CA Business Service Insight

Guía de implementación

8.2

Esta documentación, que incluye sistemas incrustados de ayuda y materiales distribuidos por medios electrónicos (en adelante, referidos como la "Documentación") se proporciona con el único propósito de informar al usuario final, pudiendo CA proceder a su modificación o retirada en cualquier momento.

Queda prohibida la copia, transferencia, reproducción, divulgación, modificación o duplicado de la totalidad o parte de esta Documentación sin el consentimiento previo y por escrito de CA. Esta Documentación es información confidencial, propiedad de CA, y no puede ser divulgada por Vd. ni puede ser utilizada para ningún otro propósito distinto, a menos que haya sido autorizado en virtud de (i) un acuerdo suscrito aparte entre Vd. y CA que rija su uso del software de CA al que se refiere la Documentación; o (ii) un acuerdo de confidencialidad suscrito aparte entre Vd. y CA.

No obstante lo anterior, si dispone de licencias de los productos informáticos a los que se hace referencia en la Documentación, Vd. puede imprimir, o procurar de alguna otra forma, un número razonable de copias de la Documentación, que serán exclusivamente para uso interno de Vd. y de sus empleados, y cuyo uso deberá guardar relación con dichos productos. En cualquier caso, en dichas copias deberán figurar los avisos e inscripciones relativas a los derechos de autor de CA.

Este derecho a realizar copias de la Documentación sólo tendrá validez durante el período en que la licencia aplicable para el software en cuestión esté en vigor. En caso de terminarse la licencia por cualquier razón, Vd. es el responsable de certificar por escrito a CA que todas las copias, totales o parciales, de la Documentación, han sido devueltas a CA o, en su caso, destruidas.

EN LA MEDIDA EN QUE LA LEY APLICABLE LO PERMITA, CA PROPORCIONA ESTA DOCUMENTACIÓN "TAL CUAL" SIN GARANTÍA DE NINGÚN TIPO INCLUIDAS, ENTRE OTRAS PERO SIN LIMITARSE A ELLAS, LAS GARANTÍAS IMPLÍCITAS DE COMERCIALIZACIÓN, ADECUACIÓN A UN FIN CONCRETO Y NO INCUMPLIMIENTO. CA NO RESPONDERÁ EN NINGÚN CASO, ANTE VD. NI ANTE TERCEROS, EN LOS SUPUESTOS DE DEMANDAS POR PÉRDIDAS O DAÑOS, DIRECTOS O INDIRECTOS, QUE SE DERIVEN DEL USO DE ESTA DOCUMENTACIÓN INCLUYENDO A TÍTULO ENUNCIATIVO PERO SIN LIMITARSE A ELLO, LA PÉRDIDA DE BENEFICIOS Y DE INVERSIONES, LA INTERRUPCIÓN DE LA ACTIVIDAD EMPRESARIAL, LA PÉRDIDA DEL FONDO DE COMERCIO O LA PÉRDIDA DE DATOS, INCLUSO CUANDO CA HUBIERA PODIDO SER ADVERTIDA CON ANTELACIÓN Y EXPRESAMENTE DE LA POSIBILIDAD DE DICHAS PÉRDIDAS O DAÑOS.

El uso de cualquier producto informático al que se haga referencia en la Documentación se regirá por el acuerdo de licencia aplicable. Los términos de este aviso no modifican, en modo alguno, dicho acuerdo de licencia.

CA es el fabricante de esta Documentación.

Esta Documentación presenta "Derechos Restringidos". El uso, la duplicación o la divulgación por parte del gobierno de los Estados Unidos está sujeta a las restricciones establecidas en las secciones 12.212, 52.227-14 y 52.227-19(c)(1) - (2) de FAR y en la sección 252.227-7014(b)(3) de DFARS, según corresponda, o en posteriores.

Copyright © 2012 CA. Todos los derechos reservados. Todas las marcas registradas y nombres comerciales, logotipos y marcas de servicios a los que se hace referencia en este documento pertenecen a sus respectivas compañías.

Información de contacto del servicio de Soporte técnico

Para obtener soporte técnico en línea, una lista completa de direcciones y el horario de servicio principal, acceda a la sección de Soporte técnico en la dirección http://www.ca.com/worldwide.

Contenido

Capítulo 1: Introducción 9

Roles ... 9
- Gestor del catálogo de servicios/gestor de la detección de servicios ... 10
- Gestor de contratos ... 10
- Experto en la lógica de negocios ... 11
- Experto de la fuente de datos ... 12
- Administrador del sistema ... 13

Proceso de solución ... 14
- Planificación ... 15
- Diseño ... 16
- Implementación ... 18
- Instalación e implementación ... 18

Capítulo 2: Planificación y diseño 19

Sesión de recopilación de requisitos (Todos los roles) ... 20
Recolección de datos de muestra (experto en el origen de datos) ... 22
Diseño ... 23
- Introducción al diseño (gestor de contratos, experto en la lógica de negocios y experto en el origen de datos) ... 24
- Modelado del contrato (gestor de contratos) ... 25
- Resultados de la etapa de modelado del contrato (gestor de contratos y experto en el origen de datos) ... 48
- Modelado de datos (experto en el origen de datos, experto en la lógica de negocios) ... 50
- Resultados de la etapa de modelado de datos (experto en el origen de datos y experto en la lógica de negocios) ... 67

Capítulo 3: Implementación 69

Implementación: introducción ... 69
Configuración del marco (gestor de contratos) ... 72
Configuración de la biblioteca de plantillas (gestor de contratos) ... 73
Cómo crear contratos (gestor de contratos) ... 74
- Creación de contratos a partir de un servicio ... 76
- Creación de plantillas de nivel de servicio ... 78
- Ciclo de vida del contrato y control de versiones ... 79

Recolección de datos (experto en el origen de datos) 82
Funcionalidad del adaptador 82
Entorno del adaptador 83
Archivos principales 87
Archivos de trabajo 88
Comunicación del adaptador 94
Valores de configuración del registro del adaptador 97
Modos de ejecución del adaptador 100
Configuración y herramientas de mantenimiento 102
Cómo configurar adaptadores y traducciones 103
Traducciones automáticas con scripts de traducción 144
Temas avanzados de la función de adaptador 145
Scripting de lógica de negocios (experto en la lógica de negocios) 153
Flujo de trabajo del scripting de lógica de negocios 155
Módulos de lógica de negocios 156
Dentro de la lógica de negocios 157
Cómo activar los contratos (gestor de contratos) 197
Recálculo completo de la métrica de contrato 200
Creación de entregas (gestor de contratos) 201
Definición de opciones de seguridad (administrador) 201
Cómo crear informes 202
Configuración de páginas de cuadro de mandos 215
Adición de perfiles de alertas de nivel de servicio 218

Capítulo 4: Instalación e implementación 223

Introducción 224
Preparativos 226
Instalación 228
Cómo importar o exportar entre entornos (experto en el origen de datos) 230
Integración: configuración del servidor de correo (administrador del sistema) 232
Establecimiento de las preferencias del sistema (administrador del sistema) 233
Opciones de seguridad (administrador del sistema) 234
Especificación de valores de configuración para sincronización de SSA 235
Valores de configuración del uso compartido global 236

Capítulo 5: Gestión y detección del servicio 239

Detección del servicio 239
Cómo funciona Detección y gestión de servicios 239

Comprensión de las categorías de servicios 240
Hoja de cálculo de categorías de servicio 241
Establecer la visibilidad y el ámbito de sus servicios 244
Establecimiento de vistas de servicios 245
Establecimiento de opciones de visualización de columnas 246
Selección de una acción para un servicio seleccionado 248
Trabajar con opciones de atributos y metadatos 250
Gestión de los servicios 254
Uso compartido de datos de comparación en Cloud Commons 256
Adición de su servicio a Cloud Commons 257
Adición de servicios en la página Detección y gestión de servicios 258
Exportación e importación de datos de servicio manualmente 260

Apéndice A: Dominios del servicio de muestra y categorías de dominio 265

Apéndice B: Ejemplos de casos prácticos 267

Ejemplos de modelado del contrato 267
Caso práctico 1: Disponibilidad del sistema 268
Caso práctico 2: Disponibilidad del sistema 2 269
Caso práctico 3: Tiempo de respuesta del sistema 271
Caso práctico 4: Rendimiento del servicio de asistencia 276
Caso práctico 5: Copia de seguridad del sistema 278
Ejemplo de modelado de métrica financiera 279
Caso práctico 6: Modelado de las condiciones financieras de modelado de un contrato/servicio 279
Ejemplos de modelado de datos 287
Caso práctico 7: Rendimiento del servidor 287
Caso práctico 8: Rendimiento del servicio de asistencia 291
Uso del ejemplo de atributos personalizados 298
Caso práctico 9: Varios estilos dinámicos 298
Ejemplos de scripts de traducción 302
Caso práctico 10: Traducción automática básica 302
Caso práctico 11: Actualizaciones del modelo de recurso 305
Ejemplos de scripting de lógica de negocios 308
Caso práctico 12: Uso de la lógica de contador para calcular el número de errores 309
Caso práctico 13: Tratamiento de un grupo de componente dinámico 313
Caso práctico 14: Tratamiento del reloj de acumulación de tiempo 318
Ejemplos de escritura de lógica de negocios vigente 324

Caso práctico 15: Métricas en clúster 327
Caso práctico 16: Patrones de diseño de lógica de negocios 331
Caso práctico 17: Funcionalidad incorporada 336
Caso práctico 18: Registro 338
Caso práctico 19: Asistente del adaptador para origen de datos basado en archivo 341
Creación del adaptador 343
Implementación del adaptador 354
Programación del adaptador 356
Caso práctico 21: Ejemplo de integración de LDAP 359
Caso práctico 22: Generación de informes mediante PERL 366

Apéndice C: Especificaciones de configuración del adaptador 369

Estructura global 369
Sección General 370
Sección Interface de CA Business Service Insight 376
Sección DataSourceInterface 379
Sección de la interfaz de SQL 382
Sección InputFormatCollection 387
Sección TranslationTableCollection 392
Sección TranslatorCollection 394

Apéndice D: Definición de fórmulas de lógica de negocios (Experto en la lógica de negocios) 397

Qué evitar al crear fórmulas de lógica de negocios 397
Métricas en clúster y efectividad de recursos 397

Glosario 401

Capítulo 1: Introducción

En esta guía se explican las etapas del proceso de implementación y los roles, responsabilidades e interacciones relacionados que están implicados en la planificación, el diseño, la instalación y la implementación de CA Business Service Insight.

Esta sección contiene los siguientes temas:

Roles (en la página 9)
Proceso de solución (en la página 14)

Roles

En este documento, rol hace referencia a una persona que realiza un conjunto común de tareas que se realizan durante una implementación típica. La misma palabra también puede hacer referencia al conjunto de dichas tareas. CA Business Service Insight dispone de un número de roles predeterminados que el administrador del sistema puede editar. Además, se pueden crear nuevos roles y permisos específicos.

Los siguientes roles son necesarios en el proceso de implementación:

- Gestor del catálogo de servicios/gestor de la detección de servicios
- Gestor de contratos
- Experto en la lógica de negocios
- Experto de la fuente de datos
- Administrador del sistema

A continuación se describen las responsabilidades y la cualificación de cada rol.

Nota: La misma persona puede realizar varios roles, tal y como los defina el administrador del sistema.

Gestor del catálogo de servicios/gestor de la detección de servicios

Responsabilidades:

- Gestión del catálogo de servicios de la organización.
- Definición del catálogo de oferta de servicios de la organización.
- Descubrimiento y clasificación de los servicios mediante las funciones de detección de servicios.
- Gestión de los requisitos para las definiciones de contrato y la generación de informes.

Cualificación obligatoria:

- Conocimiento del proceso de entrega de servicios de E2E de la organización.
- Conocimiento de los conceptos de CA Business Service Insight.

Gestor de contratos

Responsabilidades:

- Interacción con el usuario final.
- Estar a cargo de la creación del catálogo de servicios basado en requisitos definidos.
- Creación de nuevos contratos y manteniendo de los existentes.
- (Opcional, recomendado) Estar a cargo de la implementación de acuerdos de nivel de servicio de clientes específicos.

Cualificación:

- Comprensión de los principios y la lógica de las métricas.
- Usuario experto en las funciones de CA Business Service Insight basadas en la interfaz gráfica de usuario.

Experto en la lógica de negocios

Responsabilidades:

- Implementación de métricas.
- Escritura de scripts de lógica de negocios; mantenimiento de scripts de lógica de negocios existente.

Cualificación:

- Experiencia de desarrollo básico, sólido conocimiento práctico de idiomas de generación de scripts, como VB-Script.
- Buena comprensión del flujo de datos de CA Business Service Insight.
- Experto en scripting de lógica de negocios de CA Business Service Insight.
- Buena comprensión de la arquitectura y los componentes de CA Business Service Insight.
- Usuario experto en las funciones de CA Business Service Insight basadas en la interfaz gráfica de usuario.

Experto de la fuente de datos

Responsabilidades:

- Diseño de la infraestructura de CA Business Service Insight.
- Creación de adaptadores nuevos; mantenimiento de los adaptadores existentes, interfaz con la infraestructura operacional para obtener mediciones.
- Elaboración y mantenimiento de la infraestructura de CA Business Service Insight.

Cualificación:

- Comprensión del entorno y la estructura de los orígenes de datos del cliente.
- Buen conocimiento práctico de las bases de datos, XML y los idiomas de generación de scripts.
- Comprensión del flujo de datos de CA Business Service Insight y de los procesos de recopilación de datos.
- Experto en adaptadores de CA Business Service Insight.
- Compresión de la arquitectura y los componentes de CA Business Service Insight.
- Usuario experto en las funciones de CA Business Service Insight basadas en la interfaz gráfica de usuario.
- Experiencia en desarrollo (ventaja).

Administrador del sistema

Responsabilidades:

- Gestión de la instalación y las actualizaciones.
- Cumplimiento de los requisitos del sistema de hardware y software.
- Gestión de la configuración de seguridad: definiciones de usuario y de roles.
- Control y gestión del sistema.

Cualificación:

- Comprensión de las infraestructuras y redes del cliente.
- Compresión de la arquitectura y los componentes de CA Business Service Insight.
- Conocimiento práctico de DB, XML e idiomas de generación de scripts.
- Comprensión del flujo de datos de CA Business Service Insight.
- Usuario de funciones de CA Business Service Insight basadas en la interfaz gráfica de usuario.

Proceso de solución

El proceso de solución incluye tres etapas básicas:

- Planificación y diseño
- Implementación
- Instalación e implementación

Los roles descritos previamente participan en etapas diferentes del proceso de implementación, cada uno según sus especialidades. Estos roles interactúan para completar el proceso completo de principio a fin, desde la definición contractual a informe de salidas.

La estructura de esta guía se centra en los roles y los procesos y se basa en el flujo de etapas general que construye el proceso de implementación. Describe la implicación de cada rol y cómo interactúan los roles.

Los párrafos siguientes proporcionan una descripción general de las etapas incluidas en el proceso de implementación y las actividades de cada rol.

El resto de la guía se estructura por etapas con una indicación qué roles participan en cada etapa. Cada capítulo indica el rol responsable de cada actividad.

A continuación se incluye una descripción breve de las tareas básicas de cada fase y las funciones de los diferentes roles. Podrá encontrar más detalles dentro de cada capítulo.

Planificación

La finalidad de la etapa de planificación es identificar y cuantificar todos los aspectos que se deberán abordar como parte de la implementación.

En esta etapa, el gestor de contratos recopila requisitos de negocio del gestor del catálogo de servicios para entender las expectativas de CA Business Service Insight tras una implementación correcta. En esta etapa es importante entender tanto los requisitos actuales como los planes futuros para poder verificar que la implementación permite evolucionar y crecer de manera fácil y fluida.

Según los requisitos de implementación definidos, el gestor de contratos deberá examinar el proceso relacionado (si existe) revisando las entradas y las salidas del proceso existentes. En esta etapa es necesario identificar todos los orígenes de información necesarios y obtener muestras, contratos, informes de salida y orígenes de entrada utilizados para calcular las salidas existentes. Esta información se deberá especificar cuidadosamente para que el gestor de contratos identifique todas las entradas necesarias, de modo que se pueda derivar la salida esperada.

En esta etapa, el gestor de contratos examina los contratos para asegurarse de que el catálogo de servicios y las plantillas proporcionadas como parte de la implementación son compatibles con las necesidades existentes y futuras, y que la implementación actual cubre todas las áreas del contrato.

El gestor de contratos examina los informes y sus formatos para verificar que se realizan todas las mediciones definidas para poder producirlos con la implementación existente.

A continuación, el experto en el origen de datos examina muestras de datos de entrada según las mediciones necesarias y los cálculos que ha proporcionado el gestor de contratos. Esto permite al experto en el origen de datos identificar posibles problemas con las entradas que haya que solucionar, como formatos personalizados o carencias, y determinar si se requieren orígenes de datos adicionales.

La finalidad de este examen es garantizar que los orígenes contienen la información obligatoria para calcular la información inicial y de métricas obligatoria en el proceso de extracción, como los métodos de comunicación con los orígenes de datos y la estructura del origen de datos.

Diseño

Durante la fase de diseño, los requisitos recopilados de entradas y salidas se traducen a la estructura de modelo de CA Business Service Insight, con contratos, métricas y recursos. Esto implica transformar datos reales y conceptualizarlos para ajustarlos al marco de CA Business Service Insight.

El diseño del sistema incluye lo siguiente:

Modelado del contrato

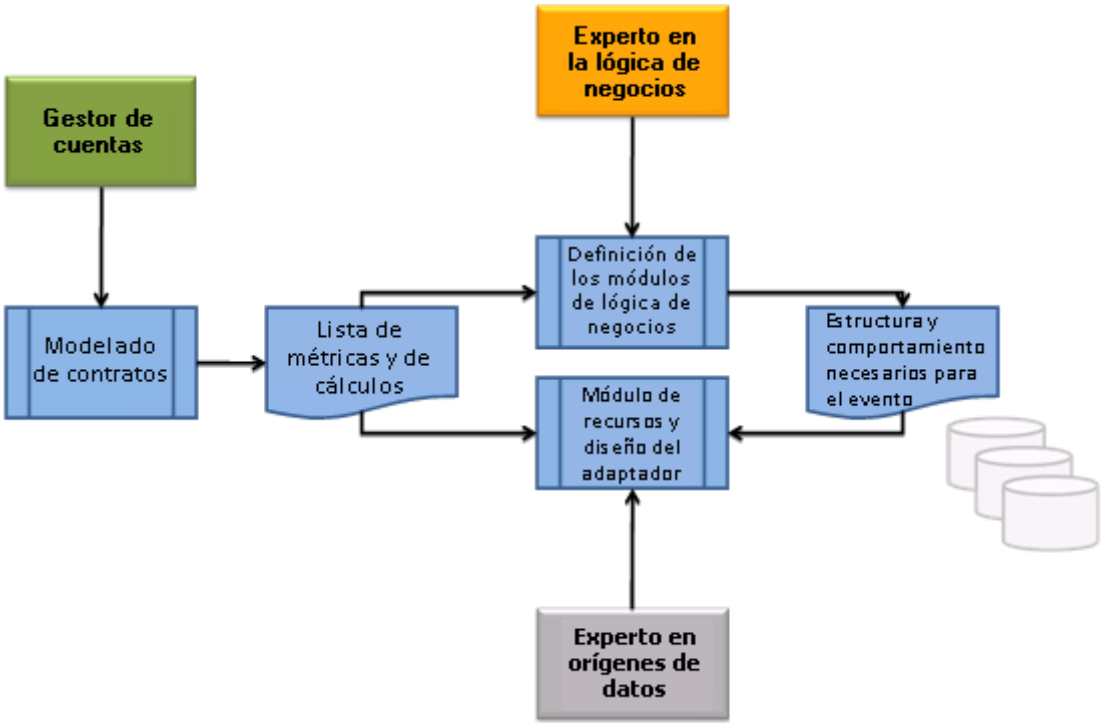
Los acuerdos de nivel de servicio de cliente se interpretan como contratos de CA Business Service Insight y se definen entidades de catálogo de servicios (como las carpetas de plantillas). Esto lo realiza principalmente el gestor de contratos.

Modelado de datos

Los datos de recurso se examinan y modelan en el modelo de recurso de CA Business Service Insight. Esto lo realiza el experto en el origen de datos y el experto en la lógica de negocios.

Puede haber varios métodos disponibles para modelado de contratos o de datos. Sin embargo, existe a menudo una práctica óptima que no solamente resuelve el problema, sino que también brinda más flexibilidad o productividad, proporcionando así un sólido marco para un crecimiento futuro.

Para ayudar al experto en el origen de datos a elegir los métodos más apropiados, se utilizan casos prácticos junto con sugerencias de soluciones.



Como muestra el diagrama de flujo de trabajo anterior, como resultado del proceso de modelado del contrato, el gestor de contratos proporciona como salida la lista de métricas que se deberán configurar en el sistema y la definición de su perfil de cálculo.

Ejemplo:

Cliente A, tiempo de respuesta promedio de sistema CNP.

La lista de métricas se proporciona como entrada al experto en la lógica de negocios, quien define la estructura de evento necesaria y el comportamiento del evento, lo cual permitirá definir el script de cálculo necesario.

La lista de métricas y los requisitos de comportamiento y estructura de evento son la entrada para el diseño del modelo de recurso y los adaptadores de datos por parte del experto en el origen de datos.

Implementación

En la fase de implementación, el sistema de CA Business Service Insight se configura de acuerdo con los resultados de la fase de diseño. La fase de implementación conlleva la toma de los resultados de la etapa de diseño teórico y su uso para construir los requisitos de funcionamiento para CA Business Service Insight.

Algunos de los elementos que se deben crear o tratar incluyen:

Gestor de contratos

- Configuración y detección del servicio
- Creación del contrato
- Configuración de informes y cuadros de mandos

Experto en el origen de datos

- Configuración del adaptador
- Configuración de la infraestructura

Experto en la lógica de negocios

- Módulos de lógica de negocios

Instalación e implementación

La fase de instalación e implementación abarca la instalación e integración del sistema de producción, realización de pruebas, control del rendimiento y formación de usuarios. El administrador del sistema y el experto en el origen de datos llevan a cabo la mayor parte de las actividades en esta etapa.

Capítulo 2: Planificación y diseño

Esta sección contiene los siguientes temas:

Sesión de recopilación de requisitos (Todos los roles) (en la página 20)
Recolección de datos de muestra (experto en el origen de datos) (en la página 22)
Diseño (en la página 23)

Sesión de recopilación de requisitos (Todos los roles)

La sesión de recopilación de requisitos es el primer esencial en la implementación de CA Business Service Insight y debería incluir a todo el personal clave involucrado en la implementación.

La gente que está muy familiarizada con los acuerdos de nivel de servicio seleccionados deberá proporcionar la información obligatoria para esta sesión, incluidos la lógica de negocios del acuerdo de nivel de servicio, cómo se están gestionando actualmente el acuerdo de nivel de servicio, los informes que se están generando actualmente a partir de ellos y todos los requisitos de generación de informes que se esperan en el futuro.

Deberá participar el personal siguiente:

- Gestores del catálogo de servicios
- Gestores de contratos
- Expertos en el origen de datos
- Expertos en lógica de negocios/generación de scripts
- Algunos otros miembros del personal relevantes familiarizados con los acuerdos de nivel de servicio seleccionados
- Gestor de proyectos (si se nombra)

CA recomienda lo siguiente:

- Es importante asegurar que hay personal que puede tomar decisiones en cuanto a algunas definiciones confusas que podrían surgir mientras se revisan los acuerdos de nivel de servicio seleccionados.
- Es aconsejable que el equipo de implementación reciba, antes de esta reunión, una copia los acuerdos de nivel de servicio seleccionados para este proyecto. Los asistentes deberían estar familiarizados con todos los orígenes de datos relevantes que conciernen a los acuerdos de nivel de servicio seleccionados, ser capaces de proporcionar volcados de datos de ellos y explicar sus estructuras interiores, cuando sea aplicable.

Los objetivos principales de la sesión de planificación son definir y especificar:

- El alcance del trabajo planificado.
- Los criterios de éxito.
- Los roles y las responsabilidades de todas las partes implicadas.
- El proceso de implementación y programación.

- Los requisitos de hardware y software obligatorios.

Esto se logra mediante:

- La revisión de los acuerdos de nivel de servicio que se usarán en la implementación.
- La revisión de la "lógica de negocios" para cada objetivo dentro de los acuerdos de nivel de servicio seleccionados.
- La revisión de informes, alertas y requisitos del cuadro de mandos relacionados con los acuerdos de nivel de servicio seleccionados.
- La revisión de los orígenes de datos relevantes que se usarán en la implementación.
- La revisión de la topología de infraestructura relevante.
- La recolección de volcados de datos relevantes a partir de orígenes de datos relevantes.
- La identificando de personas clave y de sus responsabilidades.
- La definición de rutas de comunicación para la duración de la implementación, las reuniones de revisión, el proceso de dudas y respuestas, etc.
- La revisión del proceso de implementación y la programación.
- La revisión del estado actual de los asuntos, es decir, cómo se gestionan estos acuerdos de nivel de servicio y qué es lo que falta.
- La definición de los criterios de éxito de la implementación.
- La definición de los requisitos de hardware y software obligatorios.
- La definición de la línea de hora para la implementación.

Recolección de datos de muestra (experto en el origen de datos)

Para realizar la configuración inicial de los adaptadores fuera del sitio, es importante obtener muestras de datos pasados para poder recopilar datos. Estas muestras deben tener la misma estructura que la de los datos que más tarde aparecerán como entrantes desde el sistema real a partir del cual CA Business Service Insight tiene que recuperar datos. Además de las muestras, el experto en el origen de datos deberá tener información acerca de los orígenes de datos a partir de sus propietarios, para poder así aclarar las cuestiones siguientes:

Tipo:

Base de datos, archivo, flujo u otro.

Ubicación:

¿Dónde está? ¿Cómo llegar allí (detalles de conexión)? ¿Hay obstáculos de seguridad? ¿Plataforma?

Convención de denominación:

¿Cómo se da nombre a los archivos? (¿Es un origen de datos basado en archivo?) ¿Cómo se da nombre a las tablas? ¿Nombres de campos de tabla?

Disponibilidad:

¿Cuándo se puede acceder a él? ¿Cuándo se debería acceder a él? (¿Existen consideraciones de carga?) ¿Actualización continua o una vez cada x tiempo? ¿Durante cuánto tiempo es persistente?

Comportamiento:

¿Se acumulan los datos? ¿Se sobrescriben los datos? ¿Se guarda un archivo de archivado?

Ámbito:

¿Cuántos datos históricos hay disponibles?

Volúmenes:

¿Volumen actual de datos? ¿Tasa de crecimiento? ¿Pronósticos?

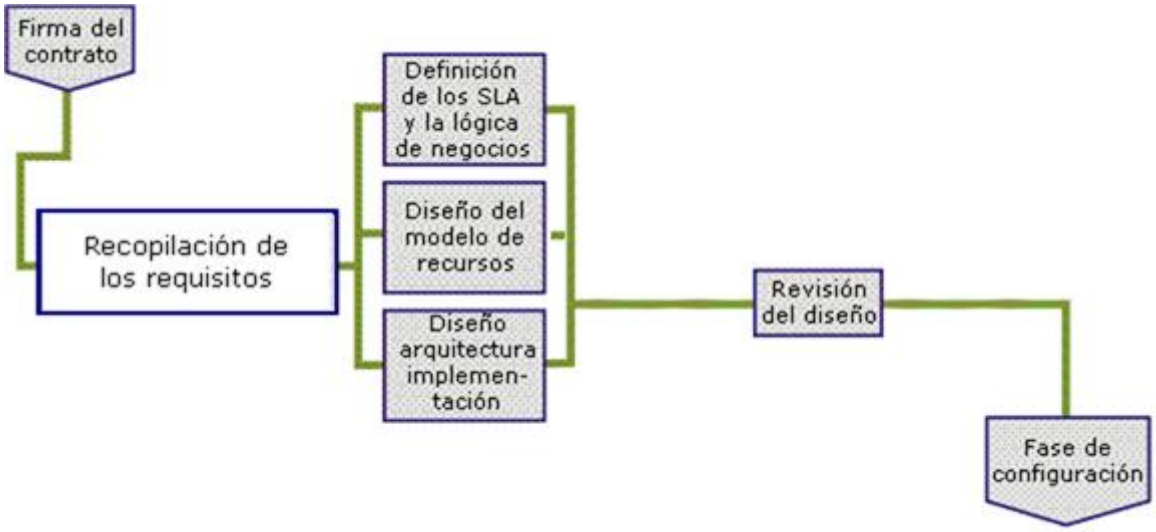
Estructura y formato:

¿Cómo están organizados los datos dentro del origen de datos? ¿Cuáles son los campos de datos? ¿Cuáles son los nombres de tabla? ¿Qué separa los campos de datos?

Streaming:

¿El adaptador obtiene los datos o los recopila?

Diseño



Introducción al diseño (gestor de contratos, experto en la lógica de negocios y experto en el origen de datos)

Esta sección explica el proceso y razonamiento detrás de la fase de diseño del proceso de solución. La fase de diseño sigue a la fase de planificación, y después le sigue la fase de implementación, en el capítulo siguiente.

El objetivo de la fase de diseño es que el equipo de implementación pueda asignar los contratos reales, sus cláusulas y los datos de rendimiento existentes al sistema de CA Business Service Insight.

Antes de empezar este proceso, el equipo de implementación tiene que ser consciente de los diversos métodos y opciones disponibles para garantizar que no sólo se tienen en cuenta todos los requisitos, sino que también el diseño resultante sea lo más óptimo posible, al tiempo que debe permitir cambiarse o crecer en el futuro.

El proceso de diseño conlleva la realización de los pasos siguientes por parte del equipo de implementación:

- Examen de los contratos y conversión de los mismos a los objetos de CA Business Service Insight necesarios, a los que se hace referencia en esta guía como "modelado del contrato". Esto es responsabilidad del gestor de contratos.
- Toma de los conjuntos de datos existentes y determinación de qué elementos relevantes extraer y cómo hacerlo, de acuerdo con los resultados deseados, a lo cual se hace referencia en esta guía como "modelado de datos". Esto es responsabilidad del experto en el origen de datos y el experto en la lógica de negocios.

La sección Modelado del contrato explica:

- La terminología utilizada (parte esencial para una implementación correcta).
- Módulos y plantillas de lógica de negocios.
- Plantilla de nivel de servicio y cómo utilizarla.
- La importancia de crear un catálogo de servicios sólido.
- Métricas de gestión financiera (sanciones, incentivos y costes y ejemplos) según se apliquen a los contratos de cliente y a otras métricas contractuales.

La sección Modelado de datos explica:

- Los eventos según se aplican a CA Business Service Insight y su flujo a través del sistema de CA Business Service Insight.

- Las métricas y cómo se registran.
- Los recursos de CA Business Service Insight y cómo identificarlos.
- Sugerencias sobre cómo automatizar la recolección y definición de estos recursos.

Todos los puntos anteriores se explican en mayor profundidad en las secciones siguientes.

Es importante ser consciente de que una elección pobre en esta etapa puede afectar negativamente al funcionamiento de CA Business Service Insight y puede hacer difícil o imposible retroceder en una etapa posterior.

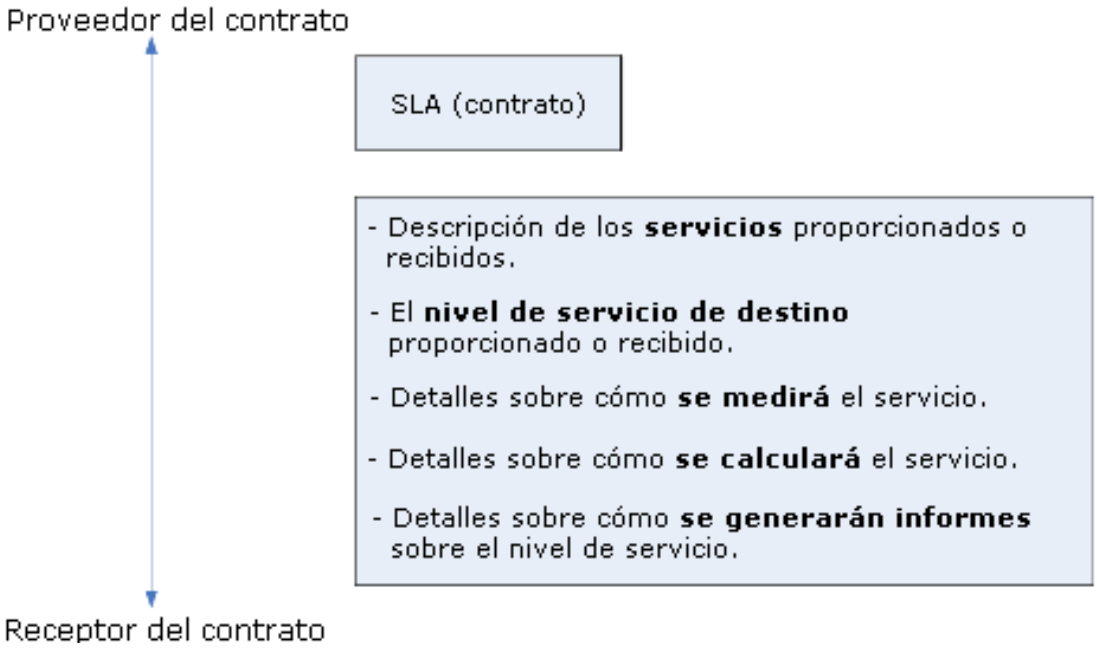
Modelado del contrato (gestor de contratos)

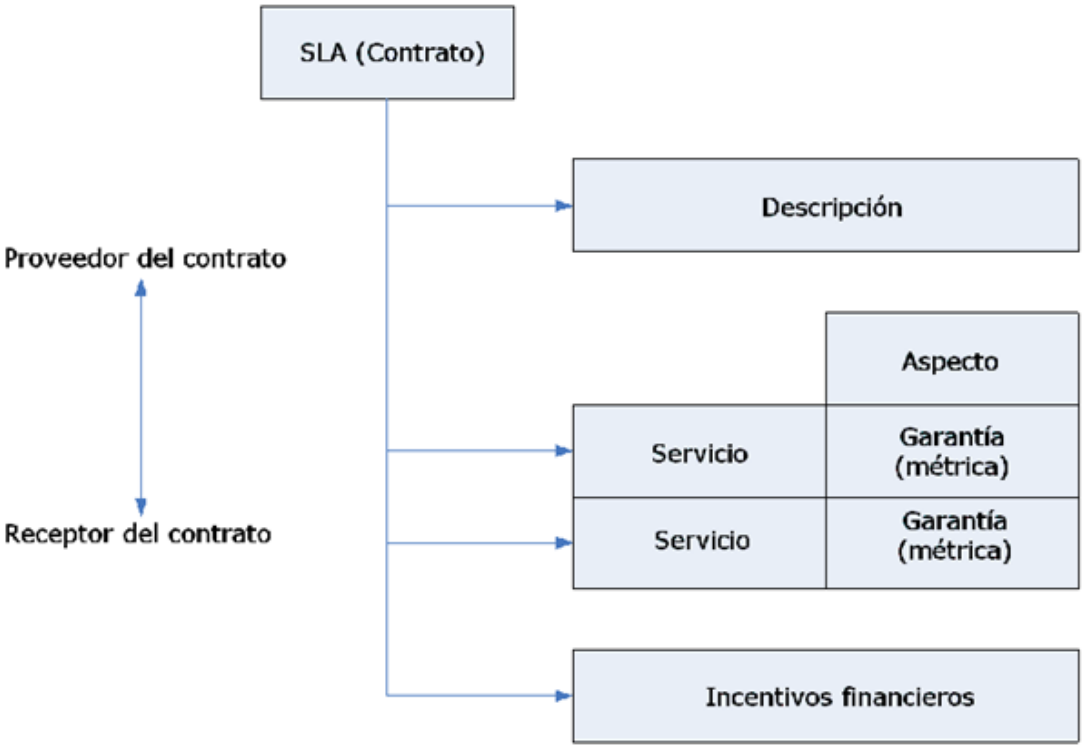
El gestor de contratos realiza las tareas siguientes para modelar el contrato.

Terminología del contrato

Un contrato es una recolección de objetivos. Un objetivo es un objetivo contractual u operativo (métrica) o un objetivo financiero (incentivo). El proceso de modelado del contrato implica tomar el contenido completo del contrato dentro del ámbito de la implementación y convertirlo al modelo de CA Business Service Insight.

Los diagramas siguientes describen este proceso.

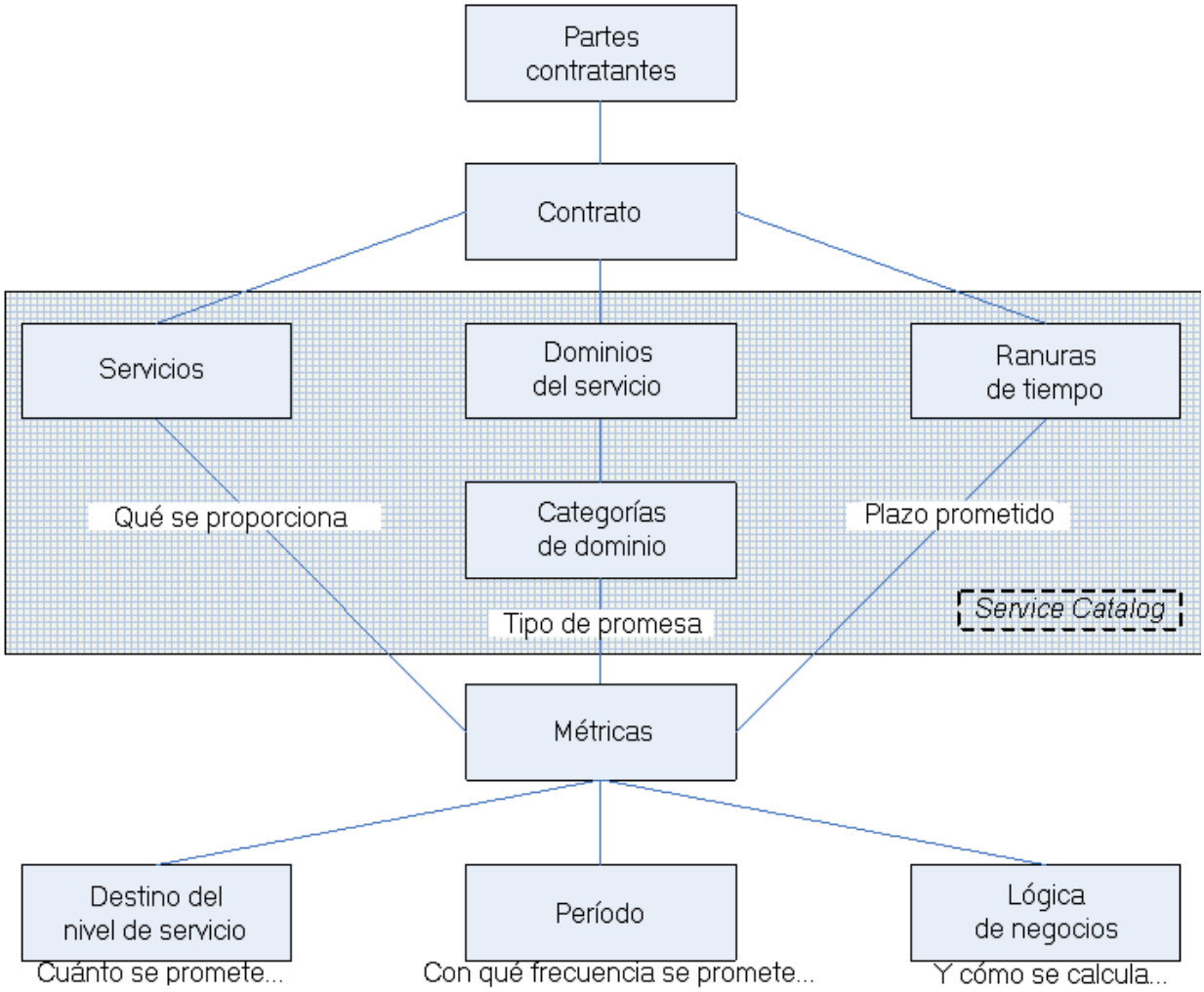




Nota: Deberán tenerse en cuenta los posibles planes futuros para el sistema, aunque solamente se modelarán los aspectos encuadrados dentro del ámbito de implementación. El ámbito debe incluir los requisitos de gestión de contratos generales de la organización para poder diseñar un modelo lo bastante amplio como para permitir desarrollos futuros. Esto minimiza y simplifica los cambios que deban hacerse en el futuro.

Convertir contratos de cliente al modelo de CA Business Service Insight es un proceso por el cual el gestor de contratos agrupa las garantías (métricas) ofrecidas dentro de límites comunes en grupos lógicos, componentes de servicio comunes, aspectos del contrato, etc. Esta agrupación lógica permite llevar a cabo una gestión de nivel de servicio muy flexible.

El modelo del contrato de CA Business Service Insight comprende las entidades mostradas en la ilustración siguiente:



Contrato

Define el acuerdo y la recolección de métricas. Un contrato puede corresponder a tres tipos diferentes dependiendo de la relación entre las partes contratantes. Puede ser un contrato SLA (acuerdo de nivel de servicios entre la organización y sus clientes, OLA (acuerdo de nivel operativo entre divisiones dentro de la organización) o UC (contrato de soporte entre la organización y sus suministradores, donde generalmente el UC forma parte de un servicio en el cual la organización suministra a otro cliente mediante un SLA).

Partes contratantes

Define el cliente de los servicios proporcionados, con el cual se firma el contrato. Se puede añadir una sola parte contratante a más de un contrato. Tenga en cuenta que dentro del contrato se puede definir el proveedor y receptor de los servicios que conciernen al contrato.

Métrica

Define un sólo objetivo de nivel de servicio o garantía. Cada métrica es una solicitud de medición que produce el resultado del nivel de servicio real en el cual se elabora un informe.

Una métrica comprende los elementos siguientes:

Tipo

Las métricas pueden ser de uno de los ocho tipos siguientes:

SLO	Objetivo de nivel de servicio.
informativo	Objetivo de nivel de servicio sin destino.
KPI	Indicador clave de rendimiento. Se utiliza para admitir conceptos industriales diferentes. En Telco KPI=SLO, y significa obligación del cliente, mientras que en TI significa obligación técnica.
KQI	Indicador clave de la calidad, una métrica global para medir el resultado agregado a partir de indicadores diferentes.
Provisional	Medición provisional para utilizar en otras métricas. Estos resultados de métricas no se pueden incluir en informes.
Consumo	Lo utilizan las métricas financieras para calcular el consumo de servicios/recursos. Se utiliza conjuntamente con las métricas de elemento del precio.
Incentivo	Métrica financiera que se utiliza para cálculo de incentivos (término positivo para sanción).

Elemento del precio Métrica financiera que se utiliza para calcular resultados basados en consumos. Precios diferentes por período o por número de unidades.

Período de seguimiento

Período de informe contractual o período para el cual se compara el resultado del nivel de servicio calculado real con el destino. El período de seguimiento en CA Business Service Insight se puede definir con una granularidad de hora, día, semana, mes, trimestre o año. Para el análisis de la causa raíz, además del período de seguimiento, la métrica se calcula también en las otras seis granularidades, pero los resultados para estos períodos se marcan como resultados operacionales solamente.

Nota: Esto se realiza solamente si estas granularidades de cálculo se especifican para la métrica.

Ranura de tiempo

Tiempo dentro del período de seguimiento durante el cual se puede aplicar la garantía específica u obligación, incluidas definiciones de ranura de tiempo excepcionales, como ventanas de mantenimiento previsto, días festivos estándar, etc.

Lógica de negocios

Fórmula que define cómo evaluar los datos sin procesar para calcular el valor de nivel de servicio real para ese aspecto del servicio, y los supuestos específicos que la calculadora debe tener en cuenta. Durante la fase de diseño, solamente se identifica el perfil del cálculo. Esto se define en más detalle en la fase de configuración.

Destino

Obligación de nivel de servicio contractual. El destino puede ser obligatorio o no, dependiendo del tipo de métrica en cuestión. En caso de que no se defina un destino, la métrica proporciona resultados solamente para elaboración de informes, y no puede compararse con una obligación contractual o garantía. Los destinos pueden ser estáticos o dinámicos.

Nota: Para seguir algunos de los conceptos presentados hasta aquí, consulte el Apéndice B: Casos prácticos (en la página 267).

Cada métrica va relacionada con las siguientes entidades de sistema contenidas dentro de la carpeta de plantillas del sistema:

Componente del servicio

Describe aquello que se tiene obligación de proporcionar.

Grupo de componentes del servicio

Recolección de componentes del servicio. Los componentes del servicio pueden formar parte de más de un grupo. Los grupos de componentes del servicio son una entidad opcional; si se utilizan, pueden brindar mayor flexibilidad a la hora de realizar informes.

Dominio del servicio

Aspecto específico de los componentes del servicio que tienen que medirse para controlar el nivel de servicio (por ejemplo, Rendimiento o Disponibilidad).

Categoría del dominio

Unidad de medida específica. Esto define la unidad de medida predeterminada y si el objetivo de destino corresponde a un valor mínimo o máximo. Esencialmente ésta es la dimensión específica del dominio del servicio que se mide (es decir, un % del tiempo disponible, número de interrupciones, tasa de rendimiento media, etc.).

Cada una de las entidades anteriores, así como sus relaciones con todos los objetivos de nivel de servicio que se van a controlar en
CA Business Service Insight, deberán identificarse durante la etapa de modelado del contrato.

Funcionamiento del proceso de modelado

Durante el proceso de modelado, todas las métricas que se tienen que configurar en el sistema se deberán definir junto con sus entidades relacionadas y de acuerdo con el contrato bajo consideración y los requisitos de elaboración de informes.

Antes de esta etapa, o durante ella, es recomendable decidir qué estándar de denominación utilizar para los nombres de las métricas, de modo que el sistema resulte ordenado y sea fácil desplazarse por él. También deberá pensar en la descripción de la métrica que se puede utilizar dentro de la sección Declaración del objetivo de la métrica.

El proceso de análisis del contrato incluye los pasos siguientes:

1. Identificación de los objetivos contractuales.

   Para cada objetivo, identifique:

   - Un nombre apropiado (tenga en cuenta los estándares de denominación).
   - Destino (opcional).
   - Período de seguimiento.
   - Unidad de medida.
   - Componente del servicio
   - Dominio del servicio
   - Categoría del dominio (y unidad de medida).
   - Ranura de tiempo (cuando: 24 x 7, ¿en horarios comerciales solamente?).
   - Perfil de cálculo (qué variables se necesitan y cómo se calcula el nivel de servicio).

   Nota: Algunos de estos elementos pueden no resultar claros en un principio, pero se pueden ajustar más tarde al afinar el catálogo de servicios.

2. Identificación, a partir del contrato, de los objetivos financieros y determinación de lo siguiente para cada objetivo financiero:
   - ¿Es una sanción, un incentivo o un objetivo de coste?
   - ¿Cuáles son las condiciones que lo activan?
   - ¿Para qué componentes del servicio se produce?
   - Dominio del servicio
   - Categoría del dominio (la unidad aquí puede ser una cantidad correspondiente a una moneda o un coste, un porcentaje de pago, etc.).

Una vez todos los objetivos se hayan anotado y definido, se recomienda revisar la lista completa de métricas e intentar optimizar/uniformizar el catálogo.

Durante este paso es importante asegurarse que cada uno de los componentes del servicio, dominios del servicio y categorías de dominio se definen acertadamente, así como de que pueden formar un catálogo claro y conciso de lo que se ofrece en el sistema. Esto incluye TODAS las métricas y contratos en el sistema. Esto garantiza la creación de un catálogo de servicios muy sólido dentro del sistema para permitir que éste en el futuro.

Los dominios del servicio y las categorías de dominio NO deben definirse en un nivel granular muy bajo. Deberán ser descriptivos pero en un nivel más alto que el de métrica.

Por ejemplo, si tiene las tres métricas siguientes:

- % de informes de interrupciones proporcionados puntualmente.
- % de informes de excepciones proporcionados puntualmente.
- % de informes de gestión proporcionados puntualmente.

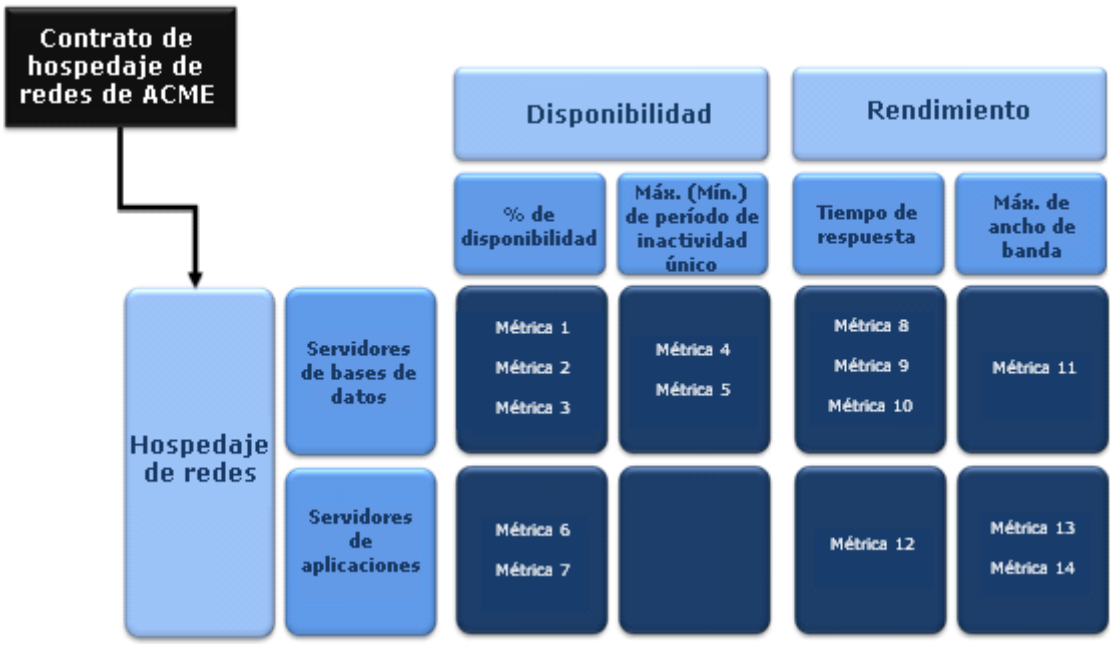
La mejor categoría donde se incluirían la tres métricas sería probablemente el dominio del servicio Rendimiento y la categoría del dominio % informes proporcionados puntualmente. La categoría del dominio no deberá mencionar el tipo de informes. Estas tres métricas tendrán probablemente el mismo método de cálculo y usarán la misma lógica de negocios (con la excepción de quizás un sólo parámetro para distinguir entre los diferentes tipos de informe).

Los dominios del servicio y las categorías de dominio adecuados se pueden tomar del estándar ITIL (Biblioteca de infraestructura de tecnologías de información). No obstante, estas son meras sugerencias de ejemplo; cada organización puede tener sus propios estándares definidos para ello. Consulte el Apéndice A: Dominios de servicio de muestra y categorías de dominio (en la página 265) para ver algunas sugerencias sobre dominios de servicio y categorías de dominio.

Nota: En este punto puede ser útil reunirse con todos los interesados para confirmar que el catálogo elegido es adecuado para sus necesidades actuales y futuras.

En los apéndices podrá ver otros ejemplos con puntos importantes. En estos ejemplos, se describe un sólo objetivo de contrato junto con su modelado. Al modelar situaciones reales, es necesario ser consciente de todos los objetivos para que las entidades del catálogo representen todos los objetivos de manera más amplia e integradora.

Una vez se complete el proceso de identificar todas las métricas y sus entidades relacionadas, el gestor de contratos dispone de una matriz que describe todas las métricas, tal y como muestra el diagrama siguiente.



En las secciones siguientes se describen otras cuestiones que deben tenerse en cuenta en el proceso de modelado.

Cuestiones para el gestor de contratos

Las siguientes son las cuestiones que el gestor de contratos debe tener en cuenta para asegurarse de que se respetan todos los aspectos:

¿Cómo sé que he elegido el componente del servicio correcto?

Si el componente del servicio se puede aplicar a más de un contrato y se puede medir en más de un aspecto, se ha definido correctamente.

Por ejemplo, el sistema X es un sistema proporcionado a más de un cliente y se puede medir por su disponibilidad, fiabilidad, rendimiento, etc.

¿Cómo sé que he elegido el dominio del servicio correcto?

Si el dominio del servicio se puede medir o calcular de más de una manera, está describiendo un aspecto general del servicio proporcionado y es por lo tanto el dominio del servicio correcto.

Por ejemplo, la disponibilidad se puede medir de distintas formas, una de las cuales es el porcentaje de tiempo disponible. Otras pueden ser el porcentaje de disponibilidad durante o fuera de horarios comerciales, el número de errores, tiempo medio entre errores (MTBF), tiempo medio de reparación, tiempo de inactividad máximo, tiempo de inactividad promedio, tiempo de inactividad total, etc. Todas estas son formas diferentes de evaluar la disponibilidad de un sistema concreto.

Casos que tener en cuenta durante el proceso de modelado

A continuación se incluyen varios ejemplos de casos, algunos comunes y otros más concretos, que describen conceptos que han de tenerse en cuenta en el proceso de modelado. Estos conceptos pueden resultar en una definición más precisa de las métricas y en un marco estable.

Métricas sin destino

Puesto que el campo de destino dentro de la definición de la métrica no es obligatorio, si no se ha definido habrá informes de nivel de servicio disponibles para la métrica. Sin embargo, no es posible generar ningún informe de nivel de servicio frente a destino y desviación (puesto que no hay ningún destino con el cual comparar el nivel de servicio calculado real). Estos tipos de métricas se definen cuando se requieren informes para obtener información que no forma parte de las obligaciones contractuales.

Definir este tipo de métrica proporciona al usuario todas las capacidades de obtención de detalles necesarias en los informes, así como proporciona al gestor del nivel del servicio la opción de aplicar mediciones a un destino en el futuro, en cualquier etapa.

***Por ejemplo*:**

La garantía contractual es proporcionar un 99 % de disponibilidad de la red e informar del número de tiempos de inactividad por cada mes.

Se definen dos métricas: una con un destino de 99 % de disponibilidad, y otra para contar el número de tiempos de inactividad para cada mes sin un destino. Se pueden emitir informes de las dos métricas, si bien sólo la primera tiene cálculos de desviación, debido a su obligación contractual.

Nota: Otro método posible para abordar este tipo de situación es utilizar salidas de lógica de negocios e informes de formato libre sobre estos datos. No obstante, esto impide la obtención de detalles de los datos en el informe, además de que no se puede utilizar el asistente de informes simples. La ventaja de utilizar salidas de lógica de negocios, por otra parte, es ahorrar potencia del motor gracias a un menor número de métricas.

Para obtener más información acerca de este método, consulte el caso práctico de Tablas de salidas de usuario (en la página 175).

Métrica con destinos

Cuando se define un destino para una métrica, hay dos formas de especificar el destino. Se puede especificar como destino estático o dinámico. El destino estático es el escenario más común, donde el destino se puede acordar según un valor válido durante la duración del contrato.

Por ejemplo:

La disponibilidad de red no debe inferior a un 98 % todos los meses.

El destino en este caso es 98 %.

También, el destino puede depender del rendimiento en los meses anteriores, o simplemente cambiar su valor a lo largo del año. Se podrían dar muchas situaciones alternativas, pero en general se implementan todas por medio de una fórmula. CA Business Service Insight admite esta función gracias a la adición de una llamada de función adicional desde la plantilla de lógica de negocios estándar. La función de destino puede acceder a otros parámetros desde el contexto de la lógica de negocios y puede admitir cualquier escenario que haga falta.

Por ejemplo, el tiempo de resolución de los tickets en el servicio de Help Desk que depende de la carga del Help Desk: el tiempo de resolución promedio para tickets de prioridad alta es 1 día si no hay más de 1000 tickets durante el mismo mes. Si se emiten más de 1000 tickets en el Help Desk dentro del mes, el tiempo de resolución medio para los tickets de prioridad alta es 2 días.

En este caso, la métrica se define con un destino dinámico que se evalúa dentro del script de la lógica de negocios según el número de tickets emitidos ese mes.

Nota: Para obtener más detalles sobre el método de implementación de destinos dinámicos, consulte el caso práctico Implementación de destinos dinámicos (en la página 185).

