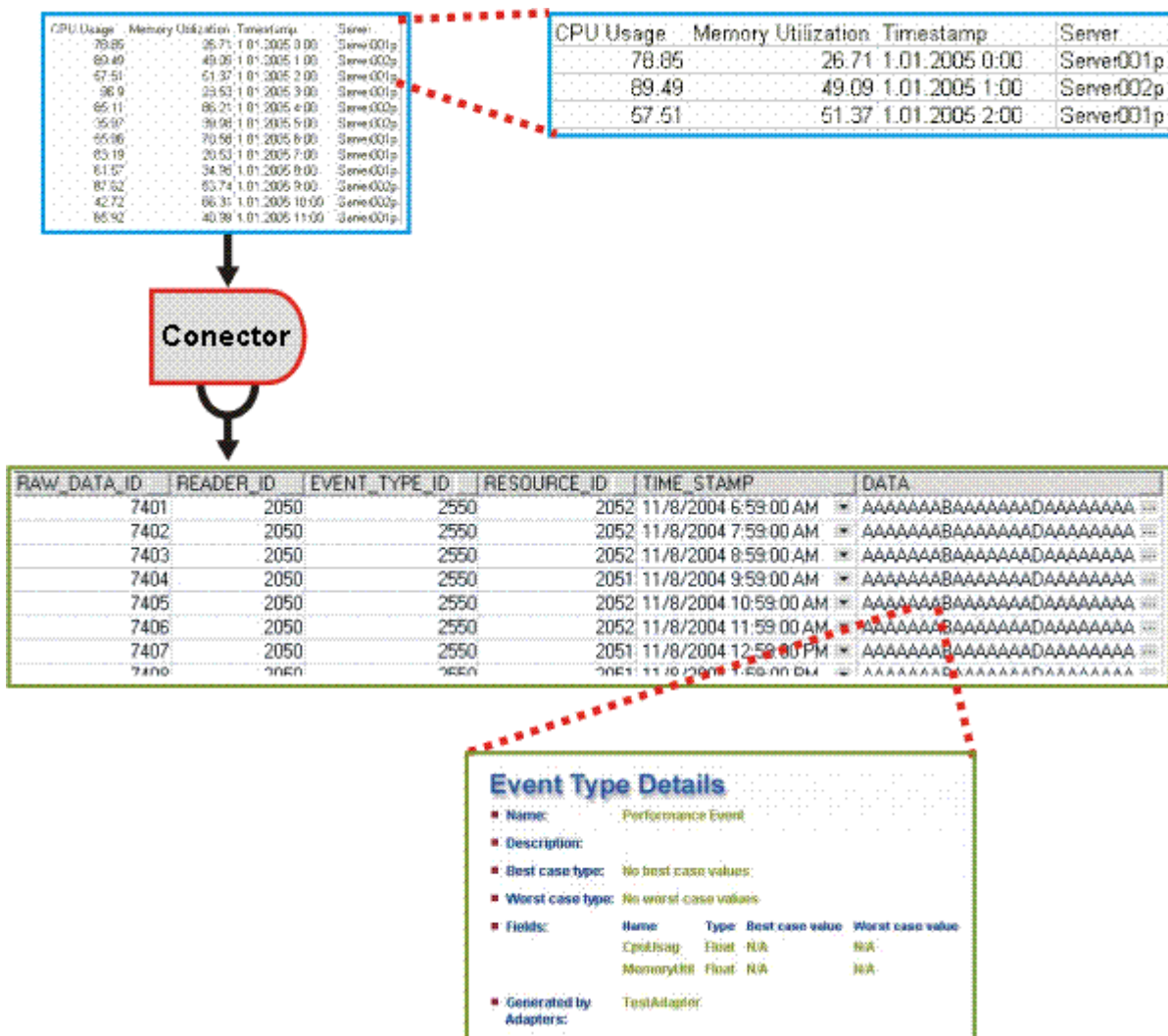
Os dados brutos são recuperados pelos conectores de várias fontes de dados distintas e mantidos em uma variedade de formatos. Esses dados diversos precisam ser recuperados e homogeneizados em uma única tabela de banco de dados. Portanto, os conectores são necessários para ler e normalizar os dados em um modelo de dados unificado, conforme mostrado na figura a seguir.

Fontes de dados



Como parte desse processo, todos os campos de dados são inseridos no mesmo campo da tabela de banco de dados, mas são criptografados. Cada linha inserida no banco de dados do CA Business Service Insight tem um identificador do evento vinculado a ela. A definição de Evento contém descrições dos campos de dados. Também permite que o mecanismo de correlação interprete corretamente os campos de dados e identifique quando eles são exigidos pela lógica de negócios para os cálculos.

A figura a seguir mostra uma representação gráfica da recuperação dos dados e da seção de preenchimento do banco de dados desse processo. Uma seção ampliada também é exibida e mostra o que os dados representam em termos reais, em vez de como os dados brutos são exibidos.



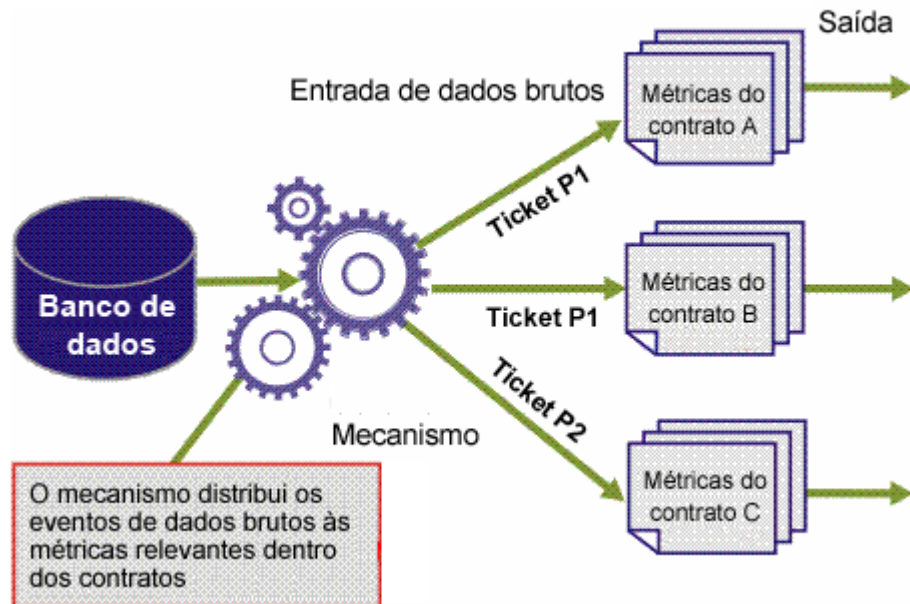
O sistema do CA Business Service Insight também inclui todos os contratos e métricas que exigem avaliação com base nos dados brutos para produzir informações resultantes de desempenho de nível de serviço. Cada métrica deve receber apenas o subconjunto de dados relevantes para o seu cálculo. Os dados brutos contêm um número potencialmente alto de registros de vários tipos. Usar a métrica para filtrar os eventos relevantes por seus valores é muito ineficiente. Portanto, o mecanismo do CA Business Service Insight distribui os dados brutos relevantes para cada métrica específica.

Exemplo:

Para as duas métricas a seguir em um contrato:

- Tempo médio de resolução de tickets com prioridade 1 (P1).
- Tempo médio de resolução de tickets com prioridade 2 (P2).

A primeira métrica é necessária para avaliar apenas os tickets com prioridade 1, e a segunda métrica, apenas os tickets com prioridade 2. Portanto, o mecanismo precisa distribuir os registros de maneira apropriada. Além disso, o tempo de resolução em um contrato é calculado para tickets P1 que foram abertos para a parte contratual A, ao passo que no segundo contrato, tickets P1 para a parte contratual B e, no terceiro, tickets P2 para a parte contratual C. Portanto, o mecanismo será solicitado a selecionar o tipo de ticket e o cliente para o qual ele foi relatado, conforme mostrado na figura a seguir.



Conforme explicado anteriormente, os registros dos dados brutos vincularam identificadores que permitem que o mecanismo identifique os registros e os eventos relevantes para cada lógica de negócios da métrica. Os dois identificadores são Evento e Recurso.

Conversão e normalização do conector

A função do conector é ler os dados da fonte de dados e normalizá-los no formato de um evento. Cada evento no CA Business Service Insight é composto dos seguintes campos:

- ID do recurso
- ID do evento
- Carimbo de data/hora
- Campo(s) de valor de acordo com o evento

O conector precisa vincular os campos originais na fonte de dados aos campos de Recurso correspondentes no CA Business Service Insight. Para fazer isso, ele usa uma tabela de conversões que contém o valor encontrado na fonte de dados e a ID do recurso correspondente no CA Business Service Insight.

O processo de vinculação da ID do recurso e da ID do evento ao valor de fonte de dados relevante é chamado de *Conversão de conector*. Durante esse processo, a tabela de conversões é criada com os valores correspondentes. Essa tabela é usada pelo conector para preencher a ID do evento relevante e as IDs do recurso no registro de evento que está criando. Tabelas independente são criadas para a conversão de Recursos e Eventos.

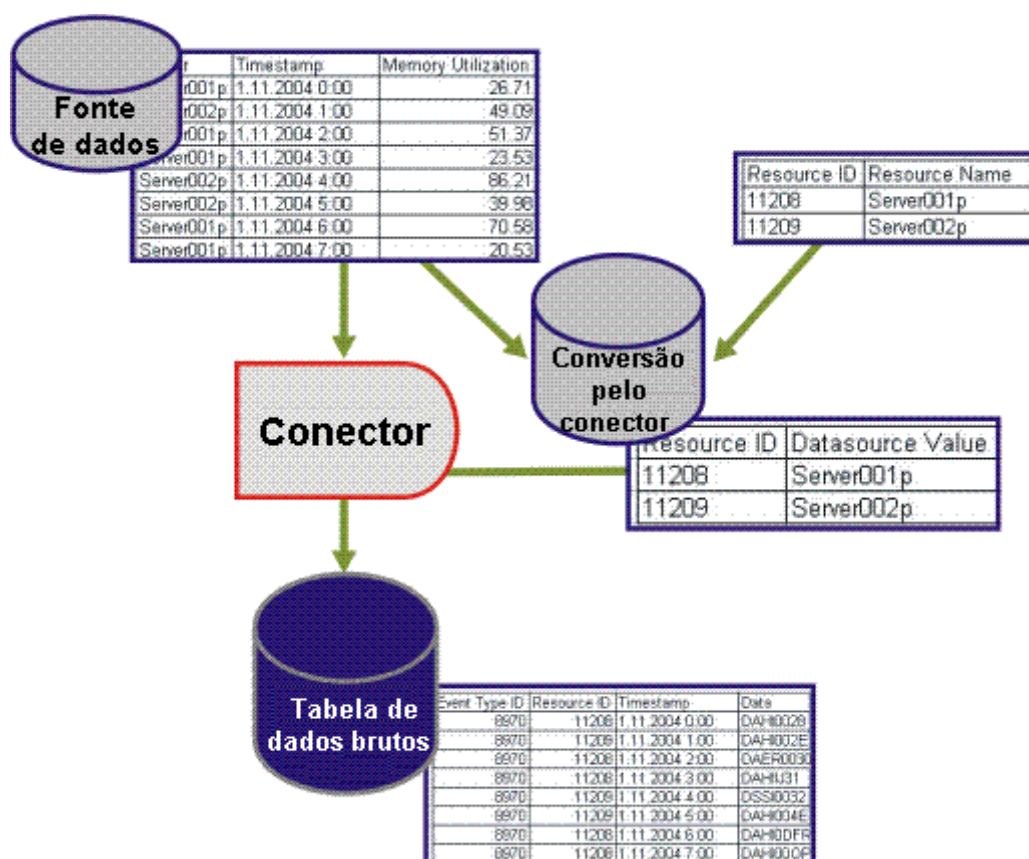
Conforme mencionado anteriormente, a ID do recurso e a ID do evento são usadas para fins de registro, ao passo que os valores do campo de dados e de data e hora são usados para cálculos reais.

O campo Data e hora também é usado pelo mecanismo para determinar a ordem e a duração dos eventos enviados para os cálculos.

A definição de Evento é feita manualmente no CA Business Service Insight com base na fonte de dados de entrada e na saída necessária.

Observação: a definição de Recurso pode ser feita manualmente ou automaticamente usando scripts de conversão (consulte [Recursos e seu gerenciamento](#) (na página 52) para obter detalhes adicionais).

A figura a seguir mostra a interação entre a fonte de dados, a tabela de conversões do conector, o conector e a tabela de dados brutos do CA Business Service Insight.



Registro de métrica

Para que o mecanismo de correlação saiba quais dados podem ser solicitados, a métrica deve registrar sua presença e requisitos com o mecanismo de correlação.

O registro de métrica é a solicitação de uma métrica para receber eventos, e apenas aqueles que precisam ser incluídos no cálculo. Essa solicitação é feita pelo estado do Evento e Recurso do identificador de evento.

O registro pode ser feito com base em um único recurso ou grupo de recursos.

Exemplo:

Para a métrica informativa a seguir: "Número de downtime do servidor X" e na premissa de que a fonte de dados fornece uma notificação quando um servidor fica ativo ou inativo, e que a notificação indica se ele está ativo ou inativo em um determinado momento, o registro é:

Evento: evento de servidor ativo/inativo

Recurso: servidor X

Com base no registro acima, o que acontece é que o mecanismo filtra todos os eventos que tiverem as definições ativo/inativo no campo Evento, além de terem servidor X no campo Recurso.

Quando um contrato está ativado no sistema do CA Business Service Insight, todas as métricas registram os eventos relevantes e necessários para os seus cálculos. Com base nessas solicitações, o mecanismo de correlação marca os eventos relevantes para cada lógica de negócios. Quando os cálculos começam, os eventos relevantes são enviados a cada métrica para cálculo.

Recursos e seu gerenciamento

Para permitir que o registro seja dinâmico, os recursos podem ser alocados individualmente, pelo seu próprio nome/identificador exclusivo, ou pela relação com os grupos lógicos.

Exemplo:

Para a métrica: "Número geral de downtime de servidores do data center", o registro é:

Evento: eventos ativos/inativos

*Recurso: todos os recursos que são identificados em servidores do data center.
(Provavelmente seria um grupo de recursos.)*

Noções básicas de ciclo de vida do recurso

Um recurso é uma entidade lógica ou física que pode alterar suas características ao longo do tempo. Pode ser alocado para determinados componentes de serviço ou partes contratuais, etc. em uma ocasião e depois ser realocado em algum momento no futuro. Cada uma dessas alterações ou realocações é capturada pelo CA Business Service Insight para poder executar cálculos a qualquer momento, com base na configuração e alocação de recursos naquele exato momento.

Alterações em um recurso ou em suas alocações podem ser realizadas a qualquer momento, mas exigem que uma nova versão desse recurso seja criada. Cada nova versão deve ter uma data de efetivação definida, que indica quando as alterações ocorrerão. As alterações serão realizadas no futuro, a menos que outras alterações sejam encontradas em uma versão posterior do mesmo recurso. Todas as alterações só ficarão visíveis e disponíveis para o mecanismo de cálculo quando essa nova versão tiver sido ativada e estiver em vigor. Esse processo é chamado de confirmação de recurso.

No CA Business Service Insight, também há uma maneira de lidar com várias alocações de recursos em uma única etapa. Esse método é realizado por meio do uso de um "Conjunto de alterações". Conjuntos de alterações permitem que grandes volumes de alterações de recursos sejam realizadas em uma única "transação", semelhante ao funcionamento de um banco de dados transacional. Todas as alterações podem ser feitas em todos os recursos que estiverem alocados para um conjunto de alterações ao executar a operação no conjunto de alterações como um todo, e, em seguida, confirmar o conjunto de alterações em uma única etapa.

Ao lidar com recursos e suas alterações, é importante considerar os seguintes pontos em relação ao mecanismo de cálculo:

- Ao ativar (confirmar) as alterações em um recurso específico ou em um grupo de recursos (conjunto de alterações), tente considerar o que será afetado no sistema. Como alterar o módulo do recurso pode acionar um recálculo, é importante otimizar a ativação do(s) recurso(s) e o número de alterações ativadas em uma operação.
- **Atualização em massa:** é possível aplicar a mesma alteração em muitos recursos (atualização em massa). Como alterar o módulo do recurso pode acionar um recálculo, é importante otimizar essa operação.

O exemplo anterior não abordou um recurso diretamente, mas por sua alocação lógica à sua função ou local (nesse caso, à sua função, um servidor do data center).

A solicitação de registro pode ser muito complexa se os eventos forem solicitados para cada servidor individual mantido no data center. Um problema é o número de recursos que são referenciados. O outro é que a infraestrutura do data center muda regularmente, de modo que um servidor que fazia parte do data center pode não estar mais no local, ou um novo servidor pode ter sido adicionado. Portanto, a lista deve ser dinâmica.

Com base no exemplo anterior, está claro que os recursos precisam ser vinculados a um grupo lógico, de modo que possam ser tratados por meio dessa entidade lógica. Além disso, o próprio grupo lógico pode precisar de gerenciamento se for alterado com frequência.

A alocação de recursos é o método que o CA Business Service Insight usa para identificar os recursos. Um recurso pode ser alocado a um ou mais grupos, tipos de recurso, partes contratuais ou serviços. As alocações de recursos são gerenciadas usando o controle da versão do CA Business Service Insight.

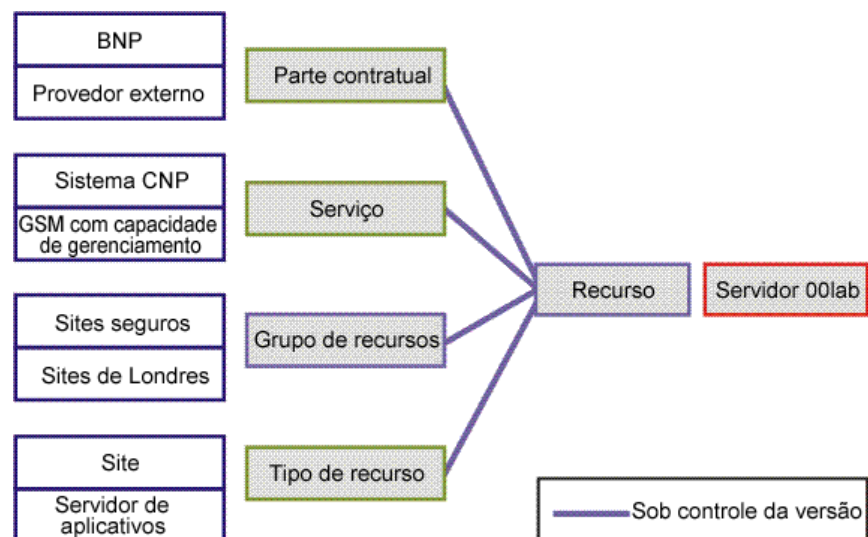
Os recursos disponíveis para inclusão em cálculos são determinadas pelos recursos atualmente em vigor no sistema (em relação ao intervalo de datas que está sendo calculado no momento).

Agora, voltando ao exemplo anterior:

Número geral de downtime de servidores do data center

O data center pode ser representado no sistema como um serviço para o qual todos os servidores no data center serão alocados. Também pode ser definido como um grupo de recursos chamado de "servidores do data center". Esses são dois métodos alternativos a escolher para a alocação de recursos nesse caso específico, mas há mais opções disponíveis.

O diagrama a seguir ilustra a quais entidades um recurso pode ser vinculado e seus usos lógicos.



Um grupo de recursos pode refletir qualquer aspecto do recurso que é necessário para os cálculos, como seu local ou a tecnologia que contém.

A principal finalidade da alocação de recursos a essas entidades é garantir que os requisitos de cálculo serão atendidos e que o modelo permanecerá o mais dinâmico possível.

Atributos personalizados

Outro recurso disponível para um recurso é a capacidade de fornecer atributos personalizados. Cada Tipo de recurso pode ter esses atributos personalizados adicionados a ele e, por sua vez, cada recurso definido como sendo desse Tipo de recurso específico, herda esses atributos personalizados.

Usando o exemplo anterior, para cada um dos servidores do data center, associe também seu endereço IP. Se cada um dos servidores do data center estiver definido com o Tipo de recurso de "Servidor do data center", verifique se um atributo personalizado chamado "IP_Address" está adicionado ao Tipo de recurso de Servidor do data center. Dessa forma, cada recurso (servidor) pode ser associado ao seu endereço IP por meio do atributo personalizado.

Observação: para obter mais detalhes e um exemplo de estudo de caso, consulte o [Estudo de caso de uso de atributos personalizados](#) (na página 244).

O que é uma métrica agrupada?

Uma métrica agrupada é uma métrica definida para calcular os níveis de serviço de um grupo de recursos. Os cálculos são feitos para cada um dos recursos individualmente, sem a necessidade de duplicar a métrica sempre com o registro de um recurso diferente.

O agrupamento pode ser feito apenas em um grupo de recursos que não tiver um nível de serviço de destino vinculado a ele ou que tiver o mesmo nível de serviço de destino. Por exemplo, a disponibilidade de todos os servidores de aplicativos deve ser de pelo menos 99,9% por servidor. Nesse caso, uma métrica é agrupada em recursos que contêm todos os servidores de aplicativos. O mecanismo calcula um resultado de nível de serviço separado para cada servidor, em que o destino para cada servidor é 99,9%.

Além desse tipo de agrupamento, o CA Business Service Insight oferece suporte a um tipo de agrupamento de "acúmulo", permitindo a geração de relatórios de vários níveis de recurso (e grupo) em uma única métrica. Isso permite que o nível de serviço seja calculado em vários níveis da hierarquia do recurso e um recurso de detalhamento até/de entre as entidades relacionadas da estrutura desse recurso. Na guia Agrupamento de métricas, você pode ativar essa opção ao selecionar as opções relevantes nessa página.

☐ Esta métrica não está agrupada.

☒ Esta métrica está agrupada como segue:

O agrupamento foi aplicado aos itens de:

☒ De modo dinâmico

☐ Primeiro nível apenas: integrantes diretos

☐ Todos os níveis: inclui recursos apenas

☒ Todos os níveis: inclui recursos e grupos de recursos

☐ Estaticamente. Apenas os seguintes itens estão incluídos:

Criando o modelo de dados do sistema

Como parte do processo de Modelagem de dados, os componentes necessários são identificados com base na fonte de dados e nos requisitos de cálculo.

A seguir, está uma lista de componentes que devem ser identificados no processo de Modelagem de dados e suas definições.

Nome do evento	Nome do evento como ele aparece no CA Business Service Insight; deve ser o mais descritivo possível.
Comportamento do evento	Comportamento do evento especificado; quando foi recebido da fonte de dados, em que condições, etc.
Campo Data e hora	Campo na fonte de dados usado como Data e hora do evento.

Campo Evento	Campo na fonte de dados a ser convertido em um evento, descrevendo o tipo de geração de relatórios. É importante que o número de eventos diferentes seja minimizado o máximo possível, pois a definição de evento é manual e, em condições ideais, deve ser feita apenas uma vez.
Campos de dados	Campos na fonte de dados a serem recuperados como campos de dados.
Campo Recurso	Campo na fonte de dados a ser convertido em um recurso. Contém uma entidade que precisa ser relatada com uma vida útil relativamente fixa. O recurso é uma entidade com um ciclo de vida definido, no qual as alterações podem ser gerenciadas dinamicamente no sistema. A referência ao ciclo de vida de um recurso indica a frequência com que novos recursos são adicionados e alterados na alocação a um serviço diferente ou qualquer outra entidade de alocação, conforme mencionado acima. O campo a ser convertido em um recurso no CA Business Service Insight deve possuir poucas alterações na alocação e outros tipos limitados.
Registro por	E por fim, com base em todas as definições acima: Define o critério de registro. O critério de registro define o evento e os recursos que a métrica registra. Solicitações de recursos podem ser executadas diretamente por meio do registro por recurso, por entidade de alocação, como serviço, parte contratual, grupo de recursos, tipo de recurso, etc. Esta definição é determinada pelos recursos de registro. Outro método é o Registro por métrica, em que a métrica atual obtém saídas (resultado de nível de serviço) de outra métrica e as usa como entrada. É possível também usar o resultado de várias métricas como entrada.

Diretrizes para definição de registros

Use as seguintes diretrizes para definir registros:

- Nunca defina o registro para ser somente por evento. Mesmo que o requisito de cálculo não precise da filtragem de recursos, adicione a filtragem de recursos pelo menos no Tipo de recurso.

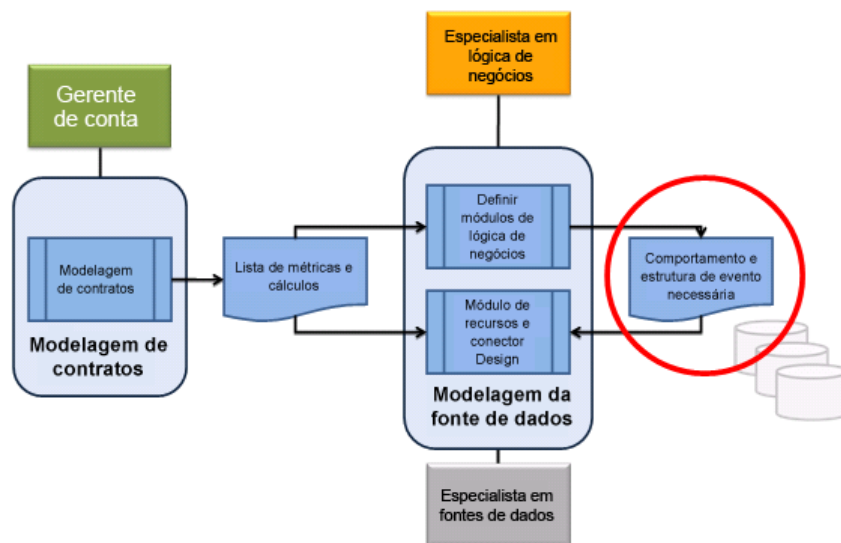
Por exemplo, ao calcular os tempos médios de resposta gerais de servidores de aplicativos, o evento de tempo de resposta deve ser relatado somente quando estiver associado a um desses servidores de aplicativos (isto é, quando o tipo de recurso do recurso associado com o evento for "Servidor de aplicativos"). Nesse caso, o sistema pode ter outros tipos de recursos, como sites ou roteadores que usam o mesmo evento para o envio de dados, de modo que seja possível distinguir entre eles. O tipo de recurso a ser relatado ("Servidor de aplicativos") deve ser adicionado ao comando de registro como o 3º parâmetro.

O motivo para isso é que quando uma alteração ocorre em um recurso, o mecanismo marca as métricas associadas a esse recurso como "exigir um recálculo". Em um caso em que a métrica registra apenas por evento, o mecanismo considera a métrica registrada para todos os recursos e, portanto, a marca para recálculo após a ativação de cada recurso. Para evitar que isso aconteça, você deve usar o 3º parâmetro opcional de tipo de recurso.

- O método mais eficiente para registro é por parte contratual e serviço. Organizar os recursos dessa maneira permite expressar o relacionamento lógico entre a camada de dados e a camada de negócios no sistema. Registrar os recursos por essas entidades não exige alterações nas fórmulas quando usadas em diferentes contratos ou para diferentes serviços. O contexto do contrato e serviço da métrica define a parte contratual e serviço relevantes. As fórmulas de lógicas de negócios definidas nesse tipo de registro são facilmente reutilizáveis, pois não exigem alterações no registro.

Observação: os registros também podem ser gerenciados pela guia *Registro* em cada métrica. Essa interface fornece um assistente que orienta você durante o processo para a métrica selecionada.

Saídas da etapa de modelagem de dados (Especialista em fonte de dados e Especialista em lógica de negócios)



Para prosseguir para a etapa de Implementação da solução, o Especialista em fonte de dados precisa ter o seguinte:

- Uma compreensão profunda das fontes de dados a serem utilizadas, incluindo:
 - Exemplo de dados históricos.
 - Método de comunicação.
 - Noções básicas dos principais campos de cada fonte de dados a serem usados para os principais componentes do modelo de recurso.
- Para cada fonte de dados, há uma lista inerente de eventos descrevendo a estrutura e o comportamento, incluindo as definições a seguir:
 - Nome do evento
 - Comportamento do evento
 - Campo Data e hora
 - Campo Evento
 - Campos de dados
 - Campo Recurso

Observação: você deve criar um documento com o resumo das informações sobre que você adquiriu.

Capítulo 3: Implementação

Esta seção contém os seguintes tópicos:

[Implementação - Introdução](#) (na página 61)

[Configurando a estrutura \(Gerente de contratos\)](#) (na página 64)

[Configurando a biblioteca de modelos \(Gerente de contratos\)](#) (na página 64)

[Como criar contratos \(Gerente de contratos\)](#) (na página 65)

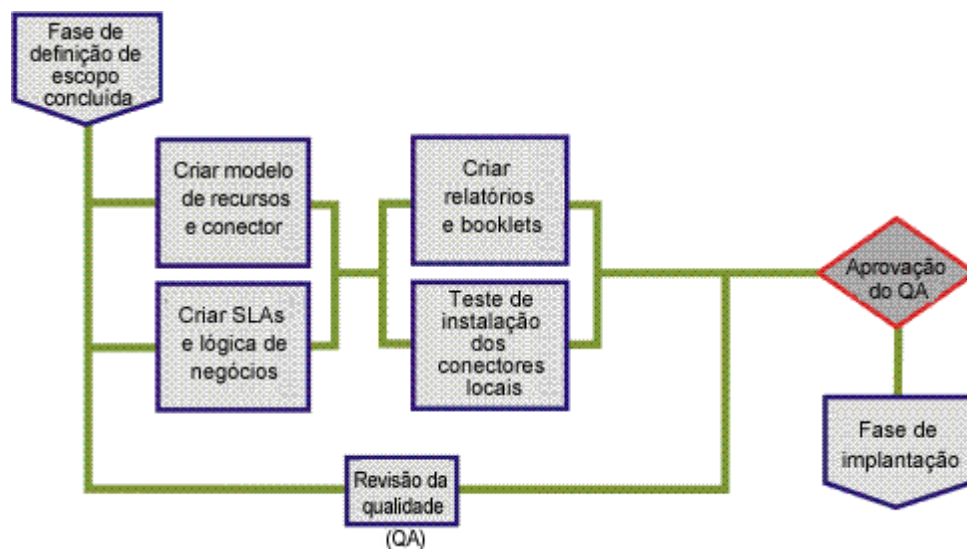
[Coleta de dados \(Especialista em fonte de dados\)](#) (na página 70)

[Criação de script de lógica de negócios \(Especialista em lógica de negócios\)](#) (na página 136)

[Como ativar os contratos \(Gerente de contratos\)](#) (na página 177)

[Criar produtos a serem entregues \(Gerente de contratos\)](#) (na página 180)

Implementação - Introdução



Este capítulo explica o processo e o raciocínio por trás da fase de implementação do projeto. Conforme mostrado na figura anterior, a fase de implementação segue a fase de design, que é seguida pela fase de instalação e implantação.

O objetivo da fase de implementação é permitir que o Gerente de contratos conclua a criação real de todos os itens e objetos no sistema do CA Business Service Insight que foram previamente definidos durante a etapa de design. Durante a fase de implementação, a equipe fica envolvida na preparação do sistema para a implantação e a instalação completas.

Essa fase não deve ser iniciada antes que a fase de design completa seja concluída e todos os objetivos necessários tenham sido organizados e definidos corretamente. Se a fase de design não tiver sido concluída corretamente ou nem todos os contratos, métricas, conectores, entre outros, tiverem sido claramente definidos, haverá problemas com o sistema ou os itens ausentes que deveriam ter sido implementados durante a fase de implementação. A fase de implementação só deve iniciar após a aprovação da verificação de design.

Também é importante que a fase de implementação seja concluída corretamente antes de continuar com a instalação e a implantação do sistema. Isso deve ser feito somente depois de executar uma verificação de qualidade.

O processo de configuração envolve o Gerente de contratos, que executa as seguintes etapas:

- Instalação do catálogo de serviços
- Criando contratos
- Ativando os contratos
- Configurando definições de segurança
- Criando relatórios/alertas/painéis

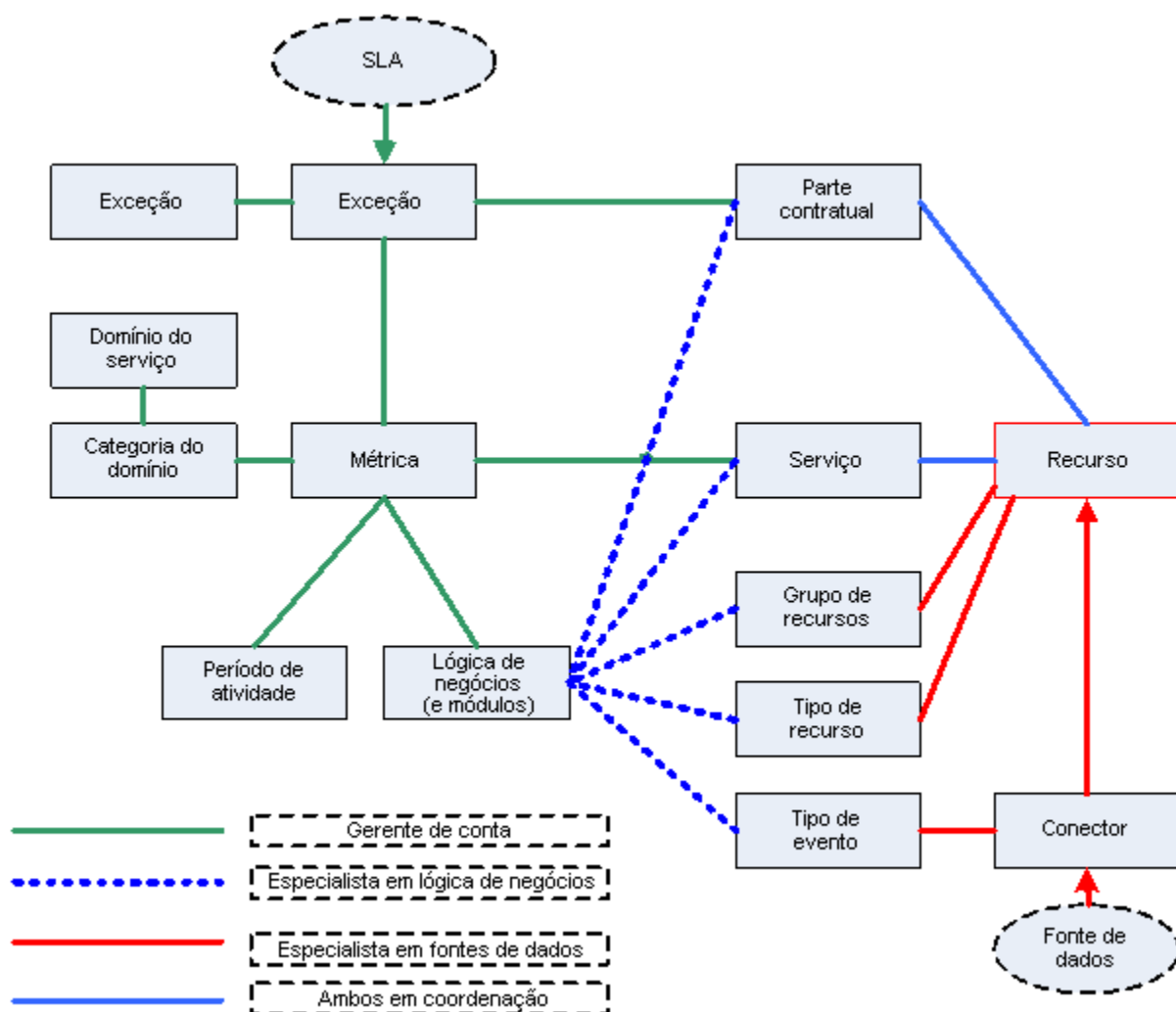
O Especialista em fonte de dados deve executar a configuração e a integração do conector com as fontes de dados. O Especialista em fonte de dados também deve executar a etapa de conversão do recurso para vincular as estruturas da fonte de dados com as entidades definidas no sistema da CA. Essa etapa é crucial para o processo inteiro e talvez exija algumas orientações do Gerente de contratos.

Além disso, o Especialista em lógica de negócios é necessário para escrever a lógica de negócios para todas as métricas, de acordo com os planos da fase de design. Isso pode incluir a criação de todos os módulos necessários e a configuração dos parâmetros associados para fornecer as funcionalidades de cálculo esperadas.

Todos os pontos acima são explicados com mais profundidade nas seções deste capítulo.

Observação: é importante que o Gerente de contratos esteja ciente de que escolhas erradas feitas nesta etapa podem causar um impacto negativo na operação do CA Business Service Insight, e pode ser difícil ou impossível revertê-lo em uma etapa posterior.

A figura a seguir descreve o fluxo de trabalho lógico geral.



Configurando a estrutura (Gerente de contratos)

A estrutura permite:

- Definir serviços, grupos de serviços, domínios do serviço e unidades.
- Criar e manter modelos, incluindo de lógica de negócios e período de atividade, e módulos de lógica de negócios.
- Gerenciar atributos personalizados para os tipos de recurso.

Nessa etapa, as entidades de todo o sistema identificadas durante a fase de design são criadas na seção Estrutura do aplicativo. Somente quando o sistema contiver essas entidades da estrutura será possível prosseguir com a criação de contratos e suas métricas relacionadas.

A criação da estrutura envolve a adição dos novos itens a seguir:

- Serviços
- Grupos de serviços
- Domínios do serviço e categorias do domínio
- Unidades de medida
- Modelos de período de atividade
- Partes contratuais
- Atributos personalizados

Para obter mais informações sobre cada um dos itens acima, consulte a Ajuda Online.

Configurando a biblioteca de modelos (Gerente de contratos)

As bibliotecas de modelos permite definir e gerenciar:

- Bibliotecas de modelos
- Pastas de modelos
- Modelos de nível de serviço
- Modelos de contrato
- Configurações de segurança para os direitos de acesso do usuário

Para obter mais informações sobre cada um dos itens acima, consulte a Ajuda Online.

Como criar contratos (Gerente de contratos)

Nessa etapa, os contratos e suas entidades relacionadas que foram definidas na fase de design são criados no sistema.

Siga estas etapas:

1. Adicionar um novo contrato e aplicar os detalhes gerais do contrato.
2. Para cada contrato, definir suas métricas e aplicar os detalhes gerais.

Somente os detalhes gerais do conteúdo de um contrato são aplicados nessa etapa, sem incluir a lógica de negócios e o agrupamento das métricas do contrato.

A seguinte descrição dessas etapas enfatiza alguns pontos importantes que devem ser considerados nessa etapa. Essas etapas são descritas em detalhes na Ajuda Online.

Etapa 1: Adicionar um novo contrato e aplique os detalhes gerais do contrato.

A definição de contrato deve incluir:

- Definir o nome do contrato.
- Selecionar as partes contratuais relacionadas.
- Vincular os serviços relacionados.
- Definir as datas de efetivação do contrato. As datas de efetivação dos contratos são o intervalo de datas para o qual o mecanismo de correlação calculará o nível de serviço para esse contrato; os resultados para a geração de relatórios estão disponíveis apenas para essas datas. Ao definir as datas, é importante levar em consideração os requisitos em termos de disponibilidade dos relatórios relacionados ao contrato e os dados brutos disponíveis.
- Definir o fuso horário e a moeda do contrato. Essa definição é para fins de geração de relatórios e permite que a geração de relatórios desse contrato siga o fuso horário relevante. A definição de moeda permite que o mecanismo de geração de relatórios determine em que moeda exibir os resultados de penalidade para as fórmulas de penalidade.

Etapa 2: Adicionar detalhes gerais das métricas.

Quando o contrato estiver em vigor, é hora de criar as métricas nele. No processo de definição das métricas, as seguintes etapas devem ser executadas:

- Definir o nome da métrica.
- Aplicar os detalhes da métrica (serviço, domínio, unidade, período de atividade, fuso horário, etc.).
- Definir os limites do painel (consulte [Configurando páginas do painel](#) (na página 194)).
- Vincular métricas relacionadas (se aplicável) e o relacionamento entre elas.

- Definir a granularidade em que a métrica é calculada pelo mecanismo.
- Definir a declaração de objetivo e os parâmetros.

Nessa etapa, a definição de métrica ainda não possui os seguintes itens:

- Fórmula/módulo de lógica de negócios e registro.
- Definição de agrupamento.

Esses itens são incluídos nas métricas do contrato apenas depois que a infraestrutura do sistema tiver sido criada e as fórmulas de lógica de negócios tiverem sido desenvolvidas e testadas.

Observação: um método alternativo para a abordagem acima é desenvolver modelos de nível de serviço no sistema, em vez de definir diretamente os contratos. Isso permite criar o modelo que depois poderá ser usado para criar outros contratos. Em alguns casos, modelos de nível de serviço preexistentes podem ser importados para o sistema de outra instância da CA. Isso pode permitir uma vantagem inicial, quando comparado com a criação de contratos a partir do zero. Para obter mais detalhes sobre a criação de seus próprios modelos de nível de serviço, consulte [Criando modelos de nível de serviço](#) (na página 68).

Em alguns casos, é recomendável criar um modelo de contrato inicialmente, para testar se tudo está funcionando corretamente e conforme o esperado. Em seguida, é possível criar os modelos de nível a partir desse contrato e armazená-los no catálogo de serviços, proporcionando um excelente ponto de partida para todos os contratos no sistema.

Criar contratos de um serviço

Criar contratos no sistema usando um catálogo de serviços, a partir de um modelo ou sem um modelo (com base em diversos modelos de nível de serviço), fornece uma eficácia muito maior e alto nível de reutilização, aplicável em muitos contratos diferentes. Em geral, os modelos de nível de serviço contêm um conjunto de métricas predefinidas aplicáveis a determinados componentes de serviço. Se necessário, você também pode ter mais de um serviço (e métricas associadas) em um único modelo de nível de serviço. Em geral, o conteúdo de um modelo de nível de serviço é definido pela maneira como será usado, e pode variar de acordo com os requisitos.

Como exemplo, considere um serviço de hospedagem de aplicativos que é oferecido por uma organização. A organização pode oferecer aos seus clientes três pacotes de serviço diferentes, como Ouro, Prata e Bronze, dependendo do que está incluído no pacote. Um bom exemplo do uso dos modelos de nível de serviço pode ser criar um modelo para cada um dos pacotes.

Uma vez criadas, essas definições podem ser usadas para criar um contrato de cliente com muita eficácia. Por exemplo, o Cliente ABC decide se inscrever em um pacote de hospedagem de aplicativos Ouro. É possível criar esse contrato no sistema usando diretamente o modelo de nível de serviço, da seguinte forma:

Na página Contratos, clique em Adicionar, usando o catálogo de serviços e selecione Com base em modelo ou Sem modelo. Em seguida, execute o Assistente de contrato para concluir a criação do contrato. Se você escolher Com base em modelo, será necessário especificar as configurações do modelo.

O Assistente de contrato exibirá uma lista de modelos de nível de serviço, e você poderá especificar quais deles deseja incluir no contrato. Observe que é possível selecionar métricas específicas de vários modelos de nível de serviço ou selecionar a definição inteira para inclusão. Nesse exemplo, todas as métricas do pacote de hospedagem Ouro foram usadas. Observe que a seleção do nível superior seleciona automaticamente todos os nós filho. Além disso, observe que é possível atribuir as métricas para um serviço diferente, se necessário. No entanto, o padrão é mantê-las atribuídas aos mesmos componentes de serviço, como na definição.

Depois que todas as métricas necessárias estiverem selecionadas, clique no botão Avançar para transferir essas métricas para o novo modelo de contrato e solicitar que um nome do contrato e detalhes gerais sejam inseridos. Clique em Salvar e Avançar para criar o contrato.

Assim que isso for concluído, você terá as seguintes opções:

- Continuar com o Assistente de contrato, definir os parâmetros e executar o Assistente de métrica, que abre a interface do assistente e permite a personalização das métricas dos contratos. O assistente permite que cada uma das métricas seja verificada e modificada alterando os campos disponíveis, como os parâmetros da métrica na declaração de objetivo, o nome da métrica, o período de atividade e a descrição. Uma vez concluído para cada métrica, o assistente o levará de volta à página Métricas do contrato, onde você poderá fechar e salvar o novo contrato.
- Abra a página Contrato para exibir ou editar o contrato.

Criar modelos de nível de serviço

A criação de um modelo de nível de serviço é um processo bastante simples. Em um contrato existente (na página Detalhes do contrato), selecione todas as métricas que deseja incluir e clique em Salvar como modelo de nível de serviço.

A próxima janela exige que você nomeie o modelo de nível de serviço. Em seguida, é possível salvar o modelo de nível de serviço. Quando estiver salvo, todos os períodos de atividade associados às métricas selecionadas serão incluídos na guia Período de atividade. A partir desse momento, talvez seja necessário personalizar o modelo de nível de serviço, a fim de torná-lo totalmente flexível. Isso pode incluir a adição de parâmetros às métricas e a exposição desses parâmetros por meio da declaração de objetivo de cada métrica. Além disso, talvez a criação de parâmetros em nível do modelo de nível de serviço (semelhantes aos parâmetros do contrato), que depois podem ser referenciados por algumas ou todas as métricas. Depois de concluído, o modelo de nível de serviço ficará disponível para implantação em outros contratos.

Ciclo de vida do contrato e controle de versão

Depois que um contrato tiver sido completamente configurado, deverá ser confirmado. Essa ação permite que o mecanismo de cálculo inicie o cálculo dos níveis de serviço para todas as métricas no contrato. Confirmar o contrato altera o status de "Preliminar" (quando é editável) para "Vigente" (quando não é editável). As outras alterações necessárias nesse contrato exigem que uma nova versão do contrato seja criada. Se as mesmas datas de efetivação forem escolhidas para a nova versão, depois que as alterações tiverem sido concluídas e a nova versão tiver sido confirmada, a versão anterior será completamente substituída. Isso também aciona o mecanismo para recalcular as métricas que são diferentes da versão anterior. Versões vigentes também podem ser parcialmente sobrepostas, por exemplo, quando uma alteração é feita nos destinos de algumas métricas do contrato na metade das datas de efetivação. Nesse caso, a versão antiga ainda é usada até que a data de efetivação da segunda versão se torne ativa. Nesse momento, a segunda versão assume o status de vigente para os cálculos.

A tabela a seguir mostra as alterações do usuário com base no escopo e intervalo de tempo do impacto do recálculo. Uma alteração pode afetar uma determinada métrica em uma versão específica do contrato ou pode afetar métricas entre contratos e versões do contrato.

Mudança	Escopo do impacto	Intervalo de tempo do impacto
Alterações feitas em métricas		
Alterando os detalhes da métrica - fórmula de lógica de negócios	Afeta todas as métricas em uma versão específica do contrato	A partir do início da versão vigente do contrato
Alterando os detalhes da métrica - valor do destino	Afeta todas as métricas em uma versão específica do contrato	A partir do início da versão vigente do contrato

Alterando os detalhes da métrica - período de monitoramento	Afeta todas as métricas em uma versão específica do contrato	A partir do início da versão vigente do contrato
Alterando os detalhes da métrica - parâmetros de métrica	Afeta todas as métricas em uma versão específica do contrato	A partir do início da versão vigente do contrato
Alterando os detalhes da métrica - fuso horário	Afeta todas as métricas em uma versão específica do contrato	A partir do início da versão vigente do contrato
Alterando os detalhes da métrica - agrupamento	Afeta todas as métricas em uma versão específica do contrato	A partir do início da versão vigente do contrato
Alterando os detalhes do contrato - datas de efetivação	Afeta todas as métricas em uma versão específica do contrato	A partir do início da versão vigente do contrato
Alterando os detalhes do contrato - períodos de atividade	Afeta todas as métricas em uma versão específica do contrato	A partir do início da versão vigente do contrato
Alterando parâmetros em nível de contrato	Afeta todas as métricas em uma versão específica do contrato	A partir do início da versão vigente do contrato
Alterando módulos de SLALOM	Afeta todas as métricas vinculadas ao módulo de SLALOM alterado, em todos os contratos e versões de contrato	A partir do início da versão vigente do contrato

Operações gerais

Alterando o modelo de recurso da métrica / Conjunto de alterações (CA Business Service Insight 4.0)	Afeta todas as métricas que registram o recurso, em todos os contratos e versões de contrato	Do estado mais próximo antes da data em que a alteração ocorreu no recurso
Obtendo eventos atrasados da métrica Eventos com data e hora passadas (dados brutos ou eventos de reutilização)	Afeta todas as métricas que usam o recurso, em todos os contratos e versões de contrato	Do estado mais próximo antes da data em que a alteração ocorreu no evento
Adicionar dados de correções da métrica	Afeta todas as métricas que usam o recurso, em todos os contratos e versões de contrato	Do estado mais próximo antes da data em que a alteração ocorreu na correção
Alterar, atualizar ou excluir período de exceções da métrica (ativação e desativação)	Dependendo das configurações e da implementação específica da exceção, afeta métricas específicas em uma versão específica do contrato ou pode afetar métricas entre contratos e versões do contrato	O mais próximo possível do período de exceção
Atualização de atributo personalizado	Afeta todas as métricas que registram o recurso, em todos os contratos e versões de contrato	Do estado mais próximo antes da data em que a alteração ocorreu no recurso

Por fim, alguns pontos principais para lembrar sobre as versões de contrato:

- Se a nova versão tiver a mesma data de efetivação, somente as métricas que tiverem sido alteradas serão recalculadas, e serão recalculadas a partir do início da versão.
- Se a nova versão tiver datas diferentes, todas as métricas da nova versão serão calculadas a partir do início da versão em questão e todas as métricas da versão anterior serão recalculadas a partir de algum ponto dessa versão até que a nova versão se torne válida. A quantidade exata de recálculo depende da configuração dos estados.
- É recomendável criar contratos com datas de efetivação de 1 ano e renová-los quando expirarem. Isso evita períodos de recálculo maiores do que um ano.
- Versões não vigentes (a data atual ultrapassa o término das datas de vigência do contrato) de contratos são calculadas (e, portanto, ainda contam como métricas ativas no sistema, pois têm dados de nível de serviço calculados associados a eles para a geração de relatórios).

Os valores associados com variáveis globais em métricas não são transmitidos entre versões de contrato (isto é, a rotina OnLoad na lógica de negócios é chamada no início de cada versão de contrato).

Observação: para obter uma série de cenários com passo a passo e estudos de caso, consulte os [Estudos de caso de Modelagem de contratos](#) (na página 215).

Coleta de dados (Especialista em fonte de dados)

Na fase de coleta de dados do processo de implementação, você trabalha com conectores. Os tópicos a seguir descrevem esse processo.

Funcionalidade do conector

Os conectores são módulos responsáveis pela coleta de dados de fontes de dados e pela transmissão deles ao sistema do CA Business Service Insight. Os conectores filtram os dados provenientes da fonte de dados e os manipulam para que contenham apenas as informações necessárias para cálculos de nível de serviço na estrutura correta quando chegarem ao sistema.

A plataforma do conector fornece a flexibilidade para:

- Receber dados online ou offline com a frequência necessária.
- Receber dados em vários níveis; brutos, calculados ou agregados.
- Receber dados de uma grande variedade de tipos de ferramenta.

Basicamente, cada conector consiste em dois componentes:

- **Componente genérico do conector:**

Há dois tipos de componentes genéricos do conector: um componente do conector de um arquivo ASCII e um componente do conector do SQL com base em ODBC. Esses componentes permitem a conexão a uma fonte de dados e a análise como um arquivo ASCII ou a execução de uma consulta SQL nela.

- **Arquivo de configuração do conector:**

Todo conector exige um arquivo de configuração para saber onde e como se conectar, o que recuperar e como transformar e converter os dados em transações e eventos genéricos da CA. O CA Business Service Insight fornece a cada tipo de conector genérico um arquivo de modelo de configuração XML que pode ser ajustado para representar informações específicas do cliente sobre a fonte de dados com a qual deve se conectar. O arquivo de configuração XML define quais campos devem ser recuperados, como devem ser identificados, como devem ser convertidos para o banco de dados normalizado do sistema, etc.

Observação: um Assistente de conector é integrado à Interface com o usuário, permitindo a personalização básica desse modelo XML online. Isso tem a mesma finalidade de criar o arquivo de configuração XML para o conector. Mais detalhes sobre esse recurso podem ser encontrados posteriormente neste capítulo.

A plataforma do conector inclui um mecanismo de reinicialização/recuperação e pode lidar com problemas nos dados recebidos de ferramentas de terceiros, como travamento de ferramenta, problemas de rede, dados ausentes, dados duplicados, dados incorretos, lacunas nos dados, validação de dados, etc. Cada conector fornece integridade total de dados e monitoramento e registro completos de todas as mensagens do conector e será abordado posteriormente neste guia com mais detalhes.

Os conectores do CA Business Service Insight podem ser executados como serviço ou aplicativo (visível ou não visível). A tecnologia de conectores do CA Business Service Insight oferece suporte a mecanismos de segurança avançados, como criptografia, handshake e processos de autenticação.

É importante observar nessa etapa que o Assistente de conector é um mecanismo de automatização dos seguintes processos e tarefas. Embora determinados elementos mencionados possam não estar sempre visíveis ao usar a abordagem orientada ao assistente, todos ainda estão presentes "nos bastidores" da interface do assistente.

Ambiente do conector

As entidades a seguir estão relacionadas ao conector e aos seus parâmetros de configuração e execução.

Fonte de dados:

A fonte de dados com a qual o conector se conecta e da qual recupera dados no formato original.

Arquivos de trabalho:

Arquivos de saída produzidos pelo conector e que são gravados nos processos (para obter mais detalhes, consulte [Arquivos de trabalho](#) (na página 75)).

Listener de conectores do CA Business Service Insight:

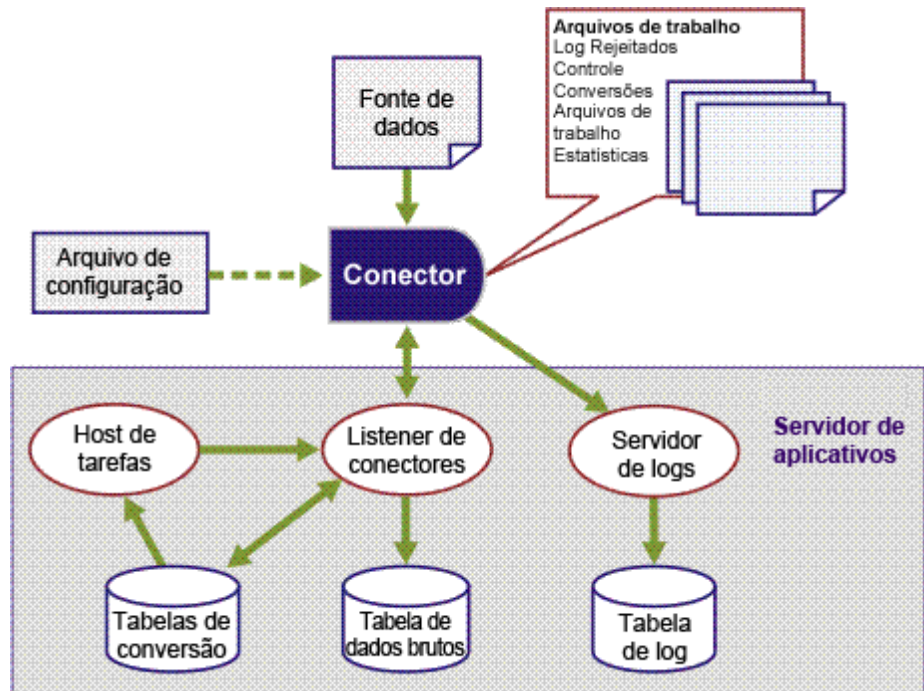
Três tipos de mensagens são transferidos entre o conector e o listener de conectores:

- **Controle:** iniciar\interromper\pausar mensagens enviadas pelo listener para o conector e de volta novamente quando o conector altera seu status.
- **Conversões:** o conector envia solicitações para o conteúdo da tabela de conversões e solicita valores de conversão específicos. O listener retorna as tabelas e o valor convertido. O listener de conectores recebe a indicação do host da tarefa de que uma entrada de conversão foi convertida. Em seguida, envia a mensagem para o conector.
- **Dados brutos:** eventos unificados de dados brutos enviados pelo conector. Esses eventos são enviados em pacotes e incluem confirmações.

Servidor de logs do CA Business Service Insight

O conector pode ser configurado para enviar mensagens de log para o log do sistema, bem como registrá-las em um arquivo local. Se a porta e o endereço IP do servidor de logs estiverem especificados e definidos nas configurações de registro do conector, ele também enviará mensagens para o servidor de logs.

O diagrama a seguir descreve o processo do conector em relação a cada uma das entidades com as quais interage.



A seguir, há uma descrição da interação do processo do conector com essas entidades:

Arquivo de configuração:

Contém configurações para alguns ou todos os parâmetros de configuração do conector. O conector usa o arquivo de configuração para determinar o método de conexão utilizado pelo conector e as métricas de análise para criar o evento de saída. Esse é um arquivo XML, e o formato contém seis elementos básicos:

Geral:

Diversos atributos do conector (pasta de trabalho, arquivos de saída, sinalizador de depuração).

OblicoreInterface:

Atributos de conexão com o servidor do CA Business Service Insight.

DataSourceInterface:

Atributos de conexão com a fonte de dados (caminho e padrão de arquivo, textos da conexão, consultas SQL, etc.)

InputFormatCollection:

Métricas de análise para analisar e manipular dados de origem.

TranslatorCollection:

Métricas para criar o evento unificado composto de campos de dados analisados e manipulados.

TranslationTableCollection:

Métricas para mapear dados entre dados originais e entidades do CA Business Service Insight.

Cada uma dessas seis seções contém todas as informações relevantes que permitem que o conector se conecte à fonte de dados, recupere as informações necessárias, analise-as nas estruturas de eventos unificados do CA Business Service Insight e armazene-as na tabela de dados brutos do CA Business Service Insight.

Arquivos principais

O conector consiste em dois arquivos principais: o executável e os arquivos de configuração. O arquivo executável é um arquivo genérico. Há dois desses arquivos executáveis: conector do SQL e conector de arquivo.

Um arquivo de configuração XML é personalizado para cada conector para armazenar os requisitos específicos da fonte de dados. O arquivo de configuração especifica as informações relacionadas à fonte de dados (nome, local, método de conexão e estrutura) e a estrutura dos eventos de saída que serão gerados pelo conector.

O arquivo de configuração inclui os parâmetros e os valores definidos para os atributos dentro de um arquivo XML estruturado e predefinido.

Ao criar um novo conector, é necessário usar o executável existente e relevante (de acordo com o tipo de fonte de dados de destino, arquivo para fontes de dados de arquivos simples, SQL para fontes de dados de banco de dados) e, em seguida, modificar o arquivo de configuração, conforme necessário. As duas estruturas contêm elementos de configuração um pouco diferentes para um conector de texto ou do SQL. Isso geralmente é configurado automaticamente ao criar o conector usando o utilitário do gerenciador de conectores.

Outros arquivos relacionados aos conectores são arquivos de trabalho criados pelo conector no processo de leitura da fonte de dados e do registro de eventos no sistema do CA Business Service Insight.

Observação: para obter detalhes sobre a modificação do arquivo de configuração, consulte [Especificações de configuração do conector](#) (na página 313).

Arquivos de trabalho

Os arquivos de trabalho do conector são criados quando o conector é executado pela primeira vez e são atualizados constantemente durante a execução de cada conector.

Cada conector tem seus próprios arquivos de trabalho. Os nomes dos arquivos de trabalho podem ser definidos no arquivo de configuração do conector (opcional) ou podem manter seus nomes padrão. O local do arquivo de trabalho é definido pela "pasta de trabalho" e também pode ser definido no arquivo de configuração. Observe que o caminho especificado pode ser relativo ao diretório atual em que o conector está localizado. O caminho especificado já deve existir (ou você deve criá-lo) para que o conector seja executado corretamente.

Observação: a pasta não será criada automaticamente se não existir.

Todos os parâmetros relevantes do arquivo de configuração estão contidos na seção Geral. Apenas o local do arquivo de log é definido no registro ou transmitido por meio da linha de comando.

AdapterStatistics.txt

O conector grava as informações estatísticas nesse arquivo em intervalos de um minuto. A última linha é gravada quando o conector é interrompido. Cada linha do arquivo contém uma série de:

- Eventos recebidos
- Eventos ignorados
- Eventos com erros
- Mensagens enviadas
- Pacotes enviados

Cada vez que o conector é executado, inicia as estatísticas.

RejectedEvents.txt

Este arquivo contém todos os eventos que o conector não pôde enviar para o CA Business Service Insight porque o valor de Evento, definido como um que exige conversão, não tem nenhuma ID correspondente na tabela de conversões. Isso significa que a conversão relevante não foi executada. Cada evento que tiver pelo menos um valor aguardando conversão é gravado no arquivo rejectedEvents.

No início de cada execução, o conector primeiro tentará enviar ao CA Business Service Insight os eventos do arquivo rejectedEvents, pois tentará localizar a ID correspondente do valor relevante na tabela de conversões. Se o valor for encontrado, o conector enviará o evento e o excluirá do arquivo. Se um valor correspondente não for encontrado, o evento permanecerá no arquivo rejectedEvents.

É possível configurar o limite superior para o número de eventos rejeitados que pode ser atingido ao definir o parâmetro RejectedEventsUpperLimit no arquivo de configuração do conector. Quando o limite é atingido, o conector interrompe a leitura de novos registros e entra no status "bloqueado". Isso pode ser visto quando a saída de depuração é exibida na tela durante a execução do conector. Se você vir um texto contínuo de letras B em caixa alta, o conector está bloqueado e exige que algumas das entradas pendentes sejam convertidas antes de carregar quaisquer outros dados.

Os eventos pendentes são gravados no arquivo no formato XML. A seguir, há um exemplo de um evento do arquivo:

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

Log do conector

O log do conector é o arquivo no qual o conector registra as mensagens de log.

É recomendável navegar pelo arquivo de log do conector usando o utilitário do navegador de logs.

É possível definir o nível de geração de relatórios para esse arquivo de log modificando um parâmetro na configuração file-LogDebugMode do conector. Quando definido como "sim", o conector registra as mensagens de indicação normal no log, bem como o registro original, os resultados de análise e o evento de destino.

Esse parâmetro é normalmente definido como "sim" durante o teste e o monitoramento do conector.

Por padrão, o tamanho do arquivo é limitado a 1 MB. Quando esse limite é atingido, o conector altera seu nome, anexa "_old" e cria um arquivo de log. O conector pode armazenar até 2 MB de mensagens de log; 1 MB para o arquivo antigo e 1 MB para o arquivo atual.

O limite de tamanho do arquivo de log pode ser configurado como uma entrada no registro para cada conector com um máximo de 10 MB. A entrada no registro é nomeada LogFileMaxSize e é definida no conector específico, com o seu valor como um múltiplo de KB.