Parámetro de métrica

Un parámetro de métrica es un valor que se puede abordar desde dentro de la lógica de negocios de la métrica y que se puede cambiar fácilmente desde la definición de la métrica, sin necesidad de cambiar el código en sí. Se utiliza en lugar del valor codificado y se puede cambiar fácilmente.

Es importante identificar los parámetros de la métrica para identificar con facilidad los módulos de lógica de negocios y crear contenido reutilizable. Además, a los parámetros de las métricas se puede acceder mediante el asistente del contrato, el cual permite que el usuario final realice cambios fácilmente.

***Por ejemplo*:**

- Los incidentes de severidad 1 se deben resolver en 24 horas.

  En la obligación anterior el destino es una tasa de resolución de 24 horas y el nivel de gravedad (gravedad 1) se puede definir como parámetro.

- El número de tiempos de inactividad durante un mes no deberá ser más de 3 veces, donde el tiempo de inactividad es el tiempo que el sistema no ha estado disponible durante más de 5 minutos.

  En la obligación anterior, el tiempo que se considera definido como tiempo de inactividad se puede definir como parámetro.

Parámetro de contrato

Un parámetro de contrato es un valor que pueden abordar todas las métricas dentro de un contrato. Dentro de la métrica se puede usar un parámetro de contrato que emplee el mismo método que un parámetro de métrica, si bien se define como parámetro dinámico.

Se recomienda utilizar un parámetro de contrato cuando haya más de una métrica que requiera utilizar el mismo valor. Otro incentivo importante para usar un parámetro de contrato es facilitar el mantenimiento del contrato. Dado que los parámetros tienden a cambiar a menudo y se tiene que actualizar dentro del sistema, es más fácil acceder a una sola ubicación en el contrato y cambiar todos los valores de los parámetros simultáneamente que acceder a cada métrica en el contrato y cambiar los valores de los parámetros a nivel de métrica.

Por lo tanto, el modelado más recomendado es definir los parámetros en el nivel de contrato como parámetros de contrato y acceder a sus valores a través de los parámetros dinámicos de nivel de métrica.

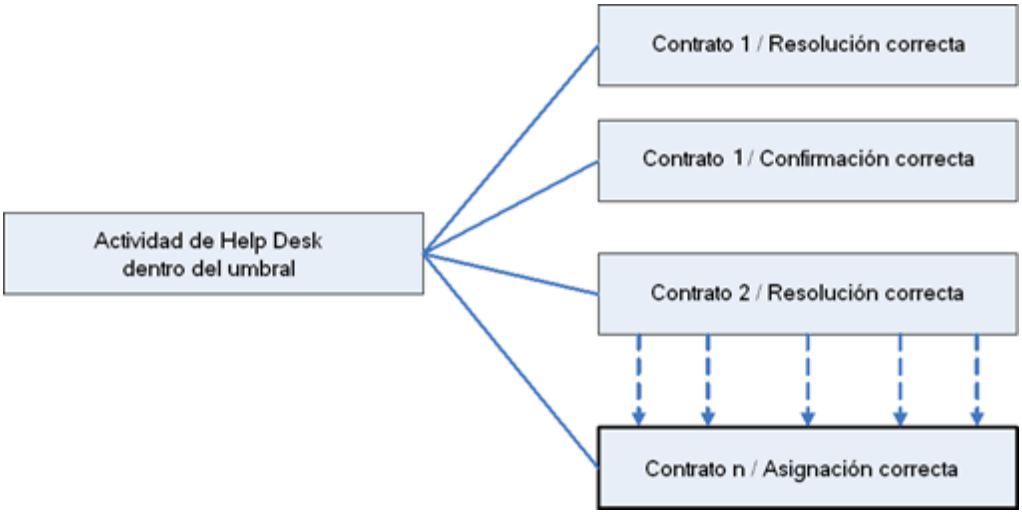
Para ver un ejemplo, consulte el caso práctico Rendimiento del Help Desk (en la página 276).

Módulos y plantillas de lógica de negocios

Las plantillas de lógica de negocios son una forma sencilla de almacenar un método de cálculo para una métrica. Son un componente de lógica de negocios integral y una forma muy práctica de crear una línea de referencia para otros componentes de lógica de negocios. Los nuevos componentes de lógica de negocios creados a partir de una copia de plantilla copian el código y crean una nueva instancia del mismo. Sin embargo, en general, la flexibilidad al utilizar plantillas es bastante pobre, y se deberán utilizar módulos de lógica de negocios cuando sea posible.

Los módulos de lógica de negocios son componentes de código independientes que permiten que otra lógica de negocios reutilice la misma base de código. Los módulos pueden incluir también otros módulos, de modo que los niveles de jerarquía pueden ser varios. Al utilizar módulos, el código queda contenido en un sitio y cada uno de los otros componentes que vinculan a él lo pueden reutilizar. Esta reutilización de secciones de código facilita el mantenimiento, ya que elimina la duplicación de código. Además, permite aplicar los cambios en la lógica en todo el sistema de forma rápida y sencilla.

Durante la etapa de diseño, es necesario identificar los módulos de lógica de negocios principales y sus parámetros relacionados. Una vez se completa el modelado de contrato y el gestor de contratos ve claramente la lógica que se tiene que utilizar, se pueden identificar los cálculos en común y se pueden definir en módulos aparte.



El diagrama anterior muestra un módulo que calcula la tasa de éxito de la actividad del Help Desk para cumplir con un objetivo dentro de umbrales concretos. Para hacer una implementación según lo descrito, es necesario de definir dos parámetros, conocidos como "parámetros de métricas": uno que define el tipo de actividad del Help Desk y otro para el umbral de comparación (consulte la definición de *parámetro de métrica* en Casos que tener en cuenta durante el proceso de modelado (en la página 37)).

Preste atención a los tipos de cálculos implementados en el sistema. Probablemente descubrirá que se pueden aplicar tipos similares cambiando una pequeña sección del código y usando un parámetro que actúe como "conmutador" entre ellos. De esta manera, puede minimizar la cantidad de código que necesita crear y maximizar la reutilización de código.

Plantillas de nivel de servicio

Una plantilla de nivel de servicio es una recolección empaquetada de componentes de servicio y la métrica asociada definida para medirlos. Estas plantillas de nivel de servicio se pueden crear según haga falta y se definen a menudo según los aspectos de servicio más comúnmente medidos.

La cuestión clave a la hora de definir plantillas de nivel de servicio es que se deben identificar y mostrar todos los parámetros posibles que se podrían utilizar para cambiar el comportamiento de las métricas. Esto proporciona máxima flexibilidad al sistema y facilita la configuración para sus usuarios. Al utilizar la plantilla de nivel de servicio para crear una nueva métrica, la interfaz del asistente muestra cada uno de los parámetros de configuración, con lo que hace falta personalizarla muy poco, o nada, para activar el contrato. Los parámetros mostrados al usuario mediante el asistente están en la declaración del objetivo. Por lo tanto, incluya en la declaración del objetivo de la métrica todos los parámetros que se desea que pueda cambiar el usuario final.

Para disfrutar de una eficiencia máxima con las plantillas de nivel de servicio, deberá aspirar a realizar toda la lógica de negocios a través de la funcionalidad de módulo, como se ha descrito previamente.

A continuación se incluye un escenario de ejemplo para utilizar plantillas de nivel de servicio, donde se ofrecen componentes de servicio de alojamiento de aplicaciones en los tres niveles siguientes: Bronce, Plata, y Oro, dependiendo de la cantidad que el cliente desee pagar. Se pueden proporcionar métricas adicionales en cada nivel de alojamiento por encima, junto con destinos más estrictos. Cada uno de estos niveles podrá ser un buen candidato para crear una plantilla de nivel de servicio como la que incluye el escenario de muestra siguiente:

- Alojamiento de aplicaciones: Bronce (5 métricas)
- Alojamiento de aplicaciones: Plata (8 métricas)
- Alojamiento de aplicaciones: Oro (12 métricas)

Ahora, cuándo se añade un cliente nuevo al sistema, es fácil elegir la definición necesaria basándose en lo que el cliente desea. Al usar la interfaz de asistente podrá personalizar cada una de las métricas incluidas para ese cliente concreto. Tenga en cuenta que también es posible elegir solamente algunas de las métricas en las definiciones, o incluso agregar métricas de dos definiciones diferentes al contrato del nuevo cliente, si se necesita una personalización especial.

La importancia de un catálogo de servicios sólido

Como se ha descrito previamente, el catálogo de servicios es un componente clave del sistema y se deberá configurar de manera estructurada. Esto permite usar el sistema eficazmente por parte de todos los usuarios, evitando confusiones. También permite que el sistema crezca con la organización y gestionar futuras mejoras y adiciones con un impacto mínimo en lo ya creado.

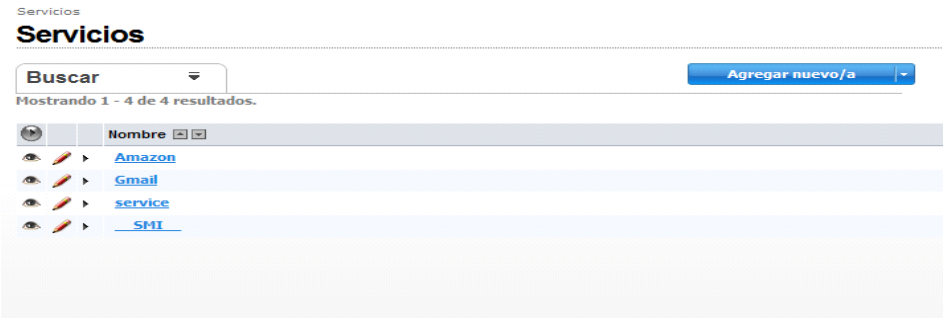
El sistema ofrece un grado alto de flexibilidad a la hora de crear y gestionar el catálogo de componentes del servicio y acuerdos SLA. Sin embargo, puesto que su calidad dependerá del diseño, dedicar tiempo a planificar para el futuro es esencial en las primeras etapas de diseño.

Definir el catálogo de servicios de CA Business Service Insight de una forma eficaz y estructurada proporciona flexibilidad a su sistema para agregar servicios y dominios al catálogo de servicios más adelante. También permite agregar contratos, métricas, alertas e informes en el futuro. Además, conduce a un acercamiento más estructurado a la lógica de negocios y prepara el camino para poder utilizar plantillas y módulos estandarizados para gestionar datos de negocio y cálculos asociados.

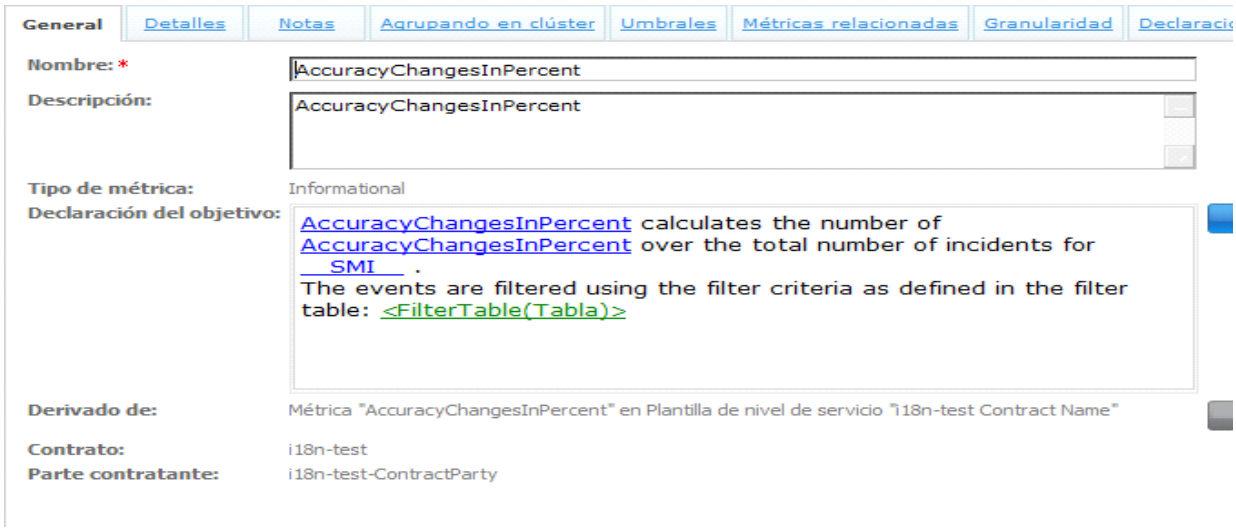
Conjuntamente con el catálogo, las plantillas de nivel de servicio definidas en el sistema proporcionan al gestor de contratos una forma excelente de crear nuevos contratos muy fácilmente y con poco o ningún conocimiento de los niveles subyacentes de datos requeridos. Tras una buena configuración, deberá ser posible configurar un nuevo contrato para un cliente modificando solamente los parámetros en la plantilla de nivel de servicio. Todo esto se basa, no obstante, en que el catálogo y las definiciones se hayan establecido de la manera más eficaz posible. Estos parámetros se muestran todos mediante la declaración del objetivo para cada métrica en la plantilla de nivel de servicio y se pueden modificar o allí, o desde el asistente al usar la definición.

Ejemplo:

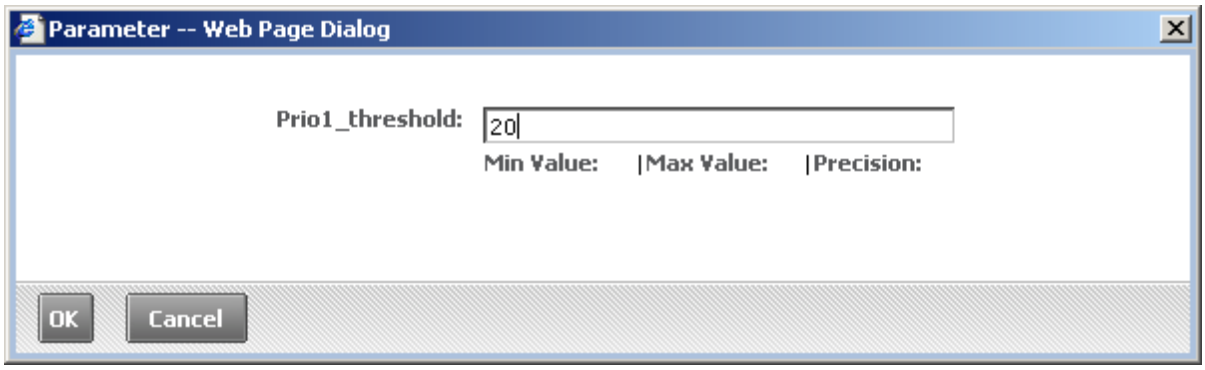
Cuando se usa una plantilla de nivel de servicio para implementar métricas de soporte al cliente en un contrato, se pueden seleccionar las métricas necesarias a partir de una definición existente.



Dentro de esta plantilla de nivel de servicio, hay una métrica llamada "*% de incidentes respondidos puntualmente*". Puede ver que aquí hay cierto nivel de subjetividad, puesto que el componente "*puntualmente*" puede quedar sujeto a interrogación. El ejemplo siguiente explica la medición hecha en esta métrica:



La declaración del objetivo incluida al final de la ficha General de la métrica (o también en la ficha Declaración del objetivo) incluye todos los parámetros mostrados en esta métrica. En la ilustración anterior, la definición de "*puntuales*" corresponde a 20 minutos. Este es un parámetro personalizable para permitir hacer una reinterpretación propia de esta métrica predefinida. Para cambiar este valor, se puede hacer clic en el vínculo del parámetro de 20 minutos.



De esta manera, la nueva métrica creada a partir de la plantilla de nivel de servicio se puede personalizar sin necesidad de modificar la lógica de negocios subyacente de la métrica. Tenga en cuenta que esto también asume que la lógica de negocios se ha escrito para incorporar estos parámetros en el cálculo del nivel de servicio.

Este ejemplo sencillo muestra claramente lo importante que es crear un conjunto de plantillas de nivel de servicio potente y flexible para que el catálogo del sistema pueda crear contratos futuros a partir ellas aprovechando su información.

Gestión financiera (sanciones, incentivos y costes)

En versiones anteriores de CA Business Service Insight, había entidades de contrato conocidas como "sanciones" que se implementaban mediante fórmulas similares a las de Excel. Las sanciones basaban sus resultados únicamente en la entrada de las métricas del contrato y se apoyaban en funciones básicas para calcular la cantidad de sanción resultante. A partir de la versión 4.0 en adelante, se han reemplazado las sanciones por métricas financieras plenas que el usuario puede crear, lo cual aporta mayor flexibilidad. Estas métricas financieras se pueden utilizar para proporcionar información de incentivo o coste acerca del contrato.

Nota: el incentivo reemplaza el término "sanción" de CA Business Service Insight 3.0 y versiones anteriores, y puede ser positivo o negativo dependiendo del rendimiento. Un incentivo negativo, sin embargo, es fundamentalmente igual que una sanción. Es también importante tener en cuenta que, si se está implementando una sanción con el tipo de métrica de incentivo, no se *debe* olvidar hacer que la función Result() devuelva un valor negativo. Esto permite que las funciones de resumen que combinen resultados de métricas diferentes puedan ajustar los totales en la dirección correcta. Es decir, un incentivo aumenta el valor, mientras que una sanción lo reduce.

La versión de CA Business Service Insight 4.0 también permite crear métricas de consumo que midan el uso de recursos y componentes del servicio, así como combinar esto con una métrica de elemento del precio para determinar el coste de ese servicio o recurso. La combinación de éstas con la funcionalidad de previsión mejorada permite ahora crear métricas de gestión financiera muy completas.

Las métricas financieras pueden adquirir también el resultado de otras métricas contractuales y determinar los valores de sanción o incentivo relacionados basándose en el rendimiento de estas métricas contractuales. Pueden trabajar también con otros tipos de información para determinar su resultado, como información de precio por unidad y modelos de previsión, lo cual permite elaborar informes como informes de costes esperados frente a costes reales.

***Ejemplo de elemento de coste*:**

Una aplicación de riesgos particular tiene un coste asociado basado en el número de usuarios simultáneos del sistema. Se calcula mensualmente y existe un valor pronosticado para esta aplicación. El precio por unidad de esta aplicación (coste por usuario simultáneo) se proporciona en la tabla a continuación (supongamos que esta aplicación queda bajo el 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

El número pronosticado de usuarios simultáneos para este período está también disponible (de nuevo, 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

Al modelar esta métrica de coste mediante estas tablas de cálculo de costes es posible determinar el coste de aplicación mensual, basado en el número real de usuarios simultáneos. Esta información proviene del origen de datos y se puede multiplicar por las cifras de precio por unidad anteriores para dar las cifras de costes. Se puede comparar también con los valores pronosticados para proporcionar un análisis del coste esperado frente al coste real. En este caso, se requiere lógica de negocios para determinar el número real de usuarios simultáneos encontrados durante el período y multiplicarlo por los valores de coste por unidad. Además, la función de previsión dentro de la lógica de negocios hace referencia a la información de previsión. Este es un ejemplo de cómo se aplica una métrica de elemento de coste.

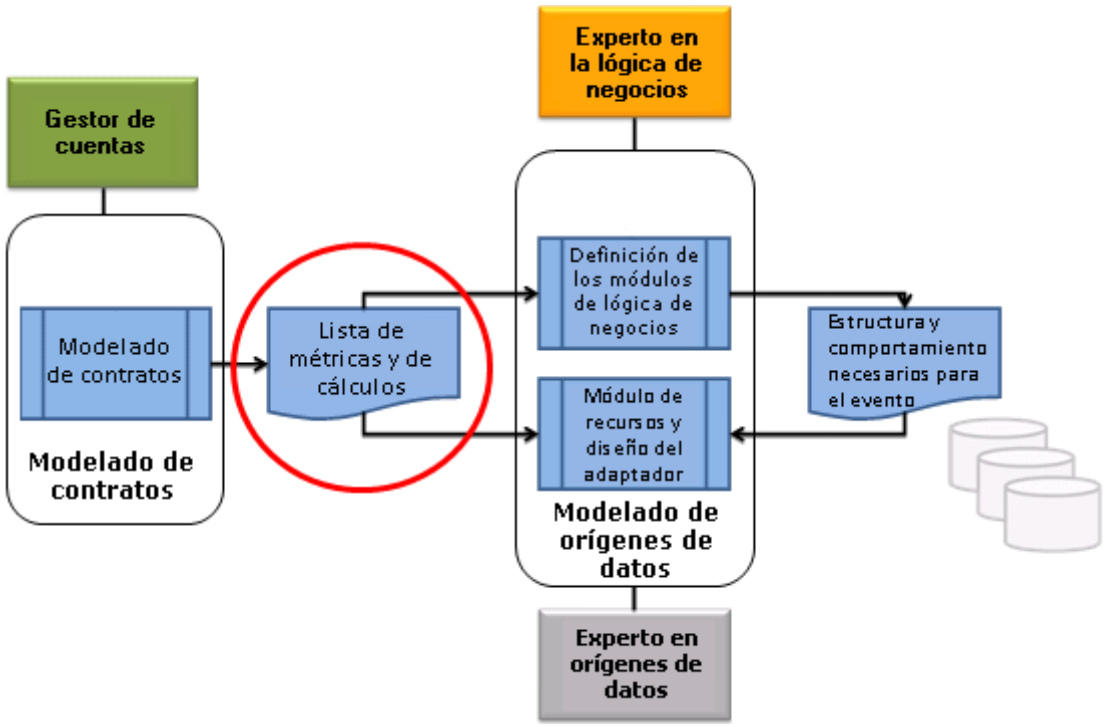
***Ejemplo de escenario de sanción*:**

El SLA de cliente tiene una cláusula en caso de que no haya rendimiento para garantizar que la red esté disponible el 98 % del tiempo en cualquier mes durante horarios comerciales. Todo nivel de servicio mensual por debajo de este valor incurre en el pago de una sanción basado en una fórmula (sanción = 1000 $ por cada porcentaje por debajo del destino (96,5 % = (98-redondeo de(96,5)) * 1000 = (98-97) * 1000 = -1000 $)).

Para implementar esta condición de sanción, se puede crear una métrica de incentivo financiero que tome su entrada de una métrica existente (*disponibilidad de la red* >= 98 %). El proceso de registro para esta métrica utiliza el proceso RegisterByMetric() para recibir los valores de nivel de servicio de esa métrica y realizar las comparaciones. Esto envía los valores de nivel de servicio para el período de seguimiento de esa métrica a esta métrica financiera como eventos que se utilizan a continuación como parte del cálculo para determinar la sanción para ese mismo período mediante la fórmula del escenario.

Nota: Para ver otros casos prácticos, consulte el caso práctico Caso práctico de modelado de métricas financieras (en la página 279).

Resultados de la etapa de modelado del contrato (gestor de contratos y experto en el origen de datos)



Como se muestra en el diagrama anterior, para continuar con la etapa de modelado de datos de la solución, es necesario que el gestor de contratos proporcione la lista de métricas y sus cálculos necesarios al final de la etapa de modelado del contrato. La lista incluye la siguiente información:

- Una definición terminada del catálogo de servicios, que incluye:
  - Una lista completa de componentes del servicio que se van a ofrecer.
  - Enumeración de los dominios del servicio y las categorías de dominio.
  - Definición de todas las ranuras de tiempo requeridas para cada una de las métricas.
  - Una lista de módulos de lógica de negocios para gestionar cada tipo de cálculo (incluidos los parámetros necesarios para implementarlos).
- Una lista completa de métricas que implementar, con lo siguiente:
  - Estándares de denominación bien definidos para las métricas
  - Categorización de cada métrica según componentes de catálogo de servicios predefinidos

Este documento es una útil herramienta para garantizar que todas las definiciones clave para un contrato se han especificado y todas sus métricas se han definido por completo.

La información dentro de esta hoja de cálculo es la entrada que el experto en el origen de datos necesita para continuar con la etapa siguiente, el modelado de datos.

Una vez completados estos elementos, es posible continuar con la etapa siguiente, en la cual puede empezar a modelar con datos reales provenientes de los orígenes. Sin haber terminado un modelo de contrato es muy difícil saber exactamente lo que se requiere del origen de datos para poder satisfacer las obligaciones contractuales.

Modelado de datos (experto en el origen de datos, experto en la lógica de negocios)

La sección Modelado de datos describe la segunda parte del proceso de diseño. La fase de modelado de datos es el proceso de tomar los datos de los orígenes del cliente, identificar los componentes específicos de los datos obligatorios, decidir cómo tienen estos datos que ser normalizados y evaluar cómo asignar los datos relevantes a la métrica correspondiente (mediante el proceso de registro).

Esta fase incluye las tareas siguientes:

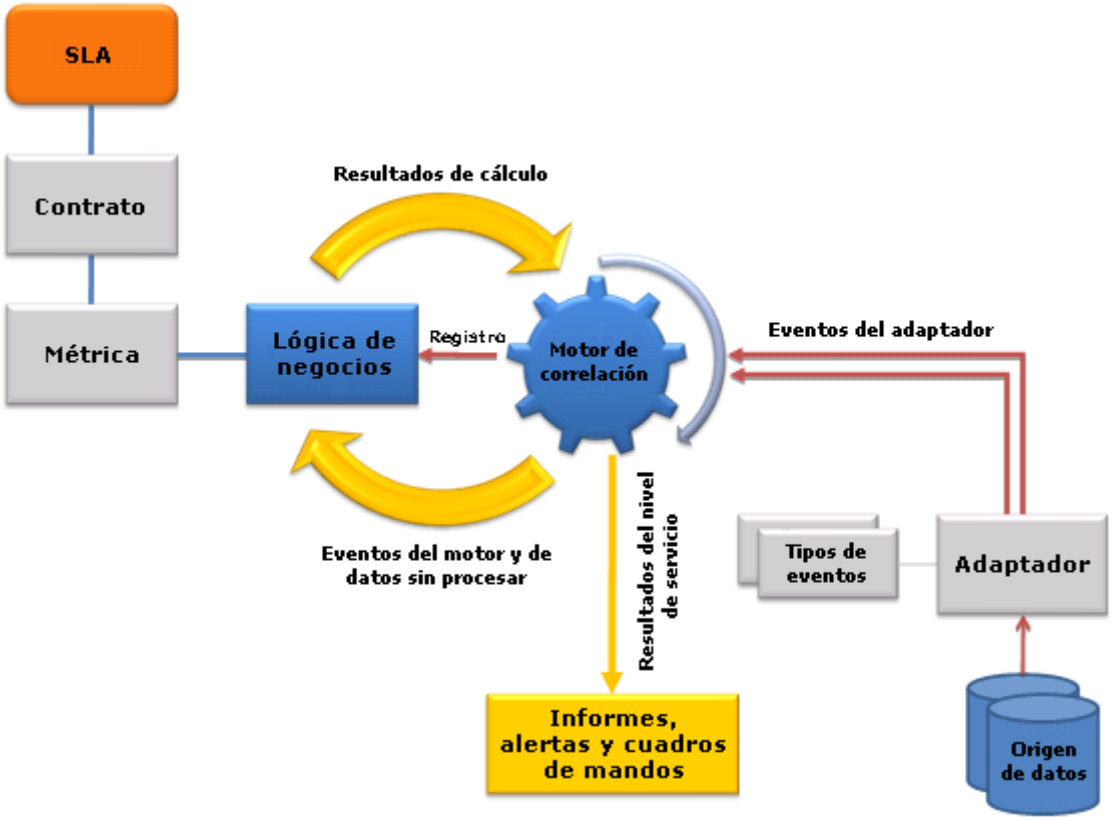
- **Experto en la lógica de negocios**:
  - Identificación y definición de la estructura del evento de entrada obligatorio para que el cálculo se defina más tarde como tipos de eventos.
  - Construcción de los campos obligatorios en tipo de evento para proporcionar los datos necesarios de todos los cálculos en los que se utilizan.
  - Definición del proceso de registro de métrica para maximizar el flujo de eventos.
  - Determinación de cómo construir el modelo de recurso a partir de los datos disponibles para satisfacer todos los requisitos de generación de informes y lógica.
- **Experto en el origen de datos**:
  - Identificación del tipo de origen de datos y tomar la decisión de cómo lo debería normalizar el adaptador en los tipos de eventos definidos y se debería llevar a la base de datos de CA Business Service Insight.
  - Determinación de los identificadores de tipo de evento que deberían estar adjuntos a los datos.

Eventos y su flujo

Los flujos de datos dentro del sistema toman la forma de eventos. Un evento es un mensaje de información creado por el adaptador a partir de los datos de origen, en un formato que CA Business Service Insight puede utilizar para sus cálculos de nivel de servicio. Los datos sin procesar siempre constan de eventos.

El centro del diseño debe estar por lo tanto en este flujo de eventos dentro del sistema.

Antes modelar los requisitos de datos, tanto el experto en la lógica de negocios como el experto en el origen de datos tienen que tener una comprensión sólida de los eventos y de su flujo dentro del sistema de CA Business Service Insight. El diagrama siguiente ilustra en un alto nivel este flujo de eventos básico.



El diagrama anterior muestra cómo recuperan los eventos del origen de datos los adaptadores y cómo se normalizan en una estructura de eventos estándar definida como el tipo de evento. Los adaptadores envían estos eventos a CA Business Service Insight. Estos eventos se conocen como eventos de datos sin procesar.

Los cálculos de lógica de negocios en cada métrica se basan en un subconjunto de eventos de datos sin procesar. La lógica de negocios por lo tanto pide este subconjunto realizando un *registro*.

Basado en la declaración de registro, el motor de correlación envía solamente los eventos de datos sin procesar relevantes para los cálculos de lógica de negocios.

Los tipos adicionales de eventos enviados a la lógica de negocios son los eventos de motor. Todos los conceptos implicados en este proceso se tratan en detalle en este capítulo.

Esta sección se centra en las partes siguientes del diagrama:

- Fuente de datos
- Adaptador
- Tipos de eventos
- Registro de métrica

El modelo de datos de CA Business Service Insight se ha diseñado para maximizar la eficiencia de este flujo de datos dentro del sistema.

En general, CA Business Service Insight funciona en dos capas: la capa de infraestructura y la capa de modelo de negocio. Por desglosarlo de una manera sencilla y explicativa, la capa de infraestructura incluye adaptadores, recursos y objetos de tipo de evento, mientras que la capa de negocio incluye contratos, métricas y objetos de servicios. Entre las dos capas hay una capa de corrección de compatibilidad virtual, llamada "capa de correlación".

El identificador de evento corresponde al objeto de tipo de evento. El tipo de evento determina cómo se definen los eventos y cómo se informa de ellos a CA Business Service Insight. También define la estructura del campo de datos de evento para que la lógica de negocios la pueda interpretar durante el proceso.

Otro identificador de evento es el recurso, la entidad más pequeña utilizada en los cálculos. Por ejemplo, al calcular la disponibilidad del servidor, la definición lógica de la entidad más pequeña que requiere emisión de informes puede ser un servidor específico, o puede ser un cliente al emitir un informe sobre la gestión del ticket de ese cliente. Un recurso es una definición de una entidad de CA Business Service Insight que se deriva tanto del origen de datos como del requisito de cálculo. Cada recurso recibe un tipo de recurso que es el identificador de recurso que determina exactamente qué "tipo" de recurso se ha definido. Cada recurso debe tener asociado un tipo de recurso, lo cual también permite agregar atributos personalizados que asociar con cada recurso. Para obtener más información acerca de estos atributos, consulte Recursos y su gestión (en la página 60).

La correlación se produce entre eventos de adaptador entrantes y la métrica de contrato. El núcleo de este proceso de correlación es la adjudicación de recursos y el registro de métricas.

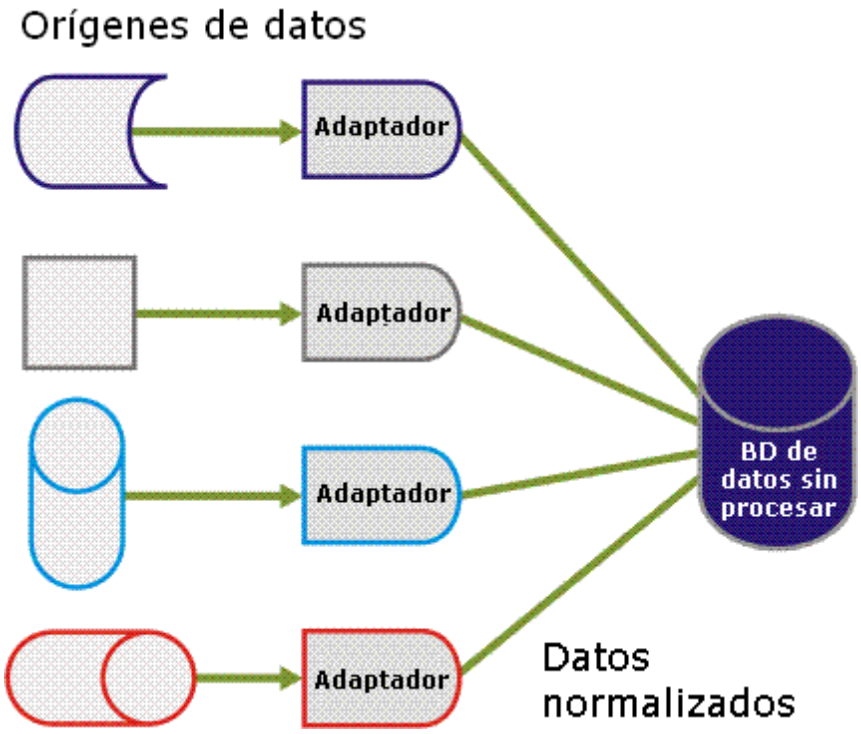
La adjudicación de recursos y el registro de métricas especifican qué flujos de eventos de recurso se miden y por medio de qué métrica.

Tenga en cuenta que con el registro de métrica, puede existir cierto grado de reutilización y codependencia con otras métricas, ya que es posible utilizar la salida de una métrica como la entrada de otra. De forma parecida, hay eventos provisionales que no se utilizan como la salida de una métrica para la medición del nivel de servicio, sino más como un paso de cálculo intermedio que pueden utilizar después otras métricas.

Modelo de datos: descripción general

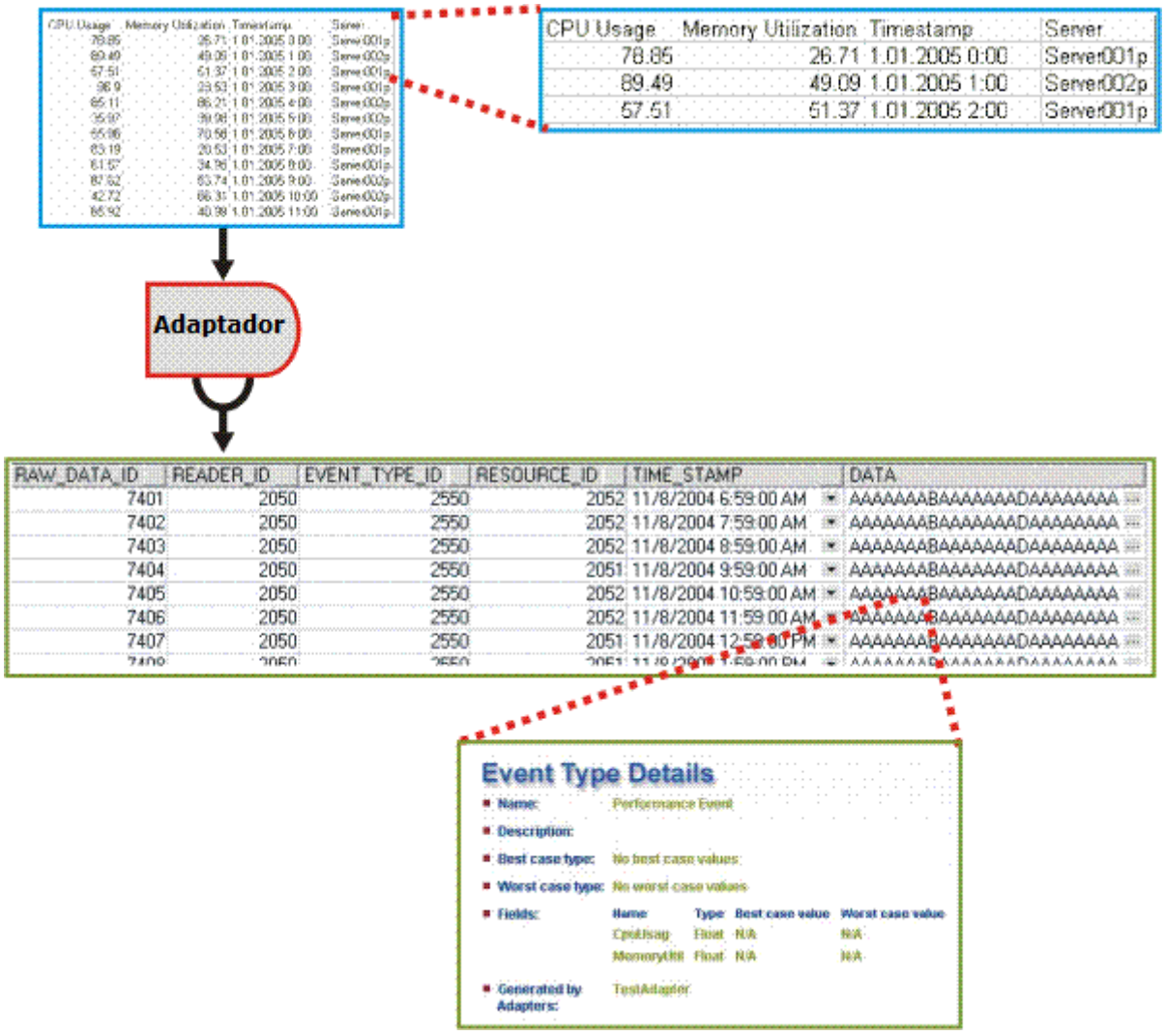
El modelo de datos de CA Business Service Insight se ha diseñado para superar el siguiente reto.

Los adaptadores recuperan datos sin procesar a partir de orígenes de datos diversos, dispares y en una variedad de formatos. Estos datos diversos tienen que recuperarse y homogeneizarse en una sola tabla de base de datos. Por lo tanto, los adaptadores son necesarios para leer y normalizar los datos en un modelo de datos unificado, como muestra la ilustración siguiente.



Como parte de este proceso, todos los campos de datos se insertan en el mismo campo de la tabla de base de datos, pero se cifran. Cada línea insertada en la base de datos de CA Business Service Insight tiene un identificador de tipo de evento adjunto. La definición del tipo de evento contiene descripciones de los campos de datos. También permite que el motor de correlación interprete correctamente los campos de datos e identifique cuándo los necesita la lógica de negocios para realizar los cálculos.

La ilustración siguiente muestra una representación gráfica de la recuperación de los datos y la sección de cumplimentación de la base de datos de este proceso. También se ilustra una sección ampliada que muestra qué representan los datos en términos reales, más que cómo aparecen los datos sin procesar.



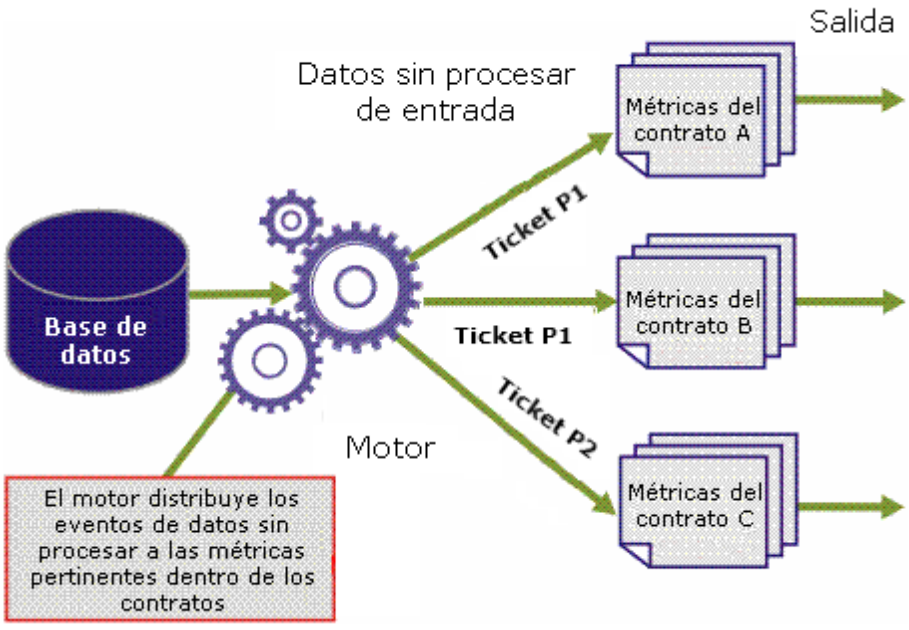
El sistema de CA Business Service Insight también incluye todos los contratos y métricas que requieren evaluación frente a los datos sin procesar para producir la información de rendimiento del nivel de servicio resultante. Cada métrica debe recibir solamente el subconjunto de datos relevante para su cálculo. Los datos sin procesar incluyen un número potencialmente amplio de registros de diversos tipos. Utilizar la métrica para filtrar los eventos relevantes por sus valores es muy poco eficaz. Por lo tanto, el motor de CA Business Service Insight distribuye los datos sin procesar relevantes a cada métrica específica.

Ejemplo:

Para las dos métricas siguientes en un contrato:

- Tiempo de resolución promedio de tickets de prioridad 1 (P1)
- Tiempo de resolución promedio de tickets de prioridad 2 (P2)

La primera métrica es necesaria para evaluar solamente los tickets con prioridad 1, y la segunda métrica, solamente tickets con prioridad 2. Por lo tanto, el motor tiene que distribuir los registros en consecuencia. Además, el tiempo de resolución dentro de un contrato se calcula para los tickets P1 abiertos para la parte contratante A, en el segundo contrato los tickets P1 para la parte contratante B y en el tercero los tickets P2 para la parte contratante C. Por lo tanto, el motor es necesario para seleccionar el tipo de ticket y cliente para el cual se emitió el informe, como muestra la ilustración siguiente.



Como se ha explicado previamente, los registros de datos sin procesar llevan identificadores adjuntos que permiten al motor identificar los registros y los eventos relevantes para la lógica de negocios de cada métrica. Los dos identificadores son el tipo de evento y el recurso.

Traducción del adaptador y normalización

La función del adaptador es leer los datos del origen de datos y normalizarlos en formato de evento. Cada evento de CA Business Service Insight se compone de los campos siguientes:

- ID del recurso
- ID de tipo de evento
- Marca de tiempo
- Campos de valor según tipos de evento

El adaptador tiene que vincular los campos originales en el origen de datos con los campos de recurso de CA Business Service Insight correspondientes. Para hacerlo, usa una tabla de traducción que contiene el valor encontrado en el origen de datos y el ID de recurso de CA Business Service Insight correspondiente.

El proceso de adjuntar el ID de recurso y el ID de tipo de evento al valor de origen de datos relevante se llama *traducción del adaptador*. Durante este proceso se crea la tabla de traducción con los valores coincidentes. El adaptador usa esta tabla para rellenar el ID de tipo de evento relevante y el ID de recurso en el registro del evento que está creando. Para traducir recursos y tipos de eventos se crean tablas independientes.

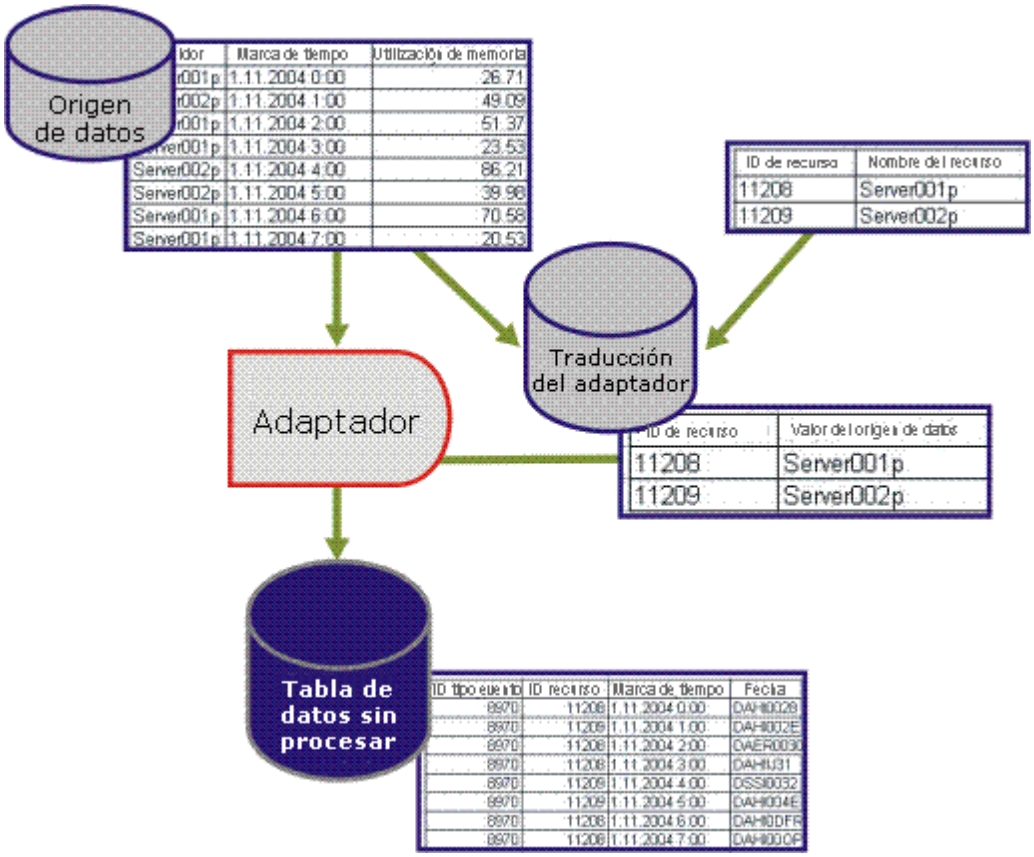
Como hemos mencionado anteriormente, el ID de recurso y ID de tipo de evento se utilizan para el registro, mientras que el campo de datos y los valores de marca de tiempo se utilizan para realizar cálculos reales.

El motor utiliza también el campo de marca de tiempo para determinar en qué orden y en qué momento enviar eventos para los cálculos.

La definición del tipo de evento se hace manualmente en CA Business Service Insight basándose en el origen de datos de entrada y la salida necesaria.

Nota: La definición de recurso se puede realizar manualmente o automáticamente utilizando scripts de traducción (consulte Recursos y su gestión (en la página 60) para ver más detalles).

La ilustración siguiente muestra la interacción entre el origen de datos, la tabla de traducción del adaptador, el adaptador y tabla de datos sin procesar de CA Business Service Insight.



Registro de métrica

Para que el motor de correlación sepa qué datos se pueden solicitar, la métrica debe registrar su presencia y sus requisitos con el motor de correlación.

El registro de métrica es la solicitud que realiza una métrica para recibir eventos, y solamente los que tienen que incluirse en su cálculo. Esta solicitud se realiza a través del tipo de evento indicado por el identificador del evento y el recurso.

El registro sólo se pueden realizar en base a un sólo recurso o un grupo de recursos.

Ejemplo:

Para la métrica informativa "Número de tiempos de inactividad del servidor X" y en el supuesto de que el origen de datos proporcione una notificación cuándo un servidor entre en funcionamiento o deje de funcionar, y la notificación indica si está activo o inactivo en un momento concreto, entonces el registro será:

Tipo de evento: evento de servidor activo/inactivo

Recurso: servidor X

De acuerdo con el registro anterior, lo que sucede es que el motor filtra todos los eventos que tienen definiciones de activo/inactivo en su campo tipo de evento, además de tener el servidor X en el campo Recurso.

Una vez se activa un contrato en el sistema de CA Business Service Insight, todas las métricas registran los eventos relevantes necesarios para sus cálculos. Basándose en estas solicitudes, el motor de correlación marca los eventos relevantes para cada lógica de negocios. Una vez empiezan los cálculos, los eventos relevantes se envían a cada métrica para realizar el cálculo.

Recursos y su gestión

Para permitir realizar un registro dinámico, los recursos se pueden adjudicar de manera individual por su identificador o su nombre único, o bien por su relación con grupos lógicos.

Ejemplo:

Para la métrica "Número global de tiempos de inactividad de servidores de centro de datos", el registro sería:

Tipo de evento: eventos de activo/inactivo

Recurso: todos los recursos etiquetados bajo servidores de centro de datos (probablemente un grupo de recursos).

Comprensión del ciclo de vida de un recurso

Un recurso es una entidad física o lógica que puede cambiar sus características con el transcurso del tiempo. Los recursos se pueden adjudicar a ciertos componentes de servicio o partes contratantes, etc., en un momento, y después readjudicarlos en otro momento. CA Business Service Insight captura cada uno de estos cambios o readjudicaciones para poder realizar cálculos en cualquier momento dado, basándose en la configuración de los recursos y la adjudicación en ese momento exacto.

Se pueden hacer cambios en un recurso o en sus adjudicaciones en cualquier momento, si bien en este caso es necesario crear una nueva versión del recurso. Cada nueva versión tiene que tener una fecha de vigencia definida para cuando se produzcan los cambios. Los cambios seguirán produciéndose en el futuro, a menos que se encuentren otros cambios en una versión posterior de ese mismo recurso. Los cambios sólo aparecerán visibles y disponibles para el motor de cálculo una vez que esta nueva versión se haya activado y esté vigente. Este proceso se denomina "confirmar" el recurso.

Dentro de CA Business Service Insight hay también una forma de gestionar varias adjudicaciones de recursos en un paso. Este método utiliza "conjuntos de cambios". Los conjuntos de cambios permiten aplicar grandes volúmenes de cambios en los recursos en una sola "transacción", de forma similar a como funciona una base de datos transaccional. Se pueden aplicar todos los cambios a todos los recursos adjudicados a un conjunto de cambios realizando la operación en el conjunto de cambios en su totalidad, y a continuación confirmando el conjunto de cambios en un paso.

Al tratar con recursos y sus cambios, vale la pena tener en cuenta los puntos siguientes en relación con el motor de cálculo:

- Al activar (confirmar) cambios en un recurso específico o en un grupo de recursos (conjunto de cambios) procure tener en cuenta qué parte del sistema se verá afectada. Dado que cambiar el módulo del recurso podría activar un recálculo, es importante optimizar la fecha de activación de los recursos y el número de cambios activados en una operación.
- **Actualización masiva**: es posible aplicar el mismo cambio a muchos recursos (actualización masiva). Dado que al cambiar el módulo de recurso podría tener que hacerse un recálculo, es importante optimizar esta operación.

El ejemplo anterior gestionaba un recurso de forma no directa, pero por la adjudicación lógica a su función o su ubicación (en este caso, a su función, un servidor de centro de datos).

La solicitud de registro podría ser complicada si se piden eventos para cada servidor individual incluido en el centro de datos. Un problema sería el número de recursos a los cuales se hace referencia. El otro sería que la infraestructura del centro de datos cambia regularmente, porque un servidor que formaba parte del centro de datos deje de estar allí, o porque se haya agregado un servidor nuevo. Por lo tanto, la lista tiene que ser dinámica.

Basándonos en el ejemplo anterior, está claro de que los recursos tienen que estar adjuntos a un grupo lógico para que se puedan dirigir mediante esta entidad lógica. Además, puede que el grupo lógico en sí deba gestionarse si está cambiando constantemente.

La adjudicación de recursos es el método de CA Business Service Insight de etiquetar recursos. Se puede adjudicar un recurso a uno o más grupos, tipos de recurso, partes contratantes o servicios. Las adjudicaciones de recursos se gestionan mediante el control de versión de CA Business Service Insight.

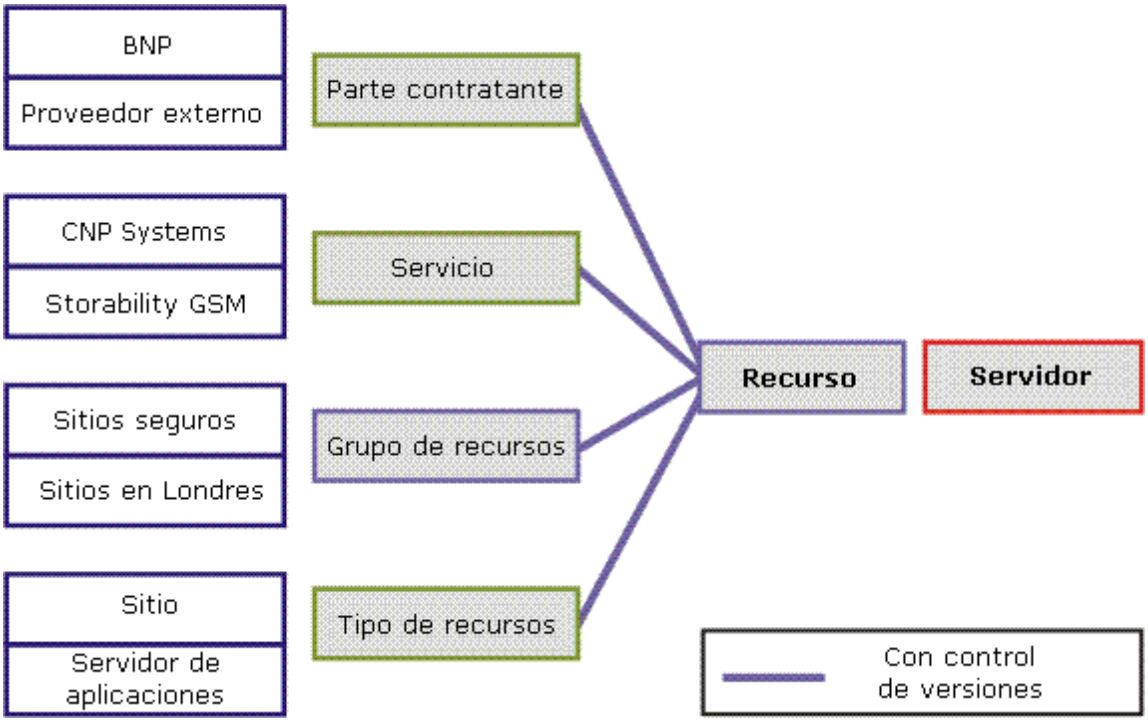
Los recursos disponibles para su inclusión en los cálculos se determinan según los recursos vigentes dentro del sistema (en relación con el intervalo de tiempo calculado en ese momento).

Volviendo ahora al ejemplo anterior:

Número global de tiempos de inactividad en servidores de centro de datos

El centro de datos se puede representar en el sistema como un servicio al cual se adjudican después todos los servidores dentro del centro de datos. Se puede definir también como un grupo de recursos llamado "servidores de centro de datos". Estos son dos métodos alternativos para elegir la adjudicación de recursos en este caso concreto, pero hay más opciones disponibles.

El diagrama siguiente muestra a qué entidades se puede adjuntar un recurso y sus usos lógicos.



Un grupo de recursos puede reflejar cualquier aspecto del recurso necesario para los cálculos, como su ubicación o la tecnología que contiene.

La finalidad principal de adjudicar recursos a estas entidades es garantizar tanto la correspondencia con los requisitos de cálculo como que el modelo permanece lo más dinámico posible.

Atributos personalizados

Otra función disponible para un recurso es la capacidad de proporcionar atributos personalizados. Cada tipo de recurso puede tener estos atributos personalizados agregados; y, a su vez, cada recurso definido como de ese tipo de recurso particular hereda estos atributos personalizados.

Mediante el ejemplo anterior, para cada uno de los servidores de centro de datos, se asocia también su dirección IP. Si cada uno de los servidores de centro de datos se define con el tipo de recurso de "servidor de centro de datos", asegúrese de que un atributo personalizado denominado "IP_Address" se agrega al tipo de recurso de servidor de centro de datos. De esta manera, cada recurso (servidor) se puede asociar con su dirección IP mediante el atributo personalizado.

Nota: Para obtener más información y ver un caso práctico, consulte Caso práctico: Uso de atributos personalizados (en la página 298).

¿Qué es una métrica en clúster?

Una métrica en clúster es una métrica definida para calcular niveles de servicio para un grupo de recursos. Los cálculos se hacen para cada uno de los recursos por individual, sin necesidad de duplicar la métrica cada vez con el registro de un recurso diferente.

La agrupación en clúster se puede hacer solamente en un grupo de recursos que no tenga ningún nivel de servicio de destino adjunto o que tenga el mismo nivel de servicio de destino. Por ejemplo, la disponibilidad de todos los servidores de aplicaciones debe ser como mínimo 99,9 % por servidor. En este caso, se agrupa en clúster una métrica en base a un grupo de recursos que contiene todos los servidores de aplicaciones. El motor calcula un resultado de nivel de servicio por separado para cada servidor, donde el destino para cada servidor es 99,9 %.

Además de este tipo de agrupación en clúster, CA Business Service Insight admite un tipo de agrupación en clúster por "acumulación", lo cual permite emitir informes en varios niveles de recursos (y grupos) a partir de una sola métrica. Esto permite calcular el nivel de servicio en varios niveles de la jerarquía de recursos, así como usar una función de obtención o reducción de detalles entre las entidades relacionadas de esta estructura de recursos. Desde la ficha de agrupación en clúster de las métricas, podrá activar esta opción seleccionando las opciones relevantes en esta página.

Construcción del modelo de datos del sistema

Como parte del proceso de modelado de datos, se identifican los componentes necesarios basándose en el origen de datos y los requisitos de cálculo.

A continuación se incluye una lista de los componentes que se deberán identificar en el proceso de modelado de datos y sus definiciones.

Nombre del evento El nombre del evento como aparece en CA Business Service Insight; deberá ser lo más descriptivo posible.

Comportamiento del evento	Comportamiento del evento especificado; cuándo se recibe desde el origen de datos, las condiciones, etc.
Campo Marca de tiempo	Campo en el origen de datos utilizado como marca de tiempo del evento.
Campo Tipo de evento	Campo en el origen de datos que se tiene que traducir a un tipo de evento que describe el tipo de informe. Es importante que el número de tipos de eventos diferentes sea el mínimo, ya que la definición del tipo de evento es manual y lo ideal es establecerla solamente una vez.
Campos de datos	Campos en el origen de datos que se tienen que recuperar como campos de datos.
Campo Recurso	Campo en el origen de datos que se tienen que traducir en un recurso. Contiene una entidad requerida para elaborar un informe con una duración relativamente fija. El recurso es una entidad con un ciclo de vida definido, cuyos cambios se pueden gestionar dinámicamente dentro del sistema. Al hacer referencia a un ciclo de vida del recurso se indica con qué frecuencia se agregan nuevos recursos y se cambian en su adjudicación a un servicio diferente o cualquier otra entidad de adjudicación, como se ha mencionado anteriormente. El campo que se debe traducir como un recurso en CA Business Service Insight deberá tener pocos cambios de adjudicación y otros limitados.

Y por último, en base a todas las definiciones anteriores:

Registro de	Define el criterio de registro. El criterio de registro define el tipo de evento y el recurso que registra la métrica. Los recursos se pueden solicitar directamente registrando por recurso o por entidad de adjudicación, como servicio, parte contratante, grupo de recursos, tipo de recurso, etc. Las capacidades de registro determinan esta definición. Otro método es hacer el registro por métrica, donde la métrica actual obtiene resultados (resultado del nivel de servicio) de otra métrica y los usa como entrada. También se puede utilizar el resultado de varias métricas como entrada.

Directrices para definir registros

Siga las directrices a continuación para definir registros:

- Nunca establezca el registro solamente por tipo de evento. Aunque el requisito de cálculo no requiera filtración de recursos, agregue el filtro de recursos como mínimo en el tipo de recurso.

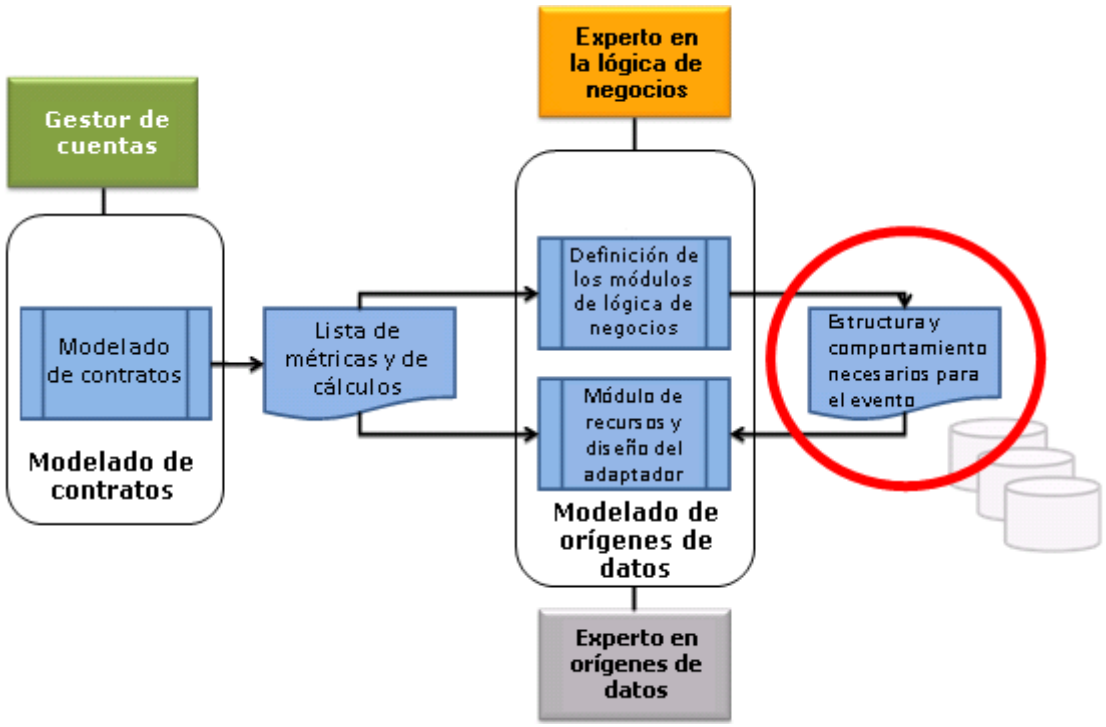
  Por ejemplo, cuando se calculen tiempos de respuesta globales promedio de servidores de aplicaciones, sólo se debe informar del evento de tiempo de respuesta cuando se asocie con uno de estos servidores de aplicaciones (es decir, cuando el tipo de recurso del recurso asociado con el evento sea "servidor de aplicaciones"). En tal caso, para distinguirlos entre sí el sistema puede tener otros tipos de recursos, como sitios o enrutadores, que utilicen el mismo tipo de evento para enviar datos. El tipo de recurso del que se debe informar ("servidor de aplicaciones") se deberá agregar al comando de registro como el 3er parámetro.

  La razón para esto es que cuando se produzca un cambio de recurso, el motor marque la métrica asociada con este recurso como que necesita recálculo. En un caso donde la métrica registre solamente por tipo de evento, el motor ve la métrica como registrada en todos los recursos, por lo que la marca para recalcularse al activarse cada recurso. Para evitar esta situación, se debe utilizar el 3er parámetro "opcional" del tipo de recurso.

- El método más eficaz para la tarea de registro es por Servicio y parte contratante. Organizar los recursos en esta manera permite expresar la relación lógica entre la capa de datos y la capa de negocio en el sistema. Al registrar recursos a través de estas entidades no es necesario hacer cambios en las fórmulas cuándo se usan en contratos diferentes o para servicios diferentes. El servicio y el contrato de contexto de métrica definen el servicio y parte contratante. Las fórmulas de lógica de negocios definidas en este tipo de registro son fácilmente reutilizables, ya que no requieren cambios en el registro.

Nota: La ficha *Registro* puede gestionar también registros dentro de cada métrica. Esta interfaz proporciona un asistente que le guía a través del proceso para la métrica seleccionada.

Resultados de la etapa de modelado de datos (experto en el origen de datos y experto en la lógica de negocios)



Para continuar con la etapa de implementación de la solución, es necesario contar con el experto en el origen de datos para garantizar que se cumple lo siguiente:

- Se comprenden totalmente los orígenes de datos con los que se tiene que tratar, incluidos:
  - Datos históricos de muestra
  - Método de comunicación
  - Se comprenden los campos clave de cada origen de datos que se tienen que utilizar para componentes clave del modelo de recurso.
- Para cada origen de datos se incluye una lista inherente de tipos de eventos que describen la estructura y el comportamiento, incluidas las definiciones siguientes:
  - Nombre del evento
  - Comportamiento del evento
  - Campo Marca de tiempo
  - Campo Tipo de evento
  - Campos de datos
  - Campo Recurso

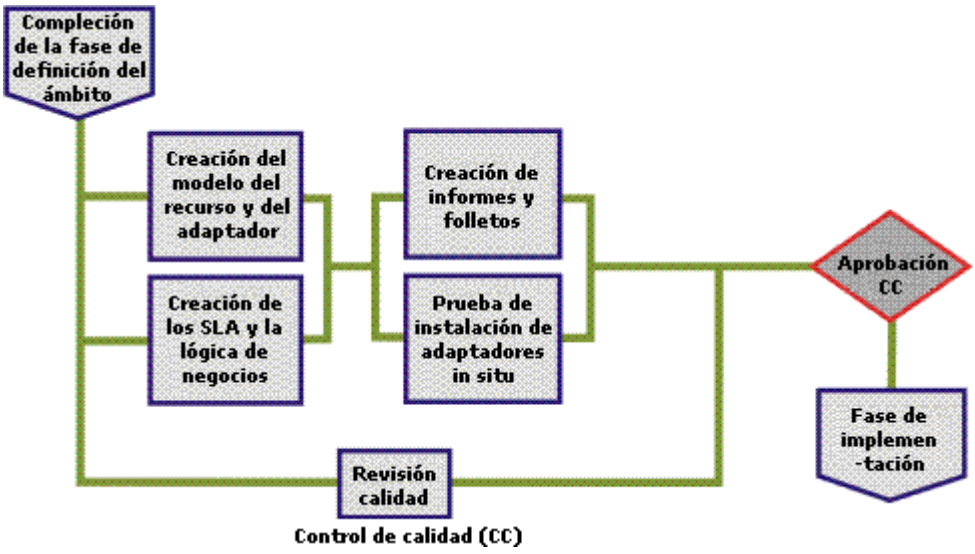
Nota: Deberá crear un documento de resumen de la información obtenida.

Capítulo 3: Implementación

Esta sección contiene los siguientes temas:

Implementación: introducción (en la página 69)
Configuración del marco (gestor de contratos) (en la página 72)
Configuración de la biblioteca de plantillas (gestor de contratos) (en la página 73)
Cómo crear contratos (gestor de contratos) (en la página 74)
Recolección de datos (experto en el origen de datos) (en la página 82)
Scripting de lógica de negocios (experto en la lógica de negocios) (en la página 153)
Cómo activar los contratos (gestor de contratos) (en la página 197)
Creación de entregas (gestor de contratos) (en la página 201)

Implementación: introducción



Este capítulo explica el proceso y el razonamiento detrás de la fase de implementación del proyecto. Como se muestra en la ilustración anterior, la fase de implementación sigue a la fase de diseño, y después le sigue la fase de instalación e implantación.

El objetivo de la fase de implementación es que el gestor de contratos pueda crear todos los elementos y los objetos en el sistema de CA Business Service Insight que se hayan definido previamente durante la etapa de diseño. Durante la fase de implementación, el equipo se dedica a preparar el sistema para una implementación e instalación total.

Esta fase no se deberá iniciar antes de que la fase de diseño haya finalizado y se hayan gestionado y definido correctamente todos los objetivos necesarios. Si la fase de diseño no ha finalizado correctamente o no se han definido claramente todos los contratos, métricas, adaptadores, etc., habrá problemas con el sistema o con los elementos que falten y que deberían haberse implementado durante la fase de implementación. La fase de implementación se deberá iniciar solamente después de verificar la revisión del diseño.

Es también importante que la fase de implementación se complete correctamente antes de continuar con la instalación e implementación del sistema. Esto se deberá hacer solamente después de haber realizado una revisión de calidad.

El proceso de configuración implica que el gestor de contratos realice los pasos siguientes:

- Configuración del catálogo de servicios
- Creación de contratos
- Activación de los contratos
- Configuración de las opciones de seguridad
- Creación de informes/alertas/cuadros de mandos

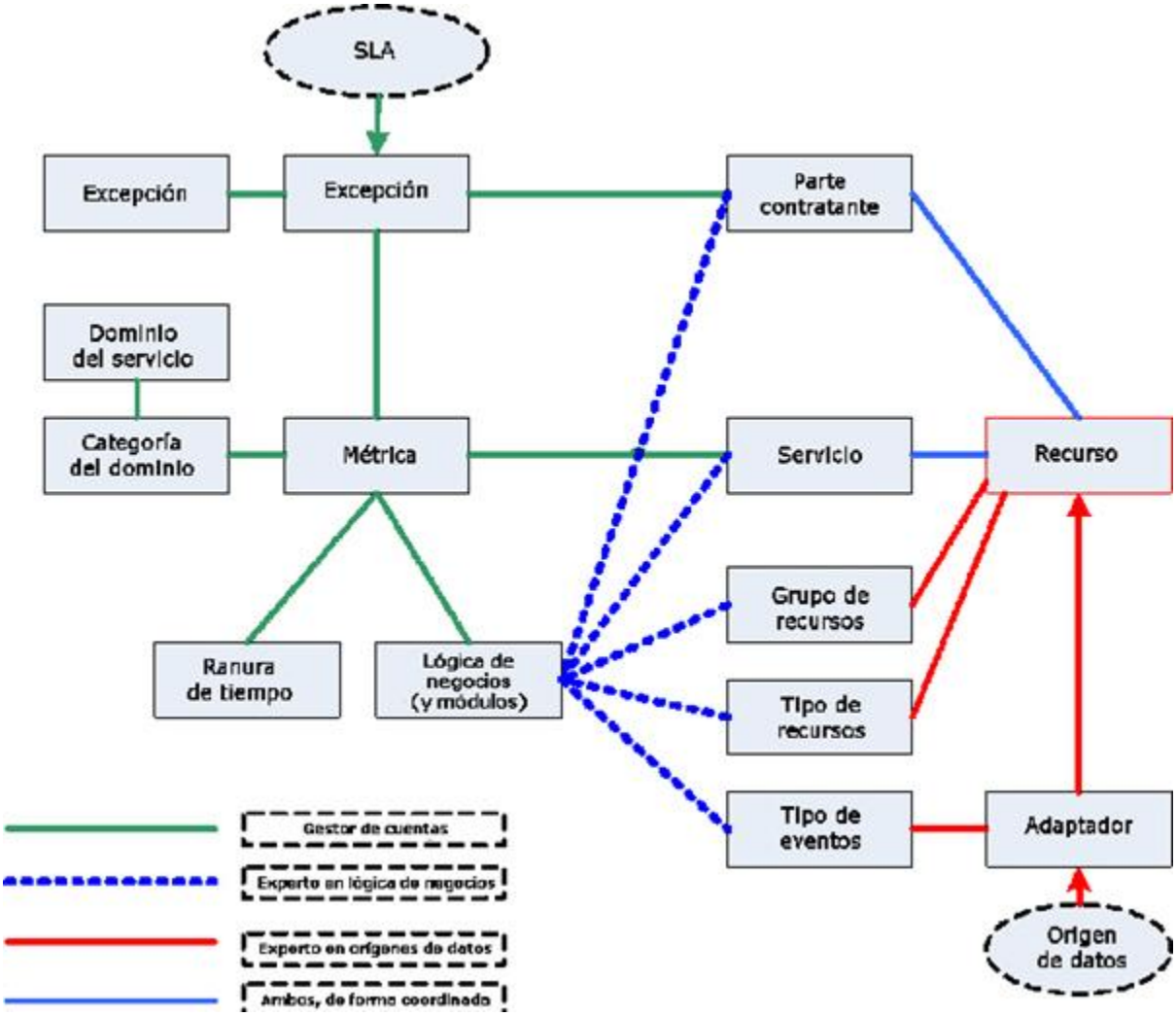
El experto en el origen de datos deberá configurar el adaptador e integrarlo con los orígenes de datos. El experto en el origen de datos debe realizar también el paso de traducción de recursos para vincular las estructuras de orígenes de datos con las entidades definidas dentro del sistema de CA. Este paso es crucial para el proceso en su totalidad y puede requerir también un poco de orientación por parte del gestor de contratos.

Además de esto, el experto en la lógica de negocios tiene que escribir la lógica de negocios para todas las métricas, según los planes de la fase de diseño. Esto puede incluir la creación de todos los módulos obligatorios y la configuración de los parámetros asociados para proporcionar las funciones de cálculo esperadas.

Todos los puntos anteriores se explican en mayor profundidad en las secciones de este capítulo.

Nota: Es importante que el gestor de contratos sea consciente de que una elección pobre en esta etapa puede afectar negativamente al funcionamiento de CA Business Service Insight y puede hacer difícil o imposible retroceder en una etapa posterior.

La ilustración siguiente esboza el flujo de trabajo lógico global.



Configuración del marco (gestor de contratos)

El marco permite:

- Definir servicios, grupos de servicios, dominios de servicio y unidades.
- Crear y mantener plantillas, incluidas plantillas de lógica de negocios y de ranuras de tiempo y módulos de lógica de negocios.
- Gestionar atributos personalizados para los tipos de recurso.

En esta etapa, todas las entidades del sistema identificadas durante la fase de diseño se crean en la sección de marco de la aplicación. Solamente cuando el sistema contiene estas entidades de marco es posible continuar con la creación de contratos y sus métricas relacionadas.

Construir el marco permite agregar los elementos nuevos a continuación:

- Servicios
- Grupos de servicios
- Categorías de dominio y dominios de servicio
- Unidades de medida
- Plantillas de ranuras de tiempo
- Partes contratantes
- Atributos personalizados

Para obtener más información acerca de cada uno de los elementos anteriores, consulte la ayuda en línea.

Configuración de la biblioteca de plantillas (gestor de contratos)

Las bibliotecas de plantillas permiten definir y gestionar lo siguiente:

- Bibliotecas de plantillas
- Carpetas de plantillas
- Plantillas de nivel de servicio
- Plantillas de contrato
- Opciones de seguridad para derechos de acceso de usuario

Para obtener más información acerca de cada uno de los elementos anteriores, consulte la ayuda en línea.

Cómo crear contratos (gestor de contratos)

En esta etapa, se crean en el sistema los contratos y sus entidades relacionadas, los cuales se definieron en la fase de diseño.

Siga estos pasos:

1. Agregue un nuevo contrato y aplique sus detalles generales.
2. Para cada contrato, defina sus métricas y aplique sus detalles generales.

En esta etapa solamente se aplican los detalles generales del contenido de un contrato, sin incluir la lógica de negocios ni la agrupación en clúster de las métricas del contrato.

La descripción a continuación de estos pasos enfatiza algunos puntos importantes que se deberán tener en cuenta en esta etapa. Estos pasos se describen en detalle en la ayuda en línea.

Paso 1: Agregue un nuevo contrato y aplique sus detalles generales.

La definición del contrato deberá incluir:

- Establecimiento del nombre del contrato.
- Selección de las partes contratantes relacionadas.
- Anexión de los servicios relacionados.
- Establecimiento de las fechas de vigencia del contrato. Las fechas de vigencia de los contratos corresponden al intervalo de fechas para el cual el motor de correlación calculará el nivel de servicio para el contrato; los resultados para elaboración de informes estarán disponibles solamente para estas fechas. Al fijar las fechas, es importante tener en cuenta los requisitos en cuanto a disponibilidad de los informes relacionados con el contrato y los datos sin procesar disponibles.
- Establecimiento de la zona horaria del contrato y la moneda. Esta definición obedece a fines de elaboración de informes y permite que los informes en el contrato sigan la zona horaria relevante. La definición de la moneda permite al motor de elaboración de informes decidir en qué moneda mostrar los resultados de sanción en el caso de fórmulas de sanción.

Paso 2: Agregue los detalles generales de las métricas.

Una vez tenemos el contrato es hora de crear las métricas incluidas. Al definir las métricas se deberán realizar los pasos siguientes:

- Establecer el nombre de las métricas.

- Aplicar los detalles de las métricas (servicio, dominio, unidad, ranura de tiempo, zona horaria, etc.).
- Establecer los umbrales del cuadro de mandos (consulte Configuración de las páginas del cuadro de mandos (en la página 215)).
- Adjuntar las métricas relacionadas (si es oportuno) y la relación entre ellas.
- Definir la granularidad con la que el motor calcula la métrica.
- Establecer la declaración del objetivo y los parámetros.

En esta etapa todavía faltan los elementos siguientes en la definición de las métricas:

- Fórmula/módulo de la lógica de negocios y registro
- Definición de agrupación en clúster

Estos elementos se incluyen en las métricas del contrato solamente después de haber creado la infraestructura del sistema y después de haber desarrollado y probado las fórmulas de la lógica de negocios.

Nota: Un método alternativo al anterior es desarrollar primero las plantillas de nivel de servicio dentro del sistema, en lugar de definir los contratos directamente. Esto permite crear la plantilla y después utilizarla para crear otros contratos. En algunos casos, se pueden importar plantillas de nivel de servicio que ya existan en el sistema desde otra instancia de CA. Esto puede permitir un inicio más rápido que al crear los contratos desde cero. Para ver más información acerca de cómo crear sus propias plantillas de nivel de servicio, consulte Creación de plantillas de nivel de servicio (en la página 78).

En algunos casos, es aconsejable crear un contrato de muestra al principio, para comprobar que todo esté funcionando correctamente y según lo previsto. A continuación es posible crear las plantillas de nivel a partir de este contrato y almacenarlas en el catálogo de servicios, lo cual proporciona un sólido punto de partida para todos los contratos en el sistema.

Creación de contratos a partir de un servicio

Crear contratos en el sistema mediante un catálogo de servicios aplicable a muchos contratos diferentes, bien desde una plantilla o sin plantilla (basándose en varias plantillas de nivel de servicio) es mucho más eficiente y ofrece una reutilización mucho mayor. En general, las plantillas de nivel de servicio contienen una recolección de métricas predefinidas aplicables a ciertos componentes del servicio. Si es necesario, se puede tener también más de un servicio (y métricas asociadas) en una sola plantilla de nivel de servicio. Generalmente el contenido de una plantilla de nivel de servicio se define según su uso, y puede variar según distintos requisitos.

Por ejemplo, imaginemos una organización que ofrece un servicio de alojamiento de aplicaciones. La organización puede ofrecer a sus clientes tres paquetes de servicio diferentes, como Bronce, Plata y Oro, dependiendo de lo que se incluye en el paquete. Un buen ejemplo en cuanto al uso de plantillas de nivel de servicio podría ser crear una para cada uno de los paquetes.

Una vez definidas, estas definiciones se pueden utilizar para crear un nuevo contrato de cliente muy eficazmente. Por ejemplo, el cliente ABC decide inscribirse a un paquete de alojamiento de aplicaciones Oro. Podrá crear este nuevo contrato en el sistema directamente utilizando la plantilla de nivel de servicio como sigue:

En la página de contratos, haga clic Agregar nuevo/a, Mediante el catálogo de servicios, y seleccione Basado en plantilla o Sin plantilla. A continuación siga el asistente del contrato para terminar de crear el contrato. Si elige Basado en plantilla, a continuación tendrá que especificar los valores de configuración de la plantilla.

El asistente del contrato muestra una lista de plantillas de nivel de servicio y se puede especificar qué plantillas de nivel de servicio incluir en el contrato. Tenga presente que es posible seleccionar métricas específicas a partir de distintas plantillas de nivel de servicio, o sólo seleccionar la definición completa para incluirla. En este ejemplo, todas las métricas del paquete de alojamiento Oro. Tenga en cuenta que al seleccionar el nivel superior automáticamente se seleccionan todos los nodos secundarios. Sepa también que es posible asignar las métricas a un servicio diferente. Sin embargo, el valor predeterminado es mantenerlas asignadas a los mismos componentes de servicio que en la definición.

Una vez seleccionadas todas las métricas necesarias, haga clic en el botón Siguiente para transferir esas métricas a la nueva plantilla de contrato e indicar el nombre del contrato y los detalles generales. Haga clic Guardar y Siguiente para crear el contrato.

Una vez terminado esto tiene las opciones siguientes:

- Continuar con el asistente del contrato, definir parámetros y ejecutar el asistente de métrica, el cual abre la interfaz de asistente para que pueda personalizar las métricas de los contratos. El asistente permite revisar y modificar cada una de las métricas cambiando los campos disponibles, como los parámetros de métricas en la declaración del objetivo, el nombre de la métrica, la ranura de tiempo y la descripción. Una vez finalizado este proceso para cada métrica, el asistente le devuelve a la página de métricas del contrato, dónde podrá cerrar y guardar el nuevo contrato.
- Abra la página del contrato para consultarlo o editarlo.

Creación de plantillas de nivel de servicio

Crear una plantilla de nivel de servicio es un proceso bastante sencillo. Desde un contrato existente (en la página Detalles del contrato), seleccione todas las métricas que desee incluir y seleccione Guardar como plantilla de nivel de servicio.

La ventana siguiente requiere que dé un nombre a la plantilla de nivel de servicio. A continuación se podrá guardar la plantilla de nivel de servicio. Una vez guardada, todas las ranuras de tiempo asociadas a las métricas seleccionadas se incluyen en la ficha Ranura de tiempo. Desde aquí puede ser necesario personalizar aún más la plantilla de nivel de servicio para hacerla totalmente flexible. Esto podría incluir agregar parámetros a las métricas y exponer estos parámetros mediante la declaración del objetivo para cada métrica. También quizás crear parámetros de nivel de plantilla de nivel de servicio (similares a los parámetros de contrato) a los cuales podrán después hacer referencia algunas de las métricas o todas ellas. Una vez terminada, la plantilla de nivel de servicio estará disponible para implementarse en otros contratos.

Ciclo de vida del contrato y control de versiones

Una vez se ha terminado de configurar un contrato, se debe confirmar. Esta acción permite al motor de cálculo empezar a calcular los niveles de servicio para todas las métricas en el contrato. Al confirmar un contrato su estado cambia de "preliminar" (editable) a "vigente" (no editable). Para realizar otros cambios en este contrato, se deberá crear una nueva versión del contrato. Si se eligen las mismas fechas de vigencia para la nueva versión, una vez se terminan de hacer los cambios y se confirma la nueva versión, ésta sobrescribe totalmente la versión anterior. Esto también hace que el motor vuelva a calcular las métricas diferentes con respecto a la versión anterior. Las versiones vigentes también pueden solaparse parcialmente, por ejemplo cuándo se hace un cambio en los destinos de algunas métricas del contrato en mitad de las fechas de vigencia. En este caso, la versión antigua se seguirá utilizando hasta que la fecha de vigencia de la segunda versión se vuelva activa. En este momento, la segunda versión asume el estado vigente para los cálculos.

La tabla siguiente muestra cómo afectan los cambios del usuario al ámbito de impacto de recálculos y al marco de tiempo. Un cambio puede afectar a una métrica específica dentro de una versión concreta del contrato o puede afectar a métricas en diferentes contratos y versiones.

Cambiar	Ámbito de impacto	Marco de tiempo de impacto
Cambios realizados en métricas		
Cambio de detalles de métricas: fórmula de la lógica de negocios	Afecta a todas las métricas dentro de una versión específica de contrato.	Desde el inicio de la versión de contrato vigente.
Cambio de detalles de las métricas: valor de destino	Afecta a todas las métricas dentro de una versión específica de contrato.	Desde el inicio de la versión de contrato vigente.
Cambio de detalles de las métricas: período de seguimiento	Afecta a todas las métricas dentro de una versión específica de contrato.	Desde el inicio de la versión de contrato vigente.
Cambio de detalles de las métricas: parámetros de métrica	Afecta a todas las métricas dentro de una versión específica de contrato.	Desde el inicio de la versión de contrato vigente.
Cambio de detalles de las métricas: zona horaria	Afecta a todas las métricas dentro de una versión específica de contrato.	Desde el inicio de la versión de contrato vigente.

Cambio de detalles de las métricas: agrupación en clúster	Afecta a todas las métricas dentro de una versión específica de contrato.	Desde el inicio de la versión de contrato vigente.
Cambio de detalles de las métricas: fechas de vigencia	Afecta a todas las métricas dentro de una versión específica de contrato.	Desde el inicio de la versión de contrato vigente.
Cambio de detalles de las métricas: ranuras de tiempo	Afecta a todas las métricas dentro de una versión específica de contrato.	Desde el inicio de la versión de contrato vigente.
Cambio de parámetros de nivel de contrato	Afecta a todas las métricas dentro de una versión específica de contrato.	Desde el inicio de la versión de contrato vigente.
Cambio de módulos SLALOM	Afecta a todas las métricas adjuntas al módulo slalom modificado, en todos los contratos y las versiones del contrato.	Desde el inicio de la versión de contrato vigente.
Operaciones generales		
Cambio del modelo de recurso de métrica/conjunto de cambios (CA Business Service Insight 4.0)	Afecta a todas las métricas que registran el recurso, en todos los contratos y las versiones del contrato.	Desde el estado más próximo antes de la fecha en la cual se produjo el cambio en el recurso.
Obtención de eventos retrasados para la métrica Eventos con marca de tiempo vencida (datos sin procesar o eventos de reutilización)	Afecta a todas las métricas que usan el recurso, en todos los contratos y las versiones del contrato.	Desde el estado más próximo antes de la fecha en la cual se produjo el cambio en el evento.
Adición de datos de correcciones de métricas	Afecta a todas las métricas que usan el recurso, en todos los contratos y las versiones del contrato.	Desde el estado más próximo antes de la fecha en la cual se produjo el cambio en la corrección.

Cambio, actualización o supresión de horas de excepciones de métricas (activación y desactivación)	Dependiendo de los valores de configuración de excepción y la implementación específica, afecta a métricas concretas dentro de la versión de contrato particular o puede afectar a métricas presentes en varios contratos y versiones de contrato.	Lo más próximo posible a la hora de la excepción.
Actualización de atributo personalizado	Afecta a todas las métricas que registran el recurso, en todos los contratos y las versiones del contrato.	Desde el estado más próximo antes de la fecha en la cual se produjo el cambio en el recurso.

Por último, algunas cuestiones clave que recordar sobre las versiones del contrato:

- Si la nueva versión tiene la misma fecha de vigencia, solamente se volverán a calcular las métricas que hayan cambiado, y se volverán a calcular desde el inicio de la versión.
- Si la nueva versión tiene fechas diferentes, todas las métricas de la nueva versión se calcularán desde el inicio de esa versión, y todas las métricas en la versión anterior se volverán a calcular desde algún punto en esa versión y hasta que la nueva versión pase a ser válida. La cantidad exacta de recálculo depende de la configuración de los estados.
- Se recomienda crear contratos con fechas de vigencia para un año y renovarlos cuando caduquen. Esto evita períodos de recálculo para más de un año.
- Se calculan versiones no vigentes (la fecha actual es posterior a las fechas de vigencia del contrato finales) de los contratos (y por consiguiente siguen contándose como métricas activas en el sistema, ya que incluyen datos del nivel de servicio calculados asociados para elaboración de informes).

Los valores asociados con variables globales dentro de las métricas no se pasan a otras versiones del contrato (es decir, se invoca a la rutina OnLoad en la lógica de negocios al inicio de cada versión del contrato).

Nota: Para ver distintos escenarios y casos prácticos, consulte Casos prácticos de modelado de contratos (en la página 267).

Recolección de datos (experto en el origen de datos)

En la fase de recolección de datos del proceso de implementación trabajará con adaptadores. Los temas siguientes describen ese proceso.

Funcionalidad del adaptador

Los adaptadores son módulos encargados de recopilar datos de los orígenes de datos y pasarlos al sistema de CA Business Service Insight. Los adaptadores filtran los datos que llegan del origen de datos y los manipulan de manera que cuando lleguen al sistema contengan solamente la información necesaria para realizar los cálculos de nivel de servicio en la estructura correcta.

La plataforma del adaptador proporciona flexibilidad para:

- Recibir datos en línea o sin conexión con cualquier frecuencia.
- Recibir datos en diversos niveles: sin procesar, calculados o agregados.
- Recibir datos desde una amplia gama de tipos de herramientas.

Los adaptadores están formados fundamentalmente por dos componentes:

- **Componente de adaptador genérico**:

  Hay dos tipos de componentes de adaptador genérico, un componente de adaptador de archivos ASCII y un componente de adaptador de SQL basado en ODBC. Estos componentes permiten conectarse a un origen de datos y analizarlo como un archivo ASCII o ejecutar en él una consulta SQL.

- **Archivo de configuración del adaptador**:

  Todos los adaptadores necesitan un archivo de configuración para saber dónde y cuándo se deben conectar, qué deben recuperar y cómo transformar y traducir los datos en transacciones y eventos de CA genéricos. CA Business Service Insight proporciona a todos los tipos de adaptador genéricos un archivo de plantilla de configuración XML predeterminado que se puede ajustar para representar datos específicos de cliente en cuanto al origen de datos al que se tienen que conectar. El archivo de configuración XML define qué campos se deberán recuperar, cómo se deberán identificar, cómo se deberán convertir al sistema de la base de datos normalizada, etc.

  Nota: La interfaz de usuario incorpora un asistente del adaptador que permite personalizar de forma básica esta plantilla XML en línea. Esto cumple el mismo objetivo de crear el archivo de configuración XML para el adaptador. Más adelante en este capítulo podrá ver más detalles sobre esta función.

La plataforma de adaptador incluye un mecanismo de reinicio/recuperación y puede encargarse de los problemas en datos recibidos de herramientas de terceros, como bloqueos de herramientas, problemas de red, datos perdidos, datos duplicados, datos basura, vacíos en los datos, validación de datos, etc. Todos los adaptadores proporcionan capacidad de integración total de datos y seguimiento integral, así como registro de todos los mensajes del adaptador. Esto se detalla más adelante en esta guía.

Los adaptadores de CA Business Service Insight pueden ejecutarse como servicio o como aplicación (visible o no visible). La tecnología de adaptador de CA Business Service Insight admite avanzados mecanismos de seguridad, como cifrado, protocolo de enlace y procesos de autenticación.

En esta etapa es importante ser consciente de que el asistente del adaptador es un mecanismo para automatizar los procesos y las tareas siguientes. Si bien algunos elementos mencionados pueden no siempre estar visibles al utilizar el método del asistente, seguirán estando presentes "entre bastidores" en la interfaz del asistente.

Entorno del adaptador

Las entidades siguientes están relacionadas con el adaptador y sus parámetros de configuración y ejecución.

Origen de datos**:**

El origen de datos al cual se conecta el adaptador y del cual recupera datos en su formato original.

Archivos de trabajo**:**

Los archivos de salida que produce el adaptador y que se escriben dentro de los procesos (para ver más detalles, consulte Archivos de trabajo (en la página 88)).

Escucha del adaptador de CA Business Service Insight**:**

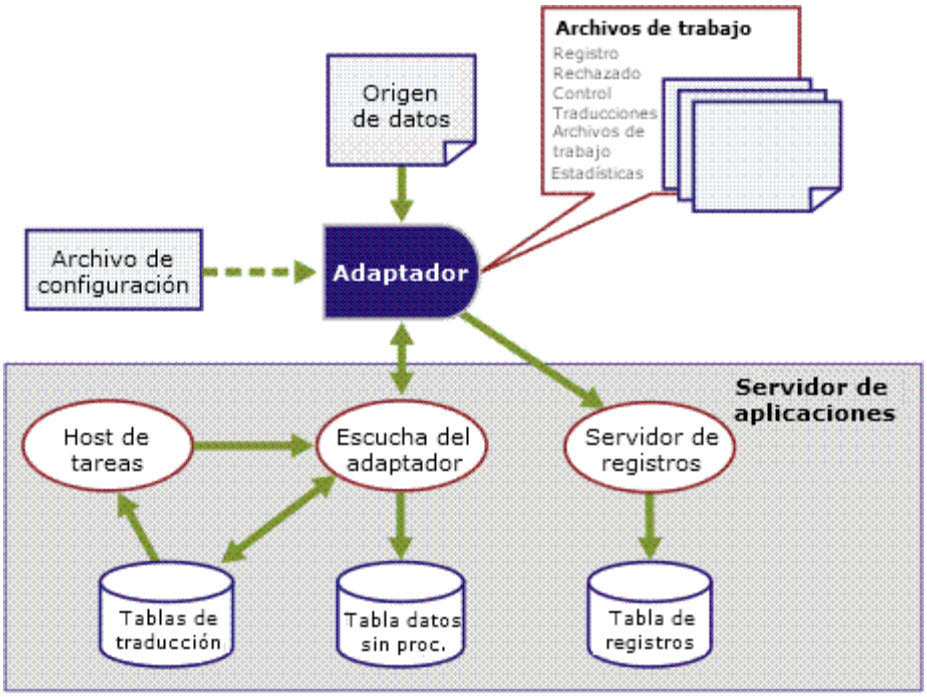
Entre el adaptador y la escucha del adaptador se transfieren tres tipos de mensajes:

- **Control**: mensajes de inicio\detención\pausa enviados por la escucha al adaptador y que se vuelven a enviar cuando el adaptador cambia su estado.
- **Traducciones**: el adaptador envía solicitudes del contenido de la tabla de traducción y solicitudes de valores de traducción específicos. La escucha devuelve las tablas y el valor traducido. La escucha del adaptador recibe indicación del host de la tarea de que se ha traducido una entrada de traducción. A continuación envía el mensaje al adaptador.
- **Datos sin procesar**: eventos de datos sin procesar unificados que envía el adaptador. Estos eventos se envían en paquetes e incluyen reconocimientos.

Servidor de registros de CA Business Service Insight

Se puede configurar el adaptador para enviar mensajes de registro al registro del sistema así como para escribirlos en un archivo local. Si se especifican y definen el puerto y la dirección de IP del servidor de registros en los valores de configuración de registro del adaptador, a continuación el adaptador envía también mensajes al servidor de registros.

El diagrama siguiente describe el proceso del adaptador en relación con cada una de las entidades con las cuales interactúa.


Archivos de trabajo
Registro
Rechazado
Control
Traducciones
Archivos de trabajo
Estadísticas
Origen de datos
Archivo de configuración
Adaptador
Servidor de aplicaciones
Host de tareas
Escucha del adaptador
Servidor de registros
Tablas de traducción
Tabla datos sin proc.
Tabla de registros

La siguiente es una descripción de la interacción del proceso del adaptador con estas entidades:

Archivo de configuración**:**

Contiene valores de configuración para algunos de los parámetros de configuración del adaptador o todos ellos. El adaptador utiliza el archivo de configuración para determinar el método de conexión que utilizan el adaptador y las métricas para realizar el análisis correspondiente para crear la salida del evento. Este es un archivo XML y su formato contiene seis elementos básicos:

General:

Diversos atributos de adaptador (directorio de trabajo, archivos de salida e indicador de depuración).

OblicoreInterface:

Atributos de conexión con el servidor de CA Business Service Insight.

DataSourceInterface:

Atributos de conexión con el origen de datos (ruta de archivos y patrón, cadenas de conexión, consultas de SQL, etc.).

InputFormatCollection:

Métricas de análisis y manipulación de los datos de origen.

TranslatorCollection:

Métricas para construir el evento unificado compuesto de los campos de datos analizados y manipulados.

TranslationTableCollection:

Métricas para asignar datos entre datos originales y entidades de CA Business Service Insight.

Cada una de estas seis secciones contiene toda la información relevante que permite al adaptador conectarse al origen de datos, recuperar la información necesaria, analizarla en estructuras de eventos unificadas de CA Business Service Insight y almacenarla dentro de la tabla de datos sin procesar de CA Business Service Insight.

Archivos principales

El adaptador está formado por dos archivos principales: el archivo ejecutable y los archivos de configuración. El archivo de configuración es un archivo de texto. Hay dos archivos ejecutables: el adaptador de SQL y el adaptador de archivos.

Un archivo de configuración de XML se adapta para cada adaptador y guardar así los requisitos del origen de datos específico. El archivo de configuración especifica la información relacionada con el origen de datos (nombre, ubicación, método de conexión y estructura) y la estructura de los eventos de salida que debe generar el adaptador.

El archivo de configuración incluye esos parámetros y los valores que se establecen para los atributos dentro de un archivo de XML de estructura predefinida.

Al crear un adaptador nuevo, es necesario usar el archivo ejecutable relevante existente (según el tipo de origen de datos de destino, el archivo para orígenes de datos de archivos fijos, SQL para orígenes de datos de base de datos) y, a continuación, modificar el archivo de configuración según sea necesario. Las dos estructuras contienen elementos de configuración un poco diferentes para un texto o adaptador de SQL. Por lo general, se configura automáticamente cuando se crea el adaptador mediante la utilidad Gestor del adaptador

Otros archivos relacionados con los adaptadores son los archivos de trabajo creados por el adaptador en el proceso de lectura del origen de datos y escritura de eventos al sistema CA Business Service Insight.

Nota: Para obtener más información sobre la modificación del archivo de configuración, consulte Especificaciones de configuración del adaptador (en la página 369).

Archivos de trabajo

Los archivos de trabajo del adaptador se crean cuando el adaptador se ejecuta por primera vez y se actualizan constantemente cada vez que se ejecuta el adaptador.

Cada adaptador tiene sus propios archivos de trabajo. Los nombres de los archivos de trabajo se pueden establecer en el archivo de configuración del adaptador (opcional) o pueden conservar sus nombres predeterminados. La ubicación del archivo de trabajo se establece a través de la "carpeta de trabajo" y se puede establecer también en el archivo de configuración. Tenga en cuenta que la ruta especificada puede ser relativa al directorio actual donde se encuentra el adaptador. La ruta especificada debe existir ya (o la deberá crear) para que el adaptador se ejecute correctamente.

Nota: La carpeta no se crea automáticamente si no existe.

Todos los parámetros relevantes en el archivo de configuración están dentro de la sección General. Solamente la ubicación del archivo de registro se define en el registro o se pasa mediante la línea de comandos.

AdapterStatistics.txt

El adaptador escribe información de estadísticas en este archivo en intervalos de un minuto. La última línea se escribe cuando se detiene el adaptador. Cada línea en el archivo contiene los números de:

- eventos recibidos;
- eventos ignorados;
- eventos con errores;
- mensajes enviados;
- paquetes enviados.

El adaptador inicia las estadísticas cada vez que se ejecuta.

RejectedEvents.txt

Este archivo contiene todos los eventos que el adaptador no pudo enviar a CA Business Service Insight debido al valor del evento, definido como de traducción obligatoria o que no tiene ID coincidente en la tabla de traducción. Esto significa que no se ha realizado la traducción relevante. En el archivo rejectedEvents se escribe cada evento que tiene como mínimo un valor en espera de traducción.

Al inicio de cada ejecución, el adaptador primero intenta enviar a CA Business Service Insight los eventos desde el archivo rejectedEvents al tiempo que intenta encontrar el ID coincidente para cada valor relevante en la tabla de traducción. Si se encuentra el valor, el adaptador envía el evento y lo suprime del archivo. Si no se encuentra un valor coincidente, el evento queda en el archivo rejectedEvents.

Es posible configurar el límite superior para el número de eventos rechazados que se puede alcanzar estableciendo el parámetro RejectedEventsUpperLimit en el archivo de configuración del adaptador. Cuando se alcanza el límite, el adaptador deja de leer nuevos registros e introduce un estado de "bloqueado". Esto se puede ver cuando la salida de depuración se muestra en la pantalla durante la ejecución del adaptador. Si ve una cadena continua de letras "B" en mayúsculas, entonces el adaptador se ha bloqueado y requiere que algunas de las entradas pendientes se traduzcan antes de cargar más datos.

Los eventos pendientes se escriben en el archivo en formato XML. A continuación se incluye un ejemplo de un evento del archivo:

```
<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>
```

Registro del adaptador

El registro del adaptador es el archivo donde el adaptador escribe los mensajes de registro.

Se recomienda examinar el archivo de registro del adaptador mediante la utilidad del explorador de registros.

Se puede establecer el nivel de elaboración de informes en este archivo de registro modificando un parámetro en el archivo de configuración del adaptador, LogDebugMode. Cuándo se establece en "yes", el adaptador escribe mensajes de indicación normales en el registro, así como el registro original, los resultados del análisis y el evento de destino.

Este parámetro normalmente se define afirmativamente en la fase de prueba y control del adaptador.

De forma predeterminada, el tamaño de archivo se limita a 1 MB. Cuando se alcanza este límite, el adaptador cambia su nombre para añadir "_old" y crea un nuevo archivo de registro. El adaptador puede almacenar hasta 2 MB de mensajes de registro; 1 MB para el archivo antiguo y 1 MB para el archivo en uso.

El límite de tamaño del archivo de registro se puede configurar como una entrada en el registro para cada adaptador, con un máximo de 10 MB. La entrada en el registro se llama LogFileMaxSize y se define bajo el adaptador específico con un valor múltiplo de kB.

DataSourceControl.xml

El adaptador utiliza el archivo DataSourceControl.xml para controlar su acceso al origen de datos y garantizar que, cuando se ejecute, siempre continúe hacia adelante desde el último punto leído.

El adaptador de archivos conserva el nombre de la última lectura, la última línea leída y la posición alcanzada en el archivo. La siguiente vez que se ejecute el adaptador, accederá al archivo desde la ubicación mediante la información incluida en el archivo DataSourceControl.xml. Mediante este mecanismo, el adaptador puede leer sólo nuevos registros cada vez que se ejecuta.

El adaptador no trabaja directamente en los archivos de origen, sino que primero copia el archivo actual en un archivo de trabajo. Por consiguiente, se guarda la misma información para el archivo de trabajo y para el archivo de origen. Si se adjunta el archivo de origen, sólo se copian los registros nuevos en el archivo de trabajo.

Si el adaptador se ha configurado para suprimir el archivo de origen una vez procesado, al establecer el parámetro en el archivo de configuración DeleteAfterProcessing en "yes", no se guardará la información en el archivo de origen. Cuando termine, leerá todos los archivos nuevos en la carpeta de trabajo que coincidan con el patrón de archivo definido en el archivo de configuración.

Solamente cuando DeleteAfterProcessing se establezca en "no" se buscarán registros nuevos en el último archivo. Si no hay ninguno, pasará al archivo siguiente en orden lexicográfico. Por lo tanto, al nombrar los archivos de los datos de origen, para garantizar que se lean en la secuencia correcta, intente imponer un patrón de denominación de aumento secuencial. Por ejemplo, para que esto sea posible añada los archivos con un valor de fecha en formato inverso (aaaammdd-hhmmss). Por ejemplo:

- DataSourceABC20070517-14:00:00.csv
- DataSourceABC20070517-15:30:00.csv
- DataSourceABC20070517-17:00:00.csv, y así.

A continuación sigue un ejemplo de DataSourceControl.xml para un adaptador de archivos.

```
<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>
```

Un adaptador de SQL conserva el último valor de los campos clave de consulta para cada consulta ejecutada. Los campos clave son identificadores únicos para registros en la tabla de base de datos de destino. El adaptador utiliza estos valores al construir la consulta para la ejecución siguiente. Esto permite al adaptador leer sólo registros nuevos.

Por ejemplo, supongamos que usamos la declaración SQL siguiente para obtener algunos datos de tickets de incidencia.

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

En este ejemplo, para capturar todos los últimos registros del origen de datos, el campo clave de consulta elegido para garantizar que se obtiene la última información es record_modified_date, ya que genera tickets nuevos emitidos desde la última ejecución del adaptador, así como actualiza los tickets existentes. Al seleccionar este campo como el campo clave de la consulta, el adaptador automáticamente añade la sección siguiente al final de la consulta durante su ejecución:

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

Así, solamente recupera los registros más nuevos. Tenga en cuenta que hay cierto número de cuestiones que tener en cuenta al elegir el campo clave de la consulta, y que siempre dependen del comportamiento del origen de datos y de lo que se pretende conseguir con los datos recuperados. Tenga en cuenta también que los campos elegidos en el ejemplo anterior no son siempre la mejor elección para *todas* las situaciones.

A continuación se muestra un ejemplo del archivo DataSourceControl.xml para un adaptador de SQL.

```
<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 los eventos construidos y listos para enviarse a CA Business Service Insight se escriben primero en el archivo de envío.

SendControl.xml

El archivo sendcontrol.xml contiene todas las líneas enviadas y reconocidas por CA.

El archivo permite al adaptador seguir el protocolo de reconocimiento mediante ventana deslizable para transferir los datos a CA. Dispone de más información acerca de este mecanismo en Comunicación del adaptador (en la página 94).

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

Archivo de salida IllegalEvents (.txt)

El adaptador escribe en el archivo de salida IllegalEvents todos los registros leídos pero con error de análisis. Esto normalmente se debe a la lógica de validación introducida en el archivo de configuración del adaptador. El adaptador guarda estos registros si el parámetro SaveIllegalEvents en el archivo de configuración se establece en "yes". Tenga en cuenta que la ruta para esta opción se deberá establecer también mediante "IllegalEventsDirectoryName". Esta carpeta debe existir ya, puesto que no se crea automáticamente. Si la carpeta no existe, el adaptador da error al ejecutarse.

Dentro de un adaptador de archivos, el archivo que contiene los registros de errores tiene el mismo nombre que el archivo de origen, mientras que en el adaptador de SQL, lleva el nombre de la consulta.

Translations.xml

El archivo translation.xml contiene las tablas de traducción del adaptador.

Cuando el adaptador se ejecuta en el modo en línea, el archivo contiene una copia de la tabla de traducción de la base de datos. Si la tabla de traducción se configura como remota, el adaptador carga la tabla de traducción desde la base de datos a este archivo, reemplazándola. Si es independiente, lee el archivo local.

Cuando el adaptador se ejecuta en modo sin conexión, utiliza el archivo solamente como su tabla de traducción (para ver más información acerca de los modos en línea y sin conexión, consulte Modos de ejecución del adaptador (en la página 100)).