DataSourceControl.xml

O arquivo DataSourceControl.xml é usado pelo conector para controlar o acesso à fonte de dados e garantir que sempre que for executado, continuará a partir do último ponto que foi lido.

O conector de arquivo mantém o nome do último arquivo lido, a última linha lida e a posição atingida no arquivo. Na próxima vez em que o conector for executado, acessará o arquivo do local usando as informações encontradas no arquivo DataSourceControl.xml. Usando esse mecanismo, o conector pode ler apenas os novos registros em cada execução.

O conector não trabalha diretamente nos arquivos de origem, mas primeiro copia o arquivo atual para um arquivo de trabalho. Portanto, as mesmas informações são mantidas para o arquivo de trabalho e para o arquivo de origem. Se o arquivo de origem for anexado, apenas os novos registros serão copiados para o arquivo de trabalho.

Se o conector estiver configurado para excluir o arquivo de origem assim que tiver sido processado, ao definir o parâmetro no arquivo de configuração DeleteAfterProcessing como "sim", não salvará as informações no arquivo de origem. Ao terminar, ele lerá qualquer novo arquivo existente na pasta de trabalho que corresponder ao padrão de arquivo definido no arquivo de configuração.

Apenas quando DeleteAfterProcessing estiver definido como "não", novos registros serão verificados no último arquivo. Se não houver nenhum, ele passará para o próximo arquivo em ordem lexicográfica. Portanto, ao nomear arquivos de dados de origem, a fim de garantir que eles serão lidos na sequência correta, tente aplicar um padrão de nomenclatura sequencial. Por exemplo, anexe os arquivos com um valor de data inverso (aaaammdd-hhmmss) para oferecer suporte a isso. Por exemplo:

- DataSourceABC20070517-14:00:00.csv
- DataSourceABC20070517-15:30:00.csv
- DataSourceABC20070517-17:00:00.csv, etc.

A seguir, há um exemplo do DataSourceControl.xml para um conector de arquivo.

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

</AdapterControl>

Um conector do SQL mantém o último valor dos campos chave de consulta para cada consulta executada. Os campos chave são identificadores exclusivos para registros na tabela de banco de dados de destino. O conector usa esses valores ao criar a consulta para a próxima execução. Isso permite que o conector leia apenas novos registros.

Por exemplo, considere a seguinte instrução SQL usada para buscar alguns dados de ticket com problema.

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

Nesse exemplo, para capturar todos os registros mais recentes da fonte de dados, o campo chave de consulta escolhido para garantir a obtenção das informações mais recentes é record_modified_date, pois gera tickets criados desde a última execução do conector, bem como atualizações de tickets existentes. Ao selecionar esse campo como o campo chave de consulta, o conector anexa automaticamente a seção a seguir ao final da consulta durante a execução:

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

Portanto, recupera apenas os registros mais recentes. Observe que há uma série de considerações ao escolher o campo chave de consulta, e elas sempre dependem do comportamento da fonte de dados e do que você está tentando realizar com os dados recuperados. Além disso, observe que os campos escolhidos no exemplo anterior nem sempre são a melhor opção para *todas* as situações.

Um exemplo do arquivo DataSourceControl.xml para um conector do SQL é mostrado abaixo.

```
<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

Todos os eventos que são criados e estão prontos para serem enviados ao CA Business Service Insight são gravados primeiro no arquivo send.

SendControl.xml

O arquivo sendcontrol.xml contém todas as linhas que foram enviadas e confirmadas pela CA.

O arquivo permite que o conector siga o protocolo de confirmação de janela deslizante para a transferência de dados para a CA. Informações adicionais sobre esse mecanismo podem ser encontradas em [Comunicação do conector](#) (na página 82).

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

Arquivo de saída IllegalEvents (.txt)

O conector registra no arquivo de saída IllegalEvents todos os registros que foram lidos, mas tiveram um erro de análise. Isso é normalmente causado pela lógica de validação inserida no arquivo de configuração do conector. O conector salva esses registros se o parâmetro SaveIllegalEvents no arquivo de configuração estiver definido como "sim". Observe que o caminho para essa opção também deve ser definido usando IllegalEventsDirectoryName. Essa pasta já deve existir, pois não será criada automaticamente. Se a pasta não existir, o conector apresentará um erro durante a execução.

Em um conector de arquivo, o arquivo que contém os registros de erro tem o mesmo nome que o arquivo de origem, ao passo que no conector do SQL, leva o nome da consulta.

Translations.xml

O arquivo translation.xml contém as tabelas de conversões do conector.

Quando o conector é executado no modo online, o arquivo contém uma cópia da tabela de conversões do banco de dados. Se a tabela de conversões estiver configurada como remota, o conector carregará a tabela de conversões do banco de dados para esse arquivo, substituindo-a. Se for independente, o conector lerá o arquivo local.

Quando o conector é executado no modo offline, usa o arquivo apenas como tabela de conversões (para obter informações adicionais sobre os modos online/offline, consulte [Modos de execução do conector](#) (na página 87)).

```
<TranslationTableCollection>

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

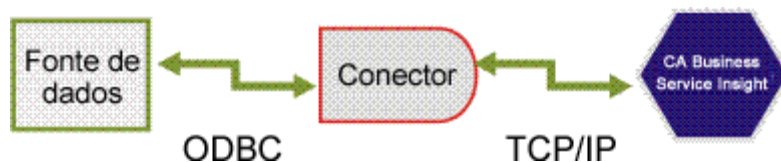
</AssystResourceTable>

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

</AssystSupplierEventTypeTable></TranslationTableCollection>
```

Comunicação do conector

Os conectores interagem com a fonte de dados de um lado e com o listener de conectores e servidor de logs do CA Business Service Insight do outro, conforme ilustrado no diagrama a seguir.



O conector se comunica com a fonte de dados para a recuperação de dados por meio de uma conexão ODBC e pode estar localizado local ou remotamente em relação à fonte de dados, contanto que o conector possa estabelecer a conexão ODBC.

O conector se comunica com o servidor de aplicativos do CA Business Service Insight usando o protocolo TCP/IP e, portanto, pode estar localizado local ou remotamente em relação a ele, contanto que possa estabelecer a conexão TCP/IP.

O conector deve ter duas portas abertas: uma para o listener de conectores e outra para o servidor de logs. As portas do listener de conectores devem ser exclusivas por conector e não devem entrar em conflito com outras operações ou aplicativos de rede que também podem usar essas portas. Por exemplo, você não deve usar a porta 1521, uma vez que, em geral, essa porta é usada pelo protocolo TNS do Oracle para comunicação com o banco de dados, etc. Talvez também seja necessário considerar os firewalls locais que podem bloquear o tráfego.

Observação: verifique com o administrador local se não tiver certeza de quais portas estão disponíveis para uso, ou se você precisará solicitar que as portas sejam abertas para permitir que essa comunicação seja realizada.

A porta e o endereço do listener de conectores são definidos no arquivo de configuração do conector. A porta e o endereço IP do servidor de logs são definidos por meio das entradas do conector no registro.

A operação de cliente/servidor em relação ao listener de conectores é configurável, o que torna possível configurar o conector para funcionar como um cliente ou um servidor. A configuração da operação de cliente/servidor é executada no lado do conector nos parâmetros do arquivo de configuração. Para fazer isso, as variáveis Port, Address e ConnectionInitiator devem ser configuradas de maneira apropriada.

Se ConnectionInitiator estiver definida para ser o conector, apenas uma porta de destino será necessária. Se estiver definida para ser o CA Business Service Insight, uma porta e um endereço IP do listener de conectores no CA Business Service Insight serão necessários. Por padrão, o servidor é definido para ser o conector. Algumas vezes, esse é um recurso importante para ativar uma regra de firewall a ser acionada (um recurso conhecido como acionamento de portas). Às vezes, um firewall permite apenas uma solicitação de chegada em uma porta, se a mensagem tiver sido enviada de "dentro" do firewall nessa mesma porta. Em seguida, ela aciona o firewall para permitir que a comunicação seja realizada.

Observação: consulte o administrador de rede para obter mais informações sobre as condições locais que podem afetar as comunicações do conector.

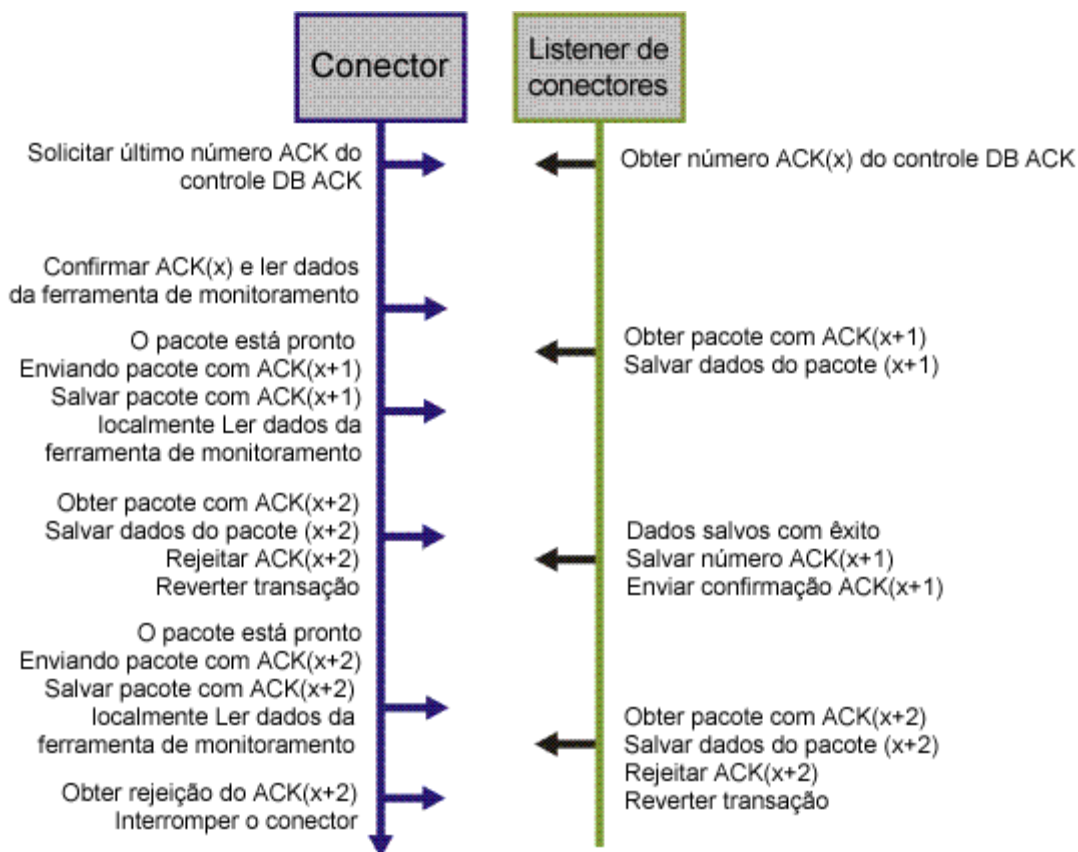
De um ponto de vista de segurança, é recomendável que o conector seja definido para ser o cliente, pois isso garante o destino de eventos ao trabalhar em um ambiente com várias implantações para testes e produção.

Para verificar o êxito da transmissão de registros de dados do conector para o listener de conectores do CA Business Service Insight, o conector incorpora um algoritmo de ACKs/janela deslizante na camada de TCP/IP. Esse algoritmo basicamente envia pacotes de dados e, em seguida, aguarda uma confirmação do listener de conectores antes de passar para o próximo pacote. Cada pacote contém várias mensagens de dados brutos. O número de mensagens em um pacote pode ser configurado por meio da configuração do parâmetro Tamanho do pacote. Cada pacote tem uma sequência que está contida na mensagem de confirmação. Todos os parâmetros relevantes que controlam o processo estão contidos na seção de interface do CA Business Service Insight do arquivo de configuração. Em geral, no entanto, não é necessário alterar esses parâmetros.

O listener de conectores registra os dados brutos no pacote em uma única transação.

Observação: a operação de ACK pode ser executada apenas nas mensagens de dados brutos enviadas ao CA Business Service Insight.

A figura a seguir mostra o processo de comunicação do conector.



Configurações de registro do conector

Nos casos em que informações estiverem faltando na linha de comando, o conector usa algumas das definições armazenadas no registro do sistema no servidor onde o conector está instalado.

As entradas do registro são gravadas pelo utilitário do gerenciador de conectores se o utilitário tiver sido usado para a instalação do conector. Se não tiver sido usado para a instalação do conector, essas entradas podem ser adicionadas manualmente ao registro.

Observação: ao instalar um conector em um ambiente UNIX, essas entradas devem ser adicionadas manualmente, uma vez que não há nenhum gerenciador de conectores para esse ambiente.

A seguir, são listadas as entradas do registro usadas pelo conector e pelo utilitário do gerenciador de conectores.

Entradas gerais do servidor

As seguintes entradas são gravadas em
`\HKEY_LOCAL_MACHINE\SOFTWARE\Oblicore\Adapters:`

Propriedades possíveis:

Nome	Tipo	Descrição
AdaptersDir	Sequência de caracteres	Diretório raiz para todos os conectores.*
FileAdapterConfTemplate	Sequência de caracteres	Caminho do modelo de configuração do conector de arquivo.* O gerenciador de conectores usa essas informações para copiar o modelo de configuração para a pasta do novo conector onde for especificado, como parte do processo de criação do conector.
GenericFileAdapter	Sequência de caracteres	Executável do conector de arquivo.* O gerenciador de conectores cria um atalho para o executável ou o copia para a pasta do novo conector onde for especificado, como parte do processo de criação do conector.
GenericSqlAdapter	Sequência de caracteres	Executável do conector do SQL.* O gerenciador de conectores cria um atalho para o executável ou o copia para a pasta do novo conector onde for especificado, como parte do processo de criação do conector.

Nome	Tipo	Descrição
LogServerAddress	Sequência de caracteres	Endereço de rede do servidor de logs. (Opcional)** Porta do servidor de logs - geralmente 4040. (Opcional)** Nos casos em que esses parâmetros estão definidos, o conector relata mensagens de log ao servidor de logs do CA Business Service Insight.
LogServerPort	Sequência de caracteres	
SqlAdapterConfTemplate	Sequência de caracteres	Caminho do modelo de configuração do conector do SQL.* O gerenciador de conectores usa essas informações para copiar o modelo de configuração para a pasta do novo conector onde for especificado, como parte do processo de criação do conector.

* Usado apenas pelo utilitário do gerenciador de conectores.

** Usado pelo conector.

Entradas individuais do conector

As seguintes entradas são gravadas em
\HKEY_LOCAL_MACHINE\SOFTWARE\Oblicore\Adapters\<Adapter Name>:

Propriedades possíveis:

Nome	Tipo	Descrição
ConfigurationFileName	Sequência de caracteres	Nome do arquivo de configuração do conector.**
Diretório	Sequência de caracteres	Diretório do conector.*
LogFileName	Sequência de caracteres	Nome do arquivo de log do conector.**
Caminho	Sequência de caracteres	Caminho do executável do conector.*
RunAs	Número	Modo de execução.* Serviço/Aplicativo de console/Aplicativo do Windows

Tipo	Número	Tipo de conector.* File/SQL
LogFileMaxSize	Número	O valor é um número em KB.** O intervalo permitido é 1.000-100.000 e o valor padrão é 1000.

* Usado apenas pelo utilitário do gerenciador de conectores.

** Usado pelo conector.

Modos de execução do conector

O conector pode ser executado por:

Serviço:

O conector pode ser instalado como um serviço regular do Windows. Isso permite que o sistema controle o seu status (Executar, Pausar, Interromper, Automático) como faria com um serviço regular do Windows.

Instalar o conector como um serviço do Windows, que é feito por meio do executável do conector na linha de comando usando **-i** para instalá-lo como serviço e **-u** para desinstalá-lo.

Aplicativo:

Usar o executável do conector na linha de comando. A linha de comando do conector pode ser executada da seguinte maneira:

Opções de linha de comando:

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

Parâmetro	Função
-i	Instalar o serviço
-u	Remover o serviço
-v	Mostrar a versão
-d	Executar o conector como aplicativo de console
-t	Verificar o arquivo de configuração e interromper
-f	Definir o nome do arquivo de configuração
-l	Definir o nome do arquivo de log
-n	Definir o nome do serviço

Parâmetro	Função
Uma	Definir o endereço do servidor de aplicativos
-p	Definir o número da porta do servidor de aplicativos (1024-49151)
-la	Definir o endereço do servidor de logs
-lp	Definir o número da porta do servidor de logs (1024-49151)

Esse tipo de execução é normalmente usado em projetos. Isso permite executar o conector por meio de um arquivo .bat e também permite usar o Agendador do Windows para controlar a duração da execução do conector. Para programar o conector usando o Agendador do Windows, é necessário configurar o modo de execução para Executar uma vez.

RunOnce: (opcional [sim/não]). Quando definido no arquivo de configuração como "não", o conector é executado continuamente. Quando definido como "sim", o conector de arquivo é executado, lê os registros e é interrompido automaticamente quando novos registros não são exibidos. Um conector de arquivo lê o arquivo inteiro e, em seguida, aguarda alguns segundos e tenta ler novos registros (dependendo das configurações de SleepTime). Se não houver novos registros, será interrompido. Um conector do SQL executa cada uma das consultas apenas uma vez. Se RepeatUntilDry estiver definido como "não", ele será interrompido imediatamente. Se RepeatUntilDry estiver definido como "sim", ele aguardará (dependendo do SleepTime). Ele tentará executar a consulta novamente (dependendo do tempo de suspensão da consulta) e, se não houver novos registros, será interrompido.

Para obter detalhes sobre os atributos SleepTime e RepeatUntilDry, consulte [Especificações de configuração do conector](#) (na página 313).

A seção de interface do CA Business Service Insight do arquivo de configuração consiste em atributos que especificam os dois modos de conexão com o CA Business Service Insight: online e offline.

No modo online, o conector se conecta ao CA Business Service Insight, recupera as tabelas de conversões e os comandos de controle do CA Business Service Insight, e envia eventos, status e solicitações de conversão de volta ao CA Business Service Insight. No modo offline, o conector trabalha com um arquivo local de tabela de conversões e registra os eventos em um arquivo de saída.

O modo offline é normalmente usado para desenvolver e testar um conector.

O uso do ConsoleDebugMode definido como "sim" permite que as mensagens de depuração sejam exibidas no console.

Para obter detalhes sobre os vários indicadores, consulte [Especificações de configuração do conector](#) (na página 313), especificamente o atributo ConsoleDebugMode.

Ferramentas de configuração e manutenção

As ferramentas necessárias como parte do processo de configuração e manutenção de conectores são principalmente utilitários independentes, como os utilitários do CA Business Service Insight, que são simples executáveis do Windows.

Observação: ao configurar conectores em um ambiente UNIX, esses utilitários não estarão disponíveis, e a configuração deverá ser executada manualmente.

Ao trabalhar em um servidor de aplicativos do CA Business Service Insight, os utilitários serão instalados como parte da instalação do aplicativo e ficarão localizados na pasta Arquivos de programas\CA\Cloud Insight\Utilitários.

Como parte da instalação, um atalho será criado no menu Iniciar. É recomendável usar esse atalho para executar os utilitários.

Em uma situação em que alguém não esteja trabalhando no servidor de aplicativos do CA Business Service Insight, esses utilitários podem ser copiados para qualquer computador Windows como arquivos padrão, ou podem ser instalados com o pacote fornecido do CA Business Service Insight e a escolha de uma instalação personalizada. No entanto, a rotina de instalação não é obrigatória, e a cópia dos arquivos .exe para um local conveniente na estação de trabalho normalmente é suficiente. Com essa opção, no entanto, talvez você perceba que alguns arquivos .dll estão ausentes e também devem ser copiados, se disponíveis, do servidor para a pasta local na estação de trabalho.

Como configurar conectores e conversões

Essa etapa inclui a execução das seguintes etapas:

1. Configurar o conector usando o Assistente de conector ou editar manualmente o arquivo de configuração XML (descrito no próximo capítulo).
2. Implantar o conector.
3. Testar o conector.
4. Executar as conversões.
5. Criar scripts de conversão para oferecer suporte a um processo de conversão automática (opcional).

Observação: a implantação do conector pode ser feita automaticamente usando o Assistente de conector quando o novo serviço de implantação do conector for executado como um serviço em segundo plano no servidor de aplicativos para realizar essa tarefa.

Implantar um novo conector (Assistente de conector)

Quando você criar um conector usando o assistente de conector, verifique se os serviços do "Listener de conectores" e de "Implantação do conector" estão em execução.

Implantar um novo conector de dados (manualmente)

Pré-requisitos para a criação de um conector

Todos os itens a seguir devem estar em execução antes de começar a criação de um conector:

- **Pasta raiz do conector:** se o CA Business Service Insight estiver instalado no servidor, essa pasta existirá na pasta raiz Arquivos de programas\CA\Cloud Insight, caso contrário, deverá ser criada.
- **Pasta do conector individual:** crie uma pasta sob a pasta raiz do conector para o conector específico.

Observação: se estiver usando o utilitário AdapterManager, ele criará automaticamente a pasta ao adicionar o novo conector.

- **Executáveis do conector:** TextFileAdapter.exe, o executável do conector de arquivo de texto para o conector de arquivo de texto; SQLAdapter.exe, o executável do conector do SQL para o conector do SQL. Podem ser encontrados no servidor de aplicativos na pasta Arquivos de programa\CA\Cloud Insight\Conectores.

Observação: durante o processo de criação do conector com o utilitário do gerenciador de conectores, sempre que possível, você deverá selecionar a opção para criar um atalho para os executáveis, em vez de criar uma cópia. Isso garante que, quando uma atualização ou service release for aplicada ao CA Business Service Insight, todos os componentes binários serão atualizados corretamente.

- **Modelos de configuração:** os modelos do arquivo de configuração do conector. Coloque os arquivos na pasta raiz do conector. Podem ser encontrados no servidor de aplicativos na pasta Arquivos de programa\CA\Cloud Insight\Conectores. Os modelos de configuração são usados para criar o arquivo de configuração para evitar que seja necessário criá-lo a partir do zero. Também é comum usar um arquivo de configuração do conector existente para essa finalidade.
- **Utilitário do gerenciador de conectores:** um executável independente. É o suficiente para fazer uma cópia do AdaptersManager.exe da pasta Utilitários em Arquivos de programas\CA\Cloud Insight\Utilitários no servidor de aplicativos. Não é necessário criar o conector usando esse utilitário, e esse utilitário pode ser usado apenas em um servidor Windows.
- **Duas portas TCP/IP abertas:** uma porta para AdapterListener no servidor de aplicativos e outra para LogServer. A porta LogServer geralmente é 4040.
- Verifique quais componentes de serviço de aplicativos do CA Business Service Insight estão em execução. Para executar o conector, os seguintes componentes de serviço devem estar em execução:
 - **AdapterListener**
 - **TaskHost**
 - **LogServer**

Siga estas etapas:

1. Executar o utilitário AdapterManager (consulte a seção Utilitário AdapterManager acima).
2. Preparar todos os pré-requisitos acima e criar um arquivo em lote com a linha de comando do executável (consulte [Modos de execução do conector](#) (na página 87)).

Modificar o arquivo de configuração do conector

Ao criar um conector, a maior parte do trabalho é editar o arquivo de configuração.

Esse trabalho envolve a configuração de atributos no arquivo XML para controlar o comportamento do conector, de modo que ele seja executado conforme necessário. O arquivo de configuração é um arquivo XML com cada seção correspondente a uma etapa em seu fluxo de trabalho interno.

As seções são:

- **Seção Geral:** diversos atributos do conector, pasta de trabalho, especificações de arquivos de saída; sinalizador de depuração, padrões, etc.

Oblicore:

- **Seção OblicoreInterface:** atributos de conexão com o servidor do CA Business Service Insight (porta TCP/IP, modo de segurança, etc.).
- **Seção DataSourceInterface:** atributos de conexão com a fonte de dados (caminho e padrão de arquivo, textos da conexão, consultas SQL, etc.).
- **Seção InputFormatCollection:** regras de análise para analisar e manipular o formato original dos dados (delimitadores, tipos de campo, ordem dos dados, expressões regulares, etc.).
- **Seção TranslatorCollection:** regras para o evento unificado do CA Business Service Insight composto de campos de dados analisados e manipulados.
- **Seção TranslationTableCollection:** regras de mapeamento de dados entre a terminologia de dados originais e as entidades de dados do CA Business Service Insight.

Essas seções são descritas em detalhes em [Especificações de configuração do conector](#) (na página 313).

Observação: a ordem dos nós XML em cada seção não é importante.

Conectores de arquivo

Os conectores de arquivo usam o componente genérico FileAdapter (executável) e o arquivo de configuração para analisar arquivos ASCII.

Fluxo de trabalho do conector de arquivo:

- Copiar e renomear o arquivo de origem para um arquivo de trabalho.
- Ler o registro lógico.
- Analisar o registro de acordo com delimitadores ou expressões regulares.
- Localizar o inputFormat correto.
- Criar o registro de evento.
- Converter o registro de evento.
- Atualizar o arquivo de controle.

Exemplo do arquivo de configuração

O exemplo a seguir usa uma fonte de dados simples do arquivo ASCII com os requisitos do evento de saída e verifica as configurações necessárias para o arquivo de configuração do conector.

O arquivo de configuração pode ser visualizado e editado com o utilitário XMLPad.

Para obter uma visão geral da estrutura e do conteúdo do arquivo de configuração, consulte as seções relevantes.

Observação: as configurações verificadas são apenas as dos atributos principais e mais importantes. Para encontrar todas as especificações de atributos, consulte [Especificações de configuração do conector](#) (na página 313).

Estudo de caso do conector de arquivo

Considere o seguinte sistema de monitoramento de servidores que gera arquivos de log com as informações de acordo com a seguinte estrutura:

Server	Timestamp	Memory Utilization	CPU Usage
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

Os arquivos são entregues na pasta do conector do CA Business Service Insight, em uma subpasta denominada "dados". Os nomes de arquivo recebem o prefixo "ServerData" e o sufixo de data e hora. Os arquivos também são CSV com a extensão .csv.

Os eventos de saída a seguir são necessários:

- **Campo de conversão de recurso:** Servidor
- **Campo de data e hora:** Data e hora
- **Campos de dados:** Uso da memória, Uso da CPU

Além disso, pressupõe-se que a fonte de dados esteja localizada na Bélgica (fuso horário CET que tem uma diferença de fuso horário de + 1 hora e inclui períodos de horário de verão com uma hora adicional durante o verão).

Seção Geral do arquivo de configuração

- **MajorVersion e MinorVersion:** devem ser definidos por padrão para a versão do aplicativo.
- **WorkingDirectoryName (opcional):** especifica o caminho padrão para os arquivos de saída do conector (controle da fonte de dados, conversão, envios). Nesse caso, é definido como "saída" e, portanto, essa pasta será criada na pasta principal do conector.

Os seguintes indicadores controlam a maneira como o conector lê e converte os registros:

- **RunOnce:** se definido como "sim", o conector será executado uma vez, lendo todos os dados disponíveis e, em seguida, será interrompido.
- **DataSourceDifferenceFromUTC:** indica a diferença de horário entre UTC (Universal Time Constant - Hora Universal Coordenada, sempre sem diferença de horário, equivalente ao GMT - Greenwich Mean Time - Hora de Greenwich) e o fuso horário da fonte de dados. O fuso horário da fonte de dados é o fuso horário com o qual os campos de hora nela são identificados. Esse atributo é necessário porque o conector normaliza todas as datas em UTC. Ter todas as datas em UTC permite que o aplicativo tenha a flexibilidade para exibir as horas de acordo com os requisitos do usuário. Os atributos a seguir fornecem ao conector os detalhes de como transformar os campos de hora da entrada para UTC:
 - **DefaultOffset:** diferença de horário do UTC quando não estiver em horário de verão. Nesse caso, é definido como "1" para CET (Central European Time - Hora da Europa Central).
 - **TimeFormat:** o formato com o qual os detalhes de horário de verão (descritos a seguir) devem ser analisados. O formato de hora europeu é "dd/mm/aaaa hh:mi:ss", ao passo que as especificações de formato do CA Business Service Insight são definidas como "%d/%m/%A %H:%M:%S".
 - **DayLightSaving:** um período de horário de verão no fuso horário da fonte de dados. Esse elemento é opcional (o que significa que, se não for selecionado, não haverá horário de verão), e pode existir mais de uma vez; uma vez para cada período de horário de verão inserido. Quando alguns elementos existem, devem ser classificados por hora, e os períodos não devem ser sobrepostos. Em geral, ao configurar conectores, esse elemento é especificado para alguns anos à frente. Nesse caso, apenas um ano é configurado como um exemplo.
- **De:** data de início do período. Especificado usando o formato de hora definido acima "25/03/2007 02:00:00".
- **Até:** a data de término do período. Especificado usando o formato de hora definido acima "28/10/2007 03:00:00".
- **Turno:** mudança de horário adicionada ao DefaultOffset dentro do período de horário de verão. Digite "1" e o horário avançará uma hora durante esse período de horário de verão especificado entre as duas datas especificadas.

Seção OblicoreInterface do arquivo de configuração

Essa seção define a conexão com o servidor do CA Business Service Insight.

- **Modo** - no modo online, o conector se conecta ao CA Business Service Insight, recupera as tabelas de conversões e o comando de controle do CA Business Service Insight, e envia eventos, status e solicitações de conversão ao CA Business Service Insight. Configure o conector nesse modo e defina o valor para online.
- **Porta** - o número da porta TCP/IP que o conector usa para se comunicar com o servidor do CA Business Service Insight (onde AdapterListener está localizado).

Se não houver problemas em fazer isso, configure o conector para usar a porta 5555 (escolhida arbitrariamente). Isso também deve ser especificado no servidor na GUI para o conector, para permitir a comunicação.

Seção DataSourceInterface do arquivo de configuração

A seção DataSourceInterface consiste em atributos que especificam a conexão e o tipo de conexão entre o conector e a fonte de dados. Há dois tipos de interface: arquivo e SQL. A diferença principal entre os dois é que, no caso dos arquivos, o conjunto de arquivos é necessário e, no caso do SQL, é o conjunto de consultas que é necessário.

A seção DataSourceInterface também define a maneira como o conector gerencia o arquivo de origem (se exclui o arquivo original quando tiver sido criado apenas para o conector ou se deixa os dados quando forem necessários para outros usos, etc.).

Para que os conectores de arquivo leiam e analisem os arquivos ASCII, a interface dos arquivos é usada conforme mostrado na figura a seguir. Escolha os seguintes valores para as configurações da seguinte maneira:

A seção Arquivos no nó DataSource Interface está relacionada à conexão com a fonte de dados. Configure os atributos a seguir.

Observação: essa seção é completamente diferente para um conector do SQL.

- **DeleteFileAfterProcessing:** define a maneira como o conector lida com o arquivo de origem e determina como o conector controla a leitura para ler apenas novos registros. Nesse caso, os arquivos de origem são deixados no servidor e esse valor é definido como "não".

Nos casos em que um arquivo é criado apenas para o conector e pode ser excluído após o processamento, deve ser definido como "sim". Em seguida, o arquivo é renomeado, processado e excluído.

Quando definido como "não", o arquivo é copiado e o processamento ocorre usando o arquivo copiado. Se novos registros forem anexados ao final desse arquivo, o conector copiará esses novos registros no arquivo de trabalho durante o próximo ciclo. Caso novos registros não sejam anexados ao arquivo, o conector pesquisará o próximo arquivo com o mesmo padrão e nome (em ordem lexicográfica) que o arquivo atual. Se o conector encontrar um arquivo como esse, começará a trabalhar nesse arquivo. O conector não reverte para o arquivo anterior, mesmo que novos registros tenham sido anexados.

Defina como "não" quando for necessário manter a integridade do arquivo de origem e quando o arquivo precisar ser anexado.

- **InputFormat:** refere-se ao nome especificado para o elemento InputFormat no próximo InputFormatCollection que lida com os dados dessa conexão com a fonte de dados. O formato de entrada é a estrutura dos campos dos dados de entrada provenientes da fonte de dados depois de serem analisados pelo conector. O método de análise é especificado no atributo de delimitadores, conforme explicado a seguir: Ao lidar com mais de uma conexão usando diferentes formatos de interface, esse campo se torna mais importante e determina qual estrutura de formato de entrada lida com cada dado das fontes de dados.
- **Path:** caminho físico dos arquivos de fonte de dados. Por exemplo: C:\Arquivos de programas\CA\Cloud Insight\Adapters\ServersAdapter\data\.

- **NamePattern:** especifica o nome do arquivo de fonte de dados. Caracteres curinga podem ser usados se mais de um arquivo usar o mesmo formato de entrada. Se o nome do arquivo for especificado sem caracteres curinga, o conector pesquisará apenas por um arquivo com esse nome. Quando caracteres curinga forem usados, o conector pesquisará todos os arquivos que correspondem ao padrão, os classificará em ordem lexicográfica e, em seguida, irá lê-los um por um. Durante a próxima execução, pesquisará novos registros no último arquivo antes de prosseguir para o próximo.

Nesse exemplo, se o caractere curinga "*" for usado, os valores de atributo serão "ServerData*.csv". (O conector lê todos os arquivos com nomes que começam com ServerData e têm a extensão .csv.)

Importante: É recomendável que data e hora sejam adicionadas ao final dos nomes de arquivos usando o formato AAAAMMDD-HHMISS para garantir que os arquivos serão classificados corretamente, lidos na ordem correta e que nenhum dos arquivos ficará ausente. A parte de hora também pode ser adicionada se vários arquivos forem produzidos a cada dia.

- **Delimiters:** método de determinação de como o arquivo será analisado. Um ou mais caracteres podem ser especificados para servir como delimitadores, de acordo com as linhas de dados a serem analisadas nos campos. Se não estiver especificado, o valor padrão é "/t".

O arquivo de fonte de dados desse exemplo é um arquivo CSV (separado por vírgula). A maneira mais simples de analisar esses arquivos é especificar a vírgula como um delimitador.

Outros métodos disponíveis para análise são:

- **RegexForParser:** usa uma expressão regular para definir a regra de análise.
- **LogicLineDefinition:** usado quando uma linha no arquivo é composta de várias linhas.
- **TitleExists** (opcional): especifica se a primeira linha não em branco no arquivo de fonte de dados é uma linha de título. O título pode ser usado pelo conector ao analisar os dados. Nesse exemplo, cada arquivo de dados contém a linha de título, portanto, "sim" deve ser especificado para esse atributo.

Seção InputFormatCollection do arquivo de configuração

Essa seção especifica a estrutura de dados recuperados da fonte de dados - como uma linha de dados deve ser recortada em campos e quais são os tipos de campos e formatos. A filtragem de dados iniciais e as manipulações de dados podem ser executadas nessa seção, usando os campos InputFormatSwitch e Composto, respectivamente.

Nessa seção, é possível definir as métricas de validação para os registros de entrada usando InputFormatValidation e ValidationCase, que determinam se o registro é válido.

O nó InputFormatCollection pode conter um ou mais nós InputFormat.

O fluxo de trabalho geral dessa seção que o conector segue é:

- A linha de dados é comparada ao InputFormat especificado na DataSourceInterface.
- Os dados são divididos em campos de acordo com a correspondência da especificação de InputFormat. A ordem de InputFormatFields deve corresponder à ordem de campos analisados da fonte de dados.
- Os campos compostos recebem valores por meio da combinação e divisão dos campos de dados especificados anteriormente. As definições de campos compostos devem vir após a definição de campo normal.
- Os dados processados são verificados em relação às condições de TranslatorSwitch para determinar qual conversor será usado para criar o evento de saída com o registro de entrada.
- Os dados processados são enviados para o conversor correspondente ou ignorados.

Trabalhe com os seguintes parâmetros:

- **InputFormatName:** nome para esse formato, referido pela seção DataSourceInterface.
- **InputFormatFields:** InputFormatFields pode conter um ou mais nós de campo, de acordo com o número de fonte de dados dos campos de entrada.
- **InputFormatField:** especifica um campo de dados da linha de dados original ou um campo composto.

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

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

- **Nome:** nome para esse campo, referido por outros elementos, como o elemento composto ou o TranslatorFields como campo de origem.
- **Tipo:** tipo de dados do campo - texto/número inteiro/decimal/data e hora.
- **Origem:** valor da origem para esse campo, sendo que o valor padrão usado é "evento". Possíveis valores:

- **evento:** o valor do campo é obtido do evento proveniente da fonte de dados. Os valores dos campos são obtidos na mesma ordem em que vêm da fonte de dados.
- **composto:** o valor do campo é criado a partir de outros campos, com base na manipulação dos valores ou constantes de outros campos. Os campos manipulados já devem ter sido definidos.
- **título:** o valor do campo é obtido dos nomes do campo de título. O campo referido já deve ter sido definido.
- **nome do arquivo:** o valor do campo é obtido do nome do arquivo de fonte de dados; apenas para conectores de arquivo de texto.
- **constante:** o valor do campo é constante e obtido de `ConstantValue`, cuja propriedade será exibida a seguir.
- **TranslatorSwitch:** determina qual conversor é usado para converter a linha de dados em um evento unificado.
 - **DefaultTranslator:** conversor a ser usado nos casos em que não há critérios correspondentes. Se o valor estiver definido como "Ignorar", nenhum conversor será usado e a linha será ignorada e não será enviada para o CA Business Service Insight.

Seção TranslatorCollection do arquivo de configuração

A seção do conjunto de conversores define como os registros da fonte de dados analisados e manipulados extraídos nas seções anteriores serão convertidos em um evento do CA Business Service Insight.

Essa seção também define como lidar com eventos duplicados e como usar o mecanismo de singularidade do evento (para obter mais detalhes, consulte [Singularidade do evento](#) (na página 129)).

Quando o modo de interface estiver definido como online, o evento do CA Business Service Insight terá uma estrutura unificada que contém os seguintes campos:

- **Timestamp**: a hora da ocorrência do evento.
- **ResourceId**: a ID do recurso associada ao evento (o recurso que foi avaliado nesse evento).
- **EventTypeId**: o tipo de evento associado ao evento e descreve o tipo de evento (tipo de medição do recurso, tipo de ação do ticket, etc.).
- **DataSourceId** (opcional): qualquer valor. Fornece outros critérios de filtragem para eventos de dados brutos.
- **Value** (vários): valores do evento (resultado da medição, número do ticket, etc.). Esse campo normalmente aparece mais de uma vez.

A estrutura do conversor corresponde à estrutura do evento no CA Business Service Insight e também à tabela de banco de dados T_RAW_DATA que armazena o evento, conforme mostrado na figura a seguir:

Structure

- Translator
 - TranslatorName
 - TranslatorFields
 - TranslatorField
 - Name
 - SourceType
 - SourceName
 - TranslatorField
 - Name
 - SourceType
 - SourceName
 - TranslatorField
 - Name
 - SourceType
 - SourceName
 - TranslatorField
 - Name
 - SourceType
 - SourceName
 - TranslatorField
 - Name
 - SourceType
 - SourceName
 - TranslatorField
 - Name
 - SourceType
 - SourceName
 - TranslatorField
 - Name
 - SourceType
 - SourceName
 - TranslatorField
 - Name
 - SourceType
 - SourceName
 - TranslatorField
 - Name
 - SourceType
 - SourceName

Values

RMTranslator

ResourceId

table

ResourceTable

EventTypeId

table

EventTypeTable

DataSourceId

field

FILENAME

Timestamp

field

RaisedDateTime

Value

field

RAISED_DATE

Columns of T_RAW_DATA

Name	Type	Nullable
RAW_DATA_ID	NUMBER	
READER_ID	NUMBER	Y
EVENT_TYPE_ID	NUMBER	
RESOURCE_ID	NUMBER	
TIME_STAMP	DATE	
READER_TIME_STAMP	DATE	Y
CREATED_BY	NUMBER	
CORRECTED_BY	NUMBER	
DATA	VARCHAR2(1000)	
CREATE_DATE	DATE	
MODIFY_DATE	DATE	
DATA_SOURCE_ID	VARCHAR2(160)	Y
HASH_DATA_KEY	VARCHAR2(116)	Y

For Help, press F1

- **Translator:** descreve como converter o conjunto de campos que recebe no evento de saída.
- **TranslatorName:** nome usado pelo InputFormat para enviar conjuntos de campos para esse conversor. Esse exemplo usa o nome "Translator1". Consulte a figura anterior para encontrar os valores que podem ser usados para esse cenário.
- **TranslatorFields:** contém uma lista de elementos TranslatorField, cada um dos quais contém os seguintes atributos:
 - **Name:** o nome do campo. Na interface online, deve ser Data e hora, ResourceId, EventTypeId, ResourceId ou Valor.
 - **SourceType:** especifica a origem do valor do campo. Pode ser um dos seguintes tipos:
 - **Field:** o valor desse campo é obtido do campo no formato de entrada. O atributo SourceName contém o nome do campo InputFormat.
 - **Table:** o valor do campo é obtido da tabela de conversões. O atributo SourceName contém o nome da tabela de conversões que faz referência a um nome que é definido na próxima seção de TranslationTableCollection. Esse tipo é usado para valores que são escolhidos para serem convertidos em recursos e eventos no evento.
 - **Lookup:** o valor desse campo é obtido da tabela de conversões. O atributo SourceName contém o nome da tabela. O valor a ser convertido do atributo LookupValue e não do formato de entrada. Normalmente, é usado quando um valor constante é necessário para fins de conversão.
 - **Constant:** o valor do campo é constante e seu valor está no atributo ConstantValue. Ao usar um valor constante, é necessário especificar o tipo de campo, usando o atributo Tipo.
 - **SourceName:** contém o nome do campo ou da tabela de conversões.
 - **LookupValue:** contém o valor de pesquisa quando SourceType="lookup".
 - **ConstantValue:** contém o valor constante quando SourceType=constant. Quando o tipo de campo for hora, o valor constante será um texto formatado de acordo com TimeFormat (definido na seção Geral do conector) ou "Agora" ou "NowUtc", onde "Agora" é a hora atual no local da fonte de dados e "NowUtc" é a hora atual em UTC.

Essa seção também contém as tabelas de mapeamento que definem o mapeamento dos valores da fonte de dados em campos de evento no CA Business Service Insight e armazenam a definição da tabela com o valor da fonte de dados referido a ser convertido.

- **LoadingMode:** o padrão para uma interface online é "remoto" e para interface offline é "independente".

O método de carregamento especificado das tabelas de conversões é o seguinte:

- **Independente:** o conector carrega as tabelas de conversões localmente. Não há conexão com o servidor do CA Business Service Insight relacionada à conversão. Alterações nas tabelas de conversões são armazenadas apenas no arquivo local.
- **Remoto:** o conector envia uma solicitação para carregar todas as tabelas do servidor do CA Business Service Insight. As alterações feitas nas tabelas de conversões também são armazenadas localmente.
- **TranslationTable:** vincula o valor do evento com a tabela de mapeamento.
 - **Nome:** nome da tabela de conversões usado e referido pelo conversor.
 - **DestinationType:** [recurso, event_type, contract_party, service, time_zone, valor]. Contém o tipo da tabela de conversões. Nesse exemplo, a coluna **Servers** no arquivo de fonte de dados é convertida em um recurso do CA Business Service Insight. Portanto, o tipo da tabela de conversões é Recurso, e contém conversões de valores para IDs do recurso no CA Business Service Insight.
 - **TranslationField:** o nome do campo a ser convertido e que é obtido dos campos do formato de entrada. Pode conter no máximo cinco campos.

Cada tabela de conversões definida no arquivo de configuração deve ter uma definição correspondente na interface com o usuário do CA Business Service Insight.

A representação XML de um arquivo de configuração de exemplo é a seguinte:

```
<?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"/>
    </InputFormatFields>
  </InputFormat>
</InputFormatCollection>
```

```
        <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>
```


Conectores do SQL

Os conectores do SQL são basicamente o uso do componente genérico do conector do SQL (conector do SQL executável) com as definições apropriadas no arquivo de configuração. O conector do SQL pode se conectar a todas as fontes de dados que oferecem suporte a ODBC e OLEDB. A conexão é estabelecida por meio de um texto da conexão. O driver apropriado deve ser instalado no servidor em que o conector está instalado.

Exemplos de fontes de dados:

- Oracle
- SQL Server
- Acesso
- Excel
- Arquivos de texto, arquivos CSV (também podem ser conectados via conector de texto, mas a conexão ODBC geralmente fornece recursos adicionais de filtragem/consulta).

Fluxo de trabalho do conector do SQL:

- Abrir a conexão.
- Substituir as variáveis locais pelos últimos valores do campo chave.
- Conclusão automática, criar as cláusulas where das consultas e concatenar no final das consultas.

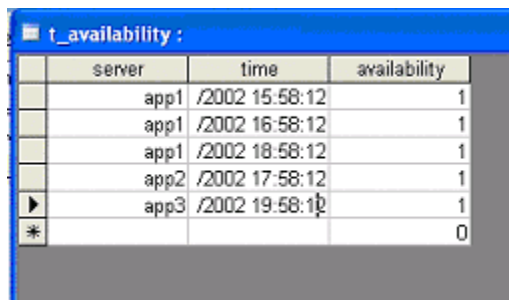
Executar a consulta e obter o conjunto de registros.

Para cada registro no conjunto de registros retornado pela consulta:

- Localizar o InputFormat correto.
- Criar o registro de evento.
- Converter o registro.
- Atualizar o valor dos campos chave no arquivo de controle.
- Fechar a conexão.
- Ir para o modo de suspensão.

Exemplo de arquivo de configuração do conector do SQL

Considere um banco de dados MS Access (.mdb) com a seguinte tabela:



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

A única diferença no arquivo de configuração em um arquivo de configuração do conector do SQL e o arquivo de configuração do conector é a seção `DatasourceInterface`.

A seção `DatasourceInterface` em um conector de arquivo armazena o conjunto de arquivos em que o arquivo do conector do SQL tem `ConnectionString` e `QueryCollection`.

A principal diferença entre os dois arquivos de configuração é o método de recuperação e análise. O restante do arquivo é igual.

A interface do SQL define a conexão com o banco de dados e as consultas usadas para recuperar os dados.

Os detalhes são:

Observação: a seção é definida com base no banco de fontes de dados acima.

Elemento de texto da conexão

ConnectionString: texto da conexão para acessar o banco de dados de origem.

ConnectionString pode ser definido no elemento DataSourceInterface e/ou nos elementos de consulta. A definição de ConnectionString no elemento DataSourceInterface é o padrão e é usada apenas em uma consulta sem uma definição de ConnectionString. Isso permite que o conector se conecte a vários bancos de dados, em que cada consulta pode ter seu próprio texto da conexão. Para obter mais detalhes sobre o mecanismo de texto da conexão, consulte a seção a seguir.

Seção do conjunto de consultas do arquivo de configuração

Query: especifica os atributos de consulta.

- **QueryName:** nome para a consulta.
- **InputFormat:** InputFormat associado a essa consulta. O conector usa o InputFormat para extrair os dados da origem.

Observação: a ordem dos campos InputFormat deve corresponder à ordem dos campos selecionados da consulta.

- **SleepTime:** tempo em segundos durante o qual o conector fica ocioso para aguardar a chegada de novos dados.
- **SelectStatement:** contém a declaração selecionada para ser executada no banco de dados de origem.

Observação: é necessário inserir os campos chave de consulta como os primeiros valores selecionados na instrução.

- **AutoCompleteQuery:** quando definido como "sim", o conector automaticamente concatena uma instrução where na consulta especificada da seguinte maneira:
 - Criar uma instrução where que recupera apenas os novos valores com base nos campos chave.
 - Ordenar os resultados da instrução com base nos campos chave.
- **QueryKeyFields:** define os campos para iniciar a próxima recuperação de dados.
 - **KeyField:**
 - **Nome:** o nome do campo, obtido dos campos da consulta.
 - **Classificação:** ordem da classificação de dados (crescente/decrescente).
 - **ValueLeftWrapper:** concatena caracteres antes do valor do campo. O padrão é ' (apóstrofe).

- **ValueRightWrapper:** concatena caracteres depois do valor do campo. O padrão é ' (apóstrofe).
- **Crítica:** interrompe a execução de outras consultas se determinada consulta falhar.
- **SleepTime:** tempo de suspensão entre as operações necessárias. O padrão é ' (apóstrofe).

Observação: campos "Data" geralmente exigem o uso de caracteres especiais para delimitar a data. Os caracteres necessários para identificar o campo como um campo de data dependem da fonte de dados. O caractere #, conforme mostrado na figura no início da seção, pode ser usado para ajustar o campo de valor no Excel. Outra fonte de dados, no entanto, pode exigir métodos diferentes. Consulte [Especificações de configuração do conector](#) (na página 313) para obter mais informações.

- **SelectInitialValues:** uma instrução select que fornece os valores iniciais para os campos chave para a primeira instrução where (quando o arquivo de controle está vazio).

Exemplo de uma consulta (ODBC com base no Excel) que é compilada com AutoCompleteQuery="yes":

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

Essa instrução select deve ser executável no banco de dados de destino em que a consulta está em execução. Isso pode diferir entre as fontes e os drivers ODBC usados para se conectar a elas. Por exemplo, no Oracle é possível selecionar os valores na tabela especial "dupla" (selecionar "aaa", "1-jan-1970" da dupla), mas no Excel, basta selecionar os valores diretamente, sem uma tabela. (selecionar "aaa")

A seguir, há um arquivo de configuração completo com layout XML:

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

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

    </ConnectionString>
```

```

<QueryCollection>
  <Query QueryName="Query" InputFormat="InputFormat" SleepTime="5">
    <SelectStatement AutoCompleteQuery="yes">

      select time,server,availability from t_availability

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

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

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

```

```
</TranslationTableCollection>  
</AdapterConfiguration>
```

Texto da conexão do conector SQL

O mecanismo de texto da conexão do conector do SQL é projetado para fornecer os seguintes recursos:

- Trabalhar com vários textos de conexão no mesmo conector.
- Trabalhar com um modelo de texto da conexão, no qual o nome do arquivo pode ser alterado sem alterar o arquivo de configuração.
- Excluir a fonte de dados do arquivo, uma vez que o conector terminar de lê-la.
- Mover a fonte de dados do arquivo para outro local, uma vez que o conector terminar de lê-la.

Além da simples definição do texto da conexão como um texto no elemento `ConnectionString`, o texto da conexão pode ser definido por segmentos, onde cada segmento contém os valores específicos que são concatenados com o texto da conexão. Isso permite que o conector crie o texto da conexão dinamicamente.

Há dois tipos de segmento. Um é texto, que contém o texto que é concatenado com o texto da conexão como está. O segundo é um arquivo com um nome de arquivo com ou sem caracteres curinga. O segmento do arquivo pode aparecer apenas uma vez. Contém outros atributos que definem o que deve ser feito com o arquivo de leitura.

`ConnectionString` pode ser definido no elemento `QueryCollection` e/ou nos elementos de consulta. A definição de `ConnectionString` no elemento `QueryCollection` é o padrão e é usada apenas em uma consulta sem uma definição explícita de `ConnectionString`.

Elementos e atributos

Elemento `ConnectionString`

Esse elemento pode conter elementos filho de segmento. Se contiver pelo menos um elemento de segmento, o texto da conexão será concatenado a partir dele. Caso contrário, o texto da conexão será obtido do texto do elemento `ConnectionString` (como na versão atual).

Elemento de segmento

Esse elemento contém um atributo obrigatório chamado *Tipo*, que tem dois valores válidos: texto e arquivo. Quando `Type="file"`, o elemento de segmento deve conter pelo menos um elemento de arquivo. Cada elemento de arquivo é considerado uma consulta diferente.

Elemento de arquivo

Esse elemento contém atributos que definem quais arquivos devem ser usados no texto da conexão e o que deve ser feito com o arquivo quando o conector terminar de lê-lo.

- **Caminho:** define o caminho do arquivo (diretório).
- **NamePattern:** define o nome de arquivo com o caminho especificado. Pode conter caracteres curinga.
- **UsePath:** valores válidos: sim/não. O padrão é "sim". Se definido como "sim", o caminho do arquivo será concatenado com o texto da conexão.
- **UseFileName:** valores válidos: sim/não. O padrão é "sim". Se definido como "sim", o nome do arquivo será concatenado com o texto da conexão (conforme necessário para arquivos do Excel).
- **UpdateSchemaFile:** valores válidos: sim/não. O padrão é "não". Se definido como "sim", o conector atualizará o arquivo `schema.ini` com o nome do arquivo atual.

Observação: use esse atributo apenas quando desejar que o conector altere o arquivo `schema.ini`.

Variáveis internas

Duas variáveis internas foram adicionadas e podem ser usadas nos elementos `SelectStatement` e `SelectInitialValues`. São elas:

- `:file_name`: substituído pelo nome do arquivo atual e uma extensão.
- `:file_name_no_ext`: substituído pelo nome do arquivo atual sem a extensão.

Exemplos

Exemplo 1: texto da conexão simples para Oracle:

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

ou

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

Exemplo 2: texto da conexão simples para Excel com um único arquivo:

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

ou

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

Exemplo 3: texto da conexão simples para uso com vários arquivos do Excel:

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

Exemplo 4: usando uma entrada DSN de ODBC padrão:

Usando uma entrada DSN de ODBC padrão, é possível conectar-se a qualquer origem que tenha uma entrada DSN criada no gerenciador de ODBC no servidor de aplicativos. A entrada DSN de ODBC padrão pode ser encontrada na seção Ferramentas administrativas do servidor.

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


Lendo registros de uma fonte de dados do arquivo

Desde que haja uma interface de ODBC para a fonte de dados, é possível configurar um conector do SQL para arquivos de consulta. Para configurar o conector para a leitura de vários arquivos, é necessário usar os elementos de segmento no elemento `ConnectionString`. Para encontrar um exemplo, consulte a seção anterior que descreve o texto da conexão.

O conector do SQL trabalha com os arquivos da seguinte maneira:

1. Em cada consulta, o conector tenta ler o arquivo até que não seja mais possível recuperar outros registros.
2. O conector, em seguida, passa para o próximo arquivo e tenta lê-lo.
3. Quando não houver mais arquivos, o conector entrará no modo de suspensão para essa consulta de acordo com o atributo `SleepTime`.

Arquivo `Schema.ini`

Quando um driver ODBC de texto for usado, o formato do arquivo de texto será determinado pelo arquivo de informações do esquema (`schema.ini`). Esse arquivo de informações do esquema deve ser colocado no mesmo diretório que a fonte de dados de texto.

Arquivos `Schema.ini` são compostos de entradas, com cada entrada descrevendo a estrutura de um único arquivo de texto. A primeira linha em cada entrada é o nome do arquivo de origem de texto entre colchetes.

Quando o conector trabalha com arquivos que foram definidos usando caracteres curinga, o nome do arquivo muda constantemente. Como o nome no arquivo `schema.ini` não pode conter caracteres curinga, o conector deve alterar o arquivo `schema.ini` antes de se conectar ao banco de dados.

Por isso, é necessário adicionar uma linha indicadora antes da linha de entrada do arquivo. Essa linha indicadora contém o padrão de nome do elemento de texto da conexão no arquivo de configuração do conector, entre colchetes. O conector substitui a próxima linha pelo novo nome do arquivo entre colchetes.

Observação: o arquivo `schema.ini` pode conter várias entradas. É sua responsabilidade adicionar as linhas no local correto.

Exemplo de `Schema.ini`

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

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

Arquivo DataSourceControl

Para cada consulta, o arquivo de controle da fonte de dados contém o padrão de nome do arquivo e o nome do arquivo de entrada atual para continuar com o mesmo arquivo quando o conector for reiniciado.

A seguir, há um exemplo do arquivo de controle no formato XML:

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

Retendo arquivos de entrada

Quando o conector termina de ler o arquivo atual, pesquisa pelo próximo. O próximo arquivo lido é o primeiro arquivo que atende ao padrão de nome e cujo nome é maior (em ordem lexicográfica) do que o nome do arquivo anterior. O conector não retorna para arquivos anteriores, mesmo que novos registros sejam adicionados a eles.

O conector usa o atributo InitialFileName apenas quando esses dois atributos são iguais a "não", e o arquivo de controle não contém o nome do último arquivo.

Verificação de consultas

O conector verifica o texto da conexão e a consulta apenas se o arquivo definido existir. Se for definido usando caracteres curinga, o conector verificará apenas em relação ao primeiro arquivo.

Erros podem ocorrer quando o conector tentar ler de um novo arquivo. Nesse caso, o conector será interrompido imediatamente se o atributo crítico for igual a "sim". Se for igual a "não", o conector não continuará a executar essa consulta, mas continuará com as outras consultas. Como alternativa, o conector deixa o arquivo atual e passa para o próximo arquivo.

Variáveis internas do conector

Há duas variáveis internas que podem ser usadas no arquivo de configuração para se referir ao contexto atual do nome do arquivo. Elas são: `file_name` e `:file_name_no_ext`, que se referem ao nome do arquivo atual e ao nome do arquivo atual menos a extensão do arquivo, respectivamente.

Essas variáveis podem ser usadas no elemento `SelectStatement` e no atributo `SelectInitialValues QueryKeyField\Function`.

O conector substitui a variável pelo nome do arquivo nas consultas.

Por exemplo:

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

Criar um conector usando a interface com o usuário do CA Business Service Insight

Cada conector configurado para ser executado no ambiente do CA Business Service Insight deve ser registrado na interface com o usuário, além de ser definido no registro. O motivo por trás dessa etapa é principalmente estabelecer as configurações no lado do listener de conectores para que esteja pronto para receber eventos do conector. Durante essa etapa, todas as configurações do conector, como tabelas de conversões e eventos, são definidas.

Siga estas etapas:

1. Crie o conector.
2. Escolha recursos/conectores.
3. Verifique os conectores existentes na lista para ver se nenhum está definido na mesma porta que o seu conector, isto é, a mesma porta que está definida no arquivo de configuração do conector no diretório OblicoreInterface\OnlineInterface\Port.
4. Clique em Adicionar e, em seguida, selecione o método que deseja usar para criar o conector. Existem diversas opções:
 - a. **Criar manualmente:** define o listener de conectores para estabelecer conexão com o conector definido (ou que será definido) manualmente.
 - b. **Usando o assistente:** permite criar o conector usando uma interface do assistente tela a tela. Consulte a próxima seção para obter mais detalhes sobre isso.
 - c. **A partir de um modelo**
 - d. **A partir de um arquivo de configuração existente:** permite fazer upload de um modelo de conector pré-configurado que preenche automaticamente os campos do assistente de conector com as opções definidas no arquivo de configuração especificado.
 - e. A tela resultante varia de acordo com a opção escolhida.
5. Preencha os campos:

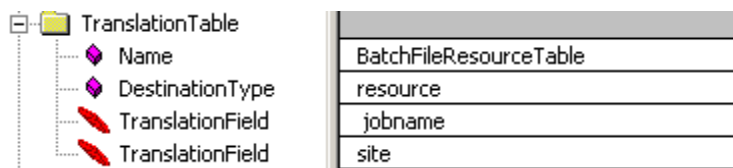
Endereço de rede - digite o endereço IP do conector. Host local, caso seja local para o servidor de aplicativos, caso contrário, digite a porta definida.
6. Clique em Salvar.

Para criar as tabelas de conversões necessárias:

Observação: essas etapas devem ser executadas para cada tabela de conversões definida no arquivo de configuração:

1. No menu Design, clique em Conversão, Tabelas de conversões e clique no botão "Adicionar".
2. Todas as configurações de tabela de conversões devem corresponder à definição de tabela equivalente no arquivo de configuração do conector:

- *Nome*: deve corresponder ao atributo Nome no arquivo de configuração, Tabela de conversões e Nome.
- Campos de origem: devem ter todos os campos de TranslationField da tabela de conversões. Adicionam todos os campos. Pode haver uma combinação de dois ou mais campos que compõem o valor de conversão. A seguir, há um exemplo de como isso pode ser:



3. Tipo de destino: deve novamente corresponder ao atributo DestinationType da definição da tabela de conversões no arquivo de configuração. (recurso, event_type, etc.).
4. Conectores registrados: adicionam os conectores que usarão essa tabela de conversões. Mais de um conector pode usar uma única tabela de conversões.
5. Clique em Salvar.
6. Importe a definição dos campos Evento para cada tipo de evento.

Para importar a definição de campos de um arquivo de configuração de conector específico, esse conector deve ter sido executado e conectado ao CA Business Service Insight ao menos uma vez. Quando o conector se conecta ao CA Business Service Insight, envia o arquivo de configuração ao CA Business Service Insight, o que permite que o sistema use a definição de campos dele.

Observação: como alternativa, você pode especificar as definições de campo manualmente para o tipo de evento.

7. Ative o conector:
 - a. No menu Design, clique em Aquisição de dados, Conectores.
 - b. Clique no botão Iniciar para o conector.

A tabela a seguir apresenta os vários status dos conectores:

Status	Descrição
Inativo	Estado inicial. O conector está inativo e ainda não foi iniciado.
Listener inativo	O serviço do listener de conectores (Dispatcher) não foi iniciado.
Iniciando	O conector está iniciando.
Iniciado	O conector foi iniciado.
Interrompendo	Interrompendo.
Pausando	Pausando.
Pausado	Pausado.

Não está em execução	O conector não está em execução no computador do conector.
Erro	Erro no arquivo de configuração do conector; o conector não pode iniciar.
Erro de conexão	Erro ao conectar o conector (host/porta errada) ou antes o conector foi executado pela primeira vez e nenhum sinal foi detectado. Status quando o conector é executado pela primeira vez.
Bloqueado	O número máximo de eventos rejeitados foi atingido.

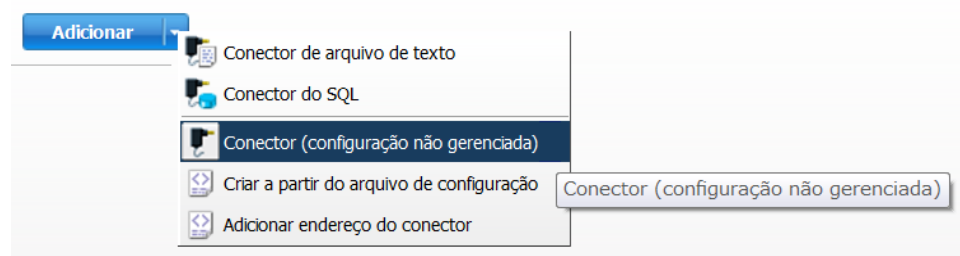
Criar um conector usando o assistente de conector

O assistente de conector é um novo recurso da interface com o usuário do CA Business Service Insight e fornece orientação para a criação de um novo conector usando uma interface mais intuitiva do que o editor de XML. O assistente o orienta por meio de uma série de guias que contêm todas as informações necessárias para criar um conector e, no final, permite que você baixe uma cópia do arquivo de configuração XML compilado. Há, no entanto, algumas limitações para o que o assistente pode fazer.