```
<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>
```

Comunicación del adaptador

Los adaptadores interactúan con el origen de datos por un lado y con la escucha del adaptador de CA Business Service Insight y el servidor de registros por otro lado, como se muestra en el diagrama siguiente.

El adaptador se comunica con el origen de datos para recuperar los datos mediante una conexión ODBC y puede encontrarse en una ubicación local o remota con respecto al origen de datos, siempre que el adaptador pueda establecer la conexión ODBC.

El adaptador se comunica con el servidor de aplicaciones de CA Business Service Insight mediante el protocolo TCP/IP y se puede encontrar por lo tanto en una ubicación local o remota, siempre que pueda establecer conexión TCP/IP.

El adaptador debe tener abiertos dos puertos, uno para la escucha y uno para el servidor de registros. Los puertos de escucha del adaptador deben ser únicos por adaptador y no deberán entrar en conflicto con otras operaciones de red o aplicaciones que puedan estar usando también estos puertos. Por ejemplo, no deberá usar el puerto 1521, ya que generalmente el protocolo TNS de Oracle utiliza este puerto para comunicarse con la base de datos, etc. También puede ser necesario ver si hay cortafuegos locales que puedan bloquear este tráfico.

Nota: Consulte con su administrador local si no está seguro de qué puertos hay disponibles, o si necesita solicitar abrir puertos para que se pueda establecer esta comunicación.

El puerto y la dirección de escucha del adaptador se establecen en el archivo de configuración del adaptador. El puerto y la dirección de IP del servidor de registros se establecen a través de las entradas del adaptador en el registro.

El funcionamiento cliente/servidor para la escucha del adaptador es configurable. Así, se puede configurar el adaptador para que funcione como cliente o como servidor. La configuración del funcionamiento como cliente/servidor se lleva a cabo en el lado del adaptador, en los parámetros del archivo de configuración. Para hacerlo, se deberán establecer en consecuencia las variables de puerto, dirección e iniciador de conexión.

Si el iniciador de conexión (ConnectionInitiator) se establece para ser el adaptador, entonces solamente hace falta un puerto de destino. Si se establece para que sea CA Business Service Insight, entonces es necesario un puerto y una dirección de IP para la escucha del adaptador en CA Business Service Insight. De forma predeterminada, el servidor se establece para ser el adaptador. A veces esta es una importante función para que se pueda activar una regla de cortafuegos (una función conocida como "activación de puerto"). A veces los cortafuegos solamente permiten solicitudes entrantes en un puerto si se ha enviado un mensaje desde "dentro" del cortafuegos en ese mismo puerto. Después activa el cortafuegos para que se pueda establecer la comunicación.

Nota: Consulte con el administrador de la red para obtener más información acerca de las condiciones locales que pueden afectar a las comunicaciones del adaptador.

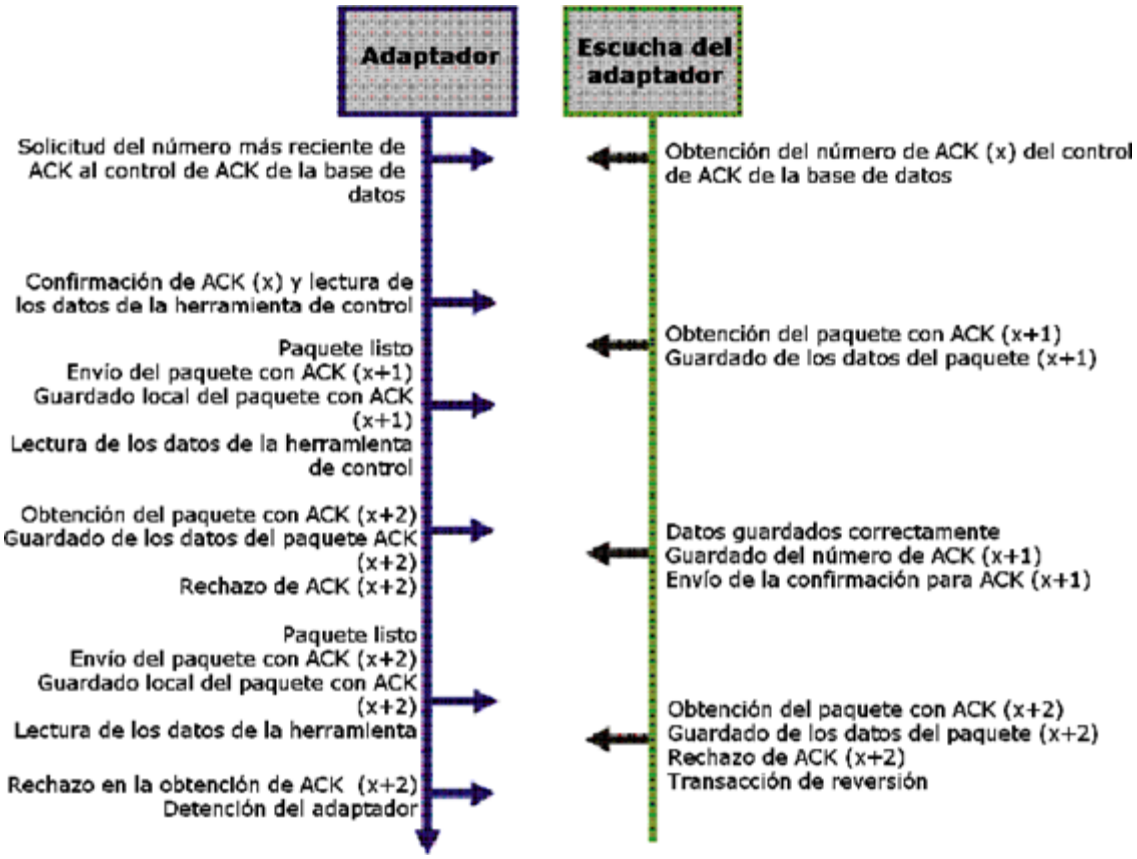
Desde un punto de vista de seguridad, se recomienda que el adaptador se establezca para ser cliente, ya que esto garantiza el destino de los eventos al trabajar en un entorno de varias implementaciones para elaboración de pruebas y producción.

Para verificar el éxito de la transmisión de los registros de datos del adaptador a la escucha del adaptador de CA Business Service Insight, el adaptador incorpora un algoritmo de ventana deslizable/reconocimientos en la capa TCP/IP. Este algoritmo básicamente envía los datos en paquetes y a continuación espera un reconocimiento por parte de la escucha del adaptador antes de pasar al paquete siguiente. Cada paquete contiene varios mensajes de datos sin procesar. El número de mensajes en un paquete se puede configurar estableciendo el parámetro Tamaño del paquete. Cada paquete incluye una secuencia en el mensaje de reconocimiento. Todos los parámetros relevantes que controlan el proceso se incluyen en la sección de interfaz de CA Business Service Insight del archivo de configuración. No obstante, en general no es necesario modificar estos parámetros.

La escucha del adaptador escribe los datos sin procesar en el paquete en una sola transacción.

Nota: La operación de reconocimiento se puede llevar a cabo solamente en los mensajes de datos sin procesar enviados a CA Business Service Insight.

La ilustración siguiente muestra el proceso de comunicación del adaptador.



Valores de configuración del registro del adaptador

En los casos en los que falta la información de la línea de comandos, el adaptador utiliza algunas de las definiciones almacenadas en el registro del sistema, en el servidor donde está instalado el adaptador.

La utilidad Gestor del adaptador escribe las entradas de registro, si la utilidad se utilizó para la instalación del adaptador. Si no se utilizó para la instalación del adaptador, estas entradas se pueden agregar manualmente al registro.

Nota: Si se instala un adaptador en un entorno UNIX, estas entradas se deberán agregar manualmente, ya que no hay ningún Gestor del adaptador para este entorno.

A continuación, se presentan las entradas del registro utilizadas por el adaptador y la utilidad Gestor del adaptador.

Entradas de servidor generales

Las entradas siguientes se escriben en el registro \HKEY_LOCAL_MACHINE\SOFTWARE\Oblicore\Adapters:

Propiedades posibles:

Nombre	Tipo	Descripción
AdaptersDir	Cadena	Directorio raíz para todos los adaptadores*.
FileAdapterConfTemplate	Cadena	Ruta de la plantilla de configuración del adaptador de archivos*. El gestor del adaptador utiliza esta información para copiar la plantilla de configuración a la carpeta del nuevo adaptador donde se haya especificado, como parte del proceso de creación del adaptador.
GenericFileAdapter	Cadena	Ejecutable del adaptador de archivos*. El gestor del adaptador crea un acceso directo al archivo ejecutable o lo copia a la carpeta del nuevo adaptador según lo especificado, como parte del proceso de creación del adaptador.
GenericSqlAdapter	Cadena	Ejecutable del adaptador de SQL*. El gestor del adaptador crea un acceso directo al archivo ejecutable o lo copia a la carpeta del nuevo adaptador según lo especificado, como parte del proceso de creación del adaptador.
LogServerAddress	Cadena	Dirección de red del servidor de registros. (Opcional)** Puerto del servidor de registros, normalmente 4040. (Opcional)** En los casos en los que se han establecido estos parámetros, el adaptador envía los mensajes de registro al servidor de registros de CA Business Service Insight.
LogServerPort	Cadena	
SqlAdapterConfTemplate	Cadena	Ruta de la plantilla de configuración del adaptador de SQL*. El gestor del adaptador utiliza esta información para copiar la plantilla de configuración a la carpeta del nuevo adaptador donde se haya especificado, como parte del proceso de creación del adaptador.

* Lo utiliza la utilidad de gestor del adaptador.

** Lo utiliza el adaptador.

Entradas individuales del adaptador

Las entradas siguientes se escriben en el registro \HKEY_LOCAL_MACHINE\SOFTWARE\Oblicore\Adapters\<Nombre del adaptador>:

Propiedades posibles:

Nombre	Tipo	Descripción
ConfigurationFileName	Cadena	Nombre del archivo de configuración del adaptador**.
Directorio	Cadena	Directorio del adaptador.*
LogFileName	Cadena	Nombre del archivo de registro del adaptador**.
Ruta	Cadena	Ruta del ejecutable del adaptador*.
RunAs	Número	Modo de ejecución*. Servicio/aplicación de consola/aplicación de Windows
Tipo	Número	Tipo de adaptador*. De archivos/de SQL
LogFileMaxSize	Número	El valor es un número en kB**. El intervalo permitido es 1000-100 000 y el valor predeterminado es 1000.

* Lo utiliza la utilidad de gestor del adaptador.

** Lo utiliza el adaptador.

Modos de ejecución del adaptador

El adaptador se puede ejecutar por:

Servicio**:**

El adaptador se puede instalar como un servicio de Windows normal. Esto permite al sistema controlar su estado (en ejecución, en pausa, detenido o automático), como lo haría un servicio de Windows normal.

Instalación del adaptador como un servicio de Windows, para lo cual se inicia el archivo ejecutable del adaptador desde la línea de comandos mediante **-i** para instalarlo como servicio y **-u** para desinstalarlo.

Aplicación**:**

Ejecución del archivo ejecutable del adaptador desde la línea de comandos. La línea de comandos del adaptador se puede ejecutar de la manera siguiente:

Opciones de la línea de comandos:

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

Parámetro	Función
-i	Instalar servicio.
-u	Suprimir servicio.
-v	Mostrar versión.
-d	Ejecutar adaptador como aplicación de consola.
-t	Comprobar archivo de configuración sólo de comprobación y detenerse.
-f	Establecer nombre del archivo de configuración.
-l	Establecer nombre del archivo de registro.
-n	Establecer nombre del servicio.
-a	Establecer dirección de servidor de aplicaciones.
-p	Establecer número de puerto de servidor de aplicaciones (1024-49151).
-la	Establecer dirección de servidor de registros.
-lp	Establecer número de puerto de servidor de registros (1024-49151).

Este tipo de ejecución se utiliza habitualmente en proyectos. Esto permite ejecutar el adaptador mediante un archivo .bat y también permite utilizar el programador de Windows para controlar el momento de ejecución del adaptador. Para programar el adaptador mediante el programador de Windows, es necesario configurar su modo de ejecución en Ejecutar una vez.

RunOnce: (opcional [yes/no]). Cuando se ha definido en el archivo de configuración en "no", una vez ejecutado, el adaptador se ejecuta de manera continua. Cuando se define en "yes", el adaptador de archivos se ejecuta, lee los registros y se detiene automáticamente cuando deja de haber registros nuevos. El adaptador de archivos lee el archivo completo, a continuación espera unos cuantos segundos e intenta leer nuevos registros (dependiendo de los valores de configuración del tiempo de inactividad). Si no hay ningún registro nuevo, se detiene. El adaptador de SQL ejecuta cada una de las consultas solamente una vez. Si se define RepeatUntilDry en "no", se detiene inmediatamente. Si se define RepeatUntilDry en "yes", espera (dependiendo del valor de SleepTime). Intenta ejecutar la consulta otra vez (dependiendo del tiempo de inactividad de la consulta), y si no hay ningún nuevo registro, se detiene.

Para obtener detalles sobre los atributos SleepTime y RepeatUntilDry, consulte Especificaciones de configuración del adaptador (en la página 369).

La sección de interfaz de CA Business Service Insight del archivo de configuración está formada por atributos que especifican los dos modos de conexión a CA Business Service Insight: en línea y sin conexión.

En el modo en línea, el adaptador se conecta a CA Business Service Insight, recupera las tablas de traducción y los comandos de control de CA Business Service Insight, a continuación envía eventos, estados y solicitudes de traducción de vuelta a CA Business Service Insight. En el modo sin conexión, el adaptador trabaja con un archivo de tabla de traducción local y escribe los eventos en un archivo de salida.

El modo sin conexión se utiliza habitualmente al desarrollar por primera vez un adaptador y al probarlo.

Al definir ConsoleDebugMode en "yes", se activa la aparición de mensajes de depuración en la consola.

Para obtener información detallada sobre los diversos indicadores, consulte Especificaciones de configuración del adaptador (en la página 369), particularmente el atributo **ConsoleDebugMode**.

Configuración y herramientas de mantenimiento

Las herramientas necesarias como parte del proceso de configuración de los adaptadores y su configuración son principalmente utilidades independientes, como las utilidades de CA Business Service Insight que son simples ejecutables de Windows.

Nota: Al configurar adaptadores en un entorno UNIX, estas utilidades no están disponibles y se debe realizar la configuración manualmente.

Al trabajar en un servidor de aplicaciones de CA Business Service Insight, las utilidades se instalan como parte de la instalación de aplicaciones y se ubican en la carpeta Archivos de programa\CA\Cloud Insight\Utilities.

Como parte de esta instalación se crea un acceso directo en el menú Inicio. Se recomienda utilizar este acceso directo para ejecutar las utilidades.

Cuando no se está trabajando en el servidor de aplicaciones de CA Business Service Insight, estas utilidades se pueden copiar a cualquier equipo de Windows como archivos estándar, o se pueden instalar mediante el paquete de CA Business Service Insight proporcionado y eligiendo una instalación personalizada. No obstante, la rutina de instalación no es obligatoria, y normalmente es suficiente copiar los archivos .exe a una ubicación conveniente en la estación de trabajo. No obstante, esta opción puede dejar algunos archivos .dll sin copiar. En este caso, también se deben copiar, si están disponibles, desde el servidor a la carpeta local de la estación de trabajo.

Cómo configurar adaptadores y traducciones

Esta etapa conlleva la realización de los pasos siguientes:

1. Configurar el adaptador mediante el asistente del adaptador o editar manualmente el archivo de configuración XML (descrito en el capítulo siguiente).
2. Implementar el adaptador.
3. Probar el adaptador.
4. Realizar traducciones.
5. Escribir scripts de traducción para admitir un proceso de traducción automática (opcional).

Nota: El adaptador se puede implementar automáticamente cuando se usa el asistente del adaptador, ya que el servicio de implementación del adaptador nuevo se ejecuta como servicio de fondo en el servidor de aplicaciones para gestionar esta tarea.

Implementación de un adaptador nuevo (asistente del adaptador)

Cuando cree un adaptador nuevo mediante el asistente del adaptador, asegúrese de que se estén ejecutando los servicios de "escucha del adaptador" e "implementación del adaptador".

Implementación de un adaptador nuevo (manualmente)

Requisitos previos para crear un adaptador nuevo

Antes de empezar a crear un adaptador nuevo es necesario cumplir con todo lo siguiente:

- **Carpeta raíz del adaptador**: si se ha instalado CA Business Service Insight en el servidor, esta carpeta se encuentra bajo la carpeta raíz Archivos de programa\CA\Cloud Insight; si no, se deberá crear.
- **Carpeta del adaptador individual**: cree una carpeta en la carpeta raíz del adaptador para el adaptador específico.

  Nota: Si está utilizando la utilidad AdapterManager, la utilidad automáticamente crea la carpeta al añadir el nuevo adaptador.
- **Archivos ejecutables del adaptador**: TextFileAdapter.exe, el archivo ejecutable del adaptador de archivos de texto para adaptadores de archivos de texto; SQLAdapter.exe, el archivo ejecutable del adaptador de archivos SQL para adaptadores de SQL. Estos se encuentran en el servidor de aplicaciones, bajo la carpeta Archivos de programa\CA\Cloud Insight\Adapters.

  Nota: Durante el proceso de creación del adaptador, a través de la utilidad del gestor del adaptador, siempre que sea posible deberá elegir la opción para crear un acceso directo hacia los archivos ejecutables, en lugar de crear una copia. Esto garantizará que cuando se aplique una actualización o una versión de servicio (SR) a CA Business Service Insight, todos los componentes binarios se actualicen correctamente.
- **Plantillas de configuración**: plantillas del archivo de configuración del adaptador. Coloque los archivos en la carpeta raíz del adaptador. Estos se encuentran en el servidor de aplicaciones, bajo la carpeta Archivos de programa\CA\Cloud Insight\Adapters. Las plantillas de configuración se utilizan para crear el archivo de configuración sin tener que hacerlo desde cero. Es también corriente utilizar un archivo de configuración del adaptador existente para este fin.
- **Utilidad del gestor del adaptador**: un archivo ejecutable independiente. Es suficiente hacer una copia del archivo AdaptersManager.exe desde la carpeta de utilidades bajo la carpeta Archivos de programa\CA\Cloud Insight\Utilities en el servidor de aplicaciones. No es necesario crear el adaptador mediante esta utilidad. Esta utilidad se puede utilizar solamente en un servidor Windows.
- **Dos puertos TCP\IP abiertos**: un puerto para la escucha del adaptador en el servidor de aplicaciones y otro para el servidor de registros. El puerto del servidor de registros es normalmente 4040.

- Compruebe qué componentes de servicios de aplicaciones de CA Business Service Insight se están ejecutando. Para ejecutar el adaptador, se tienen que estar ejecutando los componentes de servicio siguientes:
  - **AdapterListener**
  - **TaskHost**
  - **LogServer**

Siga estos pasos:

1. Ejecute la utilidad AdapterManager (consulte sección sobre la utilidad AdapterManager más arriba).
2. Prepare todos los requisitos previos anteriores y cree un archivo por lotes con la línea de comandos del ejecutable (consulte Modos de ejecución del adaptador (en la página 100)).

Modificación del archivo de configuración del adaptador

Al crear un adaptador, la parte más importante del trabajo es editar el archivo de configuración.

Este trabajo conlleva establecer los atributos en el archivo XML para controlar el comportamiento del adaptador para que se ejecute según sea necesario. Cada una de las secciones del archivo de configuración, un archivo XML, corresponde a un paso en su flujo de trabajo interno.

Las secciones son:

- **Sección General**: diversos atributos de adaptador, directorio de trabajo, especificaciones de archivos de salida; indicador de depurador, valores predeterminados, etc.

Oblicore:

- **Sección OblicoreInterface**: atributos de la conexión con el servidor de CA Business Service Insight (puerto TCP/IP, modo de seguridad, etc.).
- **Sección DatasourceInterface**: atributos de conexión con el origen de datos (ruta de archivos y patrón, cadenas de conexión, consultas de SQL, etc.).
- **Sección InputFormatCollection**: reglas para analizar y manipular el formato de datos original (delimitadores, tipos de campos, orden de los datos, expresiones regulares, etc.).
- **Sección TranslatorCollection**: reglas para compuesto unificado de eventos de CA Business Service Insight a partir de los campos de datos analizados y manipulados.
- **Sección TranslationTableCollection**: reglas de asignación de datos entre terminología de datos originales y entidades de datos de CA Business Service Insight.

Estas secciones se describen en detalle en **Especificaciones de configuración del** adaptador (en la página 369).

Nota: No importa el orden de los nodos XML dentro de cada sección.

Adaptadores de archivos

Los Adaptadores de archivos utilizan el componente genérico FileAdapter (archivo ejecutable) y el archivo de configuración que se emplea para analizar archivos ASCII.

Flujo de trabajo del adaptador de archivos:

- Copiar/cambiar el nombre del archivo de origen a un archivo de trabajo.
- Leer el registro lógico.
- Analizar el registro según los delimitadores o las expresiones regulares.
- Buscar el inputFormat correcto.
- Construir el registro de eventos.
- Traducir el registro de eventos.
- Archivo de control de actualización.

Ejemplo de archivo de configuración

El ejemplo siguiente utiliza un origen de datos de archivos ASCII sencillo con sus requisitos de salida del evento y revisa los valores de configuración obligatorios para su archivo de configuración del adaptador.

El archivo de configuración se puede consultar y editar mediante la utilidad XMLPad.

Para obtener una descripción general de la estructura y del contenido del archivo de configuración, consulte las secciones relevantes.

Nota: Los valores de configuración revisados son solamente los valores de configuración de atributos principales y más importantes. Para ver todas las especificaciones de atributos, consulte Especificaciones de configuración del adaptador (en la página 369).

Caso práctico de adaptador de archivos

Supongamos que tiene el sistema de control del servidor siguiente para producir archivos de registro con información de acuerdo con la estructura siguiente:

Servidor	Marca de tiempo	Utilización de memoria	Uso de CPU
Server001p	1.11.2004 0:00	26.71	78.85
Server002p	1.11.2004 1:00	49.09	89.49
Server001p	1.11.2004 2:00	51.37	57.51
Server001p	1.11.2004 3:00	23.53	98.9
Server002p	1.11.2004 4:00	86.21	85.11
Server002p	1.11.2004 5:00	39.98	35.97
Server001p	1.11.2004 6:00	70.58	55.96

Los archivos se entregan a la carpeta del adaptador de CA Business Service Insight en una subcarpeta llamada "data". Los nombres de archivo llevan todos el prefijo 'ServerData' y un sufijo con la fecha y hora. Los archivos son también archivos CSV con extensión .csv.

Se requiere el siguiente evento de salida:

- **Campo de traducción de recurso**: Servidor.
- **Campo de marca de tiempo**: Marca de tiempo.
- **Campos de datos**: Utilización de la memoria, Uso de la CPU.

Además, se supone que el origen de datos se encuentra en Bélgica (zona horaria CET con compensación de hora +1 y que incluye períodos de horario de verano con una hora añadida en el verano).

Sección General del archivo de configuración

- **MajorVersion y MinorVersion**: se debe establecer de forma predeterminada en la versión de la aplicación.
- **WorkingDirectoryName (opcional)**: especifica la ruta predeterminada para los archivos de salida del adaptador (control de origen de datos, traducción y envíos). En este caso, se define en "output", ya que esta carpeta se crea a continuación bajo la carpeta principal de adaptadores.

Los indicadores siguientes controlan la forma en que la que el adaptador lee y traduce los registros:

- **RunOnce**: si se define en "yes", el adaptador se ejecuta una vez, lee todos los datos disponibles y, a continuación, se detiene.
- **DataSourceDifferenceFromUTC**: indica la diferencia de hora entre UTC (hora universal coordinada, siempre con compensación de hora cero y equivalente a GMT) y la zona horaria del origen de datos. La zona horaria del origen de datos es la zona horaria de acuerdo con la cual se indican los campos de hora. El motivo de tener este atributo es que el adaptador normaliza todas las fechas en UTC. Tener todas las fechas en UTC dota a la aplicación de flexibilidad para mostrar a continuación las horas según los requisitos del usuario. Los atributos siguientes proporcionan al adaptador los detalles de cómo transformar los campos de hora a partir de la entrada a UTC:
  - **DefaultOffset**: variación con respecto a la hora UTC fuera del horario de verano. En este caso, se establece en "'1" para el horario centroeuropeo (CET).
  - **TimeFormat**: el formato por el cual se deberán analizar los detalles de horario de verano (descritos a continuación). El formato de hora europeo es "dd/mm/aaaa hh:mi:ss", mientras que las especificaciones de formato de CA Business Service Insight se definen como "%d/%m/%Y %H:%M:%S".
  - **DayLightSaving**: período de horario de verano de la zona horaria del origen de datos. Este elemento es opcional (lo que significa que, si no se selecciona, no hay horario de verano) y puede existir más de una vez; una vez por cada período de horario de verano introducido. Cuando hay varios elementos, se deben ordenar por tiempo, y los períodos no se deben solapar. Normalmente, al configurar adaptadores, este elemento se especifica para un número de años por delante. En este caso, solamente se ha configurado un año como ejemplo.
- **From**: fecha de inicio del período. Se especifica utilizando el formato de hora establecido arriba, "25/03/2007 02:00:00".

- **To**: fecha de finalización del período. Se especifica utilizando el formato de hora establecido arriba, "28/10/2007 03:00:00".
- **Shift**: el cambio de hora añadido a DefaultOffset dentro del período de horario de verano. Introduzca "1" cuando la hora se adelante una hora durante el período de horario de verano indicado entre las dos fechas especificadas.

Sección OblicoreInterface del archivo de configuración

Esta sección define la conexión al servidor de CA Business Service Insight.

- En el modo en línea, el adaptador se conecta a CA Business Service Insight, recupera las tablas de traducción y los comandos de control de CA Business Service Insight, y a continuación envía eventos, estados y solicitudes de traducción de vuelta a CA Business Service Insight. Configure el adaptador en este modo y establezca el valor en "online".
- Port: número de puerto TCP/IP que usa el adaptador para comunicarse con el servidor de CA Business Service Insight (donde se encuentra la escucha del adaptador).

  Suponiendo que no surjan problemas al hacerlo, configure este adaptador para que use el puerto "5555" (elegido arbitrariamente). Para poder establecerse la comunicación, esto se deberá especificar también en el servidor en la interfaz gráfica de usuario del adaptador también.

Sección DataSourceInterface del archivo de configuración

La sección DatasourceInterface está formada por atributos que especifican la conexión y el tipo de conexión entre el adaptador y el origen de datos. Hay dos tipos de interfaz, de archivos y SQL. La diferencia principal entre las dos es que, en el caso de la interfaz de archivos hace falta una recolección de archivos, mientras que en la interfaz SQL es necesaria una recolección de consultas.

La sección DataSourceInterface también define cómo gestiona el adaptador el archivo de origen (si suprime el archivo original si sólo lo creó el adaptador, o si deja los datos por si hacen falta para otros usos, etc.).

Para que los adaptadores de archivos puedan leer y analizar archivos ASCII, la interfaz de archivos se utiliza como muestra la ilustración siguiente. Elija los valores de configuración siguientes:

La sección Files bajo el nodo DataSourceInterface hace referencia a la conexión al origen de datos. Configure los atributos siguientes:

Nota: Esta sección tiene un aspecto completamente diferente en el caso de adaptadores de SQL.

- **DeleteFileAfterProcessing**: define cómo gestiona el adaptador el archivo de origen y determina cómo controla la lectura para poder leer sólo registros nuevos. En este caso, los archivos de origen se quedan en su sitio en el servidor y este valor se establece en "no".

  En aquellos casos en los que se cree un archivo solamente para el adaptador y se pueda suprimir después de terminar de procesarse, se deberá establecer en "yes". A continuación se cambia el nombre del archivo, se procesa y se suprime.

  Cuándo se define en "no", el archivo se copia y se procesa utilizando el archivo copiado. Si se añaden nuevos registros al final de este archivo, el adaptador copia estos nuevos registros al archivo de trabajo durante el ciclo siguiente. Si no se añaden registros nuevos al archivo, el adaptador busca el primer archivo con el mismo patrón y nombre (en orden lexicográfico) que el archivo actual. Si el adaptador encuentra dicho archivo, empieza a trabajar desde él. El adaptador no vuelve al archivo anterior, ni siquiera si se añaden nuevos registros.

  Defínalo en "no" cuando sea necesario mantener la integridad del archivo de origen y cuando deba adjuntarse el archivo.

- **InputFormat**: hace referencia al nombre dado al elemento InputFormat en la siguiente InputFormatCollection que gestiona los datos desde esta conexión de origen de datos. El formato de entrada es la estructura de campos de los datos de entrada provenientes del origen de datos después de haber sido analizados por el adaptador. El método de análisis se especifica en el atributo de delimitadores, tal como se explica a continuación. Al gestionar más de una conexión mediante formatos de interfaz diferentes, este campo cobra mayor importancia y determina qué estructura de formato de entrada gestiona cada dato del origen de datos.
- **Path**: ubicación física de los archivos de origen de datos. Por ejemplo: C:\Archivos de programa\CA\Cloud Insight\Adapters\ServersAdapter\data\.
- **NamePattern**: especifica el nombre de archivo del origen de datos. Se pueden utilizar caracteres comodín si más de un archivo utiliza el mismo formato de entrada. Si se especifica un nombre de archivo sin caracteres comodín, el adaptador busca solamente un archivo con este nombre. Cuando se usen caracteres comodín, el adaptador buscará todos los archivos que correspondan al patrón, los clasificará lexicográficamente y, a continuación, los leerá uno a uno. En la siguiente ejecución, buscará nuevos registros en el último archivo antes de pasar al siguiente.

  En este ejemplo, si se usa el carácter comodín "*", entonces el valor de atributo es "ServerData*.csv". (El adaptador lee todos los archivos cuyo nombre empiece por ServerData y tengan la extensión .csv).

  Importante: Se recomienda agregar una fecha y hora al final de los nombres de archivo mediante el formato siguiente, AAAAMMDD-HHMISS, para garantizar que los archivos se ordenen correctamente, se lean en el orden correcto y que no quede ninguno sin leer. La parte de la hora se puede agregar también si se agregan varios archivos cada día.
- **Delimiters**: método para determinar cómo analizar el archivo. Se pueden especificar uno o más caracteres como delimitadores, según las filas de datos que se analizarán en los campos. Si no se especifica, el valor predeterminado es "/t".

El archivo de origen de datos en este ejemplo es un archivo CSV (delimitado por comas). La forma más sencilla de analizar estos archivos es especificar la coma como delimitador.

Estos son otros métodos disponibles para realizar el análisis:

- **RegExForParser**: utiliza una expresión regular para establecer la regla de análisis.
- **LogicLineDefinition**: se utiliza cuando se construye una línea en el archivo a partir de varias líneas.

- **TitleExists** (opcional): especifica si la primera línea del archivo de origen de datos que no esté en blanco es una línea de título. El adaptador puede utilizar el título al analizar los datos. En este ejemplo, cada archivo de datos contiene la fila de título, por lo que se deberá especificar "yes" para este atributo.

Sección InputFormatCollection del archivo de configuración

Esta sección especifica la estructura de los datos recuperados desde el origen de datos, cómo se dividirán las filas de datos en campos y qué tipos de campos y formatos hay. En esta sección se puede hacer un filtrado y una manipulación inicial de los datos mediante los campos InputFormatSwitch y Compound respectivamente.

En esta sección se pueden definir las métricas de validación para los registros de entrada mediante InputFormatValidation y ValidationCase, que determinan si un registro es válido o no.

El nodo InputFormatCollection puede contener uno o más nodos InputFormat.

El flujo de trabajo general que el adaptador sigue en esta sección es:

- Se hace corresponder la fila de datos con el campo InputFormat especificado en DataSourceInterface.
- Los datos se diseccionan en campos siguiendo la especificación InputFormat correspondiente. El orden de InputFormatFields deberá coincidir con el orden de campos analizados desde el origen de datos.
- Se asignan valores a los campos compuestos combinando y dividiendo los campos de datos previamente especificados. Los campos compuestos se deberán definir tras definir los campos normales.
- Se comprueban los datos procesados con las condiciones de TranslatorSwitch para determinar qué traductor se utiliza para construir el evento de salida a partir del registro de entrada.
- Los datos procesados se envían al traductor coincidente o se ignoran.

Se trabaja con los parámetros siguientes:

- **InputFormatName**: nombre para este formato al que hará referencia la sección DatasourceInterface.
- **InputFormatFields**: InputFormatFields puede contener uno o más nodos de campos, según el número de orígenes de datos de los campos de entrada.
- **InputFormatField**: especifica un campo de datos de la fila de datos de origen, o un campo compuesto.

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

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

  - **Name**: nombre para este campo al que harán referencia otros elementos, como el elemento compuesto o TranslatorFields como campo de origen.
  - **Type**: tipo de datos del campo: cadena, entero o tiempo real.
  - **Source**: valor de origen para este campo, el valor predeterminado tomado es "event", y los valores posibles son:
    - **event**: el valor del campo se toma del evento que viene del origen de datos. Los valores de los campos se toman en el mismo orden de llegada del origen de datos.
    - **compound**: el valor del campo se construye a partir de otros campos basándose en una manipulación de los valores o constantes de otros campos. Los campos manipulados ya deben haberse definido.
    - **title**: el valor del campo se toma de los nombres del campo de título. El campo de referencia ya debe haberse definido.
    - **filename**: el valor del campo se toma del nombre de archivo del origen de datos; solamente para adaptadores de archivos de texto.
    - **constant**: el valor del campo es constante y se toma de ConstantValue, cuya propiedad deberá aparecer a continuación.
- **TranslatorSwitch**: determina qué traductor se utiliza para traducir la fila de datos a un evento unificado.
  - **DefaultTranslator**: traductor que se debe utilizar cuando no hay criterios coincidentes. Si el valor se establece en "Ignore", no se usará ningún traductor, y la línea se ignorará y no se enviará a CA Business Service Insight.

Sección TranslatorCollection del archivo de configuración

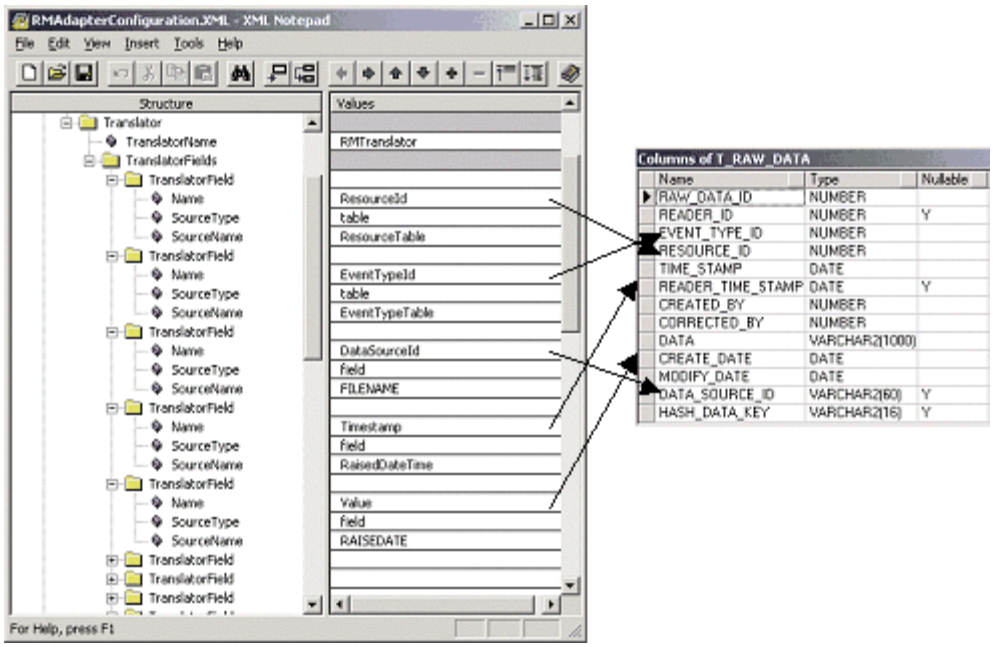
La sección TranslatorCollection define cómo se traducirán a eventos de CA Business Service Insight los registros de orígenes de datos analizados y manipulados que se han extraído en las secciones anteriores.

Esta sección también define cómo gestionar los eventos duplicados y cómo utilizar el mecanismo de singularidad de eventos (para ver más información, consulte Singularidad del evento (en la página 146)).

Cuando el modo de interfaz se ha establecido como en línea ("online"), el evento de CA Business Service Insight tiene una estructura unificada que contiene los campos siguientes:

- **Timestamp**: hora en la que se produjo el evento.
- **ResourceId**: ID de recurso asociado al evento (el recurso que se midió dentro de ese evento).
- **EventTypeId**: el tipo de evento asociado al evento. Describe el tipo de evento (tipo de medición del recurso, tipo de acción de ticket, etc.).
- **DataSourceId** (opcional): cualquier valor. Proporciona criterios de filtrado adicionales para eventos de datos sin procesar.
- **Value** (varios): valores del evento (resultado de medición, número de ticket, etc.). Este campo a menudo aparece más de una vez.

La estructura del traductor se corresponde con la estructura del tipo de evento dentro de CA Business Service Insight, y también con la tabla de base de datos T_RAW_DATA que almacena el evento, tal y como muestra la ilustración siguiente:


RMAdapterConfiguration.XML - XML Notepad
File Edit View Insert Tools Help
Structure
Values
Translator
TranslatorName
TranslatorFields
TranslatorField
Name
SourceType
SourceName
RMTranslator
ResourceId
table
ResourceTable
EventTypeId
table
EventTypeTable
DataSourceId
field
FILENAME
Timestamp
field
RaisedDateTime
Value
field
RAISEDATE
For Help, press F1
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(60) Y
HASH_DATA_KEY VARCHAR2(16) Y

- **Translator**: describe cómo traducir el conjunto de campos recibidos en el evento de salida.
- **TranslatorName**: nombre utilizado por InputFormat para enviar conjuntos de campos a este traductor. Este ejemplo utiliza el nombre "Translator1". Consulte la ilustración anterior para ver los valores que se podrían utilizar para este escenario.
- **TranslatorFields**: contiene una lista de elementos TranslatorField, cada uno de los cuales contiene los atributos siguientes:
  - **Name**: nombre del campo. En la interfaz en línea debe ser Timestamp, ResourceId, EventTypeId, ResourceId o Value.
  - **SourceType**: especifica el origen del valor del campo. Puede ser uno de estos valores:
    - **Field**: el valor de este campo se toma del campo en el formato de entrada. El atributo SourceName contiene el nombre del campo InputFormat.
    - **Table**: el valor del campo se toma de la tabla de traducción. El atributo SourceName contiene el nombre de la tabla de traducción que hace referencia a un nombre que se define en la sección siguiente de TranslationTableCollection. Este tipo se utiliza para los valores que se traducirán a recursos y tipos de eventos en el evento.
    - **Lookup**: el valor de este campo se toma de la tabla de traducción. El atributo SourceName contiene el nombre de tabla. El valor que se tiene que traducir del atributo LookupValue y no del formato de entrada. Se utiliza habitualmente cuando se requiere un valor constante para realizar traducciones.
    - **Constant**: el valor del campo es constante, y su valor está en el atributo ConstantValue. Al utilizar un valor constante, es necesario especificar el tipo de campo mediante el atributo Type.
  - **SourceName**: contiene el nombre del campo o el nombre de la tabla de traducción.
  - **LookupValue**: contiene el valor de búsqueda cuando SourceType=lookup.
  - **ConstantValue**: contiene el valor constante cuando SourceType=constant. Cuando el tipo de campo es de tiempo, el valor constante es una cadena con formato según TimeFormat (establecido en la sección General del adaptador) o "Now" o "NowUtc", donde "Now" es la hora en curso en la configuración regional del origen de datos y "NowUtc" es la hora en curso en UTC.

Esta sección también contiene las tablas de asignación que definen la asignación de valores del origen de datos a campos evento de CA Business Service Insight, y guarda la definición de la tabla con el valor del origen de datos al que se hace referencia y debe traducirse.

- **LoadingMode**: el valor predeterminado para la interfaz en línea es "remote", y para la interfaz sin conexión es "standalone".

  El método de carga de las tablas de traducción se especifica como sigue:

  - **Standalone**: el adaptador carga las tablas de traducción localmente. No hay conexión al servidor de CA Business Service Insight para la traducción. Los cambios en las tablas de traducción se almacenan solamente en el archivo local.
  - **Remote**: el adaptador envía una solicitud para cargar todas las tablas desde el servidor de CA Business Service Insight. Los cambios realizados en las tablas de traducción se almacenan también localmente.

- **TranslationTable**: vincula el valor del evento con la tabla de asignación.

  - **Name**: nombre de tabla de traducción utilizada a la que hace referencia el traductor.
  - **DestinationType**: [resource, event_type, contract_party, service, time_zone, value]. Guarda el tipo de la tabla de traducción. En este ejemplo, la columna **Servers** en el archivo del origen de datos se traduce en un recurso de CA Business Service Insight. Por lo tanto, el tipo de tabla de traducción es un recurso que guarda traducciones de valores para los ID de recurso en CA Business Service Insight.
  - **TranslationField**: nombre del campo de donde traducir y que se toma de los campos de formato de entrada. Puede contener un máximo de cinco campos.

Cada tabla de traducción definida en el archivo de configuración debe tener una definición correspondiente en la interfaz de usuario de CA Business Service Insight.

La representación XML de un archivo de configuración de muestra es la siguiente:

```
<?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:\Archivos de
programa\Oblicore\Adapters\ServersAdapter\data\" TitleExists="yes" SleepTime="60"
Delimiters=","/>
      </Files>
</DataSourceInterface>
<InputFormatCollection>
     <InputFormat InputFormatName="InputFormat1">
          <InputFormatFields>
               <InputFormatField Name="resource" Type="string"/>
                    <InputFormatField Name="timestamp" Type="time"
TimeFormat="%d.%m.%Y %H:%M"/>
                        <InputFormatField Name="memory_util" Type="real"/>
                        <InputFormatField Name="cpu_util" Type="real"/>
                </InputFormatFields>
     <TranslatorSwitch DefaultTranslator="Translator1"/>
     </InputFormat>
</InputFormatCollection>
<TranslatorCollection>
<Translator TranslatorName="Translator">
         <TranslatorFields>
            <TranslatorField Name="ResourceId" SourceType="table"
SourceName="ResourceTable"/>
            <TranslatorField Name="EventTypeId" SourceType="lookup"
SourceName="EventTable" LookupValue="PerformanceEvent"/>
            <TranslatorField Name="Timestamp" SourceType="field"
SourceName="timestamp"/>
            <TranslatorField Name="Value" SourceType="field"
SourceName="memory_util"/>
            <TranslatorField Name="Value" SourceType="field"
SourceName="cpu_util"/>
         </TranslatorFields>
      </Translator>
   </TranslatorCollection>
   <TranslationTableCollection LoadingMode="remote">
      <TranslationTable Name="ResourceTable" DestinationType="resource">
         <TranslationField>resource</TranslationField>
      </TranslationTable>
      <TranslationTable Name="EventTable" DestinationType="event_type">
         <TranslationField>resource</TranslationField>
      </TranslationTable>
   </TranslationTableCollection>
</AdapterConfiguration>
```

Adaptadores de SQL

Los adaptadores de SQL son, fundamentalmente, el uso del componente genérico del adaptador de SQL (archivo ejecutable de adaptador de SQL) con los valores de configuración apropiados en el archivo de configuración. El adaptador de SQL se puede conectar a todos los orígenes de datos que sean compatibles con ODBC y OLEDB. La conexión se establece mediante cadena de conexión. El controlador adecuado debe instalarse en el servidor en el que se encuentra instalado el adaptador.

Ejemplos de orígenes de datos:

- Oracle
- SQL Server
- Acceso
- Excel
- Archivos de texto, archivos CSV (también se pueden conectar mediante el adaptador TEXT, pero la conexión ODBC a menudo proporciona capacidades de filtración extra/consulta).

Flujo de trabajo del adaptador de SQL:

- Abrir la conexión.
- Reemplazar las variables locales por los últimos valores de campos clave.
- Completar automáticamente, generar las cláusulas "where" de las consultas y concatenar al final de las consultas.

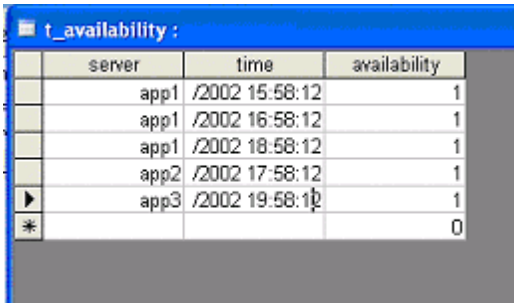
  Ejecutar la consulta y obtener el conjunto de registros.

  Para cada registro del conjunto de registros devuelto por la consulta:

  - Buscar el InputFormat correcto.
  - Construir el registro de eventos.
  - Traducir el registro.
  - Actualizar el valor de campos clave en el archivo de control.
- Cerrar la conexión.
- Pasar a modo inactivo.

Ejemplo de archivo de configuración del adaptador de SQL

Supongamos que tenemos una base de datos de MS Access (.mdb) con la tabla siguiente:


t_availability :

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

La única diferencia en el archivo de configuración entre un archivo de configuración de adaptador de SQL y un archivo de configuración de adaptador de archivos es la sección DatasourceInterface.

La sección DatasourceInterface en el adaptador de archivos almacena la recolección de archivos, mientras que el archivo del adaptador de SQL incluye ConnectionString y QueryCollection.

La diferencia principal entre los dos archivos de configuración reside en el método de recuperación y análisis. El resto del archivo es igual.

La interfaz de SQL define la conexión con la base de datos y las consultas que se utilizan para recuperar los datos.

Los detalles son los siguientes:

Nota: La definición de la sección se basa en la base de datos de origen de datos anterior.

Elemento ConnectionString

ConnectionString: cadena de conexión para acceder a la base de datos de origen.

ConnectionString se puede definir en el elemento DataSourceInterface y/o en los elementos Query. La definición de ConnectionString en el elemento DataSourceInterface es el valor predeterminado y se utiliza solamente en consultas donde no se ha definido ConnectionString. Esto permite al adaptador conectarse a varias bases de datos, donde cada consulta puede tener su propia cadena de conexión. Para obtener más información acerca del mecanismo de la cadena de conexión, consulte la sección siguiente.

Sección QueryCollection del archivo de configuración

Query: especifica los atributos de la consulta.

- **QueryName**: nombre de la consulta.
- **InputFormat**: formato de entrada asociado a la consulta. El adaptador utiliza el valor de InputFormat para extraer los datos del origen.

  Nota: El orden de los campos InputFormat tiene que corresponder con el orden de los campos de consulta seleccionados.
- **SleepTime**: tiempo en segundos durante los cuales el adaptador permanece inactivo a la espera de que lleguen nuevos datos.
- **SelectStatement**: contiene la declaración seleccionada que ejecutar en la base de datos de origen.

  Nota: Debe introducir los campos clave de la consulta como los primeros valores seleccionados en la declaración.

  - **AutoCompleteQuery**: si se establece en "yes", el adaptador concatena automáticamente una declaración where a la consulta especificada:
    - Creando una declaración where que recupera solamente valores nuevos basados en los campos clave.
    - Ordenando la declaración de resultados según los campos clave.
- **QueryKeyFields**: define los campos donde iniciar la siguiente recuperación de datos.
  - **KeyField**:

- **Name**: nombre del campo que se toma de los campos de la consulta.
- **Sort**: orden de clasificación de los datos (ascendente o descendente).
- **ValueLeftWrapper**: concatena caracteres antes del valor del campo. El valor predeterminado es ' (apóstrofo).
- **ValueRightWrapper**: concatena caracteres después del valor del campo. El valor predeterminado es ' (apóstrofo).
- **Critical**: deja de ejecutar otras consultas si esta consulta particular da error.
- **SleepTime**: tiempo de inactividad necesario entre operaciones. El valor predeterminado es ' (apóstrofo).

Nota: Los campos de fecha a menudo requieren utilizar caracteres especiales para encerrar la fecha en sí. Los caracteres necesarios para identificar el campo como un campo de fecha dependen del origen de datos. El carácter #, como muestra la ilustración al principio de la sección, se puede utilizar para envolver el campo de valor en Excel. Otros orígenes de datos, sin embargo, pueden requerir métodos diferentes. Consulte Especificaciones de configuración del adaptador (en la página 369) para obtener más información.

- **SelectInitialValues**: una declaración de selección que proporciona los valores iniciales para los campos clave para la primera declaración where (cuando el archivo de control está vacío).

  Ejemplo de una consulta (ODBC basado en Excel) construida con AutoCompleteQuery="yes":

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

  Esta declaración de selección debe ejecutarse en la base de datos de destino en la cual se está ejecutando la consulta. Esto puede variar entre los orígenes y los controladores ODBC que se utilizan para conectarse a ellos. Por ejemplo, en Oracle podría seleccionar los valores de la tabla especial dual (select 'aaa', '1-jan-1970' from dual), pero en Excel sólo se pueden seleccionar los valores directamente, sin una tabla. (select 'aaa')

A continuación se incluye un archivo de configuración completo en formato 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>
```

Cadena de conexión del adaptador de SQL

El mecanismo de cadena de conexión del adaptador de SQL está diseñado para permitir hacer lo siguiente:

- Trabajar con varias cadenas de conexión en el mismo adaptador.
- Trabajar con una plantilla de cadena de conexión, donde el nombre de archivo se puede cambiar sin cambiar el archivo de configuración.
- Suprimir el origen de datos del archivo una vez el adaptador lo ha terminado de leer.
- Mover el origen de datos del archivo a otra ubicación una vez el adaptador lo ha terminado de leer.

Además de la simple definición de la cadena de conexión como una cadena en el elemento ConnectionString, la cadena se pueden definir por segmentos, donde cada segmento guarda los valores específicos concatenados a la cadena de conexión. Esto permite al adaptador construir la cadena de conexión dinámicamente.

Hay dos tipos de segmentos. Uno es texto y contiene texto que se concatena a la cadena de conexión tal cual. El segundo es un archivo y contiene un nombre de archivo con o sin caracteres comodín. El segmento de archivo puede aparecer solamente una vez. Contiene otros atributos que definen lo que se deberá hacer con el archivo de lectura.

ConnectionString se puede definir en el elemento QueryCollection y/o en los elementos Query. La definición de ConnectionString en el elemento QueryCollection es el valor predeterminado y se utiliza solamente en consultas sin definición de ConnectionString específica.

Elementos y atributos

Elemento ConnectionString

Este elemento puede contener elementos secundarios Segment. Si contiene al menos un elemento Segment, la cadena de conexión se concatena a partir de él. De lo contrario, la cadena de conexión se toma del texto del elemento ConnectionString (como en la versión actual).

Elemento Segment

Este elemento contiene un atributo obligatorio llamado *Type*, que tiene dos valores válidos: "text" y "file". Cuando se define Type=file, el elemento Segment debe contener como mínimo un elemento File. Cada elemento File se considera una consulta diferente.

Elemento File

Este elemento contiene atributos que definen qué archivos se deberán utilizar en la cadena de conexión y lo que se deberá hacer con el archivo cuando el adaptador lo termine de leer.

- **Path**: define la ruta de archivo (directorio).
- **NamePattern**: define el nombre de archivo con la ruta dada. Puede contener caracteres comodín.
- **UsePath**: los valores válidos son "yes" o "no". El valor predeterminado es "yes". Si se establece como "yes", la ruta de archivo se concatena a la cadena de conexión.
- **UseFileName**: los valores válidos son "yes" o "no". El valor predeterminado es "yes". Si se establece en "yes", el nombre de archivo se concatena a la cadena de conexión (así lo exigen los archivos de Excel).
- **UpdateSchemaFile**: los valores válidos son "yes" o "no". El valor predeterminado es "no". Si se establece en "yes", el adaptador actualiza el archivo schema.ini con el nombre de archivo actual.

  Nota: Utilice este atributo solamente cuando desee que el adaptador cambie el archivo schema.ini.

Variables internas

Se han agregado dos variables internas adicionales que se pueden utilizar en los elementos SelectStatement y SelectInitialValues. Estas son:

- :file_name: se reemplaza con el nombre de archivo actual y la extensión.

- *:file_name_no_ext*: se reemplaza con el nombre de archivo actual sin la extensión.

Ejemplos

Ejemplo 1: Cadena de conexión simple para Oracle:

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

o

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

Ejemplo 2: Cadena de conexión simple para Excel con un único archivo:

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

o

```
<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>
```

Ejemplo 3: Cadena de conexión simple para usar con varios archivos de 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>
```

Ejemplo 4: Uso de una entrada DSN de ODBC estándar:

El uso de entradas DSN de ODBC estándar le permiten conectarse a cualquier origen que tenga una entrada DSN creada en el gestor de ODBC en el servidor de aplicaciones. La entrada DSN de ODBC estándar se puede encontrar en la sección de herramientas administrativas del servidor.

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

Lectura de registros a partir de un origen de datos de archivo

Mientras haya una interfaz de ODBC al origen de datos, es posible configurar un adaptador de SQL para consultar archivos. Para configurar el adaptador para leer varios archivos, es necesario utilizar elementos Segment en el elemento ConnectionString. Para ver un ejemplo, consulte la sección anterior que describe la cadena de conexión.

Así es como trabaja el adaptador de SQL con los archivos:

1. En cada consulta, el adaptador intenta leer desde el archivo hasta que no pueda recuperar más registros.
2. Después, el adaptador pasa al archivo siguiente e intenta leerlo.
3. Si no hay más archivos, el adaptador pasa a modo inactivo para esta consulta, de acuerdo con el atributo SleepTime.

Archivo schema.ini

Cuando se usa un controlador ODBC de texto, el formato del archivo de texto se determina según el archivo de información de esquema (schema.ini). Este archivo de información de esquema se deberá colocar en el mismo directorio que el origen de datos de texto.

Los archivos schema.ini se construyen a partir de entradas, cada una de las cuales describe la estructura de un solo archivo de texto. La primera línea de cada entrada es el nombre del archivo de origen de texto, encerrado entre corchetes.

Cuando el adaptador trabaja con archivos que se han definido mediante caracteres comodín, el nombre del archivo cambia constantemente. Puesto que el nombre en el archivo schema.ini no puede contener caracteres comodín, el adaptador debe cambiar el archivo schema.ini antes de conectarse a la base de datos.

Por lo tanto, es necesario agregar una línea de indicador delante de la línea de entrada de archivo. Esta línea de indicador contiene el patrón de nombres del elemento de cadena de conexión en el archivo de configuración del adaptador, encerrado entre corchetes. El adaptador reemplaza la línea siguiente con el nuevo nombre de archivo encerrado entre corchetes.

Nota: El archivo schema.ini puede contener varias entradas. Es su responsabilidad agregar las líneas en el lugar correcto.

Ejemplo de schema.ini

```
[sqltes*.txt]
[sqltest2.txt]
ColNameHeader = False
CharacterSet = ANSI
Format = TabDelimited
Col1=id Char Width 30
Col2=idname Char Width 30
----------------------------------------
[report_200*.txt]
[report_2003_09.txt]
ColNameHeader = False
CharacterSet = ANSI
Format = TabDelimited
Col1=date Char Width 30
Col2=service Char Width 30
Col3=responseTime Char Width 30
----------------------------------------
```

Archivo de control de origen de datos

Para cada consulta, el archivo de control de origen de datos contiene el patrón de nombres de archivo y el nombre del archivo de entrada actual para continuar con el mismo archivo cuando se reinicie el adaptador.

A continuación se incluye un ejemplo del archivo de control en su 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>
```

Conservación de archivos de entrada

Cuando el adaptador ha terminado de leer el archivo actual, busca el siguiente. La siguiente lectura de archivo corresponderá al primer archivo que se ajuste al patrón de nombre y cuyo nombre siga al nombre del archivo anterior (en orden lexicográfico). El adaptador no vuelve a archivos anteriores, aunque se le agreguen nuevos registros.

El adaptador utiliza el atributo InitialFileName solamente cuando estos dos atributos corresponden a "no" y el archivo de control no contiene el nombre del último archivo.

Comprobación de consultas

El adaptador comprueba la cadena de conexión y la consulta solamente si existe el archivo definido. Si se define mediante caracteres comodín, el adaptador lo comprueba solamente en relación con el primer archivo.

Se pueden producir errores cuando el adaptador intente leer a partir de un archivo nuevo. En este caso, el adaptador se detiene inmediatamente si el atributo "critical" es igual a "yes". Si es igual a "no", el adaptador no continúa realizando esta consulta, sino que procede con las otras consultas. El adaptador deja el archivo actual y pasa al archivo siguiente.

Variables internas del adaptador

Hay dos variables internas que se pueden utilizar en el archivo de configuración para hacer referencia al contexto actual del nombre de archivo. Estas son ":file_name" y ":file_name_no_ext", y hacen referencia respectivamente al nombre de archivo actual y el nombre de archivo actual sin la extensión de archivo.

Estas variables se pueden utilizar en el elemento SelectStatement, en el elemento SelectInitialValues y en el atributo QueryKeyField\Function.

El adaptador reemplaza la variable con el nombre de archivo en las consultas.

Por ejemplo:

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

Creación de un adaptador mediante la interfaz de usuario de CA Business Service Insight

Cada adaptador configurado para ejecutarse dentro del entorno de CA Business Service Insight deberá estar registrado dentro de la interfaz de usuario, además de definirse en el registro. El motivo para ello es principalmente establecer los valores de configuración del lado de escucha del adaptador para que esté preparado para recibir eventos por parte del adaptador. Durante este paso se definen todos los valores de configuración del adaptador, como las tablas de traducción y los tipos de eventos.

Siga estos pasos:

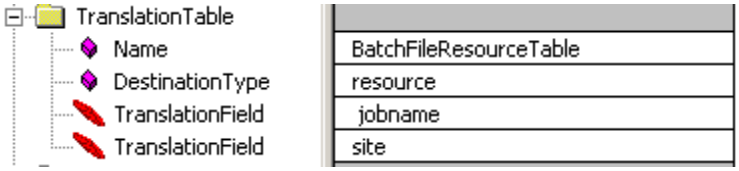
1. Cree el adaptador.
2. Elija Recursos/Adaptadores.
3. Compruebe los adaptadores existentes en la lista para asegurarse de que ninguno se haya definido en el mismo puerto que su adaptador, es decir, el mismo puerto que se ha definido en el archivo de configuración del adaptador en el directorio OblicoreInterface\OnlineInterface\Port.
4. Haga clic en Agregar nuevo/a y, a continuación, seleccione el método que desee utilizar para crear el adaptador. Hay varias opciones:
    a. **Crear manualmente**: establece la escucha del adaptador para conectarse al adaptador que se ha definido (o definirá) manualmente.
    b. **Mediante el asistente**: permite crear el adaptador mediante una interfaz de asistente pantalla a pantalla. Consulte la sección siguiente para obtener más información al respecto.
    c. **A partir de una plantilla**
    d. **A partir de un archivo de configuración existente**: permite cargar una plantilla de adaptador preconfigurada que rellenará automáticamente los campos del asistente del adaptador con las opciones establecidas en el archivo de configuración especificado.
    e. La pantalla resultante varía según la opción elegida.
5. Rellene los campos:

    En Dirección de red, introduzca la dirección IP del adaptador. "localhost" si es local con respecto al servidor de aplicaciones; de lo contrario, introduzca el puerto definido.
6. Haga clic en Guardar.

Para crear las tablas de traducción necesarias:

Nota: Estos pasos se deberán realizar para cada tabla de traducción definida en el archivo de configuración:

1. En el menú Diseño, haga clic en Traducción, Tablas de traducción y, a continuación, seleccione el botón "Agregar nuevo/a".
2. Todos los valores de configuración en la tabla de traducción deberán corresponderse con la definición de la tabla equivalente en el archivo de configuración del adaptador:
   - *Nombre*: deberá coincidir con el atributo de nombre en la tabla de traducción del archivo de configuración, Name.
   - Campos de origen: deberá tener todos los campos de TranslationField de la tabla de traducción. Agregue todos los campos. El valor de traducción puede estar formado por una combinación de dos o más campos. El siguiente es un ejemplo del aspecto que podría tener:



3. Tipo de destino: deberá corresponderse de nuevo con el atributo DestinationType de la definición de tabla de traducción en el archivo de configuración (resource, event_type, etc.).
4. Adaptadores registrados: agregue los adaptadores que usarán esta tabla de traducción. Varios adaptadores pueden usar la misma tabla de traducción.
5. Haga clic en Guardar.
6. Importe la definición de campos de tipo de evento para cada tipo de evento.

   Para poder importar la definición de campos de un archivo de configuración de adaptador específico, este adaptador deberá haberse ejecutado y haberse conectado a CA Business Service Insight como mínimo una vez. Cuando el adaptador se conecta a CA Business Service Insight, envía el archivo de configuración a CA Business Service Insight, lo cual permite al sistema utilizar la definición de campos a partir de dicho archivo de configuración.

   Nota: También se pueden especificar las definiciones de campo manualmente para el tipo de evento.
7. Active el adaptador:
   a. En el menú Diseño, haga clic en Adquisición de datos, Adaptadores.
   b. Haga clic en el botón Inicio correspondiente al adaptador.

La tabla siguiente contiene los diferentes estados de los adaptadores:

Estado	Descripción
Inactivo	Estado inicial. El adaptador está inactivo y no se ha iniciado todavía.
Escucha inactivo	El servicio de escucha del adaptador (emisor) no se ha iniciado.
Inicio	El adaptador se está iniciando.
Iniciado	El adaptador se ha iniciado.
Parada	Deteniéndose.
Pausar	Pausándose.
Pausado	Pausado:
No está en ejecución	El adaptador no se está ejecutando en el equipo del adaptador.
Error	Error en el archivo de configuración del adaptador; no se puede iniciar el adaptador.
Error de conexión	Se ha producido un error al conectar el adaptador (host o puerto equivocado), o antes de que el adaptador se haya ejecutado por primera vez y sin que se haya detectado ninguna señal. Estado al ejecutar el adaptador por primera vez.

Bloqueado	Se ha alcanzado el número máximo de eventos rechazados.

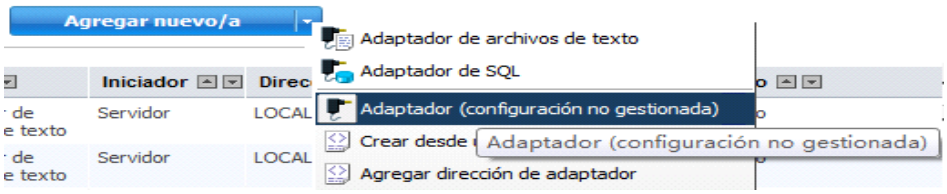
Creación de un adaptador mediante el asistente del adaptador

El asistente del adaptador es una función nueva de la interfaz de usuario de CA Business Service Insight que puede orientarle a la hora de crear un adaptador nuevo, pues usa una interfaz más intuitiva que el editor de XML. El asistente le guía por una serie de fichas que contienen toda la información necesaria para crear un adaptador, y al final le permite descargar una copia del archivo de configuración XML recopilado. Existen, sin embargo, algunas limitaciones en cuanto a lo que el asistente puede hacer.

Actualmente no permite:

- Hacer referencia a la misma estructura de entrada (formato de entrada) desde diferentes interfaces de orígenes de datos.
- Hacer referencia a la misma estructura del resultado (traductor) a partir de entradas diferentes.
- Configurar un conmutador de formato de entrada y utilizarlo para decidir qué formato de entrada utilizar desde la interfaz de origen de datos.
- Proporcionar una definición de un campo de ID de origen de datos en la estructura del resultado.
- Proporcionar una definición de más de un archivo en la cadena de conexión para consultar archivos de texto o de Excel con la misma consulta.
- Utilizar restricciones de hora UtcNow o UTC.
- Especificar valores que empiecen o acaben con un "espacio".

Al crear un adaptador nuevo a través del asistente dispondrá de cuatro opciones, como muestra la ilustración siguiente:



Las primeras dos opciones le permiten crear un adaptador de archivos de texto o un adaptador de SQL mediante la interfaz de asistente. La siguiente opción de adaptador (configuración no gestionada) es la opción que permite elegir cuándo añadir un adaptador preconfigurado que se haya creado mediante el editor de XML. Esta opción no permitirá modificar más adelante la configuración de este adaptador mediante el asistente. La última opción, Crear desde un archivo de configuración, le permite cargar una configuración de adaptador preconfigurada y hacer que el sistema la importe a la interfaz del asistente para hacer más modificaciones. Esta opción exige que no exista ninguna de las limitaciones de asistente mencionadas arriba en la configuración en cuestión.

A partir de este punto, las opciones de configuración del asistente del adaptador proporcionan la misma funcionalidad descrita para la configuración de adaptador manual. Su intención es proporcionar una interfaz más sencilla y más fácil para editar los valores de configuración. Consulte las secciones relevantes para obtener más información, ya que las funciones y los principios aplicables son los mismos que en la opción manual.

Ejecución y prueba del adaptador

El archivo de configuración del adaptador no se puede terminar de configurar normalmente en un solo ciclo. Es posible que haya que ejecutar el adaptador varias veces, así como comprobar los resultados para asegurarse de que la configuración del adaptador sea la correcta.

A continuación se incluyen algunos de los problemas más comunes que suele haber que corregir:

- Problemas de conexión (cualquiera entre el origen de datos y el adaptador o entre el adaptador y la escucha)
- Errores en el archivo de configuración, como:
  - Estructura equivocada
  - Uso equivocado de atributos
  - Uso equivocado de mayúsculas o minúsculas (los nombres de los adaptadores distinguen entre mayúsculas y minúsculas, "Yes" y "yes" son diferentes, etc.)
  - Asignación de valores equivocada
- Errores de manipulación de datos, como estructura de evento de salida, valores de evento equivocados y errores en consultas.

Siga estos pasos:

1. Establezca el adaptador en RunOnce = "yes" y LogDebugMode = "yes", así como el valor RejectedEventsUpperLimit en un número razonable (consulte Modos de ejecución del adaptador (en la página 100)).

   La ilustración siguiente muestra los valores de configuración necesarios para hacer pruebas.

2. También se puede utilizar el modo sin conexión para establecer los valores del archivo de configuración.
3. Una vez el archivo de configuración se haya cargado correctamente, cambie la configuración para volver a ponerla en línea (consulte los modos de ejecución del adaptador).
4. En cada repetición se realizan los pasos siguientes:
   a. Actualice o corrija el archivo de configuración del adaptador.
   b. Suprima todos los archivos de salida del adaptador.
   c. Ejecute el adaptador haciendo doble clic en el acceso directo al archivo ejecutable del *adaptador* o en el archivo *.bat* que se haya creado.

d. Abra el archivo de registro del adaptador mediante el explorador del registro (utilidad Browser.exe del registro en las utilidades de CA Business Service Insight) y compruebe que no haya ningún mensaje de error.

5. El primer paso es obtener un archivo de configuración correcto y llegar a un estado en el cual el adaptador cargue el archivo de configuración correctamente. Repita este paso hasta que se haya conectado correctamente a CA Business Service Insight y al origen de datos y disponga de eventos rechazados y solicitudes de traducción.
6. Para completar esta etapa, compruebe lo siguiente:
   - Que no haya ningún mensaje de error en el archivo de registro del adaptador.
   - Que el adaptador se conecte a CA Business Service Insight y al origen de datos correctamente.
   - Que el adaptador envíe solicitudes de traducción y eventos rechazados.

   Para cada registro que el adaptador haya rechazado, verá la letra "R" en la consola. Recuerde que se espera que aparezcan eventos rechazados hasta que hayan finalizado todas las traducciones necesarias.
7. Compruebe que el archivo rejectedEvents contenga registros y no esté vacío.

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

Traducciones de eventos y recursos

En el paso anterior, había varios eventos rechazados creados al ejecutar el adaptador. Estos eventos rechazados se almacenaron en un archivo RejectedEvents.txt, pero se almacenaron también como entradas de traducción pendientes en la base de datos de CA Business Service Insight. El paso siguiente del proceso de cargar los datos sin procesar en el sistema es proporcionar una traducción de lo que se ha medido, para que CA Business Service Insight pueda utilizar estos datos según sea necesario.

Los eventos de datos sin procesar se pueden rechazar porque no se ha definido en el sistema el tipo de evento o el recurso. Los eventos pendientes se definen según la tabla de traducción de donde procedan. Los ejemplos más habituales son que haya tipos de evento que procedan de una tabla de traducción y recursos que procedan de otra tabla de traducción.

Cuando el adaptador encuentra un nuevo recurso (digamos por ejemplo que se agrega un servidor nuevo a la herramienta de control de la red y aparece como una nueva entrada en el origen de datos de esta herramienta de control), se puede agregar al modelo de recurso del sistema. Se deben realizar dos pasos para que este nuevo recurso se incluya en los informes de CA Business Service Insight.

Primero es necesario crear el recurso como una entidad de CA Business Service Insight (un recurso) y, a continuación, se debe introducir una traducción. Esto proporciona el vínculo entre la representación de la cadena encontrada en el origen de datos y la entidad definida como un recurso en CA Business Service Insight. Este proceso de dos pasos se puede realizar en una sola acción a través de la interfaz de usuario en un proceso conocido como "Agregar y traducir", por el cual el nuevo recurso y la entrada de traducción necesaria se pueden crear con un solo procedimiento. Al agregar y traducir, se pueden seleccionar varias entradas, siempre que tengan los mismos valores de adjudicación. La traducción es el proceso de crear la coincidencia entre el valor del origen de datos y la entidad de CA Business Service Insight. Al crear traducciones, se agrega una entrada a la tabla de traducción con los valores coincidentes. A continuación, en las consultas subsiguientes al origen de datos, el adaptador sabrá automáticamente cómo gestionar este nuevo valor.

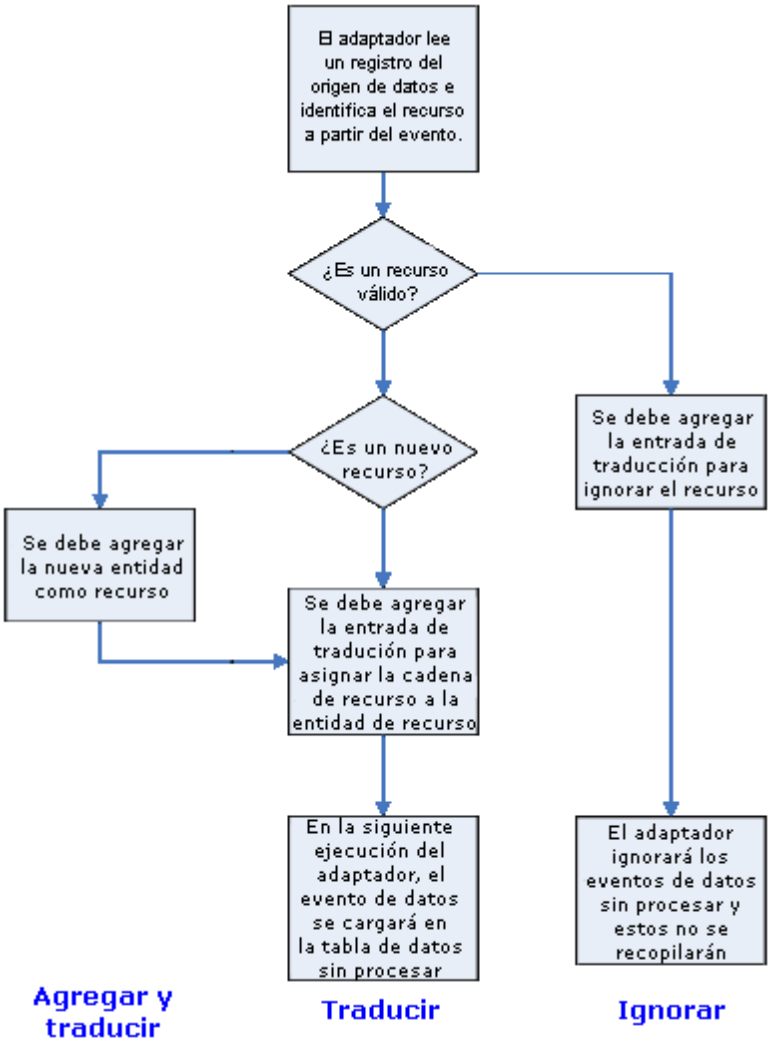
En esta etapa, el adaptador ha ejecutado y enviado ya solicitudes de traducción para cada valor en los campos de traducción. Los eventos asociados a estos valores se han rechazado y están en espera de enviarse a CA Business Service Insight una vez que terminada la traducción. Las traducciones se pueden hacer manualmente o automáticamente utilizando un script de traducción.

Estas son las acciones de traducción posibles en el caso de entradas de traducción pendientes:

Traducir: permite crear una entrada en la tabla de traducción haciendo coincidir un valor de origen de datos con la entidad de CA Business Service Insight relevante. La entidad de CA Business Service Insight para la cual se hacen las traducciones debe existir previamente. (Un buen ejemplo de esto puede ser el caso de un recurso escrito mal en el origen de datos. Puede haber nombres diferentes en el origen de datos que están haciendo referencia en realidad a una sola entidad lógica. Por ejemplo, Server001 y SERVER001. Los recursos de CA Business Service Insight distinguen entre mayúsculas y minúsculas).

- **Agregar y traducir**: permite crear una nueva entidad en CA Business Service Insight *y* al mismo tiempo añade una entrada de traducción para esa entidad en la tabla de traducción. Esta es la acción más habitual que se realiza en las entradas de traducción pendientes, ya que el mecanismo de traducción se utiliza para construir la infraestructura en CA Business Service Insight.
- **Ignorar**: al ignorar una entrada, todos los eventos asociados se ignoran y no se envían a la tabla de datos sin procesar de CA Business Service Insight. Los eventos ignorados se perderán. Por ejemplo, si el origen de datos contiene información acerca de todos los servidores en un centro de datos, pero solamente hacen falta los datos de los servidores de aplicaciones para calcular el nivel de servicio, todos los servidores llegarán al adaptador para traducirse, si bien solamente se traducirán los servidores de aplicaciones. Los demás se ignorarán, ya que CA Business Service Insight se asegura de que el evento no necesario también se ignore. Las entrada ignoradas se pueden traducir en una etapa posterior si es necesario, pero solamente se capturarán estos datos desde ese punto en adelante.
- **Suprimir**: se suprime la entrada de traducción y también el evento rechazado asociado. Si el origen de datos envía el mismo valor más tarde, se crea una nueva entrada pendiente.

El diagrama de flujo siguiente resume los casos de uso de estas opciones:


El adaptador lee un registro del origen de datos e identifica el recurso a partir del evento.
¿Es un recurso válido?
¿Es un nuevo recurso?
Se debe agregar la nueva entidad como recurso
Se debe agregar la entrada de tradución para asignar la cadena de recurso a la entidad de recurso
En la siguiente ejecución del adaptador, el evento de datos se cargará en la tabla de datos sin procesar
Se debe agregar la entrada de traducción para ignorar el recurso
El adaptador ignorará los eventos de datos sin procesar y estos no se recopilarán
Agregar y traducir
Traducir
Ignorar

Traducciones manuales

Se requieren traducciones manuales cuando la entidad ya existe dentro de CA Business Service Insight. Esto puede ocurrir en varias situaciones. Por ejemplo, cuando se ha ejecutado un script de traducción para crear automáticamente los recursos a partir de un origen externo (pero las traducciones no se han podido automatizar). O cuando un origen de datos incluye entradas dudosas y se ha definido un recurso dos veces de maneras diferentes (por ejemplo, Server001p y server001P). Podría incluso deberse al hecho de que hubiera recursos creados manualmente.

Para ejecutar una traducción manual cuando el recurso ya existe:

1. En el menú Diseño, haga clic en Traducciones, Entradas de traducción.
2. De forma predeterminada, aparecen todas las entradas pendientes.
3. Seleccione la entrada de traducción pendiente que desea traducir marcando la casilla de verificación junto a ella.
4. Haga clic en Traducir.
5. Elija la entidad relevante (recurso, tipo de evento, etc.) en la lista. (Si la lista no muestra ningún elemento puede ser necesario cambiar los criterios de búsqueda predeterminados en el recuadro).
6. Seleccione el tipo de evento o recurso en la lista de entidades haciendo clic en la línea que contiene el elemento. Una vez seleccionado, permanece resaltado.
7. Haga clic en Traducir. La entrada de traducción se almacenará ahora en el sistema.

Para ejecutar una traducción manual cuando el recurso NO exista aún:

1. En el menú Diseño, haga clic en Traducciones, Entradas de traducción.
2. De forma predeterminada, aparecen todas las entradas pendientes.
3. Seleccione la entrada de traducción pendiente que desea convertir a un recurso de CA Business Service Insight y traduzca seleccionando la casilla de verificación junto a ella.
4. Haga clic en Agregar y traducir.
5. Asegúrese de que el nombre del recurso aparezca especificado correctamente. Se puede personalizar también el campo *Mostrar nombre* para cambiar la forma en la que el recurso aparece en un informe. Si este recurso se debe gestionar como parte de un "conjunto de cambios", el conjunto de cambios concreto se deberá especificar aquí también.
6. La fecha de vigencia del recurso se deberá configurar como la fecha en la cual se deberá informar de este recurso desde el sistema. Tenga en cuenta que el recurso NO aparecerá en informes anteriores a la fecha especificada aquí.
7. Haga clic en la ficha Detalles y seleccione las opciones siguientes para la asociación del recurso: Vigente (verdadero/falso), Tipo de recurso, Pertenencia a grupo de recursos y Asociación de contrato y servicio.
8. Haga clic en Guardar y traducir. El recurso se agrega al catálogo de servicios de recursos y la entrada de traducción se almacena ahora también en el sistema. Una vez gestionadas todas las entradas pendientes, podrá comprobar que los datos se cargan en el sistema:
9. Navegue a los recursos en CA Business Service Insight y compruebe que el nuevo recurso se haya creado.
10. Ejecute el adaptador otra vez.
11. Compruebe que el archivo de eventos rechazados esté vacío, en contenido y tamaño.
12. Compruebe que el archivo de envío de eventos esté vacío, en contenido y tamaño.
13. Compruebe con la herramienta de datos sin procesar que los eventos se hayan agregado a los datos sin procesar.

Configuración de la infraestructura

La configuración de la infraestructura incluye la definición de las entidades de modelado de datos como se identificaron durante la fase de diseño.

Esta etapa no incluye la definición de todas las entidades de infraestructura, y se termina cuando se termina de configurar el adaptador.

Durante esta etapa se suministran las entidades siguientes al sistema de CA Business Service Insight:

- Tipos de evento
- Tipos de recurso
- Recursos y adjudicaciones de recursos
- Grupos de recursos

Nota: Una vez el adaptador se ha ejecutado correctamente, se puede utilizar la función de importación automática al definir los campos de tipo de evento.

Traducciones automáticas con scripts de traducción

La traducción automática corresponde a una automatización de los procesos de creación y traducción y de la infraestructura, basada en un origen de datos externo mediante scripts que ejecutan y realizan las acciones de traducción.

La traducción automática se realiza a través de scripts de traducción. Los scripts de traducción aceleran el proceso de asignación de recursos empresariales e informáticos nuevos a CA Business Service Insight. El script de traducción identifica automáticamente cuándo se recibe una nueva entrada de traducción y traduce los recursos, lo cual permite asignar el recurso de manera rápida y eficaz. La automatización es compatible con la interfaz a los CMDB, lo cual permite al sistema identificar recursos basándose en la definición configurada. La traducción automática tiene una serie de ventajas, incluida la sencillez del mantenimiento de traducciones y la posibilidad de evitar errores. También se pueden utilizar scripts de traducción para crear nuevos recursos y adjudicar cambios.

Además, los scripts de traducción se puede usar para:

- Traducir entradas a objetos de CA Business Service Insight existentes.
- Agregar nuevos objetos de CA Business Service Insight y traducirlos basándose en entradas de traducción existentes.
- Crear objetos y traducir entradas en base a tablas externas a CA Business Service Insight, por ejemplo tablas de recursos de otro CMS externo.
- Comprobar si existe un objeto.
- Crear recursos y adjudicar recursos, como tipos de recurso, grupos de recursos, partes contratantes y componentes de servicios.
- Adjudicar o anular adjudicaciones a conjuntos de cambios.

Puesto que el proceso de traducción también se puede realizar manualmente en la interfaz de usuario, es necesario decidir qué proceso de traducción elegir. Al hacerlo, tenga en cuenta las ventajas y los inconvenientes de la traducción automática:

- Los scripts de traducción para proyectos más complejos requieren los conocimientos adecuados y bastante trabajo en el desarrollo del script.
- Deberá añadir tiempo de desarrollo así como tiempo para control de calidad en las pruebas.
- No se necesita implementar en aquellos casos en los que el proceso de traducción requiere un trabajo meramente inicial.
- Se puede programar para implementarse como parte del acercamiento a la segunda fase.
- Requiere un mantenimiento menor.
- Los errores de traducción humanos son menos.
- Para obtener más información acerca de cómo crear y ejecutar scripts de traducción, consulte la *Guía* SDK en el capítulo de integración de CMDB.

Temas avanzados de la función de adaptador

Los temas siguientes tratan sobre temas avanzados de la función de adaptador.

Singularidad del evento

La singularidad del evento es un mecanismo de adaptador que proporciona un proceso que previene la inserción de datos sin procesar duplicados en la tabla T_Raw_Data.

Sin singularidad del evento, cuando el adaptador se ejecute para un origen de datos y cargue eventos en la base de datos, no se validaría la presencia de eventos idénticos. La función Singularidad del evento resuelve este problema proporcionando la posibilidad de especificar si comprobar o no la singularidad de los eventos antes de insertarlos y qué hacer si se produce esta situación. No obstante, este proceso de verificación puede afectar negativamente al rendimiento del adaptador.

La solución permite al usuario definir una clave, que puede basarse en los campos del evento. Dicha clave representa la unicidad del evento, lo que significa que, si hay dos eventos con la misma clave, son eventos idénticos.

También es posible especificar qué acción llevar a cabo si se encuentra una clave duplicada en la base de datos. Más adelante se tratan estas acciones.

Nota: La clave se puede definir como una combinación de varios campos desde el traductor.

Archivo de configuración del adaptador con singularidad del evento

TranslatorCollection/Translator/OnDuplication

Este campo determina qué se deberá hacer si el evento ya existe en la base de datos.

Los valores posibles son:

- **Add**: inserta eventos (adicionales) en la base de datos.
- **Update**: suprime (en algunos casos) el evento existente y la inserción de un evento nuevo después de validar el cambio del evento.
- **updateAlways**: suprime (en algunos casos) el evento existente y la inserción de un evento nuevo sin buscar cambios.
- **Ignore**: hace que no se inserte ningún evento nuevo en la base de datos.

El valor predeterminado es "Add".

"Ignore" frente a "Add"

Al ejecutar el adaptador en el modo "Ignore", puede producirse una leve disminución del rendimiento. Sin embargo, con el modo "Add", el sistema activa el recálculo en todos los casos de eventos duplicados.

"Update" frente a "updateAlways"

Utilizar la opción "Update" reduce el rendimiento del adaptador, mientras que "updateAlways" reduce el rendimiento del cálculo al activar el recálculo de cada caso.

TranslatorCollection > Translator > TranslatorFields > TranslatorField > IsKey

Este atributo determina si el valor del campo al cual pertenece deberá contribuir a proporcionar una clave única para los datos sin procesar. Se incluye si el valor es "yes", y no se incluye si el valor es "no". El valor predeterminado (si no se indica ningún valor) es "yes" para campos estándar (ResourceId, EventTypeId y Timestamp) y "no" para el resto. La clave se deberá elegir cuidadosamente para garantizar que los datos sin procesar mantengan la integridad necesaria y que los cálculos sean totalmente precisos.

Comportamiento de escucha del adaptador con singularidad del evento

Al recibir un nuevo evento del adaptador, la escucha comprueba el valor del campo OnDuplication. Cuando el valor es "add", se lleva cabo el proceso de inserción normal. Cuando el valor no es "add", la escucha busca la existencia de un evento con el mismo valor UniqueKey y el mismo ID de lector en la base de datos. Si la base de datos ya contiene un evento según lo descrito, el nuevo evento no se inserta en la base de datos cuando el valor de OnDuplication es "ignore".

Cuando el valor de OnDuplication es "update", se comprueban los cambios en el evento. Si todos los campos son idénticos, el nuevo evento no se inserta en la base de datos.

Cuando el valor de OnDuplication es "updateAlways", se omite la comprobación anterior y se actualiza siempre.

En los modos tanto "update" como "updateAlways" se realizan los pasos siguientes:

- Si se ha cambiado ResourceId, EventTypeId y Timestamp, se hace un recálculo de todas las métricas que utilizan la fórmula del evento antiguo.
- El evento se actualiza con los valores apropiados.

Agregación de datos de transacciones

Los datos de las transacciones a menudo se recopilan para compararlos con determinados umbrales o para poder calcular el porcentaje periódico de éxito final. Por ejemplo, cada cinco minutos se registra una transacción virtual en un sistema y el resultado (tiempo de respuesta en milisegundos) se almacena como se indica a continuación:

```
[…]
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
[…]
```

En otras situaciones, en lugar de utilizar transacciones virtuales, se puede tener acceso a transacciones reales en un sistema. En estos casos, se pueden realizar cientos o incluso miles de transacciones cada hora.

En los dos casos anteriores, debe evitarse cargar tal volumen de información en CA Business Service Insight, si se puede.

Agregar los datos por períodos de tiempo es la mejor forma de reducir la cantidad de datos. Cuando se fija el umbral de acuerdo al cual se va a medir el éxito, el proceso de agregación se puede llevar a cabo permitiendo al adaptador contar el número de transacciones dentro del período agregado que ha tenido éxito. Por ejemplo, si el umbral de éxito en el ejemplo anterior se establece en 500 milisegundos, solamente cinco de cada siete transacciones tuvieron éxito dentro del período mostrado. El problema con este enfoque es el umbral fijado: ¿qué pasaría si hubiera que cambiar el umbral más adelante? En este caso, el adaptador deberá releer y probar los datos sin procesar comparándolos con el nuevo umbral.

Por lo tanto, lo ideal sería que el adaptador agregase los datos de las transacciones de una manera flexible y sin perder datos significativos.

Una solución limitada es permitir al adaptador probar las transacciones comparándolas con varios umbrales. Hay muchas formas de hacerlo:

- Un evento con varias pruebas: tipo de evento = {TotalTransactions, TransBelow250, TransBelow500, TransBelow750, [...]}
- Varios eventos con una prueba: tipo de evento = {TotalTransactions, Threshold, TransBelowThreshold}.

Las dos opciones tiene el mismo problema: los umbrales se podrán cambiar en el futuro solamente dentro de un conjunto pequeño de valores predefinidos.

Solución recomendada

Supongamos que todos los umbrales potenciales son múltiplo de cierto número. Por ejemplo, todos los umbrales (en milisegundos) son múltiplo de 250, de modo que 0, 250, 500, 1750 y 3000 milisegundos son todos umbrales potenciales.

De acuerdo con este supuesto, la solución sugerida es agregar transacciones redondeando todos los valores al múltiplo común, y contar cuántas transacciones se incluyen dentro de cada redondeado. Tipo de evento = {RangeFrom, RangeTo, TransactionCount}

Por ejemplo, se generarán los eventos siguientes para agregar los datos mostrados anteriormente, donde el múltiplo común es 250 milisegundos:

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

Comentarios:

La marca de tiempo de esos eventos sería la misma. (Por ejemplo, todos los eventos agregados pueden ocurrir el 1/1/2007 00:00., y puede haber otro conjunto de eventos para la siguiente muestra de hora el 1/1/2007 01:00 en el supuesto de haya agregación cada hora).

RangeTo se calcula redondeando una transacción al múltiplo común (vea la información a continuación).

RangeFrom es RangeTo menos el múltiplo, más 1. Se especifica para proporcionar claridad solamente; se puede omitir.

Por ejemplo, una agregación por cada hora tendría un aspecto similar a lo siguiente (reemplace MULTIPLE por el valor del 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
```

En la lógica de negocios, se puede aplicar a los eventos la condición siguiente:

```
If eventDetails("RangeTo")<=Threshold Then
```

```
        SuccessfulTransactions=SuccessfulTransactions+eventDetails("TransactionCo
unt")
End If
```

Algunos datos decisivos:

- Dado que las transacciones tienden a distribuirse con normalidad, el número de eventos agregados deberá ser relativamente bajo.
- Al seleccionar múltiplos comunes más altos se obtendrán menos eventos agregados.
- El volumen de eventos agregados deberá ser siempre menor o igual al volumen de datos sin procesar.

Ejecución de un adaptador tras un cortafuegos

Para poder ejecutar un adaptador tras un cortafuegos se sugiere recurrir a la solución siguiente:

Abra dos puertos; el primer puerto es el puerto asignado al adaptador en CA Business Service Insight; el segundo puerto, opcional pero recomendable, es el puerto del servidor de registros. El puerto del servidor de registros se define de forma predeterminada en el puerto 4040. Abrir el puerto del servidor de registros permite al adaptador informar al registro de CA Business Service Insight y también generar alertas. Esto es opcional porque el adaptador siempre envía la información a un archivo de registro local.

Para los dos puertos, el protocolo en uso es TCP/IP.

Carga de datos de varios directorios

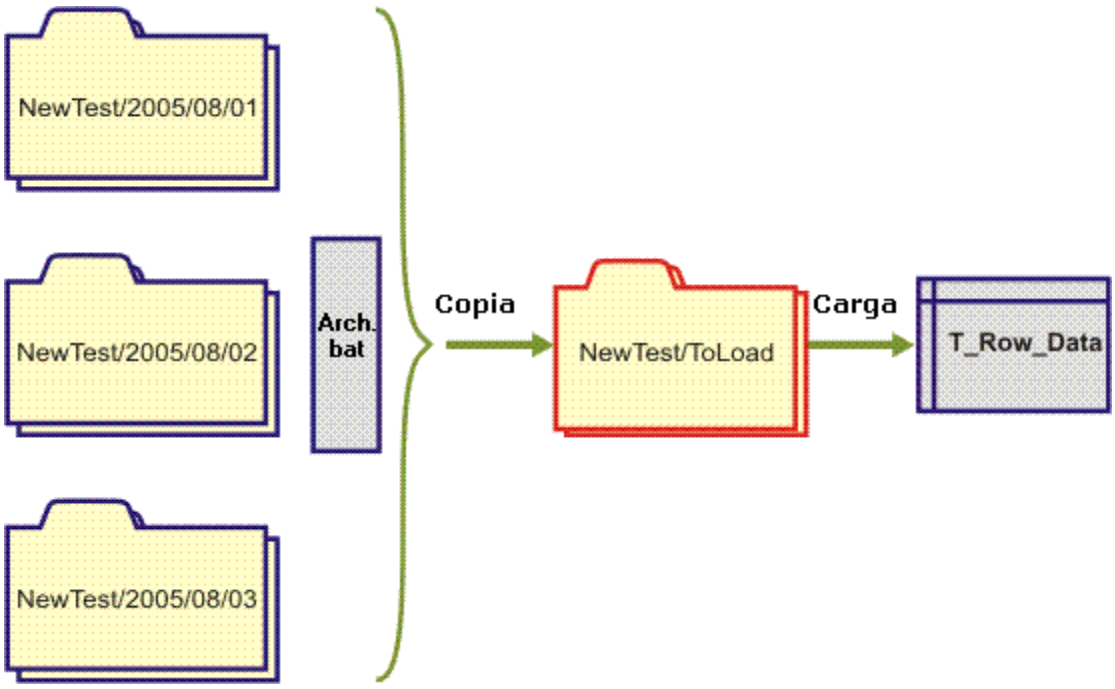
Esta sección describe una solución que se puede implementar si los archivos de origen de datos (entrada a un adaptador de CA Business Service Insight) se encuentran en directorios diferentes cada día, o para todos los períodos de tiempo definidos (según una convención de denominación específica).

Por ejemplo, si tenemos la estructura de directorios c:\NewData\AAAA\MM\DD. En esta instancia, para cada día hay un directorio nuevo donde se encuentran los archivos relevantes para ese día.

El adaptador de CA Business Service Insight tiene que explorar el directorio más nuevo y cargar los nuevos archivos.

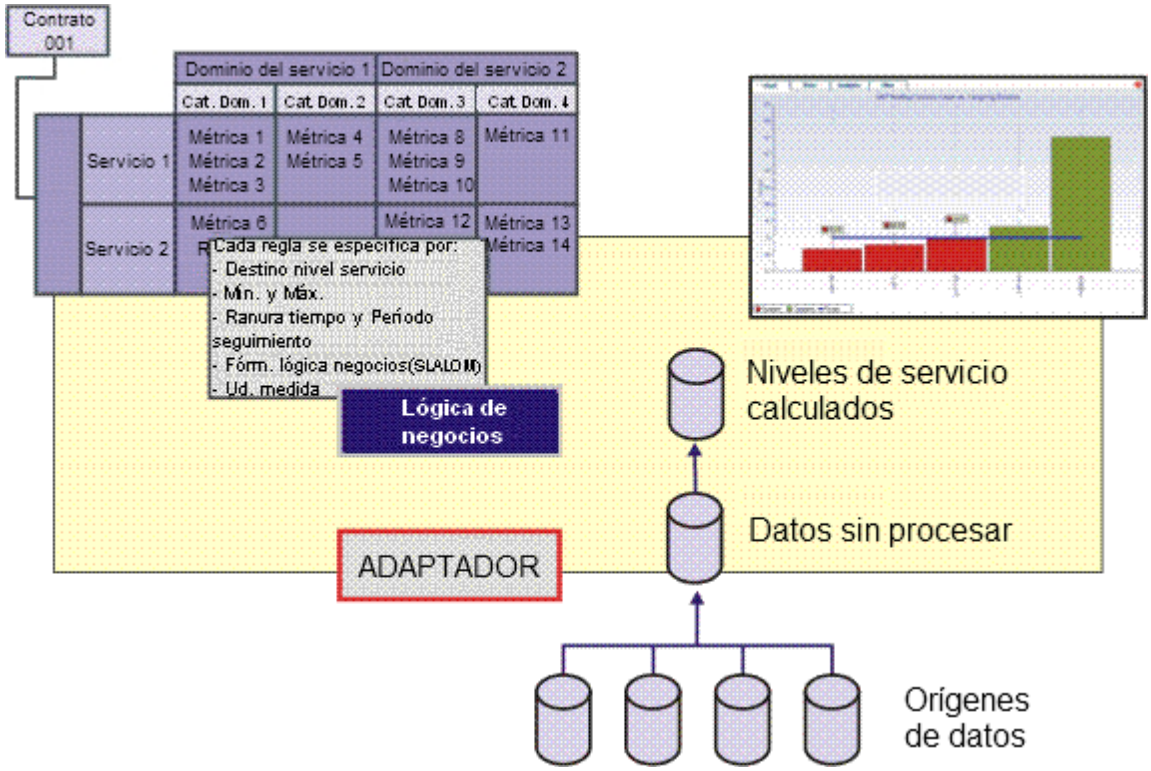
Una solución temporal podría ser agregar algunos comandos al principio del archivo bat que ejecuta el adaptador. Estos comandos copian los orígenes de datos que tienen que cargarse desde la carpeta específica (según sus convenciones) a una sola carpeta dedicada creada específicamente para este fin. El adaptador siempre carga la información desde esta carpeta.

La imagen siguiente describe esta solución:



Scripting de lógica de negocios (experto en la lógica de negocios)

La imagen siguiente describe la posición de lógica de negocios dentro de CA Business Service Insight. Se encuentra detrás de cada métrica dentro de los contratos.



En esta etapa de la implementación, se han configurado los adaptadores relevantes y los registros de datos sin procesar deberán estar ya disponibles en la tabla T_RAW_DATA de CA Business Service Insight. Ahora es posible aplicar la lógica de negocios a los eventos para producir el resultado del nivel de servicio real para cada métrica.

El scripting de lógica de negocios es el proceso de escribir código que manipula de manera lógica los datos sin procesar (datos sin procesar que envían los adaptadores) para poder calcular los niveles de servicio.

Cada métrica tiene su propia fórmula de lógica de negocios con la que se puede calcular el nivel de servicio real, aunque muchas métricas del sistema suelen tener una lógica común que se puede aplicar a diferentes conjuntos de eventos de datos sin procesar.

Por ejemplo, una métrica que calcula el tiempo de resolución de tickets de gravedad 1 y otra métrica que calcula el tiempo de resolución de tickets de gravedad 2 evalúan un conjunto diferente de registros: uno utiliza solamente tickets de gravedad 1 y el otro solamente tickets de gravedad 2. Sin embargo, los dos aplicarían probablemente el mismo método para calcular el tiempo de resolución. El script de hora de resolución se desarrollará y probará una vez, definido como módulo de lógica de negocios, y a continuación lo utilizarán las dos métricas incluyendo este módulo en las áreas de lógica de negocios de métricas.

Por lo tanto, al desarrollar scripts de lógica de negocios los módulos principales o las plantillas normalmente se desarrollan para ponerlos a disposición de todas las métricas del sistema. Además, cada categoría de dominio normalmente refleja un tipo diferente de medición y, por lo tanto, cada categoría del dominio seguirá generalmente un módulo de lógica de negocios o una plantilla diferente.

Flujo de trabajo del scripting de lógica de negocios

La etapa de scripting de lógica de negocios conlleva los pasos siguientes:

- Definir una fórmula.

  Crear la fórmula basándose en los requisitos de cálculo definidos en la fase de diseño. Las fórmulas definidas son todas las fórmulas únicas que se tienen que utilizar en sus diversas permutaciones en las métricas del contrato, cada una como un módulo de lógica de negocios.

  Por ejemplo, si el contrato tiene tres métricas para calcular el tiempo de resolución promedio por ticket y una métrica para cada prioridad de ticket, entonces se desarrolla una sola fórmula para calcular el tiempo de resolución por ticket, con la prioridad de los tickets como parámetro. Esta fórmula, una vez probada, se define como módulo y se adjunta a todas las métricas relevantes.

- Probar la fórmula.

  Se realizan pruebas para garantizar que la fórmula se ha definido correctamente y sin errores, y que los cálculos producen el resultado esperado. En el proceso de elaboración de pruebas, es importante cubrir todos los casos extremos y las condiciones límite. El ámbito de la lógica de negocios es donde la fórmula se ejecuta con fines de pruebas. Una vez en definida de manera inicial, la fórmula se prueba en su totalidad. A continuación, tras aplicarla a todas las métricas como un módulo, es importante ejecutar cada métrica en el ámbito como mínimo una vez para ver si recibe los eventos (es decir, que el registro se produce correctamente) y produce un resultado razonable.

- Definir el módulo SLALOM asociado.

  Cada módulo es un cálculo de lógica de negocios único y con una definición de parámetros que se pueden aplicar a todas las métricas relevantes. Al definir el módulo, es importante que éste se pruebe a fondo y se documente en detalle; qué hace el módulo (descripción del cálculo), qué parámetros espera (nombre, significado y uso), etc.

- Adjuntar todas las métricas al módulo de lógica de negocios relevante.

  Para cada métrica en los contratos definidos, se deberá definir un vínculo al módulo de lógica de negocios relevante. A continuación se deberá ejecutar en el ámbito de la lógica de negocios para garantizar que el vínculo se haya implementado correctamente y que el registro esté funcionando bien, es decir, esté recibiendo los eventos relevantes y produciendo los resultados esperados.

Módulos de lógica de negocios

Hay varias consideraciones importantes que tener en cuenta al construir módulos para la lógica de negocios, especialmente si está utilizando varios módulos dentro de una sola métrica:

- Para garantizar que queda claro el uso de un módulo, deberá agregar un comentario en la parte superior de la lógica de negocios para esa métrica.
- Si está utilizando un trozo de código pequeño dentro del espacio de la lógica de negocios de la métrica y ha incluido uno o varios módulos en el código, deberá asegurarse de que, en todos los controladores de eventos o subrutinas predeterminados, elimina esa sección de código de la lógica de negocios de la métrica principal. Se debe asegurar de que cada subrutina y cada controlador de eventos se haya definido solamente *una vez* en cada uno de los módulos que utilice una métrica particular. Esto es así para evitar confundir al motor de cálculo y que produzca resultados inesperados.

  Nota: Si, por ejemplo, define la función OnPeriodStart() en su módulo, pone código específico en ella y después deja la función OnPeriodStart() predeterminada sin código en la pantalla de la lógica de negocios principal de su métrica, entonces en el tiempo de ejecución el motor no sabrá qué subrutina ejecutar. Puede que no ejecute el código que quiere que ejecute.
- Hay que tener mucho cuidado si se está parametrizando la subrutina OnRegistration. En algunos casos, al hacerlo se puede romper la activación automática integrada en el sistema para recalcular métricas basadas en la agregación de nuevos datos sin procesar.

Dentro de la lógica de negocios

Toda la lógica de negocios se encuentra en un script basado en eventos, basado en la sintaxis de VBScript. Cada evento que alcanza la lógica de negocios, activa un controlador de eventos.

El motor que tiene que ser evaluado por la lógica de negocios envía los tipos siguientes de eventos:

- Eventos de datos sin procesar. Entradas de datos reales sin procesar en las que la lógica de negocios basa sus resultados. El motor envía solamente los eventos de datos sin procesar relevantes basados en el registro de la fórmula.
- Eventos de motor Eventos enviados por el motor implícitamente que reflejan las propiedades de la definición de la métrica, como definición de ranura de tiempo y período de seguimiento.
- Eventos intermedios Eventos generados por otros scripts de lógica de negocios y que se pueden utilizar dentro de otro evento.

La fórmula de lógica de negocios contenida dentro de la definición de métrica es lo que evalúa los eventos y envía un resultado de nivel de servicio en el que se basan los informes. Dependiendo de estos resultados de nivel de servicio y la definición de dominio, el motor también produce un resultado de desviación (si un destino del nivel de servicio se ha aplicado a la métrica). Los resultados producidos se almacenan en una tabla de base de datos llamada T_PSL. Es esta tabla lo que el Asistente de informes consulta cuando genera los informes; por consiguiente, todos los datos de informes se calculan previamente para asegurar que se maximiza el rendimiento de generación informes.

Flujo de eventos

Como se ha indicado previamente, las entradas en la lógica de negocios son los eventos de motor y los eventos de datos sin procesar.

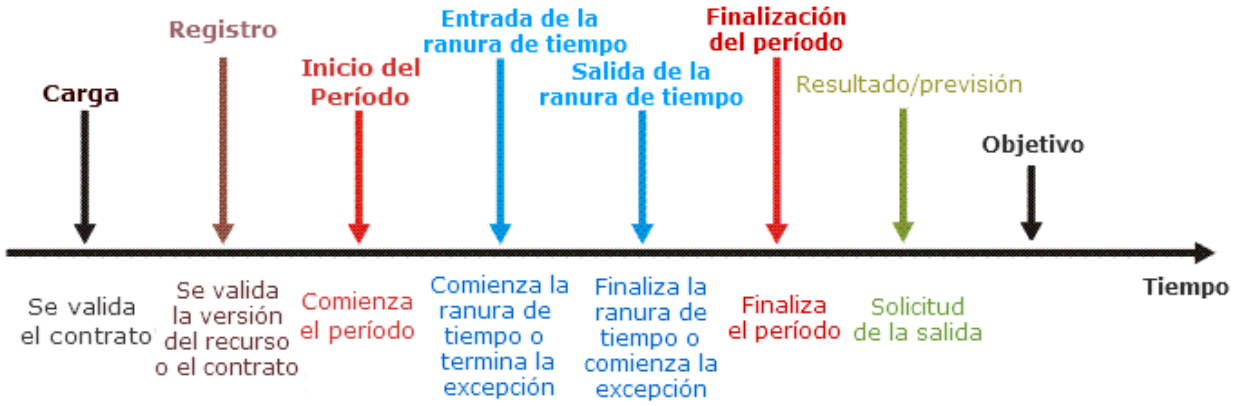
Los eventos de datos sin procesar por parte de la lógica de negocios vienen definidos por la función de registro dentro de la cual el código solicita un conjunto específico de eventos de datos sin procesar definidos por su tipo de evento y los identificadores de recurso.

En la lógica de negocios, el proceso de registro también asocia una subrutina definida por el usuario que se ejecuta para gestionar el evento de datos sin procesar una vez que se recibe un evento. (De forma predeterminada, se trata del evento OnXXXEvent, cuyo nombre se deberá cambiar por uno más significativo).

El motor activa los eventos de motor según las definiciones de contrato y las métricas asociadas. Una vez el evento de motor se activa y recibe, el motor ejecuta el controlador de eventos pertinente. Cada evento de motor tiene un controlador de eventos implícito. Estos controladores de eventos son funciones y procedimientos definidos sobre el VBScript. El controlador de eventos que gestiona el proceso de registro y la función "result" son ambos elementos de implementación obligatoria en el código. El resto de los controladores de eventos son opcionales. No obstante, la lógica de negocios no gestiona eventos de motor para los cuales no se han implementado controladores de eventos. Por lo tanto, es recomendable incluirlos todos (aunque no se usen) para permitir realizar futuras mejoras.

Nota: Al escribir el script de lógica de negocios es importante implementar todos los eventos de motor de manera que se cubran todas las posibilidades finales. Por ejemplo, si hubiera habido una definición de ranura de tiempo no aplicable durante la primera etapa de la implementación pero sí aplicable en el futuro, entonces todas las fórmulas requerirán modificación. Por lo tanto, se recomienda hacer esto para que el experto en la lógica de negocios defina por adelantado el comportamiento de períodos "en la ranura de tiempo" y "fuera de ranura de tiempo", aunque en un principio no sea aplicable; así, cuando se introduzca el comportamiento, los cambios del sistema necesarios serán de poca envergadura.

A continuación se incluyen varios eventos de motor y sus controladores de eventos:



- **OnLoad (Time)**: (opcional), se le invoca una vez al iniciar los cálculos cuando el contrato se está activando. Se puede utilizar para inicializar variables globales.
- **OnRegistration (Dispatcher)**: (necesario), procedimiento para solicitar los eventos de datos sin procesar relevantes y asociarlos con los procedimientos definidos por el usuario con los que gestionarlos. Se solicitan los eventos y se asocian con procedimientos mediante los métodos del objeto Dispatcher.
  El motor de cálculo invoca a OnRegistration una vez al iniciar el cálculo de la métrica y otra vez cada vez que entra en vigencia un recurso registrado para la métrica, tras lo cual el motor evalúa que el conjunto de cambios hechos para ese recurso que pueden influir en el conjunto de eventos transmitidos a la fórmula. El motor tiene la definición de la solicitud de evento definida por los identificadores de tipo de evento y de recursos y por si un recurso (o un conjunto de cambios con varios recursos) cambia algo relacionado con este conjunto. Una vez vigente, se activa un controlador de eventos OnRegistration.
- **OnPeriodStart (Time)**: (opcional), se le invoca al iniciarse el período de tiempo del agente (definido según la unidad de tiempo). El primer OnPeriodStart se activa una vez el contrato entra en vigencia, donde el balance de los períodos empieza con horas redondeadas como unidades de tiempo. Este controlador de eventos normalmente se utiliza para inicializar variables periódicamente, como un contador que guarda el número de incidentes abiertos dentro del período de cálculo que se deberán después inicializar en cero al principio de cada período.

- **OnPeriodEnd (Time,IsCompleteRecord)**: (opcional), se le invoca al final del período de tiempo del agente (definido según la unidad de tiempo). Se le invoca siempre al final redondeado de la unidad de tiempo en horas redondeadas (por ejemplo, 24:00). IsCompleteRecord es verdadero cuando el período de la métrica ha terminado (según el tiempo real del servidor de aplicaciones), y es falso al hacer un cálculo intermedio. Este controlador de eventos se utiliza normalmente para hacer los cálculos finales para el período final y preparar todo para el resultado final que proporcionará la función Result.
- **OnTimeslotEnter (Time)**: (opcional), se le invoca al entrar en un período TimeSlot basado en la definición de la métrica asociada. Por ejemplo, si la métrica está asociada a una definición de ranura de tiempo de horario comercial, donde cada día a las 08:00 AM empieza una ranura de tiempo, entonces este controlador de eventos se activará a esta hora cada día.
- **OnTimeSlotExit (Time)**: (opcional), se le invoca al salir de un período TimeSlot basado en la definición de la métrica asociada. Por ejemplo, si la métrica está asociada con una definición de ranura de tiempo de horario comercial, donde cada día termina a las 17:00 PM, entonces este controlador de eventos se activará a esta hora cada día.
- **Target()**: (opcional), se le invoca cuando se especifica la métrica con un destino dinámico. Le permite determinar el destino del nivel de servicio de una métrica durante el tiempo de ejecución de la lógica de negocios. Estos destinos pueden incluir el consumo de componentes de servicio o cambios de infraestructura. Tiene cuatro valores, como la función Result; uno para cada modo. En una ejecución normal, esta función se ejecuta *después de* la función Result.
- **Forecast()**: (opcional), se le invoca una vez al confirmar la versión del contrato haciendo que el motor de cálculo calcule el contrato una vez en modo de previsión. El modo Previsión está realizando un ciclo de motor de cálculo completo en el contrato (de la fecha de inicio a la fecha de finalización de la versión del contrato). La finalidad de este ciclo es calcular solo los valores de previsión. (No se realiza ningún cálculo en la tabla T_RAW_DATA). A esta función se le invoca *en lugar de* a la función Result en este modo de ejecución.
- **Result()**: (necesario), la invoca el motor de cálculo para obtener el resultado del cálculo para un período dado. A la función Result se le invoca siempre después del controlador de eventos OnPeriodEnd.

Los siguientes son los pasos que sigue el mecanismo de cálculo (servicio PslWriter) para procesar una sola fórmula de lógica de negocios:

- PslWriter carga la fórmula en la memoria y ejecuta todas las declaraciones que hay en la sección de declaraciones (la sección de declaraciones es todo el código que está fuera de una función o subrutina). Tenga también en cuenta que todos los módulos y las bibliotecas adjuntos se recopilan e incluyen en este único módulo de código para la ejecución.
- PslWriter invoca al controlador de eventos OnLoad.
- PslWriter invoca a OnRegistration.
- El cálculo del período empieza invocando a OnPeriodStart.
- Para cada evento de datos sin procesar registrado durante OnRegistration e incluido dentro del intervalo de tiempo del período, PslWriter invoca al controlador de eventos definido por el usuario asociado con ese evento.
- Si el inicio o el final de la ranura de tiempo cae dentro del intervalo de tiempo del período, se invocará a OnTimeslotEnter o a OnTimeSlotExit.
- Si se produce un cambio de infraestructura relevante dentro del intervalo de tiempo del período, se invocará a OnRegistration.
- El cálculo del período termina al invocar a OnPeriodEnd.
- Si se ha especificado un destino dinámico, entonces se invocará a esta función.
- PslWriter invoca a la función Result para obtener el resultado final del cálculo del período.

  Nota: Cuando se confirma por primera vez la versión del contrato y se selecciona una previsión, se invoca a la función Forecast en lugar de a la función Result. Esto, sin embargo, sólo se produce una vez para cada versión.

Durante cada ciclo de cálculo, el motor evalúa cuáles son los eventos de motor y los eventos de datos sin procesar relevantes basándose en el período de tiempo de cálculo. Primero los clasificará por tiempo y a continuación los enviará a las fórmulas relevantes para realizar los cálculos pertinentes. El tiempo del evento de datos sin procesar es su marca de tiempo y el tiempo del evento de motor es su momento de activación. Estos dos tipos de eventos se combinan después en una secuencia temporal y se envían para realizar el cálculo correspondiente.

Los tiempos de los eventos se basan en la métrica local asociada, pero el parámetro de tiempo de los controladores de eventos (OnPeriodStart (Time)) y la marca de tiempo del evento de datos sin procesar corresponde a su valor en UTC. El motor los compara con sus valores en UTC para utilizar un punto de referencia constante.

Ejemplo:

Si la diferencia de zona horaria de una métrica con respecto a la hora UTC es de dos horas (GMT+2), y hay un evento de apertura de incidente con marca de tiempo 10:00 AM, entonces la marca de tiempo que el controlador de eventos usará dentro del motor se cambiará en consecuencia y se iniciará en realidad a las 8:00 AM UTC. Suponiendo que el adaptador se ha configurado para usar esta misma zona horaria, entonces la hora de los eventos de datos sin procesar se retrasará 2 horas con respecto a la hora UTC también en la base de datos Cuando se pasan los eventos a la lógica de negocios, el agente de cálculo responsable del cálculo de eventos para el período que se inicia a las 10:00 AM en realidad usará el tiempo UTC para eventos, que será 8:00 AM en adelante. Sin embargo, si utiliza el mensaje out.log en el código para imprimir las marcas de hora, mostrará que la marca de tiempo se ha cambiado a la hora UTC y, por consiguiente, mostrará 08:00 AM a pesar de que el período especificado (según la métrica) era para 10:00 AM.

En el requisito de cálculo siguiente es importante convertir la marca de tiempo del evento antes de utilizarlo:

Si una métrica es calcular el número de incidentes cerrados en el mismo día de apertura, entonces es necesario comparar la hora de apertura y de cierre de cada incidente. Si la hora de apertura y de cierre de un incidente caen en el mismo día (y dentro de una ranura de tiempo definida), el incidente se contará.

Puede suceder que durante el proceso de cambiar el incidente a la hora UTC desde la hora local original, cambie el día (usando otra vez GMT+2). Un incidente abierto a la 1:00 AM se pasará a las 11:00 PM del día anterior en hora UTC. En este caso, no se contará cuando hubiera sido necesario. En esta situación, primero hay que convertir la hora de nuevo a la hora de la métrica local y después compararla. En tal caso, utilice el método GetLocaleTime(utcTime). Este método convierte una hora dada en una zona horaria UTC a la zona horaria de la métrica local.

A continuación se incluye el código del controlador 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

El registro es el proceso por el cual la lógica de negocios envía una solicitud al motor de cálculo para el conjunto de eventos de datos sin procesar que formarán parte del cálculo.

El proceso de registro se puede gestionar de dos maneras: mediante el asistente de registro, o manualmente utilizando el objeto de emisor dentro de la lógica de negocios.

Utilizar el asistente de registro es un sencillo proceso donde se van eligiendo las opciones disponibles. Las opciones disponibles son las mismas que al hacer el registro manualmente, sin posibilidad de utilizar parámetros. Si es necesario utilizar parámetros, tendrá que realizar el registro manualmente. El flujo básico del asistente exige determinar primero qué tipo de registro desea realizar y después establecer los tipos de recurso y los eventos sobre los cuales se deberá realizar el registro. Por último, deberá determinarse qué controlador de eventos se utilizará para procesar los eventos recopilados.

Una vez terminadas las operaciones de registro, aparecerán clasificadas en la ficha "Registro" de la métrica. Tenga también en cuenta que se puede tener más de una declaración de registro para una métrica.

En realidad, el asistente de registro utiliza la misma funcionalidad que el registro manual. Todas estas opciones se describen en la sección a continuación.

Cuando se realiza manualmente dentro de la lógica de negocios, el registro de la fórmula se gestiona a través del controlador de eventos *OnRegistration.* Esto se deberá implementar en la fórmula y se deberá activar cuando se active un evento de motor de registro. El evento de registro se activa una vez cuando se activa el contrato, y después cada vez que se active un recurso o un conjunto de cambios relevante. Para que un cambio en el recurso afectado se considere relevante deberá afectar a los eventos que la métrica debe recibir. Por ejemplo, si la parte contratante es quien realiza el registro *(RegisterByContractParty*), entonces formarán parte del cálculo todos los eventos del tipo definido cuyos recursos estén adjuntos a la parte contratante de la métrica. En este caso, cada vez que se adjunte un nuevo recurso a esa parte contratante, o se retire de ella, el método de registro se activará para notificar el cambio al motor.

El objeto Dispatcher proporciona los métodos de registro y se pasa a OnRegistration como argumento. Los diferentes métodos proporcionan distintas maneras de definir los criterios de filtrado basándose en la definición del tipo de evento y cualquier criterio de adjudicación de recursos, como recursos de un grupo de recursos o recursos de un cierto tipo.

Se recomienda encarecidamente utilizar los métodos de registro de servicio y parte contratante, ya que facilita el uso de la lógica de negocios como un módulo o plantilla. Al hacerlo así, se toma el servicio y la parte contratante relevante de la definición de la métrica asociada y, al reutilizar la fórmula para contratos y/o componentes de servicio diferentes, el registro no tiene que cambiar.

Otro popular método de registro es RegisterByResourceGroup, práctico para trabajar con recursos agrupados lógicamente pero que puede que no siempre estén asociados con servicios o partes contratantes. La asignación de recursos a los grupos aquí se puede gestionar a través del catálogo de recursos (individualmente o mediante conjuntos de cambios), y se podrá incluso actualizar automáticamente a través de scripts de traducción.

En general, el método de registro se determina durante la fase de diseño y se guía directamente a través del modelo de datos definido.

Nota: Para comprobar si el objeto Dispatcher se ha utilizado correctamente, se invoca también a la función OnRegistration al comprobar la sintaxis del SLALOM. Por este motivo, no se deberá asumir que OnLoad se ejecutó antes de la función OnRegistration, y no se deberán utilizar algunas de las propiedades del objeto de contexto, como "TimeUnit", "IntervalLength", "IsPeriod", "CorrectionsApply" y "ExceptionsApply" en el controlador de eventos OnRegistration.

Los métodos de registro son también responsables de asociar los eventos a un procedimiento que se activará según la marca de tiempo del evento. El procedimiento definido puede tener cualquier nombre, pero debe tener siempre el objeto eventDetails como parámetro. El objeto eventDetails refleja el evento de datos sin procesar recibido y guarda todos los datos del evento como campos de datos, tal y como se puede ver en el registro siguiente:

```
Sub OnRegistration(dispatcher)
dispatcher.RegisterByContractPartyAndService "OnMemUseEvent", "MemUse", "Server"
'Los parámetros del método son: <nombre del procedimiento>, <Tipo de evento>,
<Tipo de recurso>
End Sub
```

La declaración de registro anterior nos dice que se enviarán al controlador de eventos "OnMemUseEvent" en la lógica de negocios todos los eventos de datos sin procesar del tipo de evento "MemUse" y asociados al tipo de recurso "Server".

El procedimiento siguiente deberá haberse definido también antes en la lógica de negocios:

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

```
    End If
End Sub
```

La declaración extrae el valor del campo MemoryUsage del evento que se pasó en la función haciendo referencia al objeto eventDetails y mediante el parámetro "MemoryUsage". Estos campos son los mismos que los definidos en el tipo de evento, y los nombres de los campos distinguen entre mayúsculas y minúsculas.

Métricas de registro en clúster

Las métricas en clúster permiten ejecutar una definición de métrica para cada miembro individual de un grupo de recursos con objeto de aplicar la misma definición y la misma lógica para un conjunto de elementos. Las agrupaciones en clúster se pueden definir de manera estática en un conjunto predeterminado de recursos o de forma dinámica en los miembros del grupo de recursos, mientras que el grupo se puede cambiar con el tiempo, así como incluir o excluir miembros del grupo.

Nota: Para ver una descripción detallada, consulte el Apéndice E, Definición de fórmulas de lógica de negocios (experto en la lógica de negocios).

Las métricas en clúster se utilizan cuando es necesario calcular un resultado de nivel de servicio para cada elemento de un grupo de recursos. Los elementos dentro de un grupo de recursos pueden ser recursos u otros grupos de recursos, por lo que el método de registro en un script de lógica de negocios de una métrica en clúster debe ser RegisterByResourceGroup o RegisterByResource, donde el nombre del recurso o el nombre del grupo de recursos especificado se define como el elemento en el clúster. Esto se hace mediante la propiedad "ClusterItem" del objeto de contexto que guarda el nombre del elemento del clúster actual.

Ejemplo:

```
dispatcher.RegisterByResource  “<nombreDelProcedimiento>”, “<Nombre de Tipo de evento>”, Context.ClusterItem
```

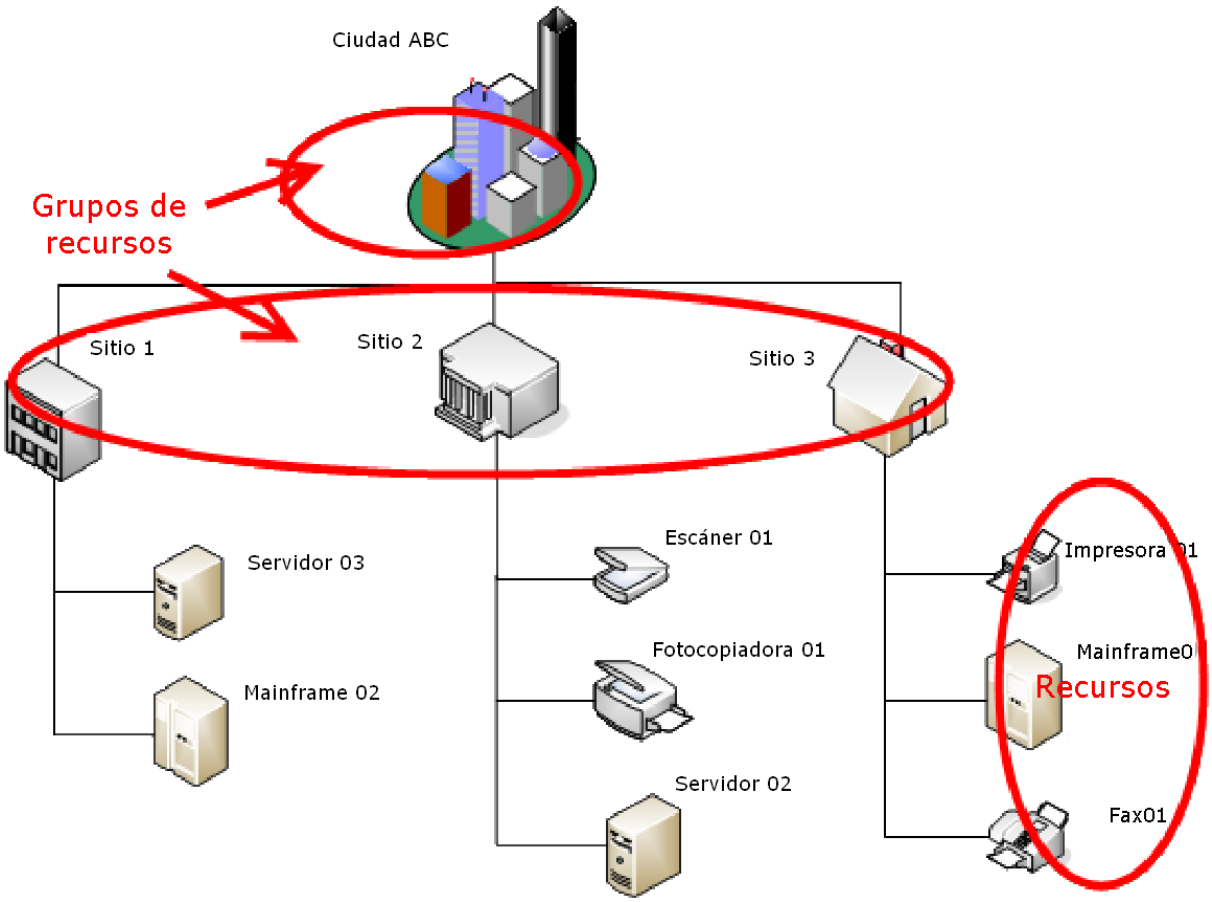
Cuando se use este método de registro, la métrica calcula un resultado para cada uno de los recursos en el grupo de recursos donde la métrica está agrupada;

o

```
dispatcher.RegisterByResourceGroup "<NombreDelProcedimiento>", "<Nombre de Tipo de evento>", Context.ClusterItem
```

Cuando se use este método de registro, la métrica calculará un resultado para cada uno de los grupos de recursos contenidos dentro del grupo de recursos donde se ha agrupado la métrica.

La agrupación en clúster puede producirse en niveles diferentes, dependiendo de cómo se cree el modelo del recurso. A menudo las organizaciones quieren representar capas de agrupación diferentes. Por ejemplo, una ciudad concreta puede incluir varios sitios, y dentro de cada sitio podría haber varios dispositivos de infraestructura (impresoras, escáneres, servidores, etc.). Mediante este tipo de agrupación podría estructurar una jerarquía de recursos definida que contuviera varios niveles y agrupaciones con estos elementos de hardware. Suponiendo que tenemos un dispositivo de infraestructura que será el recurso. La estructura descrita aquí podría tener el siguiente aspecto:


Ciudad ABC
Grupos de recursos
Sitio 1
Sitio 2
Sitio 3
Servidor 03
Mainframe 02
Escáner 01
Fotocopiadora 01
Servidor 02
Impresora 01
Mainframe0
Recursos
Fax01

Como se puede ver en el diagrama, hay varias capas de grupos. El grupo de nivel superior "City ABC" contiene tres sitios diferentes (que son también grupos de recursos). El grupo de recursos "Site 3 Resources" contiene tres recursos diferentes. Así, en el ejemplo anterior, para agrupar la métrica en los tres sitios, utilizaría el registro siguiente:

```
dispatcher.RegisterByResourceGroup  "<NombreDelProcedimiento>", "<Nombre de Tipo de evento>", Context.ClusterItem
```

En este caso, Context.ClusterItem hace referencia al grupo de recursos llamado "City ABC Sites", que contiene tres grupos de recursos más llamados "Site 01 Resources", "Site 2 Resources", etc., y que podría tener el siguiente aspecto en la ficha de agrupación en clúster de la métrica.

Observe también que la agrupación en clúster se ha establecido en dinámica, ya que esto incluirá automáticamente cualquier cambio en el grupo a medida que se fueran produciendo. La agrupación en clúster estática puede ser útil para subconjuntos de grupos de recursos o cuando no se desea que la agrupación en clúster cambie con el tiempo.

Para crear una métrica que informe sobre los recursos del grupo Site 3, utilice la declaración de registro siguiente:

```
dispatcher.RegisterByResource  “<nombreDelProcedimiento>”, “<Nombre de Tipo de evento>”, Context.ClusterItem
```

En este caso, Context.ClusterItem hace referencia a los recursos individuales, por consiguiente se registrará por recurso solamente. La ficha de agrupación en clúster de la métrica contiene una referencia al grupo "Sites 03 Resources".

Puede configurar la agrupación en clúster para que funcione en niveles diferentes de la jerarquía dentro de una sola métrica. Por ejemplo, supongamos que tenemos la situación descrita en el ejemplo anterior y se agrupa esta métrica en clúster de nuevo en el grupo "City ABC Sites". Puede incluir dentro de una métrica los miembros del recurso de niveles diferentes de la jerarquía. En este caso, hay tres opciones en cuanto a qué recursos se incluyen en esta agrupación:

- **Primer nivel solamente: miembros directos**: igual que los métodos de clúster antiguos descritos previamente.
- **Todos los niveles: Incluye recursos solamente**: incluye todos los recursos dentro de los tres grupos de recursos de sitios en un solo nivel y calcula el nivel de servicio para ellos por individual.
- **Todos los niveles: Incluye recursos y grupos de recursos**: incluye todos los recursos contenidos en los tres grupos de recursos de sitios, así como los tres grupos de recursos, y calcula el nivel de servicio para ellos por individual.

Agentes

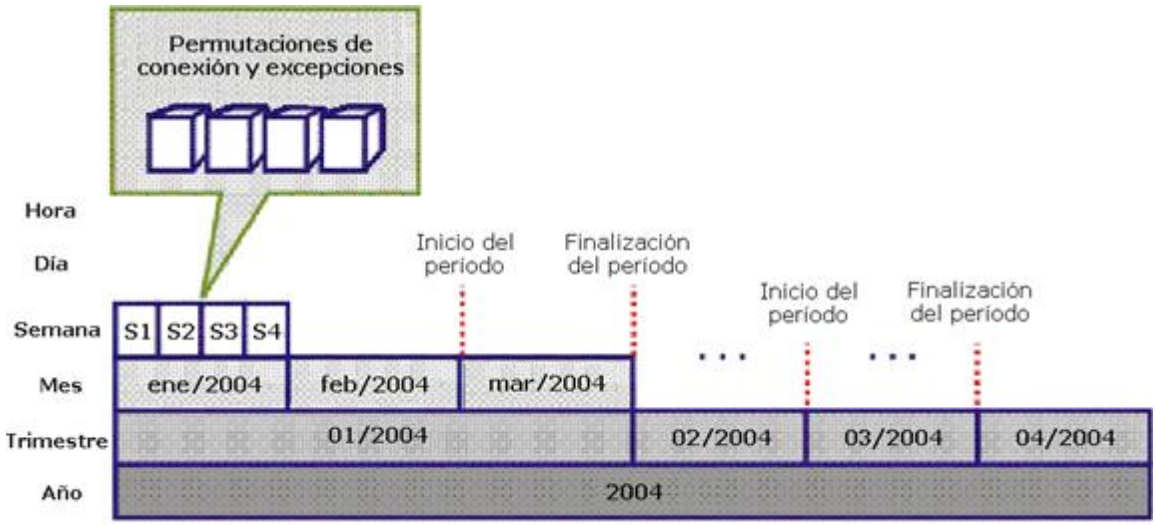
Cada métrica tiene definido un período de seguimiento. El período de seguimiento es el período durante el cual la métrica tiene que emitir un resultado del nivel de servicio; es este resultado el que se deberá comparar con el destino definido. El resultado producido para el período de seguimiento es el resultado de negocio. Dicho de otra manera, es el resultado contractual con el que se ha comprometido el proveedor para proporcionar un nivel de servicio de destino concreto. Cada una de las instancias de la lógica de negocios para cada período de tiempo se conoce como agente, y está directamente relacionado con las granularidades asociadas a cada métrica.

Por ejemplo, si la métrica se define con un período de seguimiento de un mes, entonces la métrica se ejecuta para proporcionar un resultado mensual.

Para poder obtener detalles en el resultado mensual, allí donde el usuario puede consultar el resultado diario, la métrica debe tener definidas unidades de tiempo adicionales en las cuales se deberá calcular el resultado. Las unidades de tiempo se definen en la sección Detalles generales de la métrica en la ficha Granularidad.

Para cada una de las unidades de tiempo definidas en la ficha Granularidad de la métrica, y para el período de seguimiento, el motor ejecuta una instancia de lógica de negocios aparte. Cada una de estas instancias ejecuta el mismo código de lógica de negocios, pero OnPeriodStart y OnPeriodEnd se activarán de forma diferente para cada una de estas instancias. Para la instancia diaria, se activará al inicio del día y al final del día. Para cada unidad de tiempo, se activará según los puntos de inicio y fin de la unidad de tiempo.

Se ejecuta cada una de las instancias de la lógica de negocios para producir el resultado de unidad de tiempo relevante. Además, cada período tiene que tener un resultado que ha de tener en cuenta todas las excepciones. Los períodos que no lo hagan (si se han definido excepciones), deben permitir al usuario elegir ver el resultado de nivel de servicio con o sin las excepciones que se han tenido en cuenta. Dado este requisito, el motor ejecutará potencialmente catorce agentes (instancias) por cada métrica, dos agentes para los resultados de negocio y doce para los seis períodos operacionales adicionales.



Escala de tiempo de la lógica de negocios

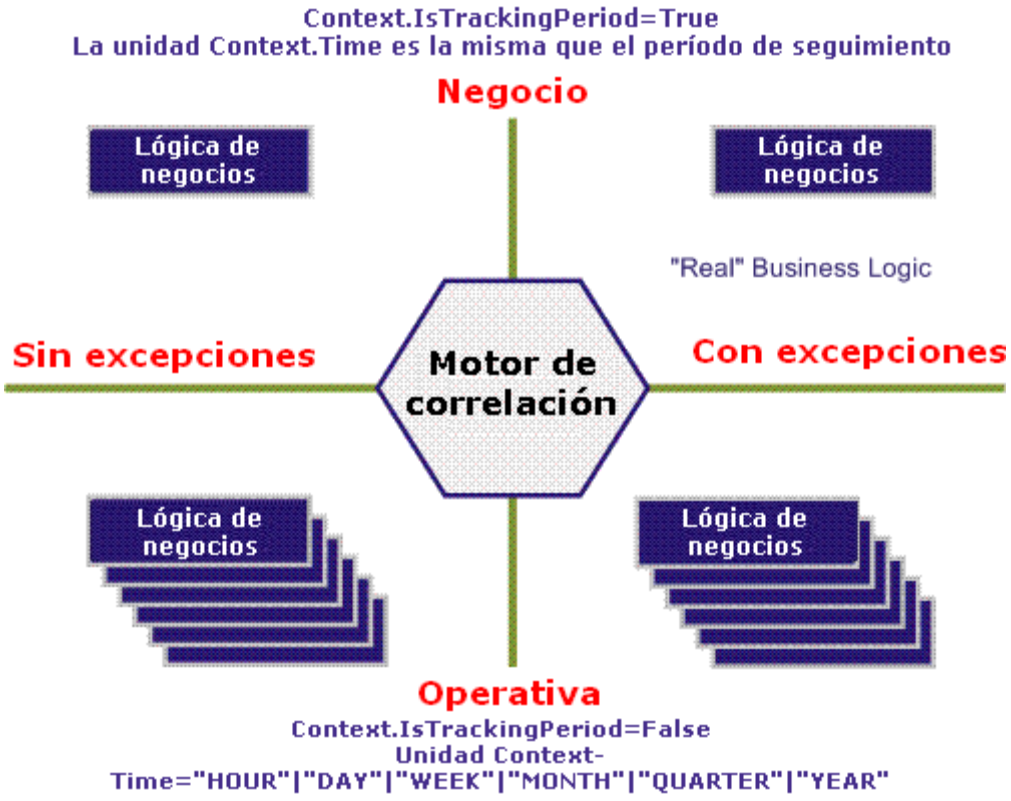
Esto significa que la definición de granularidad tiene un efecto importante en el rendimiento del motor de cálculo, ya que cada período de tiempo se calcula para un agente diferente. Así, cuando no se requiere capacidad de obtención de detalles total o parcial, se recomienda desactivar algunos de los agentes. Esto afecta especialmente a las granularidades más bajas, como los resultados por horas. También afecta mucho a las métricas en clúster, ya que el motor hace todos los cálculos anteriores para cada elemento ClusterItem que encuentra. En realidad, trata a cada elemento ClusterItem como una nueva métrica. Por ejemplo, al calcular una métrica en un grupo de recursos de 50 elementos, el motor tendrá que hacer 49 veces más trabajo si se compara con la misma métrica no en clúster.

Por ejemplo, si la métrica se establece para calcular el tiempo de resolución en número de días, entonces EL resultado por horas no tendrá sentido y deberá quitarse la marca correspondiente en la ficha de granularidades para evitar que el motor que realice cálculos innecesarios.

El atributo TimeUnit del objeto de contexto (context.TimeUnit en la lógica de negocios) devuelve la unidad de tiempo del agente que se está ejecutando actualmente, donde los valores de devolución posibles son: HOUR, DAY, WEEK, MONTH, QUARTER y YEAR.

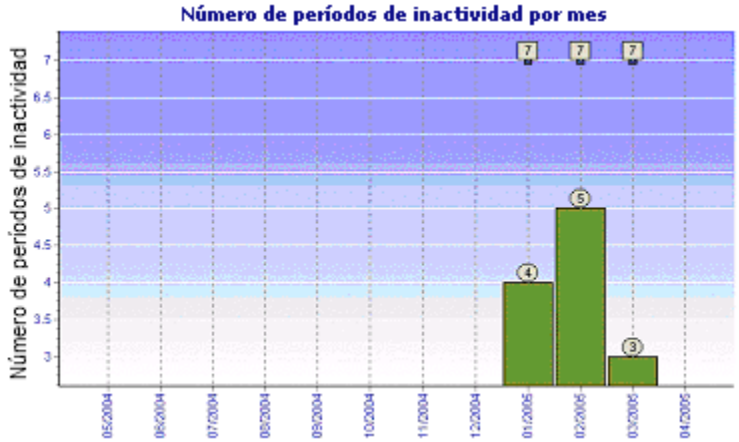
Por ejemplo, para el agente diario, Context.TimeUnit devolverá "DAY".

El atributo IsTrackingPeriod del objeto de contexto devolverá verdadero para el agente que calcula la unidad de tiempo del período de seguimiento. También es importante saber que los agentes mostrados en la ficha Granularidad de las métricas son adicionales al agente del período de seguimiento de la métrica. Así, incluso para una métrica con un período de seguimiento mensual, podrá desactivar el agente mensual y seguirá calculando el nivel de servicio mensual, pero solamente como "resultados de negocio" (resultados no operacionales).

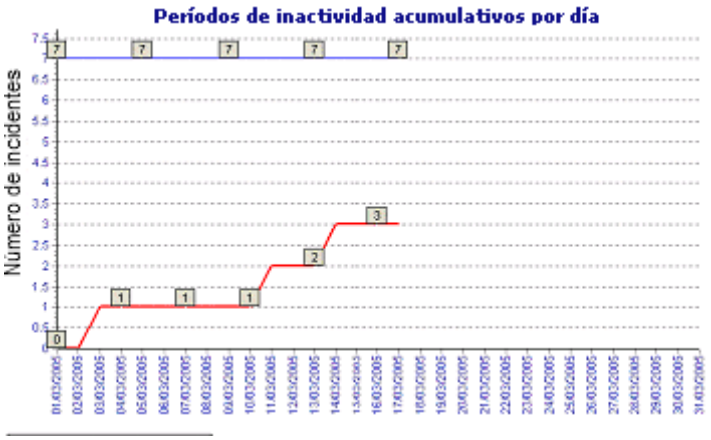


Como hemos mencionado anteriormente, los distintos agentes normalmente ejecutan el mismo código de lógica de negocios, pero hay casos en los que es necesario aplicar una lógica un poco diferente.

Por ejemplo, en el caso mensual, el resultado deberá ser el número de tiempos de inactividad para cada mes por separado, como se muestra a continuación:



Para la obtención de detalles diaria, puede ser necesario poder consultar los tiempos de inactividad acumulados, donde cada día es una suma de todos los días desde el inicio del mes. Este método se deberá aplicar a todas las unidades de tiempo más pequeñas y, a continuación, a un solo mes, tal y como se muestra a continuación:



La diferencia entre las dos unidades de tiempo es que, para el agente que calcula el período de seguimiento, el contador de tiempos de inactividad se inicializará en 0 al inicio de todos los períodos, pero para el agente diario, el contador se inicializará solamente cuando el día sea el primer día del mes.

A continuación se incluye el controlador 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
```

¿Qué es el recálculo?

El recálculo se realiza cuando el motor de correlación identifica que el resultado periódico final de una métrica no es ya válido y se deben recalcular por lo tanto los resultados.

Los motivos para hacer un recálculo pueden ser:

- Que se estén recibiendo nuevos eventos que realmente se produjeron en el pasado (antes de lo que el motor haya calculado *hasta el momento* para esa métrica).
- Que se haya modificado un recurso registrado para la métrica (nueva fecha de versión y cambios realizados en él).
- Que se haya confirmando una nueva versión del contrato que se solapa con el tiempo previamente calculado con cambios en algunas métricas (solamente se recalculan las métricas modificadas).

El método más eficaz para la tarea de registro es por Servicio y parte contratante. Organizar los recursos bajo Servicio y parte contratante es una forma de expresar la relación lógica entre la capa de datos y la capa de negocio en el sistema. Al registrar recursos a través de estas entidades no será necesario hacer cambios en las fórmulas cuando se usen en contratos diferentes o para servicios diferentes. El contexto actual de la métrica identifica el contrato y servicio, que define la parte contratante relevante y el servicio asociado. Las fórmulas de lógica de negocios definidas en este tipo de registro son fácilmente reutilizables, ya que no requieren cambios en el registro.

Salidas: tablas de usuario

El script de lógica de negocios estándar no tiene acceso a tablas de salida externas. Hay solamente dos destinos de salida:

- La tabla de PSL (T_PSL), donde el motor escribe automáticamente los resultados de nivel de servicio y donde se escribe el resultado especificado en la función Result. Los valores de nivel de servicio que se escriben en esta tabla sólo pueden ser numéricos. Los resultados escritos en la tabla T_PSL son los resultados sobre los cuales informa el asistente de informes. No se puede controlar la estructura o el método de escritura de estos resultados. Esto lo hace automáticamente el motor de correlación.
- La tabla de salida SLALOM (T_SLALOM_OUTPUTS). Las operaciones de escritura en esta tabla se realizan mediante los métodos proporcionados por el objeto de herramientas desde dentro de la fórmula de la lógica de negocios. Al enviar datos a esta tabla, se proporciona un nombre de tabla lógica a la que se hace referencia como la tabla de usuario. Estas son las tablas que se utilizan para extraer información durante el cálculo del nivel de servicio. Esta información se puede utilizar más tarde para generar informes de formato libre. Puede haber muchas tablas de usuario.

Es necesario usar la tabla externa T_SLALOM_OUTPUTS cuando se necesitan salidas adicionales para el resultado de nivel de servicio periódico y por encima de él, cuando no se pueden proporcionar salidas adicionales agregando otra métrica, o cuando al añadir otra métrica se reduce el rendimiento del cálculo al pasar por el mismo conjunto de registros solamente para proporcionar una salida diferente.

Supongamos por ejemplo un caso donde se ha definido una métrica para calcular el porcentaje de tickets que se resolvieron en menos de un día y que debe producir un informe con la lista de todos los tickets de resolución fallida en menos de un día; es necesario que la fórmula proporcione cada ticket con error en una tabla externa, además de añadirlo a las estadísticas de cálculo.

Con el requisito anterior, la tabla de nivel de servicio de salida normal no puede proporcionar esta salida porque:

- Los resultados de servicio son todos numéricos.
- Sólo se puede obtener un resultado de nivel de servicio para cada período de tiempo.

Los registros se escriben en las tablas de salida de usuario solamente para el agente que se está ejecutando para el período de seguimiento de la métrica y que calcula las excepciones y las correcciones. Por ejemplo, si la métrica se define con un período de seguimiento mensual, entonces las declaraciones de salida (Tools.SaveRecord y Tools.SaveFields) NO se ejecutan cuando el motor está ejecutando la fórmula para las otras granularidades como HOUR, DAY, WEEK, QUARTER y YEAR.

Una ventaja adicional de contar con esta tabla externamente es que se puede utilizar para varios requisitos de elaboración de informes. Además de los requisitos contractuales, se pueden generar otros requisitos de elaboración de informes típicos a partir de estas tablas. Por ejemplo, una métrica que calcule el número de tickets cerrados en un mes podría calcular también el tiempo de resolución de los tickets y enviar toda esta información a la tabla de salida. Esto permitirá realizar un análisis más detallado de los tickets individuales que se cerraron durante el período, junto con otros detalles adicionales que se puedan requerir a través de un requisito de informe aparte.

El mecanismo de recálculo del motor gestiona también la información en estas tablas. Durante este proceso de recálculo, los registros se marcan como inactivos y se escribe en su lugar una nueva fila. Esto es un punto importante que tener en cuenta, ya que todas las declaraciones SQL de informes de formato libre tienen que incluir una marca en el campo IS_ACTIVE en la tabla T_SLALOM_OUTPUTS (donde is_active = 1), ya que solamente son actuales estos registros.

Nota: Al ejecutar el ámbito de la lógica de negocios durante el proceso de depuración de fórmulas, los mensajes se escriben realmente en la tabla T_DEBUG_SLALOM_OUTPUTS, en lugar de en la tabla T_SLALOM_OUTPUTS.

Al documentar datos mediante T_SLALOM_OUTPUTS, los datos insertados son siempre texto (ya que los campos de T_SLALOM_OUTPUTS son todos varchar2). Por lo tanto, los valores de fecha se convierten a texto aplicando el formato del sistema operativo, el cual puede cambiar durante el ciclo de vida de la aplicación. T_SLALOM_OUTPUTS puede sufrir por lo tanto inconsistencias en los formatos de fecha. Además, la lógica de negocios gestiona fechas UTC, mientras que se podría esperar que T_SLALOM_OUTPUTS guardara las marcas de hora locales. Así, en algunos casos puede ser necesario utilizar la función de conversión *Tools.GetLocaleDate(date)* para solucionar este problema.

```
La función siguiente convierte las fechas en horas locales y mantiene la
coherencia del formato de fecha aplicando a las fechas el formato "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
```

Se pueden utilizar dos métodos para escribir en la tabla externa desde dentro de la fórmula de la lógica de negocios:

- Tools.SaveRecord <lista de parámetros>
- Tools.SaveFields <lista de parámetros>

Estos dos métodos del objeto de herramientas se describen a continuación en más detalle:

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

Este método guarda un registro en una tabla llamada T_SLALOM_OUTPUTS. El parámetro tableName especifica la tabla (virtual) dentro de T_SLALOM_OUTPUTS donde escribir la información. Cada registro en la tabla de usuario tiene una clave única que especifica el registro donde se debe escribir la información. Cada registro tiene también hasta 20 campos de valores de tipo cadena. El método SaveRecord recibe un nombre de tabla de usuario y una clave. También acepta todos los campos de valores en la tabla de usuario. (Estos parámetros de valores son opcionales y se pueden omitir.) Si ya existe un registro con la misma clave, se actualiza. (Solamente se actualizan los campos de valores transferidos como parámetros). Si no existe un registro con esta clave, se crea.

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

Este método es similar a SaveRecord, excepto en que en lugar de enumerar todos los valores, proporciona pares de nombres de campos y valores de campos relacionados. Los números de campo pueden sustituirse por nombres de campo. Los nombres de los campos deberán definirse previamente en la tabla T_SO_FIELD_NAMES. Esta tabla se utiliza para registrar la estructura de las tablas de salida.

Se recomienda que el experto en la lógica de negocios defina la estructura de la tabla de salida antes de escribir en T_SLALOM_OUTPUTS, ya que entonces la estructura y el significado de los campos estará ya bien definida y será mucho más sencillo consultarlos.

La estructura de la tabla es:

- **table_name**: cada tabla lógica tiene un nombre único.
- **field_name**: cada campo en una tabla es único.
- **field_id**: cada campo se numera con un número en serie desde 1.

Es preferible utilizar el método SaveFields, ya que guarda una documentación de la estructura y el significado de los valores insertados.

Ejemplo:

Supongamos un caso donde es necesario producir una lista de todos los incidentes con tiempo de resolución superior al umbral definido, además del resultado de la métrica que calcula el porcentaje de tickets resueltos por debajo de ese umbral. La escritura en las tablas de salida se hará en el procedimiento del controlador de eventos OnXXXEvent tras validar el umbral.

El ejemplo siguiente ilustra este caso:

```
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
```

Unas cuantas sugerencias relacionadas con el uso de las tablas T_SLALOM_OUTPUTS:

- Se recomienda escribir solamente los datos necesarios en las tablas de salida (no más de los necesarios).
- Escriba en las tablas de salida sólo para las granularidades de métrica que sean necesarias.
- Todos los campos de valor tienen solamente 50 caracteres de tamaño de forma predeterminada (excepto la columna VAL_1, que tiene 512). Para garantizar que los valores de los campos se ajusten a este tamaño de datos, es posible que haya que truncar algunos, ya que de lo contrario aparecerán errores de tiempo de ejecución en la lógica de negocios.
- Solamente hay 20 columnas disponibles para sus datos (VAL_1... VAL_20).

Nota: El hecho de escribir en las tablas de salida puede afectar al rendimiento del motor, ya que ésta es una operación computacionalmente intensiva en comparación con realizar un cálculo en la memoria. Por lo tanto, determine cuidadosamente si es necesario utilizar esta funcionalidad y donde será necesario y, a continuación, minimice el número de veces que se accede a las tablas.

Elaboración de informes con información de las tablas definidas por el usuario

El asistente de informes de CA Business Service Insight estándar no se puede usar para elaborar informes sobre lo escrito en las tablas de salida. Para elaborar informes con esta información es necesario crear un informe de formato libre. Esto significa fundamentalmente crear una consulta SQL sobre esta tabla. Dado que esta tabla contiene muchas tablas lógicas que se están escribiendo en diversas fórmulas, se recomienda que alguien familiarizado con SQL (experto en el origen de datos) cree una vista sobre la tabla T_SLALOM_OUTPUTS para que sea más fácil distinguir entre las diferentes tablas lógicas contenidas dentro de ella, y también para garantizar que la información se recupere según lo programado.

La definición de vista tendrá ya todos los tipos de campos de cadena para el tipo de información relevante y tendrá también todos los filtros necesarios (tabla lógica, registros activos, métrica relevante, etc.).

A continuación se incluye un ejemplo de creación de la vista:

```
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'
```

Creación de módulos de lógica de negocios

Los usuarios pueden definir módulos de lógica de negocios independientes que pueden utilizar varios objetivos de nivel de servicio (métricas). Para aplicar cambios de ámbito global en el sistema a la lógica de negocios, el usuario cambia el componente de lógica "básico" para después aplicarlo a todos los SLA relevantes con un solo clic.

Un módulo de lógica de negocios es un componente de código en el que se puede definir la lógica de negocios, además de permitir hacer un mantenimiento sencillo y reducir su redundancia. Varias métricas pueden usar un mismo módulo.

Durante la etapa de configuración, las fórmulas se configuran para definir los módulos de lógica de negocios principales. (Consulte el capítulo Diseño, Módulos y plantillas de lógica de negocios).

Hay tres tipos de módulos de lógica de negocios:

- **Lógica de negocios completa**: se utiliza cuando hace falta usar como módulo una fórmula completa. Se deberán implementar los métodos Result y OnRegistration en el script del módulo.
- **Clase**: módulo que contiene una definición de una sola clase de VB.
- **Biblioteca**: módulo que contiene cualquier recolección de procedimientos, funciones y clases.

Se pueden utilizar módulos desde dentro de cualquiera de lo siguiente:

- Métrica
- Otros módulos
- Plantillas de lógica de negocios
- Métricas dentro de plantillas de contrato y plantillas de nivel de servicio

Los módulos pueden utilizar parámetros gestionados desde el contexto de métrica Parameters("NombreDelParámetro").

Nota: Para evitar errores de tiempo de ejecución, defina siempre un valor predeterminado al utilizar parámetros en los módulos de lógica de negocios. Cuando no hay parámetros, la fórmula envía un mensaje de error de registro.

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

```
End If
```

Ejemplo de módulo de lógica de negocios

Hay un objeto de nivel de servicio descrito como " actividad de Help Desk dentro del umbral". El siguiente sistema de Help Desk tiene un ciclo de vida de tickets con los siguientes estados:

- Abierto
- Asignado
- Cerrado.

Dos de las métricas que se pueden definir para descubrir el rendimiento del Help Desk son las siguientes:

- **Nombre de la métrica**: **Éxito al ofrecer resolución de ticket a tiempo**.

  Declaración del objetivo: como mínimo, el 99% de los tickets debe resolverse en 4 horas.

  Lógica de negocios: la resolución se debe calcular de abierto a cerrado.
- **Nombre de la métrica**: **Éxito al asignar ticket a tiempo**.

  Declaración del objetivo: como mínimo, el 99% de los tickets debe asignarse en 30 minutos.

  Lógica de negocios: la asignación se debe calcular de Abierto a Asignado.

Puesto que se puede identificar la misma lógica para estas dos métricas, se puede crear un módulo para atender a las dos métricas.

El módulo requiere los siguientes parámetros dentro del contexto de la métrica:

- Primer estado
- Segundo estado
- Umbral

Una vez se cree el módulo de lógica de negocios, las métricas definidas pueden utilizar un módulo como parte de la definición.

A continuación, se puede cambiar la lógica. Por ejemplo, se deberá tener en cuenta un nuevo estado "Cliente a la espera". Un ticket se establece en Cliente a la espera cuando el Help Desk espera información adicional acerca del ticket por parte del cliente. Dentro de la lógica de negocios, mientras el ticket esté en estado Cliente a la espera, no se deberá considerar parte del cálculo. Por lo tanto, para tener en cuenta el estado y la lógica nuevos, solamente deberá cambiarse el módulo de lógica de negocios. Se crea una nueva versión del módulo con la nueva lógica.

Al confirmar el cambio, se le mostrará una lista de métricas que emplea el módulo. Puede aplicar el cambio a todas las métricas colectivamente, o bien puede optar por aplicar el cambio solamente a métricas específicas dentro de la lista.

Si selecciona solamente métricas específicas de la lista, se le pedirá que cree un nuevo módulo para las métricas seleccionadas. El módulo antiguo utilizado por las métricas seleccionadas se reemplazará por el nuevo módulo de lógica de negocios y se hará el recálculo con la nueva lógica.

Creación de parámetros

Los parámetros proporcionan a los usuarios de negocio una forma rápida e intuitiva de consultar y cambiar sus valores mediante la interfaz de usuario gráfica (GUI), sin tener que editar el script de la fórmula.

El uso de parámetros dentro de la lógica de negocios permite crear fórmulas generales que pueden tener un amplio uso en todo el sistema y optimiza el uso de los módulos.

Los parámetros se pueden definir en el nivel de contrato o de métrica. Los parámetros de nivel de métrica se muestran y se configuran en la ficha Declaración del objetivo de los detalles de la métrica. La lógica de negocios sólo tiene acceso a los parámetros de nivel de métrica; por lo tanto, para acceder a un parámetro de contrato desde dentro de una métrica, se crea un tipo de parámetro diferente localmente dentro de la métrica. Éste se denomina "parámetro dinámico" y toma su valor como referencia de los parámetros de nivel de contrato. Los valores de referencia permitidos en el parámetro dinámico son solamente los definidos en el contrato principal de la métrica.

Tipos de parámetros:

- Fecha: en modo de fecha (fecha + hora).
- Número: tamaño máximo de 15 dígitos con una precisión máxima de 5 dígitos.
- Texto: tamaño máximo de 100 caracteres.
- Tabla: tamaño máximo de 120 entradas.

Para acceder a los valores de los parámetros desde el código de fórmula, es necesario de utilizar el objeto Parameters y hacer referencia al nombre del parámetro.

Ejemplo:

```
Parameters("Threshold")
(Tenga en cuenta que este es un método abreviado de invocar al valor. Normalmente esto se hace como Parameters.Item("Threshold"))
```

O, para un parámetro de tipo tabla:

Parameters("Table")("Word")("Price")

(donde los valores de "Word" y "Price" corresponden a los nombres de fila y columna del parámetro en la tabla, respectivamente).

Los parámetros de tabla se deberán utilizar solamente confirmados una serie de puntos clave:

1. Se ha definido una variable global (p. ej. dim miParamDeTabla)
2. En la función OnLoad, se ha rellenado la variable del objeto Parameters (p. ej. "Set miParamDeTabla = Parameters("MiTablaDeParametros"))
3. Así pues, utilice solamente el objeto creado, miParamDeTabla. El parámetro en sí no se deberá utilizar fuera de la función OnLoad, y sólo deberá hacer referencia al objeto de la variable global creada a partir de él.
   (p. ej. "dim miVal: miVal = miParamDeTabla ("miFila")("miColumna")).

Esto libera memoria que el parámetro toma y evita que el motor cree una asignación de parámetro en cada invocación de parámetro para cada "OnXXXEvent", que se puede invocar miles de veces por métrica.

El código siguiente ilustra el uso correcto de un parámetro de tabla:

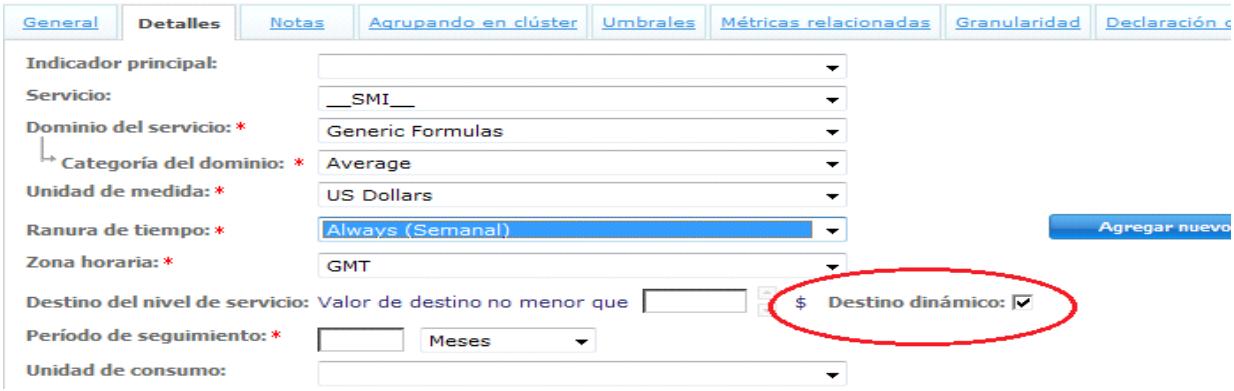
```
Opción explícita

Dim sum
Dim miTablaDeParametros

Sub OnLoad(TIME)
   Set miTablaDeParametros = Parameters("MiParamDeTabla")
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

Resultado de la función
   Result = ( sum * miTablaDeParametros("secondRow")("thirdColumn")  )
End Function
```

Los siguientes son los métodos disponibles para cada objeto de parámetro creado dentro del código.

Parámetro	Descripción
IsExist	Si existe el parámetro.
IsNumeric	Si el tipo de parámetro es numérico.
IsText	Si el tipo de parámetro es texto.
IsDate	Si el tipo de parámetro es fecha.
IsTable	Si el tipo de parámetro es tabla.
IsEntryExist	Si existe una entrada en la tabla.
IsEntryNumeric	Si el tipo de entrada en la tabla es numérico.
IsEntryText	Si el tipo de entrada en la tabla es texto.
IsEntryDate	Si el tipo de entrada en la tabla es fecha.
Dump	Devuelve una lista con todos los parámetros.
Item	Hace referencia al valor del parámetro.

Implementación de destinos dinámicos

Los destinos dinámicos se gestionan a través de la lógica de negocios mediante un controlador de eventos en el script de lógica de negocios estándar, similar a la función Result() que se usa para devolver el valor de nivel de servicio dese la métrica. El destino dinámico se deberá especificar en la ficha de detalles de las métricas, como se muestra a continuación.



Cuando se especifica un destino dinámico, el destino se toma de la función "Target()" en la lógica de negocios, en lugar de el valor estático especificado en la ficha Detalles de la métrica. La función Target tiene el siguiente aspecto.

```
Function Target
  'TAREA: AGREGAR aquí código PARA gestionar el cálculo de destinos dinámicos
  Target = Null
End Function
```

Esta función se deberá implementar basándose en el requisito para la métrica para devolver el valor de destino deseado para un período específico. La función puede devolver cualquier valor que la lógica de negocios le pueda asignar.

Ejemplo real de destinos dinámicos

Para un centro de llamadas, el destino para una métrica que mide el "*Tiempo promedio de respuesta a llamadas*" puede depender del volumen de llamadas. Si hay entre 0 y 800 llamadas, el destino deberá ser < 15 segundos; si hay entre 801 y 1500 llamadas, el destino deberá ser < 20 segundos; si hay más de 1500 llamadas, el destino deberá ser menos de 25 segundos. Esto se podría implementar como sigue (asumiendo que TotalCalls es un contador incrementado para cada evento de llamada recibido y que TotalCalls no puede ser menos de 0):

```
Function Target
  If TotalCalls >0 and TotalCalls <= 800 Then
     Target = 15
  ElseIf Total Calls > 800 and TotalCalls <= 1500 Then
     Target = 20
  Else
      Target = 25
  End If
End Function
```

Otro ejemplo de cómo usar un destino dinámico

Supongamos una situación en la que el destino de una métrica puede cambiar dependiendo de la granularidad del cálculo. Puede darse el caso de que haya un destino diario de disponibilidad de un 98 % para un grupo de servidores, pero que el destino mensual corresponda a una disponibilidad de un 99,5 %. La solución para esto requiere utilizar la función de destino dinámico conjuntamente con la invocación de la función a *Context.TimeUnit* para determinar el agente actual que se está calculando. Por consiguiente, se puede ajustar el destino en consecuencia.

```
Function Target
  If Context.TimeUnit = “DAY” Then
     Target = 98
  ElseIf
     Target = 99.5
```

```
  End If
End Function
```

Estados de seguridad estados

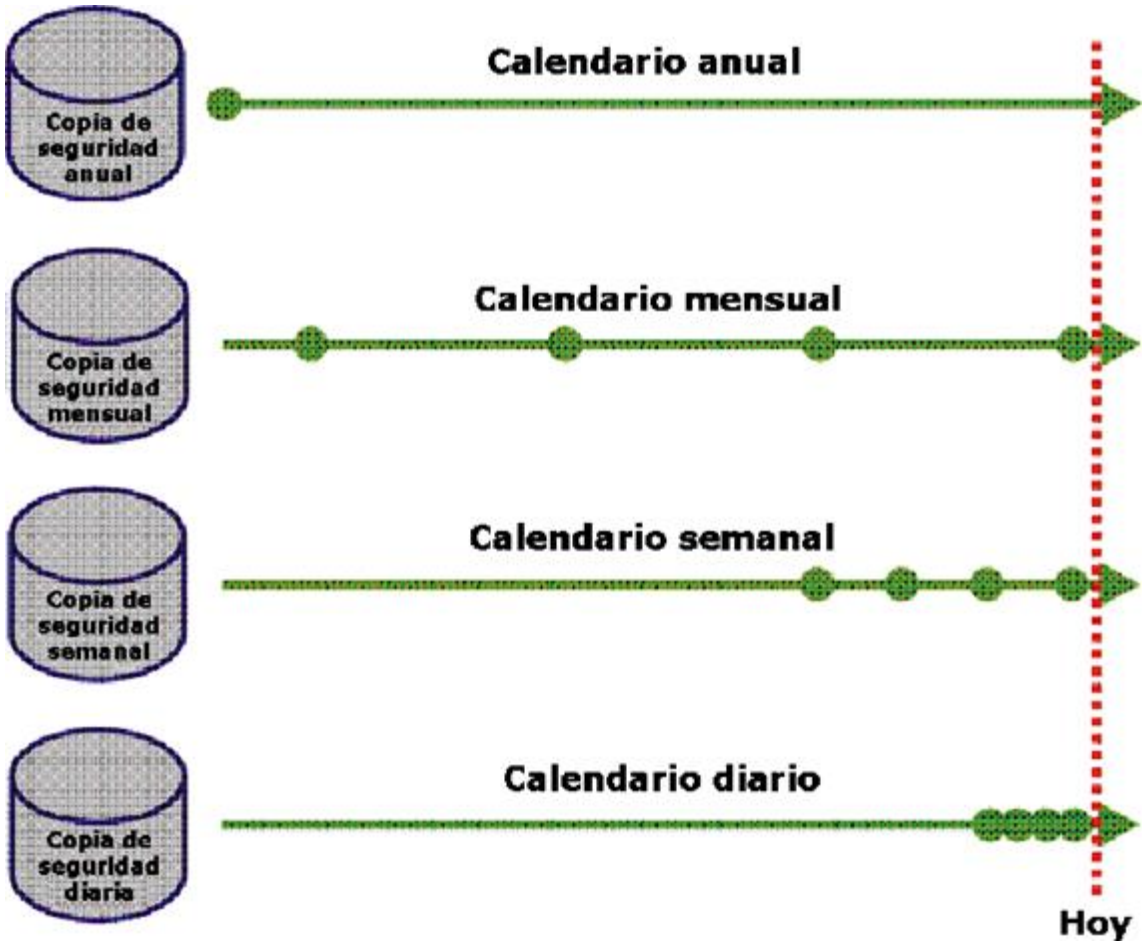
Durante el proceso en curso de calcular los niveles de servicio para cada una de las métricas, a menudo se obliga al motor a realizar un cálculo parcial para un período que no ha finalizado todavía. Para garantizar que vuelva al inicio del cálculo cuándo lleguen nuevos datos, realiza un tipo de copia de seguridad de su "estado" actual antes de proseguir con la siguiente tarea de cálculo. En este punto, toma una instantánea de las variables actuales y los valores en ese punto en el cálculo y guarda este "estado" en la base de datos.

El proceso de copia de seguridad de la lógica de negocios es un mecanismo por el cual el código de la lógica de negocios, incluyendo los valores de las variables, se codifica en un flujo binario y se guarda en la base de datos. Este mecanismo es también necesario para acelerar el rendimiento del motor de cálculo cuando hay que hacer recálculos. La copia de seguridad del estado se lleva a cabo de vez en cuando, y se utiliza en el recálculo y como medida de eficiencia para continuar con los cálculos.

Por ejemplo, si se requiere un recálculo para un mes retroactivamente, entonces en lugar de recalcular los resultados desde el inicio del contrato, se utiliza la fecha de recálculo más próxima al estado del que se ha hecho copia de seguridad y los cálculos se ejecutan a partir de ese estado en adelante.

El motor de cálculo utiliza heurística predefinida para determinar cuándo se necesita hacer una copia de seguridad y utiliza las capacidades de copia de seguridad para almacenar el estado codificado en la base de datos.

En la ilustración a continuación, los puntos rojos representan la copia de seguridad de un estado. Cuanto más atrás, menor será el número de estados con copia de seguridad tenidos en cuenta. La lógica detrás de este mecanismo es el supuesto de que normalmente se requiere recálculo para el período de tiempo con menos de un mes atrás.


Calendario anual
Copia de seguridad anual
Calendario mensual
Copia de seguridad mensual
Calendario semanal
Copia de seguridad semanal
Calendario diario
Copia de seguridad diaria
Hoy

Optimización del sistema para el recálculo

El proceso de cálculo de nivel de servicio requiere una cantidad significativa de recursos de CPU, memoria y almacenamiento. A continuación se incluye una lista de recomendaciones que pueden reducir la carga del equipo y mejorar la velocidad de cálculo.

Nota: Algunas de estas recomendaciones requieren valores de configuración internos que no están disponibles en la interfaz de usuario. En este caso, póngase en contacto con Soporte técnico de CA para obtener más información e instrucciones.

- **Cambie la configuración de almacenamiento de los estados.**

  Dependiendo de los retrasos conocidos del adaptador, se pueden establecer los parámetros de estado para adaptarlos oportunamente a sus requisitos; esto significa que puede cambiar el número y granularidad de los puntos de almacenamiento de los estados.

- **Configure el sistema para calcular solamente las unidades de tiempo que realmente se necesitan.**

  Las métricas pueden tener hasta siete períodos de granularidades diferentes: *año*, *trimestre*, *mes*, *semana*, *día*, *hora* y período de seguimiento. En algunas implementaciones, no son necesarias todas las granularidades. A través de la interfaz de usuario, se pueden desactivar las granularidades que no sean necesarias para los contratos no confirmados. Consulte la ficha de granularidades de cada métrica. Para cambiar las granularidades de las métricas de los contratos confirmados, póngase en contacto con Soporte técnico de CA.

  Nota: Evite que se calculen períodos operacionales que sean similares al período de seguimiento.

- **No calcule valores "sin correcciones" y "sin excepciones".**

  De forma predeterminada, el motor de cálculo calcula cuatro valores diferentes: los *proporcionados*, *proporcionados con correcciones*, *proporcionados con excepciones* y *proporcionados con correcciones y excepciones*. Esto se puede cambiar para calcular solamente el valor proporcionado.

  Nota: Para obtener más información, póngase en contacto con Soporte técnico de CA.

- **Cambie el orden de cálculo.**

  Se puede dar prioridad al orden de cálculo de PSL de las métricas para empezar por las métricas más importantes y continuar luego con el resto de las métricas.

Nota: Para obtener más información, póngase en contacto con Soporte técnico de CA.

- **Cree más de una instancia de PslWriter.**

  Al crear más de una instancia de PSL (del motor de cálculo), se puede dividir la carga de trabajo y aumentar la velocidad de cálculo.

  Nota: Para obtener más información, póngase en contacto con Soporte técnico de CA.

- **Planifique implementaciones para acortar el tiempo de recálculo.**

  Para optimizar el tiempo de recálculo, puede hacer lo siguiente:

  - Ejecute el adaptador más a menudo para reducir el número de eventos retrasados y evitar largas listas de trabajos pendientes de datos que el motor debe procesar.
  - Desactive los agentes que no use en la ficha de granularidad de las métricas.
  - Duplique las métricas y calcule los diferentes agentes mediante la misma métrica para equilibrar la carga de cálculo.
  - Utilice métricas intermedias para realizar cálculos habituales y compartir los resultados con el resto de las métricas que requieran los mismos datos.

- **Planifique implementaciones para reducir la cantidad de datos.**

  Para optimizar la cantidad de datos, utilice el adaptador para cargar sólo datos agregados (procesados). Agregar la información del origen de datos antes de enviarlo a la tabla de datos sin procesar de CA Business Service Insight aumenta la eficiencia de lectura de las entradas de PSL.

- **Siga las recomendaciones de configuración de PSL de CA Business Service Insight.**

  El motor de PSL se puede reconfigurar para disfrutar de un mejor rendimiento según el entorno de aplicación específico y los requisitos. Algunos de los parámetros se pueden establecer desde la interfaz de usuario, y otros solamente desde la tabla de configuración del sistema de base de datos.

  Nota: Consulte las recomendaciones de soporte para sus configuraciones de PSL.

Registros y alertas

Hay casos que requieren usar lógica de negocios para informar en el registro o para activar un mensaje de alerta. Esto es necesario cuando los mensajes se deben enviar basándose en el procesamiento del evento. Se puede enviar como alerta cualquier información recopilada durante el proceso de cálculo y que pueda ser valiosa. Por ejemplo, se puede enviar un mensaje de alerta cuando un evento específico se encuentre bajo el umbral de tiempo de resolución especificado o en el análisis de tendencias cuando se alcance cierto número de errores consecutivos.

El objeto "Out" es un objeto de lógica de negocios global que permite a la fórmula enviar alertas y mensajes de registro. Tiene dos métodos asociados con la forma siguiente:

```
Alert(<tipo de evento>, <nombre del recurso>, <valor1, valor2>, …<valor16>)
```

Este comando envía una alerta de un tipo de evento específico. No obstante, este tipo de evento se tiene que crear manualmente para la finalidad de esta alerta. El número de valores y su tipo deben corresponder a la definición de tipo de evento.

```
Log(<mensaje>,<nivel>)
```

Este comando envía un mensaje al registro del sistema. El primer parámetro es el mensaje de información comunicado, y puede ser texto libre. También se pueden añadir valores de variables a esta cadena para dar sentido contextual al mensaje. El parámetro "nivel" puede tomar uno de los valores siguientes:

	Descripción
	Se comunica un mensaje de advertencia.
	Se comunica un mensaje de error.
	Se comunica un mensaje de información solamente al funcionar dentro del ámbito de la lógica de negocios. Al funcionar en PslWriter, no se comunica ningún mensaje. Este es el nivel predeterminado. Se utiliza fundamentalmente para depuración.

Ejemplo:

El ejemplo siguiente se ha tomado de un caso donde la información de infraestructura del evento se esperaba antes que los detalles del incidente en sí. Se configuró un mecanismo de alerta para notificarle al administrador esta condición con objeto de solicitar corregir la incidencia.

```
Out.Alert "Alerta de sitio desconocido", Context.ClusterItem, Context.Rule

Out.Log("Evento fallido recibido para un sitio sin detalles de infraestructura: "
& Context.ClusterItem)
```

Comentarios de causa raíz de infracción y comentarios de evento

El comentario de causa raíz de infracción se puede establecer en la lógica de negocios para explicar los resultados de nivel de servicio. Los comentarios de causa raíz de infracción se asocian a métricas.

También es posible generar anotaciones de eventos, que son comentarios asociados a los eventos de datos sin procesar, en lugar de a la métrica. Estos dos tipos de comentarios se pueden consultar en los datos de informe.

Hay dos métodos en el objeto "Tools" de la lógica de negocios que permiten crear registros de causa raíz de infracción y de anotaciones de eventos:

- **Tools.AddRootCauseComment** (Text, Timestamp, [resourceId])
- Guarda un comentario de causa raíz. Esta información puede ser útil posteriormente en informes generados. El comentario de causa raíz guardado explica una situación específica durante la ejecución de la fórmula de lógica de negocios en un momento concreto. El parámetro de información especifica el comentario que se deberá escribir. El método recibe una marca de tiempo que se guardará juntamente con el comentario. También acepta un parámetro ResourceId en el que se especifica el recurso relacionado con el contexto de método. (Este parámetro es opcional y se puede omitir).

- **Tools.AddEventAnnotation** (EventId, Text)
- Estos métodos se pueden utilizar en cualquier sitio dentro de la lógica de negocios y debe tenerse en cuenta el contexto de dónde se aplicarán. Añadir un comentario de causa raíz puede ser más relevante al final de un período de cálculo, cuando se conoce el motivo por el que el nivel de servicio es inferior a lo esperado para ese período. Supongamos que, por ejemplo, durante el período de un mes se han producido cuatro interrupciones, todas causados por un solo dispositivo. El comentario de causa raíz se podría entonces recopilar a partir de esta información acerca de las interrupciones, para que cuando se consulten los informes para este período, se pueda ver qué ha contribuido a cualquier incumplimiento de nivel de servicio durante este tiempo. El comando AddRootCauseComment es el más recomendable para la rutina del controlador de eventos OnPeriodEnd, o para otra función similar que se ejecute hacia el final del período calculado.
- No obstante, resulta más adecuado agregar una anotación del evento para el procesamiento de eventos de datos sin procesar, además de que facilita el uso para OnXXXEvent (el controlador de eventos personalizado especificado en la declaración de registro). Dentro de este controlador de eventos, todos los campos específicos del evento en sí están disponibles mediante el objeto eventDetails.
- Este sería un ejemplo del aspecto que esto podría tener dentro de la lógica de negocios:

```
Sub OnPeriodEnd(Time)
        pctAvailable = (TotalTime-OutageTime) / TotalTime
        Tools.AddRootCauseComment “Incumplimientos causados por los
   elementos siguientes: ” & 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”) & “ no resuelto en el tiempo objetivo
    de 6 horas”
    End If
End Sub
```

Separación de excepciones de ranuras de tiempo

La lógica de negocios de CA Business Service Insight no recibe eventos de excepción. Lo que recibe es un evento OnTimeslotExit cuando se inicia un período de excepción y un evento OnTimeslotEnter cuando termina un período de excepción. La lógica de negocios, por lo tanto, no puede distinguir entre tiempos de excepción y tiempos fuera de la ranura de tiempo. Además, no puede distinguir entre tipos de excepción. Por ello, no es posible implementar una lógica diferente para el comportamiento de los tiempos de excepción y para el comportamiento de "fuera de ranura de tiempo".

Una forma de implementar excepciones especiales (es decir, una excepción que no se comporta como un período "fuera de la ranura de tiempo") es definir tipos de eventos dedicados, en lugar de utilizar el mecanismo integrado en CA Business Service Insight para gestionar excepciones. Estos eventos se generan leyéndolos de un origen de datos dedicado mediante un adaptador.

Estas excepciones se pueden almacenar en una hoja de cálculo de Excel (o en cualquier otro origen de datos); a continuación, un adaptador puede cargar los datos y generar una respuesta: eventos de entrada de excepción y eventos de salida de excepción. Las excepciones también se pueden agregar mediante correcciones. Además de la corrección, se deberá definir un recurso ficticio y asociarlo a estos eventos para hacer el registro. Este recurso no sirve a otra finalidad que la de hacer de marcador de posición, como requiere el comando.

Para poder gestionar los tiempos de excepción comunicados por estos eventos dedicados, la fórmula de lógica de negocios se deberá registrar con estos eventos de excepción además del registro normalmente necesario para los eventos de datos sin procesar utilizados en el cálculo.

Se recomienda que el experto en la lógica de negocios incluya un campo para el tipo de excepción en el tipo de evento, lo cual permitirá gestionar de manera diferente los diversos tipos de excepciones especiales.

Este enfoque presenta las características siguientes:

- Separa dos conjuntos de excepciones: las estándar y las especiales.
- Las excepciones especiales no tienen una interfaz de usuario para las tareas de mantenimiento.
- Los informes basados en las excepciones generadas como eventos por parte de un adaptador no registran comentarios sobre su existencia. En aquellos casos en los que se utilizó el mecanismo de corrección, sí hay un comentario. Se recomienda usar el método de corrección para mantener la integridad de los resultados producidos por el sistema.