Atualmente, não permite que você:

- Faça referência à mesma estrutura de entrada (formato de entrada) de diferentes interfaces de fonte de dados.
- Faça referência à mesma estrutura de saída (conversor) de diferentes entradas.
- Configure uma troca de formato de entrada e use-a para decidir qual formato de entrada da interface de fonte de dados será usada.
- Forneça uma definição de um campo ID da fonte de dados na estrutura de saída.
- Forneça uma definição de mais de um arquivo no texto da conexão para o texto da consulta ou arquivos do Excel com a mesma consulta.
- Use as restrições de tempo UTC ou UtcNow.
- Especifique valores que iniciam ou terminam com um espaço.

Ao criar um novo conector por meio do assistente, há quatro opções para escolher, conforme mostrado na figura a seguir:



As duas primeiras opções permitem criar um conector de arquivo de texto ou um conector do SQL usando a interface com o usuário do assistente. A próxima opção de conector (configuração não gerenciada) é escolher quando adicionar um conector pré-configurado que foi criado usando o editor de XML. Essa opção não permite que você edite a configuração para esse conector usando o assistente mais tarde. A última opção Criar a partir do arquivo de configuração permite que você faça upload de uma configuração de conector pré-configurado e solicite que o sistema importe-a para a interface do assistente para edição adicional. Essa opção exige que nenhuma das limitações do assistente mencionadas acima seja implementada nessa configuração específica.

A partir desse ponto, as opções de configuração do assistente de conector fornecem a mesma funcionalidade descrita para a configuração manual do conector. Sua finalidade é fornecer uma interface mais simples e amigável para editar as configurações. Como as mesmas funções e princípios da alternativa manual se aplicam, consulte as seções relevantes para obter detalhes.

Executar e testar o conector

Geralmente, não é possível concluir a configuração do arquivo de configuração do conector em um único ciclo. Uma série de iterações podem ser necessárias para a execução do conector, e os resultados são verificados para garantir que a configuração do conector está correta.

A seguir, estão algumas das ocorrências mais comuns que devem ser corrigidas:

- Problemas de conexão (entre a fonte de dados e o conector, ou entre o conector e o listener).
- Erros no arquivo de configuração, como:
 - Estrutura incorreta.
 - Uso incorreto de atributos.
 - Diferenciação incorreta de maiúsculas e minúsculas (o conector diferencia maiúsculas de minúsculas - "Sim" e "sim" são diferentes, etc.).
 - Atribuição de valor incorreta.
- Erros de manipulação de dados, como estrutura do evento de saída, valores de evento incorretos, erro em consultas.

Siga estas etapas:

1. Defina o conector para RunOnce = "yes" e LogDebugMode = "yes", bem como RejectedEventsUpperLimit para um número razoável (consulte [Modos de execução do conector](#) (na página 87)).

A figura a seguir mostra as configurações necessárias para teste.

2. Também é possível usar o modo offline para as configurações do arquivo de configuração.
3. Depois que o arquivo de configuração tiver sido carregado com êxito, altere a configuração de volta para online (consulte Modos de execução do conector).
4. Cada iteração inclui as seguintes etapas:
 - a. Atualizar/corrigir o arquivo de configuração do conector.
 - b. Excluir todos os arquivos de saída do conector.
 - c. Executar o conector, clicando duas vezes no atalho para o executável do *conector* ou no arquivo *.bat* que foi criado.
 - d. Abrir o arquivo de log do conector usando o navegador de logs (utilitário Log Browser.exe colocado em Utilitários do CA Business Service Insight) e verificar se não há mensagens de erro.
5. A primeira etapa é obter um arquivo de configuração correto e atingir um estado no qual o conector carrega o arquivo de configuração com êxito. Repita essa etapa até se conectar ao CA Business Service Insight e à fonte de dados com êxito e ter eventos rejeitados e solicitações de conversão.

6. Para concluir essa etapa, verifique o seguinte:

- Não há mensagens de erro no arquivo de log do conector.
- O conector se conecta ao CA Business Service Insight e à fonte de dados com êxito.
- O conector enviou solicitações de conversões e eventos rejeitados.

Você deve ver a letra "R" exibida no console para cada registro que o conector rejeitar. Lembre-se de que eventos rejeitados são esperados até que todas as conversões necessárias tenham sido concluídas.

7. Verifique se o arquivo rejectedEvents contém registros e não está vazio.

8. Efetue login no CA Business Service Insight, vá para a página *Entradas de conversão* e pesquise entradas de conversão pendentes do conector. Deve haver várias entradas, uma para cada solicitação de conversão que o conector enviou.

AVISO: a exclusão dos arquivos de saída do conector é muito arriscada. Deve ser feita apenas nessa etapa para fins de teste. Por exemplo, ao excluir o arquivo de controle, o conector perde o controle dos arquivos já lidos e, portanto, poderá perder dados ou ler os arquivos novamente. O único arquivo que pode ser excluído durante o modo operacional sem consequências funcionais é o arquivo de log.

Para usar o navegador de logs para exibir as mensagens de erro:

Se houver uma mensagem de erro, clique duas vezes na mensagem e leia o conteúdo. Isso geralmente é causado por um erro no arquivo de configuração.

Conversões de recursos e eventos

Na etapa anterior, havia uma série de eventos rejeitados criados durante a execução do conector. Esses eventos rejeitados eram armazenados no arquivo RejectedEvents.txt, mas também eram armazenados como entradas de conversão pendentes no banco de dados do CA Business Service Insight. A próxima etapa no processo de carregamento de dados brutos no sistema é fornecer uma conversão do que foi avaliado, de modo que o CA Business Service Insight possa usar esses dados, conforme necessário.

Eventos de dados brutos podem ser rejeitados porque o evento OU o recurso não foi definido no sistema. Os eventos pendentes são definidos de acordo com o tipo de tabela de conversões do qual são provenientes. Os exemplos mais comuns é que os eventos sejam provenientes de uma tabela de conversões e os recursos sejam provenientes de outra.

Quando um novo recurso é encontrado pelo conector (por exemplo, se um novo servidor for adicionado à ferramenta de monitoramento de rede e for exibido como uma nova entrada na fonte de dados dessa ferramenta de monitoramento), pode ser adicionado ao modelo de recurso do sistema. Há duas etapas que você precisa executar para que esse novo recurso se torne relatável no CA Business Service Insight.

Primeiro, é necessário criar o recurso como uma entidade do CA Business Service Insight (um recurso) e, em seguida, uma conversão deve ser inserida. Isso fornece o vínculo entre a representação do texto encontrado na fonte de dados e a entidade definida como um recurso no CA Business Service Insight. Esse processo de duas etapas pode ser executado em uma única ação por meio da interface com o usuário em um processo conhecido como "Adicionar e converter", com o qual o novo recurso e a entrada de conversão necessária podem ser criados em um único procedimento. Ao adicionar e converter, várias entradas podem ser selecionadas, desde que tenham as mesmas configurações de alocação. A conversão é o processo de criar a correspondência entre o valor da fonte de dados e a entidade do CA Business Service Insight. Ao criar conversões, uma entrada é adicionada à tabela de conversões com os valores correspondentes. Em seguida, em consultas subsequentes à fonte de dados, o conector saberá automaticamente como lidar com esse novo valor.

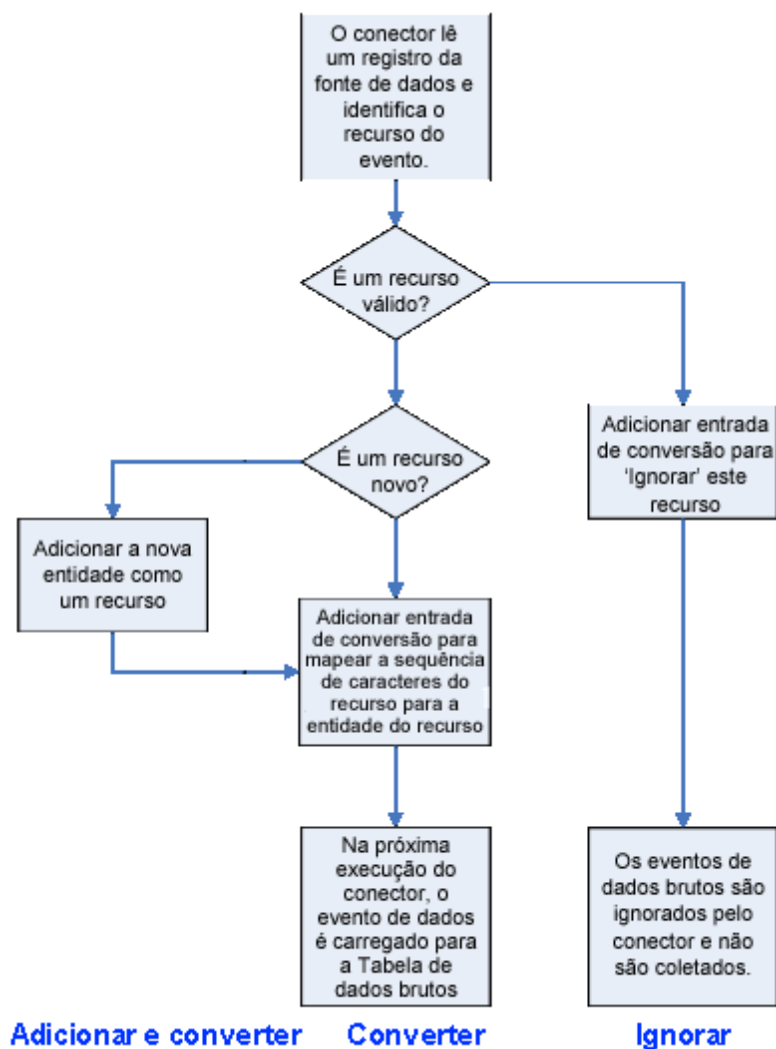
Nessa etapa, o conector já foi executado e enviou solicitações de conversão para cada valor nos campos de conversão. Os eventos associados a esses valores foram rejeitados e aguardam para serem enviados ao CA Business Service Insight assim que a conversão tiver sido concluída. As conversões podem ser feitas manualmente ou automaticamente usando um script de conversão.

As seguintes ações de conversão são possíveis para entradas de conversão pendentes:

Converter: permite a criação de uma entrada na tabela de conversões, fazendo a correspondência entre um valor da fonte de dados e a entidade do CA Business Service Insight relevante. A entidade do CA Business Service Insight para a qual conversões são feitas já deve existir. (Um bom exemplo disso pode ser o caso de um erro de digitação em um recurso de uma fonte de dados. Pode haver nomes diferentes em uma fonte de dados que, na verdade, fazem referência a uma única entidade lógica. Isto é, Server001 e SERVER001. Os recursos do CA Business Service Insight diferenciam maiúsculas de minúsculas.)

- **Adicionar e converter:** permite a criação de uma nova entidade no CA Business Service Insight e adiciona uma entrada de conversão para essa entidade na tabela de conversões ao mesmo tempo. Essa é a ação mais comum realizada em entradas de conversão pendentes porque o mecanismo de conversão é usado para criar a infraestrutura no CA Business Service Insight.
- **Ignorar:** ao ignorar uma entrada, todos os eventos associados são ignorados e não são enviados à tabela de dados brutos do CA Business Service Insight. Os eventos ignorados serão perdidos. Por exemplo, se a fonte de dados contiver informações sobre todos os servidores em um data center, mas apenas os dados dos servidores de aplicativos forem necessários para o cálculo de nível de serviço, todos os servidores chegarão ao conector para conversão, mas apenas os servidores de aplicativos serão convertidos. Todos os outros serão ignorados, pois o CA Business Service Insight garante que o evento desnecessário também será ignorado. Uma entrada ignorada pode ser convertida em uma etapa posterior, se necessário, mas só captura esses dados a partir desse ponto.
- **Excluir:** a entrada de conversão é removida e o evento rejeitado associado também é excluído. Se o mesmo valor posteriormente for enviado novamente pela fonte de dados, uma nova entrada pendente será criada.

O fluxograma a seguir resume os casos de uso dessas opções:



Conversões manuais

As conversões manuais são necessárias quando a entidade já existe no CA Business Service Insight. Isso pode ocorrer em diversas situações. Por exemplo, quando um script de conversão tiver sido executado para criar automaticamente os recursos a partir de uma fonte externa (mas as conversões não puderem ser automatizadas). Ou quando uma fonte de dados possuir entradas ambíguas e um recurso tiver sido definido duas vezes de maneiras diferentes (isto é, Server001p e server001P). Pode até mesmo ser devido a recursos que foram criados manualmente.

Para executar uma conversão manual quando o recurso já existe:

1. No menu Design, clique em Conversões, Entradas de conversão.
2. Por padrão, todas as entradas pendentes são exibidas.
3. Selecione a entrada de conversão pendente que deseja converter marcando a caixa de seleção próxima a ela.
4. Clique em Converter.
5. Escolha a entidade relevante (Recurso, Evento, etc.) na lista. (Se nenhum item for exibido na lista, talvez seja necessário alterar os critérios de pesquisa padrão na caixa.)
6. Selecione o Recurso/Evento na lista de entidades exibida, clicando na linha que contém o item. Ele permanece realçado quando selecionado.
7. Clique em Converter. A entrada de conversão agora está armazenada no sistema.

Para executar uma conversão manual quando o recurso NÃO existir ainda:

1. No menu Design, clique em Conversão, Entradas de conversão.
2. Por padrão, todas as entradas pendentes são exibidas.
3. Selecione a entrada de conversão pendente que deseja transformar em um recurso do CA Business Service Insight e converter, marcando a caixa de seleção próxima a ela.
4. Clique em Adicionar e converter.
5. Verifique se o nome do recurso está especificado como desejar. Você também pode personalizar o campo *Nome de exibição* para alterar a maneira como o recurso é exibido em um relatório. Se esse recurso precisar ser gerenciado como parte de um "Conjunto de alterações", o conjunto de alterações específico também deverá ser especificado aqui.
6. A data de efetivação do recurso deve ser configurada como a data em que esse recurso deve ser relatável no sistema. Observe que o recurso NÃO será exibido nos relatórios antes da data especificada aqui.
7. Clique na guia Detalhes e selecione as seguintes opções para a associação do recurso: Vigente (verdadeiro/falso), Tipo de recurso, inscrição de Grupo de recursos, Serviço e inscrição de Contrato.
8. Clique em Salvar e converter. O recurso será adicionado ao catálogo de serviços do recurso e a entrada de conversão também será armazenada no sistema. Depois de lidar com todas as entradas pendentes, você pode verificar se os dados foram carregados no sistema:
9. Procure os recursos no CA Business Service Insight e verifique se o novo recurso foi criado.
10. Execute o conector novamente.
11. Verifique se o arquivo de eventos rejeitados está vazio, no conteúdo e no tamanho.
12. Verifique se o arquivo de eventos enviados está vazio, no conteúdo e no tamanho.
13. Verifique com a ferramenta de dados brutos se os eventos foram adicionados aos dados brutos.

Instalação da infraestrutura

A instalação da infraestrutura inclui a definição das entidades de Modelagem de dados conforme identificado durante a fase de design.

Essa etapa não inclui a definição de todas as entidades de infraestrutura, e será concluída quando a configuração do conector estiver concluída.

Durante essa etapa, as seguintes entidades são fornecidas para o sistema do CA Business Service Insight:

- Eventos
- Tipos de recurso
- Recursos e alocações de recursos
- Grupos de recursos

Observação: uma vez que o conector tiver sido executado com êxito, você pode usar o recurso de importação automática ao definir os campos de evento.

Conversões automáticas com scripts de conversão

A conversão automática é a automação dos processos e infraestrutura de criação e conversão, com base em uma fonte de dados externa usando scripts que são executados e executam as ações de conversão.

A conversão automática é feita por meio de scripts de conversão. Os scripts de conversão aceleram o processo de mapeamento de novos recursos de TI e de negócios no CA Business Service Insight. O script de conversão automaticamente identifica quando uma nova entrada de conversão é recebida e converte os recursos, permitindo um mapeamento de recursos rápido e eficiente. A automação oferece suporte a CMDBs, permitindo que o sistema identifique os recursos com base em suas definições de configuração. A conversão automática tem diversos benefícios, inclusive facilita a manutenção das conversões e previne erros. Scripts de conversão podem ser usados para criar novos recursos e alocar as alterações.

Além disso, os scripts de conversão podem ser usados para:

- Converter entradas em objetos existentes do CA Business Service Insight.
- Adicionar novos objetos do CA Business Service Insight e fazer a conversão para eles com base nas entradas de conversão existentes.
- Criar objetos e converter as entradas com base nas tabelas de fora do CA Business Service Insight, por exemplo, tabelas de recursos de outro CMS externo.
- Verificar se um objeto existe.
- Criar recursos e realizar alocações de recursos, como tipos de recurso, grupos de recursos, partes contratuais e componentes de serviço.
- Alocar/cancelar alocação de recursos de conjuntos de alterações.

Como o processo de conversão também tem suporte como um procedimento manual na interface com o usuário, é necessário decidir qual processo de conversão deve ser escolhido. Quando fizer isso, considere os seguintes prós e contras referentes à conversão automática:

- Aumento da complexidade do projeto - os scripts de conversão exigem habilidades apropriadas e esforços de desenvolvimento de scripts.
- Mais tempo de desenvolvimento, bem como tempo de garantia de qualidade para testes.
- Não precisa ser implementada nos casos em que o processo de conversão é apenas um esforço inicial executado uma única vez.
- Pode ser programada para implementação como parte de uma abordagem de segunda fase.
- Manutenção contínua reduzida.
- Erros humanos de conversão reduzidos.
- Para obter mais informações sobre a criação e a execução de scripts de conversão, consulte o Guia de *SDK* no capítulo de integração com o CMDB.

Tópicos avançados para o recurso do conector

A seguir, são abordados tópicos avançados para o recurso do conector.

Singularidade do evento

A singularidade do evento é um mecanismo do conector que fornece um processo que impede a inserção de dados brutos duplicados na tabela T_Raw_Data.

Sem a singularidade do evento, quando o conector é executado em relação a uma fonte de dados e carrega os eventos no banco de dados, nenhuma validação será executada para um evento idêntico. O recurso singularidade do evento resolve essa questão ao fornecer a capacidade de especificar se deseja ou não verificar a singularidade dos eventos antes da inserção e o que fazer se a situação ocorrer. No entanto, o processo de verificação pode causar um impacto negativo no desempenho do conector.

A solução permite que o usuário defina uma chave, que pode ter como base os campos do evento. Essa chave representa a exclusividade do evento, o que significa que, se dois eventos tiverem a mesma chave, são eventos idênticos.

Também é possível especificar a ação que será realizada caso uma chave duplicada já tenha sido encontrada no banco de dados. Essas ações são discutidas a seguir.

Observação: a chave pode ser definida como uma combinação de vários campos do conversor.

Arquivo de configuração do conector com singularidade do evento

TranslatorCollection/Translator/OnDuplication

Esse campo determina o que deve ser feito quando o evento já existe no banco de dados.

Os valores possíveis são:

- **Adicionar** - insere eventos (adicionais) no banco de dados.
- **Atualizar** - causa a exclusão (em alguns casos) do evento existente e a inserção de um novo após a validação de que o evento foi alterado.
- **updateAlways** - causa a exclusão (em alguns casos) do evento existente e a inserção de um novo sem verificar a existência de alterações.
- **Ignorar** - não insere um novo evento no banco de dados.

O valor padrão é Adicionar.

Ignorar versus Adicionar

Pode haver pequenas reduções de desempenho ao executar o conector no modo Ignorar. No entanto, no modo Adicionar, o sistema aciona o recálculo em todos os casos de um evento duplicado.

Atualizar versus updateAlways

Usar a opção Atualizar diminui o desempenho do conector, ao passo que updateAlways diminui o desempenho do cálculo ao acionar o recálculo para todos os casos.

TranslatorCollection > Translator > TranslatorFields > TranslatorField > IsKey

Esse atributo determina se o campo ao qual pertence deve contribuir com seu valor na chave exclusiva para os dados brutos. É incluído quando value = "yes" e não incluído quando value = "no". O valor padrão (se nenhum valor for fornecido) é "sim" para os campos padrão (ResourceId, EventTypeId e Data e hora) e "não" para todos os outros. A chave deve ser escolhida com cuidado para garantir que os dados brutos mantenham a integridade necessária e que os cálculos estejam totalmente precisos.

Comportamento do listener de conectores com singularidade do evento

Ao receber um novo evento do conector, o listener verifica o valor do campo `OnDuplication`. Quando o valor for "adicionar", o processo normal de inserção será executado. Quando o valor não for "adicionar", o listener verificará a existência de um evento com a mesma `UniqueKey` e a mesma `ID` do leitor no banco de dados. Se o banco de dados já contiver um evento como descrito, o novo evento não será inserido no banco de dados quando o valor de `OnDuplication` for "ignorar".

Quando o valor de `OnDuplication` for "atualizar", uma verificação em busca de alterações no evento será executada. Se todos os campos forem idênticos, o novo evento não será inserido no banco de dados.

Quando o valor de `OnDuplication` for "updateAlways", a verificação anterior será ignorada e uma atualização ocorrerá de qualquer maneira.

Nos modos atualizar e `updateAlways`, as seguintes etapas são executadas:

- Se `ResourceId`, `EventTypeId` e `Data` e hora tiverem sido alterados, o recálculo será feito para todas as métricas que usam a antiga fórmula do evento.
- O evento é atualizado com os valores apropriados.

Agregar dados da transação

Os dados da transação geralmente são coletados para comparar aos limites ou para calcular as porcentagens periódicas eventuais de êxito. Por exemplo, a cada 5 minutos, uma transação virtual é executada em um sistema e o resultado (tempo de resposta em milissegundos) é armazenado, conforme mostrado a seguir:

```
[...]  
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  
[...]
```

Em outras situações, em vez de usar transações virtuais, pode haver acesso a transações reais em execução no sistema. Nesses casos, centenas ou até milhares de transações podem ser realizadas por hora.

Em ambos os casos acima, o carregamento desse volume de informações no CA Business Service Insight deve ser evitado, se possível.

Agregar os dados por períodos é a melhor maneira de reduzir as quantidades de dados. Quando o limite que funciona como base para medir o êxito é fixo, a agregação pode ser executada permitindo que o conector conte o número de transações que tiveram êxito dentro do período agregado. Por exemplo, se o limite de êxito no exemplo anterior estiver definido como 500 milissegundos, apenas cinco de sete transações tiveram êxito durante o período exibido. O problema dessa abordagem é o limite fixo: e se, posteriormente, alguém desejar alterar o limite? Nesse caso, os dados brutos devem ser lidos e testados novamente pelo conector com base no novo limite.

Portanto, o ideal seria que o conector agregasse os dados da transação de uma maneira flexível sem perder os dados significativos.

Uma solução limitada é permitir que o conector teste as transações com base em vários limites. Há duas maneiras de fazer isso:

- Um evento com vários testes - Event Type = {TotalTransactions, TransBelow250, TransBelow500, TransBelow750, [...]}
- Vários eventos com um teste - Event Type = {TotalTransactions, Threshold, TransBelowThreshold}.

As duas opções têm o mesmo problema - os limites podem ser alterados no futuro apenas dentro de um pequeno conjunto de valores predefinidos.

Solução recomendada

Premissa - todos os possíveis limites são múltiplos de um determinado número. Por exemplo, todos os limites (em milissegundos) são múltiplos de 250, desse modo, 0, 250, 500, 1750 e 3000 milissegundos são possíveis limites.

Com base nessa premissa, a solução sugerida é agregar as transações, arredondando todos os valores para um múltiplo comum e contando o número de transações que se enquadram em cada valor arredondado. Event Type = {RangeFrom, RangeTo, TransactionCount}

Por exemplo, os seguintes eventos serão gerados para agregar os dados exibidos acima, onde o múltiplo comum é 250 milissegundos:

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

Comentários:

A data e a hora desses eventos seriam as mesmas. (Por exemplo, todos os eventos agregados podem ocorrer em 1/1/2007 00:00, e pode haver outro conjunto de eventos para o próximo exemplo de hora, 1/1/2007 01:00, presumindo que houve a agregação de hora em hora.)

RangeTo é calculado arredondando uma transação para o múltiplo comum (consulte a seguir).

RangeFrom é RangeTo menos o múltiplo, mais 1. É especificado apenas para fins de clareza, e você pode ignorá-lo.

Por exemplo - agregar por hora pode ser semelhante ao seguinte (substituir MULTIPLE pelo valor do múltiplo):

```
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
```

Na lógica de negócios, a seguinte condição pode ser aplicada aos eventos:

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

Algumas ideias conclusivas:

- Como as transações tendem a ser distribuídas normalmente, o número de eventos agregadores deve ser relativamente baixo.

- Selecionar múltiplos comuns mais altos irá gerar eventos menos agregadores.
- O volume do evento agregador sempre deve ser menor ou igual ao volume de dados brutos.

Executar um conector com proteção de um firewall

Para que seja possível executar um conector com proteção de um firewall, a seguinte solução é sugerida:

Abra duas portas: a primeira porta é a atribuída para o conector no CA Business Service Insight; a segunda porta, que é opcional, mas recomendada, é a porta do servidor de logs. O servidor de logs é definido por padrão como porta 4040. Abrir a porta do servidor de logs permite que o conector envie relatórios ao log do CA Business Service Insight e também permite que ele gere alertas. Isso é opcional, pois o conector sempre envia relatórios para um arquivo de log local.

Para ambas as portas, o protocolo em uso é TCP/IP.

Carregar dados de vários diretórios

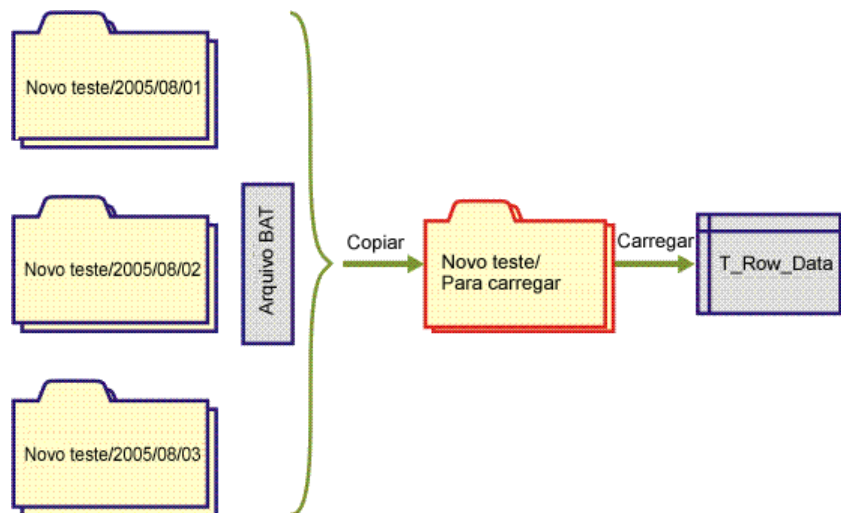
Essa seção descreve a solução que pode ser implementada se os arquivos de fonte de dados (entrada para um conector do CA Business Service Insight) estiverem localizados em diretórios diferentes a cada dia ou para todos os períodos definidos (de acordo com uma convenção de nomenclatura específica).

Por exemplo, a estrutura de diretório `c:\NewData\YYYY\MM\DD`. Nesse exemplo, para cada dia há um novo diretório, no qual os arquivos relevantes para esse dia são localizados.

O conector do CA Business Service Insight deve navegar pelo diretório mais novo e carregar os novos arquivos.

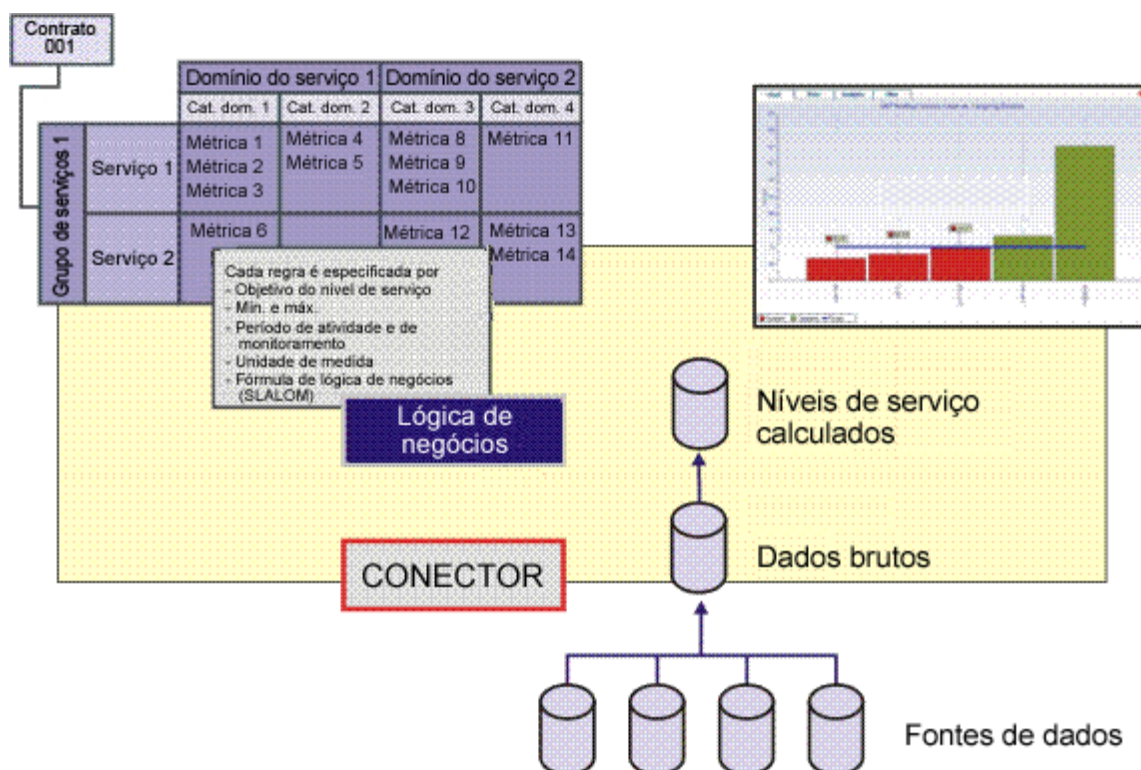
Uma solução de contorno disponível é adicionar alguns comandos ao início do arquivo bat que executa o conector. Esses comandos copiam as fontes de dados que devem ser carregadas da pasta específica (de acordo com as convenções) para uma única pasta dedicada que foi criada especificamente para essa finalidade. O conector sempre carrega as informações dessa pasta.

A imagem a seguir descreve essa solução:



Criação de script de lógica de negócios (Especialista em lógica de negócios)

A imagem a seguir representa a posição da lógica de negócios no CA Business Service Insight. Fica abaixo de cada métrica nos contratos.



Nessa etapa da implementação, os conectores relevantes são configurados e os registros de dados brutos já devem estar disponíveis na tabela T_RAW_DATA no CA Business Service Insight. É possível aplicar a lógica de negócios aos eventos a fim de gerar os resultados reais de nível de serviço para cada métrica.

A criação de script de lógica de negócios é o processo de escrever o código que opera logicamente em dados brutos (dados brutos enviados por conectores) para calcular os níveis de serviço.

Cada métrica tem sua própria fórmula de lógica de negócios com a qual calcula o nível de serviço real, embora muitas das métricas no sistema em geral tenham uma lógica em comum que pode ser aplicada a diferentes conjuntos de eventos de dados brutos.

Por exemplo, uma métrica que calcula o tempo de resolução do ticket de Gravidade 1 e uma outra métrica que calcula o tempo de resolução do ticket de Gravidade 2 avaliam um conjunto diferente de registros - uma usa apenas os tickets de Gravidade 1 e a outra apenas os tickets de Gravidade 2. No entanto, é provável que ambas apliquem o mesmo método para calcular o tempo de resolução. O script de tempo de resolução será desenvolvido e testado uma vez, definido como um módulo de lógica de negócios e, em seguida, será usado por ambas as métricas por meio da inclusão desse módulo nas áreas da lógica de negócios das métricas.

Portanto, geralmente ao desenvolver scripts de lógica de negócios os principais módulos ou modelos são desenvolvidos para torná-los disponíveis para todas as métricas do sistema. Além disso, cada categoria do domínio geralmente reflete um tipo diferente de medição e, portanto, cada categoria do domínio geralmente seguirá um módulo ou modelo de lógica de negócios diferente.

Fluxo de trabalho da criação de script de lógica de negócios

A etapa de criação de script de lógica de negócios envolve a execução das seguintes etapas:

- Definir uma fórmula

Criar a fórmula com base nos requisitos de cálculo definidos na fase de design. As fórmulas definidas são todas fórmulas exclusivas usadas em suas diversas permutações nas métricas do contrato, cada uma como um módulo de lógica de negócios.

Por exemplo, se o contrato possui três métricas para calcular o tempo médio de resolução de ticket e uma métrica para cada prioridade do ticket, uma única fórmula é desenvolvida para calcular o tempo de resolução do ticket com a prioridade do ticket como seu parâmetro. Essa fórmula, uma vez testada, é definida como um módulo e é vinculada a todas as métricas relevantes.

- Testar a fórmula

Testes são realizados para garantir que a fórmula está definida corretamente e sem erros, e que os cálculos geram os resultados esperados. É importante abranger todos os casos extremos de término e condições de limite como parte dos testes. O escopo de lógica de negócios é onde a fórmula é executada para fins de teste. Quando inicialmente definida, a fórmula é testada em sua totalidade. Em seguida, depois de ser aplicada a todas as métricas como um módulo, é importante executar cada métrica no escopo ao menos uma vez para ver se ela recebe eventos (isto é, se o registro está correto) e se está produzindo um resultado razoável.

- Definir o módulo de SLALOM associado

Cada módulo é um cálculo exclusivo de lógica de negócios, e com a sua definição de parâmetro pode ser aplicado a todas as métricas relevantes. Durante a definição do módulo, é importante que o módulo seja testado e documentado detalhadamente; o que o módulo faz (descrição do cálculo), quais parâmetros são esperados (nome, significado e uso), etc.

- Vincular todas as métricas ao módulo de lógica de negócios relevante

Para cada métrica nos contratos definidos, um vínculo para o módulo de lógica de negócios relevante deve ser definido. Então, ele deve ser executado no escopo de lógica de negócios para garantir que o vínculo foi implementado corretamente e que o registro está funcionando corretamente, recebendo os eventos relevantes e produzindo os resultados esperados.

Módulos de lógica de negócios

Há diversas considerações importantes na criação de módulos para a lógica de negócios, especialmente se você estiver usando vários módulos em uma única métrica:

- Para garantir que o uso de um módulo está claro, você deve adicionar um comentário no topo da lógica de negócios para essa métrica.
- Se você estiver usando uma pequena parte do código no espaço de lógica de negócios da métrica e incluindo um ou mais módulos no código, você deve garantir que para qualquer manipulador de eventos ou sub-rotina padrão, removerá essa seção do código da lógica de negócios da métrica principal. Você deve verificar se cada sub-rotina e manipulador de eventos está definido apenas *uma vez* em cada um dos módulos que está sendo usado por uma determinada métrica. Isso é para evitar confundir o mecanismo de cálculo e produzir resultados inesperados.

Observação: se, por exemplo, você definir a função OnPeriodStart() no módulo e colocar um código específico, e, em seguida, deixar a OnPeriodStart() padrão sem nenhum código na tela da lógica de negócios principal da métrica, o mecanismo não saberá qual sub-rotina executar em tempo de execução. Talvez ele não execute o código que você deseja.

- É necessário ser muito cuidado para parametrizar a sub-rotina OnRegistration. Em alguns casos, isso pode disparar o gatilho automático do sistema para recalcular as métricas com base na adição de novos dados brutos.

Dentro da lógica de negócios

Todas as lógicas de negócios estão localizadas em um script orientado a eventos, com base na sintaxe de VBScript. Cada evento que atinge a lógica de negócios aciona um manipulador de eventos.

Os seguintes tipos de eventos são enviados pelo mecanismo para serem avaliados pela lógica de negócios:

- Eventos de dados brutos. As entradas de dados brutos reais nas quais a lógica de negócios baseia seus resultados. O mecanismo envia apenas os eventos de dados brutos relevantes com base no registro da fórmula.
- Eventos do mecanismo. Eventos enviados pelo mecanismo implicitamente que refletem as propriedades da definição da métrica, como definição de período de atividade e período de monitoramento.
- Eventos intermediários. Os eventos gerados por outros scripts de lógica de negócios e que podem ser usados dentro de cada um deles.

A fórmula de lógica de negócios contida na definição da métrica é o que avalia os eventos e gera um resultado de nível de serviço que os relatórios usam como base. Dependendo desses resultados de nível de serviço e da definição de domínio, o mecanismo também gera um desvio como resultado (se um destino de nível de serviço tiver sido aplicado à métrica). Os resultados gerados são armazenados em uma tabela de banco de dados chamada T_PSL. Essa é a tabela consultada pelo Assistente de relatório durante a geração de relatórios, e, portanto, todos os dados do relatório são pré-calculados para garantir que o desempenho da geração de relatórios será maximizado.

Fluxo de eventos

Conforme mencionado anteriormente, as entradas para a lógica de negócios são os eventos do mecanismo e os eventos de dados brutos.

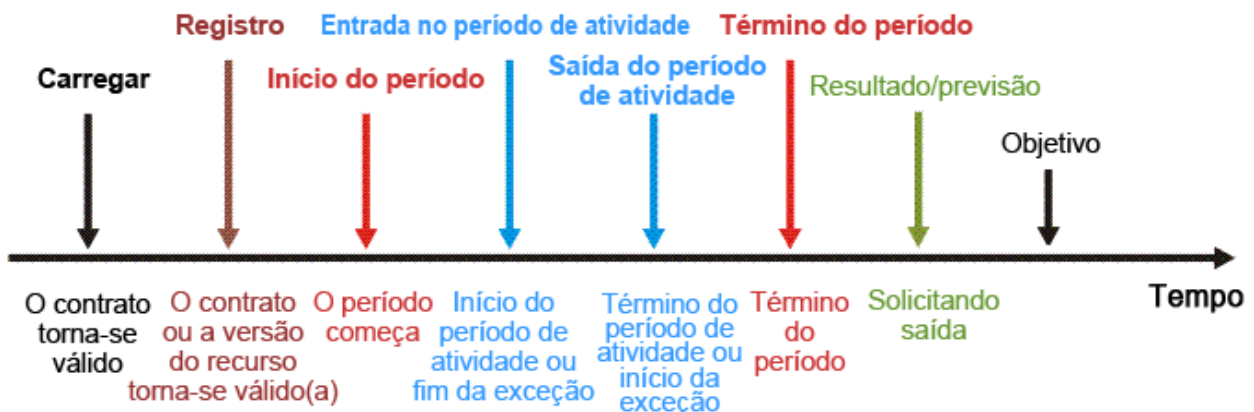
Os eventos de dados brutos recebidos pela lógica de negócios são determinados pela função de registro na qual o código solicita um conjunto específico de eventos de dados brutos definidos pelo tipo de evento e identificadores de recurso.

Na lógica de negócios, o registro também associa uma sub-rotina definida pelo usuário que é executada para lidar com o evento de dados brutos quando um evento for recebido. (Por padrão, é OnXXXEvent, que deve ser renomeado para algo mais significativo.)

Os eventos do mecanismo são acionados pelo mecanismo de acordo com o contrato e as definições de métrica associadas. Quando o evento do mecanismo é acionado e recebido, o mecanismo executa o manipulador de eventos pertinente. Cada evento do mecanismo tem um manipulador de eventos implícito. Esses manipuladores de eventos são as funções e os procedimentos definidos no VBScript. O manipulador de eventos que controla o registro e a função "Resultado" devem ser implementados no código. Todos os outros manipuladores de eventos são opcionais. No entanto, a lógica de negócios não lida com eventos do mecanismo para os quais manipuladores de eventos não estão implementados. Portanto, é uma boa prática mantê-los no local (mesmo que não estejam em uso) para permitir melhorias no futuro.

Observação: ao escrever o script de lógica de negócios, é importante implementar todos os eventos do mecanismo para que seja possível abranger todas as possibilidades finais. Por exemplo, se uma definição de período de atividade não for aplicável durante a primeira etapa da implementação, mas será aplicável no futuro, todas as fórmulas precisarão ser alteradas. Portanto, é recomendável que o Especialista em lógica de negócios defina o comportamento "no período de atividade" e "fora do período de atividade" com antecedência, mesmo que inicialmente ele não seja aplicável, para que quando o comportamento for apresentado, as alterações necessárias no sistema sejam triviais.

A seguir, há vários eventos do mecanismo e seus manipuladores de eventos:



- **OnLoad (Time)** - (opcional), chamado uma vez no início dos cálculos quando o contrato está sendo ativado. Pode ser usado para inicializar variáveis globais.
- **OnRegistration (Dispatcher)** - (necessário), procedimento para solicitar os eventos de dados brutos relevantes e associá-los com os procedimentos definidos pelo usuário com os quais serão manipulados. Os eventos são solicitados e associados a procedimentos usando os métodos do objeto Dispatcher.
OnRegistration é chamado uma vez pelo mecanismo de cálculo no início do cálculo da métrica e novamente a cada vez que um recurso registrado para a métrica se torna vigente depois que o mecanismo avalia o conjunto de alterações feitas para o recurso que possam influenciar o conjunto de eventos transmitidos para a fórmula. O mecanismo tem a definição da solicitação do evento definida pelo tipo de evento e identificadores de recurso, e no caso de um recurso (ou um conjunto de alterações que contém diversos recursos) alterar alguma coisa relacionada a esse conjunto. Quando se torna vigente, um manipulador de eventos OnRegistration é acionado.
- **OnPeriodStart (Time)** - (opcional), chamado no início do período do agente (definido de acordo com a unidade de tempo). O primeiro OnPeriodStart é acionado quando o contrato se torna vigente, em que a comparação dos períodos começa com unidades de tempo de hora arredondada. Em geral, esse manipulador de eventos é usado para periodicamente inicializar variáveis, como um contador que controla o número de incidentes abertos durante o período de cálculo que, em seguida, deve ser inicializado para zero no início de cada período.
- **OnPeriodEnd (Time,IsCompleteRecord)** - (opcional), chamado no início do período do agente (definido de acordo com a unidade de tempo). É sempre chamado no final arredondado da unidade de tempo em horas arredondadas (por exemplo, 24h00). IsCompleteRecord é Verdadeiro quando o período da métrica estiver encerrado (de acordo com o tempo real do servidor de aplicativos) e é Falso ao fazer um cálculo intermediário. Em geral, esse manipulador de eventos é usado para fazer cálculos finais para o período de término a fim de preparar o terreno para o resultado final que será fornecido pela função Resultado.
- **OnTimeslotEnter (Time)** - (opcional), chamado ao inserir um período TimeSlot com base na definição da métrica associada. Por exemplo, se a métrica estiver associada a uma definição de período de atividade do horário comercial em que todos os dias às 8h00 será o início do período de atividade, esse manipulador de eventos será acionado nessa hora todos os dias.
- **OnTimeSlotExit (Time)** - (opcional), chamado ao sair de um período TimeSlot com base na definição da métrica associada. Por exemplo, se a métrica estiver associada a uma definição de período de atividade do horário comercial em que todos os dias termina às 17h00, esse manipulador de eventos será acionado nessa hora todos os dias.
- **Target()** - (opcional), chamado quando a métrica for especificada como tendo um destino dinâmico. Permite determinar o destino de nível de serviço de uma métrica durante o tempo de execução da lógica de negócios. Tais destinos podem incluir o consumo de componentes de serviço ou alterações na infraestrutura. Possui quatro valores, assim como a função Resultado; um para cada modo. Essa função é executada *depois* da função Resultado durante a execução normal.

- **Forecast()** - (opcional), chamado uma vez ao executar uma confirmação da versão do contrato por meio do cálculo do contrato executado uma vez no modo de previsão pelo mecanismo de cálculo. O modo de previsão executa um ciclo completo do mecanismo de cálculo no contrato (desde a data de início até a data de término da versão do contrato). A finalidade desse ciclo é calcular somente os valores de previsão. (Nenhum cálculo é feito na tabela T_RAW_DATA). Essa função é chamada *no lugar* da função Resultado durante esse modo de execução.
- **Result()** - (necessário), chamado pelo mecanismo de cálculo para obter o resultado do cálculo para um determinado período. A função Resultado sempre é chamada depois do manipulador de eventos OnPeriodEnd.

A seguir, estão as etapas seguidas pelo mecanismo de cálculo (serviço PslWriter) para processar uma única fórmula de lógica de negócios:

- O PslWriter carrega a fórmula na memória e executa cada instrução que existe na seção de declarações (a seção de declarações é o código que está fora de uma função ou sub-rotina). Observe também que todas as bibliotecas e módulos vinculados são incluídos e compilados nesse módulo de código único para execução.
- O PslWriter chama o manipulador de eventos OnLoad.
- O PslWriter chama o OnRegistration.
- O cálculo do período começa com a chamada do OnPeriodStart.
- Para cada evento de dados brutos registrado durante OnRegistration que está dentro do intervalo de datas do período, o PslWriter chama o manipulador de eventos definido pelo usuário e associado ao evento.
- Se o início ou o término do período de atividade estiver dentro do intervalo de datas do período, OnTimeslotEnter ou OnTimeSlotExit será chamado.
- Se houver uma alteração na infraestrutura relevante dentro do intervalo de datas do período, OnRegistration será chamado.
- O cálculo do período termina com a chamada de OnPeriodEnd.
- Se um destino dinâmico tiver sido especificado, essa função será chamada.
- O PslWriter chama a função Resultado para obter o resultado final do cálculo do período.

Observação: quando a versão do contrato for confirmada pela primeira vez e uma projeção tiver sido selecionada, a função Previsão será chamada, em vez da função Resultado. No entanto, isso ocorre apenas dessa vez para cada versão.

Durante cada ciclo de cálculo, o mecanismo avalia quais são os eventos do mecanismo e os eventos de dados brutos relevantes com base no período de cálculo. Primeiro, irá classificá-los por hora e, em seguida, enviá-los às fórmulas relevantes para cálculo. A hora do evento de dados brutos é sua data e hora, e a hora do evento do mecanismo é sua hora de acionamento. Em seguida, esses dois tipos de eventos são combinados em uma sequência de tempo e enviados para cálculo.

Os períodos dos eventos estão de acordo com a métrica local associada, mas o parâmetro de hora dos manipuladores de eventos (isto é, `OnPeriodStart (Time)`) e a data e hora do evento de dados brutos estão de acordo com o valor em UTC. O mecanismo os compara de acordo com seus valores em UTC, de forma a usar um ponto de referência constante.

Exemplo:

Se a diferença de fuso horário de UTC de uma métrica for de duas horas (por exemplo, GMT+2), e um evento de abertura de incidente tiver data e hora de 10h00, dentro do mecanismo a data e hora que o manipulador de eventos usará na verdade será modificada de forma adequada e realmente inicia às 8h00 UTC. Supondo que o conector esteja configurado para usar esse mesmo fuso horário, os eventos de dados brutos também terão 2 horas reduzidas para UTC no banco de dados. Quando os eventos são passados para a lógica de negócios, o agente de cálculo responsável pelo cálculo dos eventos no período que inicia às 10h00 usará a hora em UTC para os eventos, que será das 8h00 em diante. No entanto, se você usar a mensagem `out.log` no código para imprimir as datas e horas, ela mostrará a data e hora modificada para UTC e, portanto, mostrará 8h00, apesar do período especificado (de acordo com a métrica) ser às 10 da manhã.

No seguinte requisito de cálculo, é importante converter a data e hora do evento antes de usá-lo:

Se uma métrica for para calcular o número de incidentes que foram fechados no mesmo dia em que foram abertos, será necessário comparar a hora de abertura e de fechamento de cada incidente. Se a hora de abertura e de fechamento de um incidente cair no mesmo dia (e em um período de atividade definido), o incidente será contado.

Há a possibilidade de que durante o processo de modificação do incidente para UTC a partir de sua hora local original, o dia mude (isto é, novamente usando GMT+2). Um incidente aberto à 1h será modificado para 23h no dia anterior em UTC. Desse modo, não será contado mesmo que devesse ter sido. Essa é uma situação em que a hora primeiro deve ser convertida de volta para a hora local da métrica e apenas depois ser comparada. Nesse caso, use o método `GetLocaleTime(utcTime)`. Esse método converte uma hora determinada em um fuso horário UTC para o fuso horário da métrica local.

A seguir, há o código do manipulador de eventos:

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


Registro

O registro é o processo pelo qual a lógica de negócios envia uma solicitação ao mecanismo de cálculo para o conjunto de dados de eventos brutos que se tornará parte do cálculo.

O processo de registro pode ser realizado de duas formas: usando o Assistente de registro ou manualmente, usando o objeto Dispatcher na lógica de negócios.

O uso do Assistente de registro é um processo simples de escolha entre as opções disponíveis. Você tem todas as opções disponíveis no registro manual, sem a capacidade de usar parâmetros. Se você precisar usar parâmetros, terá de executar o registro manualmente. O fluxo básico do assistente exige que você primeiramente determine o tipo de registro que deseja realizar e, em seguida, defina os tipos de recurso e os eventos nos quais o registro deve ser realizado e, finalmente, qual manipulador de eventos será usado para processar os eventos coletados.

Quando os registros estiverem concluídos, serão listados na guia "Registro" da métrica. Observe também que é possível ter mais de uma declaração de registro para uma métrica.

Na verdade, o Assistente de registro usa a mesma funcionalidade que o registro manual, e todas essas opções são abordadas na seção a seguir.

Quando realizado manualmente na lógica de negócios, o registro da fórmula é manipulado pelo manipulador de eventos *OnRegistration*. Ele deve ser implementado na fórmula e acionado sempre que um evento do mecanismo de registro for acionado. O evento de registro é acionado uma vez quando o contrato é ativado e depois cada vez que um recurso ou conjunto de alterações relevante se tornar ativo. Uma alteração no recurso afetado é considerada relevante se afetar os eventos que a métrica deve receber. Por exemplo, se o registro for feito pela parte contratual (*RegisterByContractParty*), isso significa que todos os eventos do tipo definido cujos recursos estão vinculados à parte contratual da métrica fazem parte do cálculo. Nesse caso, sempre que um novo recurso for vinculado ou desvinculado a essa parte contratual, o método de registro será acionado para notificar o mecanismo sobre a alteração.

Os métodos de registro são fornecidos pelo objeto Dispatcher que é transmitido para o *OnRegistration* como um argumento. Os diferentes métodos oferecem várias maneiras de definir os critérios de filtragem com base na definição de tipo de evento e qualquer critério de alocação de recursos, como recursos de um grupo de recursos ou recursos de um determinado tipo.

O uso dos métodos de registro de Parte contratual e Serviço é altamente recomendado, pois facilita o uso da lógica de negócios como um módulo ou modelo. Quando fizer isso, a parte contratual e o serviço relevantes são obtidos da definição da métrica associada, e ao reutilizar a fórmula para diferentes contratos e/ou componentes de serviço, o registro não precisa ser alterado.

Outro método de registro comum é o `RegisterByResourceGroup`, que é útil para trabalhar com os recursos que são agrupados logicamente mas nem sempre estão associados a partes contratuais ou serviços. A atribuição de recursos aos grupos aqui pode ser gerenciada pelo catálogo de recursos (individualmente ou por meio de conjuntos de alterações) e pode até mesmo ser atualizada automaticamente por meio de scripts de conversão.

Em geral, o método de registro é determinado durante a fase de design e é diretamente orientado pelo modelo de dados definido.

Observação: para verificar se o objeto `Dispatcher` foi corretamente usado, a função `OnRegistration` também é chamada durante a verificação de sintaxe de SLALOM. Por esse motivo, não suponha que o `OnLoad` foi executado antes da função `OnRegistration` e não use algumas das propriedades do objeto de contexto, como `"TimeUnit"`, `"IntervalLength"`, `"IsPeriod"`, `"CorrectionsApply"` e `"ExceptionsApply"` no manipulador de eventos `OnRegistration`.

Os métodos de registro também são responsáveis pela associação dos eventos com um procedimento que será acionado de acordo com a data e hora do evento. O procedimento definido pode ter qualquer nome, mas sempre terá o objeto `eventDetails` como seu parâmetro. O objeto `eventDetails` reflete o evento de dados brutos recebido e contém todos os detalhes do evento como campos de dados, conforme mostrado no seguinte registro:

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

A declaração de registro acima informa que todos os eventos de dados brutos do evento `"MemUse"` e associados ao tipo de recurso `"Server"` serão enviados para o manipulador de eventos `"OnMemUseEvent"` na lógica de negócios.

O procedimento a seguir também deverá ser definido adiante na lógica de negócios:

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

Ao fazer referência ao objeto `eventDetails` e usar o parâmetro `"MemoryUsage"`, a declaração extrai o valor do campo `MemoryUsage` do evento que foi passado para a função. Esses campos são os mesmos definidos no tipo de evento, e os nomes de campos diferenciam maiúsculas de minúsculas.

Registrar métricas agrupadas

As métricas agrupadas permitem a definição de uma métrica a ser executada para cada integrante de um grupo de recursos, a fim de aplicar a mesma definição e lógica para um conjunto de itens. O agrupamento pode ser definido de maneira estatística em um conjunto predefinido de recursos, ou de maneira dinâmica nos integrantes de um grupo de recursos. Com o tempo, é possível alterar o grupo e incluir ou excluir seus integrantes.

Observação: para obter uma descrição detalhada, consulte o Apêndice E - Definindo fórmulas da lógica de negócios (Especialista em lógica de negócios).

As métricas agrupadas são usadas quando é necessário calcular um resultado de nível de serviço para cada item em um grupo de recursos. Os itens em um grupo de recursos podem ser recursos ou outros grupos de recursos, portanto, o método de registro em um script de lógica de negócios de uma métrica agrupada deve ser `RegisterByResourceGroup` ou `RegisterByResource`, onde o nome do recurso ou o nome do grupo de recursos especificado é definido como o item no agrupamento. Isso é feito usando a propriedade `"ClusterItem"` do objeto de contexto que contém o nome do item de agrupamento atual.

Exemplo:

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

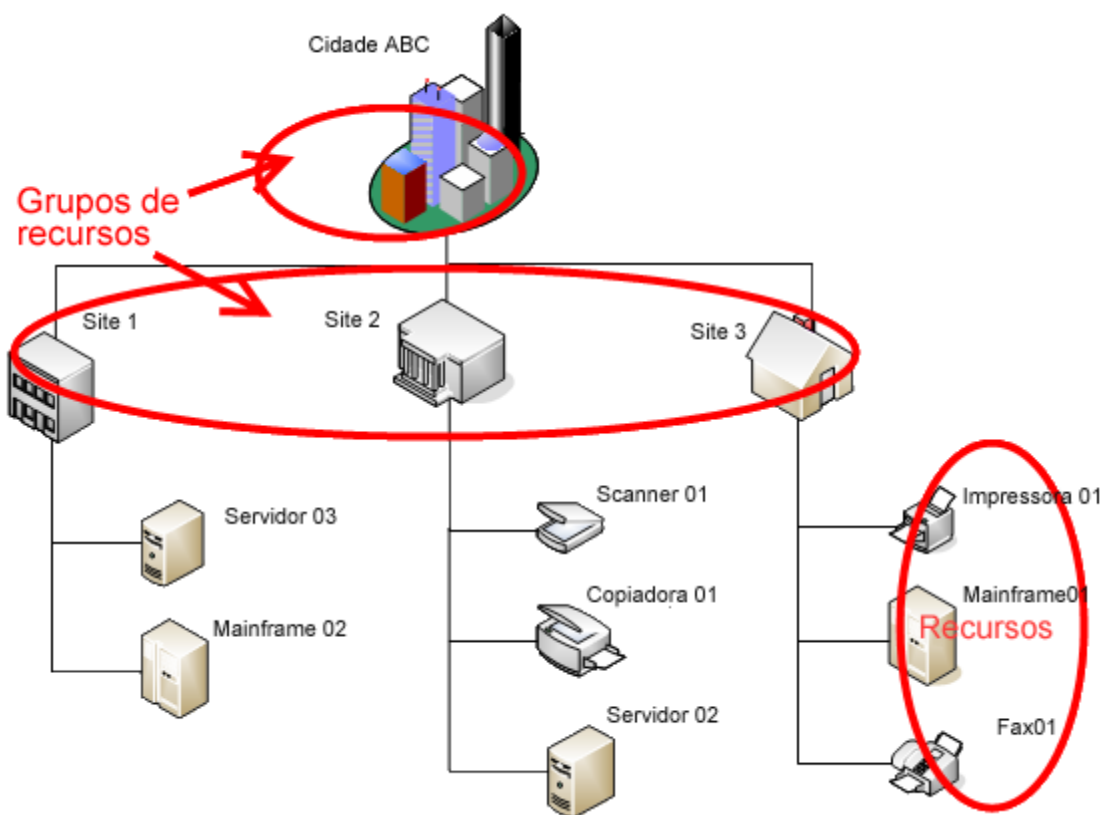
Nos casos em que esse método de registro é usado, a métrica calcula um resultado para cada um dos recursos do grupo de recursos no qual a métrica está agrupada.

-OU-

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

Nos casos em que esse método de registro é usado, a métrica calculará um resultado para cada um dos grupos de recursos contidos no grupo de recursos no qual a métrica está agrupada.

O agrupamento pode ocorrer em níveis diferentes, dependendo de como o modelo de recurso é criado. Com frequência, as organizações têm diferentes camadas de agrupamento que desejam representar. Por exemplo, em uma determinada cidade, pode haver uma série de locais, e dentro de cada local pode haver uma série de dispositivos de infraestrutura. (Impressoras, scanners, servidores, etc.) Usando esse tipo de agrupamento, você pode estruturar uma hierarquia do recurso definida que contém vários níveis e agrupamentos desses itens de hardware. (Supondo que um dispositivo de infraestrutura será o "recurso". A estrutura aqui descrita pode ter a seguinte aparência:



Como mostra o diagrama, há várias camadas de grupos. O grupo de nível superior "City ABC" contém três locais diferentes (que também são grupos de recursos). O grupo de recursos "Site 3 Resources" contém três recursos diferentes. Assim, no exemplo anterior, para agrupar a métrica nos três locais diferentes, você usaria o seguinte registro:

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

Nesse caso, Context.ClusterItem refere-se ao grupo de recursos chamado "City ABC Sites", que contém outros três grupos de recursos chamados "Site 01 Resources", "Site 2 Resources", etc., e pode ter a seguinte aparência na guia Agrupamento da métrica.

Geral	Detalhes	Notas	Agrupamento	Limites	Métricas relacionadas	Granularidade	Declaração de objetivo

☐ Esta métrica não está agrupada.

☒ Esta métrica está agrupada como segue:

O agrupamento foi aplicado aos itens de:

☒ De modo dinâmico

- ☒ Primeiro nível apenas: integrantes diretos
- ☐ Todos os níveis: inclui recursos apenas
- ☐ Todos os níveis: inclui recursos e grupos de recursos

☐ Estaticamente. Apenas os seguintes itens estão incluídos:

Observe também que o agrupamento é definido como dinâmico, pois isso inclui automaticamente quaisquer alterações no Grupo à medida que ocorrerem. O agrupamento estático pode ser útil para subconjuntos de grupos de recursos, ou quando você não deseja que o agrupamento mude ao longo do tempo.

Para criar uma métrica que relata sobre os recursos do grupo Site 3, use a seguinte declaração de registro:

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

Nesse caso, Context.ClusterItem refere-se aos recursos individuais, portanto, registrados pelo recurso apenas. A guia Agrupamento da métrica contém uma referência ao grupo "Sites 03 Resources".

Você pode configurar o agrupamento para operar em níveis diferentes da hierarquia em uma única métrica. Por exemplo, considerando a situação descrita no exemplo anterior e agrupando essa métrica entre o grupo "City ABC Sites" novamente. Você pode incluir em uma métrica os integrantes do recurso de diferentes níveis da hierarquia. Nesse caso, há três opções relacionadas aos recursos que estão incluídos nesse agrupamento:

- **Primeiro nível apenas: integrantes diretos:** igual aos antigos métodos de agrupamento descritos anteriormente.
- **Todos os níveis: inclui recursos apenas:** inclui todos os recursos contidos nos três grupos de recursos de locais em um único nível e calcula o nível de serviço deles individualmente.
- **Todos os níveis: inclui recursos e grupo de recursos:** inclui todos os recursos contidos nos três grupos de recursos de locais, bem como os três grupos de recursos, e calcula o nível de serviço deles individualmente.

Agentes

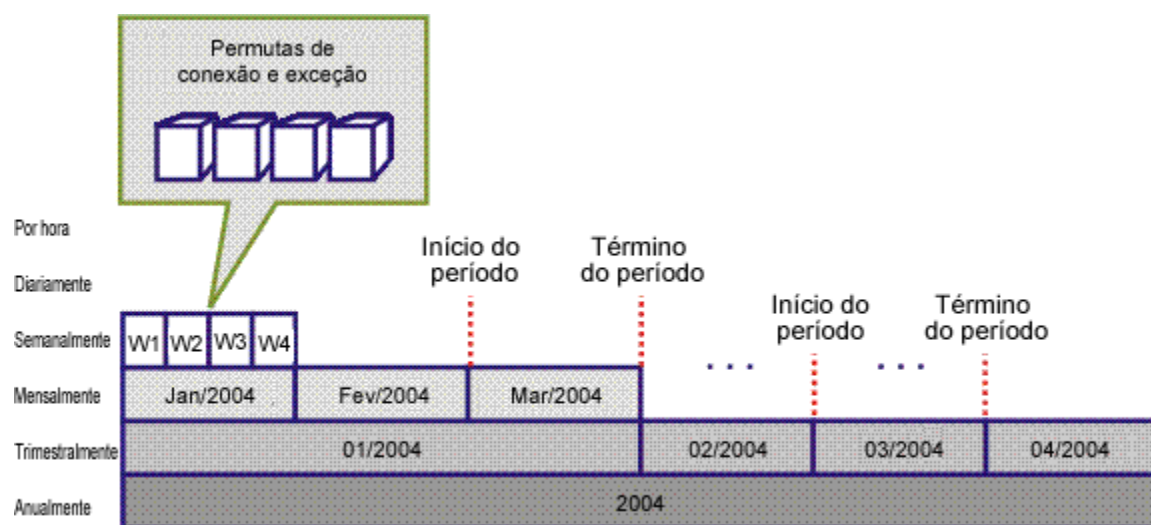
Cada métrica tem uma definição de um período de monitoramento. O período de monitoramento é o período durante o qual a métrica precisa gerar um resultado de nível de serviço, e é esse resultado que deve ser comparado com o destino definido. O resultado gerado para o período de monitoramento são os resultados comerciais; em outras palavras, é o resultado contratual com o qual o provedor se comprometeu em fornecer um determinado nível de serviço de destino. Cada instância da lógica de negócios para cada período é conhecida como um agente e está diretamente relacionada às granularidades associadas a cada métrica.

Por exemplo, se a métrica for definida com um período de monitoramento de um mês, será executada para fornecer um resultado mensal.

Para fornecer o recurso de detalhamento, no qual o usuário pode fazer uma busca detalhada nos resultados mensais para ver o resultado diário, a métrica deve ter uma definição de outras unidades de tempo nas quais deve ser calculada. As unidades de tempo são definidas na seção Detalhes gerais da métrica na guia Granularidade.

Para cada uma das unidades de tempo definidas na guia Granularidade da métrica e para o período de monitoramento, uma instância da lógica de negócios separada é executada pelo mecanismo. Cada uma dessas instâncias executa o mesmo código da lógica de negócios, mas OnPeriodStart e OnPeriodEnd serão acionados de maneira diferente para cada uma dessas instâncias. Para a instância diária, será acionado no início e no final do dia. Para cada unidade de tempo, será iniciado de acordo com os pontos de início e de término da unidade de tempo.

Cada uma das instâncias da lógica de negócios é executada para produzir o resultado relevante da unidade de tempo. Além disso, cada período deve ter um resultado que leva em consideração quaisquer exceções. Qualquer período que não fizer isso, (se as exceções estiverem definidas) deve permitir que o usuário escolha se deseja ver o resultado de nível de serviço com ou sem as exceções que foram consideradas. Devido a esse requisito, cada métrica potencialmente terá catorze agentes (instâncias) executadas pelo mecanismo: dois agentes para os resultados comerciais e doze para os seis períodos operacionais adicionais.



Escala de tempo da lógica de negócios

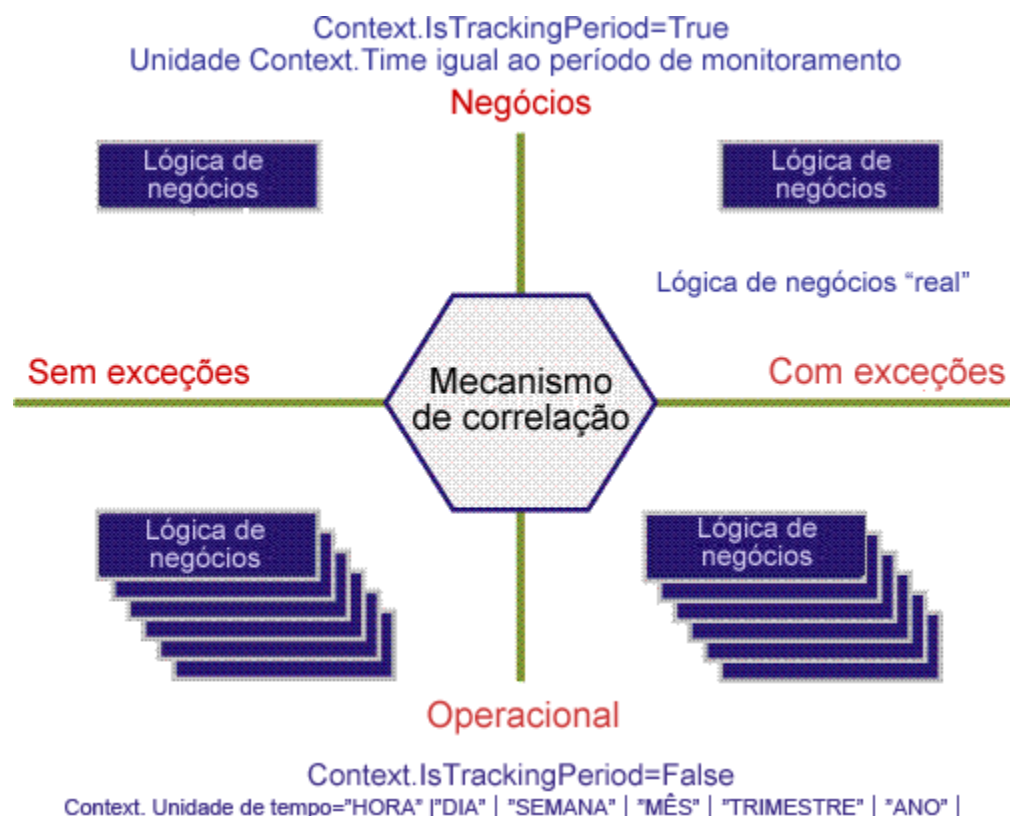
Isso significa que a definição de granularidade tem um sério impacto no desempenho do mecanismo de cálculo, pois cada período é calculado para um agente diferente. Portanto, nos casos em que o recurso de detalhamento não for necessário como um todo ou parcialmente, é recomendável que alguns dos agentes sejam desativados. Isso tem um impacto particularmente grande no caso das granularidades inferiores, como os resultados de hora em hora. Também tem um enorme impacto na métrica agrupada, pois o mecanismo faz todos os cálculos acima para cada ClusterItem que encontra. Na verdade, trata cada ClusterItem como uma nova métrica. Por exemplo, ao calcular uma métrica em um grupo de recursos de 50 itens, haverá 49 vezes mais trabalho para o mecanismo, em comparação com a mesma métrica não agrupada.

Por exemplo, se a métrica for definida para calcular o tempo de resolução em número de dias, um resultado de hora em hora não faz sentido e deve ser desmarcado na guia de granularidades para evitar que o mecanismo realize cálculos desnecessários.

O atributo TimeUnit do objeto de contexto (isto é, context.TimeUnit na lógica de negócios) retorna a unidade de tempo do agente em execução no momento, onde os possíveis valores retornados são: HORA, DIA, SEMANA, MÊS, TRIMESTRE, ANO.

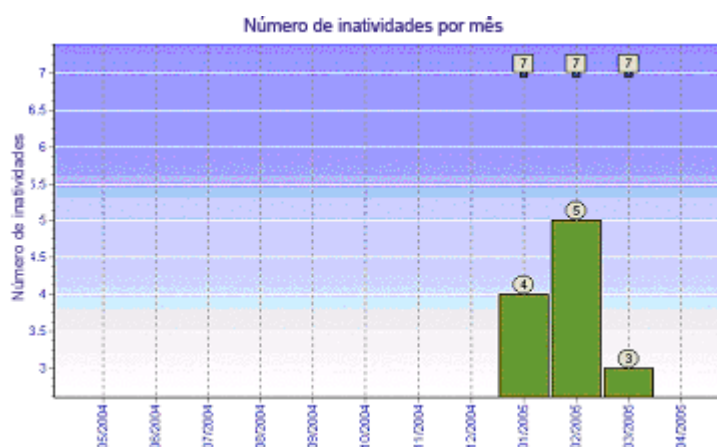
Por exemplo, para o agente diário, Context.TimeUnit retornará "DIA".

O atributo IsTrackingPeriod do objeto de contexto retornará Verdadeiro para o agente que calcula a unidade de tempo do período de monitoramento. É também importante observar que os agentes mostrados na guia Granularidade das métricas são adicionais para o agente do período de monitoramento da métrica. Portanto, até mesmo para uma métrica com um período de monitoramento mensal, você pode desativar o agente mensal e ela ainda calculará o nível de serviço mensal, mas apenas como "Resultados comerciais" (resultados não operacionais).

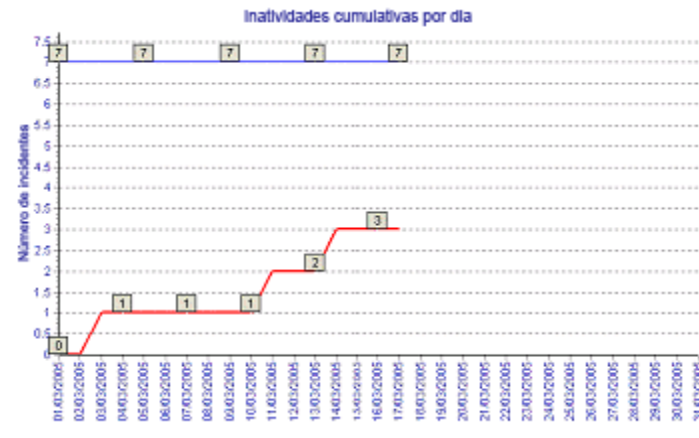


Conforme mencionado anteriormente, os vários agentes geralmente executam o mesmo código da lógica de negócios, mas existem casos em que é necessário aplicar uma lógica um pouco diferente.

Por exemplo, no caso mensal, o resultado será o número de downtime por cada mês separadamente, conforme mostrado a seguir:



Para o detalhamento diário, pode ser necessário que seja possível ver os downtimes cumulativos, onde cada dia é uma soma de todos os dias a partir do início do mês. Esse método deve ser aplicado a todas as unidades de tempo menores do que um único mês, conforme mostrado a seguir:



A diferença entre as duas unidades de tempo é que, para o agente que calcula o período de monitoramento, o contador de downtime será inicializado como 0 no início de cada período, mas para o agente diário, o contador será inicializado apenas caso o dia seja o primeiro dia do mês.

A seguir, está o manipulador de eventos OnPeriodStart:

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

O que é recálculo?

O recálculo é executado quando o mecanismo de correlação identifica que o resultado periódico final de uma métrica não é mais válido e que deve, portanto, recalcular os resultados.

O recálculo pode ser causado por:

- Recebimento de novos eventos que na realidade ocorreram no passado. (antes de o mecanismo ter calculado *até o momento* para essa métrica).
- Alteração de um recurso que está registrado para a métrica (nova data da versão e alterações feitas nele).
- Confirmação de uma nova versão do contrato que se sobrepõe ao tempo calculado anteriormente com alterações em algumas métricas (apenas métricas alteradas são recalculadas).

O método mais eficiente para registro é por parte contratual e serviço. Organizar os recursos em Parte contratual e Serviços é uma maneira de expressar o relacionamento lógico entre a camada de dados e a camada de negócios no sistema. Registrar os recursos por essas entidades não exigirá alterações nas fórmulas quando usadas em diferentes contratos ou para diferentes serviços. O contexto atual da métrica identificou o contrato e o serviço, que define a Parte contratual relevante e o Serviço associado. As fórmulas de lógicas de negócios definidas nesse tipo de registro são facilmente reutilizáveis, pois não precisam de alterações no registro.

Saídas - Tabelas de usuário

O script padrão de lógica de negócios não possui acesso a tabelas de saída externas. Existem apenas dois destinos de saída:

- A tabela de PSL (T_PSL), para a qual o mecanismo registra automaticamente os resultados de nível de serviço e onde o resultado especificado na função Resultado é gravado. Os valores de nível de serviço que são gravados nessa tabela podem ser apenas de um tipo numérico. Os resultados gravados na tabela T_PSL são os resultados que o Assistente de relatório relata. Não há nenhum controle sobre a estrutura ou o método por meio do qual esses resultados são gravados. Isso é realizado automaticamente pelo mecanismo de correlação.
- Tabela de saída de SLALOM (T_SLALOM_OUTPUTS). O registro nessa tabela é feito usando os métodos fornecidos pelo objeto de ferramentas da fórmula de lógica de negócios. Ao produzir informações para essa tabela, um nome de tabela lógica é fornecido, que é conhecido como tabela de usuário. Essas tabelas são usadas para produzir informações durante o cálculo de nível de serviço. Essas informações podem ser usadas posteriormente para gerar relatórios de formato livre. Pode haver muitas tabelas de usuário.

O uso da tabela externa T_SLALOM_OUTPUTS é obrigatória quando uma saída adicional é necessária além do resultado de nível de serviço periódico, quando a saída adicional não puder ser fornecida por meio da adição de outra métrica, ou quando a adição de outra métrica diminuir o desempenho do cálculo ao examinar o mesmo conjunto de registros, apenas para fornecer uma saída diferente.

Por exemplo, considere um caso no qual uma métrica está definida para calcular a porcentagem de tickets que foram resolvidos em menos de um dia e um relatório com a lista de todos os tickets que não puderam ser resolvidos em menos de um dia que deve ser gerado, é necessário que a fórmula gere cada ticket encontrado como uma falha para uma tabela externa, além de adicioná-lo às estatísticas de cálculo.

Com o requisito acima, a tabela normal de nível de serviço de saída não pode fornecer essa saída, pois:

- Os resultados de serviços são completamente numéricos.
- Apenas um único resultado de nível de serviço é possível para cada período.

Os registros são gravados nas tabelas de saída de usuário apenas para o agente que está em execução no período de monitoramento da métrica e calcula as exceções e as correções. Por exemplo, se a métrica for definida como tendo um período de monitoramento mensal, os parâmetros de saída (Tools.SaveRecord e Tools.SaveFields) NÃO serão executados quando o mecanismo estiver executando a fórmula para as outras granularidades, como HORA, DIA, SEMANA, TRIMESTRE e ANO.

Um outro benefício de ter essa tabela externamente é que ela pode ser usada para diversos requisitos de geração de relatórios. Outros requisitos de geração de relatórios típicos podem ser gerados a partir dessas tabelas além dos requisitos contratuais. Por exemplo, uma métrica que calcula o número de tickets fechados em um mês também pode calcular o tempo de resolução do ticket e produzir todas essas informações para a tabela de saída. Isso permitirá uma análise mais detalhada dos tickets individuais que foram fechados durante o período, juntamente com detalhes adicionais que possam ser necessários em um requisito de geração de relatórios separado.

As informações nessas tabelas também são gerenciadas pelo mecanismo de recálculo do mecanismo. Durante esse processo de recálculo, os registros são marcados como inativos, e uma nova linha é criada. Esse é um ponto importante que deve ser observado, pois todas as instruções SQL de relatório de formato livre precisam incluir uma marca no campo IS_ACTIVE na tabela T_SLALOM_OUTPUTS (onde is_active = 1) já que apenas esses registros são atuais.

Observação: ao executar o escopo de lógica de negócios durante o processo de depuração de fórmulas, as mensagens são gravadas na tabela T_DEBUG_SLALOM_OUTPUTS em vez da T_SLALOM_OUTPUTS.

Ao documentar dados usando T_SLALOM_OUTPUTS, os dados inseridos são sempre texto (pois os campos de T_SLALOM_OUTPUTS são varchar2). Portanto, os valores de data são convertidos em texto ao aplicar o formato do sistema operacional que pode ser alterado durante o ciclo de vida do aplicativo. Portanto, T_SLALOM_OUTPUTS poderá ter inconsistências nos formatos de data. Além disso, a lógica de negócios lida com datas em UTC, sendo que pode-se esperar que T_SLALOM_OUTPUTS tenha data e hora local, por isso, em alguns casos, pode ser necessário usar a função de conversão *Tools.GetLocaleDate(date)* para solucionar esse problema.

A seguinte função converte datas em horas locais e mantém a consistência do formato de data ao formatar as datas para "dd/mm/aaaa hh24:mi:ss":

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

Há dois métodos que podem ser usados para gravar informações na tabela externa a partir da fórmula de lógica de negócios, que são:

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

Esses dois métodos do objeto de ferramentas são descritos abaixo em mais detalhes:
Tools.SaveRecord tableName, key,[val1],[val2],...

Esse método salva um registro em uma tabela chamada T_SLALOM_OUTPUTS. O parâmetro tableName especifica a tabela (virtual) em T_SLALOM_OUTPUTS na qual as informações devem ser gravadas. Cada registro na tabela de usuário tem uma chave exclusiva, que especifica o registro no qual as informações devem ser gravadas. Cada registro também tem até 20 campos de valor de texto. O método SaveRecord recebe o nome de uma tabela de usuário e uma chave. Ele também aceita todos os campos de valor na tabela de usuário. (Esses parâmetros de valor são opcionais e podem ser omitidos.) Se um registro com a mesma chave já existir, ela será atualizada. (Somente os campos de valor transferidos como parâmetros são atualizados.) Se um registro com esta chave não existir, ela será criada.

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

Esse método é semelhante a SaveRecord, exceto pelo fato de que, em vez de enumerar todos os valores, ele fornece pares de nomes de campo e os valores de campo relacionados. Os números de campo podem ser substituídos por nomes de campo. Os nomes de campo devem ser definidos previamente e manualmente na tabela T_SO_FIELD_NAMES. Essa tabela é usada para registrar a estrutura das tabelas de saída.

É recomendável que o Especialista em lógica de negócios defina a estrutura da tabela de saída antes de gravar na T_SLALOM_OUTPUTS, pois depois a estrutura e o significado dos campos já estarão bem definidos e farão com que a consulta seja muito mais simples.

A estrutura da tabela é:

- **table_name** - cada tabela lógica tem um nome exclusivo.
- **field_name** - cada campo em uma tabela é exclusivo.
- **field_id** - cada campo é numerado com um número de série, começando em 1.

Usar o método SaveFields é preferível, pois mantém uma documentação da estrutura e o significado dos valores inseridos.

Exemplo:

Considere um caso no qual seja necessário gerar uma lista de todos os incidentes com o tempo de resolução mais alto do que um limite definido, juntamente com os resultados da métrica que calculam a porcentagem de tickets que foram resolvidos abaixo desse limite. A gravação nas tabelas de saída será feita no procedimento do manipulador de eventos OnXXXEvent após a validação do limite.

Isso é ilustrado no exemplo a seguir:

```
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
```

Algumas sugestões relacionadas ao uso das tabelas T_SLALOM_OUTPUTS são:

- É recomendável gravar apenas os dados necessários nas tabelas de saída (não mais).
- Grave nas tabelas de saída apenas as granularidades da métrica que forem necessárias.
- Todos os campos de valor têm apenas 50 caracteres de tamanho por padrão (exceto a coluna VAL_1, que tem 512). Talvez seja necessário reduzir alguns valores para os campos para garantir que eles irão se ajustar a esse tamanho dos dados, uma vez que, caso contrário, você receberia erros de tempo de execução na lógica de negócios.
- Há apenas 20 colunas disponíveis para os dados (VAL_1 ... VAL_20).

Observação: gravar nas tabelas de saída pode ter influência sobre o desempenho do mecanismo, já que a gravação em uma tabela exige muito do computador, comparado a um cálculo na memória. Portanto, considere cuidadosamente se é necessário usar essa funcionalidade e onde ela é necessária e, em seguida, reduza o número de vezes que as tabelas são acessadas.

Gerando relatórios com as informações das tabelas definidas pelo usuário

Não é possível usar o Assistente de relatório padrão do CA Business Service Insight para gerar relatórios com as informações gravadas nas tabelas de saída. Para gerar relatórios com essas informações, é necessário criar um relatório de formato livre.

Essencialmente, isso significa criar uma consulta SQL em cima dessa tabela. Como essa tabela contém muitas tabelas lógicas que recebem gravações de várias fórmulas, é recomendável que alguém familiarizado com SQL (Especialista em fonte de dados) crie uma exibição na T_SLALOM_OUTPUTS para facilitar a diferenciação entre as várias tabelas lógicas contidas dentro dela, e também para garantir que as informações serão recuperadas conforme planejado.

A definição da exibição já terá todas as conversões dos tipos de campo de texto para o tipo de informações relevantes e também manterá todas as filtragens necessárias (tabela lógica, registros ativos, métrica relevante, etc.).

A seguir, há um exemplo da criação da exibição:


```
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'
```

Criar módulos de lógica de negócios

Os usuários podem definir módulos de lógica de negócios independentes que podem ser usados por vários objetivos do nível de serviço (métricas). Para aplicar alterações na lógica de negócios em todo o sistema, o usuário altera o componente de lógica "base" que depois pode ser aplicado em todos os SLAs relevantes com um único clique.

Um módulo de lógica de negócios é um componente de código no qual a lógica de negócios pode ser definida e facilmente mantida, e que reduz a redundância. Um único módulo pode ser usado por várias métricas.

Durante a etapa de configuração, as fórmulas são configuradas para definir os principais módulos de lógica de negócios. (Consulte o capítulo de design: Modelos e módulos de lógica de negócios.)

Há três tipos de módulos de lógica de negócios:

- **Lógica de negócios completa** - usada quando uma fórmula completa precisa ser usada como um módulo. Os métodos Resultado e OnRegistration devem ser implementados no script do módulo.
- **Classe** - módulo que contém uma definição de uma classe VB única.
- **Biblioteca** - módulo que contém qualquer conjunto de procedimentos, funções e classes.

Os módulos podem ser usados em qualquer um dos seguintes:

- Métricas
- Outros módulos
- Modelos de lógica de negócios
- Métricas dentro de modelos de contrato e de nível de serviço.

Módulos podem usar parâmetros que são acionados dos parâmetros de contexto da métrica ("ParamName").

Observação: para evitar erros de tempo de execução, sempre defina um valor padrão ao usar parâmetros em módulos de lógica de negócios. A fórmula gera uma mensagem de erro de log para parâmetros não existentes.

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

Exemplo do módulo de lógica de negócios

Há um objeto de nível de serviço descrito como "atividade da central de ajuda dentro do limite". O sistema de central de ajuda a seguir tem um ciclo de vida de ticket com o status de:

- Aberta
- Atribuído
- Fechado.

Duas das métricas que podem ser definidas para descrever o desempenho da central de ajuda são:

- **Nome da métrica - resolução de ticket a tempo com êxito.**

Declaração de objetivo - não menos que 99% dos tickets devem ser resolvidos em 4 horas.

Lógica de negócios - a resolução deve ser calculada de Aberto para Fechado.

- **Nome da métrica - atribuição de ticket a tempo com êxito.**

Declaração de objetivo - não menos que 99% dos tickets devem ser atribuídos em 30 minutos.

Lógica de negócios - a atribuição deve ser calculada de Aberto para Atribuído.

Como a mesma lógica pode ser identificada para essas duas métricas, um módulo pode ser criado para atender às duas métricas.

O módulo requer os seguintes parâmetros dentro do contexto da métrica:

- Primeiro status
- Segundo status
- Limite

Quando o módulo de lógica de negócios está criado, as métricas definidas podem usar um módulo como parte da definição.

Em seguida, a lógica pode ser alterada. Por exemplo, um novo status "Cliente Pendente" deve ser considerado. O status Cliente Pendente é definido para um ticket quando a central de ajuda está aguardando informações adicionais sobre o ticket do cliente. Dentro da lógica de negócios, já que o período em que o ticket está em "cliente pendente" não deverá ser considerado parte do cálculo. Portanto, apenas o módulo de lógica de negócios precisa ser alterado para considerar o novo status e lógica. Uma nova versão do módulo, incluindo a nova lógica, é criada.

Ao confirmar a alteração, você é apresentado a uma lista de métricas que usam o módulo. Você pode aplicar a alteração a todas as métricas, coletivamente, ou optar por aplicar a alteração apenas a métricas específicas na lista.

Se você selecionar apenas métricas específicas da lista, será solicitado a criar um novo módulo para as métricas selecionadas. O antigo módulo usado pelas métricas selecionadas é substituído pelo novo módulo de lógica de negócios, e o recálculo é feito usando a nova lógica.

Criar parâmetros

Os parâmetros fornecem aos usuários corporativos uma maneira rápida e intuitiva de exibir e alterar seus valores por meio da interface gráfica de usuário (GUI), sem precisar editar o script de fórmula.

O uso de parâmetros na lógica de negócios permite a criação de fórmulas gerais que podem ter um amplo uso em todo o sistema e otimiza o uso de módulos.

Os parâmetros podem ser definidos em nível de contrato ou de métrica. Os parâmetros em nível de métrica são exibidos e configurados na guia Declaração de objetivo dos Detalhes da métrica. A lógica de negócios tem acesso apenas aos parâmetros em nível de métrica; portanto, para acessar um parâmetro de contrato em uma métrica, um tipo diferente de parâmetro é criado localmente na métrica. É chamado de parâmetro dinâmico e obtém seu valor como uma referência dos parâmetros em nível de contrato. Os valores de referência permitidos no parâmetro dinâmico são apenas aqueles definidos no contrato pai da métrica.

Tipos de parâmetros:

- Data - na janela restrita (data + hora).
- Número - tamanho máximo de 15 dígitos com uma precisão máxima de 5 dígitos.
- Texto - tamanho máximo de 100 caracteres.
- Tabela - tamanho máximo de 120 entradas.

Para acessar os valores do parâmetro no código da fórmula, é necessário usar o objeto de parâmetros e fazer referência ao nome do parâmetro.

Exemplo:

`Parameters("Threshold")`

(Observe que esse é um método resumido de chamar o valor - normalmente é feito como `Parameters.Item("Threshold").`)

Ou, para um parâmetro de tipo de tabela:

```
Parameters("Table")("Word")("Price")
```

(Onde os valores "Word" e "Price" são os nome de linha e coluna do parâmetro de tabela, respectivamente.)

Os parâmetros de tabela só devem ser usados após uma série de pontos principais:

1. Definir uma variável global (por exemplo, `dim myTableParam`).
2. Na função `OnLoad`, preencher a variável do objeto de parâmetros (por exemplo, `"Set myTableParam = Parameters("MyParametersTable")`).
3. Desse ponto em diante, use apenas o objeto criado, `myTableParam`. O parâmetro em si não deve ser usado fora da função `OnLoad`, e você só deve fazer referência ao objeto de variável global criado a partir dele.
(por exemplo, `"dim myVal: myVal = myTableParam ("myRow")("myColumn")`).

Isso libera memória que o parâmetro utiliza e evita que o mecanismo crie o mapa do parâmetro em cada chamada de parâmetro e para cada "OnXXXEvent", que pode ser chamado milhares de vezes por métrica.

O código a seguir ilustra o uso correto de um parâmetro de tabela:

```
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
```

Os seguintes métodos estão disponíveis para cada objeto de parâmetro criado no código.

Parâmetro	Descrição
IsExist	O parâmetro existe. Não é possível determinar a existência de parâmetros de contrato.
IsNumeric	Parâmetro de número.
IsText	Parâmetro de texto.
IsDate	Parâmetro de data.
IsTable	Parâmetro de tabela.
IsEntryExist	Há uma entrada na tabela.
IsEntryNumeric	Entrada numérica na tabela.
IsEntryText	Entrada de texto na tabela.
IsEntryDate	Entrada de data na tabela.
Dump	Retorna uma lista de todos os parâmetros.
Item	Refere-se ao valor do parâmetro.

Implementar destinos dinâmicos

Destinos dinâmicos são manipulados pela lógica de negócios, usando um manipulador de eventos do script padrão de lógica de negócios, semelhante à função Result(), que é usada para retornar o valor do nível de serviço da métrica. O destino dinâmico deve ser especificado na guia Detalhes das métricas, conforme mostrado abaixo.

Geral	Detalhes	Notas	Agrupamento	Limites	Métricas relacionadas	Granularidade	Declaração de objetivo	Registro
Indicador principal: Serviço: * Eagle App Domínio de serviço: * Service Availability → Categoria de domínio: * % of time available Unidade de medida: * Porcentagem Período de atividade: * Business Hours (Weekly) (Semanal) Fuso horário: * GMT - Sun Objetivo do nível de serviço: * Valor do objetivo não mais que 1 Semanas Porcentagem Destino dinâmico: ☒ Período de monitoramento: * 1 Semanas								

Quando um destino dinâmico está especificado, o destino é obtido da função Target() na lógica de negócios, em vez do valor estático especificado na guia Detalhes da métrica. A função de destino é semelhante à seguinte.

Objetivo da função

```
'TODO: ADD code here TO handle dynamic target calculation
Target = Null
End Function
```

Essa função deve ser implementada com base no requisito para a métrica, de modo a retornar o valor do destino desejado para um período específico. A função pode retornar qualquer número que a lógica de negócios possa atribuir a ela.

Um exemplo real de destinos dinâmicos

Para uma central de atendimento, o destino para uma métrica que mede o "*Tempo médio de atendimento de chamadas*" pode depender do volume de chamadas. Se houver entre 0 e 800 chamadas, o destino deve ser < que 15 segundos; se houver entre 801 e 1500 chamadas, o destino deve ser < que 20 segundos; com mais de 1500 chamadas, o destino deve ser menor que 25 segundos. Isso pode ser implementado da seguinte maneira: (presumindo que TotalCalls é um contador incrementado para cada evento de chamada recebido e que TotalCalls não pode ser menor que 0).

Objetivo da função

```
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
```

Outro exemplo de como um destino dinâmico pode ser usado

Considere a situação em que o destino para uma métrica pode ser alterado dependendo da granularidade do cálculo. Esse pode ser o caso se houver um destino diário de 98% de disponibilidade para um grupo de servidores, mas o destino mensal for de 99,5% de disponibilidade. A solução para isso exige o uso da função do destino dinâmico em conjunto com a chamada de função para *Context.TimeUnit* para determinar o agente atual que está sendo calculado. Portanto, você pode ajustar o destino de maneira apropriada.

Objetivo da função

```
If Context.TimeUnit = "DAY" Then
    Target = 98
ElseIf
    Target = 99,5
End If
End Function
```

Fazer backup de estados

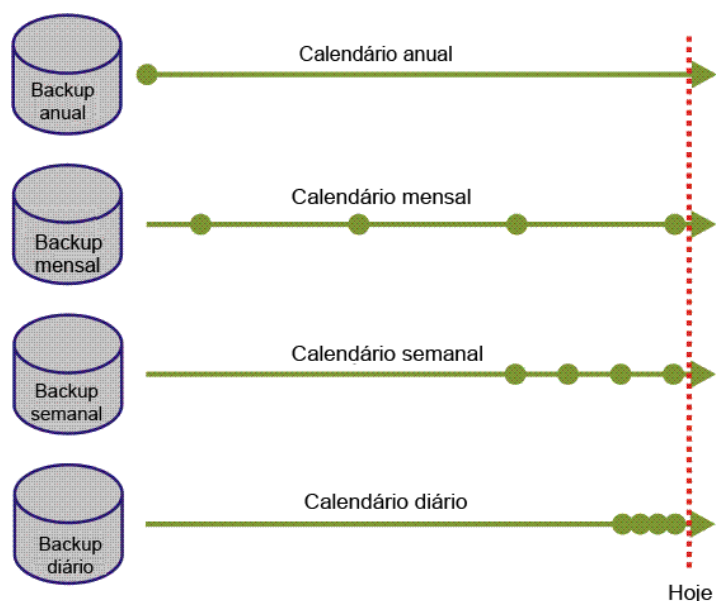
Durante o processo contínuo de calcular os níveis de serviço para cada uma das métricas, o mecanismo é frequentemente forçado a executar um cálculo parcial para um período que ainda não foi concluído. Para garantir que não precisará voltar ao início do cálculo quando novos dados chegarem ao longo do tempo, ele executa um tipo de backup de seu "estado" atual antes de passar para a próxima tarefa de cálculo. Nesse momento, obtém um instantâneo das variáveis e valores atuais naquele ponto do cálculo e salva esse "estado" no banco de dados.

O processo de backup da lógica de negócios é um mecanismo por meio do qual o código da lógica de negócios, incluindo os valores das variáveis, é codificado em um fluxo binário e salvo no banco de dados. Esse mecanismo também é necessário para acelerar o desempenho do mecanismo de cálculo em casos de recálculos. O estado é submetido a backup periodicamente e usado em recálculos e como uma medida de eficácia para cálculos contínuos.

Por exemplo, se um recálculo for necessário para o mês de forma retroativa, em vez de recalcular os resultados desde o início do contrato, o estado de backup mais próximo antes da data de recálculo é usado, e os cálculos são feitos a partir desse estado.

O mecanismo de cálculo usa heurística predefinida para determinar quando o backup é necessário, e usa os recursos de backup para armazenar o estado codificado no banco de dados.

Na figura a seguir, os pontos em vermelho representam um estado de backup. Quanto mais antiga for a consideração, menor será o número de estados submetidos a backup que são considerados. A lógica por trás desse mecanismo é a premissa de que o recálculo geralmente é necessário para o período inferior a um mês.



Otimizar o sistema para recálculo

O processo de cálculo de nível de serviço exige uma quantidade significativa de recursos de CPU, memória e armazenamento. A seguir, há uma lista de recomendações que podem reduzir a carga do computador e aumentar a velocidade de cálculo.

Observação: algumas dessas recomendações exigem configurações internas não disponíveis na interface com o usuário. Nesses casos, entre em contato com o suporte técnico da CA para obter mais detalhes e instruções.

- **Alterar os estados salvando a configuração**

Dependendo dos atrasos conhecidos do conector, você pode definir os parâmetros dos estados para melhor atender aos seus requisitos. Isso significa que você poderá alterar o número e a granularidade dos pontos salvos dos estados.

- **Configurar o sistema para calcular apenas as unidades de tempo que são realmente necessárias**

Métricas podem ter até sete períodos de granularidades diferentes: *ano*, *trimestre*, *mês*, *semana*, *dia*, *hora* e período de monitoramento. Em algumas implementações, nem todas as granularidades são necessárias. Você pode desativar as granularidades desnecessárias para contratos não confirmados por meio da interface com o usuário. Consulte a guia de granularidades de cada métrica. Para alterar as granularidades da métrica de contratos confirmados, entre em contato com o suporte técnico da CA.

Observação: evite o cálculo de períodos operacionais que forem semelhantes ao período de monitoramento.

- **Não calcular os valores "sem correções" ou "sem exceções"**

Por padrão, o mecanismo de cálculo calcula quatro valores diferentes: *fornecido*, *fornecido com correções*, *fornecido com exceções* e *fornecido com correções e exceções*. É possível alterá-lo para calcular apenas o valor fornecido.

Observação: para obter mais informações, entre em contato com o suporte técnico da CA.

- **Alterar a ordem de cálculo**

A ordem de cálculo PSL das métricas pode ser priorizada para iniciar com suas métricas mais importantes e, em seguida, continuar nas outras métricas.

Observação: para obter mais informações, entre em contato com o suporte técnico da CA.

- **Criar mais de uma instância PslWriter**

Ao criar mais de uma instância PSL (do mecanismo de cálculo), você poderá dividir a carga de trabalho e aumentar a velocidade de cálculo.

Observação: para obter mais informações, entre em contato com o suporte técnico da CA.

- **Planejar implementações para reduzir o tempo de recálculo**

Para otimizar o tempo de recálculo, você pode:

- Executar o conector com mais frequência para reduzir o número de eventos atrasados e evitar backlogs grandes de dados para o mecanismo processar.
- Desativar agentes não utilizados na guia Granularidade das métricas.
- Duplicar as métricas e calcular diferentes agentes usando a mesma métrica, para equilibrar a carga de cálculo
- Usar métricas intermediárias para executar cálculos comuns e compartilhar os resultados com todas as outras métricas que exigem os mesmos dados.

■ **Planejar implementações para reduzir a quantidade de dados**

Para otimizar a quantidade de dados, use o conector para carregar apenas dados agregados (processados). Agregar as informações da fonte de dados antes que sejam enviadas à tabela de dados brutos do CA Business Service Insight aumenta a eficácia da leitura da entrada PSL.

■ **Seguir as recomendações de configurações de PSL do CA Business Service Insight**

O mecanismo PSL pode ser reconfigurado para melhorar o desempenho de acordo com o ambiente de aplicativo específico e os requisitos. Alguns dos parâmetros podem ser definidos na interface com o usuário e outros podem ser definidos apenas na tabela de configuração do sistema do banco de dados.

Observação: consulte as recomendações de suporte para suas configurações de PSL.

Logs e alertas

Há casos em que a lógica de negócios é necessária para enviar relatórios ao log ou acionar uma mensagem de alerta. Isso é necessário quando as mensagens devem ser enviadas com base no processamento de eventos. Quaisquer informações coletadas durante o processo de cálculo e que podem ser valiosas podem ser enviadas como um alerta. Por exemplo, uma mensagem de alerta pode ser enviada quando um evento específico estiver dentro do limite de tempo de resolução especificado ou na análise de tendências quando um determinado número de falhas consecutivas tiver sido atingido.

"Out" é um objeto de lógica de negócios global que permite que a fórmula envie alertas, bem como mensagens de log. Tem dois métodos associados a ele com o seguinte formato:

`Alert(<Event type>, <Resource name>, <value1, value2>, ...<value16>)`

Esse comando envia um alerta de um tipo de evento especificado. No entanto, esse tipo de evento deve ser criado manualmente para esse alerta. O número de valores e o seus tipos devem corresponder à definição do evento.

`Log(<Message>, <Level>)`

Esse comando envia uma mensagem ao log do sistema. O primeiro parâmetro é a mensagem informativa relatada e pode ter texto livre. Você também pode anexar os valores de variáveis a esse texto para dar significado contextual à mensagem. O parâmetro "Nível" pode assumir um dos seguintes valores:

Descrição

Uma mensagem de aviso é relatada.

Uma mensagem de erro é relatada.

Uma mensagem informativa é relatada somente ao executar no escopo de lógica de negócios. Ao executar no `PsIWriter`, nenhuma mensagem será relatada. Esse é o nível padrão. É usado principalmente para fins de depuração.

Exemplo:

O exemplo a seguir foi obtido de um caso em que informações de infraestrutura do evento eram esperadas antes dos detalhes reais do incidente. Um mecanismo de alerta foi configurado para notificar o administrador dessa condição para solicitar a correção do problema.

`Out.Alert "Site Unknown Alert", Context.ClusterItem, Context.Rule`

```
Out.Log("Fault Event Received for a Site with no infrastructure details: " &  
Context.ClusterItem)
```

Comentários de causa raiz da violação e comentários de evento

O comentário de causa raiz da violação pode ser definido na lógica de negócios para explicar os resultados de nível de serviço. Os comentários de causa raiz da violação são associados às métricas.

Também é possível gerar anotações do evento que são comentários associados aos eventos de dados brutos, em vez de associados à métrica. Esses dois tipos de comentários podem ser visualizados nos dados do relatório.

Dois métodos no objeto de ferramentas da lógica de negócios permitem a criação de causa raiz da violação e registros de anotações do evento:

- **Tools.AddRootCauseComment** (Texto, Data e hora, [resourceId])
- Salva um comentário de causa raiz. Essa informação pode ser útil posteriormente na geração de relatórios. O comentário de causa raiz salvo explica uma situação específica durante a execução da fórmula de lógica de negócios em um momento específico. O parâmetro de informações especifica o comentário que deve ser gravado. O método recebe uma data e hora a serem salvas junto com o comentário. Ele também aceita um parâmetro ResourceId, especificando o recurso relacionado ao contexto do método. (Esse parâmetro é opcional e pode ser omitido).
- **Tools.AddEventAnnotation** (EventId, Texto)
- Esses métodos podem ser usados em qualquer lugar da lógica de negócios, e o contexto do local onde se aplicam precisa ser levado em consideração. Adicionar um comentário de causa raiz pode ser mais relevante no final de um período de cálculo, quando o motivo para o nível de serviço ser menor do que o esperado para esse período é conhecido. Vamos supor que, por exemplo, durante o período de um mês ocorreram quatro interrupções, todas causadas por um único dispositivo. O comentário de causa raiz pode ser compilado a partir de algumas dessas informações sobre as interrupções, de modo que quando os relatórios forem exibidos para esse período, será possível ver o que contribuiu para qualquer violação de nível de serviço durante esse tempo. O comando AddRootCauseComment é mais adequado para a rotina do manipulador de eventos OnPeriodEnd ou outra função similar que é executada no final do período que está sendo calculado.

- Adicionar uma anotação do evento, no entanto, é mais adequado para o processamento de eventos de dados brutos e favorece seu uso para o OnXXXEvent (o manipulador de eventos personalizado especificado na declaração do registro). Nesse manipulador de eventos, todos os campos específicos do evento em questão estão disponíveis por meio do objeto eventDetails.

- Veja um exemplo de como isso pode ser na lógica de negócios:

```
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
```

Separando exceções de períodos de atividade

A lógica de negócios do CA Business Service Insight não recebe eventos de exceção. O que recebe é OnTimeslotExit quando um período de exceção começa e OnTimeslotEnter quando um período de exceção termina. A lógica de negócios, portanto, não consegue distinguir entre períodos da exceção e tempos limites de períodos de atividade. Além disso, não consegue distinguir entre tipos de exceção. Como resultado, não é possível implementar uma lógica diferente para o comportamento do período da exceção e para o comportamento "fora do período de atividade".

Uma maneira de implementar exceções especiais (isto é, uma exceção que não se comporta como "fora do período de atividade") é definir tipos de eventos dedicados, em vez de usar o mecanismo integrado do CA Business Service Insight para a manipulação de exceções. Esses eventos são gerados por meio da leitura de uma fonte de dados dedicada, usando um conector.

Uma planilha do Excel (ou qualquer outra fonte de dados) pode armazenar essas exceções, e um conector pode carregar os dados e gerar uma resposta: eventos de entrada da exceção e de saída da exceção. Como alternativa, as exceções podem ser adicionadas usando correções. Além da correção, um recurso fictício deve ser definido e associado a esses eventos para fins de registro. Esse recurso não tem outra finalidade que não seja a de um espaço reservado que é necessário para o comando.

Para ser capaz de manipular os períodos da exceção relatados por esses eventos dedicados, a fórmula de lógica de negócios deve registrar esses eventos de exceção, além de fazer o registro normalmente necessário para eventos de dados brutos a serem usados no cálculo.

É recomendável que o Especialista em lógica de negócios inclua um campo para o tipo de exceção no tipo de evento, permitindo manipulações diferentes de vários tipos de exceções especiais.

Essa abordagem tem as seguintes características:

- Separa dois conjuntos de exceções - padrão e especial.
- Exceções especiais não têm uma interface com o usuário para manutenção.
- Os relatórios com base em exceções geradas como eventos por um conector não comentam sobre sua existência. Nos casos em que o mecanismo de correção tiver sido usado, haverá um comentário. O uso do método de correção é recomendável manter a integridade dos resultados produzidos pelo sistema.
- Nenhuma especificação "Com/Sem exceções" considera essas exceções.
- Quando o mecanismo de correção for usado, Com/Sem correção se aplica.

Quando implementada, é recomendável que o Especialista em lógica de negócios aplique a lógica para todas as métricas do sistema.

Há outro método de aplicar uma exceção em um único recurso, se necessário. Esse método envolve o uso do status "Vigente" dos recursos. Definir o status de um recurso como "Não está vigente" significa que, durante esse período, o mecanismo de cálculo irá ignorar todos os eventos de dados brutos que são enviados para esse recurso. Defina um período de tempo em que o recurso não está vigente criando novas versões do recurso, uma no início do período da exceção e outra no final do período da exceção.

No entanto, se o recurso fizer parte de uma métrica agrupada e o recurso estiver se tornando vigente ou não estiver vigente no mesmo período de cálculo, apenas o último período no qual o recurso estava vigente será levado em consideração no resultado, conforme definido acima. Nesse caso, é recomendável usar o recurso de atributos personalizados. Um atributo adicional para o recurso que indica o status do recurso pode ser gerenciado, e a fórmula de lógica de negócios irá consultar o status do recurso em cada lugar relevante no script.

Considerações de memória e desempenho

A seguir, há uma série de situações que devem ser consideradas ao desenvolver soluções de lógica de negócios. As situações descritas são casos em que o desempenho do mecanismo de cálculo poderá ser afetado negativamente:

- **Parâmetros**

Se um valor de um parâmetro for exigido no código, a criação de uma variável global para atribuir o valor do parâmetro a ela é recomendado. Além disso, sempre que o valor do parâmetro for exigido, use a variável global. Isso evita a situação na qual o mecanismo cria o mapa dos parâmetros para cada chamada de parâmetro.

- **Uso de mapas em métricas agrupadas**

Objetos grandes de mapa global na lógica de negócios para métricas agrupadas devem ser usados somente com muita atenção. Enquanto o mecanismo estiver calculando uma métrica agrupada, estará ocupado carregando as variáveis globais dos estados anteriores para cada item no agrupamento separadamente.

- **Registro da lógica de negócios**

Filtrar os eventos de dados brutos exclusivamente pelos métodos de registro é recomendado. Adicionar a filtragem interna usando uma instrução "if" no código resultará em mais tempo de processamento. Mais importante, despesas adicionais serão exigidas pelo mecanismo para a recuperação e o processamento dos registros de dados brutos que não são necessários.

- **Evitando o uso de `Dispatcher.RegisterByEventType`**

Melhora o desempenho. Usar esse método de registro significa que você está registrando em todos os recursos do sistema, e não apenas nos recursos que têm eventos desse tipo específico. Portanto, todas as alterações no recurso afetam os cálculos de métrica. Outra desvantagem de usar esse método de registro ocorre no tempo de execução da métrica, quando ela acessa os dados brutos. Em seguida, ela precisa filtrar nos dados brutos apenas os eventos com o tipo de evento específico e ignorar os outros eventos.

- **`Dispatcher.Register`**

Quando você estiver usando `Dispatcher.Register`, *sempre* verifique se especificou o 3º parâmetro. Registrar sem o 3º parâmetro é exatamente como fazer registros por tipo de evento (`Dispatcher.RegisterByEventType`). Em outras palavras, verifique se usou pelo menos um outro parâmetro além dos dois primeiros.

- **Granularidade de cálculo**

É importante ativar somente os agentes necessários para fins de cálculo e detalhamento. O cálculo das unidades de tempo de todos os agentes faz uso intensivo do processador.

Como ativar os contratos (Gerente de contratos)

A ativação de um contrato é feita ao confirmá-lo. (Para obter mais detalhes, consulte a Ajuda Online.)

A ativação do contrato aciona o mecanismo para iniciar os cálculos das métricas do contrato e começar a produzir resultados para o contrato.

Antes de ativar o contrato, verifique se todas as condições a seguir foram atendidas:

- Todas as métricas devem ter a lógica de negócios definida para elas (na métrica ou como um módulo vinculado) que foi testada e não contém erros.
- Todas as métricas têm um limite do painel definido. (Para obter mais detalhes, consulte a Ajuda Online.) É importante que os limites já estejam definidos para que, durante o processo de cálculo, o painel possa avaliar os limites para a métrica.
- As datas de efetivação estão de acordo com o período correto e com os registros de dados brutos disponíveis. Verifique também se os resultados são os esperados, usando o escopo de lógica de negócios.

Depois que o contrato tiver sido confirmado, verifique se a ativação foi iniciada com êxito e se os cálculos estão ocorrendo como esperado.

Siga estas etapas:

1. Verifique se todos os componentes de serviço do CA Business Service Insight foram iniciados (principalmente o mecanismo de cálculo, que inclui PSLWriter e PenaltyWriter). Recomenda-se que todos os componentes de serviço estejam em execução sempre que cálculos forem necessários.
2. Diagnostique o contrato - o recurso de diagnóstico de contrato exibe os resultados de uma série de testes executados em todas as métricas do contrato (e fórmulas de penalidade se forem usadas.) A gravidade do resultado do teste é fornecida juntamente com um procedimento de resolução sugerido. É recomendável que um diagnóstico seja executado sempre que um contrato for ativado, bem como após a conclusão do cálculo para esse contrato.
3. Gere o relatório de formato livre "Status do cálculo". Esse relatório é fornecido como parte da instalação inicial do CA Business Service Insight e está localizado na pasta de agrupamento "Relatórios administrativos". Fornece informações sobre o andamento do cálculo e pode ser usado nesse momento para verificar se o mecanismo PSL está em andamento e se o cálculo foi concluído. Revise esse relatório para determinar se há problemas potenciais nos cálculos.

O relatório contém as seguintes colunas:

Campo	Descrição
Contrato	Contém o nome do contrato. A lista contém os nomes dos contratos vigentes e não vigentes.

Métrica	Contém o nome da métrica do contrato. A lista contém todas as métricas de cada contrato.
Período de monitoramento	Contém o período de cálculo da métrica. A lista exibirá uma entrada para cada unidade de tempo de cálculo da métrica de acordo com os agentes que estão ativos e com base na definição de granularidade da métrica. Nos casos em que o período de cálculo for o período de monitoramento, isso será indicado.
Atualização até	Contém a hora de atualização do último resultado. Indica que o resultado para a métrica específica está disponível até essa data. Por exemplo, se 01/01/2006 for exibido, indica que todos os resultados para essa métrica nessa unidade de tempo serão atualizados até essa data e que os relatórios estarão disponíveis para ela até essa data.
Último ciclo inicia em	Contém a hora do ciclo quando o cálculo foi iniciado com o resultado no qual a métrica foi atualizada.
Requer recálculo de	Caso o mecanismo marque uma determinada métrica para recálculo e ela ainda não tiver sido atualizada, a data resultante será exibida aqui para especificar a hora a partir da qual os resultados não são relevantes e, portanto, precisam de atualização. Isso pode ocorrer em qualquer caso de recálculo.
Última atualização em	A hora em que o mecanismo atualizou o registro com o último resultado.
Tratado pelo mecanismo nº	Contém o número do mecanismo atribuído para lidar com o cálculo da métrica específica.

Esse relatório também pode fornecer as informações a seguir, com base nos dados brutos disponíveis:

- O tempo que o mecanismo levou para calcular uma única métrica. É possível verificar, classificando todas as métricas que foram calculadas em um único ciclo pela hora de atualização, quanto tempo o mecanismo levou para atualizar cada métrica. Todos os registros com o mesmo valor de "Último ciclo inicia em" são calculados durante o mesmo ciclo e a hora de atualização é "Última atualização em". É possível avaliar o tempo que o mecanismo levou para calcular a métrica inteira com as unidades de tempo subjacentes do agente, bem como cada uma das unidades de tempo.
- O tempo que o mecanismo levou para calcular um contrato inteiro. Isso é feito ao examinar a hora de atualização da primeira métrica do contrato e a hora da última atualização da última métrica do contrato, e calcular o tempo entre elas.

Recálculo completo das métricas do contrato

O processo de recálculo no contexto atual é o início de um recálculo completo do sistema pelo Especialista em fonte de dados ou Especialista em lógica de negócios. Não é o recálculo do mecanismo executado durante um processo de cálculo regular. Esse tipo de ação geralmente é executada durante ou após o processo de configuração do contrato quando várias falhas de funcionamento tiverem sido identificadas.

É recomendável que um recálculo completo seja iniciado somente quando o sistema estiver em um estado estável (isto é, não durante a criação do sistema) antes ativá-lo no local.

O recálculo geralmente é feito por meio da execução de um script SQL no banco de dados. Esse script faz a limpeza da tabela PSL e de todas as tabelas de suporte associadas que fazem parte do processo de cálculo. Esse script deve ser aprovado e validado pela equipe de suporte da CA antes de ser executado, pois eventuais alterações na estrutura do sistema podem ter resultado em alterações no esquema de banco de dados e/ou objetos dependentes.

Observação: antes de executar o script, todos os componentes de serviço devem ser interrompidos, e depois da execução, os componentes de serviço podem ser reinicializados para iniciar os cálculos novamente.

As seguintes situações podem indicar a necessidade de um recálculo completo:

- Problemas com as definições no contrato - se estiver nessa etapa, descobre-se que o destino de uma métrica foi definido incorretamente ou que o limite de uma métrica está errado.
- Erros no contrato.
- Um requisito para reavaliar os limites do painel ou gerar novamente alertas de SLA.

Observação: entre em contato com o suporte da CA para discutir essa opção, se necessário, e também para obter cópias dos scripts relevantes para a versão do CA Business Service Insight que está instalada.

Criar produtos a serem entregues (Gerente de contratos)

Nesse estágio, todas as saídas do sistema são configuradas para atender aos requisitos, incluindo a criação e a formatação de todos os relatórios.

A configuração de produtos a serem entregues inclui o seguinte:

- Definir as configurações de segurança (permissões do usuário e grupos).
- Criar relatórios salvos.
- Criar booklets.
- Criar modelos de exportação de contrato.
- Criar exibições de SDN (Service Delivery Navigator - Visualizador de Entrega de Serviços).
- Configurar mapas do painel.
- Criar perfis de alerta de nível de serviço.

Definir configurações de segurança (administrador)

Ao definir as configurações de segurança, os usuários da CA e suas permissões associadas precisam ser configuradas. Essas configurações incluem definir as informações que devem estar acessíveis para um usuário (que entidades no sistema o usuário pode exibir ou editar). Essas permissões podem ser definidas em vários níveis de grupos de usuários, funções ou até mesmo individualmente por usuário. As informações acessíveis são definidas em relação às partes contratuais e podem ser configuradas diretamente para o usuário, ou herdadas do grupo de usuários ao qual o usuário pertence.

É nessa etapa que as principais funções são configuradas e os grupos de usuários são associados a elas, de modo que, quando um novo usuário é adicionado, ele só precisa ser vinculado a um grupo para herdar as configurações relevantes.

As ações permitidas são configuradas nas funções e são enviadas ao usuário ao associá-lo diretamente com uma função, ou pelo grupo de usuários ao qual o usuário pertence. Ações permitidas relacionadas ao painel também são definidas na função vinculada.

É recomendável que o administrador determine quais grupos de usuários e funções devem ser definidos e suas permissões necessárias para configurar uma estrutura que permita a adição fácil de usuários.

Como criar relatórios

Use o processo a seguir para criar um relatório:

1. Criar todas as pastas necessárias do relatório - todas as pastas devem ser criadas com antecedência para que estejam disponíveis ao salvar cada novo relatório. Em geral, cada contrato tem sua própria pasta, incluindo uma pasta executiva para os relatórios de alto nível.
2. Definir os critérios de filtragem de relatórios usando o Assistente de relatório - a criação de cada relatório salvo é iniciada ao gerar o relatório usando o Assistente de relatório. No Assistente de relatório, os critérios de filtragem necessários são selecionados e o relatório é gerado. Os administradores de TI podem definir parâmetros de relatório que determinam os campos definidos pelo usuário a serem preenchidos pelo usuário/visualizador do relatório, permitindo a geração de resultados do relatório específicos para os interesses desse usuário.

Ao gerar um relatório para configurá-lo como um relatório salvo, é muito importante definir os critérios de filtragem da forma mais flexível possível. Isso evita trabalhos desnecessários à medida que o sistema cresce e evolui. O objetivo deve ser garantir que cada relatório sempre exibirá informações atuais e atualizadas. Por exemplo, se no momento um relatório mostrar três componentes de serviço, no futuro, ao adicionar novos componentes de serviço, é importante que esse serviço seja adicionado automaticamente ao relatório e não exija a instalação de um novo relatório. Ou, se a geração de relatórios for mensal e, no momento, o relatório mostrar três meses, no próximo mês deverá mostrar quatro meses, incluindo o mês anterior. Em muitos casos, os relatórios sempre devem mostrar os últimos 6 meses de dados. Esses tipos de relatórios de "janela deslizante" são muito úteis em comparação aos relatórios de duração fixa, pois não exigem atenção adicional depois de criados.

A seguir, há algumas dicas para definir os critérios de filtragem para os relatórios salvos:

Guia Critério

- Usar a opção "Apenas dados de negócios" limita os dados exibidos somente para aqueles relacionados ao período de monitoramento da métrica.
 - Sempre dê preferência para a *opção agrupar por* ou para a *opção relatar por* em vez de selecionar as métricas específicas a serem relatadas.
 - Defina um intervalo de datas dinâmico. Se essa opção estiver definida como um intervalo de datas fixo, o relatório sempre mostrará os mesmos resultados. (por exemplo, Últimos 3 meses é um intervalo dinâmico).

Guia Diversos

- Verifique se "Períodos incompletos" devem ser apresentados no relatório. Se precisarem ser apresentados, selecione essa opção na lista. Normalmente, ao configurar relatórios, o período incompleto é excluído, pois ele não é um resultado final e, portanto, não é um resultado comercial.

- Crie o layout do relatório - quando o relatório tiver sido gerado, será possível criar seu layout usando a ferramenta de design na página Relatório. O design pode ter requisitos específicos, dependendo do destinatário pretendido do relatório. É recomendável que uma série de layouts de design sejam criados, um para cada possível cenário, e esses designs devem ser aplicados ao restante dos relatórios a serem gerados. Portanto, ao definir os critérios do relatório, na guia *Exibir*, escolha *Design*.

Observação: para obter dicas sobre como usar a ferramenta de edição para criar o layout de relatórios, consulte a próxima seção.

Guia Geral

- Salve o relatório - depois que o relatório tiver sido gerado e criado, poderá ser salvo na pasta relevante.
- Durante o processo de salvar o relatório, ele é associado aos usuários que têm acesso a ele. Portanto, é importante ter um grupo de usuários já definido para que seja possível associar os relatórios com os usuários. Os usuários com as permissões apropriadas podem ser vinculados ao relatório em uma etapa posterior, usando as opções de pasta.
- Associe os relatórios relacionados para que a navegação entre relatórios que são semelhantes ou que têm um relacionamento um com o outro seja simplificada para os usuários desses relatórios.
- Na página Pasta de relatórios, é possível criar relatórios, grupos de relatórios, relatórios compostos, relatórios de formato livre, booklets, atalhos e pastas, bem como pesquisar por relatórios.

No menu Relatórios, clique em Pastas de relatórios. A página Pasta de relatórios é exibida, mostrando uma lista de relatórios salvos.

Relatório de desvio

O valor de desvio é calculado automaticamente pelo mecanismo do CA Business Service Insight para métricas que têm um destino. O valor representa a diferença entre o nível de serviço atual e o destino. A fórmula de cálculo de desvio, calculada automaticamente pela CA, muda de acordo com a definição de nível de serviço: se o destino de nível de serviço for um valor máximo (quando o nível de serviço real é "não mais que"), ou um valor mínimo (quando o nível de serviço real é "não menos que"). Veja a seguir um exemplo de como a fórmula muda:

Objetivo	Limite de nível de serviço	Fórmula de desvio
O serviço deve estar disponível em pelo menos 99% do tempo programado.	O destino é o limite mínimo.	$Service_{Dev} = (-1) \times \frac{Service_{SL} - Service_{SLT}}{Service_{SLT}}$

O MTTR médio não deve ultrapassar 4 horas por mês.

O destino é o limite máximo.

$$Service_{Dev} = \frac{Service_{SL} - Service_{SLT}}{Service_{SLT}}$$

Onde $Service_{Dev}$ = desvio de serviço

Onde $Service_{SLT}$ = desempenho do serviço esperado

Onde $Service_{SL}$ = desempenho do serviço real

O exemplo a seguir ilustra um cálculo de desvio mínimo:

O serviço deve estar disponível em pelo menos 99% do período de atividade programado. O nível de serviço real é de 98% durante o período de atividade programado.

$$Service_{Dev} = (-1) \times \frac{98\% - 99\%}{99\%} = 1.0101\%$$

Os relatórios de desvio são usados para exibições de alto nível de garantias de outra natureza (tipo diferente de cálculo) e para agregá-las em uma única barra com áreas comuns.

Se, por exemplo, no contrato houver a matriz a seguir:

Serviço - central de ajuda	Domínio do serviço	Domínio do serviço	Domínio do serviço
	Gerenciamento de ticket	Gerenciamento de ticket	Gerenciamento de ticket
	Prioridade 1	Prioridade 2	Prioridade 3
	Tempo médio de resolução de P1	Tempo médio de resolução de P2	Tempo médio de resolução de P3
	% de tickets de P1 resolvidos em T1	% de tickets de P2 resolvidos em T1	% de tickets de P3 resolvidos em T1
	% de tickets de P1 respondidos em T1	% de tickets de P2 respondidos em T1	% de tickets de P3 respondidos em T1
	Tempo médio de resposta de P1	Tempo médio de resposta de P2	Tempo médio de resposta de P3

Os resultados de gerar um relatório de domínio do serviço filtrado pelo serviço da central de ajuda são mostrados no gráfico a seguir.

O relatório acima permite que o Gerenciador de nível de serviço veja o desempenho da central de ajuda com base em várias prioridades, independentemente do contrato ou tipo de obrigação.

Todas as métricas da central de ajuda são agregadas em uma única barra com base nas respectivas prioridades.

Por exemplo, a barra de Prioridade 1 mostra as três métricas definidas na métrica e agrega seus desvios em um único valor. (O método de agregação pode ser escolhido como médio/mínimo/máximo no Assistente de relatório.)

Com esse relatório, o gerenciador pode concluir, por exemplo, se deve investir mais recursos em tickets de Prioridade 1 ou alterar os contratos relacionados a eles.

Esse exemplo mostra que a modelagem fornece os dois relatórios em uma única obrigação que mostra se um contrato foi respeitado ou violado, mas também um relatório de gerenciamento mais amplo que permite que o Gerenciador de nível de serviço gerencie seus recursos de forma mais eficiente e, portanto, melhore seus componentes de serviço.

Relatórios de formato livre

Os relatórios de formato livre permitem que os usuários gerem relatórios com base em consultas SQL do banco de dados do CA Business Service Insight ou de qualquer outra fonte de dados externa, que pode ser acessada por meio de uma conexão do servidor do CA Business Service Insight. Isso também inclui quaisquer outros tipos de fonte de dados que podem ser acessados por meio de ODBC, como Excel, Access, Lotus Notes, arquivos de texto, etc. Os relatórios de formato livre geralmente são usados para configurar relatórios de estatísticas com base em dados criados pelos comandos Tools.SaveRecord e Tools.SaveFields da lógica de negócios.

Os relatórios de formato livre se conectam por meio de um texto de conexão a um banco de dados selecionado e executam uma consulta SQL no banco de dados usando um texto de consulta. É possível adicionar parâmetros a ambos os textos para criar relatórios dinâmicos que permitem ao usuário digitar ou selecionar valores específicos para inclusão na consulta, como o nome de usuário e a senha para se conectar ao banco de dados.

Os relatórios de formato livre são exibidos nas guias Gráfico, Dados e Filtro, da mesma forma que os relatórios gerados usando o assistente de relatório.

Observação: relatórios de formato livre podem incluir gráficos somente se todas as colunas, exceto a primeira coluna, forem numéricas. Os dados na primeira coluna são usados para os títulos do eixo X. Os nomes da coluna são usados para outros títulos.

Devido ao fato de que os relatórios de formato livre usam o acesso direto a um banco de dados e uma consulta SQL aberta, a manutenção é problemática. É preciso tomar muito cuidado para não afetar os dados subjacentes que servem como origem para os relatórios de formato livre. Quando relatórios são gerados a partir de uma fonte de dados externa, recomenda-se que um processo de notificação seja configurado para garantir que essas fontes de dados não serão alteradas sem antes consultar o Gerente de contratos responsável pelos relatórios de dados de formato livre.

Informações gerais que devem ser lembradas ao criar relatórios de formato livre.