- Ninguna especificación "con/sin excepciones" tendrá en cuenta esas excepciones.
- Cuando se emplea el mecanismo de corrección, se aplica la corrección con/sin corrección.

Una vez implementado, se recomienda que el experto en la lógica de negocios aplique la lógica a todas las métricas del sistema.

Si es necesario, existe otro método para aplicar una excepción a un solo recurso. Este método implica utilizar el estado "Vigente" de los recursos. Establecer el estado de un recurso como "No vigente" significa que, durante este período, el motor de cálculo ignorará todos los eventos de datos sin procesar que se envíen para ese recurso. Establecer un período de tiempo en el que el recurso no está vigente creando nuevas versiones del recurso, una al inicio del período de excepción y otra al final del período de excepción.

Sin embargo, si el recurso forma parte de una métrica en clúster y el recurso va a entrar en vigencia y dejar de estar vigente en el mismo período de cálculo, solamente se tendrá en cuenta en el resultado el último período donde el recurso estuvo vigente, como se ha indicado anteriormente. En ese caso, se aconseja utilizar la función de atributos personalizados. Se puede gestionar un atributo adicional para el recurso que indique su estado; la fórmula de lógica de negocios solicitará el estado del recurso en cada lugar relevante del script.

Consideraciones en cuanto a rendimiento y memoria

A continuación se indican una serie de situaciones que se deberán tener en cuenta al diseñar soluciones de lógica de negocios. Las situaciones descritas corresponden a casos en los que el rendimiento del motor de cálculo se puede ver afectado negativamente:

- **Parámetros**

  Si se requiere un valor de un parámetro en el código, se recomienda crear una variable global para asignarle el valor del parámetro. Además, cuando se requiera el valor del parámetro, utilice la variable global en su lugar. Esto previene situaciones en las que el motor crea la asignación de parámetros para cada invocación de parámetro.

- **Uso de asignaciones en métricas en clúster**

  Cuando vaya a utilizar objetos de asignación globales grandes en la lógica de negocios para métricas en clúster, hágalo con sumo cuidado. Mientras el motor calcula una métrica en clúster, está ocupado cargando las variables globales desde los estados anteriores para cada elemento en el clúster por separado.

- **Registro de la lógica de negocios**

  Se recomienda filtrar los eventos de datos sin procesar únicamente por los métodos de registro. Al agregar un filtrado interno mediante una declaración "if" dentro del código, se tardará más tiempo en procesar. Y lo que es más importante, el motor necesitará más tiempo para extraer y procesar los registros de datos sin procesar no necesarios.

- **Evitar el uso de Dispatcher.RegisterByEventType**

  Mejora el rendimiento. Este método realiza un registro en todos los recursos en el sistema, y no solamente en los recursos con eventos de ese tipo concreto. Así pues, todos los cambios en el recurso afectarán a los cálculos de métricas. Otra desventaja de utilizar este método de registro es el tiempo de ejecución de las métricas al acceder a los datos sin procesar. Después tendrá que filtrar desde los datos sin procesar solamente los eventos con el tipo de evento específico, e ignorar los otros eventos.

- **Dispatcher.Register**

  Cuando esté utilizando Dispatcher.Register, compruebe *siempre*que especifica el 3er parámetro. Realizar un registro sin el 3er parámetro es exactamente como llevar a cabo registros por tipo de evento (Dispatcher.RegisterByEventType). Dicho de otra manera, asegúrese de que utiliza como mínimo otro parámetro además de los dos primeros.

- **Granularidad del cálculo**

Es importante activar solamente los agentes requeridos para el cálculo y para obtención de detalles. El cálculo de todas las unidades de tiempo del agente conlleva mucho procesamiento.

Cómo activar los contratos (gestor de contratos)

Los contratos se activan confirmándolos. (Para obtener más información, consulte la ayuda en línea).

Al activar el contrato se activa también el motor para iniciar los cálculos para las métricas del contrato y empezar a producir resultados para el contrato.

Antes de activar el contrato, asegúrese de que se cumplen todas las condiciones siguientes:

- Todas las métricas deben tener una lógica de negocios definida para ellas (en la métrica o como un módulo vinculado) que se haya probado previamente y que no contenga errores.
- Todas las métricas tienen que tener definido un umbral de cuadro de mandos. (Para obtener más información, consulte la ayuda en línea). Es importante que los umbrales se hayan definido previamente para que durante el proceso de cálculo el cuadro de mandos pueda evaluar los límites para la métrica.
- Las fechas de vigencia tienen que corresponder al período de tiempo correcto y tiene que haber registros de datos sin procesar disponibles. Compruebe también que los resultados sean los previstos al usar el ámbito de la lógica de negocios.

Una vez el contrato se haya confirmado, compruebe que la activación haya comenzado correctamente y que los cálculos vayan progresando según lo previsto.

Siga estos pasos:

1. Compruebe que todos los componentes de servicio de CA Business Service Insight se hayan iniciado (especialmente el motor de cálculo, que incluye tanto PSLWriter como PenaltyWriter). Se recomienda que todos los componentes de servicio se estén ejecutando cuando sea necesario realizar un cálculo.
2. Diagnostique el contrato: la función de diagnóstico del contrato muestra los resultados de una serie de pruebas realizadas en todas las métricas del contrato (y las fórmulas de sanción, si se utilizan). Se indica la severidad del resultado de la prueba, junto con una sugerencia de resolución. Se recomienda realizar un diagnóstico siempre que se active un contrato, así como después de finalizar el cálculo para ese contrato.
3. Genere el informe de formato libre "Estado de cálculo". Este informe se incluye en la instalación inicial de CA Business Service Insight, y se encuentra en la carpeta del paquete "Admin Reports". Proporciona información acerca del progreso de cálculo y se puede utilizar en este punto para verificar si el motor de PSL está avanzado y si el cálculo ha finalizado. Revise este informe para evaluar si hay problemas potenciales en los cálculos.

El informe tiene los campos de columna siguientes:

Campo	Descripción
Contract	Indica el nombre del contrato. La lista contiene los nombres de los contratos vigentes y no vigentes.
Metric	Indica el nombre de la métrica dentro del contrato. La lista contiene todas las métricas contenidas dentro de cada contrato.
Tracking Period	Indica el período de cálculo de la métrica. La lista mostrará una entrada para cada unidad de tiempo de cálculo de la métrica según los agentes activos y en base a la definición de granularidad de la métrica. Cuando el período de cálculo sea el período de seguimiento, se incluirá indicación al respecto.
Updated Up To	Indica la hora de la última actualización del resultado. Esto indica que el resultado para la métrica específica está disponible hasta esta fecha. Por ejemplo, si se muestra 01/01/2006, indica que todos los resultados para esa métrica en esa unidad de tiempo se han actualizado hasta esta fecha, y que hay informes disponibles para ella hasta esta fecha.
Last Cycle Begin At	Indica el momento en el que empezó el ciclo del cálculo en el que se actualizó el resultado de la métrica.

Requires Recalculation From	Cuando el motor etiqueta cierta métrica para recálculo y todavía no se ha actualizado, la fecha resultante aparecerá aquí especificando el momento a partir del cual los resultados no son relevantes y por lo tanto requieren actualización. Esto puede ocurrir en cualquier caso de recálculo.
Last Update At	Momento en el que el motor actualizó el registro con el último resultado.
Handled by Engine #	Indica el número del motor asignado para gestionar el cálculo de la métrica específica.

Este informe puede proporcionar también la información siguiente basada en los datos sin procesar disponibles:

- El tiempo que tardó el motor en calcular una sola métrica. Clasificando todas las métricas calculadas en un solo ciclo por su hora de actualización se puede ver cuánto tiempo tardó el motor en actualizar cada métrica. Todos los registros con el mismo valor "Last Cycle Begin At" se calculan durante el mismo ciclo y la hora de actualización es la hora de "Last Update At". Se puede evaluar el tiempo que tardó el motor en calcular la métrica entera con las unidades de tiempo de su agente subyacente, así como con cada una de las unidades de tiempo.
- El tiempo que tardó el motor en calcular un contrato completo. Esto se hace examinando la hora de actualización de la primera métrica del contrato y la hora de la última actualización de la última métrica del contrato, y calculando luego el tiempo entre medio.

Recálculo completo de la métrica de contrato

El proceso de recálculo en el contexto actual corresponde a la iniciación de un recálculo del sistema completo bien por parte del experto en el origen de datos o del experto en la lógica de negocios. No es el recálculo del motor que se realiza durante un proceso de cálculo normal. Este tipo de acción se realiza normalmente durante o después del proceso de configuración del contrato, tras identificar funcionamientos incorrectos.

Se recomienda comenzar una operación de recálculo completa solamente una vez el sistema esté en un estado estable (es decir, no al construir el sistema), antes de activarlo en el sitio.

Actualmente el recálculo se realiza mediante un script SQL en la base de datos. Este script limpia la tabla de PSL y todas las tablas de soporte asociadas que formen parte del proceso de cálculo. El equipo de soporte de CA deberá aprobar y validar este script antes de que se ejecute, ya que pueden haberse producido cambios eventuales de estructura en el sistema que podrían haber dado lugar a cambios en el esquema de la base de datos y/o objetos dependientes.

Nota: Antes de ejecutar el script, se deberán detener todos los componentes de servicio, y después de la ejecución se podrán reiniciar estos componentes de servicio para volver a reiniciar los cálculos.

Las situaciones siguientes pueden ser indicativas de que es necesario realizar un recálculo completo:

- Problemas con las definiciones en el contrato: si está en esta etapa, puede haber un destino de métrica definido incorrectamente o un umbral de métrica erróneo.
- Errores en el contrato.
- Necesidad de volver a evaluar los umbrales de cuadro de mandos o de volver a generar alertas de SLA.

Nota: Si es necesario, póngase en contacto con Soporte de CA para hablar de esta opción y también para obtener copias de los scripts relevantes para la versión de CA Business Service Insight instalada.

Creación de entregas (gestor de contratos)

En esta etapa se configuran todas las salidas del sistema para cumplir con los requisitos necesarios, incluido crear todos los informes y aplicar el formato correspondiente.

La configuración de entregas incluye lo siguiente:

- Definir las opciones de seguridad (permisos de usuario y grupos).
- Crear informes guardados.
- Crear folletos.
- Crear plantillas de exportación de contratos.
- Crear vistas de Navegador de entrega del servicio (SDN).
- Configurar mapas del cuadro de mandos.
- Crear perfiles de alertas de nivel de servicio.

Definición de opciones de seguridad (administrador)

Al definir opciones de seguridad, es necesario configurar los usuarios de CA y sus permisos asociados. Estos valores de configuración incluyen definir qué información deberá estar accesible para un usuario (qué entidades dentro del sistema puede ver o editar el usuario). Estos permisos se pueden establecer en diversos niveles de grupos de usuarios, roles, o incluso individualmente por usuario. La información accesible se define en relación con las partes contratantes y se puede establecer directamente en el usuario, o bien heredada del grupo de usuarios al cual pertenece el usuario.

Es en esta etapa cuando se configuran los roles principales y los grupos de usuarios asociados; así, cuando se agregue un usuario nuevo, sólo se tendrá que adjuntar a un grupo para heredar los valores de configuración relevantes.

Las acciones permitidas se configuran en los roles, y se transfieren al usuario asignándole directamente un rol o mediante el grupo de usuarios al cual pertenezca el usuario. Las acciones del cuadro de mandos permitidas relacionadas se definen también en el rol adjunto.

Se recomienda que el administrador determine qué grupos de usuarios y roles se deberán definir, así como el permiso necesario para poder configurar una estructura que permita agregar fácilmente usuarios.

Cómo crear informes

Para crear un informe, utilice el proceso siguiente:

1. Cree todas las carpetas necesarias para el informe: las carpetas se deberán crear por adelantado para que estén disponibles al guardar cada nuevo informe. Normalmente cada contrato tiene su propia carpeta, incluida una carpeta ejecutiva para los informes de alto nivel.
2. Defina los criterios de filtrado de informe mediante el asistente de informes: para crear un informe guardado se comienza por generar el informe mediante el asistente de informes. En el asistente de informes se seleccionan los criterios de filtrado necesarios y se genera el informe. Los administradores de sistemas pueden establecer parámetros de informe que designen campos definidos por el usuario que deberá rellenar el usuario o la persona que esté viendo el informe. Esto permite generar resultados de informe específicos según el interés de ese usuario.

   Al generar un informe para establecerlo como un informe guardado, es muy importante definir los criterios de filtrado lo más flexiblemente posible. Esto evita tener que realizar un trabajo innecesario a medida que el sistema aumente y evolucione. El objetivo deberá ser garantizar que cada informe muestre siempre información vigente y actualizada. Por ejemplo, si un informe actualmente muestra tres componentes de servicio, al añadir nuevos componentes de servicio en el futuro, es importante que se agreguen automáticamente al informe sin tener que configurarlo de nuevo. O, si se está generando el informe por mes y el informe muestra actualmente tres meses, entonces en el mes siguiente deberá mostrar cuatro meses, incluido el último mes. En muchos casos los informes deberán mostrar siempre los datos de los últimos 6 meses. Estos tipos de informes de "ventana deslizable" son sumamente útiles en contraposición con los informes de duración fija, ya que no requieren ningún cuidado adicional una vez creados.

A continuación se indican algunas sugerencias para establecer los criterios de filtrado para los informes guardados:

Ficha Criterios

- La opción "Sólo datos empresariales" limita los datos mostrados sólo a los datos relacionados con el período de seguimiento de la métrica.
  - Dé siempre preferencia a las opciones de *agrupar por* o *informar por*, en lugar de seleccionar que se informe sobre métricas específicas.

- Establezca un intervalo de tiempo dinámico. Si esto se establece en un intervalo de tiempo fijo, el informe mostrará siempre los mismos resultados (por ejemplo, "Los últimos 3 meses" sería un intervalo dinámico).

Ficha Varios

- Seleccione si se deberán presentar en el informe o no los "períodos incompletos". Si así es, entonces seleccione esta opción en la lista. Normalmente, al configurar informes, los períodos incompletos se excluyen, ya que no es un resultado final y por lo tanto no es un resultado de negocio.
- Establezca el diseño del informe: una vez generado el informe, se puede especificar su diseño mediante la herramienta de diseño en la página del informe. El diseño puede obedecer a requisitos específicos dependiendo de su destinatario. Se recomienda crear varios diseños, uno para cada posible escenario, así como aplicar estos diseños al resto de los informes que se tienen que generar. Por consiguiente, al definir los criterios del informe, en la ficha *Visualizar,* elija *Utilizar diseño de*.

  Nota: Para ver sugerencias sobre cómo usar la herramienta de edición para definir el diseño de los informes, consulte la sección siguiente.

Ficha General

- Guarde el informe: una vez se ha generado y diseñado el informe, se puede guardar en la carpeta relevante.
- Durante el proceso de almacenamiento del informe, éste se asocia a los usuarios que tienen acceso a él. Es por lo tanto importante tener definido previamente un grupo de usuarios, para poder asociar los informes a usuarios. Se pueden adjuntar al informe los usuarios con los permisos correctos en una etapa posterior mediante las opciones de carpeta.
- Asocie los informes relacionados para simplificar la navegación entre informes similares o que tengan relación uno con otro.
- En la página Carpeta de informes se pueden crear informes, grupos de informes, informes compuestos, informes de formato libre, folletos, accesos directos y carpetas, así como buscar informes.

  Desde el menú Informes, haga clic en Carpetas del informe. Se muestra la página Carpeta de informes con una lista de informes guardados.

Informe Desviación

El motor de CA Business Service Insight calcula el valor Desviación automáticamente para las métricas que tienen un destino. El valor representa la diferencia entre el nivel de servicio real y el destino. La fórmula de cálculo de la desviación, calculada automáticamente por CA, cambia según la definición del nivel de servicio: si el destino del nivel de servicio es un valor máximo (cuando el nivel de servicio real es "no superior a"), o un valor mínimo (cuando el nivel de servicio real es "no inferior a"). Consulte la información a continuación para ver un ejemplo de cómo cambia la fórmula:

Declaración de destino	Umbral de nivel de servicio	Fórmula de desviación
El servicio deberá estar disponible como mínimo el 99 % del tiempo programado.	El destino es el umbral mínimo.	$Service_{Dev} = (-1)x \frac{Service_{SL} - Service_{SLT}}{Service_{SLT}}$
El promedio de tiempo de reparación (MTTR) no deberá superar las 4 horas por mes.	El destino es el umbral máximo.	$Service_{Dev} = \frac{Service_{SL} - Service_{SLT}}{Service_{SLT}}$

Donde $Service_{Dev}$ = Desviación del servicio

Donde $Service_{SLT}$ = Rendimiento del servicio esperado

Donde $Service_{SL}$ = Rendimiento del servicio real

El ejemplo siguiente ilustra un cálculo de desviación mínimo:

El servicio deberá estar disponible como mínimo el 99 % de la ranura de tiempo programada. El nivel de servicio real es del 98 % durante la ranura de tiempo programada.

$$Service_{Dev} = (-1)x \frac{98\% - 99\%}{99\%} = 1.0101\%$$

Los informes Desviación se utilizan para vistas de alto nivel de garantías de una naturaleza externa (tipo diferente de cálculo), y para agregarlos en una sola barra con base común.

Si, por ejemplo, en el contrato corresponde a la matriz siguiente:

Servicio - Help Desk	**Dominio del servicio - Gestión de tickets Prioridad 1**	**Dominio del servicio - Gestión de tickets Prioridad 2**	**Dominio del servicio - Gestión de tickets Prioridad 3**
	Promedio de tiempo de resolución P1	Promedio de tiempo de resolución P2	Promedio de tiempo de resolución P3
	% de tickets P1 resueltos dentro de T1	% de tickets P2 resueltos dentro de T1	% de tickets P3 resueltos dentro de T1
	% de tickets P1 con respuesta dentro de T1	% de tickets P2 con respuesta dentro de T1	% de tickets P3 con respuesta dentro de T1
	Promedio de tiempo de respuesta P1	Promedio de tiempo de respuesta P2	Promedio de tiempo de respuesta P3

La gráfica siguiente muestra los resultados de generar un informe de dominio del servicio filtrado por servicio de Help Desk.

El informe anterior permite al gestor de nivel de servicio ver el rendimiento el Help Desk basándose en diversas prioridades, sin tener en cuenta el contrato o el tipo de obligación.

Todas las métricas del Help Desk se agregan en una sola barra según su prioridad.

Por ejemplo, la barra de prioridad 1 muestra las tres métricas definidas dentro de la métrica, y agrega su desviación en un solo valor. (El método de agregación se puede elegir en el asistente de informes como promedio, mínimo o máximo).

Con un informe así, el gestor puede concluir, por ejemplo, que tiene que invertir más recursos en los tickets de prioridad 1 o cambiar los contratos relacionados con estos tickets.

Este ejemplo muestra que el modelado proporciona tanto el informe para una sola obligación que muestra si un contrato se ha cumplido o infringido, pero también un informe de gestión más amplio que permite al gestor de nivel de servicio gestionar sus recursos más eficazmente y mejorar en consecuencia sus componentes de servicio.

Informes de formato libre

Los informes de formato libre permiten a los usuarios generar informes basados en consultas SQL de la base de datos de CA Business Service Insight o de cualquier otro origen de datos externo al cual se pueda acceder mediante una conexión de servidor de CA Business Service Insight. Esto también incluye otros tipos de orígenes de datos a los cuales se puede acceder mediante ODBC, como Excel, Access, Lotus Notes, archivos de texto, etc. Los informes de formato libre se utilizan habitualmente para configurar informes estadísticos basados en datos creados por los comandos de lógica de negocios Tools.SaveRecord y Tools.SaveFields.

Los informes de formato libre se conectan mediante una cadena de conexión a la base de datos seleccionada y ejecutan una consulta SQL en la base de datos mediante una cadena de consulta. Los parámetros pueden agregarse a las dos cadenas para crear informes dinámicos que permitan que el usuario especifique o seleccione valores determinados para incluirlos en la consulta, como el nombre de usuario y la contraseña para conectarse a la base de datos.

Los informes de formato libre se muestran en las fichas Gráfico, Datos y Filtrar de forma similar a los informes generados mediante el asistente de informes.

Nota: Los informes de formato libre pueden incluir gráficos solamente si todas las columnas, aparte de la primera columna, son numéricas. Los datos de la primera columna se utilizan para los títulos del eje X. Los nombres de la columna se utilizan para otros títulos.

Dado que los informes de formato libre acceden directamente a una base de datos y a una consulta SQL abierta, el mantenimiento resulta problemático. Se deberá tener mucho cuidado de no afectar a los datos subyacentes que sirven como origen para los informes de formato libre. Cuando se generan informes de un origen de datos externo, se recomienda configurar un proceso de notificación para garantizar que esos orígenes de datos no cambien sin primero consultar con el gestor de contratos responsable de los informes de datos de formato libre.

Información general que tener en cuenta al crear informes de formato libre.

- Por problemas de formato de fechas internacionales, es a menudo útil especificar el formato exacto de la fecha creando una fórmula personalizada en la lógica de negocios. Esto convierte la fecha a una cadena con el mismo formato siempre y evita incidencias de formato de fecha US frente a formato de fecha UE. Esta conversión se deberá utilizar para fechas que se escriban a partir de la tabla T_SLALOM_OUTPUTS a través de los comandos Tools.SaveFields o Tools.SaveRecord. A continuación se incluye una fórmula de muestra:

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
```

- Cuando se utiliza el parámetro de tiempo de los controladores de eventos de la lógica de negocios, o cualquier marca de tiempo del objeto eventDetails, es necesario tener siempre presente el efecto del cambio correspondiente a la zona horaria. Tenga en cuenta que, al escribir las fechas en la tabla puede hacerse en hora UTC en lugar de en la hora local. Puede que haya que usar la función Tools.GetLocaleTime() para corregir esta situación.
- Se puede utilizar también la utilidad Diseñador de informes (cuando un informe de formato libre produce una gráfica) para personalizar el aspecto.
- Las opciones de exportación a PDF para los informes de formato libre son personalizables a través de la sección de parámetros del informe de la pantalla de configuración de los informes de formato libre, como el diseño de página (horizontal, vertical, etc.).
- Se puede insertar HTML en los informes para ejecutar diferentes funcionalidades, como agregar hipervínculos o cambiar el color de las columnas o las filas, los tipos de letra o la configuración de visualización.
- Se pueden crear vistas de base de datos (funcionalidad de Oracle) en la tabla T_SLALOM_OUTPUTS para simplificar el SQL necesario para los informes.

- Al especificar parámetros para los informes, se pueden establecer de distintas formas, como texto libre, numérico (validar), contraseña (caracteres ocultos, útil para contraseñas para conexiones a bases de datos), fecha (mediante la utilidad de selección de fecha proporcionada) y listas (creadas especificando una declaración SQL para rellenar la lista).
- Los valores de parámetro especificados en la sección de SQL del informe de formato libre necesitan sustituirse con el nombre del parámetro, precedidos del símbolo "@". Por ejemplo, @PARAM1. Si los valores son cadenas, es posible que sea necesario usar comillas alrededor de este valor de parámetro.

Informes de formato libre de histograma genérico

La consulta siguiente se puede utilizar en un informe de formato libre para presentar la distribución de valores en una tabla por porcentaje, como muestra el gráfico siguiente:

En el gráfico superior se puede ver qué proporción (por porcentaje) de los valores están por debajo de 11,5 (0 %), por debajo de 804,74 (~50 %) y por debajo de 1435,53 (100 %).

Si el SLA especifica destinos como "x% de los valores deberá estar por debajo de y", los resultados de este formato libre ayudan a encontrar los valores x e y que garantizan la conformidad con el SLA.

En la consulta se utilizan los parámetros siguientes:

- **@Query**: una declaración de selección con el formato siguiente; select *algún_valor* **val** from *alguna_tabla*. El valor de alias es obligatorio. Esta consulta proporciona los valores cuya distribución se analiza.
- **@Buckets**: el número de valores en el eje x. Los valores de origen se redondean a estos números. Por ejemplo, si especifica @Buckets = 100, entonces los valores de los datos de origen se dividen en 100 grupos, y la consulta calculará cuántos valores se incluían dentro de cada grupo. Cuanto más alto sea el valor @Buckets, más exacto será el resultado, si bien el gráfico no será muy atractivo visualmente.
- **@Relation**: la dirección de acumulación. Posibles valores: "<=" (cuando el destino es no inferior a) o ">=" (en cualquier otro caso).

La consulta se puede ejecutar para el origen de datos o para T_SLALOM_OUTPUTS para obtener los mejores resultados.

La consulta siguiente genera la gráfica que se muestra arriba:

```
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 continuación se incluye una lista de parámetros de muestra (como XML) que se podría utilizar:

```
<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>&gt;=</value>
                                        <text>providing too little</text>
                                   </item>
                                   <item>
                                        <value>&lt;=</value>
                                        <text>providing too much</text>
                                   </item>
                              </list>
                         </param>
                    </params>
               </query>
```

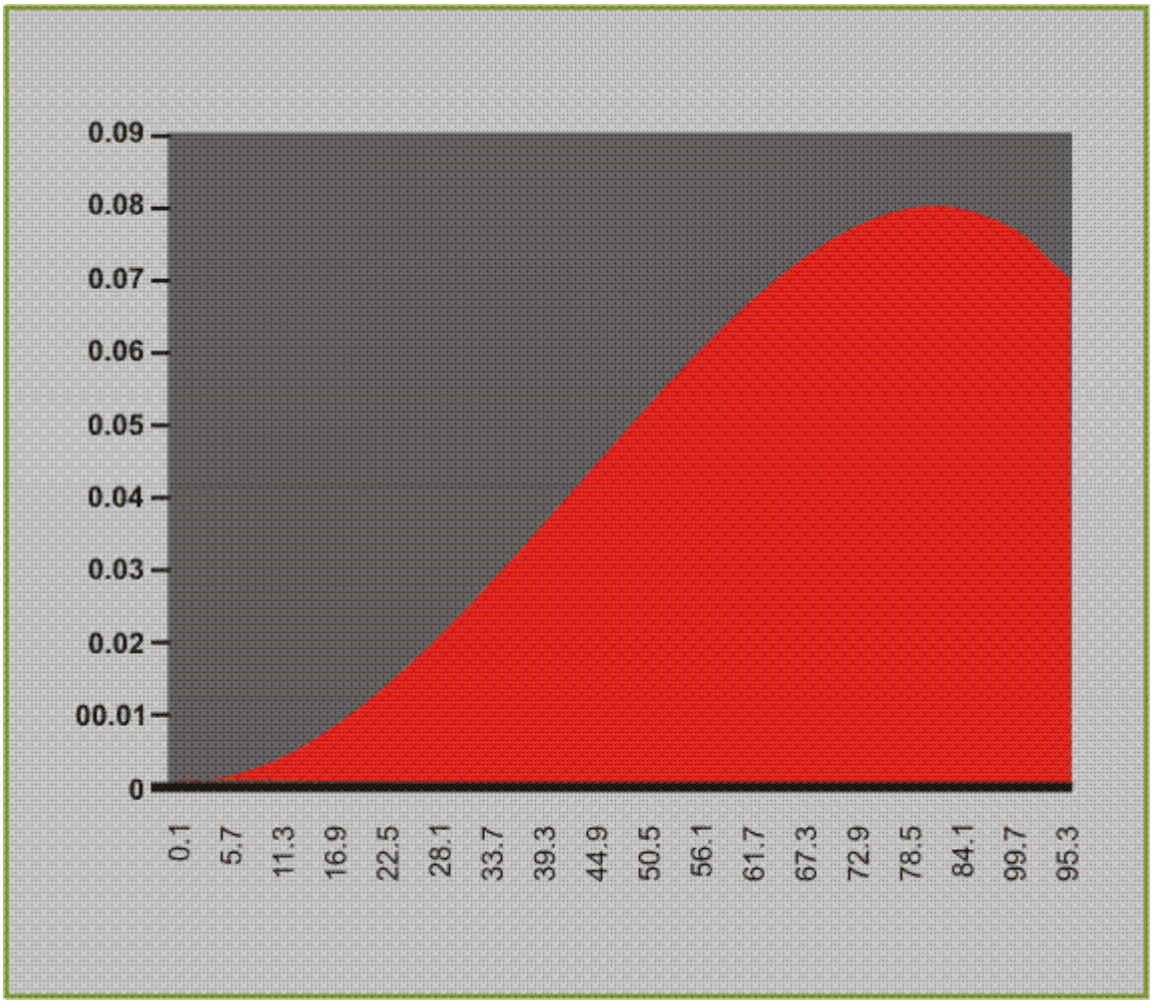
```
</custom>
```

Comentarios

- La consulta se ha optimizado para Oracle y SQL Server. Para otros orígenes de datos ODBC, es posible que haya que agregar "as" antes de los alias de columna, además de aplicar otras modificaciones.
- Al exportar resultados a Excel, se recomienda que el usuario genere un informe XY (de dispersión) para representarlos. Si el gráfico lleva puntos, también se podrá ver la dispersión de los valores.

Informe de formato libre de distribución normal genérica (gausiana)

La consulta que se detalla a continuación se puede utilizar en un informe de formato libre para presentar la distribución normal de los valores en una tabla, tal como muestra el gráfico siguiente:



En la consulta se utilizan los parámetros siguientes:

- @Query: una declaración de selección con el formato siguiente:

```
select min (algo) min_val,
          max (algo) max_val,
          avg (algo) avg_val,
          stddev (algo) stddev_val
from nombre_tabla
```

  Es obligatorio indicar **x_val**. Esta consulta proporciona los valores cuya distribución se analiza.

- @Buckets: el número de valores en el eje x. Los valores de origen se redondean a estos números. Por ejemplo, si especifica @Buckets = 100, entonces los valores de los datos de origen se dividirán en 100 grupos, y la consulta mostrará el valor de distribución normal para cada grupo. Cuanto más alto sea el valor @Buckets, más exacto será el resultado, si bien la operación de cálculo será más intensiva. 100 es una buena elección para @Buckets.

De nuevo, la consulta se puede ejecutar para el origen de datos o para T_SLALOM_OUTPUTS para obtener los mejores resultados.

La consulta siguiente generará la gráfica que se muestra arriba:

```
select (max_val-min_val)/@Buckets*serial,
     (1/stddev_val*sqrt (2*3.14159265359))*exp ( -1/(2*power (stddev_val,
2))*power (((max_val-min_val)/@Buckets*serial-avg_val), 2))
from     (
          @Query
     ),
     (
          select     rownum serial
          from  t_psl
          where  rownum < @Buckets
     )
order by serial
```

Sus parámetros XML correspondientes:

```
<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 min(success_rate) min_val,max(success_rate) max_val,avg(success_rate)
avg_val,stddev(success_rate) stddev_val
       from PDP_Context_Activation_Success.CSV
</value>
                              <text>PDP Context Activation Success</text>
                         </item>
                         <item>
                              <value>
select min(throughput) min_val,max(throughput) max_val,avg(throughput)
avg_val,stddev(throughput) stddev_val
       from [gprs throughput volume by apn.csv]
</value>
                              <text>Throughput in Kb</text>
                         </item>
                         </list>
               </param>
               <param name="@Buckets" disp_name="X Axis Values" type="LIST">
                    <value>100</value>
                    <list>
                         <item>
                              <value>25</value>
                              <text>25</text>
                         </item>
                         <item>
                              <value>50</value>
                              <text>50</text>
                         </item>
                         <item>
                              <value>100</value>
                              <text>100</text>
                         </item>
                         <item>
                              <value>250</value>
                              <text>250</text>
                         </item>
                         <item>
                              <value>500</value>
                              <text>500</text>
                         </item>
                         <item>
                              <value>1000</value>
                              <text>1000</text>
                         </item>
                    </list>
               </param>
          </params>
     </query>
</custom>
```

Comentarios

- La consulta se ha optimizado para Oracle y SQL Server. Para otros orígenes de datos ODBC, es posible que haya que agregar "as" antes de los alias de columna, además de aplicar otras modificaciones.
- Al exportar resultados a Excel, se recomienda que el usuario genere un informe XY (de dispersión) o un informe de área para representarlos.

Configuración de páginas de cuadro de mandos

La sección siguiente incluye una recomendación sobre el contenido que configurar para los usuarios de CA Business Service Insight. La recomendación es de alto nivel y deberá tener en cuenta los requisitos específicos del cliente. Las páginas se describen a continuación en el contexto de un usuario concreto descrito a través de su rol en la organización.

Páginas del gestor ejecutivo

Supongamos que un gestor ejecutivo podría estar interesado en una vista de alto nivel para todas las cuentas de los países por departamentos, etc. Los gestores ejecutivos normalmente no ejecutan tareas en sí y requieren vistas que le proporcionen información para tomar decisiones en términos de estrategia. Por lo tanto, en las asignaciones de gestores ejecutivos y en las vistas del cuadro de mandos agregadas, puede resultar más relevante mostrar un estado contractual en lugar del estado actual.

Por ejemplo, en la página Descripción general pueden aparecer las vistas del cuadro de mandos siguientes:

Vista del cuadro de mandos	Descripción
Cuentas críticas	Incluye todas las partes contratantes marcadas como sensibles. El gestor ejecutivo selecciona los contratos o las partes contratantes que desde su punto de vista se consideran sensibles.
Rendimiento global	Incluye widgets personalizados de calidad general, widgets de vista agregativa que incluyen todos los KQI de las cuentas.
Departamentos	Utiliza una imagen de fondo con el gráfico organizativo y coloca los widgets en los departamentos relevantes. Normalmente aquí resulta útil el grupo de componentes del servicio, dependiendo de qué departamento representa en la organización.
Regiones geográficas	Utiliza una imagen de fondo con una asignación geográfica y coloca widgets de grupos de partes contratantes en las ubicaciones relevantes.

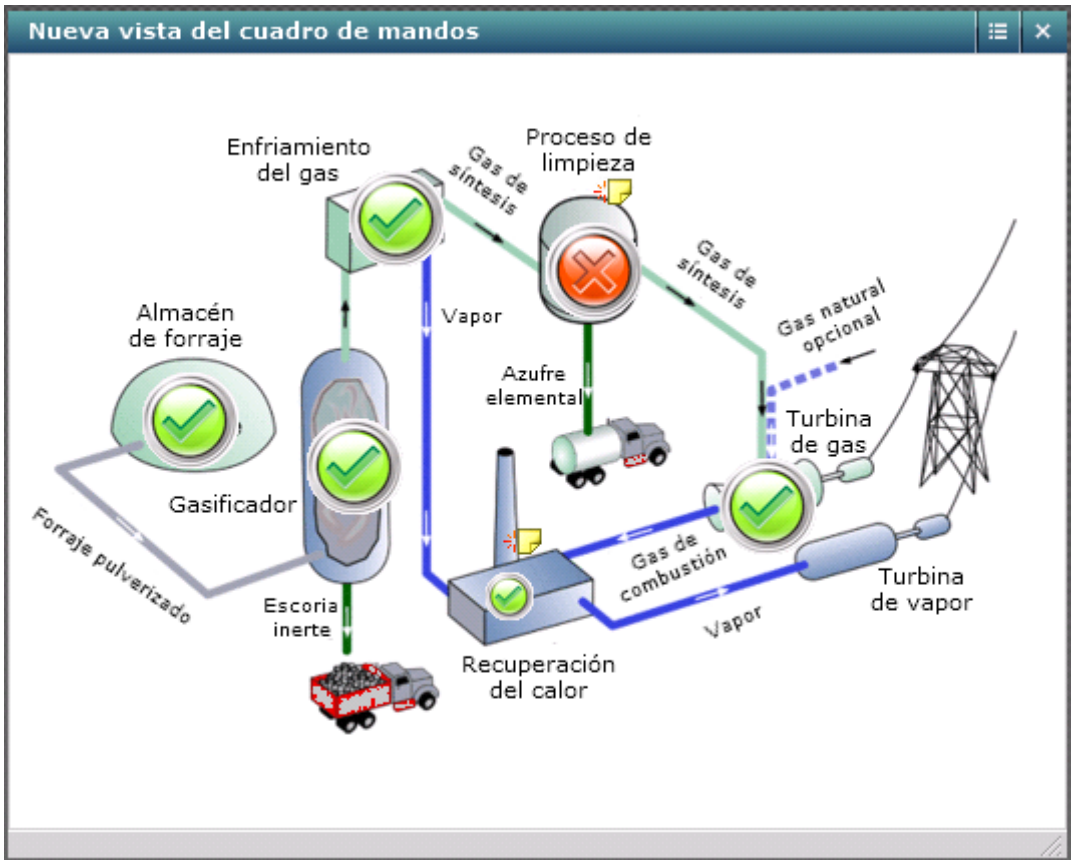
Rendimiento financiero	Incluye widgets con información agregativa en métricas financieras.
URL	Incluye páginas del portal corporativo, por ejemplo oportunidades comerciales para el equipo de ventas.

Informes recomendados para la página:

Informe	**Descripción**
Informe Desviación	Los diez peores contratos para un período concreto, proporciona información acerca de las áreas con peor rendimiento en términos de entrega de servicios.
Informe financiero para el mes actual	Resume el estado financiero agregado con el transcurso del tiempo.

Página de proceso:

Esta página deberá contener una vista del cuadro de mandos con un diagrama del proceso con widgets que representa cada cadena en el proceso, como en el ejemplo a continuación:



Páginas del gestor de contratos

Las páginas configuradas para un gestor de contratos muestran lo bien o no que se está proporcionando el servicio para cada una de las cuentas que está gestionando. También hay una vista del cuadro de mandos de los contratos relevantes para mantener informado al gestor del nivel de servicio actual proporcionado sobre las obligaciones de los contratos bajo su responsabilidad. Además, esta página muestra los informes disponibles para cada uno de los componentes de servicio que se incluyen en el contrato.

La página Descripción general incluye las vistas del cuadro de mandos siguientes:

Vista del cuadro de mandos	Descripción
Mis cuentas	Widgets de todos los contratos o las partes contratantes bajo responsabilidad del gestor de contratos.
Mis servicios	Componentes de servicio en todas las cuentas gestionadas por el gestor de contratos.
Rendimiento financiero	Widgets que incluyen información agregativa sobre métricas financieras para las cuentas gestionadas.
URL	Un portal como el sistema CRM.

Cree una página de cuenta para cada una de las cuentas gestionadas que proporcionan una vista de las obligaciones de cuenta específicas. Se recomienda combinar vistas de cálculo del estado actual, a parte del estado contractual, para que el gestor de cuentas pueda ser proactivo y actuar con rapidez. Por ejemplo:

Página de cuenta	Descripción
Obligaciones de cuenta	Incluye métricas contenidas en el contrato.
Rendimiento financiero	Widgets que incluyen información agregativa sobre métricas financieras para las cuentas gestionadas.

Páginas del Gestor de servicios

Páginas que se tienen que configurar para un gestor encargado de un servicio o dominio específico, que muestran una vista detallada de los objetivos de servicio en los diferentes contratos y los estados de sus objetivos. Además, dichas páginas incluyen informes que resaltan los parámetros de nivel de servicio esenciales que se miden.

Las vistas del cuadro de mandos incluidas en las páginas del Gestor de servicios son similares a las páginas del Gestor de cuentas, donde solamente se muestra la información y se agrega en un nivel de servicio en lugar de en el contrato o en el nivel de la parte contratante.

Páginas principales

Las páginas de cuadro de mandos se pueden asignar como páginas principales, por ejemplo para activar una puerta de enlace personalizable para interactuar con el sistema y proporcionar un acceso rápido a la información y a las acciones.

Adición de perfiles de alertas de nivel de servicio

El perfil de alertas permite definir las condiciones de envío de una notificación de alerta a un destinatario predeterminado a través de un dispositivo concreto.

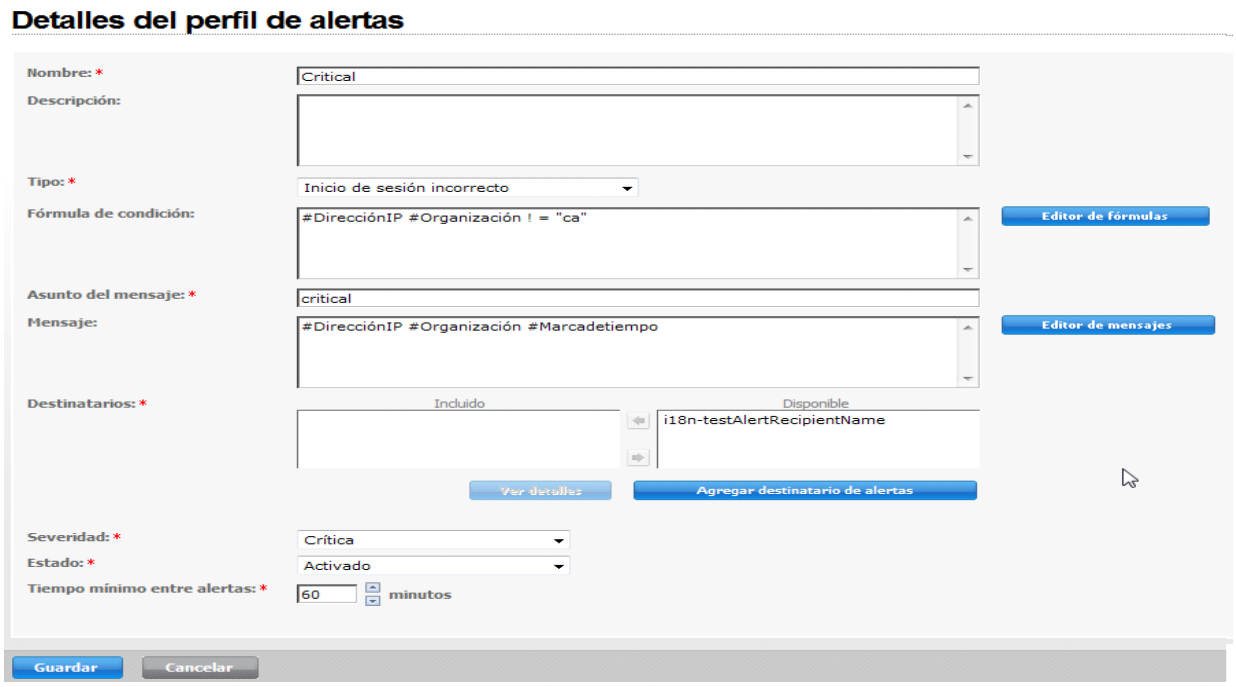
Antes de crear perfiles, vale la pena determinar qué condiciones son lo suficientemente importantes como para requerir notificación, y también quiénes deberán ser los destinatarios. Es importante definir perfiles que ayuden al gestor de contratos y al administrador del sistema a gestionar eficazmente los problemas y las incidencias de servicio. Todas las notificaciones deberán dirigir claramente a los recursos necesarios para abordar las incidencias más urgentes y prevenir infracciones.

A la hora de añadir alertas de nivel de servicio, hay una serie de campos que se pueden utilizar para determinar el estado de la métrica y para decidir si se deberá activar o no una condición de alerta. Como criterios de filtrado y condiciones se pueden utilizar cualquiera de los detalles estándar de métrica (como contrato, servicio, nombres de métricas, destinos, ranuras de tiempo, resultados del nivel de servicio, etc.). También se pueden emitir alertas sobre pronósticos de nivel de servicio, en cuyo caso se proporcionan algunos valores adicionales, como mejor o peor caso de desviación y niveles de servicio. Esta información es sumamente útil para saber si se puede recuperar o no el sistema de una infracción de nivel de servicio dentro de un período dado.

Las alertas basadas en el registro del sistema y las secciones generales de CA Business Service Insight son otras funciones sumamente útiles que permiten emitir una alerta en base a cualquier acción que tenga lugar dentro del sistema (esto se escribe en el registro del sistema). Dado que casi todas las acciones se registran, esto proporciona un amplísimo campo de actuación para los perfiles de las alertas. Los perfiles de las alertas del registro del sistema constan de lo siguiente:

- Notificación del adaptador que ejecuta la alerta para el experto en el origen de datos.
- Cualquier condición de "error" introducida en el registro. Se envía una alerta de correo electrónico al administrador del sistema.
- Errores personalizados creados por la lógica de negocios relevante para las condiciones específicas de los orígenes de datos. Pueden generar una alerta de correo electrónico que se envía al grupo responsable del servicio dentro de la organización y al gestor de contratos. Por ejemplo, si no se ha resuelto dentro de cierto marco temporal un incidente de prioridad alta, se envía una alerta al Help Desk para escalar la incidencia.
- Repetidos intentos de inicio de sesión no válidos. Se pueden enviar por correo electrónico al administrador del sistema para determinar si alguien está intentando entrar indebidamente en el sistema
- Nuevas entradas de traducción llegadas desde el adaptador (por ejemplo, si se encuentran entradas de recurso nuevas en un origen de datos que deban traducirse en el sistema para poder calcular el SLA correctamente).

A continuación se muestra la ventana Detalles del perfil de alertas con la configuración de un perfil de alertas personalizado que informa sobre un tipo de evento personalizado de los eventos de ticket abierto del Help Desk. Si coinciden los criterios, entonces en envía una alerta crítica al coordinador del Help Desk para garantizar que el ticket se gestione eficazmente. Esto podría ser útil en situaciones en las que se esté comprobando una aplicación en algún momento y se requiera un nivel especial de cuidado.



A continuación se incluye un ejemplo de implementación de una alerta de incumplimiento de nivel de servicio que se basa en si el SLA se puede cumplir todavía teniendo en cuenta el nivel de servicio actual y el tiempo restante en el período de seguimiento de negocio de la métrica.

Ejemplo: Alerta de incumplimiento de SLA reversible

Esta alerta se activa cuando se está incumpliendo un objetivo (métrica), pero puesto que el mejor escenario indica conformidad con el objetivo, el incumplimiento se puede evitar si se proporciona en adelante un nivel de servicio más apropiado.

Tipo: pronóstico de nivel de servicio

Fórmula de condición:

#Desviación>0 Y #Desviacióndelmejorcaso<=0 Y
#Niveldeserviciodelmejorcaso<>#Niveldeserviciodelpeorcaso Y
#Elementodeclúster = ""

Mensaje:

Tenga en cuenta que el objetivo "#Métrica" del contrato "#Contrato" se está incumpliendo actualmente, pero el incumplimiento es reversible.

Mientras que el destino del nivel de servicio sea #Destino, se predecirán los niveles de servicio siguientes:

- Nivel de servicio por ahora: #Niveldeservicio (Desviación: #Desviación%).
- Peor nivel de servicio de escenario: #Niveldeserviciodelpeorcaso (Desviación: #Desviacióndelpeorcaso%).
- Mejor nivel de servicio de escenario: #Niveldeserviciodelmejorcaso (Desviación: #Desviacióndelmejorcaso%).

Importación de perfiles de alertas administrativas

La opción Transferencia de contenido del menú Administración le permite importar o exportar entidades para fines administrativos a o desde la base de datos del sistema. Es útil importar algunos perfiles de alertas predeterminados. Estos perfiles de alertas proporcionan alertas al administrador del sistema como:

- **Notificación de estado del adaptador**: la alerta informa de si ha cambiado el estado de un adaptador.
- **Seguimiento de inicio de sesión incorrecto**: notificación de eventos de inicio de sesión erróneo a CA Business Service Insight.
- **Notificación del servicio de CA Business Service Insight**: notificación de inicialización de servicio interno.
- **Procesamiento de sanciones**: la alerta informa de los resultados de procesamiento de la sanción.
- **Procesamiento de nivel de servicio**: la alerta informa de los resultados del nivel de servicio.
- **Notificación de error del sistema**: la alerta informa de los mensajes de error del sistema.

Siga estos pasos:

1. Copie el archivo SupportAlertProfiles.xml desde los archivos de instalación en la carpeta Additional Files al servidor de aplicaciones en la carpeta Archivos de programa\CA\Insight\Export\Out. (Nota: Si no se le ha proporcionado este archivo, se puede poner en contacto con Soporte de CA para solicitarlo).
2. En el menú Administración, haga clic en Transferencia de contenido, Importar. En el cuadro de diálogo Importar, seleccione "Omitir y continuar" para la opción En colisión. Seleccione el archivo SupportAlertProfiles en la lista desplegable Archivo para importar.
3. Haga clic en Importar. Se importan los perfiles.

Capítulo 4: Instalación e implementación

Esta sección contiene los siguientes temas:

Introducción (en la página 224)
Preparativos (en la página 226)
Instalación (en la página 228)

Introducción

Una vez que la fase de configuración ha finalizado y los cálculos del sistema se han verificado correctamente, el sistema está preparado para implementarse en el entorno de producción. Este capítulo pretende cubrir todos los pasos que pueden ser necesarios durante esta fase. Estos pasos pueden variar de una instalación a otra; no obstante, se deberán comprobar los elementos siguientes para ver que se han implementado todos los preparativos para integrar el sistema en un entorno de producción en funcionamiento.

- El software y el hardware de producción están disponibles y listos para instalarse.
- Se ha configurado el servidor de correo para admitir el envío de correo externo/interno.
- Hay un agente de FTP (si es necesario para transferir archivos de datos).
- El suministro de orígenes de datos está disponible y funciona.

Hay dos fases dentro de la instalación. Durante la primera fase (la fase de instalación), el administrador del sistema deberá instalar e integrar los componentes de CA Business Service Insight en el entorno de producción. Durante la segunda fase, se verifica la funcionalidad del sistema y el sistema se deberá poner en estado controlado para garantizar que los procesos de sistema y las funcionalidades de la interfaz de usuario funcionen correctamente.

Durante el proceso de instalación puede ser necesario trabajar en los servidores de producción mediante herramientas de conexión remota que deberán estar instaladas en los servidores Web y de aplicaciones para poder realizar una instalación completa y segura.

Si es posible, la base de datos Oracle la deberá instalar un experto DBA de Oracle para garantizar que la configuración se realiza correctamente y que cumple con los requisitos corporativos aplicables. También se puede dejar que cree la base de datos el administrador del sistema mediante la utilidad de creación de datos incluida en la instalación de CA Business Service Insight. En este caso, el DBA deberá verificar la instalación para garantizar que haya finalizado correctamente. Si el sistema ya se ha configurado en un sistema de desarrollo y tiene que migrarse, entonces es necesario mover el contenido de la base de datos a esta nueva base de datos de producción.

A veces, una vez se ha transferido el sistema completo al nuevo entorno de producción, se aconseja eliminar todos los datos calculados y los datos sin procesar almacenados en el sistema y volver a cargar el sistema desde cero (manteniendo el catálogo de servicios, los recursos y las traducciones, por supuesto). Esto es una forma excelente de probar el funcionamiento del sistema entero dentro del entorno de producción.

Para resumir, esta fase incluye los pasos siguientes:

- Instalar y conectar el sistema.
- Cargar la base de datos del sistema configurada si es necesario.
- Integrar conexiones con el correo electrónico, Exchange, SMS, etc.
- Probar la funcionalidad y ajustar el rendimiento.
- Instalar, configurar y probar las conexiones remotas.
- Volver a iniciar el sistema listo para entrar en funcionamiento.
- Activar los adaptadores de datos para empezar a sondear los orígenes de datos y cargar los datos sin procesar.
- Encender el motor de CA Business Service Insight y permitir que empiecen los cálculos.
- Rellenar toda documentación necesaria, como, administración del sistema, base de datos y otros procedimientos de mantenimiento.
- Asegurarse de que los usuarios reciben la formación adecuada.

Preparativos

Para garantizar una instalación eficaz, es importante preparar por adelantado algunas cosas:

1. **La exportación de la base de datos al entorno de desarrollo**. Esta exportación de la base de datos se deberá realizar mediante scripts aprobados por CA que creen un archivo de extracción (volcado) para poder importarlo de vuelta al software en el sistema de producción.

   Nota: Es importante que la exportación la realice un usuario de la base de datos que disponga de privilegios suficientes, así como que se pueda acceder a él al importar a la base de datos. Para servir a este fin se puede usar la cuenta "**oblicore**", que siempre se crea en cada sistema, o también la cuenta "**sys**". No obstante, asegúrese de que la contraseña de "**sys**" esté disponible en la base de datos de destino para que se pueda llevar a cabo el proceso de importación.

2. **Scripts de base de datos**: si el DBA desea modificar los scripts de creación de la base de datos, entonces los deberá verificar antes un DBA de CA. Estos scripts se deberán verificar y quedar listos por adelantado para poder llevar a cabo una instalación fluida. Es habitual que diferentes organizaciones deseen transmitir algunos comentarios y realizar algunos cambios que el administrador de base de datos de CA deberá aprobar para garantizar que la configuración de la base de datos cumple con las políticas locales.

   - **Conectividad del servidor y accesibilidad**: tanto los servidores Web como los de aplicaciones deberán tener una conexión de cliente remota instalada que admita un acceso externo (si no se puede acceder físicamente en el momento de realizar la instalación, o en el futuro para incidencias de soporte). La posibilidad de acceder externamente es también importante para supervisar el sistema de forma continua durante un período complicado. Debido a otras implicaciones de seguridad adicionales, si se requiere acceso remoto externo esto podría exigir más tiempo de organización; por consiguiente, esta cuestión se deberá abordar por adelantado.
     Las conexiones remotas se pueden establecer mediante cualquiera de las herramientas siguientes:

   - PcAnyWhere
   - Máster y host proxy
   - Microsoft Terminal Server
   - Escritorio remoto
   - VNC

- Cualquier otra herramienta de conexión remota compatible con la organización.

Notas:

1. Microsoft Terminal Server se conecta al servidor simulando un inicio de sesión de servidor, a diferencia de PcAnyWhere, VNC, Proxy y otras herramientas.
2. Hay software de Oracle que no se puede instalar mediante Microsoft Terminal Server y que se debe instalar localmente.

3. **Accesibilidad del origen de datos**: el origen de datos deberá estar accesible para poder manipularlo manualmente y la conexión ODBC se deberá configurar para permitir que los adaptadores se conecten al origen de datos.
4. **Seguridad del servidor**: tanto en el servidor Web como en el de aplicaciones, se requiere un perfil de usuario con derechos de administrador local. Si la plataforma de base de datos es un sistema operativo Windows estándar, se requiere también un perfil de usuario con derechos de administrador local. Para plataformas de Unix, el administrador de la base de datos deberá tener los privilegios apropiados para crear la base de datos en el servidor.
5. **Acceso a Internet**: esto permite al **administrador del sistema** conectarse a Internet para conseguir las actualizaciones del sistema operativo o de aplicaciones que haga falta.
6. **Un equipo de escritorio de usuario estándar para probar la funcionalidad de aplicación**. Tenga en cuenta que la aplicación requiere usar algunos controles ActiveX que se tienen que instalar; esta función a menudo está previamente bloqueada debido a las políticas de seguridad de la organización.
7. **Programa de formación con el personal relevante**: formación introductoria que permitirá al personal seguir la formación de aceptación del usuario y empezar a trabajar con el sistema.

Instalación

El proceso de instalación se describe en detalle en la Guía de instalación e incluye las actividades siguientes:

1. **Instalación de la base de datos**:

   La instalación de base de datos es responsabilidad del experto en el origen de datos o del DBA de Oracle, y en situaciones especiales deberá realizarse bajo la dirección de un ingeniero de CA. La instalación de la base de datos incluye los pasos siguientes:

   - Instalación de Oracle en el servidor de la base de datos.
   - Instalación del parche de Oracle en el servidor de la base de datos (si es necesario; se deberán utilizar siempre las últimas versiones de servicio de soporte del software de base de datos de Oracle).
   - Instalación del cliente de Oracle en el servidor Web/de aplicaciones.
   - Instalación del parche de cliente de Oracle en el servidor Web/de aplicaciones (si es necesario; asegúrese siempre de que esta versión coincida con el servidor).

2. **Instalación de la aplicación CA Business Service Insight**:

   El administrador del sistema instala la aplicación. Si ya se ha realizado la instalación en el entorno de prueba (durante la fase de configuración) al llevar a cabo la integración y las pruebas del adaptador, entonces sólo se requiere un estado de inicialización o una importación de la base de datos. Después puede ser necesario instalar la aplicación en el entorno de producción.

   La instalación de la aplicación conlleva los pasos siguientes:

   - Creación de la base de datos de CA Business Service Insight.
   - Instalación del servidor de aplicaciones.
   - Instalación del servidor Web.
   - Instalación del adaptador.

Instale siempre la última versión del servicio de CA en los servidores de aplicaciones/Web. La instalación de la aplicación y las versiones del servicio incluye scripts SQL para garantizar que la base de datos se actualice a la estructura correcta para la versión. Estos scripts se almacenan todos en el directorio de instalación de la aplicación por si la base de datos tiene que actualizarse otra vez en otro momento. Por ejemplo, si es necesario importar la base de datos a partir de una copia de seguridad construida en una versión de servicio anterior. En este caso, deberá ejecutar el script de la última actualización de las versiones del servicio para garantizar que la estructura de la base de datos se corresponda con los componentes binarios instalados como parte de esa misma versión de servicio.