- Devido a problemas de formatação internacional de datas, muitas vezes é útil especificar o formato exato para a data, criando uma fórmula personalizada na lógica de negócios. Isso converte a data em um texto com o mesmo formato todas as vezes e evita problemas de formatação de datas dos EUA versus da Europa. Deve ser usada para as datas que são gravadas na tabela T_SLALOM_OUTPUTS por meio do comando Tools.SaveFields ou Tools.SaveRecord. A fórmula de exemplo é fornecida a seguir:

Function 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
```

- Ao usar o parâmetro de tempo dos manipuladores de eventos da lógica de negócios ou datas e horas do objeto eventDetails, sempre esteja ciente do impacto da mudança de fuso horário. Esteja ciente de que as datas, quando gravadas na tabela, podem estar em UTC, e não na hora local. Talvez você precise usar a função Tools.GetLocaleTime() para corrigir esse problema.
- Você ainda poderá usar o utilitário Criador de relatórios (quando um relatório de formato livre gerar um gráfico) para personalizar a aparência.
- As opções de exportação de PDF para o relatório de formato livre são personalizáveis usando a seção de parâmetros de relatório da tela de instalação de formato livre. Coisas como o layout da página (paisagem e retrato).
- HTML pode ser incorporado nos relatórios para executar diferentes funcionalidades, como adicionar hiperlinks ou alterar as fontes e as cores de linhas e colunas, ou exibir as configurações.
- Exibições de banco de dados (funcionalidade do Oracle) podem ser criadas na tabela T_SLALOM_OUTPUTS para simplificar o SQL necessário para os relatórios.
- Ao especificar parâmetros para os relatórios, é possível que eles sejam definidos de várias maneiras, como: texto livre, numérico (validar), senha (caracteres ocultos - útil para senhas de conexões a banco de dados), data (usando o utilitário seletor de data fornecido) e listas (criadas ao especificar uma instrução SQL para preencher a lista).
- Os valores dos parâmetros especificados na seção SQL do relatório de formato livre devem ser substituídos pelo nome do parâmetro, precedido pelo símbolo "@". por exemplo, @PARAM1. O uso de aspas nesse valor do parâmetro pode ser necessário se os valores forem textos.

Relatórios de formato livre de histograma genérico

A consulta a seguir pode ser usada em um relatório de formato livre para apresentar a distribuição dos valores em uma tabela por porcentagem, conforme exibido no gráfico a seguir:



É possível ver no gráfico acima qual proporção (em porcentagem) dos valores está abaixo de 11,5 (0%), abaixo de 804,74 (mais ou menos 50%) e abaixo de 1435,53 (100%).

Se o SLA especificar destinos como "x% dos valores devem estar abaixo de y", os resultados desse formato livre ajudam a encontrar os valores de x e y que asseguram a conformidade com o SLA.

Os seguintes parâmetros são usados na consulta:

- **@Query** - uma instrução select com o seguinte formato: `select some_value val from some_table`. O alias `val` é obrigatório. Essa consulta fornece os valores com os quais a distribuição é analisada.
- **@Buckets** - o número de valores no eixo x. Os valores da origem são arredondados para esses números. Por exemplo, se você especificar `@Buckets = 100`, os valores dos dados de origem serão divididos em 100 grupos, e a consulta irá calcular a quantidade de valores que fica em cada grupo. Quanto maior for `@Buckets`, mais preciso será o resultado, mas o gráfico não será visualmente atraente.
- **@Relation** - a direção de acúmulo. Valores possíveis: "`<=`" (quando o destino é não menos que) ou "`>=`" (quando for o contrário).

A consulta pode ser executada na fonte de dados ou na `T_SLALOM_OUTPUTS` para obter os melhores resultados.

A consulta a seguir gera o gráfico, conforme mostrado acima:

```
select val,100*records/(select count(*) from (@Query))
from
(
  select x.bucket_val val,
        sum(y.records) records
  from
  (
    select round(val/bucket_size,0)*bucket_size bucket_val,
          count(*) records
    from
    (
      select (max(val)-min(val))/@Buckets bucket_size
      from
      (
        @Query
      )
    ) params,
    (
      @Query
    ) source
  group by round(val/bucket_size,0)*bucket_size
  order by round(val/bucket_size,0)*bucket_size
) x,
```

```

select round(val/bucket_size,0)*bucket_size bucket_val,
       count(*) records
from
(
  select (max(val)-min(val))/@Buckets bucket_size
  from
  (
    @Query
  )
) params,
(
  @Query
) source
group by round(val/bucket_size,0)*bucket_size
order by round(val/bucket_size,0)*bucket_size
) y
where y.bucket_val @Relation x.bucket_val
group by x.bucket_val
order by x.bucket_val
)

```

A seguir, há um exemplo de lista de parâmetros (como XML) que pode ser usada:

```

<custom>
  <connection>
    <params/>
  </connection>
  <query>
    <params>
      <param name="@Query" disp_name="Data Type" type="LIST">
        <value>PDP Context Activation Success</value>
        <list>
          <item>
            <value>select success_rate as val from
PDP_Context_Activation_Success.CSV</value>
            <text>PDP Context Activation Success</text>
          </item>
          <item>
            <value>select throughput as val from [gprs
throughput volume by apn.csv]</value>
            <text>Throughput of a Single APN</text>
          </item>
          <item>
            <value>select throughput as val from [Generic GPRS
Throughput.CSV]</value>
            <text>Generic Throughput</text>
          </item>
        </list>
      </param>
      <param name="@Buckets" disp_name="X Axis Values" type="LIST">

```

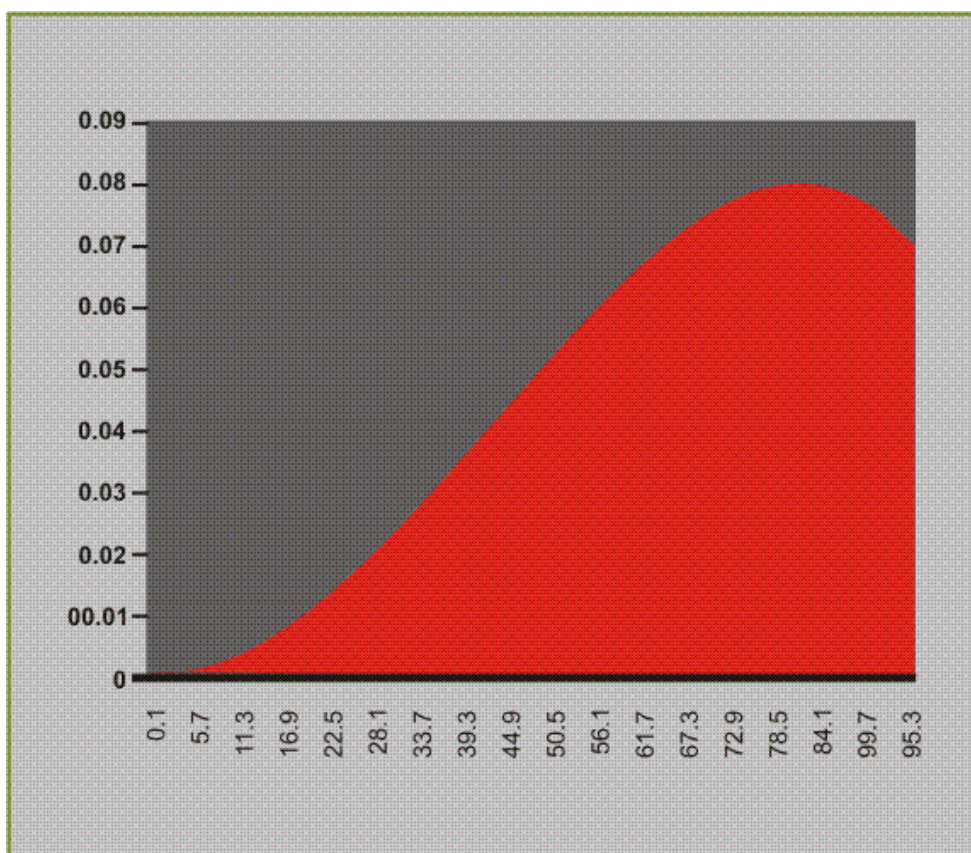
```
<value>100</value>
<list>
  <item>
    <value>25</value>
    <text>25</text>
  </item>
  <item>
    <value>50</value>
    <text>50</text>
  </item>
  <item>
    <value>100</value>
    <text>100</text>
  </item>
  <item>
    <value>250</value>
    <text>250</text>
  </item>
  <item>
    <value>500</value>
    <text>500</text>
  </item>
  <item>
    <value>1000</value>
    <text>1000</text>
  </item>
</list>
</param>
<param name="@Relation" disp_name="Violation of threshold means"
type="LIST">
  <value>providing too little</value>
  <list>
    <item>
      <value>>=</value>
      <text>providing too little</text>
    </item>
    <item>
      <value><=</value>
      <text>providing too much</text>
    </item>
  </list>
</param>
</params>
</query>
</custom>
```

Comentários

- A consulta foi otimizada para o Oracle e o SQL Server. Para outras fontes de dados ODBC, pode ser necessário adicionar 'como' antes de aliases de colunas e aplicar outras modificações.
- Ao exportar os resultados para o Excel, é recomendável que o usuário gere um relatório XY (de dispersão) para representá-lo. Quando o gráfico for pontilhado, também será possível exibir a dispersão dos valores.

Relatório em formato livre da distribuição normal genérica (Gaussiano)

A consulta detalhada a seguir pode ser usada em um relatório em formato livre para apresentar a distribuição normal de valores em uma tabela, conforme mostrado no gráfico a seguir:



Os seguintes parâmetros são usados na consulta:

- @Query-a selecione a declaração do formulário a seguir:

```
selecione min (something) min_val,  
          max (something) max_val,  
          avg (something) avg_val,  
          stddev (something) stddev_val  
do table_name
```

x_val é obrigatório. Esta consulta fornece os valores para que a distribuição seja analisada.

- @ Buckets - o número de valores no eixo x. Os valores de origem são arredondados para esses números. Por exemplo, se você especificar @Buckets = 100, os valores dos dados de origem serão divididos em 100 grupos, e a consulta irá mostrar o valor normal de distribuição para cada grupo. Quanto maior o @Buckets, mais preciso será o resultado, porém o cálculo será maior. 100 é uma escolha adequada para @Buckets.

Novamente, a consulta pode ser executada em dados de origem, ou o T_SLALOM_OUTPUTS para obter os melhores resultados.

A consulta a seguir irá gerar o gráfico como mostrado a seguir:

```
selecione (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))  
de      (  
    @Query  
),  
(  
    selecione      rownum serial  
    de      t_psl  
    em que rownum < @Buckets  
)  
classificar por número de série
```

Parâmetros do XML correspondentes:

```
<custom>  
  <connection>  
    <params/>  
  </connection>  
  <query>  
    <params>  
      <param name="@Query" disp_name="Data Type" type="LIST">  
        <value>PDP Context Activation Success</value>  
        <list>  
          <item>  
  
          </item>  
        </list>  
      </param>  
    </params>  
  </query>  
</custom>  
selecione min(success_rate) min_val,max(success_rate) max_val,avg(success_rate)  
avg_val,stddev(success_rate) stddev_val  
de PDP_Context_Activation_Success.CSV
```



```

</value>
    <text>PDP Context Activation Success</text>
  </item>
  <item>
    <value>
selecione min(throughput) min_val,max(throughput) max_val,avg(throughput)
avg_val,stddev(throughput) stddev_val
de [gprs throughput volume by apn.csv]
</value>
    <text>Throughput in Kb</text>
  </item>
</list>
</param>
<param name="@Buckets" disp_name="X Axis Values" type="LIST">
  <value>100</value>
  <list>
    <item>
      <value>25</value>
      <text>25</text>
    </item>
    <item>
      <value>50</value>
      <text>50</text>
    </item>
    <item>
      <value>100</value>
      <text>100</text>
    </item>
    <item>
      <value>250</value>
      <text>250</text>
    </item>
    <item>
      <value>500</value>
      <text>500</text>
    </item>
    <item>
      <value>1000</value>
      <text>1000</text>
    </item>
  </list>
</param>
</params>
</query>
</custom>

```

Comentários

- A consulta foi otimizada para o Oracle e o SQL Server. Para outras fontes de dados ODBC, pode ser necessário adicionar 'como' antes de aliases de colunas e aplicar outras modificações.
- Ao exportar os resultados para o Excel, é recomendável que o usuário gere um relatório XY (de dispersão) ou um relatório de área para representá-lo.

Configurar páginas do painel

A seção a seguir contém recomendações sobre o conteúdo a ser configurado para usuários do CA Business Service Insight. A recomendação é de alto nível e deve levar em conta dos requisitos do cliente específico. As páginas são descritas a seguir no contexto de um determinado usuário que é descrito por meio de sua função na empresa.

Páginas do gerente executivo

Pressupõe-se que um gerente executivo esteja interessado em exibições de alto nível em departamentos, países, contas, etc. Normalmente, este gerente não é operacional e requer exibições que lhe forneçam informações para tomar decisões estratégicas. Portanto, o status do contrato, em vez de status atual, pode ser mais relevante para a exibição em mapas e painel de visualização agregado do gerente executivo.

Por exemplo, os painéis de visualização a seguir podem ser incluídos na página Visão geral:

Painel de visualização	Descrição
Contas críticas	Inclui todas as partes contratuais marcadas como importante'. O gerente executivo seleciona os contratos ou as partes contratuais que considera importante.
Desempenho geral	Inclui medidores personalizados de qualidade geral, medidores de exibição agregada que incluem todos os KQI de contas.
Departamentos	Usa a imagem de plano de fundo do organograma e coloca os medidores em departamentos relevantes. Em geral, o grupo de componente de serviços é útil aqui, dependendo do que um departamento representa na organização.
Regiões	Usa a imagem do plano de fundo com um mapa geográfico e localiza medidores de grupos de partes contratuais em locais relevantes.
Desempenho financeiro	Inclui medidores que incluem informações agregadas sobre métricas financeiras
URL	Inclui páginas do portal corporativo, por exemplo, leads no sales force.

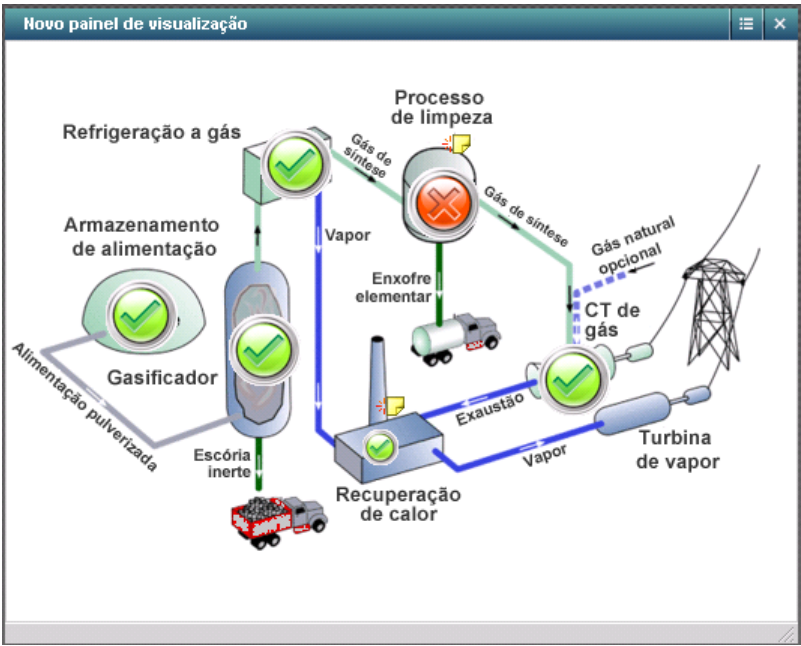
Relatórios recomendados para a página:

Relatório	Descrição
Relatório de desvio	Dez piores contratos de um determinado período fornece informações sobre as áreas com pior desempenho em entrega de serviços

Relatório financeiro do mês atual Resume o status financeiro agregado por hora

Página de processo:

Esta página deve conter um painel de visualização que apresenta um diagrama de processo que representa cada cadeia do processo, como no exemplo a seguir:



Páginas do gerente de contratos

Páginas configuradas para que um gerente de contrato exiba como o serviço está sendo entregue para cada conta que ele gerencia. Painel de visualização de contratos relevantes para atualizar o gerente sobre o nível de serviço atual fornecido mediante as obrigações de contratos que estão sob sua responsabilidade. Além disso, essa página exibe os relatórios disponíveis para todos os componentes de serviço incluídos no contrato.

A página Visão geral contém os seguintes painéis de visualização:

Painel de visualização	Descrição
Minhas contas	Medidores de todos os contratos ou partes contratuais que estão sob a responsabilidade do gerente de contratos.
Meus serviços	Os componentes de serviço em todas as contas gerenciadas pelo gerente de contratos.
Desempenho financeiro	Medidores que incluem informações agregadas sobre métricas financeiras para as contas gerenciadas.

URL

Portal como o sistema CRM

Cria uma página de conta para todas as contas gerenciadas que fornecem uma exibição das obrigações da conta específica. É recomendável combinar exibições de cálculo do status atual com o status contratual para fornecer ao gerente de contas a capacidade de ser proativo e agir de forma rápida. Por exemplo:

Página da conta	Descrição
Obrigações da conta	Inclui métricas incluídas no contrato
Desempenho financeiro	Medidores que incluem informações agregadas sobre métricas financeiras para as contas gerenciadas.

Páginas do gerente

Páginas a serem configuradas por um gerente encarregado de um serviço ou domínio específico, que exibe uma visualização detalhada dos objetivos do serviço em diferentes contratos, bem como os estados de seus objetivos. Além disso, essa página contém relatórios que destacam os parâmetros de nível de serviços essenciais que estão sendo medidos.

Os painéis de visualização incluídos nas páginas do gerente de serviços são semelhantes às páginas do gerente de contas, em que apenas as informações são exibidas e agregadas em um nível de serviço, em vez de nível de contrato ou parte contratual.

Páginas iniciais

Uma página do painel pode ser atribuída como uma página inicial para ativar um gateway personalizável para a interação com o sistema, permitindo acesso rápido às informações e ações.

Adicionar perfis de alerta de nível de serviço

O perfil de alerta permite a definição de condições sob as quais uma notificação de alerta é enviada a um destinatário predefinido usando um dispositivo especificado.

Antes de criar perfis, é importante considerar que condições são importantes o suficiente para exigir a notificação e também quem são os destinatários pretendidos. É importante definir perfis que podem auxiliar o gerente de contratos e o administrador do sistema a gerenciar, de forma eficiente, os problemas e as questões de serviço encontrados. Todas as notificações devem ser específicas no que se refere aos recursos necessários para resolver as ocorrências mais urgentes de forma a evitar uma situação de violação.

Ao adicionar alertas de nível de serviço, há um número de campos que pode ser usado para determinar o status da métrica e para decidir se uma condição de alerta deve ser disparada ou não. Qualquer um dos detalhes de métrica padrão pode ser usado para os critérios de filtragem e condição (como nomes de contrato, de serviço, de métrica, metas, períodos de atividade, resultados do nível de serviço, etc.). As previsões do nível de serviços também podem ser alertadas, caso em que alguns valores adicionais como desvio de melhor/pior caso são fornecidos, e os níveis de serviço também podem ser estimados. Eles são muito úteis para decidir se uma violação de nível de serviço pode ser recuperada ou não em um determinado período.

Alertas com base no log de sistema e seções gerais do CA Business Service Insight são outros recursos extremamente úteis e permitem que um alerta seja enviado com base em qualquer ação que ocorra no sistema (registrada no log do sistema). Como praticamente todas as ações são registradas, isso fornece um amplo escopo dos perfis de alerta. O log do sistema comum que alerta os perfis, consiste em:

- Notificação de um conector que executa o alerta para o especialista em fontes de dados.
- Qualquer condição de erro registrada no log envia um alerta por email ao administrador do sistema.
- Erros personalizados criados pela lógica de negócios, relevantes às condições específicas das fontes de dados, podem gerar um alerta por email para o grupo responsável pelo serviço dentro da organização e ao gerente de contratos. (por exemplo, um incidente de alta prioridade não foi resolvido dentro do prazo > envia um alerta à central de ajuda para escalar a ocorrência)
- Tentativas repetidas de logon inválidas podem ser enviada por email para o administrador do sistema para determinar se alguém está tentando invadir o sistema
- Novas entradas de conversão oriundas do conector (por exemplo, novas entradas de recurso) foram encontradas em uma origem de dados, e podem precisar de conversão no sistema para permitir que os SLAs sejam calculados corretamente.

A janela a seguir Detalhes do perfil de alerta, mostra a configuração de um perfil de alerta personalizado que relata um evento personalizado de eventos de abertura do ticket pela central de atendimento. Se o critério for correspondido, um alerta crítico é enviado ao coordenador da central de atendimento para garantir que este ticket seja gerenciado de forma eficiente. Isso pode ser útil para as situações em que um aplicativo está sob investigação minuciosa em algum momento e requer um nível de cuidado especial.

Detalhes do perfil de alerta

Nome: *	Critical fidessa Outage					
Descrição:	All severity 1 incidents raised for Fidessa application are sent to Helpdesk Coordinator for analysis and possible escalation					
Tipo: *	Helpdesk Ticket Open Event	Recurso: * Equity Traders - Transl				
Fórmula de cálculo da condição:	#ID = "FIDESSA" && #Severity = 1	Editor de fórmulas				
Assunto da mensagem: *	Critical Fidessa Outage					
Mensagem:	Critical Fidessa Outage - Equity Traders Down	Editor de mensagens				
Destinatários: *	<table><thead><tr><th>Incluído</th><th>Disponível</th></tr></thead><tbody><tr><td>Helpdesk Coordinator</td><td>Obicore Support Sadmin</td></tr></tbody></table> Exibir detalhes Adicionar destinatário de alerta		Incluído	Disponível	Helpdesk Coordinator	Obicore Support Sadmin
Incluído	Disponível					
Helpdesk Coordinator	Obicore Support Sadmin					
Gravidade: *	Crítica					
Status: *	Ativado					
Intervalo mínimo entre alertas: *	10 minutos					

O exemplo a seguir refere-se à implementação de um alerta de violação de nível de serviço, com base se o SLA ainda pode ser atendido, considerando o nível de serviço atual e o tempo restante para o período de monitoramento de negócios da métrica.

Exemplo: alerta reversível de violação ao SLA

Esse alerta é disparado quando um objetivo (métrica) está sendo violado, mas uma vez que os melhores cenários de caso indicam a conformidade com o objetivo, a violação pode ser evitada se um nível de serviço mais adequado for fornecido.

Tipo: previsão do nível de serviço

Fórmula de cálculo da condição:

`#Deviation>0 And #BestCaseDeviation<=0 And
#BestCaseServiceLevel<>#WorstCaseServiceLevel And #ClusterItem = ""`

Mensagem:

Observe que o objetivo #Métrica do contrato #Contracto está sendo violado, mas a violação é reversível.

Embora o objetivo do nível de serviço seja #Objetivo, os seguintes níveis de serviço são previstos:

- Nível de serviço agora: #ServiceLevel (Desvio: #Deviation%).
- Nível de serviço do pior cenário: #WorstCaseServiceLevel (Desvio: #WorstCaseDeviation%).
- Nível de serviço do melhor cenário: #BestCaseServiceLevel (Desvio: #BestCaseDeviation%).

Capítulo 4: Instalação e implantação

Esta seção contém os seguintes tópicos:

[Introdução](#) (na página 202)

[Preparações](#) (na página 204)

[Instalação](#) (na página 206)

Introdução

Quando a fase de configuração for concluída e o cálculo do sistema for verificado corretamente, o sistema estará pronto para implantar em um ambiente de produção. Este capítulo tem como objetivo abranger todas as etapas que podem ser necessárias durante essa fase. Essas etapas podem variar entre instalações. No entanto, os seguintes itens devem ser verificados para ver se todas as preparações foram feitas para integrar o sistema em um ambiente de produção em tempo real.

- O hardware e o software de produção estão disponíveis e prontos para instalação.
- O servidor de email está configurado para oferecer suporte ao envio de email externo/interno.
- A existência do agente FTP (para transferências de arquivo de dados, se necessário).
- Os feeds de origem de dados estão disponíveis e funcionando.

Há duas fases na instalação. Durante a primeira fase, a fase de instalação - o administrador do sistema deve instalar e integrar os componentes do CA Business Service Insight ao ambiente de produção. Durante a segunda fase, a funcionalidade do sistema é verificada e o sistema deve ser colocado em um status monitorado para garantir que os processos do sistema e as funcionalidades da interface do usuário estejam funcionando corretamente.

Durante o processo de instalação, talvez seja necessário trabalhar em servidores de produção por meio de ferramentas de conexão remota que devem ser instaladas em aplicativos e servidores web para permitir uma instalação completa e seguro.

A instalação do banco de dados Oracle deve ser feita por um especialista DBA da Oracle, se possível, para garantir que a configuração seja executada corretamente e esteja em conformidade com os requisitos da empresa, os quais podem ser atendidos. Como alternativa, a criação de banco de dados pode ser deixada para o administrador do sistema usando o utilitário de criação de dados fornecido com a instalação do CA Business Service Insight. Nesse caso, a instalação deve ser verificada pelo DBA para garantir que ela foi concluída com êxito. Se o sistema já foi configurado em um sistema em desenvolvimento e precisa ser migrado, é necessário mover o conteúdo do banco de dados para o novo banco de dados de produção.

Às vezes, uma vez que o sistema completo foi transferido para o novo ambiente de produção é aconselhável descartar todos os dados brutos e calculados armazenados no sistema e recarregar o sistema a partir do zero (manter todo o catálogo de serviços, recursos e as conversões do curso). Esta é uma excelente maneira de testar a operação de todo o sistema no ambiente de produção.

Resumindo, as seguintes etapas estão incluídas nessa fase:

- Instalar e se conectar ao sistema.
- Carregar o banco de dados do sistema configurado, se necessário.
- Integrar as conexões para email, Exchange, SMS, etc.

- Testar a funcionalidade e ajustar o desempenho.
- Instalar, configurar e testar conexões remotas.
- Reinicializar o sistema pronto para a operação online.
- Ativar o conector de dados para iniciar a sondagem de origens de dados e o carregamento de dados brutos.
- Alternar o mecanismo do CA Business Service Insight e permitir que os cálculos sejam iniciados.
- Preencher a documentação necessária, como a administração do sistema, o banco de dados e outros procedimentos de manutenção.
- Assegurar que o treinamento seja fornecido aos usuários.

Preparações

Para garantir uma instalação eficiente, é importante que algumas coisas sejam preparadas com antecedência:

1. **A exportação de banco de dados do ambiente de desenvolvimento.** Essa exportação deve ser executada usando o os scripts aprovados da CA, que criam um arquivo de extração (despejo) para que seja possível importá-lo de volta ao software do sistema de produção.

Observação: é importante que a exportação seja executada por um usuário do banco de dados que tenha privilégios suficientes, e que esse mesmo usuário esteja acessível ao importar no banco de dados. Para este fim, a conta **oblicore** pode ser usada, uma vez que é sempre criada em cada sistema ou, como alternativa use a conta do **sistema**. No entanto, certifique-se de que a senha **do sistema** esteja disponível no banco de dados de destino para permitir o processo de importação.

2. **Scripts do banco de dados** - se o DBA deseja alterar os scripts de criação do banco de dados, eles devem ser verificados por um DBA da CA com antecedência. Esses scripts devem ser verificados e estarem prontos, com antecedência, para permitir uma instalação tranquila. É comum que diferentes organizações tenham comentários e alterações que devem ser aprovados pelo administrador de banco de dados da CA, a fim de garantir que a configuração do banco de dados esteja em conformidade com as diretivas locais.

- **Conectividade e acessibilidade do servidor** - tanto o aplicativo quanto os servidores web devem ter uma conexão cliente remota instalada para oferecer suporte à acessibilidade externa (se não for possível o acesso físico no momento da instalação, ou no futuro para questões de suporte). A acessibilidade externa também é importante para permitir o monitoramento contínuo do sistema durante a instalação. Se for necessário o acesso remoto externo, observe que isso pode levar mais tempo para ser organizado devido às outras implicações de segurança; portanto, ele deve ser tratado antes do tempo.

As conexões remotas podem ser estabelecidas usando uma das seguintes ferramentas:

- PcAnywhere
- Host / Master de proxy
- Microsoft Terminal Server
- Área de trabalho remota
- VNC
- Qualquer outra ferramenta de conexão remota suportada pela organização.

Observações:

1. O Microsoft Terminal Server se conecta ao servidor simulando um logon do servidor, diferentemente do PcAnyWhere, VNC, Proxy e outras ferramentas.
2. Alguns softwares Oracle não podem ser instalados por meio do Microsoft Terminal Server e devem ser instalados localmente.
3. **Acessibilidade da origem de dados** - a origem de dados deve ser acessível para manipulação manual, e a conexão ODBC deve ser configurada para permitir que os conectores se conectem à origem de dados.
4. **Segurança do servidor** - um perfil de usuário com direitos de administrador local é necessário tanto no aplicativo quanto no servidor web. Se a plataforma do banco de dados for um sistema operacional Windows padrão, também será necessário um perfil de usuário com direitos de administrador local. Em plataformas UNIX, o administrador do banco de dados deve ter os privilégios adequados para criar o banco de dados no servidor.
5. **Acesso à internet** - permite que o **administrador do sistema** se conecte à internet para fazer atualizações no sistema operacional ou no aplicativo, se necessário.
6. **Uma área de trabalho de usuários padrão para testar a funcionalidade do aplicativo.** Observe que o aplicativo requer o uso de controles ActiveX a serem instalados, os quais normalmente estão bloqueados por diretivas de segurança da organização.
7. **Programação de treinamento com equipe relevante** - treinamento de introdução que permite que a equipe faça o treinamento de aceitação do usuário e comece a trabalhar com o sistema

Instalação

O processo de instalação é descrito em detalhes no Guia de Instalação, e inclui as seguintes atividades:

1. Instalação do banco de dados:

A instalação do banco de dados é responsabilidade do especialista em fontes de dados ou do DBA Oracle. Em situações especiais, ela deve estar sob a orientação de um engenheiro da CA. A instalação do banco de dados inclui as seguintes etapas:

- Instalação do Oracle no servidor de banco de dados.
- Instalação do patch Oracle no servidor do banco de dados (se necessário - as releases de serviço de suporte mais recentes do software de banco de dados Oracle devem ser usadas).
- Instalação do cliente Oracle no servidor do aplicativo/web.
- Instalação do patch Oracle Client no aplicativo/servidor web (se necessário - certifique-se de que esta versão sempre corresponda ao servidor).

2. Instalação do aplicativo do CA Business Service Insight:

A instalação do aplicativo é realizada pelo administrador do sistema. Nos casos em que a instalação já foi realizada no ambiente de teste (na fase de configuração) durante a integração e o teste do conector, somente um estado de inicialização ou importação de banco de dados é necessário. Pode ser necessária a instalação em um ambiente de produção.

A instalação do aplicativo inclui as seguintes etapas:

- Criação do banco de dados do CA Business Service Insight.
- Instalação do servidor de aplicativos.
- Instalação do servidor web.
- Instalação do conector.

A instalação da release mais recente de serviço da CA no aplicativo/servidores web. Durante a instalação do aplicativo e de releases de serviços, os scripts SQL são fornecidos para garantir que o banco de dados seja atualizado para a estrutura correta da release. Eles são armazenados no diretório de instalação do aplicativo quando o banco de dados precisar ser atualizado novamente no futuro. (Por exemplo, se for necessário importar o banco de dados de um backup criado em uma release de serviço anterior. Nesse caso, você precisará executar o script de atualização das releases mais recentes do serviço para garantir que a estrutura do banco de dados esteja em linha com os componentes binários instalados como parte da mesma release do serviço)

Como opção, é possível também instalar outras ferramentas de suporte, como o desenvolvedor PLSQL ou outro utilitário SQL, para auxiliar em qualquer manipulação de dados de nível inferior que possa ser necessário.

Como importar ou exportar entre ambientes (especialista em fontes de dados)

Essa situação abrange a migração de conectores configurados e testados em um ambiente de teste/desenvolvimento, em um ambiente online ou em um ambiente de produção. Supõe-se que o ambiente de produção já foi instalado com uma instalação padrão do CA Business Service Insight e o banco de dados está presente.

O processo compreende as seguintes etapas:

Para exportar o banco de dados e importar ao novo ambiente

1. Interrompa todos os trabalhos no sistema de desenvolvimento e selecione um ponto lógico no sistema para executar uma exportação que pode ser usada no sistema de produção. Encerre todos os componentes de serviço e os componentes COM+ do CA Business Service Insight. Exporte usando os scripts de exportação do sistema padrão do CA Business Service Insight. (Se necessário, entre em contato com o suporte da CA)
2. Pegue a cópia da extração do banco de dados e coloque-a no servidor de produção pronto para importação. Observe que as versões de banco de dados Oracle de produção e de desenvolvimento devem corresponder. Importe o banco de dados usando scripts de importação padrão do sistema (fornecidos com os scripts de exportação)
3. Depois de concluído, verifique se há ocorrências de importação. Se não houver nenhuma, certifique-se de executar os scripts SQL da release de serviço mais recente (se necessário)
4. Execute o atalho Little Wizard para configurar o banco de dados para as configurações do novo sistema de produção.
5. Inicie todos os componentes de serviço do CA Business Service Insight e efetue logon no sistema de produção para confirmar se o sistema foi importado corretamente.

Para migrar os conectores

1. Instale o conector usando o utilitário AdapterManager com as configurações similares ao conector que você está importando (certifique-se de que o nome seja exatamente o mesmo para que a próxima etapa funcione corretamente).
2. Copie o arquivo de configuração do conector do sistema de desenvolvimento para a nova pasta do conector no sistema de produção. Substitua a configuração padrão fornecida. (Certifique-se de que o arquivo seja substituído. Caso contrário, isso significa que o nome não é mesmo, o que causará problemas durante a execução.)
3. Atualize o arquivo de configuração do conector - a comunicação com o servidor do CA Business Service Insight e com a origem de dados deve ser atualizada para atender ao novo ambiente. A seção OblicoreInterface deve ser atualizada com a porta correta do conector. A seção DataSourceInterface deve ser atualizada com a ConnectionString correta ou padrão de nome de arquivo ou caminho, se necessário.
4. Certifique-se de que todos os nomes de origens de dados ODBC (Nome da origem de dados) esteja configurado e funcionando no computador novo.
5. Teste a conectividade do conector.
6. Teste a execução do conector.
7. **Conversões** - em casos em que houver scripts de conversão, eles devem ser ativados. Verifique se eles estão sincronizados com o conector e se as conversões estão executando conforme o esperado. Quando a conversão manual foi usada e não foi concluída previamente, outras conversões devem ser executadas nesta etapa.
8. **Programação do conector** - programa o conector para ser executado conforme o esperado. Se o conector foi definido como um aplicativo em um modo de execução única, ele pode ser programado usando o Agendador do Windows. Isso pode ser visualizado em Painel de controle > Tarefas agendadas. Consulte o tópico [Modos de execução do conector](#) (na página 87) para obter mais detalhes.

Integração - Instalação do servidor de email (Administrador do sistema)

Para ativar estes recursos de notificação do sistema, é necessário saber qual servidor de email e qual caixa de correio são usados para enviar emails do CA Business Service Insight. Este servidor de email deve permitir que o email seja retransmitido por ele, pois será enviado como SMTP de servidores do CA Business Service Insight a este servidor de email usando a conta especificada. Depois de concluir a instalação do servidor de email, é possível usar os recursos de email nos recursos Alertas e Aprovação de contrato no CA Business Service Insight

Clique no menu Administrador e selecione Configurações do site, Alertas. Configure as definições de email na seção de definições de alerta, incluindo o servidor de email, o endereço de envio e o nome do remetente juntamente com as informações do fornecedor SMS para usar os gateways SMS.

Verificar se o servidor de aplicativos pode acessar o servidor de email da organização e usar esta opção para enviar alertas do CA Business Service Insight é parte do teste de integração.

Para testar a conectividade entre o servidor de email e o servidor de aplicativos, execute o seguinte comando no servidor de aplicativos:

```
C:\Documents and Settings>telnet ORGANIZATION-MAIL 25
```

ORGANIZATION-MAIL

Define o servidor de email definido em seu cliente de email. Se não tiver certeza sobre a configuração, entre em contato com o administrador do sistema para obtê-la.

Para verificar o servidor de email cliente do Outlook:

1. No Microsoft Outlook, vá para Ferramentas > Contas de email e selecione a opção Exibir ou alterar as contas de email existentes.
2. Clique em Alterar.
3. Copie o servidor Microsoft Exchange. Este servidor é o servidor de email da sua organização.

Se houver uma resposta do servidor com esse comando significa que a conexão foi realizada com êxito. A resposta deverá ser igual ao seguinte:

Se você receber qualquer outra mensagem, significa que não há conectividade entre os dois servidores. Entre em contato com o administrador do sistema.

Definir preferências do sistema (Administrador do sistema)

Configurações de mecanismos

Listener de conectores:	☐ Mostrar mensagens informativas.
Mecanismo de correlação:	1
Número de instâncias do ACE:	1440 minuto(s)
Calcular nível de serviço a cada: *	Continuar
Quando o cálculo terminar: *	Normal
Método de programação: *	
☒ Limitar o intervalo entre os cálculos.	
Período consecutivo máximo a ser calculado: *	10 dia(s)
☒ Limitar a quantidade de memória usada pelo mecanismo de correlação.	
Começar a liberar memória quando exceder: *	800 MB
Interromper a liberação de memória quando estiver abaixo: *	400 MB
Calculador de penalidade:	
Calcular penalidades a cada *	60 minuto(s)
Supressor de log do sistema:	
☐ Inicializar parâmetros de tarefa	
Excluir as mensagens de depuração após: *	3 dia(s)
Excluir as mensagens informativas após: *	15 dia(s)

O processo de definir as preferências do sistema inclui a aplicação de valores relevantes para as variáveis do sistema. No menu Administração, clique em Configurações do site, Mecanismos, para abrir a caixa de diálogo Engine Preferences.

Para obter detalhes sobre as várias recomendações de parâmetros, consulte o Guia do administrador.

Configurações de segurança (Administrador do sistema)

As configurações de segurança incluem a criação de usuários, grupos de usuários e funções. Por padrão, todos os usuários estão associados à organização especificada durante a instalação do aplicativo. No entanto, também é possível criar outras organizações se necessário.

A maioria das definições necessárias já foi concluída durante a fase de configuração, dessa forma, no geral, apenas um pequeno ajuste é necessário para definir as configurações adicionais que foram identificadas desde então.

Para obter mais detalhes sobre a configuração de segurança, consulte a Ajuda online ou o Guia do Administrador.

Especificar configurações de sincronização do SSA

Ao usar o CA Spectrum Service Assurance (SSA) para detectar serviços para o CA Business Service Insight, é possível definir as configurações para ativar a sincronização automática. Os recursos de automação permitem manter a lista de serviços e os dados de serviço atualizados.

Observação: é necessário ter acesso à API restful no SSA para editar essas configurações.

Siga estas etapas:

1. No menu Administrador, clique em Configurações do site, Configurações do SSA.

A caixa de diálogo Configurações do SSA é aberta.

2. Digite as seguintes informações na área de autenticação de usuário do SSA:

URL do servidor

Especifica o URL do servidor do SSA de destino.

ID do usuário

Especifica a ID de usuário para o servidor do SSA.

Senha

Especifica a senha para a ID de usuário do servidor do SSA.

3. Digite as seguintes informações na área Opções de sincronização:

Sincronizar automaticamente

Especifica que você deseja que a sincronização ocorra automaticamente, de acordo com a frequência de sincronização (consulte o próximo parágrafo).

Definir a frequência de sincronização

Especifica a frequência das pesquisas por novos serviços. É possível indicar um valor em horas ou dias.

Sincronização manual

Ativa a sincronização manual na caixa de diálogo.

4. Digite as seguintes informações na área Opções de dados padrão:

Serviços padrão para:

Permite definir o padrão para serviços detectados como gerenciados ou não gerenciados.

Ativar o cálculo de métricas de comparação

Permite definir a ação padrão para a configuração das métricas de comparação de serviços do SSA.

5. Clique em OK.

As configurações do SSA são ativadas.

Apêndice A: Domínios de serviço e categorias de domínio da amostra

A tabela a seguir é uma compilação dos domínios comuns de serviço e categorias que são normalmente usadas.

Domínio do serviço	Categoria do domínio	Comentários
Disponibilidade do sistema	Percentual do tempo disponível	
	Número de interrupções/tempo de inatividade	
	Tempo médio para correção	
	Tempo Médio entre falhas	
Disponibilidade de serviço	Minutos de inatividade	
	Quantidade de dias interrompidos	
Gestão financeira	Custo do serviço	
Desempenho do processo	Percentual de processos concluídos no prazo	
	Número de ciclos do processo concluído	
Gerenciamento de incidentes	Percentual de incidentes resolvidos < T1	
	Percentual de incidentes respondidos < T1	
	Quantidade de incidentes	
	Percentual de incidentes convertidos em problema	
Satisfação do cliente	Percentual de satisfação do cliente	
	Pontuação média de CSI	
Desempenho da Central de atendimento	Percentual de chamadas abandonadas	
	Média de tempo de atendimento de chamadas	
	Percentual de FLLR	(First Level Resolution Rate - Taxa de resolução de primeiro nível)
Qualidade de dados	Percentual de precisão	
	Percentual de pontualidade	

Domínio do serviço	Categoria do domínio	Comentários
Número total de erros/defeitos		

Apêndice B: Exemplos de estudo de caso

Esta seção contém os seguintes tópicos:

[Exemplos de modelagem do contrato](#) (na página 215)

[Exemplo de modelagem de métricas financeiras](#) (na página 226)

[Exemplos de modelagem de dados](#) (na página 234)

[Usando exemplo de atributos personalizados](#) (na página 244)

[Exemplos de script de conversão](#) (na página 247)

[Exemplos de script de lógica de negócios](#) (na página 253)

[Criando exemplos de lógica de negócios efetivas](#) (na página 270)

[Caso de estudo 19: assistente de conector para fonte de dados com base em arquivo](#) (na página 288)

[Estudo de caso 21: exemplo de integração LDAP](#) (na página 303)

[Estudo de caso 22: gerando relatórios com o PERL](#) (na página 310)

Exemplos de modelagem do contrato

Para cada um dos estudos de caso a seguir, use os seguintes itens de categorização para os objetivos descritos:

- Serviço
- Domínio do serviço
- Categoria do domínio
- Período de atividade
- Destino
- Período de monitoramento
- Unidade de medida
- Estrutura de tópicos de cálculo
- Parâmetros de métrica/contrato

Estudo de caso 1: Disponibilidade do sistema

Considere a garantia contratual a seguir:

O sistema X deve estar disponível pelo menos 99.6% do mês durante o horário comercial.

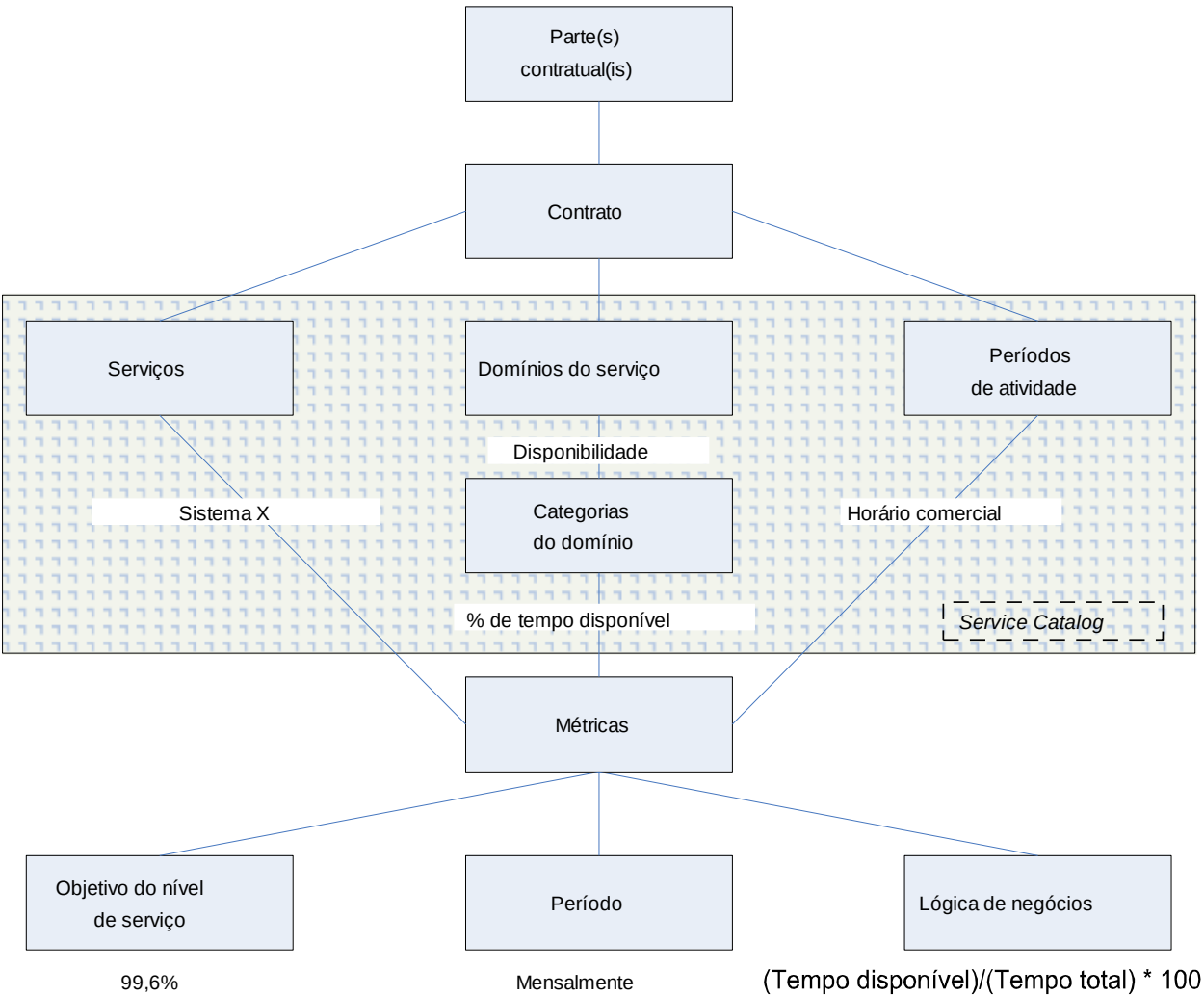
Isso pode ser descrito usando as seguintes entidades do CA Business Service Insight:

- **Nome da métrica:** percentual de tempo disponível do sistema X.
- **Período de monitoramento:** 1 mês
- **Período de atividade:** horário comercial (precisa ser definido posteriormente)
- **Lógica de negócios:** $((\text{tempo disponível})/(\text{tempo total})) * 100$
Observação: este estudo de caso se preocupa somente quando o monitoramento falha no horário comercial (que é o tempo de atividade da métrica)
- **Objetivo:** 99.6%

Além das informações sobre a métrica anterior, os seguintes itens do catálogo de serviço do sistema também podem ser deduzidos a partir das descrição acima:

- **Serviço:** sistema X.
- **Domínio de serviço:** disponibilidade.
- **Categoria de domínio:** percentual de tempo disponível.
- **Grupo de serviço:** qualquer grupo que identifica mais de um sistema que pode haver monitoramento. Nessa etapa, é difícil determinar se um grupo adequado pode ser criado.

Assim, agora é possível reproduzir o diagrama a partir da seção Modelamento de contrato deste documento exibindo essas entidades em um diagrama.



Estudo de caso 2: Disponibilidade do sistema 2

A disponibilidade do sistema CNP deve manter os níveis a seguir:

Ambiente	Dias da semana	Finais de semana
Produção	99.9%	98.9%
Desenvolvimento	90%	ND
Teste e controle de qualidade	Sem garantia	ND
Rede	99.9%	ND

Soluções sugeridas:

Métrica	Média de produção do sistema CNP nos dias da semana
Objetivo	99.9
Período de monitoramento	1 mês
Unidade de medida	%
Serviço	Produção do sistema CNP
Domínio do serviço	Disponibilidade
Categoria de domínio	Disponibilidade do aplicativo
Período de atividade	dias úteis
Métrica	Média de produção do sistema CNP nos finais de semana
Destino	98.9
Período de monitoramento	1 mês
Unidade de medida	%
Serviço	Produção do sistema CNP
Domínio do serviço	Disponibilidade
Categoria de domínio	Disponibilidade do aplicativo
Período de atividade	Finais de semana
Métrica	Média de desenvolvimento do sistema CNP durante nos dias da semana
Destino	90
Período de monitoramento	1 mês
Unidade de medida	%
Serviço	Desenvolvimento do sistema CNP
Domínio do serviço	Disponibilidade
Categoria de domínio	Disponibilidade do aplicativo
Período de atividade	dias úteis
Métrica	Média de teste/ controle de qualidade do sistema CNP nos dias da semana
Destino	none
Período de monitoramento	1 mês
Unidade de medida	%

Serviço	Teste e controle de qualidade do sistema CNP
Domínio do serviço	Disponibilidade
Categoria de domínio	Disponibilidade do aplicativo
Período de atividade	dias úteis
Métrica	Disponibilidade de rede do sistema CNP
Destino	99.9
Período de monitoramento	1 mês
Unidade de medida	%
Serviço	Rede do sistema CNP
Domínio do serviço	Disponibilidade
Categoria de domínio	Disponibilidade de rede
Período de atividade	Sempre

Estudo de caso 3: Tempo de resposta do sistema

O estudo de caso a seguir mostra a métrica do tempo de resposta. Um contrato pode ser modelado de diversas maneiras, cada uma com suas vantagens.

O exemplo a seguir analisa vários métodos de modelagem:

Sugestão de modelagem -Solução A	Valor máximo
Tempo de resposta das transações do sistema CNP	Não se pode exceder mais de 750 milissegundos por mês
Nome da métrica	Tempo máximo de resposta das transações
Destino	750
Período de monitoramento	1 mês
Unidade de medida	Milissegundos
Período de atividade	Sempre
Serviço	Sistema CNP
Domínio do serviço	Desempenho
Categoria do domínio	Tempo máximo de resposta das transações

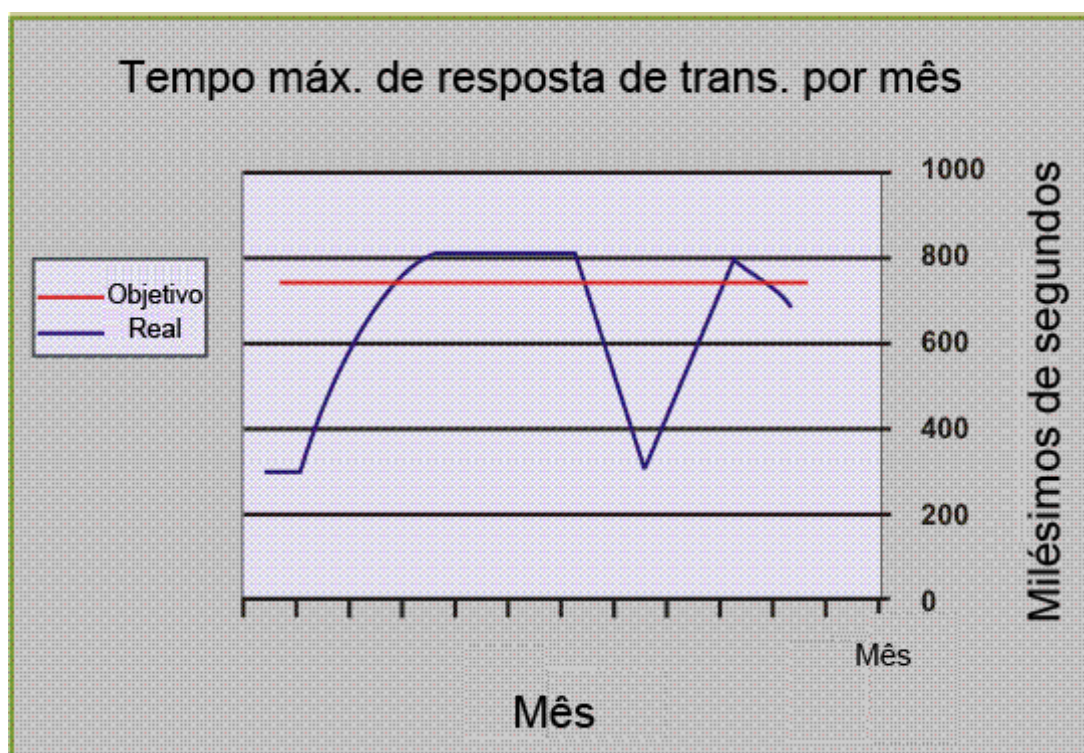
Com base na matriz acima, como o nível do serviço real é calculado?

Com base na definição de categoria de domínio, parece que o nível de serviço real é calculado como um valor máximo. Isso significa que para todas as transações realizadas durante um mês, a transação com o máximo valor é capturado e esse valor é comparado com o objetivo.

Observação: o cálculo do nível de serviço tem como base uma agregação de dados brutos em um determinado período de tempo. Para cada período, a métrica fornece um único resultado. O objetivo de uma métrica não é comparado a uma única transação, mas é comparado ao resultado mensal, que é uma agregação periódica de todas as transações daquele período. O gerente do contrato deve certificar-se de que este resultado reflete o contrato de um lado, e a qualidade de serviço de outro.

Observe que medir o tempo de resposta como um valor máximo é uma obrigação muito restrita e muito difícil de se atingir na prática. Medir um nível máximo significa que uma única transação de 751ms dentre um milhão de transações realizadas durante um mês é o suficiente para violar o contrato. Todas as barras em relatórios serão vermelhas e não refletirão a qualidade real do serviço prestado.

A figura a seguir mostra um relatório típico sob essas circunstâncias.



Qualquer transação que exceder o objetivo será considerada uma violação do contrato, mas como uma base necessária para se compreender a qualidade real do serviço fornecido é muito ruim, ela reflete somente uma única transação e nada se sabe a respeito das outras transações, como, por exemplo, isso foi uma falha única ou uma tendência? Se isso não foi um incidente isolado, quantas falhas existiram ou qual é a proporção de transações com falha em relação ao número total de transações executadas durante o mês? Pode haver alguns meses em que existem tais ocorrências e consequentemente, uma violação de contrato, mas qual é a tendência? Está melhorando ou piorando? Todas essas perguntas podem ser feitas pelo gerente de nível de serviço e o relatório deve fornecer as respostas.

Observação: ao definir a métrica e a estrutura de tópico de cálculo associado, é muito importante determinar como os resultados serão exibidos em um relatório. Este relatório deve fornecer dois itens essenciais:

- Houve uma violação do contrato?
- Ele deve fornecer aos gerentes a transparência dos dados e a capacidade suficiente para investigar a causa raiz da falha, bem como as ferramentas necessárias para entender completamente os componentes do serviço.

Sugestão de modelagem - Solução B

Tempo de resposta das transações do sistema CNP

Métrica

Destino

Período de monitoramento

Unidade de medida

Categoria do domínio

Tempo médio de resposta

Não deve ultrapassar 750 milésimos de segundos por mês

Tempo de resposta médio das transações

750

1 mês

Milissegundos

Tempo de resposta médio das transações

O cálculo do tempo médio de resposta fornece um melhor entendimento sobre a qualidade de serviço mensal e, ao mesmo tempo, é possível refletir esses meses com tempos de resposta extremos ou fora do contrato.

Sugestão de modelagem - Solução C

Tempo de resposta das transações do sistema CNP

Métrica

Destino

Período de monitoramento

Unidade de medida

Porcentagem de transações que foram concluídas com êxito no limite.

Não deve ultrapassar 750 milésimos de segundos por mês

Tempo de resposta das transações bem-sucedidas

100

1 mês

% de êxito

Parâmetro de métrica	750 ms
Serviço	Sistema CNP
Domínio do serviço	Desempenho
Categoria do domínio	Tempo de resposta das transações bem-sucedidas
Período de atividade	Sempre

Ao usar este método, o cálculo será a porcentagem de transações que foram concluídas com êxito sob o limite de 750 ms durante o período especificado, determinado pela fórmula:

$$((\text{Número de transações em 750ms.})/(\text{Número total de transações}))*100$$

Expressar a garantia como uma taxa de sucesso fornece a capacidade de manter uma garantia estrita (meta 100%), e também permite que o valor real represente quão bom ou ruim foi o serviço.

Com esse método, o objetivo agora é um limite máximo de 750 ms, mas é a proporção a ser mantida. Nos casos em que a garantia deve ser estrita, o objetivo deve ser 100%, o que não deixará brecha para nenhuma falha. Observe que neste caso uma variável adicional foi introduzida, o Parâmetro da métrica. Esse parâmetro deve ser implementado como um parâmetro de métrica para permitir modificações simples, se necessário.

Forçar um modelo de tipo de escalonamento pode ser um modelo alternativo para este método:

As seguintes soluções definem as métricas em vez de uma única métrica, como nas soluções anteriores.

Métrica	Tempo de resposta das transações bem-sucedidas
Destino	95
Período de monitoramento	1 mês
Unidade de medida	% de êxito
Parâmetro de métrica	750 ms
Métrica	Tempo de resposta das transações bem-sucedidas
Destino	99
Período de monitoramento	1 mês
Unidade de medida	% de êxito
Parâmetro de métrica	850 ms

Métrica	Tempo de resposta das transações bem-sucedidas
Destino	100
Período de monitoramento	1 mês
Unidade de medida	% de êxito
Parâmetro de métrica	1000 ms

Em uma situação em que é necessário relatar as obrigações contratuais, bem como o número de transações que excedem o limite de 750 ms, deve-se definir uma métrica adicional para contar o número de transações que falharam.

Observação: cada métrica gera um único resultado em um determinado período de tempo. Se ele for definido para calcular a porcentagem de transações, não poderá fornecer o número de transações.

A única forma de gerar relatórios adicionais de uma métrica é usar as saídas da lógica de negócios. (Consulte o tópico [Saídas - Tabelas de usuário](#) (na página 157) que aborda os resultados de saída da lógica de negócios).

Métrica	Número de transações com falha
Destino	Sem destino
Período de monitoramento	1 mês
Unidade de medida	Número de transações
Parâmetro da métrica	750ms
Serviço	Sistema CNP
Domínio do serviço	Desempenho
Categoria do domínio	Número de transações
Período de atividade	Sempre

Estudo de caso 4: Desempenho da Central de atendimento

O estudo de caso mostra uma situação da Central de atendimento

O suporte técnico deve atingir 100% de sucesso dos itens a seguir:

Tipo de ticket	Tempo de resolução
Prioridade 1	1 hora
Prioridade 2	2 horas

Prioridade 3	4 horas
Sugestão de modelagem, solução A:	
Métrica	Tempo de resolução para prioridade 1
Destino	100
Período de monitoramento	1 mês
Unidade de medida	% de êxito
Parâmetro do contrato	Matriz do tempo de resolução
Serviço	Central de atendimento
Domínio do serviço	Desempenho da central de atendimento
Categoria do domínio	Tempo de resolução do ticket
Período de atividade	Sempre

A matriz acima se aplica a três métricas. Para cada prioridade, uma métrica é definida com todas as prioridades dentro da mesma categoria.

Sugestão de modelagem - Solução B

A definição de métrica permanece a mesma, conforme mostrado na solução A.

Opção 1:

Serviço	Central de atendimento
Domínio do serviço	Gerenciamento de tickets de prioridade 3
Categoria do domínio	Tempo de resolução do ticket
Categoria do domínio	Tempo de resposta do ticket
Período de atividade	Sempre

Opção 2:

Serviço	Central de atendimento
Domínio do serviço	Gerenciamento de ticket
Categoria do domínio	Tempo de resolução do ticket de prioridade 3
Categoria do domínio	Tempo de resposta do ticket
Período de atividade	Sempre

Estudo de caso 5: Backup do sistema

O backup é executado da seguinte forma:

Intervalo de tempo	Número de backups
semanal	6
mensal	27
anual	350

Soluções sugeridas:

Métrica	Contagem de backup semanal
Destino	6
Período de monitoramento	1 semana
Unidade de medida	Backups
Serviço	Backup
Domínio do serviço	Backup
Categoria de domínio	Quantidade de backups por semana
Período de atividade	Sempre
Métrica	Contagem de backup mensal
Destino	27
Período de monitoramento	1 mês
Unidade de medida	Backups
Serviço	Backup
Domínio do serviço	Backup
Categoria de domínio	Quantidade de backups por semana
Período de atividade	Sempre
Métrica	Contagem de backup anual
Destino	350
Período de monitoramento	1 ano
Unidade de medida	Backups
Serviço	Backup
Domínio do serviço	Backup

Categoria de domínio	Quantidade de backups por semana
Período de atividade	Sempre

Exemplo de modelagem de métricas financeiras

O estudo de caso a seguir apresenta um exemplo de modelagem financeira.

Estudo de caso 6: Condições de modelagem financeira de um contrato/serviço

Há três tipos gerais de métricas financeiras usadas para criar as condições financeiras de um serviço ou contrato. São elas:

- Custos do preço fixo
- Custos de consumo
- Cobranças de incentivo/penalidade

Veja o seguinte exemplo:

Uma nova empresa é aberta e requer que um serviço de email seja fornecido juntamente com a configuração e manutenção de suas caixas de correio. Como a nova equipe foi contratada, o número de caixas de correio certamente é aumentado. O fornecimento de um serviço de email para um contrato incorre um custo de preço fixo de \$1000, e há um custo adicional por caixa de correio a ser cobrado por mês. Esse custo por caixa de correio é um modelo de preços com vários níveis, como segue:

Número de caixas de correio	Custo por caixa de correio
1-1.000	\$1,00
1,001 - 5,000	\$0,80
5,001+	\$0,50

Desse modo, quanto mais caixas de correio forem adicionadas, menor será o custo adicional. (Por exemplo: 1500 caixas de correio custarão $(1000 \times \$1) + (500 \times \$0.80) = \$1400$. Usando este modelo, duas métricas podem ser produzidas para refletir isso no contrato.

- Custo do serviço de email (fixo)
- Custo do uso da caixa de correio (com base no consumo/variável)

Além disso, há uma estimativa feita pela equipe de gerenciamento do número de integrantes da equipe em todo o ano (2007), conforme indicado a seguir. A tendência se deve ao crescimento inicial da organização em razão da contratação de funcionários e de novos escritórios abertos em outras regiões:

Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
50	100	500	900	1600	1700	1800	2500	2600	3500	3600	5800

Para criar essas métricas, faça o seguinte:

Crie a métrica de custo fixo (use o tipo de item de preço) no contrato, usando os seguintes detalhes:

Geral	Detalhes	Notas	Agrupamento	Limites	Métricas relacionadas	Granularidade	Declaração de objetivo	Registro	Lógica de negócios
Indicador principal: Serviço: Email Domínio de serviço: * Service Consumption → Categoria de domínio: * Number of Mailboxes Unidade de medida: * Dólar norte-americano Período de atividade: * Sempre (Semanal) Fuso horário: * GMT Objetivo do nível de serviço: Valor do objetivo não mais que \$ Destino dinâmico: ☐ Período de monitoramento: * 1 mês/meses Unidade de consumo: Mailboxes Preço por unidade: Adicionar Cálculo de previsão: ☐ Previsão: Adicionar									

Para especificar o custo fixo para o contrato de serviço, implemente essa opção como um parâmetro para a lógica de negócios (em que o custo fixo deve ser retornado da função de resultados). Este parâmetro pode ser exibido por meio da declaração de objetivo da métrica, como mostrado a seguir:

Declaração de objetivo

TheEmail Service Consumption Item de preço 1000 | Dólar norte-americano por 1 mês/meses

Campos

Parâmetros

Adicionar parâmetro

Classificar parâmetros

SERVICE_COST: 1000

Valor mín.: | Valor máx.: | Precisão:

Retornar o valor do parâmetro para essa métrica é simplesmente um caso de retornar o valor do custo de serviço por meio da função de resultados.

Lógica de negócios

```
Function Target
    Target= Null
End Function

Function Result
    If Parameters.IsExist("SERVICE_COST") Then
        Result = Parameters("SERVICE_COST")
    Else
        Result = Null
    End If
End Function
```

Em seguida, crie a métrica de preço variável (novamente use o tipo de item de preço) para determinar os custos de consumo das caixas de correio usadas. Nomeie essa métrica como *Custo de consumo da caixa de correio* e crie-a com os seguintes detalhes:

Geral	Detalhes	Notas	Agrupamento	Limites	Métricas relacionadas	Granularidade	Declaração de objetivo	Registro
Indicador principal:								
Serviço:		Email						
Domínio de serviço: *		Service Consumption						
Categoria de domínio: *		Number of Mailboxes						
Unidade de medida: *		Dólar norte-americano						
Período de atividade: *		Sempre (Semanal)						
Fuso horário: *		GMT						
Objetivo do nível de serviço:		Valor do objetivo não mais que \$ Destino dinâmico: ☐						
Período de monitoramento: *		1 mês/meses						
Unidade de consumo:		Mailboxes						
Preço por unidade:		 						
Cálculo de previsão:		☒						
Previsão:		 						

Nesse caso, é necessário digitar os parâmetros de consumo nos detalhes da métrica. Eles vão para a tabela Preço por unidade. Para criar a tabela acima de número de caixas de correio em relação ao custo, crie uma coluna para o limite máximo de caixas de correio e uma para os preços por unidade.

Adicionar coluna						
		Nome	Tipo	Índice	Padrão	
		Mailboxes	Número decimal	Sim		
		UnitCosst	Número decimal	Não		

Em seguida, digite os valores para cada nível. Nesse caso, o limite máximo de caixas de correio determina o custo de suporte associado a ele. Como não há 3 camadas, elas são adicionadas à tabela desta maneira:

Adicionar linha					
		Mailboxes	UnitCosst		
		1000	100		
		5000	080		
		999999999	05		

Além disso, implemente a função de previsão no consumo de caixas de correio. Faça isso criando a tabela Previsão com o layout mensal predefinido.

Adicionar coluna						
		Nome	Tipo	Índice	Padrão	
		Consumption	Número decimal	Sim		
		Feb-17-2011	Número decimal	Não		
		Mar-01-2011	Número decimal	Não		
		Apr-01-2011	Número decimal	Não		
		May-01-2011	Número decimal	Não		
		Jun-01-2011	Número decimal	Não		
		Jul-01-2011	Número decimal	Não		
		Aug-01-2011	Número decimal	Não		
		Sep-01-2011	Número decimal	Não		
		Oct-01-2011	Número decimal	Não		
		Nov-01-2011	Número decimal	Não		
		Dec-01-2011	Número decimal	Não		
		Jan-01-2012	Número decimal	Não		
		Feb-01-2012	Número decimal	Não		
		Mar-01-2012	Número decimal	Não		

Ele é preenchido com os valores de tabelas fornecidos na descrição do cenário.

Consumo	01-Jan-2007	01-Feb-2007	01-Mar-2007	01-Abr-2007	01-Mai-2007	01-Jun-2007	01-Jul-2007	01-Ago-2007	01-Sep-2007	01-Out-2007	01-Nov-2007	01-Dez-2007
1	50	100	500	900	1600	1700	1800	2500	2600	3500	3600	5800

Agora é possível adicionar a declaração do objetivo para a métrica. Nesse caso, não há valores de parâmetro necessários, pois eles são derivados das tabelas Preço por Unidade e Previsão.

DetalhesNotasAgrupamentoLimitesMétricas relacionadasGranularidadeDeclaração de objetivoRegistroLógica de negócios



TheEmail Service Consumption cost is baseed upon thePreço por unidade table, and is measured inDólar norte-americano por 1 mês/meses . The cost is accumulate are added.

Campos

Parâmetros

Valor de destino:

↑

Unidade de medida:

Dólar norte-americano

↑

Símbolo da unidade:

\$

↑

Período de atividade:

Sempre (Semanal)

↑

Fuso horário:

GMT

↑

Por fim, complete a lógica de negócios, como a seguir:

Option Explicit

```
Dim PPUMap1, PPUMap2, PPUMap3, PPUkey, FCmap, periodFC, TierPPU
Dim currentMonth, TotalMailboxes, MailboxesThisPeriod, TotalPrice
```

```
Sub OnRegistration(dispatcher)
    registro de amostras apenas
    dispatcher.RegisterByMetric "OnMailboxAddedEvent", "NewMailboxEventType", _
        "MailboxResource", "MONTH", "MetricName", "MetricContract", _
        "MetricContractParty"
End Sub
```

```
Sub OnLoad(TIME)
    Inicie os mapas de níveis de preço e os mapas de previsão
    Set PPUMap1 = Context.Field ("Price Per Unit")(1)
    Set PPUMap2 = Context.Field ("Price Per Unit")(2)
    Set PPUMap3 = Context.Field ("Price Per Unit")(3)
    Set FCmap = Context.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)
    Calcule o preço atual de todas as caixas de correio usando o modelo de preços
    em vários níveis
    Isto usa uma abordagem cumulativa, pois passa por cada nível para
    determinar o custo total.
    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
```


Função de previsão

```
Forecast = getMailboxCost (periodFC)
```

End Function

Objetivo da função

```
Target = Null
```

End Function

Function Result

```
result = TotalPrice
```

End Function

Function getforecastvalue(q)

```
getforecastvalue = FCmap (q)
```

End Function

Function getmonth(time)

esta função recupera o mês

```
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)

A função calcula o custo das caixas de correio usando o modelo de preços em vários níveis

```
Dim returnValue
```

```
If num_boxes <= PPUMap1 ("Mailboxes") Then
```

Primeiro nível

```
returnValue = num_boxes * PPUMap1 ("UnitCost")
```

```
'Out.Log "Tier1: " & num_boxes
```

```
Else If num_boxes > PPUMap1 ("Mailboxes") And num_boxes <= PPUMap2 ("Mailboxes") Then
```

'segundo nível apenas

```
returnValue = (PPUMap1 ("Mailboxes") * PPUMap1 ("UnitCost")) + _  
((num_boxes - PPUMap1 ("Mailboxes")) * PPUMap2 ("UnitCost"))
```

```
'Out.Log "Tier2: " & num_boxes
```

```
Else If num_boxes > PPUMap2 ("Mailboxes") Then
```

'terceiro nível

```
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

Observação: o script de lógica de negócios trata tanto o cálculo de previsão (usando a tabela Previsão) quanto os resultados do custo de consumo financeiro. Ambos usam a mesma fórmula `getMailboxCost()`, que calcula o preço com base em vários níveis de acordo com a tabela Preço por Unidade definida para essa métrica.

Exemplos de modelagem de dados

Os dois casos a seguir ilustram o processo básico de modelagem de dados, bem como os melhores pontos que podem estar envolvidos.

Como o processo de modelagem de dados é realizado após o processo de modelagem de contrato, os requisitos do cálculo derivados da garantia do contrato já são conhecidos, foram identificados e definidos como parte do processo de modelagem do contrato.

O modelo de dados deve levar em conta todos os requisitos do cálculo.

Nos dois estudos de caso a seguir, um único ou alguns requisitos são escolhidos para demonstrar o processo de modelagem de dados.

Estudo de caso 7: Desempenho do Servidor

Este é um estudo de caso típico de desempenho do servidor.

Dada a seguinte estrutura da fonte de dados:

Indicação	Servidor	Medida	Carimbo de data/hora
disponibilidade	Appserv01	1	03/01/2001 07:44
tempo de resposta	Appserv01	354.6	03/01/2001 09:58
Carga da CPU	Dbserve02	83%	03/01/2001 12:12
disponibilidade	Appserv01	0	03/01/2001 14:26
Carga da CPU	Dbserve02	94.30%	03/01/2001 16:40
capacidade	Firewall01	10%	03/01/2001 18:54
tempo de resposta	Dbserve02	476.89	03/01/2001 21:08
disponibilidade	Appserv02	1	03/01/2001 21:24
tempo de resposta	Appserv01	774.4	03/01/2001 21:48
Carga da CPU	Dbserve01	83%	03/01/2001 21:52

Além dos itens mencionados acima, os requisitos de cálculo são:

Calcular o percentual de disponibilidade de cada servidor de aplicativos.

A disponibilidade de cada servidor deve ser calculada separadamente. Portanto, para calcular a disponibilidade de um único servidor, é necessário receber eventos somente para este servidor específico. Além disso, as fontes de dados contêm outros indicadores de desempenho que não são relevantes para os cálculos de disponibilidade (tempo de resposta, capacidade, e assim por diante). Assim, os indicadores de disponibilidade e o servidor relevante devem ser filtrados.

Como os critérios de filtragem no CA Business Service Insight são evento e recursos, converta o critério de filtragem a partir dos valores da fonte de dados em uma definição de recurso e evento.

Nesse caso, o indicador é um valor ideal para converter um evento no CA Business Service Insight uma vez que descreve logicamente o tipo de evento. Há um número limitado de tipos, como tempo de resposta, disponibilidade, capacidade e carga da CPU. Isso significa que para as métricas que calculam a disponibilidade do servidor, o registro deve ser o evento de disponibilidade.

Nesse caso, quando houver um grande número de servidores e for necessário calcular a disponibilidade de cada um, cada servidor deve ser definido como um recurso. Ele precisa ser agrupado em um grupo de recursos e a métrica será agrupada naquele grupo de recursos.

Sugestão de modelagem:

Nome do evento	Disponibilidade do evento.
Comportamento	Relatado como a alteração de status de 0 ou 1.
Campo Data e hora	Data e hora (somente o campo na origem de dados).
Campo Recurso	Servidor (cada servidor que é exibido na fonte de dados é convertido em um recurso do CA Business Service Insight).
Campo Evento	Indicador (cada um desses valores desse campo são convertidos em um evento no CA Business Service Insight. Há quatro tipos de evento).
Campos de dados	A medida é 0 ou 1 (somente para registros de disponibilidade).

As alocações de recursos a seguir devem ser definidas:

Atributo do tipo de recurso	Servidor de aplicativos
Alocação para parte contratual	Cada servidor de aplicativo está alocado à parte contratual, em que o servidor relevante executa o aplicativo. Isso permite o registro pela parte contratual que recupera, de modo adequado, todos os servidores.
Alocação para serviço	Igual à anterior.
Alocação para grupo de serviços	Opcional. Geralmente, é necessário agrupar recursos em que o agrupamento é necessário.

E por fim, com base em todas as definições acima:

Registro por

Para a métrica agrupada, calcular a disponibilidade de cada servidor individualmente, o registro é feito por recurso.

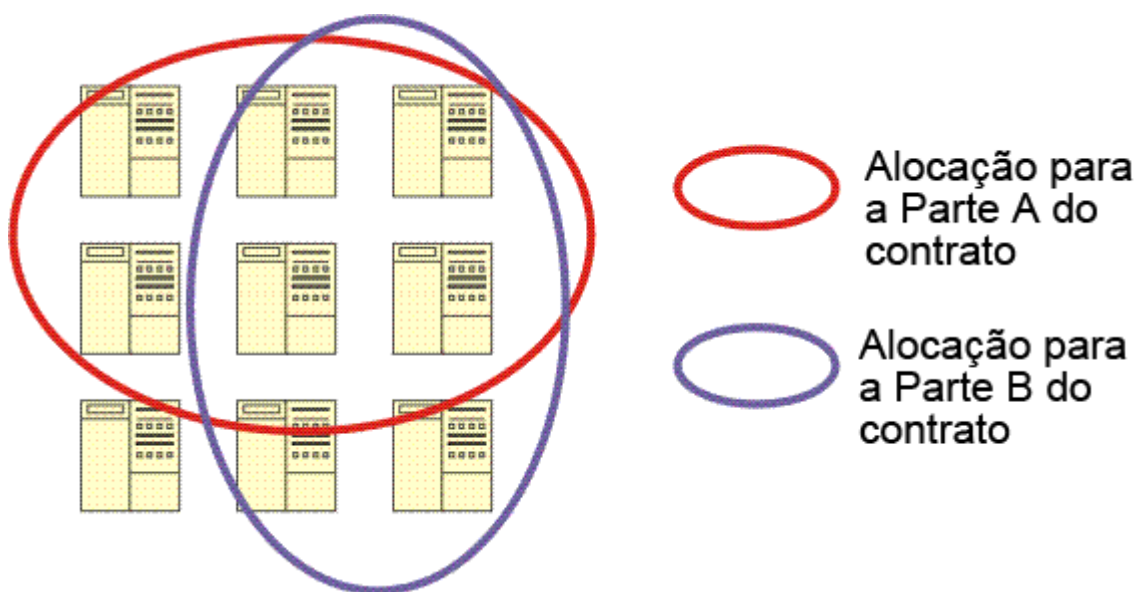
Para que seja possível cumprir o requisito acima, os seguintes critérios foram adicionados:

Calcular o tempo médio de resposta médio dos servidores de aplicativos para cada parte contratual separadamente.

Para esse requisito, é necessário receber eventos de tempos de resposta para todos os servidores de aplicativos que são parte do grupo de servidores que executam aplicativos da parte contratual específica. O recebimento do tempo de resposta dos eventos é feito ao registrar o evento criado no campo de indicação com o valor de tempo da resposta. Isso garante o recebimento somente de eventos que informam os tempos de resposta de servidores.

Para receber apenas os eventos que informam sobre os servidores relevantes de uma parte contratual específica, os recursos precisam ser registrados por meio da alocação da parte contratual.

Um recurso pode ser alocado a mais de uma única parte contratual, serviço ou grupo. Assim, pode ocorrer de um evento enviado para o cálculo de tempo de resposta como parte do contrato da parte contratual A também seja parte dos cálculos da parte contratual B.



Estudo de caso 8: Desempenho da Central de atendimento

Este é um caso de estudo típico sobre o desempenho da Central de atendimento.

Dada a origem de dados mostrada a seguir:

No. do ticket	Ticket Prioridade	Relatado em	Fechado em	Resolvido em	Referência do cliente	Local
3800968	1	19/12/2003 07:51	05/01/2004 11:31	22/12/2003 12:00	CM3	Londres
38000509	1	18/12/2003 09:21	05/01/2004 11:33	22/12/2003 12:00	CM1	Ipswich
38084199	2	07/01/2004 11:20	14/01/2004 09:10	09/01/2004 12:00	CM2	Ipswich
38188329	3	21/01/2004 09:27	27/01/2004 09:09	24/01/2004 12:00	CM3	Leeds
37964069	3	12/12/2003 14:04	05/01/2004 11:35	19/12/2003 12:00	CM286	Birmingham
3796448	1	12/12/2003 14:18	05/01/2004 11:39	19/12/2003 12:00	CM263	Luton
37965039	2	12/12/2003 14:57	14/01/2004 15:05	18/12/2003 12:00	CM264	Leeds
37970699	2	15/12/2003 09:26	05/01/2004 11:22	22/12/2003 12:00	CM288	Londres
37997259	1	17/12/2003 15:58	05/01/2004 11:27	23/12/2003 12:00	CM302	Ipswich
38000259	1	18/12/2003 09:11	06/01/2004 14:44	29/12/2003 12:00	CM340	Londres
38021049	1	22/12/2003 09:32	06/01/2004 14:28	31/12/2003 12:00	CM341	Londres

A origem de dados exibida acima lista os detalhes dos tickets da central de atendimento que são gerenciados para cada cliente em vários locais que o cliente presta serviço. Um único local também pode ser compartilhado entre os clientes.

Os três requisitos a seguir são necessárias para a geração de relatórios ao usar essa fonte de dados:

- Percentual de tickets com prioridade 1 resolvidos em até quatro horas para o cliente CM3.
- Percentual de tickets com prioridade 1 resolvidos em até quatro horas para o cliente CM3 em cada local.
- Percentual de tickets com prioridade 1 fechados em até um dia para o cliente CM3 em cada local.

Os requisitos acima indicam que os eventos devem ser filtrados por:

- prioridade
- cliente
- local

É necessário especificar quais desses critérios de filtragem serão convertidos como um evento e qual será o recurso relevante.

Como selecionar um evento?

Dentre os três critérios de filtragem necessários, a prioridade de ticket é o mais adequado para ser convertido em evento por uma das seguintes razões:

- Ele descreve o tipo de evento que está sendo manipulado (os tickets com prioridade 1).
- O intervalo de prioridades diferentes que existe é relativamente pequeno (prioridade 1,2,3).
- O tipo de evento é relativamente constante (a central de atendimento raramente altera as prioridades dos tickets que ela manipula).

Como um recurso é selecionado?

Os dois outros critérios de filtragem necessários são o cliente e o local, e eles são as menores entidades que exigem a geração de relatórios. Portanto, o recurso consiste na combinação de cliente e a local.

O cliente e o local são entidades relativamente fixas e precisam ter um ciclo de vida definido no qual novos clientes ou novos sites podem ser adicionados. Além disso, o relacionamento entre um site e um cliente pode alterar.

É possível usar mais de um campo na fonte de dados para fins de conversão. Enquanto que no estudo de caso anterior o campo do servidor foi convertido em um recurso do CA Business Service Insight, neste caso o recurso foi criado usando a combinação dos dois campos. Cada permutação gera um novo recurso.

A lista de recursos é mostrada a seguir:

Campo Cliente	Campo Site	Recurso de saída
CM3	Londres	CM3@London
CM1	Ipswich	CM1@Ipswich
CM2	Ipswich	CM2@Ipswich
CM3	Leeds	CM3@Leeds
CM286	Birmingham	CM286@Birmingham
CM263	Luton	CM263@Luton
CM264	Leeds	CM264@Leeds
CM288	Londres	CM288@London
CM302	Ipswich	CM302@Ipswich
CM340	Londres	CM340@London
CM341	Londres	CM341@London

O recurso de saída é uma combinação de campos do cliente e do site usando o símbolo '+' para combiná-los. É importante estar ciente do nome do recurso, pois é extraído da origem de dados e ele aparece posteriormente nos relatórios. A capacidade de geração de relatórios necessária para atender às expectativas.

Observação: um dos erros mais comuns ao modelar uma central de atendimento ou qualquer sistema de ticket ou incidente é definir um único incidente como um recurso.

As suposições a seguir frequentemente levam a erros:

1. O único incidente é o que foi relatado. Não defina a entidade a ser relatada como a entidade para a qual os cálculos são gerados, de modo que um único incidente não seja a base de cálculos para o site do cliente. Em geral, o SLA tem como base o desempenho geral, não o desempenho individual do ticket manipulado.
2. A garantia é por nível de ticket. Isto é um erro, pois as garantias são periódicas, e o que é calculado é a agregação no decorrer do tempo.

Definindo alocações para os recursos

Para o primeiro requisito de cálculo:

1. Percentual de tickets com prioridade 1 resolvidos em até quatro horas para o cliente CM3.
 - Os eventos de tickets atribuíveis a um cliente específico precisam ser recebidos. O cliente é parte desse recurso que também indica o site do cliente. Portanto, para capturar todos os eventos relacionados a recursos e atribuí-los ao cliente, há duas opções:
 - Nos casos em que o cliente representa uma parte contratual na origem de dados, os recursos podem ser anexados à parte contratual relevante. Isso permite o registro pela parte contratual. Esta é uma ação preferível quando for possível.
 - Criar um grupo de recursos para os clientes na origem de dados e agrupar todos os recursos relevantes. Com esse método, o registro é feito por grupo de recursos.

Para os próximos dois requisitos de cálculo:

2. Percentual de tickets com prioridade 1 resolvidos em até quatro horas para o cliente CM3 em cada local.
3. Percentual de tickets com prioridade 1 fechados em até um dia para o cliente CM3 em cada local.

Para esses requisitos, colete os recursos nos grupos de recursos, pois as métricas precisam ser agrupadas, uma vez que o cálculo é necessário para cada site individualmente.

Observação: mesmo que as alocações de recursos atuais para o modelo e os requisitos não sejam necessários, é importante criá-los para requisitos futuros. Isso evita a sobrecarga no desenvolvimento futuro do sistema.

Selecionando o campo Data/hora

Conforme mencionado anteriormente, a data/hora é muito importante para o modo como o mecanismo de correlação manipula os eventos. As métricas sempre calculam o nível de serviço por período de tempo, e os eventos que são processados nesse período são aqueles em que a data/hora está dentro do período.

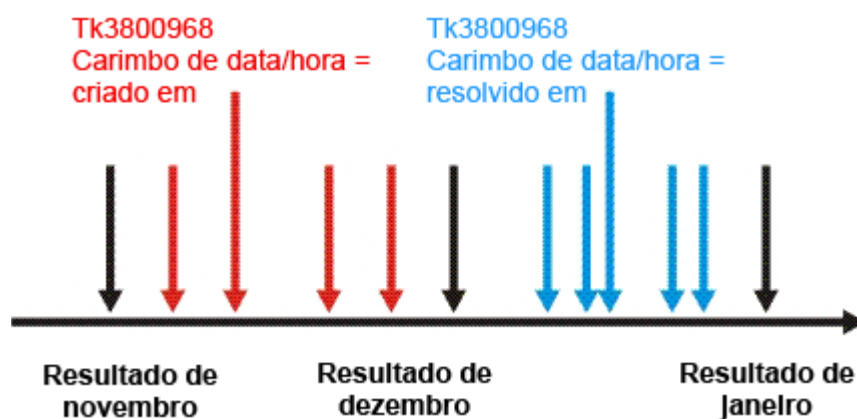
No primeiro estudo de caso, a origem de dados tem somente um campo de horas. No entanto, neste caso, há três campos possíveis que podem ser definidos como data/hora. Analisando os dois primeiros registros:

No. do ticket	Ticket Prioridade	Relatado em	Fechado em	Resolvido em	Referência do cliente	Local
3800968	1	19/12/2003 07:51	05/01/2004 11:31	22/12/2003 12:00	CM3	Londres
38000509	1	18/12/2003 09:21	05/01/2004 11:33	22/12/2003 12:00	CM1	Ipswich

Para calcular o tempo de resolução é necessário a hora de geração e de resolução. Para os fins do evento, é possível anexar somente um carimbo de data e hora a ele. O outro pode ser usado como um valor nos campos de valor.

Se o valor Criado em for usado como o carimbo de data e hora, o ticket será incluído nos resultados de dezembro. Se o valor Resolvido em for usado como o carimbo de data e hora, o ticket será incluído nos resultados de janeiro. As duas opções são viáveis. A seleção precisa apenas atender às expectativas com relação ao local em que os tickets devem ser exibidos nos relatórios.

É um ponto muito importante a ser considerado durante a implementação, pois ele determina quando os eventos serão usados nos cálculos. Por exemplo, se um ticket foi gerado em novembro, mas não foi fechado até dezembro, quando ele deverá contribuir para o resultado de SLA? Ele será contabilizado nos dados de novembro ou dezembro?



Nesse caso, uma vez que o ticket é relatado à origem de dados somente depois de seu fechamento, os dados poderão ser capturados somente após o fechamento do ticket. Em geral, a data de fechamento é posterior às datas de geração e resolução do ticket. No caso em que a data de geração do ticket foi escolhida para ser a data e hora, ela deve ser processada como parte dos resultados de dezembro. A data de fechamento foi em janeiro, o que significa que dezembro já tinha passado quando este ticket foi relatado. Assim, os resultados de dezembro já foram publicados. O mecanismo de correlação avisa que o evento estava no passado, pois o carimbo de data e hora pertence a dezembro e aciona o recálculo. Portanto, os resultados de dezembro se alteram de forma retroativa.

Essas consequências precisam ser entendidas completamente para que seja possível definir qual campo deve ser escolhido como data/hora. Em geral, a data de fechamento é selecionada para obter relatórios finais que não se alteram de forma retroativa.

Por outro lado, usar a data de fechamento como data/hora atrasa a inserção dos tickets nos cálculos. Um ticket que foi resolvido só pode ser fechado algum tempo mais tarde.

Lembre-se, no entanto, de que o uso de data de fechamento também pode disparar um processo na organização que acelera a hora de fechamento do ticket.

A última sugestão é:

Nome do evento	Ticket com prioridade 1 (também pode ser definido para outras prioridades, se necessário)
Comportamento	Relatado quando o ticket é fechado
Campo Data e hora	Fechado em
Campo Recurso	campo cliente + campo site
Campo Evento	Prioridade
Campos de dados	Tudo
Atributo do tipo de recurso	Site da parte contratual
Alocação para parte contratual	Cada site é alocado à parte contratual relevante
Alocação para serviço	Igual à anterior
Alocação para grupo de serviços	O grupo de recursos é criado para cada parte contratual para ativar o agrupamento
Registro por	Para as métricas agrupadas, o registro é feito por recurso e métrica está vinculada a um grupo de recursos chamado sites do cliente CM3 Para métricas da hora de fechamento, o registro é feito por parte contratual e serviço

Usando exemplo de atributos personalizados

O estudo de caso a seguir apresenta um exemplo para vários objetivos dinâmicos.

Estudo de caso 9: Vários objetivos dinâmicos

Considere o exemplo de um cenário onde todos os dispositivos de infraestrutura de hardware em um ambiente de cliente tenha objetivos individuais definidos conforme os requisitos de disponibilidade. Usar a abordagem de modelagem padrão é uma tarefa muito difícil de atingir e envolveria vários agrupamentos lógicos para os dispositivos e gerenciamento usando o modelo de recurso. Para adicionar à complexidade, os objetivos desses dispositivos podem mudar ao longo do tempo. Esses valores alvo são atualizados no CA Business Service Insight por um script de conversão conforme os detalhes são armazenados em um CMDB externo (consulte o tópico [Exemplos de melhores práticas de script de conversão](#) (na página 247) para obter o exemplo do script de conversão)

Nessa instância, a métrica pode ser:

O percentual de disponibilidade por dispositivo de hardware.

Um modo de modelar efetivamente isso é usar o recurso Atributos personalizados junto com um dos outros recursos principais Destinos dinâmicos Ambos podem ser usados com uma métrica agrupada para atingir os resultados desejados. A adição do objetivo do nível de serviço diretamente ao recurso permite que a lógica de negócios compare os níveis de serviço de cada recurso (dispositivo de hardware) com seu próprio objetivo. Uma métrica agrupada fornece a conformidade do serviço individual de cada parte do hardware usando uma única métrica.

Portanto, é necessário criar primeiro o atributo personalizado adicionando-o ao tipo de recurso destes dispositivos (em que todos os dispositivos são um recurso do tipo Dispositivo de infraestrutura). O atributo personalizado criado é chamado de DeviceTarget e pode ser adicionado a partir do menu em *Catálogo de serviço > Atributos personalizados*. Observe que ao criar o atributo personalizado, é necessário vinculá-lo ao tipo(s) de recurso que precisam dele.

Detalhes do atributo personalizado

Entidade	Tipo	Obrigatório	Valor padrão
Recursos	Infrastructure Devices	Sim	

Agora, ao exibir os recursos no sistema, é possível ver que o novo atributo personalizado está disponível para o tipo de recurso a que foi vinculado.

Recursos

Localizar Adicionar

Exibindo 1 - 4 de 4 resultados.
(Limitado a 1000)

Resultados aplicados a 03/08/2012 20:26

		Nome	Data de criação	Tipo de recurso	Grupo de recursos	Serviço	Parte contratual	DeviceTarget	Fuso horár	Conj
☐		Server111	03/08/2012 10:46	Infrastructure Devices	Infrastructure				GMT	Defau
☐		Server222	03/08/2012 10:56	Infrastructure Devices	Infrastructure				GMT	Defau
☐		Server333	03/08/2012 10:56	Infrastructure Devices	Infrastructure				GMT	Defau
☐		Server444	03/08/2012 10:57	Infrastructure Devices	Infrastructure				GMT	Defau

Resource Type

Custom Attribute Added

Além disso, os recursos individuais têm um campo que pode ser atualizado.

Detalhes do recurso

Geral **Detalhes**

Adicionar Filtrar

Dados

01/01/2006 00:00

recurso: * Infrastructure Devices

g11n-RT1

Adicionar tipo de recurso

Grupos de recursos: Selecionado Infrastructure Disponível g11n-RG1
plsSanity_RGxx

Serviços: Selecionado Disponível Apps Hosting Bronze
Apps Hosting Gold
Apps Hosting Silver
Customer Support

Partes contratuais: Selecionado Disponível g11n-CP1
g11n-CP2
g11n-CP3
g11n-CP4

Custom Attribute Added

DeviceTarget: * 98

Nesse exemplo, este campo normalmente será inserido/atualizado pelo script de conversão.

Agora que os recursos têm um objetivo especificado, é possível desenvolver a lógica para executar o cálculo necessário (após confirmar as alterações do recurso). O código de amostra a seguir mostra como extrair o valor de atributo personalizado do recurso (em negrito).

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
```

```
Objetivo da função
    Target = eventDetails.CustomAttribute ("DeviceTarget")
End Function
```

```
Function Result
    If TotalTime > 0 Then
        Result = (TotalTime - OutageTime) / TotalTime
    Else
        Result = Null
    End If
End Function
```

Exemplos de script de conversão

Os estudos de caso a seguir incluem exemplos de conversão automática básica e as atualizações do modelo de recursos.

Estudo de caso 10: Conversão automática básica

O exemplo de script de conversão fornecido aqui é um caso muito simples de processamento de entradas pendentes na tela de entradas de conversão.

O manipulador `OnTranslationEvent` executa uma verificação simples no primeiro caractere do recurso e executa uma ação de acordo com o valor: se a entrada de conversão 'a' estiver definida como ignorar, se 'b' for excluída, 'c' será convertida, ou caso contrário, permanecerá inalterada para ser convertida manualmente. Observe que os contadores de todo o código controlam as ações que serão executadas durante a execução do script. Isso é muito útil para depuração ou documentação das execuções de script sempre que ele for executado, especialmente quando o script for automatizado. O comando *Tools.Commit* no final da função é muito importante, pois sem ele nenhuma alteração feita pelo script será salva no banco de dados.

A função `TranslateResource()` verifica se existe no sistema um recurso com o mesmo nome de outro transmitido a ele por meio da entrada de conversão pendente (juntamente com o prefixo E2E-). Se não existir, o script adiciona este recurso e executa a conversão. Se ele já existir, ele criará uma entrada de conversão da cadeia de caracteres do recurso ao recurso existente do CA Business Service Insight.

A função final resultado do script envia uma descrição das tarefas executadas pelo script. O código é conforme segue:

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
```

Estudo de caso 11: Atualizações do modelo de recurso

Outro uso de scripts de conversão é a atualização do modelo de recurso do CA Business Service Insight com os valores obtidos de uma origem de dados externa. Embora não esteja mais estritamente relacionado à conversão, esta amostra é um valioso recurso para executar atualizações automáticas no sistema.

A seção anterior sobre atributos personalizados descreveu um cenário em que os dispositivos de infraestrutura de hardware de uma organização foram armazenados em um CMDB externo, juntamente com um objetivo de disponibilidade esperado para cada dispositivo. Esta informação deve ser replicada no modelo de recurso do CA Business Service Insight de forma a manter o mapeamento da infraestrutura (e os objetivos do dispositivo) atualizado.

Nesse caso, o script é necessário para executar as seguintes tarefas:

- Adicionar novos dispositivos de hardware que não existem atualmente no sistema.
- Atualizar o objetivo de disponibilidade esperado de dispositivos já existentes (se o objetivo for alterado).

O script é conforme segue:

Option Explicit

```
'*****
'Variáveis globais e constantes
'*****

Dim added
Dim updated
Dim ChangeSetName

added = 0
updated = 0

Const RESOURCE_TYPE = "Infrastructure Devices"
Const RESOURCE_GROUP = "InfraServers"
Const CHANGESSET_NAME = "Infrastructure Devices"
Const CHANGESSET_EFFECTIVE_DATE = "01/01/07"

'*****
'Sub OnLoad :
'Preparar as entidades da infraestrutura base
'*****

Sub OnLoad()
    Tools.log "Translation Script : In OnLoad procedure", "D"

    'Pesquisar a versão preliminar de recursos existentes
    Dim ChangeSetMap
```

```

Set ChangeSetMap=Tools.SearchChangeSets(CHANGESET_EFFECTIVE_DATE,
CHANGESET_NAME)

Se não existir nenhuma versão, crie uma
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")
        Adicionar recursos à versão mais recente, se eles não existirem
        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
    End While
End Sub

```

```
        Tools.Commit
        rs.MoveNext
    Loop

    rs.Close
    conn.Close
    Set rs = Nothing
    Set conn = Nothing
End Sub

Function Result
    'Confirmar a transação
    Tools.CommitChangeSets CHANGESET_NAME
    Result = added & " resources added and " & updated & " resources updated."
End Function

'*****
```

Exemplos de script de lógica de negócios

Veja, a seguir, algumas diretrizes gerais para o script de lógica de negócios.

Variáveis globais

- Certifique-se de assinar os valores globais que você declarar. O mecanismo de estado PSL não pode salvar variáveis com valores nulos

Codificação geral

- Antes de usar um dos objetos listados abaixo, verifique se ele existe (usando um método `IsExist` adequado):
 - Todos os tipos de parâmetro (por exemplo, tabelas, listas, etc.)
 - Atributo personalizado
 - Recurso
- Verifique se você forneceu um módulo de lógica de negócios com todos os parâmetros necessários
- Antes de alterar o nome de um recurso, verifique quais métricas estão utilizando-o e faça a atualização adequadamente.

Mapas

- Uso de mapas em métricas agrupadas: os mapas requerem mais esforço de cálculo do mecanismo. Certifique-se de que ao agrupar em uma métrica você multiplique o esforço pelo número de itens de agrupamento.
- Grandes mapas globais na lógica de negócios de métricas agrupadas devem ser usados somente depois de uma verificação elaborada. Enquanto o mecanismo está executando o cálculo, uma métrica agrupada está ocupada carregando as variáveis globais de estados para cada item no agrupamento separadamente
- Não se esqueça de limpar os mapas e vetores após terminar de usá-los
- Se for necessário usar mapas grandes, certifique-se de gerenciar o mapa de maneira eficiente dividindo-o em intervalos lógicos.

Estudo de caso 12: Usando a lógica do contador para calcular o número de falhas

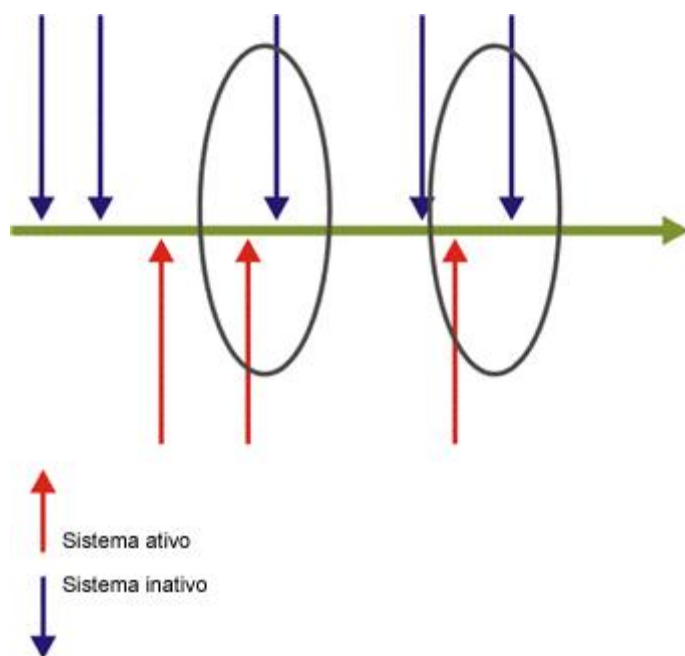
O exemplo a seguir calcula o número de falhas ocorridas em um determinado período do cálculo. A fórmula e os métodos usados para implementar podem ser usados como um exemplo de uma fórmula obrigatória sempre que for necessário calcular alguma coisa.

As suposições de cálculo a seguir são usadas:

- Eventos de entrada:
 - Disponibilidade do evento, status (0/1)
 - O evento de disponibilidade chega com intervalo de alguns minutos. Não foi possível presumir a frequência do evento (o evento pode ocorrer a cada 5 minutos ou uma vez por hora) e também pode haver eventos redundantes (por exemplo: 0, 0, 0, 1, 1, 0, etc.).
- Período de atividade - horário comercial; as falhas que ocorrerem fora do período de atividade não são contadas.
- O resultado obrigatório é a contagem de quantas falhas ocorreram durante o período.'

Para calcular as falhas ocorridas durante o cálculo do período é necessário armazenar uma variável de contador periódico, bem como uma variável que armazena o status do sistema. Uma vez que as informações redundantes de eventos podem ser recebidas (ou seja, um evento Ativo seguido por outro evento Ativo), também é necessário contar o número de locais nos quais ocorreu uma alteração de status do sistema de Ativo para Inativo e não contar cada vez que um evento Inativo é recebido, pois isso pode ser um evento redundante que representa uma falha que já foi registrada.

A figura a seguir ilustra graficamente a contagem de horas ativas e inativas do sistema.



Pontos importantes que devem ser considerados:

- **Constantes** - é recomendado o uso de definições constantes, em vez de valores constantes no código. Dessa forma, se o valor precisar de alteração, é fácil alterá-lo apenas na definição constante e não é preciso pesquisá-lo em todo o código. Além disso, é fácil alterá-lo a um parâmetro, se necessário. No exemplo anterior, os valores que representam o status do sistema recebido no evento são definidos como constantes a fim de tornar o código mais compreensível, e também para diferenciar quando o número zero foi usado como um número para fins de contagem em vez de quando ele representa um status do sistema.
- **Variáveis globais:**
 - **Contador** - a definição da variável de contador é uma definição global. Na fórmula, isto é declarado no início do código e fora de quaisquer subrotinas ou funções. É importante definir a variável de contador nesse caso como uma variável global. Isso permite que ele seja usado em várias subrotinas/funções no código, e também permite que o sistema mantenha seu valor durante todo o período do cálculo e forneça seu resultado no final do período.

Neste exemplo, a variável de contador precisa ser usada em três locais no código:
 - Para ser inicializada no início de um período.
 - Para ser antecipada no caso de uma falha no evento no manipulador de eventos.
 - Para ser enviada pela função de resultados no final do período.
 - **Status anterior do sistema** - esta variável contém o status do sistema e será atualizada sempre que um novo evento com o status do sistema for recebido. Essa variável também precisa ser global, pois é usada em algumas subrotinas/funções no código, como:
 - Para ser inicializada no início do cálculo.
 - Para ser atualizada quando um evento for recebido.
- **Considerações sobre o período de atividade** - para verificar se uma falha ocorre em um período de atividade, é possível usar o valor da propriedade *context.IsWithinTimeSlot*. O contexto é um objeto global que pode ser usado em qualquer lugar no código e, nesse caso, é usado para indicar quando uma falha é recebida e se ela está dentro ou fora do período de atividade. Se na data e hora do evento o sinalizador retornar o valor TRUE, significa que o evento ocorre dentro do período de atividade; no entanto, se ele retornar o valor FALSE, indica que o evento ocorreu fora do período de atividade. De acordo com a lógica necessária, qualquer falha que ocorrer fora do período de atividade é ignorada e, portanto, não é contabilizada.
- **Iniciar variáveis:**

- **Variável de contador** - contém um valor periódico e é iniciada no manipulador *OnPeriodstart* para garantir que cada período de cálculo conte as falhas separadamente. Cada período se inicia com zero e registra apenas o número de falhas desse período.

Nos casos em que é necessário calcular as falhas acumuladas em cada período (o que significa que o resultado de cada período consiste em todas as falhas ocorridas até o fim deste período, incluindo todos os períodos antes disso) é necessário inicializar o contador apenas no manipulador *OnLoad* e removê-lo do manipulador *OnPeriodStart*. Assim, o contador continua a contagem e a acumulação entre os períodos, como mostrado a seguir:

```
Sub OnLoad(time)
    FingerprInted=0
End Sub
```

- **Variável de status do sistema** - deve ser iniciada uma vez no início do cálculo e, a partir desse ponto, ser atualizada em cada evento de status. Essa variável retém seu valor em todo o cálculo e não está restrita a um determinado período do cálculo. Ela também deve conter seu valor entre os períodos de cálculo. Como o status do sistema antes de receber o primeiro evento é desconhecido, deve-se fazer uma suposição para o status do sistema. A suposição comum é de que o sistema está Ativo. Essa suposição deve ser definida de comum acordo e verificada, pois ela pode conduzir ao seguinte:

Se quando o cálculo for iniciado, o status do sistema estiver, de fato, Inativo e o primeiro evento chegar para indicar esse status, ele será contado como uma falha, pois o status presumido era Ativo. A falha será contabilizada no primeiro período do cálculo, mesmo se não ocorreu necessariamente durante esse período.

```
Option Explicit
```

Definições de constantes

```
Const UP=1
Const DOWN=0
```

Variável global de contagem de falhas

```
Dim FingerprInted
Dim SystemStatus
```

```
Sub OnRegistration(dispatcher)
    'Os parâmetros de método são: <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)
    Se ocorrer uma falha e estiver dentro do período de atividade, ela será
    contabilizada
    If context.IsWithinTimeSlot and SystemStatus=UP and _
        eventDetails("Status")=DOWN Then
        FingerprInted = FingerprInted + 1
    End If
    'atualizar o status do sistema para a próxima comparação
    SystemStatus = eventDetails("Status")
End Sub

Function Result
    Result = FingerprInted
End Function
```

Estudo de caso 13: Tratamento do grupo dinâmico do componente

É frequentemente necessário armazenar valores para um grupo de recursos em que os integrantes deste grupo podem ser dinâmicos e alterados durante o período do cálculo. No exemplo cálculo obrigatório a seguir, é necessário executar um cálculo intermediário em cada um dos recursos para obter o resultado agregado final.

A seguir estão alguns exemplos:

- **incidentes em sites** - calcula a porcentagem de sites com os incidentes que levaram mais de X horas para serem solucionados, ou conta os sites que tiveram incidentes com média de tempo de resolução maior que X.

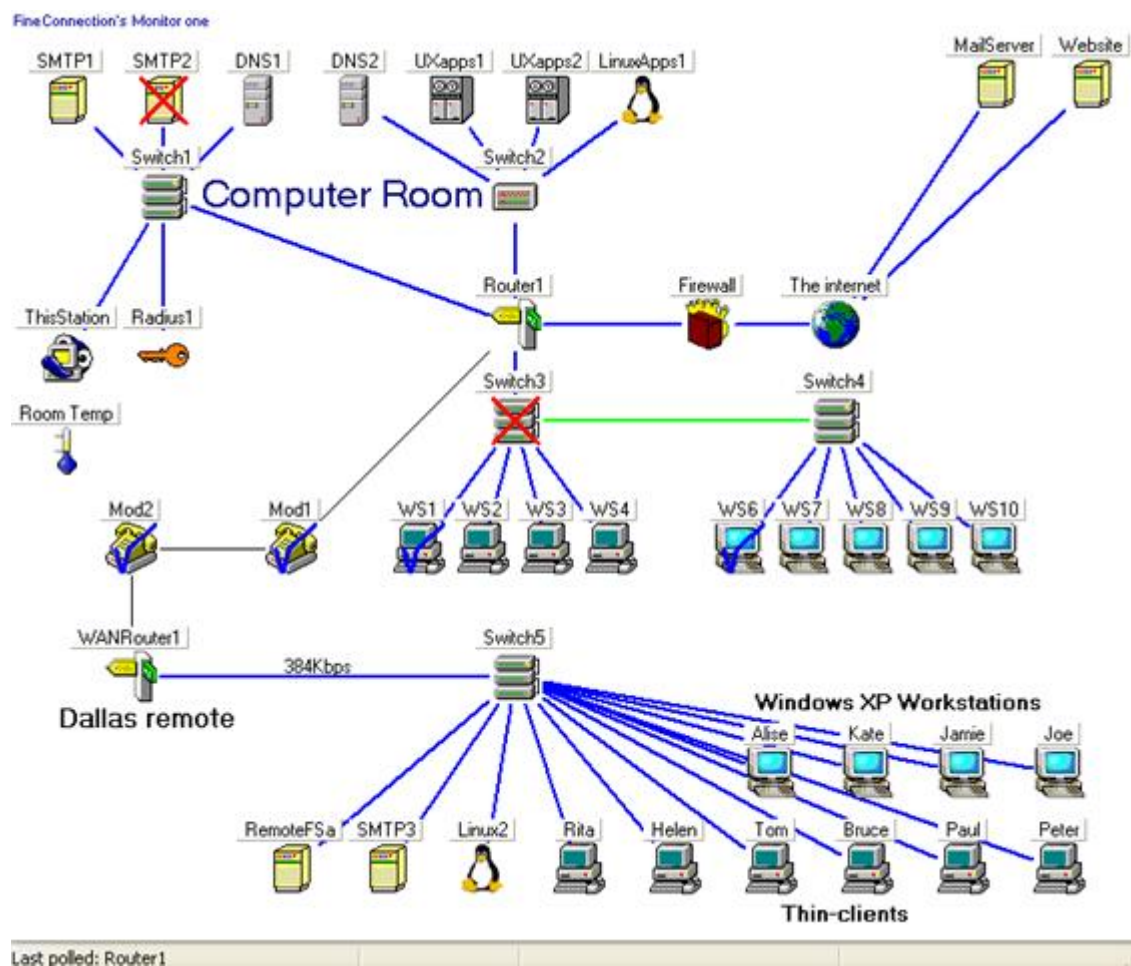
Nestes exemplos, sites são recursos que têm incidentes associados a eles.

- **Disponibilidade do servidor** - conta o número de servidores cuja disponibilidade foi maior que X%.

Um servidor é um recurso cujo percentual de disponibilidade deve ser avaliado.

- **Tipos de transação** - calcula o percentual dos tipos de transações que falharam mais de X vezes.

Nesse caso, um tipo de transação é um recurso com eventos de falha associados a ele. Para cada tipo de transação, um contador de falhas é armazenado como um resultado intermediário e a quantidade de diferentes tipos de transação com mais de X falhas é contabilizada.



Exemplo:

Para o requisito de cálculo a seguir:

Calcular a porcentagem de disponibilidade de um sistema que é composto de um agrupamento de servidores. O sistema é considerado disponível somente quando todos os servidores subjacentes estiverem disponíveis.

O design da solução será implementado como segue:

A disponibilidade do sistema é avaliada pela disponibilidade de seus recursos de agrupamento adjacentes. Nesse caso, é necessário gerenciar e armazenar o status de todos os recursos agrupados para avaliar o status do sistema o tempo todo.

Para fazer isso, a fórmula é necessária para manter um mapa (mapa ServerAvailabilityIndicators no código de exemplo a seguir) que tem uma entrada para cada um dos recursos monitorados. Como todos os objetos do mapa requerem um campo chave para fazer referência ao valor associado, este mapa terá o indicador de recursos como a chave (que é a ID do recurso) e o valor será o componente de status. Sempre que o status de um componente é alterado, a entrada relevante do mapa deve ser alterada. De acordo com o evento de disponibilidade, a fórmula atualizará o status do recurso relevante no mapa e irá reavaliar o status do sistema adequadamente.

Uma vez que os recursos monitorados podem ser alterados (alguns podem ser removidos e outros podem ser adicionados durante o período de cálculo), isto deve ser gerenciado dentro da fórmula adicionando uma função que identifica a alteração e atualiza o mapa do componente monitorado por meio da adição de uma nova entrada para um novo componente ou da exclusão de uma entrada, se um componente foi removido.

O OnRegistration é o manipulador de eventos que gerencia um registro de evento e é acionado quando ocorre uma alteração nos recursos monitorados. Tais alterações podem ocorrer como resultado da confirmação de um recurso (ou o conjunto de alterações) que pode gerar alterações nos recursos incluídos no cálculo de acordo com o método de registro da fórmula.

Durante cada registro é necessário atualizar o mapa que contém o status do recurso com as alterações necessárias. Ou seja, comparar o mapa usado para manter o status do recurso com o mapa que contém os recursos no momento em que o registro é executado (com base no método de registro), bem como incluir todos os recursos que foram adicionados ou excluir os recursos que foram removidos.

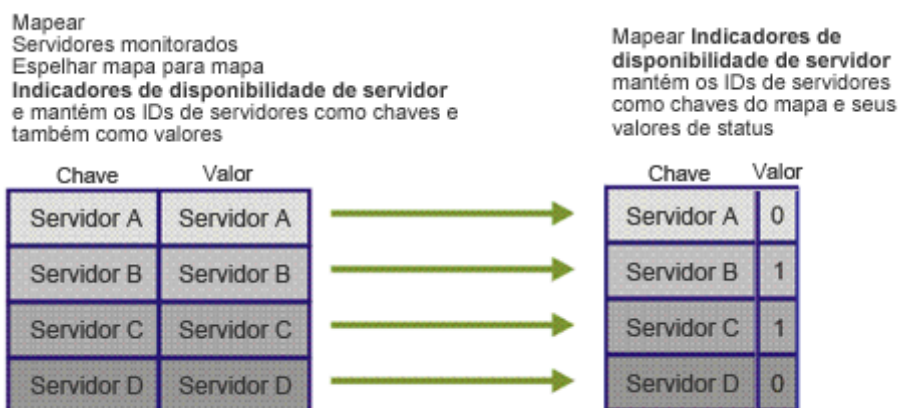
O procedimento do OnRegistration deve executar uma função que compara os recursos monitorados com os novos recursos alocados a fim de estruturar os recursos monitorados adequadamente.

O objeto Contexto tem um conjunto de métodos que compara os métodos de registro. Eles retornam um mapa com todos os recursos que são parte dos recursos de acordo com o método de registro.

No exemplo a seguir, o registro da fórmula é `ByContractParty` e o mesmo método é usado pelo `Context.ResourcesOfContractParty`. Isso retorna um mapa com todos os recursos que são parte deste conjunto no momento do registro.

Para comparar os dois mapas, é necessário realizar a iteração por meio dos mapas em paralelo. A iteração dos mapas é feita usando a declaração `For each`. Essa declaração permite a iteração dos valores de um mapa. Assim, é necessário outro mapa como um espelho ao mapa de status para poder fazer a iteração dos recursos e não de seus valores de status. Essa declaração reitera os valores de um mapa e não de suas chaves. Portanto, é necessário um mapa adicional que contenha as IDs como uma chave e também como um valor. Além disso, mantenha o mapa de espelhamento para refletir, de modo contínuo, o mapa de status, de modo que ele sempre contenha o mesmo conjunto de recursos. Isso significa que quando o mapa de status for atualizado, o mapa de espelhamento também deverá ser atualizado.

A figura a seguir mostra o conceito dos mapas de espelhamento.



Exemplo:

```

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)
    Servidor Dim
    Para cada servidor no allocatedServers
        If Not MonitoredServers.Exist(Server) Then
            MonitoredServers(Server) = Server
            ServerAvailabilityIndicators(Server)=True
        End If
        Próximo

    Para cada servidor no MonitoredServers
        If Not allocatedServers.Exist(Server) Then
            MonitoredServers.Erase Server
            ServerAvailabilityIndicators.Erase Server
        End If
        Próximo
    End Sub

```

Exemplo:

A função a seguir mostra como o mapa de espelhamento é usado para iteração do mapa de status quando necessário para avaliar o status de todo o sistema com base no status de cada recurso monitorado.

Neste exemplo, o sistema é considerado disponível se todos os recursos estiverem disponíveis. Ter um único componente desativado é suficiente para considerar o sistema desativado:

```

Function SystemAvailability
    Servidor Dim
    Para cada servidor no MonitoredServers
        If ServerAvailabilityIndicators(Server) = DOWN then
            SystemAvailability=DOWN
            Função de saída
        End if
        Próximo
    End Sub

```

End Function

Um exemplo completo de lógica de negócios com tratamento dinâmico do recurso é descrito no exemplo de padrões de design a seguir.

Estudo de caso 14: Controle do relógio de acumulação de tempo

O padrão de design descrito nesta seção é adequado sempre que o resultado obrigatório for uma função de um período que tenha expirado entre eventos, por exemplo:

- Tempo disponível calculado como o tempo passado entre uma falha e outra.
- Tempo de resolução calculado como o tempo entre a abertura e o fechamento de incidentes.

Para acumular tempo, é necessário atribuir uma variável na qual o tempo pode ser acumulado (em segundos) e implementar uma função que verifica as condições e o tempo acumulado desde a hora da última atualização transcorrida. Esta função é executada para cada evento recebido na fórmula.

A figura a seguir demonstra o relógio que trata do acúmulo de tempo.



A variável `LastUpdateTime` armazena a última hora em que a atualização foi executada, independentemente se o contador de tempo foi atualizado ou não. A função possui a condição que determina se a hora deverá ser atualizada e acumulada. Por exemplo, o tempo não deve ser considerado se exceder o período atividade, se o status do sistema estiver Inativo ou se o incidente tiver um status pendente.

Embora a situação detalhada aqui normalmente usa a função `Tools.NetTime()` para calcular o tempo, poderá haver casos em que é preferível usar a função VB padrão `DateDiff()`.

A função `Tools.NetTime` provoca uma sobrecarga de verificação do período de atividade cada vez que é usada. É recomendável evitar o uso de `NetTime` nos procedimentos do evento de dados, pois eles são chamados para qualquer novo evento de entrada e, portanto, invoca a chamada `NetTime`. Se o período de atividade for 24/7, recomenda-se que você use a função `DateDiff` para evitar a sobrecarga de verificação do período de atividade.

Exemplo 1:

A rotina dos contadores de atualização a seguir acumula o período de tempo total na variável `PeriodNetTime`. Em `AvailabilityTime`, a rotina acumula a hora em que o status do sistema estava Ativo, significando que o sistema estava disponível.

O objeto `Tools` contém o método `NetTime`, `NetTime(beginTime, endTime)`. Este método retorna o número de segundos entre `beginTime` e `endTime` que estiverem dentro do período de atividade da métrica atual. Qualquer hora entre essas duas variáveis é parte do período de atividade excluído, por isso, é muito usada para os cálculos de duração em que um período de atividade é usado. (Por exemplo: para tickets com prioridade 1 resolvidos em até quatro horas do horário comercial, mesmo que o ticket tenha sido gerado no fim de um dia útil, e não puder ser solucionado até a próxima manhã, ele ainda estará dentro do SLA, devido às horas do período de atividade a ser excluído).

```
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
```

Exemplo 2:

O exemplo a seguir calcula a disponibilidade de aplicativos tratando de eventos ativos e inativos de vários componentes essenciais, bem como eventos de manutenção desses componentes. Se todos os componentes estiverem em manutenção, o tempo não é considerado como tempo de disponibilidade esperado.

A subrotina UpdateCounters avança os contadores de tempo quando necessário e é executada com cada evento recebido para a fórmula (evento de dados brutos/ evento do mecanismo). Ele também atualiza a hora disponível esperada nos casos em que a hora estiver dentro do período de atividade e os componentes não estiverem dentro de um período de inatividade planejado. A fórmula atualiza o tempo de disponibilidade real somente quando também tiver o status do sistema como Ativo.

O DateDiff é uma função VB padrão que retorna a hora entre duas datas, mas não exclui nenhuma informação do período de atividade.

'Forçar a declaração da variável

Option Explicit

Variáveis globais

Dim ExpectedAvailabilityTime

Dim ActualAvailabilityTime

Dim LastUpdateTime

Dim AvailabilityIndicators

Dim MonitoredComponents

Dim DowntimeStatuses

'Mapear criação de objetos

Set AvailabilityIndicators=CreateObject("SlalomMap.Map")

Set MonitoredComponents=CreateObject("SlalomMap.Map")

Set DowntimeStatuses=CreateObject("SlalomMap.Map")

Após o carregamento e sempre que ocorrer uma alteração na infraestrutura

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")

Certifique-se de que a fórmula está monitorando apenas e todos os recursos
relevantes

UpdateMonitoredComponents AllocatedComponents

End Sub

Sub OnLoad(time)

Quando o sistema for ativado pela primeira vez, supõe-se que a disponibilidade
está OK

```
LastUpdateTime = time
End Sub

Sub OnPeriodStart(time)
    inicializar contadores para renovar cálculos periódicos
    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
    'gravar status de disponibilidade do recurso relatado
    AvailabilityIndicators(eventDetails.ResourceId) = _
        AvailabilityIndicators(eventDetails.ResourceId)+1
End Sub

Sub OnComponentUpEvent(eventDetails)
    UpdateCounters eventDetails.Time
    'gravar status de disponibilidade do recurso relatado
    AvailabilityIndicators(eventDetails.ResourceId)= _
        AvailabilityIndicators(eventDetails.ResourceId)-1
End Sub

Sub OnMaintenanceStartEvent(eventDetails)
    UpdateCounters eventDetails.Time
    'gravar status de disponibilidade do recurso relatado
    DowntimeStatuses(eventDetails.ResourceId)= _
        DowntimeStatuses(eventDetails.ResourceId)+1
End Sub

Sub OnMaintenanceEndEvent(eventDetails)
    UpdateCounters eventDetails.Time
    'gravar status de disponibilidade do recurso relatado
    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
        atualizar contadores de segundos no período (quando a disponibilidade é
        esperada)
        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)
    Componente Dim
    'adicionar ao mapa de componentes monitorados todos os novos componentes a
    serem monitorados
    Para cada componente no allocatedComponents
        If Not MonitoredComponents.Exist(Component) Then
            MonitoredComponents(Component) = Component
            AvailabilityIndicators(Component) = 0
            DowntimeStatuses(Component) = 0
        End If
        Próximo

    'remove do mapa de componentes monitorados todos os componentes que não são
    mais relevantes
    Para cada componente no MonitoredComponents
        If Not allocatedComponents.Exist(Component) Then
            MonitoredComponents.Erase Component
            AvailabilityIndicators.Erase Component
            DowntimeStatuses.Erase Component
        End If
        Próximo
    End Sub

Function SystemIsAvailable
    Dim SystemAvailability
    SystemAvailability = True
```

```
Componente Dim
Dim ComponentAvailability
Para cada componente no MonitoredComponents
    ComponentAvailability = AvailabilityIndicators(Component) = 0 _
        Or DowntimeStatuses(Component) > 0
    ' a disponibilidade do sistema é avaliada com disponibilidade
    SystemAvailability = SystemAvailability And ComponentAvailability
Próximo
SystemIsAvailable = SystemAvailability
End Function

Function AllComponentsAreInPlannedDowntime
    Dim ComponentsInPlannedDowntime
    ComponentsInPlannedDowntime = 0
    Componente Dim
    Para cada componente no MonitoredComponents
        If DowntimeStatuses(Component) > 0 Then
            ComponentsInPlannedDowntime = ComponentsInPlannedDowntime + 1
        End If
    Próximo

    If ComponentsInPlannedDowntime = MonitoredComponents.Count Then
        AllComponentsAreInPlannedDowntime = True
    Else
        AllComponentsAreInPlannedDowntime = False
    End If
End Function
```

Criando exemplos de lógica de negócios efetivas

As recomendações de melhor prática sobre como criar uma lógica de negócios efetiva são fornecidas nos seguintes tópicos:

■ Métricas agrupadas

- Ao avaliar o volume do sistema, contabilize uma métrica agrupada como o número de itens na métrica e lembre-se de que tudo é multiplicado.
- Calcular novamente um item de agrupamento significa recalculer todo o agrupamento. Portanto, ao planejar o agrupamento e alterar a estrutura dos recursos, lembre-se de como os conectores serão configurados
- Os mesmos eventos de dados brutos que vão para vários itens de agrupamento têm alto custo de desempenho (alternância de contexto)

■ Variáveis globais

- Obter os parâmetros e os valores de atributo em cada lugar no código em que são necessários. Evite mantê-los em uma variável global, especialmente em métricas agrupadas (aumenta o tamanho dos estados)
- Evite a lógica que processa mapas grandes. Em vez disso, processe cada evento no método OnXXEvent
- Remova os itens dos mapas o quanto antes. Por exemplo, quando um ticket é fechado e não no final do período

■ Padrões de design

O pacote de conteúdo predefinido contém vários padrões de design para cenários comuns. Usar esses padrões de design pode melhorar o desempenho

■ Funcionalidade integrada

ACE possui funcionalidade integrada e ferramentas para vários propósitos conforme segue:

- Funcionalidade do período de atividade
 - NetTime
 - IsWithinTimeslot
- Hora do evento
 - TimeOfLastEvent
 - TimeOfLastEventHandler
- Objeto de contexto
 - Contém vários métodos importantes ao ambiente
 - Use esses métodos e evite a opção Safe ODBC

■ Saídas da lógica de negócios

Manter a estrutura em T_SLALOM_OUTPUTS. Isso significa que, se você possui várias tabelas lógicas em T_SLALOM_OUTPUTS com uma estrutura semelhante, é muito útil colocar campos lógicos similares no mesmo campo físico. Isso facilita a indexação de relatórios de desempenho melhorados

■ Capacidade de reutilização de eventos

Usar quando:

- Várias métricas estiverem executando o cálculo da primeira fase, o qual, por si só, é necessário para o contrato, e enviam eventos a uma métrica de resumo que calcula o resultado (cálculo financeiro, KPI de alto nível, por exemplo)
- Uma métrica única que executa uma agregação preliminar de dados brutos e envia eventos a várias outras métricas. Isto é razoável quando a métrica envia consideravelmente menos eventos do que recebe ou executa cálculos significativos que, de outra forma, são executados várias vezes

Evitar quando:

- Reduzir o número de métricas de modo significativo
- Implementar mais de três níveis
- Aumentar a complexidade da implementação e a manutenção se tornar mais difícil

■ Novos cálculos

- Evite novos cálculos grandes como parte da operação normal do sistema
- Os motivos para novos cálculos são:
 - Dados brutos com data e hora no passado
 - Singularidade do evento que altera os dados brutos no passado
 - Correções
 - Exceções
 - Alterações nos módulos de lógica de negócios
 - Alterações em um contrato
 - Reutilização do evento com data e hora no passado
 - Alterações na estrutura dos recursos
- Considere o uso da funcionalidade Bloqueio de dados calculados

■ Módulos de lógica de negócios

- Os módulos de lógica de negócios devem ser criados de uma forma que, após completamente revisados, não precisam ser alterados.
- A diretriz consiste em: um módulo igual a uma lógica genérica
- Os módulos de lógica de negócios que são muito específicos não podem atender a muitas métricas e não promovem a reutilização do código e da lógica

- É difícil manter os módulos de lógica de negócios que são muito genéricos. Além disso, se um módulo de lógica de negócios implementa várias lógicas complexas, uma correção em um fluxo (usado por parte das métricas), gera o novo cálculo de todas as métricas.

■ **Registro**

- Execute a filtragem de eventos usando o registro. Permite que a filtragem do código tenha um enorme impacto sobre o desempenho.
- Torne mais simples.
- Para o código que não é o próprio registro, use os métodos `dispatcher.IsRunTimeMode` e `OnResourceStructureChange`, principalmente quando houver muitas alterações de recursos.
- Evite usar o método `RegisterByEventType`.
- Nos módulos de lógica de negócios, use um formulário genérico (por parte contratual, serviço, tipo de recurso), parâmetros ou deixe o registro a ser executado usando a interface do usuário (opção preferencial para a Capacidade de reutilização de eventos).

Estudo de caso 15: Métricas agrupadas

Normalmente, ao descrever um determinado item de software, a descrição pode ser dividida em duas partes, O QUE e COMO. A parte O QUE significa a descrição do que este item de código faz. Já a parte COMO significa como ele faz isso. Há uma tendência para se concentrar na parte O QUE e ignorar a parte COMO. O motivo para que seja simples e, em muitos casos, justificável. Ao fazer isso, você reduz o acoplamento entre os componentes e não precisa se preocupar com informações que, em muitos casos, são irrelevantes. Em muitos casos, o custo de ignorar a parte COMO é um desempenho ruim.

Este estudo de caso aborda a maneira como o mecanismo calcula as métricas agrupadas (responder a parte COMO) e descreve o custo de desempenho que ele implica em algumas implementações. Também aborda várias maneiras de reduzir o custo alterando a implementação.

O que são métricas agrupadas

Métricas agrupadas são métricas que se incluem na definição de um determinado grupo de recursos. Esse grupo é chamado de agrupamento da métrica e cada um dos recursos deste grupo é chamado de item de agrupamento. Ao calcular uma métrica agrupada, um cálculo à parte é executado para cada item de agrupamento. Os cálculos para estes itens são semelhantes entre si, exceto:

- `Context.ClusterItem` - o valor da propriedade `Context.ClusterItem` que é igual ao item de agrupamento que está sendo calculado.
- Registros por item de agrupamento - quando uma métrica é registrada em eventos por um item de agrupamento, cada item recebe dados brutos e eventos de reutilização que são enviados para este item de agrupamento.

Como as métricas agrupadas são calculadas

Um fato importante de se entender sobre o cálculo de uma métrica agrupada é que todos os itens do agrupamento são calculados em paralelo. O termo em paralelo não significa que eles são calculados por segmentos diferentes, mas que durante o processamento de vários eventos que devem ser tratados por vários itens de agrupamento, os eventos são processados em sequência, e para cada evento, os itens de agrupamento relevantes são chamados e processam esse evento. Por exemplo, existem muitos eventos que devem ser tratados por muitos itens de agrupamento. Há duas maneiras de fazer isso:

Exemplo: opção 1

Para cada item de agrupamento C
 Para cada evento E que deve ser tratado por C
 Permitir que C gerencie E

Exemplo: opção 2

Para cada evento E
 Para cada item de agrupamento C que deve gerenciar E

Permitir que C gerencie E

O mecanismo manipula as métricas agrupadas usando a opção 2.

Outro ponto importante de se entender é que a execução do VBScript em PslWriter é realizada por um componente chamado Script Control. Existe apenas uma única instância deste componente por cada métrica e a instância é reutilizada para o cálculo de vários itens de agrupamento. Conforme mencionado anteriormente, uma vez que os itens de agrupamento são calculados em paralelo e o componente Script Control pode conter somente os dados de um único item de agrupamento em cada momento, é necessário alterar os dados do Script Control com frequência.

Para explicar isso, um pseudo código que executa o cálculo, é apresentado a seguir com mais detalhes.

- 1- Para cada métrica M
- 2- Permitir que X seja o evento mais recente , não controlado ainda por M
- 3- Permitir que T seja a data/hora do estado mais recente antes de X
- 4- Permitir que L seja a lista de todos os eventos registrados por M (todos os itens de agrupamento) a partir da data e hora T até a hora atual.
- 5- Para cada evento E em L
- 6- Para cada item de agrupamento C que deve manipular E
- 7- Permitir que C seja o item de agrupamento atualmente carregado no controle de script
- 8- Pegue os valores das variáveis globais do scrip control e armazene-os à parte para C
- 9- Pegue os valores das variáveis globais armazenados à parte para C e carregue-os no Script control.
- 10- Manipule o evento E

Todo o processo de localizar qualquer tempo de um evento que ainda não foi considerado, e executar o cálculo a partir desse ponto em diante é chamado de Recálculo. O processo de substituição de valores das variáveis globais (etapas 8 e 9 no código acima) é chamado de alternância de contexto.

Os dois principais problemas que podem ser facilmente identificados no código acima são:

- O recálculo será feito para todos os itens de agrupamento. Como o ponto no tempo T (etapa 3 no código acima) é encontrado uma vez, todos os itens do agrupamento realizam um recálculo com base nesse ponto. Isso significa que sempre que um item de agrupamento possui um novo evento, todos os itens do agrupamento serão recalculados.
- A alternância de contexto é feita com frequência. Isso pode ser facilmente observado, pois a alternância de contexto (etapas 8 e 9 no código acima) está localizada no loop mais interno.

Recálculo de métricas agrupadas

- Descrição do problema

Como já explicado, todos os itens de agrupamento em uma métrica agrupada são recalculados como um todo. Isso significa que, se houver uma métrica que está agrupada com mais de 1000 itens e um deles precisa de um novo cálculo do ano passado devido a alguns novos eventos que chegaram, todos os 1000 itens agrupados serão recalculados com base no ano passado.

- **Soluções possíveis**

As soluções a seguir podem reduzir este problema, mas nem sempre são aplicáveis, e cada uma possui suas desvantagens. O fator importante é entender o problema e o custo estimado e comparar este custo com o custo das soluções propostas.

- Quando o número de itens de agrupamento for pequeno, pode-se considerar a opção de definir cada um deles como uma métrica separada. A desvantagem dessa abordagem é o custo de manutenção de diversas métricas, bem como o fato de que não é possível executar um relatório para toda a métrica e detalhar um item de agrupamento específico.
- Quando o número de itens de agrupamento for grande, mas apenas um (ou alguns) for(em) recalculado(s) com frequência, pode-se considerar colocar esse item em uma métrica separada e deixar todos os outros itens de agrupamento na outra métrica
- Use, com frequência, um ponto de congelamento de cálculo para o/a contrato/métrica relevante de modo que essa métrica nunca tenha recálculos longos.
- Executar uma alteração nos conectores e nas origens de dados para que não haja mais recálculos (ou seja, não enviar os eventos cuja data/hora é mais antiga que um mês)

Alternância de contexto

- **Descrição do problema**

Como já explicado, a troca de contexto é feita no loop mais interno. Em outras palavras, para cada evento que deve ser controlado por cada item de agrupamento. Quando uma métrica recebe muitos eventos e quando cada evento é controlado por vários itens de agrupamento, esse valor poderá ser muito alto. Além disso, a operação de troca de contexto é relativamente cara (em relação à manipulação do evento na lógica de negócios), e você terá um problema.

O custo da operação de troca de contexto é proporcional ao tamanho dos dados que foram trocados. Os dados que foram trocados durante a troca de contexto são os valores de todas as variáveis globais na lógica de negócios (também denominado "o estado"). Portanto, quanto mais variáveis globais você tiver e maior o tamanho das variáveis globais, mais cara será a operação de troca de contexto.

Em especial, não é recomendável usar os mapas de lógica de negócios em métricas agrupadas, especialmente se o tamanho dos mapas for grande.

- **Soluções possíveis**

- Reduzir o tempo de trocas de contexto

A ideia é reduzir o tamanho do estado (variáveis globais). Essa abordagem pode ser feita recriando a lógica de negócios, de forma que ela não contenha os mapas. Obviamente, isso nem sempre é possível, mas é recomendável fazê-lo quando for.

- Reduzir a quantidade de trocas de contexto

Quando o agrupamento é pequeno, é possível criar uma métrica separada para item de agrupamento.

Evite métricas agrupadas com muitos itens de agrupamento que registram os mesmos eventos. A ideia é o seguinte:

Se cada evento for controlado por um único item de agrupamento, o valor da troca de contexto será proporcional ao número de eventos

Se os eventos forem controlados por todos os itens de agrupamento, a quantidade de troca de contexto será proporcional ao número dos eventos vezes o número de itens de agrupamento

Crie uma métrica não agrupada que calcula os resultados de todos os itens agrupados original (que são recursos simples e não itens de agrupamento). Faça com que essa métrica envie os resultados de cada item de agrupamento como um evento. Crie outra métrica que está agrupada e que recebe os eventos da primeira métrica e reporte o valor recebido nesses eventos como o resultado. A ideia aqui é que a grande quantidade de eventos de dados brutos seja controlada por uma métrica não agrupada e a métrica agrupada controle um único evento por período, por item de agrupamento.

Estudo de caso 16: padrões de design da lógica de negócios

Há diversos padrões de design que podem ser usados em muitas lógicas de negócios. Esses padrões foram testados e usá-los quando aplicável pode economizar muito tempo e, na maioria dos casos, pode criar uma lógica de negócios mais eficiente. Este estudo de caso concentra-se em um padrão de design.

Atualizar padrão de design de contadores

Esse design padrão é útil em quase todas as lógicas de negócios, cujo objetivo é medir o tempo entre determinados eventos. Os exemplos de lógicas de negócios são: lógicas de negócios para a medição da disponibilidade, o tempo de inatividade, o tempo médio entre falhas, o tempo médio de restauração, o tempo médio de resposta, o tempo médio de resolução, a porcentagem de componentes com disponibilidade menor que X, o número de ocorrências não resolvidas na hora, etc.

A parte comum para todas as lógicas de negócios é que o resultado depende da data e hora de vários eventos recebidos..

O princípio básico para decidir se a lógica de negócios pode se beneficiar desse padrão de design consiste em: se a lógica de negócios depender da data/hora dos diversos eventos que recebe, ela provavelmente deve usar este padrão de design.

Esqueleto do padrão de design

O código de uma lógica de negócios que utiliza este padrão pode ser dividida em duas partes: uma estrutura e uma implementação. A estrutura contém o código que, na maioria dos casos, é fixo e não é alterado para as diversas lógicas de negócios. Essa parte é a mesma para o cálculo de disponibilidade e para o número de tickets não resolvidos na hora. A implementação contém o código que é específico para cada lógica de negócios.

É recomendável colocar essas duas partes do código em módulos de lógica de negócios distintos e reutilizar o módulo da estrutura em diferentes métricas.

O código da estrutura é demonstrado abaixo:

```
Dim g_PrevEventTimestamp

Sub OnLoad(time)
    g_PrevEventTimestamp = time
    InitializeStatusVariables
End Sub

Sub OnRegistration(Dispatcher)
    Se houver uma cópia separada de variáveis do status
    'para que cada recurso registrado dependa dos recursos
    'registrados, é necessário definir valores iniciais para
    as variáveis de status dos recursos recém adicionados aqui
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
```

Aqui está a estrutura do módulo de implementação:

```
' Defina aqui as variáveis de status. Ela pode ser uma
' variável global simples ou diversas variáveis globais complexas
' dependendo da lógica de negócios
Dim g_StatusVar_1, g_StatusVar_2, ... ,g_StatusVar_n

' Defina os contadores aqui.
'Isso pode ser uma variável global simples ou muitas
' variáveis globais complexas dependendo da lógica de negócios
Dim g_Counter_1, g_Counter_2, ... , g_Counter_n

Sub InitializeStatusVariables ()
    'Defina os valores iniciais para as diversas variáveis de status
End Sub

Sub InitializeCounters ()
    'Defina os valores iniciais para os diversos contadores
    g_Counter_1 = 0
```

```
g_Counter_2 = 0
'...
g_Counter_n = 0
End Sub

Function CalculateResult ()
'Calcular o resultado. O resultado deve depender
'dos valores dos contadores. Ele não deve depender
'do valor das variáveis de status. Ele não deve
'alterar os valores dos contadores ou das variáveis
' de status
End Function

Sub UpdateStatus(method, eventDetails)
    Atualizar o valor das variáveis de status com base
    ' nos parâmetros (e possivelmente no valor antigo das
    ' variáveis de status)
End Sub

Sub UpdateCounters(diff)
    Atualizar os valores dos contadores com base nos
    'valores anteriores, no valor das variáveis de status
    'e no valor do parâmetro diff.
    'Em muitos casos, esse cálculo tem como base
    ' o valor de Context.IsWithinTimeslot
End Sub

Sub OnEvent_1(eventDetails)
    HandleEvent (eventDetails.Time)
    UpdateStatus("OnEvent_1",eventDetails)
End Sub

Sub OnEvent_2(eventDetails)
    HandleEvent (eventDetails.Time)
    UpdateStatus("OnEvent_2",eventDetails)
End Sub

'...

Sub OnEvent_n(eventDetails)
    HandleEvent (eventDetails.Time)
    UpdateStatus("OnEvent_n",eventDetails)
End Sub
```

Para melhor explicar este padrão de design, tem-se um exemplo da implementação deste padrão para calcular a disponibilidade. Esse exemplo supõe que há eventos para UP e DOWN que são tratados por dois manipuladores de eventos diferentes na lógica de negócios. A disponibilidade é definida como o percentual de tempo no qual o sistema está ativo a partir do tempo total no período de atividade. O status do sistema é considerado o status do último evento recebido (ATIVO OU INATIVO).

Veja a seguir o código da implementação (o código da estrutura não é alterado):

```
' variável de status
Dim g_Status
' Contadores.
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
```

Há diversas variações neste padrão. Uma das variações mais comuns é quando um contador de tempo distinto deve ser mantido para entidades diferentes. Por exemplo, ao medir o tempo de uma solução, um contador distinto deve ser mantido para cada ticket aberto. Nesse caso, ao manipular um evento relevante apenas para um ticket, é mais eficiente atualizar apenas o contador daquele ticket. Quando um evento comum é manipulado (como OnPeriodEnd ou OnTimeslotEnter) os contadores de todos os tickets devem ser atualizados.

Observação: essa variação do padrão requer que uma cópia separada da variável global `g_PrevEventTimestamp` seja mantida para cada ticket.

Alguns exemplos úteis sobre o uso deste padrão podem ser encontrados no conteúdo predefinido. Tenha em mente que esse padrão é usado de modo diferente no conteúdo predefinido, e a separação entre a estrutura e a implementação não é tão evidente.

Estudo de caso 17: Funcionalidade integrada

O ACE possui funcionalidade integrada e ferramentas para diversos fins. É recomendável usar essa funcionalidade ao gravar em VBS. Como o VBS é uma linguagem interpretada, a reprodução dela no VBS compromete o desempenho.

Encontra-se a seguir, uma lista de funções integradas e o modo apropriado para usá-las:

IsWithinTimeslot

Essa é a forma mais simples de funções integradas. Seu objetivo é permitir que lógica de negócios indique se, no momento, o sistema encontra-se dentro de um período de atividade ou não. Isso elimina a necessidade de gerenciar uma variável na entrada do período de atividade e as funções de saída do período de atividade para fazer a mesma coisa. Por exemplo, em vez de executar o seguinte código:

```
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
```

É possível executar esse código mais simples:

```
Sub OnEvent(eventDetails)
    If context.IsWithinTimeslot Then
        count = count + 1
    End if
End sub
```

Essa funcionalidade não atenderá às suas necessidades, se deseja usar ou manter as informações sobre a data e hora da entrada e saída do período de atividade. Mas, em geral, isto não é necessário, e esse código é suficiente.

TimeOfLastEvent

Esta função apresenta a data e hora do último evento de dados brutos ou intermediários que foi manipulado. Isso significa que não é necessário salvar essas informações no manipulador de eventos, pois elas estão diretamente disponíveis por meio desta função. Por exemplo:

```
Function result
    Dim LastEventTimestamp
    LastEventTimestamp = Context.TimeOfLastEvent
End function
```

TimeOfLastEventHandler

Essa função retorna a data e hora do manipulador do último evento chamado pelo ACE. Isso inclui não somente os manipuladores de evento de dados brutos e intermediários, mas também os eventos do sistema que foram chamados. Isso é especialmente útil em manipuladores de evento que não recebem a hora da função de resultado, por exemplo. Por exemplo:

```
Function result
    Dim LastEventHandlerTimestamp
    LastEventHandlerTimestamp= Context.TimeOfLastEventHandler
End function
```

NetTime

Essa função permite especificar dois carimbos de data e hora e receber o tempo líquido (em segundos) em que o sistema estava dentro do período de atividade da regra atual entre os dois carimbos de data e hora. Essa é uma funcionalidade muito complicada e não deve ser implementada no VBS. Se ela for implementada no VBS, será necessário manter uma lista de todas as entradas e saída do período de atividade ou calcular a diferença entre o tempo de entrada e saída do período de atividade a fim de calcular o intervalo entre eles. Em condições extremas, isso pode ocorrer diversas vezes e não é bom para o cálculo de desempenho. A função interna faz o mesmo após a otimização significativa, e, por isso, é muito eficiente. Por exemplo:

```
Function result
    Dim MyNetTime
    MyNetTime = Tools.NetTime(MyBeginTimestamp, MyEndTimestamp)
End function
```

O objeto de contexto

O objeto de contexto tem uma variedade de parâmetros que fornecem informações sobre:

- A métrica atual
- O contrato.
- O estado do cálculo atual.
- O domínio de serviço, a categoria de domínio e os valores relacionados (por exemplo, limite).
- Informações sobre o agrupamento - o agrupamento em geral e o item de agrupamento específico a serem manipulados.
- Funções que retornam listas de recursos conforme os requisitos do usuário.
- Funções que permitem converter nomes de recursos em IDs de recurso e vice-versa.

Acessar essas informações diretamente a partir do banco de dados usando o ODBC seguro não é nada produtivo e não faz sentido, pois as informações estão prontamente disponíveis pelo objeto de contexto. Se possível, use sempre a funcionalidade integrada como uma forma de obter mais informações.

Estudo de caso 18: Registro

A lógica de negócios é muitas vezes criada de modo a manter um mapeamento da estrutura do recurso da métrica para uso em cálculos. Considerando que a estrutura do recurso muda com o tempo, tal lógica de negócios precisa atualizar a estrutura no mapeamento mediante alterações na estrutura do recurso.

O método OnRegistration é chamado quando a estrutura do recurso é alterada, uma vez que é responsável por gerenciar o comportamento do mecanismo associado às alterações nos registros e agrupamentos decorrentes de mudanças na estrutura do recurso. O fato de esse método ser chamado a cada alteração na estrutura do recurso o torna conveniente para atualizar o mapeamento mencionado acima. No entanto, a criação do mapeamento não é relevante para o processo de registro. Isso significa que criar o mapeamento prejudica o desempenho da função OnRegistration. Isso não é importante durante o tempo de execução, uma vez que não acontece com muita frequência. Contudo, o método OnRegistration também é chamado durante o processamento da infraestrutura pelo mecanismo. Enquanto isso, o sistema identifica se as alterações na estrutura do recurso são relevantes para o registro de cada métrica específica pela qual a instância é responsável. Durante esse processo, o método OnRegistration é chamado a cada alteração na estrutura do recurso, mesmo se ela não for relevante à métrica atual. Isso significa que o método pode ser chamado muitas vezes por métrica.

Se essa lógica for implementada no método OnRegistration, uma pequena redução no desempenho em tempo de execução pode se tornar algo significativo durante o processamento da infraestrutura.

Para resolver esse problema, a criação de mapeamentos ou qualquer outra inicialização que precise ser executada quando uma alteração na estrutura do recurso ocorrer, mas não for relevante ao registro, pode ser feita de duas maneiras:

Usando a propriedade IsRunTimeMode no objeto de dispatcher.

Ela permite ao usuário identificar se a execução atual é ou não de cálculo e embutir a lógica que não for relevante ao registro em uma instrução IF, garantindo, assim, que a execução ocorra apenas em tempo de execução.

No exemplo abaixo, o que está marcado em azul é a parte da lógica de negócios relevante ao registro e que precisa estar sempre em execução. O que está marcado em verde não é relevante ao registro e pode ser embutido na nova instrução IF.

```
Sub OnRegistration (dispatcher)
    Dim MyResource
    MyResource = Context.ClusterItem
    Dispatcher.RegisterByResource "OnEvent", "My Event Type", MyResource
    Dim ThisResourceMap
    Set GlobalResourceVector= CreateObject("SlalomVector.Vector")
    Dim resource
    Set ThisResourceMap = Context.ResourcesOfResourceGroup(Context.ClusterItem)
    Para cada recurso em ThisResourceMap
```

```
GlobalResourceVector.Add resource
Próximo
End Sub
```

Esse código pode ser aprimorado da seguinte maneira:

```
Sub OnRegistration (dispatcher)
    Dim MyResource
    MyResource = Context.ClusterItem
    Dispatcher.RegisterByResource "OnEvent", "My Event Type", MyResource
    If Dispatcher.IsRunTimeMode Then
        Dim ThisResourceMap
        Set GlobalResourceVector= CreateObject("SlalomVector.Vector")
        Dim resource
        ThisResourceMap = Context.ResourcesOfResourceGroup(Context.ClusterItem)
        Para cada recurso em ThisResourceMap
            GlobalResourceVector.Add resource
        Próximo
    End If
End Sub
```

Usando o método OnResourceStructureChanged.

Este método é executado logo após OnRegistration e, portanto, oferece a mesma funcionalidade que a metodologia original. Contudo, é executado apenas em tempo de execução. Esse método não é chamado durante o processamento da infraestrutura e, assim, o desempenho não é afetado.

No exemplo abaixo, o que está marcado em azul é a parte da lógica de negócios relevante ao registro e que precisa permanecer no método OnRegistration. O que está marcado em verde não é relevante ao registro e pode ser colocado na nova função.

```
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)
    Para cada recurso em ThisResourceMap
        GlobalResourceVector.Add resource
    Próximo
End Sub
```

Esse código pode ser aprimorado da seguinte maneira:

```
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)
    Para cada recurso em ThisResourceMap
        GlobalResourceVector.Add resource
    Próximo
End Sub
```

Caso de estudo 19: assistente de conector para fonte de dados com base em arquivo

Este estudo de caso examina os métodos recomendados para integração a uma fonte de dados com base em arquivo. O cenário de exemplo manipula um arquivo de dados CSV produzido pelo sistema de origem. Para a maioria das integrações com base em arquivo, a CA recomenda algumas diretrizes básicas para garantir que a integração tenha o mínimo de risco. São elas:

- Quando a opção estiver disponível, solicite que os dados sejam entregues no sistema de arquivos do servidor de aplicativos do CA Business Service Insight. Isso garante que o mecanismo de entrega não dependa do conector que estiver tentando obter dados de um armazenamento remoto, minimizando, assim, problemas relativos a permissões de contas de usuários, problemas de sincronização, etc.
- A convenção de nomenclatura de arquivos é importante, pois o conector solicita os arquivos conforme a nomenclatura alfanumérica. Se isso puder ser controlado, a CA recomenda solicitar duas partes:
 - Uma convenção de nomenclatura inteligente, com base no conteúdo do arquivo de origem (especialmente se mais de um arquivo vier da fonte).
 - Uma marca de data e hora em ordem inversa para garantir que os arquivos estejam ordenados de modo que o mais recente fique por último. (por exemplo, <nome_do_arquivo>_yyyymmdd_hhmiss.csv). A abrangência da marca de data usada depende da frequência dos dados entregues.

Nesse cenário, os arquivos de origem provêm de uma fonte de dados Topaz (agora conhecido como Mercury Global Monitor, propriedade da HP). Ele foi criado usando uma API do produto para incluir os arquivos necessários para as KPIs específicas solicitadas. Os arquivos são entregues ao servidor de aplicativos do CA Business Service Insight diretamente por um processo automatizado externo. Os arquivos de origem são nomeados assim: *Topaz_yyyymmdd_hhmiss.csv*.

Observação: a marca de data e hora do arquivo corresponde à data e hora em que ele foi criado, portanto, todas as entradas no arquivo referem-se à essa data e hora.

Uma amostra dos dados contidos pelo arquivo pode ser vista abaixo.

	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	

Observação: a CA recomenda verificar os arquivos CSV no Notepad (não apenas no Excel) para garantir que o formato de data corresponda ao esperado. O Excel tenta formatar as datas conforme as configurações regionais do computador, as quais podem não corresponder ao formato reconhecido pelo conector.

```

Topaz_20050330_200000.csv - Notepad
File Edit Format View Help
Profile,Status,TopazTransaction,Time,ResponseTime,wtdConnectionTime,wtdDNS
54,Pass,yh_L,30/03/2005 06:09:00,218,6,1,23,179,2
54,Pass,yh_L,30/03/2005 06:24:00,270,7,1,25,226,4
54,Pass,yh_L,30/03/2005 06:39:00,209,6,1,21,171,3
54,Pass,yh_L,30/03/2005 06:54:00,243,5,0,22,204,3
54,Pass,yh_L,30/03/2005 07:09:00,207,6,1,21,167,4
54,Pass,yh_L,30/03/2005 07:24:00,245,5,1,24,203,4
54,Pass,yh_L,30/03/2005 07:39:00,218,6,1,22,180,3
54,Pass,yh_L,30/03/2005 07:54:00,207,7,1,24,163,6
54,Pass,yh_L,30/03/2005 08:09:00,248,6,1,26,202,6
54,Pass,yh_L,30/03/2005 08:24:00,177,6,1,22,136,3
54,Pass,yh_L,30/03/2005 08:39:00,208,6,1,24,166,4
54,Pass,yh_L,30/03/2005 08:54:00,252,5,0,22,210,7
54,Pass,yh_L,30/03/2005 09:09:00,173,7,0,25,124,5
54,Pass,yh_L,30/03/2005 09:24:00,211,7,1,20,171,6
54,Pass,yh_L,30/03/2005 09:39:00,233,10,1,28,183,5
54,Pass,yh_L,30/03/2005 09:54:00,166,7,0,23,124,6
54,Pass,yh_L,30/03/2005 10:09:00,177,7,1,26,127,7

```

Antes de iniciar qualquer processo de criação do conector, também é importante que todas as análises e investigações necessárias tenham sido executadas na fonte de dados e nas KPIs relacionadas para garantir que o seguinte seja identificado:

- Quais campos são necessários para a lógica de negócios
- Qual é o formato das datas no arquivo
- Qual é o fuso horário dos valores de data e hora no arquivo de origem (esses fuso horários devem ser criados no sistema antes de iniciar o processo de criação do conector)
- Há algum campo de data que pode ter entradas em branco ou nulas
- Qual é o comportamento da fonte de dados (por exemplo, todos os registros foram adicionados ou alguns registros são uma atualização de um evento anterior?)
- Onde é necessário mais detalhamento para as KPIs (isso pode afetar a sua escolha de “recurso”)
- Quais “tipos de evento” são necessários para a filtragem eficaz dos eventos na lógica de negócios

Assim que esses pontos tiverem sido esclarecidos, você já pode dar início ao processo de criação.

No momento, é possível executar o processo de criação do conector com base nesse cenário usando o assistente de conector.

Em nosso cenário, usaremos “TopazTransaction” como recurso, “Profile” como tipo de evento e o campo “Time” como marca de data e hora. Também iremos inserir os campos “Status”, “ResponseTime” e “WtdDownloadTime” na estrutura do evento para usar com a lógica de negócios.

Criação do conector

Primeiro, assegure-se de que o sistema esteja pronto para criar o conector e implantá-lo no servidor corretamente verificando se os serviços abaixo foram iniciados no servidor de aplicativos.

- Serviço de implantação de conector
- Serviço do listener de conectores

Em seguida, vá até a página Conectores e crie um conector. Nesta mesma página, selecione a opção Adicionar > Conector de arquivo de texto.

Etapa Geral

Na etapa Geral do assistente de conector, insira os seguintes campos:

- **Nome:** forneça um nome adequado ao conector
- **Endereço do conector:** LOCALHOST é a opção padrão (para a implantação do servidor de aplicativos), contudo, outros endereços podem ser inseridos usando-se o botão ao lado, se necessário.
- **Formato de data e hora:** refere-se ao formato padrão usado nos campos de data e hora na fonte de dados. A seleção correta garante que os campos sejam detectados automaticamente mais tarde, durante o processo do assistente. Novos formatos de data e hora podem ser inseridos aqui agora, se necessário, usando o botão ao lado do campo.
- **Fuso horário:** refere-se ao fuso horário em que os registros de dados são gravados. É necessário especificá-lo para que o conector possa mudar corretamente o campo event_timestamp (entre outros campos de data e hora) de volta para UTC com o intuito de corrigir a mudança de horário internamente. O campo deve ter sido preenchido com antecedência, conforme a lista de verificação da fonte de dados na seção anterior.

Observações:

Há também um botão Avançado nesta página que fornece acesso a um certo número de parâmetros de configuração para o conector. A maioria deles pode manter seus valores padrão, a menos que seja necessário alterá-los.

O conector “Porta” é atribuído automaticamente ao conector a partir do CA Business Service Insight, mas pode ser substituído aqui, se necessário. Outros parâmetros importantes que podem ser alterados: configurações regionais, modo online/offline, detalhes da conexão, opções de monitoramento e registro em log, configurações de execução única/contínua, limites de erro, comentários e nomes de arquivo.

Geral - Configurações avançadas -- Caixa de diálogo Página da Web

Geral | Conectividade | Monitoramento | Processamento | Outros | Notas

Modo: ☒ Online ☐ Offline

Porta: *

Detalhes das configurações regionais:

Símbolo decimal *

Símbolo de agrupamento de dígitos: *

Uma análise de cada uma dessas configurações ultrapassaria o escopo deste estudo de caso, contudo, ela pode ser encontrada no tópico [Especificações de configuração do conector](#) (na página 313).

Clique em Avançar para prosseguir para a próxima etapa do assistente. A próxima etapa fornece acesso à interface da fonte de dados do conector.

Etapa Interface de origem de dados

Na etapa Interface da fonte de dados do assistente de conector, insira os seguintes campos:

- *Nome da fonte de dados*: um nome para esse arquivo de origem específico (é possível ter vários arquivos de origem no mesmo conector)
- *Caminho do arquivo*: o caminho no servidor de aplicativos (ou outro servidor) que contém os arquivos de dados de origem. No caso de um servidor que não seja o de aplicativos, use uma notação UNC (por exemplo, //server01/sourcefolder)
- *Padrão de nome*: para uso com caractere curinga com o intuito de filtrar os arquivos localizados no caminho do arquivo carregado pelo conector.

A guia Definição de análise também ajuda a definir a estrutura do arquivo sendo importado. Os campos podem ser usados da seguinte maneira:

- *Título*: caixa de seleção de valores booleanos para especificar se há uma linha de título no arquivo de dados ou não (por exemplo, a primeira linha no CSV é um título, seguida por valores nas linhas subsequentes).
- *Delimitadores*: especifique o delimitador no arquivo que separa os campos individuais.

Observação: há também dois outros botões Avançado aqui, que fornecem outras opções de configuração. Um botão está na guia Detalhes, e o outro está na guia Definição de análise. O botão Avançado na guia Detalhes fornece acesso aos seguintes parâmetros:

Origem de dados ativa: caixa de seleção booleana que permite ativar e desativar essa seção de origem específica do conector

Após o processamento: permite especificar se o arquivo deve ser excluído ou mantido após o processamento

Nome do arquivo inicial: define o nome do arquivo inicial a partir do qual iniciar o processamento (caso não queira carregar todos os arquivos em uma determinada pasta).

Verificar novos dados a cada: define o intervalo entre cada verificação de novos arquivos quando o conector está definido para ser executado continuamente.

O botão Avançado, na guia Definição de análise, fornece acesso para definir as opções de término do registro, por exemplo, registros em mais de uma linha, etc.

Após a configuração dos detalhes e da definição de análise, é possível carregar uma amostra do arquivo de origem no assistente para testar as opções de configuração e visualizar seu conteúdo.

Ao clicar no botão Procurar, um arquivo de amostra pode ser carregado na janela de visualização abaixo. Vá para o arquivo de amostra e, em seguida, clique no botão Arquivo de teste. Esta é uma etapa opcional.

Observação: a função Procurar abre a partir do computador local em que você estiver trabalhando. Caso o servidor não seja o de aplicativos, é preciso ter uma cópia dos arquivos de origem que estão no servidor para que a operação funcione. Não é possível procurar diretamente no servidor.

Assim que o arquivo é carregado, pode-se exibir o conteúdo mostrado na janela de visualização, como se pode ver abaixo.

Observação: o nome do campo é mostrado no cabeçalho, juntamente com o tipo de campo detectado (número inteiro, texto, data e hora, etc.). É preciso verificar se esses itens foram detectados corretamente e conforme seus requisitos antes de prosseguir. Não deixe de verificar o seguinte:

- A marca de data e hora detectada como texto – isso é feito de acordo com o formato de data e hora especificado na primeira etapa do assistente.
- Recurso detectado como texto.

Se estiver satisfeito com a visualização do arquivo, clique em Avançar. A etapa Mapeamento é exibida.

Etapa Mapeamento

A etapa seguinte (e final) do assistente de conector executa tarefas de conversão e permite mapear os campos de entrada a partir da fonte de dados para os campos de saída, o que constitui o evento do CA Business Service Insight. Para prosseguir, pode-se usar duas opções, dependendo se o tipo de evento já foi criado no sistema ou não. Há também outras opções que se pode escolher para configurar e garantir que a nomenclatura padrão seja seguida. Contudo, a maioria é opcional, de modo a facilitar o processo e reduzir a quantidade de etapas necessária. As etapas obrigatórias são apontadas abaixo.

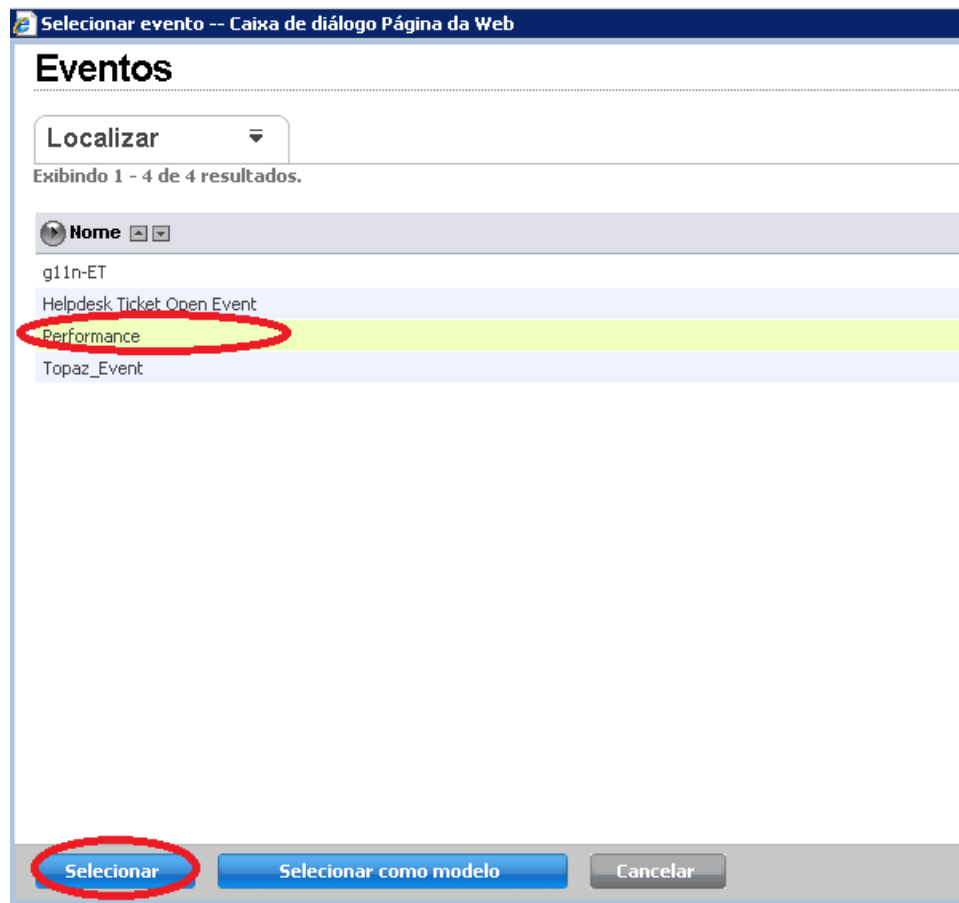
As etapas (inclusive as opcionais) seguidas para a configuração do mapeamento são:

1. Forneça um nome para o formato de entrada (útil para conectores com mais de uma entrada).
2. Adicione qualquer outro campo necessário (por exemplo, os tipos de campo Valores constantes, Fonte de dados, Título, Nome de arquivo ou Composto).
3. Crie critérios de validação de entrada, conforme necessário.
4. Selecione um tipo de evento existente ou crie um. (Obrigatório)
5. Faça o mapeamento dos campos da entrada à saída. (Obrigatório)
6. Forneça um nome para o formato de saída.
7. Faça o mapeamento para ResourceId, Timestamp e Event Type.
8. Crie critérios de validação de saída, conforme necessário.
9. Defina a configuração “OnDuplication” para os eventos.

Se optar por criar um tipo de evento, uma nova janela pop-up é exibida, pré-preenchida conforme o formato de entrada na tela principal. Você ainda precisa inserir o nome do tipo de evento, além de atribuir um tipo de recurso ao tipo de evento.

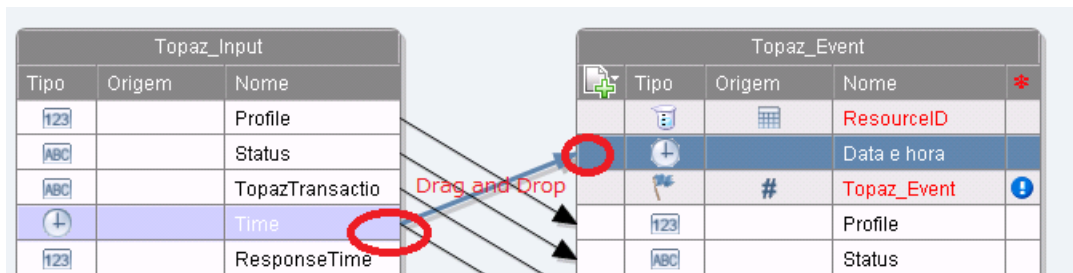
Após o preenchimento, clique em Salvar e fechar para que o mapeamento seja concluído automaticamente.

Caso selecione a opção Selecionar evento, será apresentada uma lista dos tipos de evento existentes no sistema a partir da qual escolher. No entanto, ao prosseguir, apenas os campos de entrada com o nome e o tipo de dados correspondentes a partir do tipo de evento serão vinculados automaticamente. Os campos restantes devem ser mapeados manualmente.

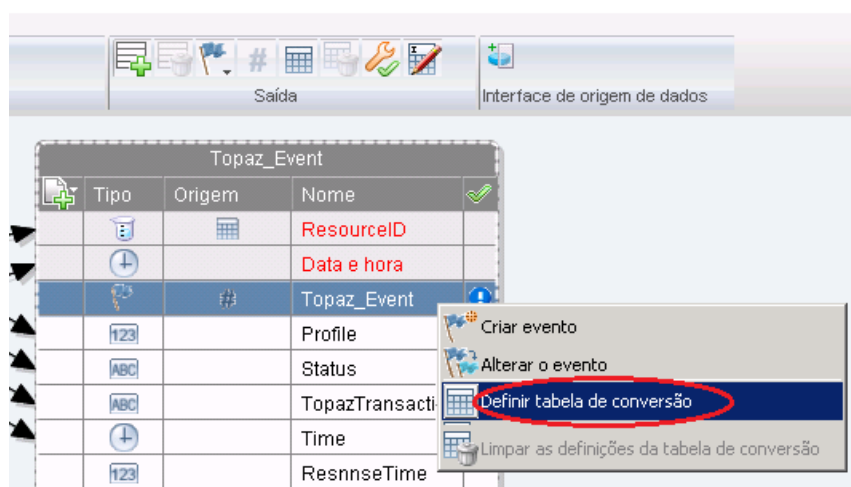


A próxima etapa é configurar a ID de recurso, a marca de data e hora e o tipo de evento. Se os campos já existirem (eles devem ter sido criados durante etapas anteriores), eles podem ser vinculados a esses campos, conforme necessário.

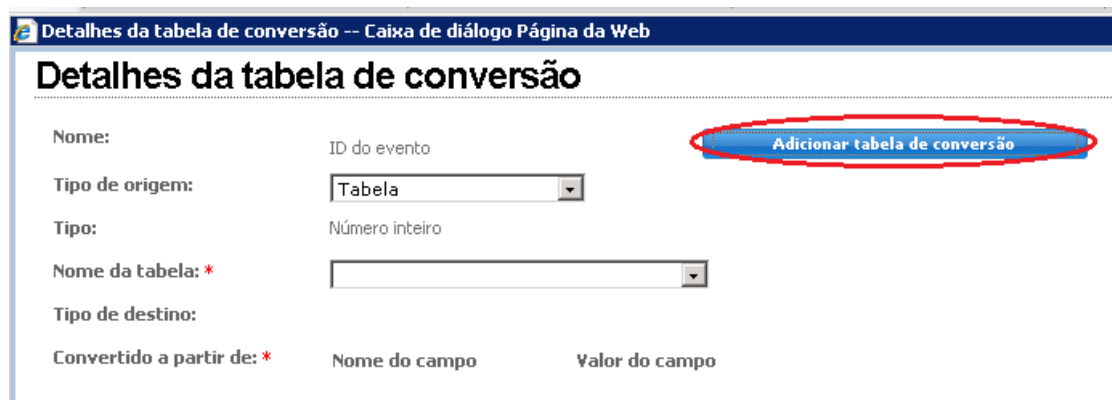
A interface oferece suporte ao estilo de associação "arrastar e soltar". Se os itens associados não permanecerem após a configuração, verifique se há correspondência entre o tipo de cada item. (Por exemplo, data e hora, texto, número inteiro, etc.)



- A ID do recurso deve ser mapeada conforme os requisitos (identificados durante a análise) a partir de um valor de texto na entrada.
- A marca de data e hora deve ser definida para a variável Time que define a data e a hora em que o evento ocorreu e deve ser usada para fazer cálculos.
- O tipo de evento é definido com o valor de uma constante por padrão, conforme o campo Nome da fonte de dados das telas anteriores. Esse valor pode ser substituído, se preciso for. Isso pode ser necessário se diversos eventos precisarem ser enviados a partir desse conector, de acordo com o valor de um campo específico (por exemplo, você quer enviar um evento diferente, dependendo do valor de um campo de entrada). Para fazê-lo, clique com o botão direito do mouse na linha de tipo de evento (ou clique no botão acima, conforme indicado na figura abaixo) e selecione Definir tabela de conversão.



Em seguida, uma janela pop-up é exibida.



Se a tabela de conversão para esse valor já existir, selecione-a aqui ou configure uma nova tabela de conversão. O botão na tela acima pode ser usado para isso.

 Detalhes da tabela de conversão -- Caixa de diálogo Página da Web

Detalhes da tabela de conversão

Nome: *

Descrição:

Campos de origem: *

	Nome do campo
	event_type

Tipo de destino: Evento

Depois de criar a tabela, é preciso especificar quais campos de entrada são mapeados para os Campos de origem especificados acima.

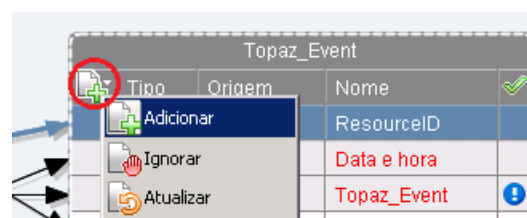
Se desejar especificar outra tabela de conversão para os recursos do conector, também é possível fazê-lo neste momento. Isso é feito usando-se um processo semelhante ao descrito acima para o tipo de evento.

Observação: por padrão, o assistente de conector aloca todos os recursos a uma tabela de conversão existente chamada Tabela de conversão padrão, caso o contrário não seja especificado. Isso pode aplicar-se a implementações simples, contudo, em casos mais complexos (com o intuito de separar os dados), a CA recomenda usar uma tabela diferente. Isso também é obrigatório quando os Campos de origem da seção de mapeamento do conector são diferentes ou contém um valor superior ao outro.

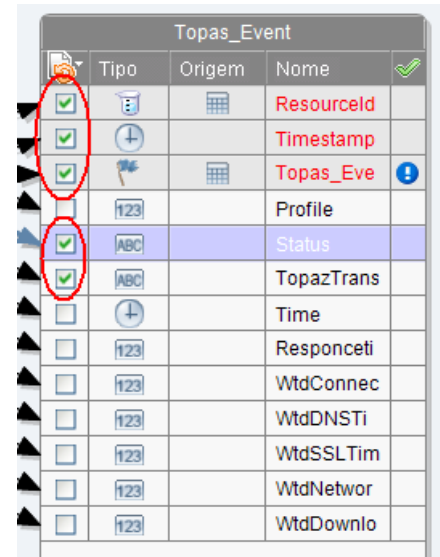
A última etapa do mapeamento é configurar o item “OnDuplication” para o conector. Ele descreve a ação a ser executada quando um segundo evento, com valores correspondentes para todos os campos Chave, é recebido. Essa chave exclusiva pode ser definida para cada saída do conector (continue lendo abaixo para obter mais informações sobre isso). Por padrão, o valor de “OnDuplication” fica definido como Adicionar, de modo que só é preciso alterá-lo se uma ação diferente for necessária. Os valores disponíveis são:

- *Adicionar:* o novo evento é adicionado apenas como um evento separado.
- *Ignorar:* o novo evento é ignorado (descartado) pelo conector.
- *Atualizar:* o novo evento substitui o evento carregado anteriormente apenas se os campos Valor dos eventos forem diferentes.
- *Atualizar sempre:* o novo evento substitui o evento carregado anteriormente.

O uso de opções que não sejam Adicionar pode afetar o desempenho do conector e deve ser considerado com cuidado antes da implementação, especialmente quando o volume de dados é muito grande.



Se o valor padrão não for usado, a estrutura de saída exibe uma série de caixas de seleção usadas para definir a chave exclusiva. A chave é constituída pelos itens contendo uma marca de seleção ao lado. A chave deve ser escolhida conforme os requisitos identificados após uma análise cuidadosa.



Tipo	Origem	Nome	
		ResourceId	☒
		Timestamp	☒
		Topaz_Eve	☒
123		Profile	
ABC		Status	☒
ABC		TopazTrans	☒
		Time	☐
123		Responceti	☐
123		WtdConnec	☐
123		WtdDNSTi	☐
123		WtdSSLTim	☐
123		WtdNetwor	☐
123		WtdDownlo	☐

Após concluir a configuração da seção de mapeamento, clique no botão Concluído na parte inferior direita da tela. Você retornará para a lista de conectores no sistema e deverá ver o conector criado com o status Inativo.

Conectores

Localizar Adicionar

Exibindo 1 - 3 de 3 resultados.

	Nome	Modo	Tipo	Iniciador	Endereço do conector	Porta	Status
	Topaz Adapter	Online	Conector de arquivo de texto	Servidor	LOCALHOST	6207	Inativo
	Topaz Adapter	Online	Conector de arquivo de texto	Servidor	LOCALHOST	6218	Inativo
	Topaz_Event	Online	Conector de arquivo de texto	Servidor	LOCALHOST	6208	Inativo

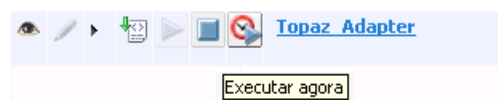
Implantação de conectores

Agora que o conector foi configurado, é preciso implantá-lo no servidor de aplicativos. Clicar no botão Iniciar conector faz com que o serviço de implantação de conector o implante no sistema de arquivos. Isso inclui o seguinte:

- Criação das pastas necessárias para executar o conector
- Cópia da configuração em XML na pasta
- Criação de um atalho para executar o conector

Depois de finalizadas essas operações, suas alterações serão refletidas no servidor.

Agora é possível executar o conector a partir da GUI clicando no botão Executar agora, que se tornou ativo. Isso faz com que serviço de implantação de conector execute o conector no servidor e dê início à coleta de dados.



Uma vez executado, é possível ver os recursos e eventos pendentes sob a seção de conversões pendentes do sistema.

Entradas de conversão

Localizar

Adicionar entrada

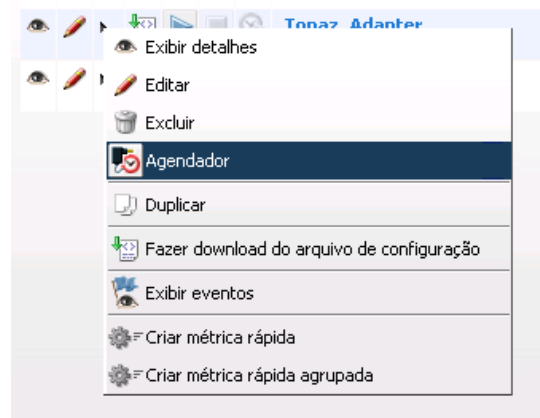
Exibindo 1 - 9 de 9 resultados.

  	Nome da tabela	Tipo de destino	Status	Convertido a partir de
☐ 1 	g11n-ResourceName	Recurso	Pendente	ResName
☐ 1 	Topaz Events Table	Evento	Pendente	hp_A
☐ 1 	Topaz Events Table	Evento	Pendente	hp_L
☐ 1 	Topaz Events Table	Evento	Pendente	hpmta_A
☐ 1 	Topaz Events Table	Evento	Pendente	hpmta_L
☐ 1 	Topaz Events Table	Evento	Pendente	mororNBA
☐ 1 	Topaz Events Table	Evento	Pendente	motormta_A
☐ 1 	Topaz Events Table	Evento	Pendente	motormta_L
☐ 1 	Topaz Events Table	Evento	Pendente	yh_A

As entradas pendentes podem ser convertidas conforme a configuração normal do sistema. Após a conclusão da conversão, o conector deve ser executado novamente para carregar os dados brutos para o sistema.

Programação de conectores

Além de executar o conector, também é possível definir uma programação para ele a partir da GUI. Contudo, para que isso seja possível, seu status deve ser Interrompido. Uma vez interrompido, pode-se definir a programação pelo menu de contexto do conector selecionando a opção Agendador.



As opções de programação são as mesmas que as fornecidas pelo Agendador de tarefas do Windows. Na verdade, o serviço de implantação de conector cria essa programação como um item do Agendador de tarefas.

A screenshot of the 'Agendador' dialog box, titled 'Agendador -- Caixa de diálogo Página da Web'. The dialog is used for scheduling a connector. It contains the following fields and options:

- Data e hora de início:** A date and time selector showing '00' for hour and '00' for minute, with a dropdown for 'de acordo com o fuso horário' set to 'GMT'.
- Padrão de recorrência:** Radio buttons for 'Diário', 'Semanal', and 'Mensal'. The 'Diário' option is selected, and there is a field for 'A cada' followed by a dropdown set to '1' and the text 'dia(s)'. There is also an option for 'Todos os dias da semana'.
- Intervalo entre recorrências:** A section with 'Início' set to '10/02/2009' and a calendar icon. It has two radio buttons: 'Sem data de término' (selected) and 'Termina em:' followed by a date field.
- Repetir tarefa:** A checkbox labeled 'Repetir tarefa' which is checked.
- A cada:** Radio buttons for 'minuto(s)' and 'hora(s)'. The 'minuto(s)' option is selected, and there is a dropdown set to '05'.
- Data e hora de término:** A section with 'hora(s)' set to '23' and 'minuto(s)' set to '59'.
- Buttons:** At the bottom, there are three buttons: 'OK', 'Cancelar', and 'Remover programação'.

Após a inclusão de uma programação, e quando o conector for implantado em seguida, é solicitado ao usuário digitar as credenciais da conta no servidor que executa a tarefa. Deve ser usada uma conta de serviço, como a criada para executar o sistema do CA Business Service Insight (OblicoreSrv é o padrão) ou outra conta administrativa, se necessário.

Esse agendador integrado é um recurso muito útil, pois permite ao usuário da GUI controlar a programação do conector sem ter que acessar diretamente a área de trabalho do servidor de aplicativos (conforme as permissões do usuário, naturalmente).

Estudo de caso 21: exemplo de integração LDAP

Requisito de organização

Utilize os usuários existentes definidos no servidor LDAP da organização. Além disso, o portal da organização é usado para o acesso e o logon no CA Business Service Insight por meio dos recursos de logon silencioso do CA Business Service Insight para os portais SSO (Single Sign On).

Defina um script de conversão de VB (Visual Basic) para a criação do usuário automático no sistema do CA Business Service Insight (sincronização LDAP). O script de conversão é usado para estabelecer conexão com o servidor LDAP da organização e extrair a lista de usuários. Os métodos do pacote de ferramentas do CA Business Service Insight são usados para a criação de usuários, grupos e funções.

Exemplo de código VB da conexão LDAP

```
Option Explicit On
Imports System.DirectoryServices

Public Function GetLDAPUsers(ByVal ldapServerName As String, ByVal pFindWhat
As String) As ArrayList
    Dim oSearcher As New DirectorySearcher
    Dim oResults As SearchResultCollection
    Dim oResult As SearchResult
    Dim RetArray As New ArrayList
    Dim mCount As Integer
    Dim mIdx As Integer
    Dim mLDAPRecord As String
    Dim ResultFields() As String = {"securityEquals", "cn"}
    Try
        With oSearcher
            .SearchRoot = New DirectoryEntry("LDAP://" & ldapServerName & _
"/dc=lippogeneral,dc=com")
            .PropertiesToLoad.AddRange(ResultFields)
            .Filter = "cn=" & pFindWhat & "*"
            oResults = .FindAll()
        End With
        mCount = oResults.Count
        If mCount > 0 Then
            For Each oResult In oResults
                mLDAPRecord =
oResult.GetDirectoryEntry().Properties("cn").Value & " " &
oResult.GetDirectoryEntry().Properties("mail").Value
                RetArray.Add(mLDAPRecord)
            Next
        End If
    Catch e As Exception
        MsgBox("Error is " & e.Message)
        Retornar RetArray
    End Try
End Function
```

```
Retornar 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
```

Métodos do script de conversão de VB do CA Business Service Insight

- Métodos de organização
AddOrganization/IsOrganizationExists
- Métodos de função
IsRoleExists/SearchRoles
- Métodos de usuário
AddUserByMap/GetUserName
GetOrganizationName/IsUserExists
GetUserDetails/SearchUsers
GetUserFullName/UpdateUserByMap
- Métodos de grupo de usuários
AddUserGroupByMap/IsUserGroupExists
DeleteUserGroup/SearchUserGroups
GetUserGroupDetails/UpdateUserGroupByMap

Crie o código de “logon silencioso” e integre-o ao portal da organização para ser usado no logon do CA Business Service Insight.

Exemplo de código C# no gateway do CA Business Service Insight (a ser integrado ao portal da organização)

```
using System;  
using System.Data;  
using System.Configuration;  
using System.Collections;
```



```
using System.ComponentModel;
using System.Drawing;
using System.Web;
using System.Web.Security;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Security.Cryptography.X509Certificates;
using OblicoreAuthenticationWebService;

namespace Oblicore.SSO
{
    /// <summary>
    /// This sample page is a sample gateway to Oblicore Guarantee(tm)
    application interface
    /// The page should be called prior navigating to any page at Oblicore
    Guarantee website
    /// or any page using Web Services supplied by Oblicore
    /// The OblicoreGateway page should perform the following actions:
    ///      1) Call Oblicore Authentication Web service in order to
    authenticate current user
    ///      2) Call SilentLogin.asp page at Oblicore website to login
    silently at Oblicore website
    ///      and create user session context
    ///      3) Redirect to desired page
    /// </summary>
    public partial class _Default : System.Web.UI.Page
    {

        /// <summary>
        /// Oblicore user credentials
        /// </summary>
        struct UserCredentials
        {
            public string UserName;
            public string Organization;
        }

        private void Page_Load(object sender, System.EventArgs e)
        {

            if (Request["OGSESSIONID"]!=null)
            {
                //We have been redirected back to this page from
                SilentLogin.asp after authentication.
                //Save OGSESSIONID in cookie for further use
            }
        }
    }
}
```

```

        HttpCookie SessionCookie = new
HttpCookie("OGSESSIONID", Request["OGSESSIONID"]);
        Response.Cookies.Add(SessionCookie);
        //Redirect to desired page
        Response.Redirect("/");
    }
    else
    {
        //First time we enter the page.
        //Let's perform authentication.

        string sAuthToken = string.Empty;

        // Obtain OG user name and organizations from portal user
directory
        UserCredentials ucOblicoreUser =
GetOblicoreUserCredentials();

        //Initialize Oblicore Authentication WebService
        //Project should include Web Reference to the service
        //Service is located on Oblicore Guarantee website at
/WebServices/OblicoreAuth.asmx
        OblicoreAuth oAuthService = new OblicoreAuth();
        // oAuthService.ClientCertificates.Add(x509);
        oAuthService.Url = "https://" + "localhost" +
/WebServices/OblicoreAuth.asmx";
        try
        {
            //Invoke authentication Web Service.
            //The AuthenticateUser method returns encrypted
token, which should be passed to
            //SilentLogin.asp page, located in root folder of
Oblicore Guarantee website
            sAuthToken =
oAuthService.AuthenticateUser(ucOblicoreUser.UserName, ucOblicoreUser.Organization
);
        }
        catch (Exception ex)
        {
            //Proceed authentication error if any
            Response.Write("The error has occurs during
Oblicore authentication: " + ex.Message);
            Response.End();
        }

        //Call SilentLogin.asp page along with passing it
authentication folder
        //SilentLogin.asp page is located Oblicore Guarantee
website root folder
    
```

//After logging in, SilentLogin.asp page will redirect us back to the current page along with passing OGSESSIONID parameter

```
//Response.Redirect(ConfigurationSettings.AppSettings["OGURL"].ToString() +
"/SilentLogin.asp?AuthToken="+Server.UrlEncode(sAuthToken)+"&DesiredPage="+GetCurrentPageURL());
```

```
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 always 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);
    }

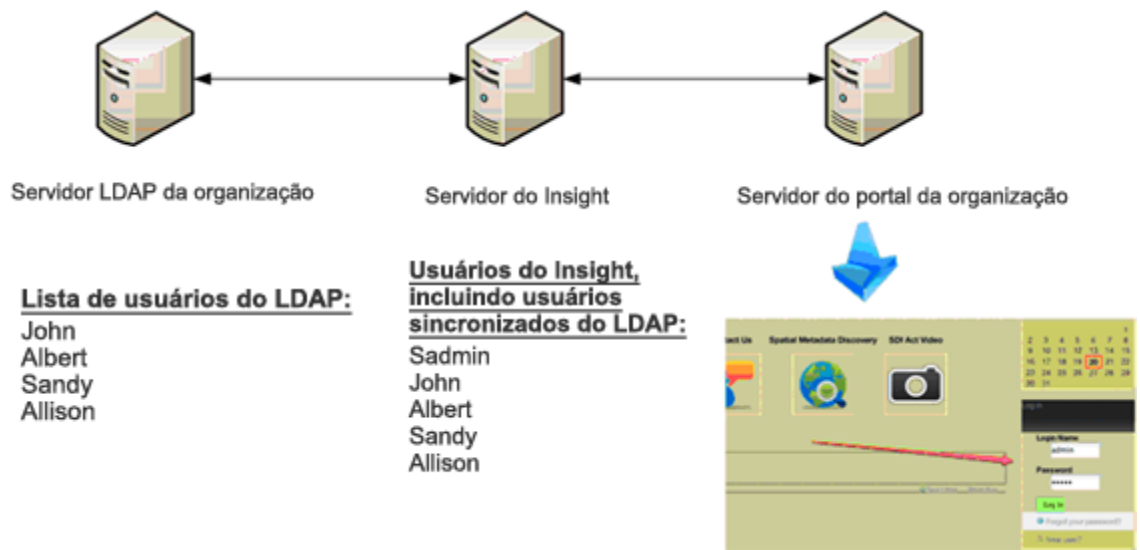
    /// <summary>
    /// Required method for Designer support - do not modify
    /// the contents of this method with the code editor.
    /// </summary>
    private void InitializeComponent()
    {
        this.Load += new System.EventHandler(this.Page_Load);
    }
    #endregion
}

```

■ Diagrama de fluxo

O diagrama abaixo mostra o processo de sincronização de usuários e o fluxo de acessos ao portal. O script de conversão é configurado para ser executado periodicamente. O script mantém a lista de usuários de LDAP atualizada e adiciona e remove os usuários, conforme necessário.

Os usuários executam um logon no portal da organização. O portal pode ser executado para redirecioná-los para o servidor do CA Business Service Insight ou exibir uma lista dos outros aplicativos disponíveis. O servidor do CA Business Service Insight usa as credenciais fornecidas ao efetuar logon no portal inicial.



Estudo de caso 22: gerando relatórios com o PERL

O exemplo abaixo mostra como usar o script do PERL para estabelecer conexão com o serviço web do relatório do CA Business Service Insight e transferir o parâmetro do xml de critérios em um envelope SOAP usando um fluxo HTTP.

Observação: o código do XML transferido no envelope SOAP não pode conter os símbolos "<" e ">". Use o código HTML para representá-los (por exemplo, <=> >=<).

```
#!/usr/bin/perl

#use strict;

use LWP::UserAgent;
use HTTP::Request::Common;
use XML::Simple;
use Data::Dumper;

my $userAgent = LWP::UserAgent > new(agent => 'Mozilla/5.0');

### Creating a Oblicore Session Context - Oblicore Session ID is stored in $scid
###

my $message = "<?xml version=\"1.0\" encoding=\"utf-8\"?>
<soap:Envelope xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\"
xmlns:xsd=\"http://www.w3.org/2001/XMLSchema\"
xmlns:soap=\"http://schemas.xmlsoap.org/soap/envelope/\"
  <soap:Body>
    <CreateSessionContext xmlns=\"http://www.oblicore.com\"
      <userName>sadmin</userName>
      <organizationName>Oblicore</organizationName>
    </CreateSessionContext>
  </soap:Body>
</soap:Envelope>";

my $response = $userAgent > request(POST
'http://obonob/WebServices/OblicoreAuth.asmx',Content_type => 'text/xml', Content
=> $message);

print $response > error_as_HTML unless $response > is_success;

my $result = $response > as_string;

my $xmlstart      = index $result, "<?xml";
my $xmlend        = length $result;

my $xmltext       = substr $result, $xmlstart, ($xmlend - $xmlstart);

my $xml           = new XML::Simple;
```

```

my $data          = $xml > XMLin($xmltext);

my $scid = ($data > {'soap:Body'} > {'CreateSessionContextResponse'} >
{'CreateSessionContextResult'});

print "Session ID : ".$scid."\n";

### Try to get contract list from Oblicore ###

my $criteria_xml =
"<&lt;Criteria&&gt;&&lt;Name&&gt;a*&&lt;/Name&&gt;&&lt;Status&&gt;EFFECTIVE&&lt;/Status&&gt;
&&lt;/Criteria&&gt;";

print "<?xml version=\"1.0\" encoding=\"utf-8\"?>
<soap:Envelope xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\"
xmlns:xsd=\"http://www.w3.org/2001/XMLSchema\"
xmlns:soap=\"http://schemas.xmlsoap.org/soap/envelope/\">
  <soap:Body>
    <GetContractsAdvanced xmlns=\"http://www.oblicore.com\">
      <sessionID>".$scid."</sessionID>
      <criteriaXML>".$criteria_xml."</criteriaXML>
    </GetContractsAdvanced>
  </soap:Body>
</soap:Envelope>";

my $message = "<?xml version=\"1.0\" encoding=\"utf-8\"?>
<soap:Envelope xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\"
xmlns:xsd=\"http://www.w3.org/2001/XMLSchema\"
xmlns:soap=\"http://schemas.xmlsoap.org/soap/envelope/\">
  <soap:Body>
    <GetContractsAdvanced xmlns=\"http://www.oblicore.com\">
      <sessionID>".$scid."</sessionID>
      <criteriaXML>".$criteria_xml."</criteriaXML>
    </GetContractsAdvanced>
  </soap:Body>
</soap:Envelope>";

#my $message = "<?xml version=\"1.0\" encoding=\"utf-8\"?>
#<soap:Envelope xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\"
#xmlns:xsd=\"http://www.w3.org/2001/XMLSchema\"
#xmlns:soap=\"http://schemas.xmlsoap.org/soap/envelope/\">
#  <soap:Body>
#    <GetContracts xmlns=\"http://www.oblicore.com\">
#      <sessionID>".$scid."</sessionID>
#    </GetContracts>
#  </soap:Body>
#</soap:Envelope>";

### is_well_formed($message)

```

```
print $message;

my $response = $userAgent > request(POST
'http://obonob/WebServices/Contracts.asmx', Content_Type => 'text/xml', Content
=> $message);

print $response > error_as_HTML unless $response > is_success;

my $result = $response > as_string;

print Dumper($result); # Output of contract list

### Close the Oblicore Session ###

my $message = "<?xml version=\"1.0\" encoding=\"utf-8\"?>
<soap:Envelope xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\"
xmlns:xsd=\"http://www.w3.org/2001/XMLSchema\"
xmlns:soap=\"http://schemas.xmlsoap.org/soap/envelope/\">
  <soap:Body>
    <ClearSessionContext xmlns=\"http://www.oblicore.com\">
      <sessionContextID>\".$scid.\"</sessionContextID>
    </ClearSessionContext>
  </soap:Body>
</soap:Envelope>";

my $response = $userAgent > request(POST
'http://obonob/WebServices/OblicoreAuth.asmx', Content_Type => 'text/xml',
Content => $message);

print $response > error_as_HTML unless $response > is_success;
```


Apêndice C: Especificações de configuração do conector

Esta seção contém os seguintes tópicos:

[Estrutura geral](#) (na página 313)

[Seção Geral](#) (na página 314)

[Seção Interface do CA Business Service Insight](#) (na página 320)

[Seção DataSourceInterface](#) (na página 323)

[Seção da interface SQL](#) (na página 326)

[Seção InputFormatCollection](#) (na página 330)

[Seção TranslationTableCollection](#) (na página 335)

[Seção TranslatorCollection](#) (na página 337)

Estrutura geral

A estrutura geral do arquivo de configuração XML é a seguinte:

```
< AdapterConfiguration>
  <General...>
  <OblicoreInterface...>
  <DataSourceInterface...>
  <InputFormatCollection...>
  <TranslatorCollection...>
  <TranslationTableCollection...>
</ AdapterConfiguration>
```

Cada um dos nós é descrito na seção abaixo.

Seção Geral

A seção Geral consiste em atributos e subnós, como segue:

Estrutura do XML:

```
<General
  MajorVersion="2345"
  MinorVersion="01245"
  WorkingDirectoryName=" output"
  DataSourceControlFileName="DatasourceControl.xml"
  SendFileName="Send.txt"
  SendControlFileName="SendControl.xml"
  LogDebugMode="no"
  ConsoleDebugMode="no"
  RunOnce="yes"
  RepeatUntilDry="yes"
  RuntimeErrorLimit="1"
  RetryRejectedEvents="yes"
  RejectedEventsFileName="rejectedEvents.txt"
  RejectedEventsUpperLimit="1000"
  RegExUnlimitedSearch (V3.1 Patch)="no"
  HoldRejectedEventsSpan="24"
  GenerateStatistics="yes"
  StatisticsFileName="MyStatistics.txt" >
  KeepHistoricState="yes" >
  DefaultTimeFormat="%Y/%m/%d-%H:%M:%S"
  DefaultDecimalSymbol="."
  DataSourceIdMaxLength="60"
  DefaultDigitGroupingSymbol=","
  SaveIllegalEvents ="no"
  WriteEventErrorToLog ="yes"
  IllegalEventsDirectoryName="xxxpath"
  <DataSourceDifferenceFromUTC
    DefaultOffset="2"
    TimeFormat="%Y/%m/%d-%H:%M:%S">
    <DayLightSaving
      From="2001/04/09-12:00:00"
      To="2001/09/01-12:00:00"
      Shift="1"/>
  </ DataSourceDifferenceFromUTC >
</General>
```

- MajorVersion - especifica o número da versão principal.
- MinorVersion - especifica o número da versão secundária.
- WorkingDirectoryName (opcional) -

especifica o caminho padrão para os arquivos de saída do conector (controle da origem de dados, conversão, envio).

Valor padrão: diretório de saída (pasta) no diretório do arquivo de configuração.

■ DataSourceControlFileName (opcional) -

especifica o nome do arquivo que o conector usa para acompanhar os últimos registros de dados recuperados da origem de dados.

Valor padrão: DataSourcecontrol.xml (se não for especificado, é usado o valor padrão).

■ SendFileName (opcional) -

especifica o nome do arquivo contendo os eventos antes de serem enviados ao CA Business Service Insight.

Valor padrão: Send.txt (se não for especificado, é usado o valor padrão)

■ SendControlFileName (opcional) -

especifica o nome do arquivo que o conector usa para monitorar o processo de envio.

Valor padrão: SendControl.xml (se não for especificado, é usado o valor padrão)

■ DataSourcecldMaxLength -

define o comprimento máximo para o campo DataSourcecld na tabela t_raw_data. Se o usuário inserir um texto mais longo, haverá erro no conector.

Este valor deve ser inferior ou igual ao tamanho real da tabela.

Valor padrão: 60

■ SaveIllegalEvents - se selecionado, este atributo indica que os eventos inválidos devem ser gravados no arquivo.

Valor padrão: no

■ WriteEventErrorToLog - indica se erros nos dados devem ser gravados na tabela T_Log.

Valores válidos: "yes" e "no"

Valor padrão: "yes"

■ IllegalEventsDirectoryName - não há valor padrão

■ SendFileName (opcional) -

especifica o nome do arquivo contendo os registros antes de serem enviados ao CA Business Service Insight.

Valor padrão: Send.txt (se não for especificado, é usado o valor padrão).

■ \$SendControlFileName (opcional) -

especifica o nome do arquivo que o conector usa para monitorar o processo de envio.

Valor padrão: SendControl.xml (se não for especificado, é usado o valor padrão).

■ LogDebugMode (opcional)

Valores válidos: "yes" e "não"

quando definido como "yes", o item seguinte é enviado à linha log-original da origem de dados, dos resultados da análise, CA Business Service Insight Unified Event.

- ConsoleDebugMode (opcional)

Valores válidos: "yes" e "não"

quando definido como "yes", são exibidas mensagens de depuração no console.

- **Indicadores para a leitura e a conversão de registros:**

- .: para um registro lido e convertido em um evento de saída.
- i: para um registro ignorado pelo analisador de expressão regular.
- l: para um registro lido e ignorado pelas tabelas de conversão.
- R: para um registro lido, mas recusado pela tabela de conversão (não pode ser convertido pelas tabelas de conversão).
- X: para um registro que apresentou problema durante a análise. Ele será ignorado e perdido ou será salvo na pasta de eventos inválidos.

Observação: quando o registro lido passa por mais de um conversor, o indicador do registro é colocada entre parênteses. Por exemplo: [...] ou [RRI].

- **Indicadores de status do conector:**

- O: o conector está ativo e não leu nenhum registro no último segundo.
- E: status Erro.
- P: status Pausado.
- S: comando Interromper recebido do CA Business Service Insight.
- B: status Bloqueado. A tabela de eventos recusados está completa.
- **Indicadores de tabelas de conversão:**
- L: aguardando tabelas de conversão.
- T: tabela de conversão enviada pelo/recebida do CA Business Service Insight.
- t: alterações na tabela de conversão enviadas pelo/recebidas do CA Business Service Insight.
- **Indicadores de conexão:**
- <Connecting CA Business Service Insight: tentando estabelecer conexão com o CA Business Service Insight.
- <Connecting CA Business Service Insight>: conexão estabelecida.
- <Connecting DataSource: tentando estabelecer conexão com a origem de dados.
- <Connecting DataSource>: conexão estabelecida.

- RunOnce (opcional)

Valores válidos: "yes" e "não"

quando definido como "no", o conector é executado continuamente.

Quando definido como "yes", o conector do arquivo de texto é executado, lido, registrado e interrompido automaticamente. O conector de arquivo lê arquivos inteiros, aguarda alguns segundos, tenta ler novos registros (conforme o SleepTime). Se tais registros não existirem, ele é interrompido.

Um conector de SQL executa cada consulta apenas uma vez. Se RepeatUntilDry estiver definido como "no", ele é interrompido imediatamente. Se RepeatUntilDry estiver definido como "yes", ele aguarda (conforme o SleepTime), tenta executar as consultas novamente (conforme o SleepTime da consulta) e, se não houver novos registros, ele é interrompido.

- RepeatUntilDry (opcional)

Valores válidos: "yes" e "no" - valor padrão: "yes"

Consulte o atributo RunOnce acima.

- RuntimeErrorsLimit (opcional) -

determina o limite de erros (por exemplo, erros em eventos de entrada), antes de o conector ser bloqueado. Quando definido como 0, o conector não bloqueado.

Valor padrão: 1 (o que significa que o conector é bloqueado após o primeiro erro)

- RetryRejectedEvents (opcional)

Valores válidos: "yes" e "no" - valor padrão: "yes"

Quando definido como "yes", os registros que precisam de conversão são mantidos no arquivo de eventos recusados enquanto aguardam a conversão.

Não é recomendado definir este atributo como "no" para evitar a perda de dados importantes.

- RejectedEventsFileName (opcional) -

o nome do arquivo que o conector usa para acompanhar os registros de dados recuperados a partir da origem de dados que aguardam a conversão.

Valor padrão: rejected.txt (se não for especificado, é usado o valor padrão).

- RejectedEventsUpperLimit (opcional) -

determina o limite dos registros recusados que são mantidos no arquivo rejected.txt. Quando o conector atinge esse limite superior, ele é interrompido e alterado para o status bloqueado até que esses registros sejam convertidos.

Valor padrão: 1000

- RegExUnlimitedSearch - orienta o conector a realizar pesquisas completas, conforme a expressão regular.

Valores válidos: "yes" e "não"

Valor padrão: no

■ **HoldRejectedEventsSpan** (opcional) -

determina o número de horas durante as quais salvar o arquivo de eventos recusados. Se este atributo não for mencionado, o conector não apaga os eventos, os quais devem ser manipulados antes de o conector ser iniciado novamente.

Não é recomendado usar este atributo, pois há risco de perda de dados importantes.

■ **GenerateStatistics** (opcional)

Valores válidos: "yes" e "no" - valor padrão: "yes"

Quando definido como "yes", o conector cria um arquivo contendo informações estatísticas a cada minuto.

■ **StatisticsFileName** (opcional) -

especifica o nome do arquivo de estatística.

Valor padrão: statistics.txt (se não for especificado, é usado o valor padrão).

■ **KeepHistoricState** (opcional)

Valores válidos: "yes" e "no" - valor padrão: "no"

Quando definido como "yes", o conector salva todos os arquivos, exceto pelo arquivo de log, em uma nova pasta chamada Historic state yyyyymmdd-hhmmss, onde yyyyymmdd e hhmmss são os formatos de data e hora de criação.

■ **DefaultTimeFormat** (opcional) -

especifica o formato de data e hora padrão. Se este atributo for especificado, usa-se o formato de data e hora onde o atributo TimeFormat estiver ausente. Se ele não for especificado, o atributo TimeFormat em outros elementos é obrigatório.

■ **DefaultDecimalSymbol** (opcional) -

especifica o símbolo decimal padrão para os campos reais.

Valor padrão: se não for especificado, é usado o valor padrão

■ **DefaultDigitGroupingSymbol** (opcional) -

especifica o símbolo de agrupamento de dígitos padrão para números inteiros e campos reais.

Valor padrão: se não for especificado, é usado o valor padrão

■ **DataSourceDifferenceFromUTC** - indica a diferença de horário entre UTC e o fuso horário do computador da origem de dados.

- **DefaultOffset** - especifica o deslocamento do UTC fora do horário de verão.
- **TimeFormat** - especifica o formato usado para analisar o horário de verão (descrito em seguida).

- DayLightSaving - especifica um período de horário de verão do fuso horário da origem de dados. Este elemento é opcional (o que significa que, se não for selecionado, não há horário de verão) e pode ter mais de uma ocorrência. Quando há mais de um elemento, eles devem ser apresentados por ordem de horário e não devem se sobrepor.
 - From - data de início do período.
 - To - data de término do período.
 - Shift - mudança de horário adicionada a DefaultOffset durante o horário de verão.

Seção Interface do CA Business Service Insight

A seção Interface do CA Business Service Insight consiste em atributos especificando a conexão com o CA Business Service Insight. Há dois modos: online e offline. No modo online, o conector se conecta ao CA Business Service Insight, obtém as tabelas de conversão e o comando de controle do CA Business Service Insight, e envia eventos, status e solicitações de conversão ao CA Business Service Insight. No modo offline, o conector usa um arquivo de tabelas de conversões local e grava os eventos no arquivo de saída.

Estrutura do XML para o modo online:

```
<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 - o número da porta TCP/IP que o conector usa para se comunicar com o servidor do CA Business Service Insight.
- ConnectionInitiator (opcional)
Valores válidos: "server" e "adapter" - valor padrão: "server"

Define o iniciador da conexão, da parte do conector ou o listener de conectores da parte do CA Business Service Insight.

O iniciador atua como 'subordinado' ou cliente, enquanto que o outro atua como o 'mestre' ou servidor.
- Address - o endereço de rede do servidor é obrigatório quando o iniciador é o conector.
- SecurityLevel - define o nível de autenticação entre o conector e o servidor do CA Business Service Insight ("Hand Shake"). Opções de nível:
 - none - nenhuma autenticação é realizada.
 - high - a autenticação é realizada. A SecurityKey deve ser fornecida.
- SecurityKey - texto de 32 dígitos. É preciso que seja o mesmo texto que o definido para o conector no banco de dados do CA Business Service Insight.

O fluxo do processo de "handshake" é o seguinte:

- O servidor do CA Business Service Insight estabelece conexão com o conector.
- Um texto aleatório é enviado do conector ao servidor do CA Business Service Insight.
- O servidor do CA Business Service Insight criptografa o texto com uma chave pré-configurada e envia o resultado de volta ao conector.
- O conector criptografa o mesmo texto aleatório enviado ao servidor do CA Business Service Insight usando o texto da SecurityKey e compara os resultados.
- O servidor do CA Business Service Insight torna um texto aleatório e envia o texto ao conector.
- O conector criptografa o texto e o envia de volta ao servidor do CA Business Service Insight.
- O servidor do CA Business Service Insight criptografa o mesmo texto e compara os resultados com o que foi recebido do conector.
- A conexão é estabelecida.
- Se algum erro for detectado em alguma fase do fluxo, a conexão não foi estabelecida.

■ UseAcknowledgeProtocol (opcional)

Valores válidos: "yes" e "no" - valor padrão: "yes"

Quando definido como yes, o conector usa o protocolo de reconhecimento. Quando definido como no, o conector envia as mensagens e os pacotes, sem aguardar o reconhecimento do CA Business Service Insight.

Não é recomendado definir este atributo como "no" para evitar a perda de eventos importantes.

■ PacketSize (opcional)

Valores válidos: de 1 a 1000

Valor padrão: 50

Número máximo de eventos em um pacote.

■ PacketDeadline (opcional)

Valores válidos: de 1 a 3600

Valor padrão: 60

Número de segundos antes de enviar um pacote que não esteja completo.

■ PacketResendTimeout (opcional)

Valores válidos: de 1 a 3600

Valor padrão: 60

Quantidade de tempo em segundos a aguardar pela mensagem de reconhecimento antes de enviar o pacote novamente. Este atributo aplica-se apenas quando o atributo UseAcknowledgeProtocol é definido como yes.

- WindowSize (opcional)

Valores válidos: de 1 a 100

Valor padrão: 10

Número de pacotes na janela. Este atributo aplica-se apenas quando o atributo UseAcknowledgeProtocol é definido como yes.

Estrutura do XML para o modo offline:

```
<OblicoreInterface
  Mode="offline">
  <OfflineInterface
    OutputFileName="outputEvents.txt"
    OutputFileType="tsv"
    OutputTimeFormat="%Y/%m/%d %H:%M:%S"
    OutputDecimalSymbol="."
  />
</OblicoreInterface>
```

- OutputFileName (opcional) -

Nome do arquivo em que o conector grava os eventos de saída.

Valor padrão: AdapterOutput.txt (se não for especificado, é usado o valor padrão).

- OutputFileType (opcional)

Valores válidos: "csv", "tsv" e "xml"

Define o formato do evento.

- OutputTimeFormat - define o formato dos campos de data. Este atributo pode ser omitido se o atributo DefaultTimeFormat, na seção General, tiver sido definido.

- OutputDecimalSymbol (opcional) -

Define o símbolo decimal para ler os campos.

Valor padrão: . (ponto)

Seção DataSourceInterface

A seção DataSourceInterface consiste nos atributos que especificam a conexão e o tipo de conexão entre o conector e a fonte de dados (ferramenta de medição, CRM, log do sistema, etc.) e é dividida em dois tipos principais: interface de arquivo e interface SQL.

Interface de arquivo

O conector de arquivo pode ser usado para recuperar dados dos arquivos de log, relatórios programados ou qualquer outro arquivo de texto, e o DataSourceInterface define as regras usadas para analisar as informações da fonte de dados do arquivo e extraí-las para os campos.

A seção DataSourceInterface também define como o conector lida com o arquivo de origem (se ele exclui o arquivo original, caso tenha sido criado apenas para o conector, ou se mantém os dados como estão, caso sejam necessários a outros usos, e assim por diante).

Estrutura do XML:

```
<DataSourceInterface WorkFileName="MyWorkingFile.txt" >
  <Files>
    <File
      IsActive="yes"
      InputFormat="events"
      Path="D:\adapters\sample_data\"
      NamePattern="adapterXXX*.log"
      DeleteFileAfterProcessing="no"
      Delimiters=", "
      IgnoreRedundantDelimiters ="no"
      TitleExists="no"
      SleepTime="10">
    </File>
  </Files>
</DataSourceInterface>
```

- WorkFileName (opcional) - quando DeleteFileAfterProcessing está definido como 'no', o arquivo é copiado para esse nome, e quando está definido como 'yes', o arquivo é renomeado para esse nome. Se nenhum valor for especificado, será usado o padrão ('WorkFile.txt').
 - Files - coleta dos elementos do arquivo (é possível definir mais de um arquivo por conector).
 - File - especifica os atributos do arquivo.
 - IsActive [yes/no] (opcional) - define se o arquivo está ativo (quando definido como "no", esse arquivo não será lido).
 - InputFormat - especifica o formato de entrada associado ao arquivo. O conector usa o InputFormat para extrair os dados da fonte de dados.

- Path - especifica o caminho dos arquivos da origem de dados.
- NamePattern - especifica o nome do arquivo da fonte de dados. É possível usar caracteres curinga se mais de um arquivo usar o mesmo formato de entrada.
- DeleteFileAfterProcessing [yes|no] - especifica a forma como o conector lida com o arquivo de origem. Quando o arquivo for criado apenas para o conector e puder ser excluído após o processamento, a definição deve ser "yes". O arquivo será, então, renomeado, processado e excluído. Se a definição for "no", o arquivo é copiado, e o processamento ocorrerá no arquivo copiado. Caso novos registros sejam adicionados ao final deste arquivo, o conector copia esses novos registros no arquivo de trabalho no próximo ciclo. Se não forem adicionados novos registros, o conector procura o primeiro arquivo com o mesmo padrão e um nome maior que o arquivo atual (em ordem lexicográfica). Se o conector localizar tal arquivo, ele prossegue o trabalho com ele. O conector não reverte para o arquivo anterior mesmo se novos registros forem adicionados ao arquivo. Use "no" quando precisar manter a integridade do arquivo de origem.
- InitialFileName - especifica o nome do primeiro arquivo a partir do qual o conector fará uma pesquisa usando o devido padrão. Use este atributo quando NamePattern tiver caracteres curinga e você desejar que o conector leia arquivos antigos.
- Delimiters (opcional) - especifica um ou mais caracteres que atuam como delimitadores, conforme os quais as linhas de dados serão divididas em campos. Se um valor não for definido, usa-se o padrão ("t").
- IgnoreRedundantDelimiters [yes/no] (opcional) - quando definido como "yes", os delimitadores consecutivos são tratados como um só. Caso contrário, será criado um campo em branco entre os delimitadores.
- RegExForParser (opcional) - trata-se de uma expressão regular usada para extrair campos da fonte de dados substituindo os delimitadores especificados anteriormente. Por exemplo:

```
<File  
...  
RegExForParser="^(.{8}) (.{6}) (?:[ ])*([0-9]+) "  
>
```

Consulte a documentação para obter mais detalhes sobre as expressões regulares.

- TitleExists [yes/no] (opcional) - especifica se a primeira linha que não estiver em branco no arquivo de origem de dados é uma linha de título. O título pode ser usado pelo conector ao analisar os dados.
 - SleepTime - especifica o intervalo entre as recuperações de dados (em segundos), ou seja, os segundos decorridos entre cada recuperação de dados pelo conector a partir do arquivo de dados de origem.
 - LogicLineDefinition (opcional)
- ```
<File
```

```
...
<LogicLineDefinition
 FirstLine="Job server:.*"
 NumberOfLines="5"
>
/>
```

Caso o conjunto de dados tenha sido criado conforme um determinado número de linhas, e não em uma só linha, os seguintes atributos definem o ponto de início e o ponto de término da extração, bem como a quantidade de linhas de dados:

- AllFile [yes/no] (opcional) - quando definido como "yes", todo o arquivo é considerado como um só registro em uma linha lógica.
- FirstLine (opcional) - uma expressão regular que define a primeira linha da linha lógica. Ela pode ser especificada com e sem LastLine e/ou NumberOfLines.
- LastLine (opcional) - uma expressão regular que define a última linha da linha lógica. Ela pode ser especificada com e sem FirstLine e/ou NumberOfLines.
- NumberOfLines (opcional) - especifica a quantidade de linhas em uma linha lógica. Ela pode ser especificada com e sem FirstLine e/ou LastLine.
- MatchCase [yes/no] (opcional) - define se a expressão regular correspondente diferencia maiúsculas e minúsculas.

## Seção da interface SQL

Os conectores do SQL podem ser usados para recuperar dados de bancos de dados usando uma instrução SQL.

A interface do SQL define a conexão com o banco de dados e as consultas usadas para recuperar os dados como segue:

Estrutura do XML:

```
< DataSourceInterface >
 <ConnectionString ConnectionTimeout="60" QueryTimeout="30">
 <![CDATA[Driver={Microsoft Access Driver
(*.mdb)};DataBase=d:\Oblicore\database1.mdb;]]>
 </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;
DataBase="/>
 <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 - texto da conexão para acessar o banco de dados de origem.

ConnectionString pode ser definido no elemento DataSourceInterface e/ou nos elementos da consulta. A definição de ConnectionString no elemento DataSourceInterface é o padrão e é usada apenas em uma consulta sem a definição de ConnectionString.

O texto da conexão pode ser definido em uma sequência de caracteres ou por segmentos. Quando o elemento ConnectionString não contém elementos de segmento, o texto da conexão é obtido a partir do texto do elemento ConnectionString. Se ele contém pelo menos um elemento de segmento, o texto da conexão é concatenado a partir dele.

Há dois tipos de segmento. O primeiro é texto e contém texto concatenado ao texto da conexão na forma em que se encontra. O segundo é arquivo e contém um nome de arquivo com ou sem caracteres curinga. O segmento do arquivo pode aparecer apenas uma vez e contém outros atributos que definem o que fazer com o arquivo de leitura.

- ConnectionTimeout (opcional) - especifica o número inteiro positivo que indica, em segundos, quanto tempo aguardar até abrir a conexão. O valor 0 indica que se deve aguardar indefinidamente até que a conexão seja aberta. O valor padrão é 15 segundos.

**Observação:** alguns provedores não oferecem suporte a essa funcionalidade.

- QueryTimeout (opcional) - especifica o número inteiro positivo que indica, em segundos, quanto tempo aguardar até executar a consulta. O valor 0 indica que se deve aguardar indefinidamente até que a execução esteja concluída. O valor padrão é 30 segundos.

**Observação:** alguns provedores não oferecem suporte a essa funcionalidade.

- Segment - especifica os atributos do segmento.
- Type (text, file) (opcional) - especifica o tipo de segmento.
- Text - especifica o texto do segmento de texto.
- File - especifica os atributos do arquivo.
- Path - especifica o caminho dos arquivos da origem de dados.
- NamePattern - especifica o nome do arquivo de origem de dados e aceita caracteres curinga.
- InitialFileName (opcional) - Informa ao usuário a partir de qual arquivo iniciar a pesquisa e o padrão a ser pesquisado.
- UsePath [yes, no] (opcional) - quando definido como yes, o caminho do arquivo é concatenado com o texto da conexão.
- UseFileName [yes/no] (opcional) - quando definido como yes, o nome do arquivo é concatenado com o texto da conexão (necessário para arquivos em Excel).
- UpdateSchemaFile [yes/no] (opcional) - quando definido como yes, o conector atualiza o arquivo schema.ini com o nome de arquivo atual.

**Observação:** use este atributo apenas quando desejar que o conector altere o arquivo schema.ini (banco de dados para arquivo de texto).

- QueryCollection - especifica um conjunto de consultas (é possível definir mais de uma consulta por conector).
  - PreliminaryCheck-: (opcional [sim/não]). o conector verifica as consultas ao conectar-se ao banco de dados. Se este atributo estiver definido como "no", o conector verificará as consultas executando-as sem nenhuma alteração. Em seguida, o conector é executado nos registros de retorno e não os lê novamente. Se definido como "yes", o conector substitui cada ocorrência de "where" na instrução selecionada por "where 1=2 and" e só então executa as consultas. Use esta opção caso deseje verificar todas as consultas antes de o conector iniciar a execução de todos os registros e quando esta adição reduzir drasticamente o tempo da consulta. Note - alguns provedores não otimizam o processo de consulta, e quando a consulta é válida, a execução leva o mesmo tempo que levaria sem a adição.
  - ExternalCommand (opcional) - especifica o comando externo executado antes de o conector iniciar outro ciclo de leitura.
  - Command - especifica o nome do arquivo em lote a ser executado.
  - SleepTime - especifica um número inteiro positivo que indica, em segundos, quanto tempo esperar até executar o comando novamente.
- Query - especifica os atributos da pesquisa.
  - QueryName - especifica o nome da consulta.
  - IsActive [yes/no] (opcional) - quando definido como "no", o conector não lê registros a partir deste arquivo.
  - InputFormat - especifica o formato de entrada associado a esta consulta. O conector usa o atributo InputFormat para extrair dados do registro de origem.
  - SleepTime - especifica o período, em segundos, durante o qual o conector é desativado enquanto aguarda a chegada de novos dados.
  - Critical [yes/no] (opcional) - se definido como "yes", o conector é interrompido imediatamente ao encontrar um erro na consulta. A opção "no" é usada quando há um erro conhecido que pode ser resolvido sem que se altere o arquivo de configuração.
  - TransactionMode [implicit/explicit] (opcional) - se definido como "explicit", o conector inicia uma transação no banco de dados antes da consulta. Essa opção resolve vários problemas em consultas extensas e complicadas.
  - RollbackSegmentName (opcional) - define qual segmento de reversão o conector usa. Caso contrário, o banco de dados é que seleciona o segmento de reversão.
- SelectStatement: contém a declaração selecionada para ser executada no banco de dados de origem.



- AutoCompleteQuery-(Optional) [yes/no] - se definido como "yes", o conector concatena automaticamente uma instrução where com a consulta especificada que procede da seguinte maneira:
- Cria uma instrução where que retornará apenas valores novos com base em campos chave.
- Ordena os resultados da instrução com base em campos chave.
- QueryKeyFields - define os campos que iniciarão a próxima recuperação de dados.
  - KeyField:
  - Name - especifica o nome do campo obtido dos campos da consulta.
  - Sort [asc/desc] (opcional) - especifica o método de classificação dos dados (crescente/decrescente).
  - Function (opcional) - use este atributo se for necessário que uma função especial opere neste campo. Para combinar o valor do campo na função, use (:fieldname). Por exemplo, usar a função de data do Oracle com um campo chamado Ts-Function="to\_date('ts','dd/mm/yyyy)'"
  - ValueLeftWrapper (opcional) - use este atributo para concatenar caracteres antes do valor do campo. O padrão para os campos de texto e data é "" (apóstrofe).
  - ValueRightWrapper (opcional) - use este atributo para concatenar caracteres depois do valor do campo. O padrão para os campos de texto e data é "" (apóstrofe).
  - NameLeftWrapper (opcional) - use este atributo para concatenar caracteres antes do nome do campo. O padrão é a sequência de caracteres nula.
  - NameRightWrapper (opcional) - use este atributo para concatenar caracteres depois do nome do campo. O padrão é a sequência de caracteres nula.
  - IsPrimaryKey [yes/no] (opcional) - quando definido como "no", esta chave não é usada pelo conector na instrução where automática na consulta.
  - DefaultInitialValue (opcional) - se a consulta SelectInitialValues não retornar nenhum registro, é deste atributo que o conector obtém o padrão. Se houver campos chave, todos eles deverão ter este atributo.
  - SelectInitialValues - especifica uma instrução que fornece os valores iniciais para a primeira instrução where (quando o arquivo de controle está vazio).

**Observação:** a ordem dos valores deve ser a mesma que a dos elementos do campo para QueryKeyFields e deve retornar um resultado para cada campo.

## Seção InputFormatCollection

Esta seção especifica a estrutura dos dados recuperados da origem de dados, como uma linha de dados será dividida em campos e quais serão os tipos de campo e formatos. A filtragem inicial e a manipulação dos dados podem ser executadas nesta seção, usando-se InputFormatSwitch e os campos compostos, respectivamente.

O fluxo de trabalho geral desta seção é o seguinte:

- Uma linha de dados faz correspondência com um ou mais InputFormats.
- Os dados são distribuídos em campos, conforme correspondência com a especificação de InputFormat.
- Aos campos compostos são atribuídos valores por meio da combinação e da divisão de campos de dados.
- Os dados processados são verificados em relação às condições de TranslatorSwitch.
- Eles são enviados ao conversor correspondente ou são ignorados.

O nó InputFormatCollection pode conter um ou mais nós InputFormat.

### Estrutura do XML:

```
<InputFormatCollection>
 <InputFormat InputFormatName="MyInputFormat">
 <InputFormatFields>
 <InputFormatField Name="sid_id" Type="string"/>
 <InputFormatField Name="content" Type="string"/>
 <InputFormatField Name="date" Type="time"
 TimeFormat="%d/%m/%Y %H:%M:%S"/>
 <InputFormatField Name="server" Type="string"
 Source="compound">
 <Compound>
 <Segment SourceField="content"
 RegularExpression=".*Job server: ([^\n]+).*" />
 </Compound>
 </InputFormatField>
 </InputFormatFields>
 </InputFormat>
 <TranslatorSwitch DefaultTranslator="GeoTranslator">
 <TranslatorCase TranslatorName="NonGeoTranslator" Break="yes">
 <Condition SourceField="routing_info" Operator="EQ"
 Value="c nano"/>
 </TranslatorCase>
 </TranslatorSwitch>
</InputFormatCollection>
```

- InputFormat:
  - InputFormatName - especifica um nome para este formato, a ser referenciado pela seção DataSourceInterface.

- RequiredFields (opcional) - especifica o número mínimo de campos que se espera encontrar em uma linha de dados. A linha que contiver menos campos será ignorada, e um erro é registrado no log.
- InputFormatFields - ele pode conter um ou mais nós de campo, de acordo com o número de campos de entrada na origem de dados.
  - InputFormatField - especifica um campo de dados da linha de dados original ou um campo composto.  

```
<InputFormatField Name="timestamp" Type="time"
TimeFormat="%d/%m/%Y %H:%M:%S"/>
```
  - Name - especifica um nome para este campo a ser referenciado por outros elementos.
  - Type - especifica o tipo de dados de field-string/integer/real/time
  - Source (opcional) - o valor padrão obtido é "event". Valores possíveis:
    - event - o valor do campo é obtido do evento proveniente da origem de dados; os valores de campos são obtidos da origem de dados, seguindo a ordem de chegada.
    - compound - este campo é composto. Ele obtém seu valor manipulando outros valores de campo ou constantes.
    - title - o valor do campo é obtido de nomes de campos de título. O campo referido já deve ter sido definido.
    - filename - o valor do campo é obtido do nome de arquivo da origem de dados apenas no caso de conectores de arquivo de texto.
    - constant - o valor do campo é uma constante obtida da propriedade ConstantValue a ser exibida em seguida.
    - FieldTitleName - quando a origem é o título, especifica o título do campo a ser usado. O campo de origem deve ser definido com antecedência.
    - ConstantValue - quando a origem é uma constante, especifica o valor com o qual fazer correspondência. Quando o campo é de data e hora, o valor da constante é um texto formatado de acordo com TimeFormat ou "Now" ou "NowUtc", em que "Now" refere-se à data e à hora atuais na localidade de origem dos dados e "NowUtc" refere-se à data e à hora atuais em UTC.
  - TimeFormat-para campos do tipo time, especifica o formato de data a seguir ao qual o campo é analisado. Os códigos abaixo podem ser usados para definir o formato de data e hora:

| <b>Sinal</b> | <b>Descrição</b>                                            |
|--------------|-------------------------------------------------------------|
| <b>%d</b>    | <b>Dia 1-31</b>                                             |
| <b>%a</b>    | <b>Nome do dia da semana abreviado (dom., seg.)</b>         |
| <b>%A</b>    | <b>Nome do dia da semana por extenso (domingo, segunda)</b> |
| <b>%m</b>    | <b>Mês 1-12</b>                                             |
| <b>%b</b>    | <b>Nome do mês abreviado (jan., fev.)</b>                   |
| <b>%B</b>    | <b>Nome do mês por extenso (janeiro, fevereiro)</b>         |
| <b>%y</b>    | <b>Ano (2 dígitos)</b>                                      |
| <b>%Y</b>    | <b>Ano (4 dígitos)</b>                                      |
| <b>%H</b>    | <b>Hora em formato de 24 horas (00 a 23)</b>                |
| <b>%I</b>    | <b>Hora em formato de 12 horas (01 a 12)</b>                |
| <b>%P</b>    | <b>A.M./P.M. Indicador para formato de 12 horas</b>         |
| <b>%M</b>    | <b>Minutos 0-59</b>                                         |
| <b>%S</b>    | <b>Segundos 0-59</b>                                        |
| <b>%D</b>    | <b>Data completa sem separadores (ddmmaaaa)</b>             |
| <b>%T</b>    | <b>Hora cheia sem segundos e separadores (hhmm)</b>         |
| <b>%n</b>    | <b>Número de segundos desde 1970</b>                        |
| <b>%E</b>    | <b>Data e hora em formato hexadecimal.</b>                  |
| <b>%%</b>    | <b>Sinal de porcentagem</b>                                 |

- DecimalSymbol (opcional) - define o símbolo decimal dos campos. Caso não seja especificado, usa-se DefaultDecimalSymbol, da seção Geral.
- DigitGroupingSymbol (opcional) - especifica o símbolo de agrupamento de dígitos padrão para números inteiros e campos reais. Caso não seja especificado, usa-se DefaultDigitGroupingSymbol, da seção Geral.
- Compound - é necessário na presença de source=compound. Especifica que manipulações de campo sejam coletadas em um campo composto.
- Segment - especifica que uma manipulação de campo seja adicionada ao composto criado. Apenas o atributo SourceField é necessário.

- SourceField - especifica o campo usado como base. O campo referido já deve ter sido definido.
- RegularExpression - usado para manipular expressões regulares.
- MatchCase [yes/no] (opcional) - define se a expressão regular correspondente diferencia maiúsculas e minúsculas.
- SelectionStart - especifica a posição inicial da extração de texto, começando em 0.
- SelectionLength - define o tamanho do texto extraído.
- Prefix - especifica o texto a ser prefixado ao resultado da manipulação.
- Suffix - especifica o texto a ser sufixado ao resultado da manipulação.
- XpathExpression - usado para manipular expressões XPath.

- InputFormatSwitch - usado para especificar os critérios de formato quando há linhas de dados em mais de um formato.

**Observação:** ao usar InputFormatSwitch, a ordem dos nós de InputFormat é importante. InputFormat já deve ter sido definido.

DefaultInputFormat - especifica o nome do InputFormat para o qual rotear, caso não haja correspondência com nenhum critério.

- InputFormatCase - especifica o critério a ser testado em linhas de dados para determinar para qual InputFormat ele deve ser roteado.
- InputFormatName - especifica o InputFormat a ser usado quando há correspondência com o critério.
- LogicOperator [and/or] (opcional)
- and - especifica que é preciso fazer correspondência com todas as condições. (o padrão).
- or - especifica que ao menos uma condição deve ser atendida.
- Condition - especifica a condição a ser testada em uma linha de dados para determinar o formato.

SourceField - especifica o campo a ser testado.

Operator - especifica o tipo do teste. Temos as seguintes opções:

- **EQ** - igual a
- **NE** - não é igual a
- **GT** - maior que
- **LT** - menor que
- **GE** - maior ou igual a
- **LE** - menor ou igual a
- **MATCH** - especifica a correspondência com uma expressão regular

- **UNMATCH** - especifica que não será feita correspondência com uma expressão regular

ValueType [constant/field/previousValue] (opcional) -

- constant - o conteúdo do atributo Value é constante, independentemente dos dados da origem.
- field - o conteúdo do atributo Value é o nome do campo proveniente do mesmo registro.
- previousValue - o conteúdo do atributo Value é o nome do campo proveniente do registro anterior na mesma consulta e com o mesmo formato de entrada.

Value - especifica o valor ou expressão regular com os quais fazer correspondência.

MatchCase [yes/no] (opcional) - define se o teste deve diferenciar maiúsculas e minúsculas. Quando definido como "yes", os dois valores são convertidos em minúsculas antes do teste.

- **TranslatorSwitch** - determina qual conversor usar para converter a linha de dados em um evento unificado do CA Business Service Insight.
  - **DefaultTranslator** - especifica o conversor a ser usado caso não haja correspondência com nenhum critério. Se o valor for Ignore, nenhum conversor será usado, e a linha é ignorada.
  - **TranslatorCase** - especifica o critério a ser testado em dados processados para determinar para qual conversor ele deve ser roteado.

Break [yes|no]

yes - (padrão) se houver correspondência com os critérios, o próximo critério não é marcado.

no - em todos os casos, após avaliar os critérios e operar o conversor, se houver correspondência, prossegue-se para o próximo critério.

LogicOperator [and/or] (opcional)

- and - especifica que é preciso fazer correspondência com todas as condições. (o padrão).
- or - especifica que ao menos uma condição deve ser atendida.

TranslatorName - especifica o conversor para o qual redirecionar, caso as condições sejam atendidas.

Condition - especifica a condição a ser testada em dados processados para determinar o conversor relevante. Equivale à condição em InputFormatSwitch.

## Seção TranslationTableCollection

Esta seção contém tabelas de mapeamento proveniente dos valores da origem de dados nos campos de evento do CA Business Service Insight.

### Estrutura do XML:

```
<TranslationTableCollection
 LoadingMode="remote"
 TranslationTablesFileName="Translations.xml">
 <TranslationTable
 Name="ResourcesTranslateTable"
 DestinationType="resource">
 <TranslationField>nodeName</TranslationField>
 </TranslationTable>
 <TranslationTable
 Name="EventTypesTranslateTable"
 DestinationType="event_type">
 <TranslationField>eventType</TranslationField>
 </TranslationTable>
 <TranslationTable
 Name="valueUpDownTranslateTable"
 DestinationType="value"
 ValueType="string">
 <TranslationField>eventType</TranslationField>
 </TranslationTable>
</TranslationTableCollection>
```

- TranslationTablesFileName (opcional) - especifica o nome do arquivo em que as tabelas são armazenadas localmente. Caso não seja especificado, usa-se o valor padrão ("Translation.XML").

- LoadingMode [standalone, remote] (opcional)

**Observação:** o padrão para a interface online é remote, e para a interface offline é standalone. O método de carregamento especificado para as tabelas de conversão é o seguinte:

- standalone - o conector carrega as tabelas de conversão localmente. Não há conexão com o servidor do CA Business Service Insight em relação à conversão. As alterações nas tabelas de conversão serão armazenadas apenas no arquivo local.
- remote - o conector envia uma solicitação para carregar todas as tabelas do servidor do CA Business Service Insight. As alterações feitas nas tabelas de conversão também serão armazenadas localmente.
- LoadTimeout (opcional) - quando o modo de carregamento é remoto, é possível especificar aqui o tempo limite em segundos.
  - TranslationTable - vincula o valor do evento à tabela de mapeamento.

- Name - um nome que será usado e referenciado pelo conversor. Para ser válido, o nome deve começar com uma letra ou um sublinhado e conter letras, dígitos e sublinhados.
- DestinationType: [resource, event\_type, contract\_party, service, time\_zone, value].
- ValueType [integer, real, string] (opcional) - especifica o tipo de valor retornado da tabela. O texto e os valores reais são válidos apenas para tabelas com DestinationType="value"
- TranslationField - especifica o nome do campo a partir do qual converter. O nome do campo é obtido dos campos de formato de entrada. É possível ter até cinco campos.



## Seção TranslatorCollection

A seção Translator Collection descreve como converter o registro da origem de dados analisados e manipulados extraído na seção anterior para um evento do CA Business Service Insight.

Quando o modo da interface é "online", o evento do CA Business Service Insight tem uma estrutura unificada que contém os seguintes campos:

- Timestamp - especifica a data e a hora da ocorrência do evento.
- ResourceId - especifica a ID do recurso do CA Business Service Insight associada ao evento (o recurso medido, etc.).
- EventTypeId - especifica o tipo de evento do CA Business Service Insight associado ao evento e descreve o tipo de evento (tipo de medição no recurso, tipo de ação de ticket, etc.).
- DataSourceId (opcional) - especifica o nome da origem de dados de entrada (nome de arquivo, nome de tabela, etc.).
- Value - especifica o valor do evento (resultado da medição, número do tíquete, etc.). Esse campo pode ser exibido mais de uma vez.

Quando o modo da interface é "offline", não há nenhuma limitação quanto ao número de campos ou quanto ao nome.

### Estrutura do XML:

```
<TranslatorCollection>
 <Translator TranslatorName="events" OnDuplication = "ignore">
 <TranslatorFields>
 <TranslatorField Name="ResourceId" SourceType="table"
 SourceName="ResourcesTranslateTable" Iskey="yes"/>
 <TranslatorField Name="EventTypeId" SourceType="constant"
 ConstantValue="1002" Iskey="yes"/>
 <TranslatorField Name="Timestamp" SourceType="field"
 SourceName="timestamp" Iskey="yes"/>
 <TranslatorField Name="Value" SourceType="table"
 SourceName="valueUpDownTranslateTable" Iskey="yes"/>
 <TranslatorField Name="Value" SourceType="field"
 SourceName="nodeName" Iskey="yes"/>
 <TranslatorField Name="Value" SourceType="constant"
 Type="integer" ConstantValue="1000" Iskey="yes"/>
 <TranslatorField Name="Value" SourceType="field"
 SourceName="timestamp" TimeShift="-3600"
 TimeShiftFieldName="createDate" Iskey="yes"/>
 <TranslatorField Name="Value" SourceType="lookup"
 SourceName="ServiceTable" LookupValue="word"
 Iskey="yes"/>
 </TranslatorFields>
 </Translator>
</TranslatorCollection>
```

</TranslatorCollection>

- Translator - descreve como converter o conjunto de campos recebidos no evento de saída.
  - TranslatorName - especifica o nome usado por InputFormat para enviar conjuntos de campos ao conversor.
  - OnDuplication - especifica o integrante com o valor "ignore", "add", "update", ou "updateAlways" para determinar a medida a ser tomada mediante um evento de duplicação (consulte o tópico Singularidade do evento).
  - TranslatorFields - contém a lista dos elementos de TranslatorField. Cada um deles contém os seguintes atributos:
    - Name - especifica o nome do campo. Na interface online, o nome deve ser Timestamp, ResourceId, EventType, Value ou DataSourceId.
    - SourceType:
      - field - o valor deste campo é obtido do campo no formato de entrada. O atributo SourceName contém o nome do campo.
      - table - o valor do campo é obtido da tabela de conversão. O atributo SourceName contém o nome de tabela.
      - lookup - o valor deste campo é obtido da tabela de conversão. O atributo SourceName contém o nome de tabela. O valor a ser convertido é obtido do atributo e não do formato de entrada.
      - constant - o valor do campo é constante e está presente no atributo ConstantValue.
    - SourceName - contém o nome do campo para o nome da tabela de conversão.
    - Type [integer/real/string/time] - é necessário apenas quando o tipo do campo não é predefinido (pelo nome do campo ou por SourceType). Na interface online, é necessário apenas para o campo Value com SourceType=constant. Na interface offline, é necessário apenas para cada campo com SourceType=constant.
    - IsKey - representa a chave exclusiva do evento. Essa chave é coletada dos campos marcados com TranslatorFields?IsKey = "yes".  
(consulte o tópico Singularidade do evento)
    - LookupValue - contém o valor de pesquisa mediante SourceType="lookup".
    - ConstantValue - contém o valor da constante mediante SourceType=constant. Quando o campo é de data e hora, o valor da constante é um texto formatado de acordo com TimeFormat ou "Now" ou "NowUtc", em que "Now" refere-se à data e à hora atuais na localidade de origem dos dados e "NowUtc" refere-se à data e à hora atuais em UTC.
    - TimeFormat - contém o formato de data e hora, necessário apenas aos campos com SourceType=constant e Type=time.

- TimeShift - define o tempo que a mudança leva em segundos. Destina-se apenas a campos de tempo.
- TimeShiftFieldName (opcional) - contém o nome do campo, proveniente do formato de entrada, especificando o tempo levado em segundos. TimeShift e TimeShiftFieldName podem ser usados juntos.



# Apêndice D: Definindo fórmulas de lógica de negócios (especialista em lógica de negócios)

---

Esta seção contém os seguintes tópicos:

[O que evitar ao criar fórmulas de lógica de negócios](#) (na página 341)

[Vigência de métricas agrupadas e recursos](#) (na página 341)

## O que evitar ao criar fórmulas de lógica de negócios

Ao criar fórmulas de lógica de negócios, certifique-se de ter em mente o seguinte:

- Nunca atribua um valor nulo a uma variável global. A atribuição de um valor nulo pode causar falha no PSLWriter ao calcular a métrica. Se um valor não inicializado for necessário, use Vazio.
- Evite o uso de mapa e objetos vetoriais em métricas agrupadas. Se esses objetos tiverem de ser usados, mantenha-os no menor tamanho possível. Métricas agrupadas que contêm grandes mapas e vetores levam muito tempo para calcular.

## Vigência de métricas agrupadas e recursos

### O que é uma métrica agrupada?

A métrica agrupada permite a definição de uma métrica a ser executada para cada integrante de um grupo de recursos, de modo a aplicar a mesma definição e a mesma lógica a um conjunto de itens. O agrupamento pode ser definido de maneira estatística em um conjunto predefinido de recursos, ou de maneira dinâmica nos integrantes de um grupo de recursos. Com o tempo, é possível alterar o grupo e incluir ou excluir seus integrantes.

Um recurso ou grupo de recursos também podem ser excluídos do grupo e, posteriormente, voltar a ser incluídos quantas vezes forem necessárias, no mesmo período de cálculo (dia, mês, ano, etc.).

**O que acontece no Business Logic quando um item é removido do grupo base do agrupamento?**

O método OnPeriodEnd e a função de resultado são acionados para este item do agrupamento. Se isso ocorrer durante o período de cálculo, o resultado é gravado apenas no banco de dados quando o período de cálculo original tiver sido finalizado (por exemplo, no final do mês ou do ano).

**O que acontece no Business Logic quando um item é incluído no grupo base do agrupamento?**

As variáveis globais são iniciadas, e os métodos OnLoad, OnRegistration e OnPeriodStart são acionados para o item do agrupamento.

**O que ocorre na lógica de negócios quando um item de agrupamento é incluído no grupo base do agrupamento após ser removido do grupo no mesmo período de cálculo?**

O resultado definido para o período durante o qual o item do agrupamento fazia parte do grupo é substituído por um novo resultado. Em outras palavras, o resultado no final do período de cálculo refere-se apenas ao último período no período calculado em que o item do agrupamento faz parte do grupo.

**Qual é o efeito do atributo vigente de um recurso no Business Logic?**

Durante o período em que um recurso não está vigente, nenhum dado bruto é coletado para ele.

**Qual é o efeito do atributo vigente de um recurso no agrupamento?**

Alterar um recurso para que ele não seja mais vigente tem o mesmo impacto no agrupamento que excluir um recurso do grupo. O agrupamento se comporta do mesmo modo mediante a vigência do recurso e a associação ao grupo.

**Como aplicar exceções em um recurso? Usar a vigência do recurso é o método correto?**

Há alguns casos de negócios em que um período de exceção deve ser definido para um recurso individual, por exemplo, um servidor pode entrar em manutenção e não ser incluído nos cálculos durante o período de manutenção.

Uma vez que o Business Logic ignora os eventos de dados brutos de um recurso que não esteja vigente, pode-se optar por implementar as exceções em um recurso usando o mecanismo de vigência. Isso funciona em alguns casos. No entanto, se o recurso fizer parte de uma métrica agrupada e alternar entre vigente e não vigente durante o mesmo período de cálculo, apenas o último período durante o qual o recurso esteve vigente é considerado no resultado, conforme declarado acima. Nesse caso, aconselha-se usar o recurso de atributos personalizados. É possível gerenciar um atributo adicional para o recurso declarando seu status. Nesse caso, a fórmula do Business Logic consulta o status do recurso nas partes relevantes do script.





# Glossário

---

**Adaptador**

A interface entre o CA Business Service Insight e origens de dados de terceiros. Os conectores convertem os dados obtidos destas origens de dados em um formato que possa ser usado para os cálculos de nível de serviço fornecidos às partes contratuais.

**Agente**

Um objeto que representa uma métrica em uma determinada unidade de tempo.

**Alerta**

Notificações para os usuários sobre os eventos que estejam ocorrendo no sistema. Os alertas permitem que os usuários executem ações corretivas para evitar desvios do contrato, penalidades incorrentes, e assim por diante.

**Alocação de recursos**

A alocação de um recurso para um serviço direciona o fluxo de eventos do recurso para esse serviço. Um recurso pode ser alocado para um serviço, parte contratual ou grupo.

**Anotações do evento**

Informações adicionais sobre um evento específico. As anotações do eventos são definidas automaticamente ou manualmente (dentro dos relatórios).

**Assistente de métricas**

Uma interface fácil de usar, desenvolvida para que os usuários possam facilmente modificar métricas na primeira vez em que a métrica é adicionada ao contrato a partir de um modelo de nível de serviço.

**Assistente de relatório**

Interface gráfica do usuário (GUI) usada para definir os parâmetros do relatório.

**Autor do contrato**

O usuário que é responsável pela criação de um determinado contrato.

**Booklet**

Arquivo com base em RTF que pode incluir dados de contrato e relatórios relacionados em um formato conveniente, usando modelos configuráveis e pré-definidos

**Categoria do domínio**

Um determinado aspecto do nível de serviço acordado no contrato. Por exemplo, você pode ter um domínio do serviço chamado Central de ajuda e uma categoria do domínio como Tempo de resposta da central de ajuda, que descreve este aspecto específico do serviço da Central de ajuda. O administrador do sistema do fornecedor de serviços geralmente define as categorias do domínio. O CA Business Service Insight também fornece categorias do domínio predefinidas.

---

**Comentário de Root-Cause**

Um comentário definido na lógica de negócios para explicar os resultados do nível de serviço. Os comentários de causa raiz são associados às métricas.

**Comentário manual de causa raiz**

Um comentário definido manualmente pelo visualizador do relatório para explicar ou adicionar mais informações ao resultado de um nível de serviço da métrica específica. Os comentários manuais de causa raiz podem ser compartilhados por todos os visualizadores de relatórios da mesma métrica.

**Conjunto de alterações**

Um conjunto de alterações para o mapa de recursos do fornecedor de serviços.

**Consumo**

Um tipo de métrica que calcula o consumo. Usado principalmente para o módulo financeiro. Ao usar este tipo de métrica, é possível ver os relatórios de consumo e os preços separadamente. O consumo também pode ser calculado na métrica Item de preço.

**Contrato arquivado**

Um contrato cujos dados foram movidos para o arquivo. Os contratos arquivados não são incluídos nos cálculos e não podem ser exibidos nos relatórios.

**Contrato de Apoio**

Contrato de Apoio. Um contrato com um fornecedor externo que abrange a entrega de serviços que apoia a organização em sua entrega de serviços.

**Conversões**

A conversão de dados coletados pelos conectores a partir de suas origens de dados em entidades que foram definidas no CA Business Service Insight.

**Data de entrada do recurso**

A data em que o recurso foi inserido no CA Business Service Insight. Esta não deve ser confundida com a Data de vigência do recurso.

**Data de vigência do recurso**

A data a partir da qual o CA Business Service Insight pode usar o recurso. A data de vigência do recurso é definida pela versão do recurso que inclui o recurso. O recurso não será reconhecido por qualquer versão de recurso que tenha uma data de vigência anterior à versão do recurso na qual o recurso foi criado.

**Declaração de objetivo**

A descrição lógica de uma métrica que contém os parâmetros que definem a métrica. As declarações de objetivo são exibidas na interface gráfica do usuário do CA Business Service Insight para que a essência da métrica seja compreendida de forma eficaz.

**Dispositivo de alerta**

Dispositivos da rede que recebem mensagens de alerta por email, pop-ups ou SMS, ou por meio de um programa de terceiros.

---

**Entrada de conversão**

Representa uma definição de origem e valores de destino que serão utilizados pela tabela de conversão.

**Evento da métrica**

Um evento gerado por uma métrica no CA Business Service Insight.

**Evento de dados**

Eventos gerados pelos conectores do CA Business Service Insight

**Evento de dados brutos**

Um evento gerado pelo dados brutos no CA Business Service Insight.

**Eventos**

Um evento é uma medição executada em um recurso por uma ferramenta de medição de terceiros e, em seguida, convertida pelo conector para um formato usado pelo CA Business Service Insight. Os tipos de evento determinam como esses eventos são definidos e formatado para o CA Business Service Insight.

**Eventos recebidos**

Uma lista de eventos recebidos de outras métricas mostrada na janela do Escopo de lógica de negócios ao clicar na guia Eventos recebidos.

**Exceção**

Período que não conta para fins de cálculo de níveis de serviço. Por exemplo, o tempo de inatividade.

**Função**

Um conjunto de responsabilidades, atividades e autorizações. A função do usuário define o modo de exibição do CA Business Service Insight. A cada usuário do CA Business Service Insight é atribuída uma ou mais funções que determinam quais ações o usuário pode executar no CA Business Service Insight. As ações que não podem ser executadas pela função são removidas da interface com o usuário do CA Business Service Insight quando o usuário acessa o aplicativo.

**Granularidade**

A granularidade determina as unidades de tempo adicionais das quais o autor da métrica quer obter resultados para excluir o período de monitoramento da métrica.

**Grupo de partes contratuais**

Grupo de partes contratuais logicamente definido (clientes). Referenciar um grupo, em vez de uma parte contratual individual, simplifica a criação do contrato. Uma parte contratual pode pertencer a mais de um grupo de partes contratuais.

**Grupo de recursos**

Uma coleção de recursos que são agrupados em uma unidade lógica. Um grupo de recursos pode conter um ou mais recursos individuais ou grupo de recursos.

---

**Índice**

Um tipo de unidade.

**Item de preço**

Uma métrica que calcula os preços. Os preços podem basear-se no consumo ou definidos como preços fixos.

**KPI**

Principal indicador de desempenho. Uma medida significativa usada isoladamente ou em combinação com outros principais indicadores de desempenho para monitorar quão bem um negócio ou serviço está alcançando seus objetivos quantificáveis.

**KQI**

Indicador de qualidade chave. Uma medida significativa usada isoladamente ou em combinação com outros principais indicadores de desempenho para monitorar quão bem um negócio ou serviço está alcançando seus objetivos qualitativos.

**Mapa do painel**

O layout do painel. A organização de medidores no painel.

**Mecanismo do status atual**

Um processo autônomo que calcula o status atual das métricas. Não há um limite para o número de instâncias autorizadas a serem executadas em um ou mais computadores.

**Métrica**

Uma combinação de vários parâmetros que definem o objetivo de nível de serviço para um determinado serviço em um determinado momento. Cada métrica é vinculada a um único domínio do serviço. Os campos e atributos da métrica incluem: categoria do domínio, fórmula de nível de serviço, serviço, período de atividade, objetivo de nível de serviço e período de monitoramento.

**Métrica agrupada**

Um tipo de métrica que se aplica a vários recursos ou grupos de recursos.

**Métrica de consumo**

Uma métrica que fornece a capacidade de exibir consumo e preços separadamente em um relatório.

**Métrica derivada**

Uma métrica que foi criada a partir de um modelo de contrato ou modelo de nível de serviço.

**Métrica informativa**

Uma métrica que executa o cálculo informativo a ser usado somente para fins de geração de relatórios.

**Métrica provisória**

Uma métrica que pode gerar eventos para fins de cálculo de eventos. As métricas provisórias não podem ser calculadas.

---

**Métrica relacionada**

Uma métrica que é referenciada como um objetivo em uma relação.

**Modelo de contrato**

Contrato predefinido usado para criar um novo contrato.

**Modelo de lógica de negócios**

Fórmulas predefinidas de lógica de negócios que podem ser usadas para definir novas fórmulas de lógica de negócios.

**Modelo de nível de serviço**

Uma definição de serviço é um conjunto predefinido de serviços e métricas, usado para facilitar a criação e a modificação de contratos de acordo com os processos de negócios comuns.

**Módulos de lógica de negócios**

Biblioteca de funções de script que pode ser usada na lógica de negócios.

**OLA**

Contrato de Nível Operacional. Um OLA é um SLA no qual o fornecedor é a organização interna e os clientes são as unidades de negócios internas.

**Opções regionais**

Detalhes específicos da localidade da parte contratual. Os parâmetros incluem idioma, moeda, diferença de horário de GMT, horário de verão, formato de data e moeda.

**Pacote**

Uma coleção de modelos de nível de serviço e modelos compactados para serem instalados e descompactados em outra instância do CA Business Service Insight.

**Painel**

Uma interface com o usuário destinada aos gerentes de alto nível que organiza e apresenta informações em tempo real e relatórios em uma maneira fácil de ler.

**Parâmetro**

Um valor que pode ser definido fora da lógica de negócios para afetar a fórmula de lógica de negócios. Os parâmetros estão sendo usados pela lógica de negócios para determinar o resultado do nível de serviço. Um parâmetro pode ser do tipo texto, lista, número, data ou tabela. Exemplos de parâmetros são limites e nomes de recursos.

**Parâmetro em nível de contrato**

Um parâmetro de um contrato, modelo de contrato e modelo de nível de serviço regular que está sendo usado pelas métricas na lógica de negócios.

**Parâmetros em nível de métrica**

Parâmetros de uma métrica.

---

**Parte contratual**

O cliente ou o fornecedor com o qual o contrato está assinado. Uma parte contratual também pode representar uma entidade interna dentro de uma organização maior.

**Pasta de relatórios**

A pasta de relatórios é um contêiner usado para agrupar os relatórios relacionados entre si de um modo ou de outro.

**Penalidade**

A compensação devida à parte contratual se o objetivo do nível de serviço não for alcançado dentro de um período de monitoramento definido. As penalidades podem ser ativadas ou desativadas no CA Business Service Insight.

**Perfil de alerta**

Um conjunto de parâmetros que definem as condições sob as quais uma mensagem de alerta é disparada, os usuários para quem o alerta é emitido e como ele é enviado.

**Período de monitoramento**

O intervalo de tempo durante o qual o CA Business Service Insight mede o nível de serviço fornecido em comparação ao objetivo de nível de serviço definido no contrato. A medição obtida durante este período determina se há um desvio do objetivo definido (conforme definido pela lógica de negócios) e se qualquer penalidade ou bônus incorreu deste desvio. O período de monitoramento também determina como os relatórios de negócios são gerados.

**Recurso**

Um elemento único da infraestrutura total do fornecedor de serviços. Exemplos de recursos incluem os servidores individuais, switches, hubs, roteadores, uma central de ajuda ou qualquer outro elemento mensurável. Vários recursos podem ser definidos como parte de um grupo de recursos, que por sua vez, será tratado como outro recurso.

**Relação**

Uma entidade que descreve uma conexão de um determinado tipo estabelecida entre duas métricas e com algumas propriedades.

**Relatório composto**

Vários relatórios regulares que são exibidos como várias séries em um único gráfico

**Relatório de formato livre**

Um relatório que é criado por uma instrução SQL definida pelo usuário com base no banco de dados do CA Business Service Insight ou em outros bancos de dados externos

**Relatório de grupo**

Um relatório que coloca vários relatórios lado a lado em uma página.

**Relatório de grupo de relatórios**

Um relatório que encapsula vários relatórios regulares, colocados lado a lado

---

**Relatório simples**

Um meio de analisar os resultados calculados pelo CA Business Service Insight com base em um rico conjunto de critérios.

**Script de conversão**

Um script que pode ser usado para facilitar a manutenção das conversões e evitar erros.

**Service Catalog**

Uma coleção de todas as informações sobre serviços, domínios e unidades, bibliotecas de modelos e modelos de nível de serviço no CA Business Service Insight.

**SLO**

Objetivo do nível de serviço. Usado para medir os acordos contratuais.

**Tabela de conversão**

Um mecanismo usado para converter os valores de dados que vêm de dados brutos nos dados definidos no sistema.

**Tipo de métrica**

Descreve a razão para cálculo de uma determinada métrica e define a lista de campos e seus comportamentos em termos de campos obrigatórios e os padrões que uma métrica de um determinado tipo deve ter.

**Tipo de recurso**

Uma categoria de recursos integrados. O recurso deve ser alocado a pelo menos um tipo de recurso.

**Trilha de auditoria**

Um registro cronológico de atividades do sistema e do usuário do CA Business Service Insight que são salvas no sistema. A auditoria pode ser revisada posteriormente para identificar as ações dos usuários no sistema ou processos que ocorreram no sistema.

**Trilha de auditoria do contrato**

Um log de todas as atividades de um contrato.

**Unidade**

Um catálogo de unidades mensuráveis. Exemplos incluem porcentagem e moeda.

**Unidade de medida**

Uma unidade de medida para todas as métricas definidas para uma categoria do domínio.