Si lo desea, también puede instalar herramientas de soporte adicionales, como un desarrollador de PLSQL o una utilidad de SQL alternativa para ayudar en las tareas de manipulación de datos de niveles inferiores.

Cómo importar o exportar entre entornos (experto en el origen de datos)

Esta situación conlleva la migración de adaptadores configurados y probados en un entorno de desarrollo/prueba a un entorno de producción o en funcionamiento. Se supone que el entorno de producción se ha instalado ya con una instalación de CA Business Service Insight estándar y que la base de datos está presente.

El proceso es el siguiente:

Para exportar la base de datos e importarla en el nuevo entorno

1. Detenga todo el trabajo en el sistema de desarrollo y elija un punto lógico en el sistema para realizar una exportación que se pueda utilizar para el sistema de producción. Cierre todos los componentes de servicio de CA Business Service Insight y los componentes COM+ de CA Business Service Insight. Exporte mediante los scripts de exportación del sistema estándar de CA Business Service Insight. (Si fuera necesario, póngase en contacto con Soporte de CA).
2. Tome la copia del extracto de la base de datos y colóquela en el servidor de producción listo para importar. Tenga en cuenta que las versiones de Oracle de las bases de datos de producción y de desarrollo deben coincidir. Importe la base de datos mediante los scripts de importación del sistema estándar (proporcionados con los scripts de exportación).
3. Una vez finalizado el proceso, compruebe si hay incidencias de importación. Si no hay ninguna, asegúrese de ejecutar los scripts SQL de las últimas versiones de servicio (si es necesario).
4. Ejecute el acceso directo en miniatura del asistente para configurar la base de datos de acuerdo con los valores de configuración del nuevo sistema de producción.
5. Inicie todos los componentes de servicio de CA Business Service Insight y conéctese al sistema de producción para confirmar que el sistema se ha importado correctamente.

Para migrar los adaptadores

1. Instale el adaptador mediante la utilidad AdapterManager con los valores de configuración similares al adaptador que está importando (asegúrese de que la denominación sea exactamente la misma para que el paso siguiente funcione correctamente).
2. Copie el archivo de configuración del adaptador desde el sistema de desarrollo a la carpeta del nuevo adaptador en el sistema de producción. Sobrescriba la configuración predeterminada proporcionada. (Asegúrese de que se sobrescribe el archivo. De lo contrario, si la denominación no es la misma se producirán problemas durante la ejecución).
3. Actualice el archivo de configuración del adaptador. La comunicación con el servidor de CA Business Service Insight y con el origen de datos se deberá actualizar para ajustarse al nuevo entorno. La sección OblicoreInterface se deberá actualizar con el puerto de adaptador correcto. La sección DataSourceInterface se deberá actualizar con el valor ConnectionString correcto o el patrón de nombre de archivo o la ruta necesarios.
4. Asegúrese de que todos los nombres DSN (nombre de origen de datos) de ODBC están configurados y funcionan en el equipo nuevo.
5. Pruebe la conectividad del adaptador.
6. Pruebe la ejecución del adaptador.
7. **Traducciones**: se deberán activar cuando haya scripts de traducción. Compruebe que están sincronizadas con el adaptador y que las traducciones se realizan según lo previsto. Cuando se implemente una traducción manual y no haya finalizado antes, se deberán realizar más traducciones en esta etapa.
8. **Programación del adaptador**: programe el adaptador para que se ejecute según lo previsto. Si el adaptador se ha definido como una aplicación en un modo de una única ejecución, se puede programar mediante el programador de Windows. Esto se puede consultar en Panel de control > Tareas programadas. Consulte Modos de ejecución del adaptador (en la página 100) para obtener más información.

Integración: configuración del servidor de correo (administrador del sistema)

Para activar las funciones de notificación del sistema, es necesario saber qué servidor de correo y qué buzón de correo se utiliza para enviar mensajes de correo electrónico de CA Business Service Insight. Este servidor de correo debe permitir retransmitir correo a través de él, ya que se enviará como SMTP desde los servidores de CA Business Service Insight a este servidor de correo mediante la cuenta especificada. Después de completar la configuración del servidor de correo, puede utilizar funciones de correo electrónico en las funciones alertas, informes de detección de servicios y aprobación de contrato en CA Business Service Insight

Haga clic en el menú de Administrador y seleccione Configuración del sitio, Alertas. Configure las definiciones de correo electrónico en la sección de valores de configuración de alerta, incluido el servidor de correo electrónico, la dirección de envío y el nombre del remitente, junto con información del proveedor de SMS para utilizar puertas de enlace de SMS.

Parte de las pruebas de integración implican verificar que el servidor de aplicaciones puede contactar con el servidor de correo de la organización y utilizarlo para enviar alertas de CA Business Service Insight.

Para probar la conectividad entre esos servidores:

Introduzca como sigue una solicitud de línea de comandos en el servidor de aplicaciones:

ORGANIZATION-MAIL es generalmente el servidor de correo definido en su cliente de correo electrónico. Póngase en contacto con el administrador del sistema para obtener este valor si no está seguro.

Nota: El nombre ORGANIZATION-MAIL es un marcador de posición para el servidor de correo de su organización. Deberá reemplazar este marcador de posición con su nombre o la dirección del servidor de correo de su organización.

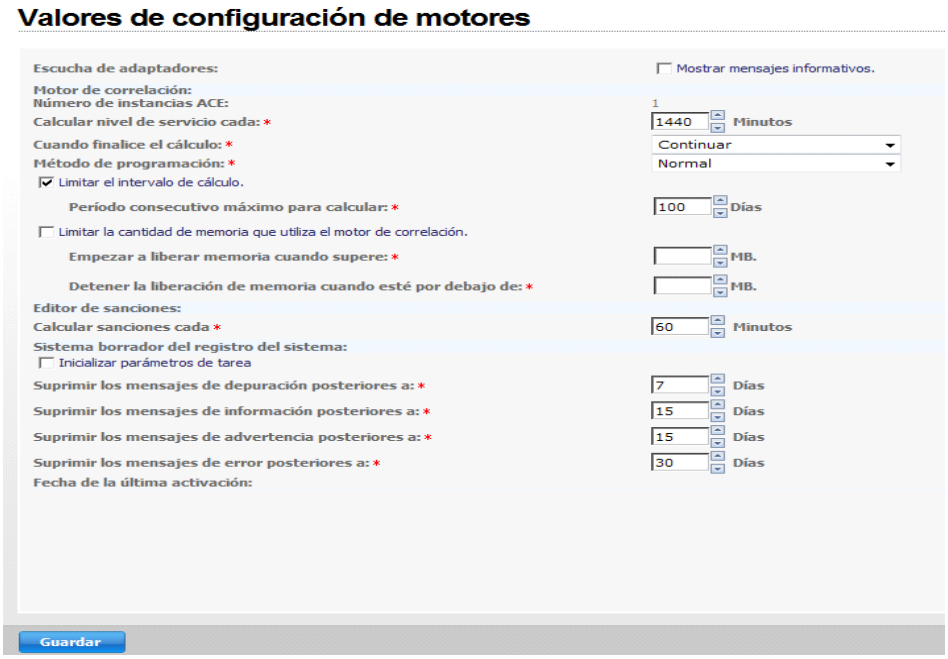
Para comprobar el servidor de correo del cliente Outlook:

1. En Microsoft Outlook, vaya a Herramientas > Cuentas de correo electrónico, y seleccione Ver o cambiar cuentas de correo electrónico existentes.
2. Haga clic en Cambiar.
3. Copie el servidor de Microsoft Exchange Server que corresponda al servidor de correo de la organización**.**

   Si hay respuesta del servidor con este comando, entonces la conexión se ha realizado con éxito. El resultado deberá ser similar al siguiente:

   Si recibe cualquier otro mensaje, significa que no hay conectividad entre los dos servidores. En este caso, consulte con el administrador del sistema.

Establecimiento de las preferencias del sistema (administrador del sistema)



El proceso de establecer las preferencias del sistema incluye aplicar los valores relevantes a las variables del sistema. Desde el menú Administración, haga clic en Configuración del sitio, Motores, para abrir el cuadro de diálogo de preferencias del motor.

Para obtener más detalles sobre las diversas recomendaciones para los parámetros, consulte la Guía del administrador.

Opciones de seguridad (administrador del sistema)

Las opciones de seguridad incluyen la creación de usuarios, grupos de usuarios y roles. De forma predeterminada, todos los usuarios se asocian con la organización especificada durante el proceso de instalación de la aplicación. No obstante, si necesario se pueden también crear organizaciones adicionales.

La mayor parte de las definiciones necesarias se habrán completado ya durante la fase de configuración, por lo que normalmente solamente será necesario hacer algunos ajustes para definir los valores de configuración adicionales que se puedan haber identificado desde ese momento.

Para obtener más información acerca de la configuración de la seguridad, consulte la ayuda en línea o la Guía del administrador.

Especificación de valores de configuración para sincronización de SSA

Cuando se utiliza CA Spectrum Service Assurance (SSA) para detectar servicios para CA Business Service Insight, se pueden establecer valores de configuración para permitir una sincronización automatizada. Las funciones de automatización le permiten mantener la lista de servicios y los datos de servicio actualizados.

Nota: Debe tener acceso a la API en reposo en SSA para editar estos valores de configuración.

Siga estos pasos:

1. En el menú Administrador, haga clic en Configuración del sitio, Configuración de SSA.

   Se abre el cuadro de diálogo Configuración de SSA.

2. Introduzca la información siguiente en el área Autenticación del usuario SSA:

   URL del servidor

   Indica la URL para el servidor de SSA de destino.

   ID de usuario

   Especifica el ID del usuario para el servidor de SSA.

   Contraseña

   Especifica la contraseña para el ID de usuario del servidor de SSA.

3. Introduzca la información siguiente en el área Opciones de la sincronización:

   Sincronizar automáticamente

   Especifica que desea que la sincronización se produzca automáticamente según la frecuencia de sincronización (consulte el siguiente párrafo).

   Establecer frecuencia de sincronización

   Especifica la frecuencia con que se buscarán nuevos servicios. Se puede indicar un valor en horas o días.

   Sincronización manual

   Permite llevar a cabo la sincronización manual del cuadro de diálogo.

4. Introduzca la información siguiente en el área Opciones predeterminadas de datos:

   Servicios predeterminados para:

Le permite establecer el valor predeterminado en gestionado o no gestionado para los servicios descubiertos.

Activar el cálculo de métricas de comparación

Le permite establecer la acción predeterminada para establecer métricas de comparación para los servicios de SSA.

5. Haga clic en Aceptar.

   Sus valores de configuración de SSA se activan.

Valores de configuración del uso compartido global

Los valores de configuración de uso compartido global sirven para establecer un vínculo con Cloud Commons. Al establecer el vínculo, puede compartir y recuperar datos de comparación de iguales que se mostrarán en toda la interfaz de usuario cada vez que se usen datos de comparación.

Los datos sobre su compañía se utilizan en algunos cálculos estadísticos en Cloud Commons, si bien se mantienen confidenciales. Haga clic en la política de privacidad de CA en el área Selección de la opción de uso compartido para obtener más información.

Siga estos pasos:

1. En el menú de administración, haga clic en Configuración del sitio, Uso compartido global.

   Se abre el cuadro de diálogo Valores de configuración del uso compartido global.

2. Introduzca la información siguiente en el área Información demográfica para la comparación de iguales:

   Industria

   Especifica su categoría de sector. Seleccione un nombre de sector en el menú desplegable.

   Tamaño de la compañía

   Especifica el tamaño de su compañía. Seleccione un intervalo de tamaño en el menú desplegable.

Ingresos

Especifica los ingresos anuales de su compañía.

País

Especifica el país de la oficina principal de su compañía.

Código ZIP/Postal

Especifica el código postal de la oficina principal de su compañía.

3. Introduzca la información siguiente en el área Registro de claves de la API:

Nombre de la organización

Especifica el nombre de su organización.

Dirección de correo del administrador

Especifica la dirección de correo electrónico del administrador para tareas de uso compartido global.

Verificar correo electrónico del administrador

Comprueba la dirección de correo electrónico introducida en el campo anterior.

4. Vaya al área Selección de la opción de uso compartido, revise el acuerdo para el uso compartido global y seleccione una de las opciones:

Uso compartido global: activado

Permite usar la opción de uso compartido global para sus servicios. Esta opción le permite marcar servicios individuales para uso compartido mediante el botón Acción del servicio en la página Detección y gestión de servicios.

Uso compartido global: desactivado

Desactiva el uso compartido global para sus servicios. Cuando se desactiva el uso compartido global, no tiene la opción de compartir servicios individuales en la página Detección y gestión de servicios.

5. Haga clic en Registrar para registrarse por primera vez, o en Guardar para actualizar los cambios.

Su configuración de uso compartido se activa.

Capítulo 5: Gestión y detección del servicio

Esta sección contiene los siguientes temas:

Detección del servicio (en la página 239)

Detección del servicio

Cómo funciona Detección y gestión de servicios

El gestor de la detección de servicios/gestor de carpetas de plantillas realiza tareas administrativas en los servicios detectados mediante funciones de la páginas Detección y gestión de servicios.

Tras completar las tareas preliminares para detectar sus servicios, se utilizan las funciones en la página Detección y gestión de servicios para lo siguiente:

- Entender las categorías de servicios (en la página 241)
- Establecer la visibilidad y el ámbito de sus servicios (en la página 244)
- Categorizar un servicio seleccionado (en la página 249)
- Trabajar con opciones de atributos y metadatos (en la página 250)
- Gestionar los servicios y habilitarlos para compartir (en la página 254)
- Detectar nuevos servicios y sincronizarlos con CA Spectrum Service Assurance (SSA) (en la página 258)

Para empezar, haga clic en Diseño, Detección del servicio, en el menú principal de CA Business Service Insight.

Comprensión de las categorías de servicios

Al trabajar con sus servicios, podrá categorizarlos para habilitar las funciones de comparación y análisis disponibles en CA Business Service Insight. La categorización de un servicio empieza cuando se detectan servicios mediante CA Service Spectrum Assurance (SSA), y se va afinando a medida que se trabaja con las funciones de la página Detección y gestión de servicios.

Puede asociar una categoría de servicio a un servicio específico. La forma en la que se trabaja con las opciones de categorización afecta a cómo se pueden ver, filtrar y comparar los servicios.

Se proporcionan las categorías de servicio siguientes, que aparecen de forma predeterminada en el panel de visualización de filtros.

- Copia de seguridad y recuperación
- Inteligencia empresarial
- Cálculo
- Gestión de la relaciones de clientes
- base de datos
- Comercio electrónico
- Correo electrónico
- Programación de recursos empresariales
- Gestión de TI
- Plataforma
- Gestión de carteras y proyectos

Consulte la Hoja de cálculo de categorías de servicios (en la página 241) para obtener más información acerca de cómo se define cada categoría.

Hoja de cálculo de categorías de servicio

Al categorizar sus servicios, la información en esta hoja de cálculo puede ayudarle a entender el significado de las diferentes categorías. Esta misma información se muestra en el cuadro de diálogo Categorización del servicio, en una ventana emergente que aparece al pasar el ratón por encima del nombre de una categoría.

Nombre de la categoría	Los servicios en esta categoría deberán tener las capacidades siguientes.
Copia de seguridad y recuperación	■ Rendimiento de almacenamiento ■ Escala automática ■ Soporte regional ■ Protocolo múltiple o soporte de sistema de archivos ■ Opción de almacenamiento sin conexión ■ Copia de seguridad en nube
Inteligencia empresarial	■ Almacenamiento de datos ■ Informes de ventas ■ Informes de datos ■ Informes ad hoc ■ Integración de SalesForce ■ Soporte de aplicaciones de empresa
Cálculo	■ Creación de instancias ■ Escalado de instancias ■ Instancia persistente ■ Integración del servicio de almacenamiento ■ Instancia de equilibrio de carga ■ Instancia de control ■ Integración del servicio de base de datos ■ Capacidad de cálculo ■ Capacidad de red ■ Rendimiento de almacenamiento

Gestión de la relaciones de clientes	■ Gestión de oportunidades ■ Gestión de servicios ■ Gestión de contactos ■ Integración de correo electrónico ■ Analíticas ■ Previsión ■ Seguimiento de oportunidades ■ Acceso móvil ■ Almacenamiento de documentos ■ Seguimiento de actividades ■ Automatización o flujo de trabajo ■ Gestión de campañas
Base de datos	■ Rendimiento de almacenamiento ■ Relacional ■ No relacional ■ Alojamiento en Amazon ■ Alojamiento en Rackspace ■ Creación o gestión de instancias de nube
Comercio electrónico	■ Procesamiento de pedidos ■ Gestión de inventarios ■ Cumplimiento ■ Servicios de pago ■ Generación de informes ■ Gestión de catálogos ■ Gestión de clientes ■ Búsqueda de catálogos ■ Integraciones

Correo electrónico	■ Envío y recepción de correo electrónico ■ Libro de direcciones ■ Calendario ■ Notes ■ Directorios ■ Integración de IM ■ Soporte Mobile Push
Programación de recursos empresariales	■ Gestión financiera ■ Programación financiera ■ Gestión de inventarios y cadena de suministro ■ Transporte y cumplimiento ■ Gestión de capital humano ■ Análisis financieros e informes
Gestión de IT	■ Control y emisión de alertas ■ Equilibrio de carga y escalado ■ Gestión de servidores físicos ■ Basado en ITIL ■ Automatización o generación de scripts ■ Implementación de nubes ■ Gestión de activos ■ CMDB ■ Gestión de costes

Plataforma	■ Entorno de desarrollo ■ Gestión de aplicaciones ■ Java ■ Ruby on Rails ■ .Net ■ Almacén de datos SQL ■ Almacenamiento persistente ■ Gestión de capacidades ■ Control y emisión de alertas ■ Rendimiento de aplicaciones
Gestión de carteras y proyectos	■ Gestión financiera ■ Gestión de tareas ■ Hitos ■ Gestión del tiempo ■ Almacenamiento de archivos ■ Tableros de mensajes ■ Gestión de proyectos ■ Gestión de la cartera ■ Gestión de recursos ■ Gestión de la demanda

Establecer la visibilidad y el ámbito de sus servicios

Cuando se abre por primera vez la página Detección y gestión de servicios, se muestran todos los servicios detectados. Para refinar el trabajo en los servicios y acceder a funciones granulares de detección del servicio, utilice las siguientes funciones en la ficha Gestión de los datos y del uso compartido de los datos:

Establecimiento de vistas de servicios (en la página 245)

Establecimiento de opciones de visualización de columnas (en la página 246)

Selección de una acción para un servicio seleccionado (en la página 248)

Establecimiento de vistas de servicios

Puede establecer vistas de servicios para trabajar con un subconjunto de todos sus servicios. Por ejemplo, se pueden mostrar solamente los servicios que comparten datos de comparación en Cloud Commons al seleccionar la vista de Datos de comparación de uso compartido de servicios.

Siga estos pasos:

1. Haga clic en Detección del servicio en el menú Diseño.

   Se abre la página Detección y gestión de servicios.

2. Haga clic en la ficha Gestión de los datos y del uso compartido de los datos.

3. Haga clic en el menú desplegable Ver y seleccione mostrar sus servicios en una de las categorías siguientes:

   Todos los servicios

   Muestra todos los servicios detectados en su sistema. Esta lista incluye servicios detectados desde CA Spectrum Service Assurance (SSA), servicios introducidos manualmente, servicios que comparten datos de comparación y servicios con Comparación del servicio activada.

   Servicios de SSA

   Muestra solamente los servicios detectados desde CA Spectrum Service Assurance (SSA). Esta lista se establece después de instalar y configurar SSA y establecer conectores en su entorno.

   Servicios especificados manualmente

   Muestra solamente los servicios especificados manualmente con el botón Agregar servicios.

   Servicios que comparten SMI

   Solamente se muestran los servicios que comparten datos de comparación de SMI. Póngase en contacto con su administrador para hablar de establecer valores de configuración de uso compartido globales para su cuenta de usuario.

   Servicios con la comparación de servicios activada

   Muestra solamente los servicios establecidos para una comparación de servicios activa. La comparación de servicios se activa y se desactiva mediante el menú desplegable en la columna Comparación del servicio.

   Servicios gestionados

   Solamente se muestran los servicios establecidos en el estado "gestionado".

Servicios no gestionados

Solamente se muestran los servicios establecidos en el estado "no gestionado".

La pantalla se actualiza para mostrar los servicios en la vista seleccionada.

Establecimiento de opciones de visualización de columnas

Puede personalizar las columnas que aparecen en la tabla de servicios en la página Detección y gestión de servicios. Por ejemplo, si todos los servicios se gestionan de forma activa, puede eliminar la columna Gestionar, ya que no es necesaria para su trabajo.

Siga estos pasos:

1. Haga clic en Detección del servicio en el menú Diseño.

   Se abre la página Detección y gestión de servicios.

2. Haga clic en el botón Personalizar.

   Se abre el cuadro de diálogo Configuración de visualización de la tabla.

3. Mueva el nombre de la columna que desee incluir en la visualización a la lista "Columnas visibles".

   Las siguientes columnas se encuentran disponibles:

   Servicios

   Especifica el nombre del servicio.

   Categoría

   Especifica la categoría de servicio que ha asignado al servicio.

Fuente

Especifica el origen del servicio. Por ejemplo, los servicios detectados mediante CA Spectrum Service Assurance (SSA) aparecen marcados como "SSA".

Métricas de comparación del servicio

Indica si el servicio se establece para la comparación del servicio o no. Puede trabajar con estos valores de configuración en la lista desplegable Acción.

gestión

Indica si el servicio está gestionado o no. Un servicio no gestionado es uno que se ha detectado, pero para el cual se han deshabilitado la métrica y el uso compartido. Puede establecer un servicio en el estado gestionado en la lista desplegable Acción.

Nombre del servicio compartido

Especifica el nombre para los datos de comparación del servicio cuando se comparte en Cloud Commons.

Uso compartido de datos

Indica si el uso compartido de la comparación del servicio está activado para el servicio o no. Puede activar o desactivar el uso compartido en la lista desplegable Acción.

Necesita aprobación

Indica si el nombre de servicio se ha enviado a Cloud Consortium, pero aún espera su aprobación.

Fecha de detección

Muestra la fecha de detección del informe.

4. Haga clic en Guardar para guardar sus valores de configuración y actualizar la visualización de la tabla.

Selección de una acción para un servicio seleccionado

Utilice las opciones en la lista desplegable de Acción del servicio para trabajar con un servicio seleccionado.

Siga estos pasos:

1. Haga clic en Detección del servicio en el menú Diseño.

   Se abre la página Detección y gestión de servicios.

2. Haga clic en la ficha Gestión de los datos y del uso compartido de los datos.

3. Haga clic en el botón Acción del servicio.

   En el menú desplegable aparecen las acciones siguientes:

   Categorizar/Editar

   Categorice un servicio seleccionado (en la página 249) para habilitar las funciones de filtrado y comparación. Después de categorizar un servicio, esta selección de menú cambia para **Editar**.

   Activar las métricas de comparación del servicio

   Habilita los cálculos de métricas en los servicios seleccionados.

   Desactivar las métricas de comparación del servicio

   Deshabilita los cálculos de métricas en los servicios seleccionados.

   Activar el uso compartido de servicios

   Habilitar el uso compartido de datos de comparación para usarlos en Cloud Commons.

   Desactivar el uso compartido de servicios

   Deshabilita el uso compartido de datos de comparación para Cloud Commons.

   Configurar el servicio para el uso compartido

   Seleccione esta opción para trabajar con funciones para compartir datos en Cloud Commons.

   gestión

   Seleccione esta opción para establecer el servicio en estado gestionado.

   Sin gestionar

   Seleccione esta opción para establecer el servicio en estado no gestionado. Cuando un servicio no se gestiona, no se recopilan datos de comparación para él, y el nombre aparece en rojo en la lista Servicios.

Volver a gestionar

Seleccione esta opción para restablecer el estado a gestionado para el servicio.

Suprimir servicio

Seleccione esta opción para suprimir el servicio de CA Business Service Insight.

Se abre un nuevo cuadro de diálogo o se actualizan las tablas de servicios según su selección.

Categorización de un servicio seleccionado

Los servicios se categorizan para activar las funciones en CA Business Service Insight que permiten recopilar datos y analizar el rendimiento del servicio. Después de categorizar un servicio, se puede aplicar un filtro por esa categoría en la página Descripción general del servicio, y compararlo con otros servicios categorizados igual.

Siga los pasos siguientes:

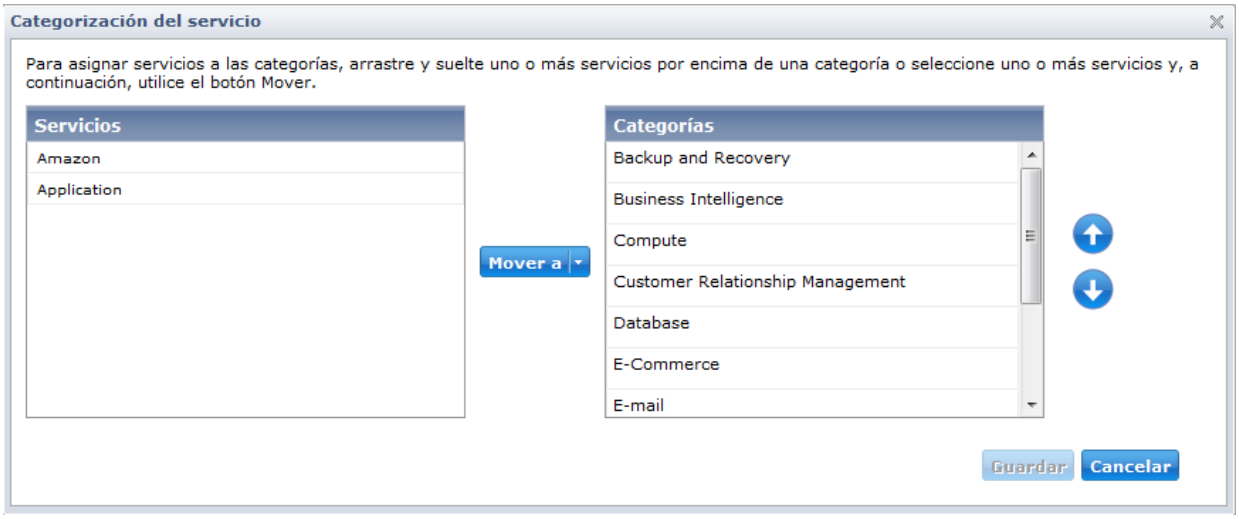
1. Haga clic en Detección del servicio en el menú Diseño.

   Se abre la página Detección y gestión de servicios.

2. Seleccione la casilla de verificación junto a uno o varios servicios que desee categorizar y, a continuación, haga clic en el botón Acción del servicio y luego seleccione Categorizar.

   Se abre el cuadro de diálogo Categorización del servicio.

   Los servicios seleccionados se muestran en el panel de la izquierda. Las categorías de servicios disponibles se muestran en el panel de la derecha.



3. Seleccione el servicio y arrástrelo a la categoría, o seleccione uno o varios nombres de servicio y haga clic en el botón Mover para seleccionar una categoría.
4. Haga clic en Guardar para guardar sus selecciones.

Nota: Utilice el icono de flechas hacia arriba o hacia abajo para cambiar el orden de los nombres de categoría.

Trabajar con opciones de atributos y metadatos

Trabaje con las opciones de la ficha Gestión de atributos y metadatos para crear y gestionar nombres de atributos. Los nombres de los atributos aparecen en el panel de filtros de servicios en la página Descripción general del servicio. Estas opciones le permiten crear filtros de vista personalizados. De forma predeterminada, se proporcionan funciones de categorías de servicios basadas en el marco SMI.

Creación de un nombre de atributo (en la página 251)

Asociación de un servicio a un atributo (en la página 252)

Gestión de atributos y valores asociados (en la página 253)

Creación de un nombre de atributo

Se puede crear un nuevo de atributo nombre y a continuación especificar una acción para él.

Siga estos pasos:

1. Haga clic en Detección del servicio en el menú Diseño.

   Se abre la página Detección y gestión de servicios.

2. Haga clic en la ficha Gestión de atributos y metadatos y, a continuación, haga clic en el botón Atributos y valores.

   Se abre el cuadro de diálogo Agregar/gestionar atributos La tabla muestra la información siguiente:

   Nombre de atributo

   Especifica el nombre del atributo que aparece en la lista de filtro en la página Descripción general del servicio.

   Editar

   Le permite editar los valores de configuración del atributo.

   Suprimir

   Suprime el atributo.

   Estado

   Muestra si el atributo está activo o inactivo.

   Valor

   Indica los valores asociados al atributo.

3. Haga clic en el botón Agregar atributo.

   Se abre el cuadro de diálogo Agregar atributo/valores. Puede trabajar con las opciones siguientes:

   Nombre de atributo

   Especifica el nombre del nuevo atributo.

   Asignar valores

   Le permite introducir valores específicos relevantes para la categoría del atributo. Por ejemplo, si crea un nombre de atributo "Gestores", puede crear nombres de gestor para los valores asociados.

4. Haga clic en Guardar.

   El cuadro de diálogo se cierra y el nuevo nombre del atributo se muestra en la columna Nombre de atributo.

Asociación de un servicio a un atributo

Tras crear nombres de atributo, se pueden asociar servicios a los atributos.

Siga estos pasos:

1. Haga clic en Detección del servicio en el menú Diseño.

   Se abre la página Detección y gestión de servicios.

2. Haga clic en la ficha Gestión de atributos y metadatos.
3. Seleccione los servicios con los que desee trabajar, haga clic en el botón Acción del servicio y, a continuación, seleccione Asignar atributos.

   Se abre el cuadro de diálogo Asociar servicios a atributos.

   Los servicios seleccionados aparecen en el panel de la izquierda, y sus atributos aparecen en el panel de la derecha.

4. Seleccione y arrastre el servicio al valor del atributo.

   Nota: Expanda el árbol del atributo para ver las categorías tras él.

5. Haga clic en Guardar para guardar sus selecciones.

   La columna en la tabla se actualiza para mostrar la asociación entre el servicio y el atributo.

Gestión de atributos y valores asociados

Tras crear nombres de atributos, podrá trabajar con opciones para gestionar atributos desde la lista desplegable Acción en el cuadro de diálogo Agregar/gestionar atributos.

Siga estos pasos:

1. Haga clic en Detección del servicio en el menú Diseño.

   Se abre la página Detección y gestión de servicios.

2. Haga clic en la ficha Gestión de atributos y metadatos y, a continuación, haga clic en el botón Atributos y valores.

   Se abre el cuadro de diálogo Agregar/gestionar atributos y valores.

3. Trabaje con las siguientes opciones en cuadro de diálogo:

   Editar

   Abre el cuadro de diálogo Editar atributos/valores, donde podrá cambiar el nombre y los valores asociados.

   Suprimir

   Suprime el atributo en la fila seleccionada.

   Establecer estado en activo

   Activa el atributo para que se muestre en la página Descripción general del servicio.

   Establecer estado en inactivo

   Desactiva el atributo para que no se muestre en la página Descripción general del servicio.

4. Haga clic en Cerrar para guardar los cambios y volver a la página Detección del servicio.

Gestión de los servicios

Puede controlar la visualización de los servicios en la página Descripción general del servicio, al activar o desactivar funciones para un servicio específico. Por ejemplo, puede elegir si desea recopilar y mostrar los datos de comparación interna, pero desactivar la visualización de iguales y los datos de comparación de categorías. Puede "no gestionar" un servicio para suspender la recopilación de todos los datos de comparación, etc.

Servicios gestionados/no gestionados

Al detectar un servicio, se puede elegir si desea establecer el estado gestionado. Se habilita un servicio gestionado para otras acciones (como la detección de métricas de comparación o para su uso compartido en Cloud Commons). Este primer paso es obligatorio antes de que pueda trabajar con otras funciones de comparación. El estado de gestión de servicios aparece reflejado en la columna Gestionar de la lista de servicios.

Los servicios con el estado Gestionar establecido en "desactivado" no aparece en la página Descripción general de servicios.

Valores de configuración de métrica de comparación del servicio

Puede activar o desactivar la opción para recopilar los datos de métricas de comparación. Esta opción es una acción que se puede aplicar a uno o varios servicios mediante el menú Acción en la ficha Gestión de los datos y del uso compartido de los datos de la página Detección y gestión de servicios. El estado de esta función aparece en la columna Métricas de comparación del servicio de la lista Servicios, marcado como "activado" o "desactivado".

Los servicios con el estado Métricas de comparación del servicio establecidos en "activado" muestran los resultados métricas de comparación en la página Descripción general de servicios, y en otras páginas que muestran resultados de métricas.

Uso compartido global

Puede activar el uso compartido global para todos los servicios en el cuadro de diálogo Uso compartido global a los que obtiene acceso al hacer clic en Configuración del sitio, Uso compartido global en el menú Administración. Uso compartido global debe establecerse en "activado" para habilitar Participación de Datos de un servicio específico el uso compartido de datos de un servicio. Active los valores de configuración de uso compartido global si planea compartir extensamente sus datos de servicio en Cloud Commons. Si debe limitar el uso compartido de datos, debe utilizar la configuración global para desconectar esta función para todos los servicios. Para trabajar con las funciones Uso compartido global, debe registrarse en primer lugar en Cloud Commons con las opciones en el cuadro de diálogo Uso compartido global.

El estado de uso compartido global para un servicio aparece en la columna Uso compartido de datos de la lista Servicios. "Configuración necesaria" aparece en esta columna para los servicios que no se han establecido todavía para el uso compartido global.

Uso compartido de datos

Cuando se activa el uso compartido global, puede activar el uso compartido de datos para un servicio (o un grupo de servicios seleccionados). Este proceso incluye la conexión a Cloud Commons para compartir los datos de comparación. Si no se encuentra una coincidencia para la categoría o el servicio seleccionados, puede agregar el servicio a Cloud Commons.

El estado de uso compartido de datos para un servicio aparece en la columna Uso compartido de datos en la lista de servicios, marcada como "activado" o "desactivado".

Los servicios cuyo estado Uso compartido de datos esté establecido en "activado" muestran datos de comparación de categorías e iguales en la página Descripción general de servicios, y en otras páginas en las que aparecen los datos de métrica.

Uso compartido de datos de comparación en Cloud Commons

Se puede configurar un servicio local para que sus datos de comparación se compartan en Cloud Commons. Cuando elija esta opción, podrá empezar a acceder a datos de comparación de categorías e iguales en Cloud Commons, así como podrá agregar información que afectará a las puntuaciones de comparación a la cual podrán acceder otros usuarios.

Nota: Debe activar el uso compartido global antes de compartir los datos de comparación en Cloud Commons. En el menú Administración, haga clic en Configuración del sitio, Uso compartido global para abrir el cuadro de diálogo Valores de configuración del uso compartido global (en la página 236).

Siga estos pasos:

1. Haga clic en Detección del servicio en el menú Diseño.

   Se abre la página Detección y gestión de servicios.

2. Haga clic en la ficha Gestión de los datos y del uso compartido de los datos.

3. Seleccione el servicio para el cual desee compartir datos y, a continuación, haga clic en el botón Acción del servicio y seleccione Configurar el servicio para el uso compartido.

   Se abre una nueva ventana con las siguientes opciones para compartir opciones para el servicio seleccionado:

   Coincidencia encontrada en Cloud Commons

   Si se encuentra una coincidencia en Cloud Commons para este servicio, se muestra una tabla de resumen con el nombre del servicio, la definición, las categorías y las capacidades. Haga clic en el botón Compartir datos.

   Coincidencia no encontrada en Cloud Commons

   Si no se encuentra una coincidencia en Cloud Commons para este servicio, haga clic en Explorar para buscar en Cloud Commons un servicio alternativo, o use las opciones disponibles para agregar su servicio a Cloud Commons (en la página 257).

4. Haga clic en el botón Compartir datos si se encuentra una coincidencia, y haga clic en Enviar a Cloud Consortium para introducir el envío de su categoría.

   Se cierra el cuadro de diálogo Uso compartido de datos.

Adición de su servicio a Cloud Commons

Cuando empiece a compartir datos de comparación en Cloud Commons, es posible que no encuentre una coincidencia para su servicio. Puede buscar en Cloud Commons una coincidencia alternativa, o solicitar que se agregue su categorización a Cloud Commons.

Siga estos pasos:

1. Trabaje con el procedimiento para compartir datos de comparación en Cloud Commons (en la página 256).
2. Si no se encuentra una coincidencia, se abrirá una nueva ventana que le permitirá explorar la lista de servicios de Cloud Commons para buscar una coincidencia. Si no encuentra una coincidencia, envíe su servicio para incluirlo en Cloud Commons mediante los pasos siguientes.
3. Haga clic en el icono Agregar servicio para abrir una nueva ventana donde puede trabajar con las opciones siguientes:

   Creación de un nombre de servicio de alias.

   Especificación del nombre del servicio que desea enviar.

   Selección de una categoría de servicio y capacidades

   Trabaje con la lista de categorías de servicio y capacidades para seleccionar aquellos que coincidan con el nombre de su servicio.

   Introducción de una descripción del servicio

   Introduzca información descriptiva adicional para aclarar el motivo de su envío.
4. Haga clic en el botón Enviar a la base de datos global de SMI.

   Su solicitud se envía y usted recibirá una confirmación por correo electrónico.

Adición de servicios en la página Detección y gestión de servicios

Aunque la mayor parte de sus servicios se puede detectar automáticamente, también podrá agregar manualmente un servicio en la página Detección y gestión de servicios.

Siga estos pasos:

1. En la página Detección y gestión de servicios, haga clic en la ficha Gestión de los datos y del uso compartido de los datos.
2. Haga clic en el menú desplegable Agregar servicios y seleccione Manualmente.

   Se abre el cuadro de diálogo Detalle del servicio.
3. Haga clic en la ficha Descripción general e introduzca la información siguiente:

   Origen (sólo lectura)

   Especifica que el origen para el servicio se ha introducido manualmente.

   Nombre

   Especifica el nombre del servicio para mostrarlo en la tabla de servicios.

   Descripción (opcional)

   Especifica la descripción del servicio.

   Estado

   Permite especificar si el servicio se gestiona o no se gestiona.

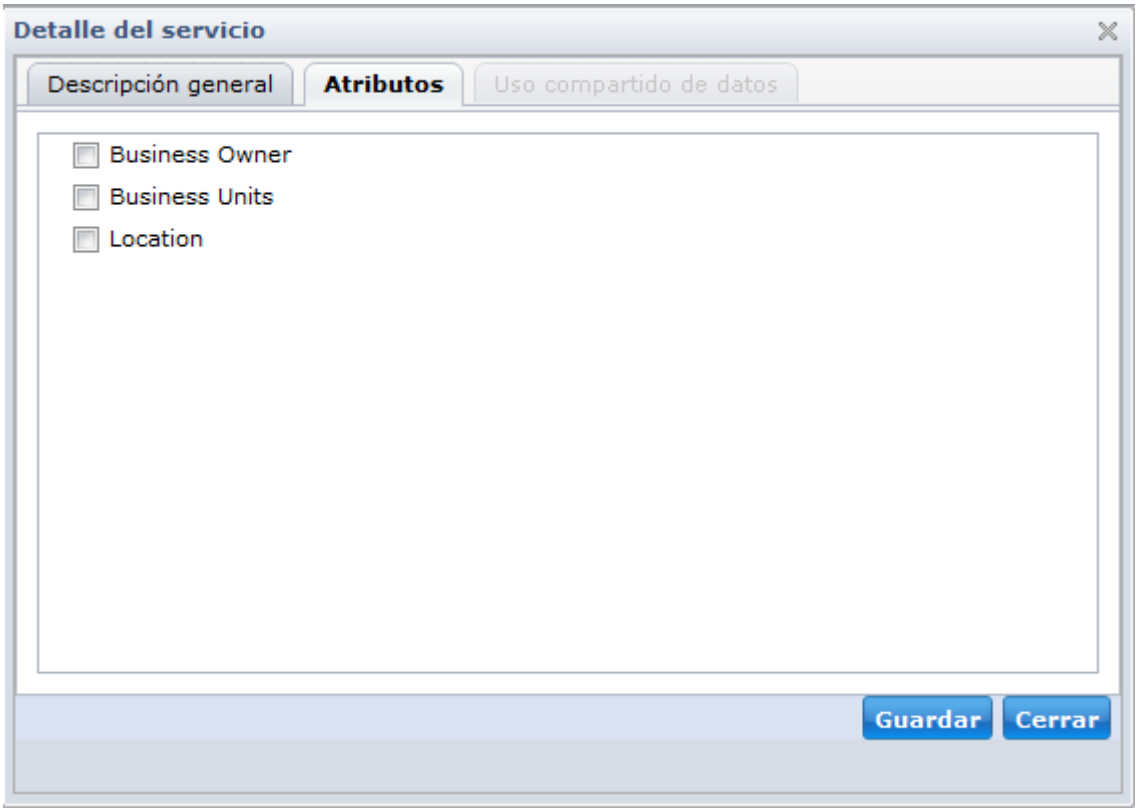
   Suministrador

   Permite seleccionar el suministrador del servicio en una lista desplegable.

   Activar el cálculo de métricas de comparación

   Permite activar o desactivar el cálculo de métricas de comparación.

4. Haga clic en la ficha Atributos para asociar un atributo existente al servicio. La asociación le permite entonces aplicar un filtro en el atributo en las opciones de vista. Por ejemplo, para agregar el atributo de ubicación Nueva York al servicio, seleccione la casilla de verificación en la lista Ubicación. Haga clic en Guardar para activar la ficha Uso compartido de datos.



5. Haga clic en la ficha Uso compartido de datos y trabaje con las opciones disponibles para compartir datos para este servicio en Cloud Commons.

Nota: Puede importar también datos de servicio manualmente (en la página 262) con la función InsertUtil.

Exportación e importación de datos de servicio manualmente

CA Business Service Insight proporciona una utilidad que le permite exportar e importar datos de servicio con un script de línea de comandos. Puede exportar datos de servicio a un archivo CSV o XML, y, a continuación, modificar el archivo fuera de CA Business Service Insight. La utilidad también le permite formatear datos de servicio de otro origen para poder importarlo manualmente a CA Business Service Insight

La utilidad Exportar e importar servicios es compatible con los siguientes datos:

Todos los atributos de servicio estándares:

- Nombre del servicio
- Descripción
- Nombre del proveedor
- Origen del servicio
- Se comparte
- Se gestiona

Metadatos del servicio: atributos personalizados:

- Adición de atributos nuevos

Categorías de servicio y capacidades

- Se genera información de categoría automáticamente si se proporciona información de capacidad pero no aparece la información de categoría.
- Se genera información de categoría automáticamente si se proporciona información de categoría pero no aparece la información de capacidad.

Jerarquía de servicio

- La información secundaria puede determinarse de forma automática desde la información principal

Exportación manual de datos de servicio

Puede exportar datos de servicio manualmente a un archivo CSV o XML.

El proceso de exportación utiliza los siguientes comandos:

```
InsightUtil /exportservice [/http|/https] /server <nombre de host> /port <número de puerto> /format csv|xml /path <ruta de archivo> /key <clave> /secret <secreto>
```

InsightUtil

El comando empieza con el nombre de la utilidad, InsightUtil.

/exportservice [/http|/https]

Especifica que el servicio de exportación utiliza HTTP o HTTPS.

/server <nombre de host>

Indica el nombre de host del servidor.

/port <número de puerto>

Indica el número de puerto.

/format csv|xml

Indica el formato de archivo de exportación: CSV o XML.

/path <ruta de archivo>

Indica la ruta para entregar el archivo de exportación.

/key <clave>

Indica la clave de administrador única de CA Business Service Insight única

/secret <secreto>

Indica la contraseña de administrador única de CA Business Service Insight única

Ejemplo: Exportación de datos de servicio a un archivo CSV local

En este ejemplo, se muestra la forma de exportar datos de servicio a un archivo CSV local.

```
InsightUtil /exportservice /https /server name001 /port 8443
/format csv /path C:\MyFiles\file.csv /key mykey /secret
mypassword
```

Importación manual de datos de servicio

Puede importar datos de servicio manualmente de un archivo de CSV o XML. Debe importar un archivo CSV o XML en el formato definido durante el proceso de exportación. Puede crear y trabajar con un archivo de plantilla de servicio mediante el procedimiento de exportación para asegurarse de que los datos tengan el formato correcto.

El proceso de importación utiliza los siguientes comandos:

```
InsightUtil /importservice [/http|/https] /server <nombre de host> /port <número de host> /format csv|xml /path <ruta de archivo> /key <clave> /secret <secreto>
```

InsightUtil

El comando empieza con el nombre de la utilidad, InsightUtil.

/importservice [/http|/https]

Especifica que el servicio de importación utiliza HTTP o HTTPS.

/server <nombre de host>

Indica el nombre de host del servidor.

/port <número de puerto>

Indica el número de puerto.

/format csv|xml

Indica el formato de archivo de importación: CSV o XML.

Importante: Al importar un archivo CSV, la primera línea (la línea de columna de encabezado) no se debe omitir o el proceso de importación generaría un error. El orden de columna del archivo CSV debe seguir el formato de un archivo de plantilla o un archivo CSV exportado, y no se puede cambiar.

/path <ruta de archivo>

Indica la ruta y nombre del archivo de importación

/key <clave>

Indica la clave de administrador única de CA Business Service Insight única

/secret <secreto>

Indica la contraseña de administrador única de CA Business Service Insight única

Ejemplo: importación de datos de servicio de un archivo CSV local

En este ejemplo, se muestra cómo importar datos de servicio desde un archivo CSV local mediante https.

```
InsightUtil /importservice /https /server name001 /port 8443
/format csv /path C:\MyFiles\file.csv /key mykey /secret
mypassword
```

Exportación de un archivo de plantilla de servicio

Si planea importar datos de servicio de forma manual a CA Business Service Insight mediante la función InsightUtil, el archivo CSV o XML deberá tener el formato correcto. Puede exportar un archivo de plantilla de servicio desde CA Business Service Insight en formato CSV o XML, y, a continuación, agregar datos de servicio antes de importarlo.

Siga estos pasos:

Utilice el siguiente comando:

```
InsightUtil /exporttemplate /format csv|xml /path <file_path>
```

El archivo de plantilla exportado contiene toda la información de formato necesaria y algunos datos de muestra.

Apéndice A: Dominios del servicio de muestra y categorías de dominio

La tabla siguiente es una compilación de dominios del servicio y categorías de dominio que se utilizan habitualmente.

Dominio del servicio	Categoría del dominio	Comentarios
Disponibilidad del sistema	% de tiempo disponible	
	Número de cortes/tiempos de inactividad	
	Promedio de tiempo de reparación	
	Tiempo medio entre errores	
Disponibilidad del servicio	Minutos de tiempo de inactividad	
	Número de días de cálculo previo	
Gestión financiera	Coste del servicio	
Rendimiento del proceso	% del proceso finalizado puntualmente	
	Número de ciclos de proceso finalizados	
Gestión de incidentes	% de incidentes resueltos < T1	
	% de incidentes respondidos < T1	
	Número de incidentes	
	% de incidentes convertidos en problemas	
Satisfacción del cliente	% de satisfacción del cliente	
	Puntación media del nivel de satisfacción del cliente	
Rendimiento del servicio de asistencia	% de llamadas abandonadas	
	Tiempo de captación de llamada medio	
	% FLLR	(Tasa de resolución de primer nivel)

Dominio del servicio	Categoría del dominio	Comentarios
Calidad de datos	% de precisión	
	% de tiempo de respuesta	
	Número total de errores/defectos	

Apéndice B: Ejemplos de casos prácticos

Esta sección contiene los siguientes temas:

Ejemplos de modelado del contrato (en la página 267)
Ejemplo de modelado de métrica financiera (en la página 279)
Ejemplos de modelado de datos (en la página 287)
Uso del ejemplo de atributos personalizados (en la página 298)
Ejemplos de scripts de traducción (en la página 302)
Ejemplos de scripting de lógica de negocios (en la página 308)
Ejemplos de escritura de lógica de negocios vigente (en la página 324)
Caso práctico 19: Asistente del adaptador para origen de datos basado en archivo (en la página 341)
Caso práctico 21: Ejemplo de integración de LDAP (en la página 359)
Caso práctico 22: Generación de informes mediante PERL (en la página 366)

Ejemplos de modelado del contrato

Para cada uno de los casos prácticos, utilice los elementos de categorización siguientes para los objetivos descritos:

- Servicio
- Dominio del servicio
- Categoría del dominio
- Ranura de tiempo
- Destino
- Período de seguimiento
- Unidad de medida
- Perfil de cálculo
- Parámetros contrato\métrica

Caso práctico 1: Disponibilidad del sistema

Considere la garantía contractual siguiente:

"El sistema X debe estar disponible como mínimo para el 99,6 % del mes durante horarios comerciales".

Esto se podría describir mediante las entidades de CA Business Service Insight siguientes:

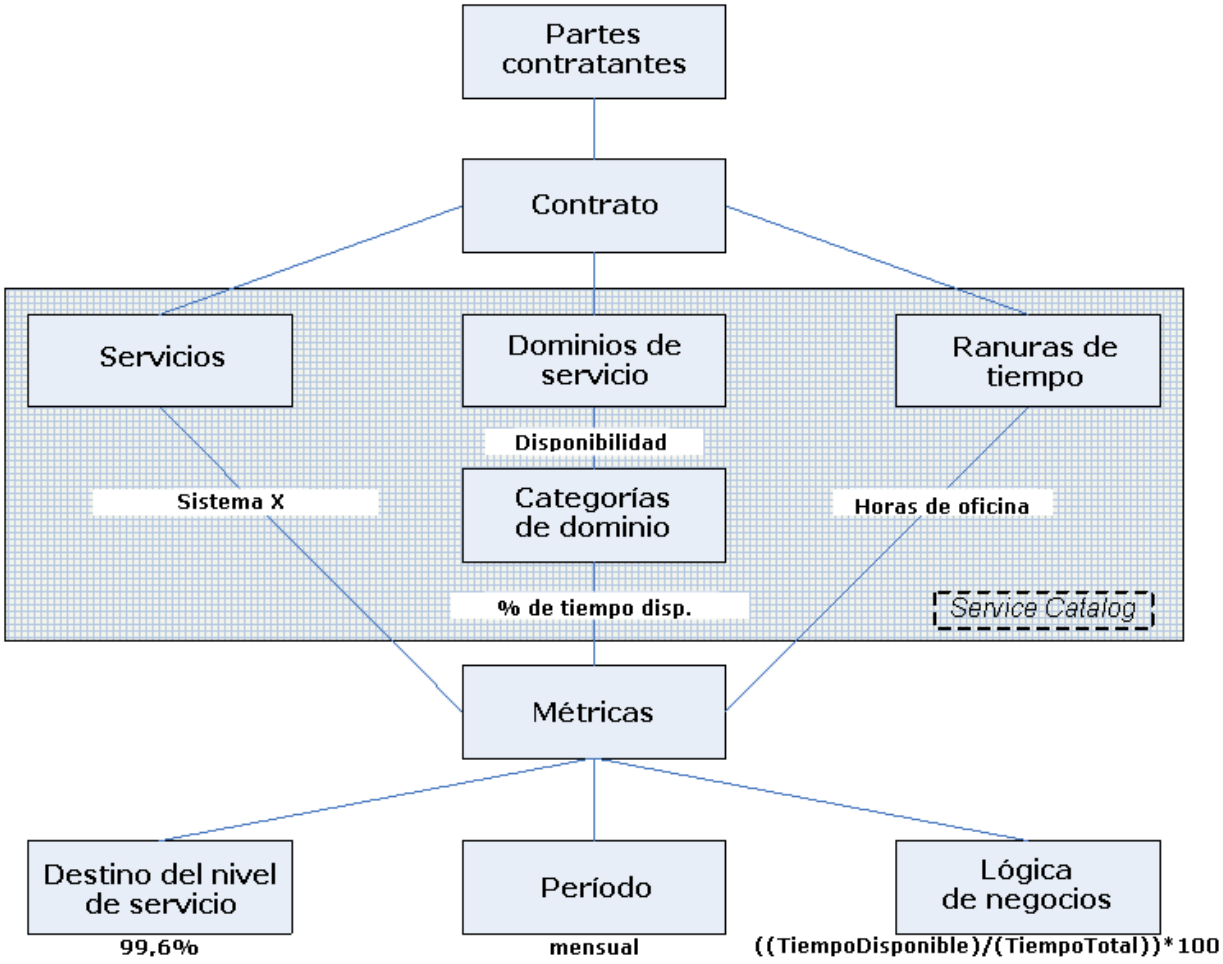
- **Nombre de la métrica**: % del tiempo disponible del sistema X
- **Período de seguimiento**: 1 mes
- **Ranura de tiempo**: horarios comerciales (es necesario definirlo más tarde)
- **Lógica de negocios**: ((tiempo disponible)/(tiempo total))*100

  Nota: Este caso práctico solamente tiene en cuenta donde cae el control en horarios comerciales (que es la ranura de tiempo de la métrica).
- **Destino**: 99,6 %

Además de la información de la métrica anterior, los elementos siguientes del catálogo de servicios del sistema se pueden deducir también de la descripción anterior:

- **Servicio**: sistema X.
- **Dominio del servicio**: disponibilidad.
- **Categoría del dominio**: % del tiempo disponible.
- **Grupo de servicios**: cualquier grupo que identifica más de un sistema que se puede monitorizar. En esta etapa, es difícil establecer si se puede crear o no un grupo apropiado.

De esta manera, ya puede reproducir el diagrama de la sección Modelado del contrato de este documento que muestra estas entidades en un diagrama.



Caso práctico 2: Disponibilidad del sistema 2

La disponibilidad del sistema CNP debería mantener los niveles siguientes:

Entorno	Días laborables	Fines de semana
Producción	99,9 %	98,9 %
Desarrollo	90 %	N/D
Prueba/control de calidad	Ninguna garantía	N/D
Red	99,9 %	N/D

Soluciones sugeridas:

Métrica	**Promedio de producción del sistema CNP durante los días laborables**
Destino	99,9
Período de seguimiento.	1 mes
Unidad de medida	%
Servicio	Producción del sistema CNP
Dominio del servicio	Disponibilidad
Categoría del dominio	Disponibilidad de la aplicación
Ranura de tiempo	días laborables
Métrica	**Promedio de producción del sistema CNP durante los fines de semana**
Destino	98,9
Período de seguimiento.	1 mes
Unidad de medida	%
Servicio	Producción del sistema CNP
Dominio del servicio	Disponibilidad
Categoría del dominio	Disponibilidad de la aplicación
Ranura de tiempo	fines de semana
Métrica	**Promedio de desarrollo del sistema CNP durante los días laborables**
Destino	90
Período de seguimiento.	1 mes
Unidad de medida	%
Servicio	Desarrollo del sistema CNP
Dominio del servicio	Disponibilidad
Categoría del dominio	Disponibilidad de la aplicación
Ranura de tiempo	días laborables

Métrica	**Promedio de pruebas/control de calidad del sistema CNP durante los días laborables**
Destino	ninguno
Período de seguimiento.	1 mes
Unidad de medida	%
Servicio	Prueba/control de calidad del sistema CNP
Dominio del servicio	Disponibilidad
Categoría del dominio	Disponibilidad de la aplicación
Ranura de tiempo	días laborables
Métrica	**Disponibilidad de la red de sistema CNP**
Destino	99,9
Período de seguimiento.	1 mes
Unidad de medida	%
Servicio	Red del sistema CNP
Dominio del servicio	Disponibilidad
Categoría del dominio	Disponibilidad de red
Ranura de tiempo	Siempre

Caso práctico 3: Tiempo de respuesta del sistema

El siguiente caso práctico ilustra las métricas de tiempo de respuesta. Los contratos se pueden modelar de varias maneras, cada una con sus propias ventajas.

El ejemplo siguiente examina los diversos métodos de modelado:

Solución de modelado sugerida A	**Valor máximo**
Tiempo de respuesta en transacciones del sistema CNP	No se puede superar los 750 milisegundos por mes
Nombre de métrica	Tiempo de respuesta en transacciones máximo
Destino	750

Período de seguimiento.	1 mes
Unidad de medida	Milisegundos
Ranura de tiempo	Siempre
Servicio	CNP System
Dominio del servicio	Rendimiento
Categoría del dominio	Tiempo de respuesta en transacciones máximo

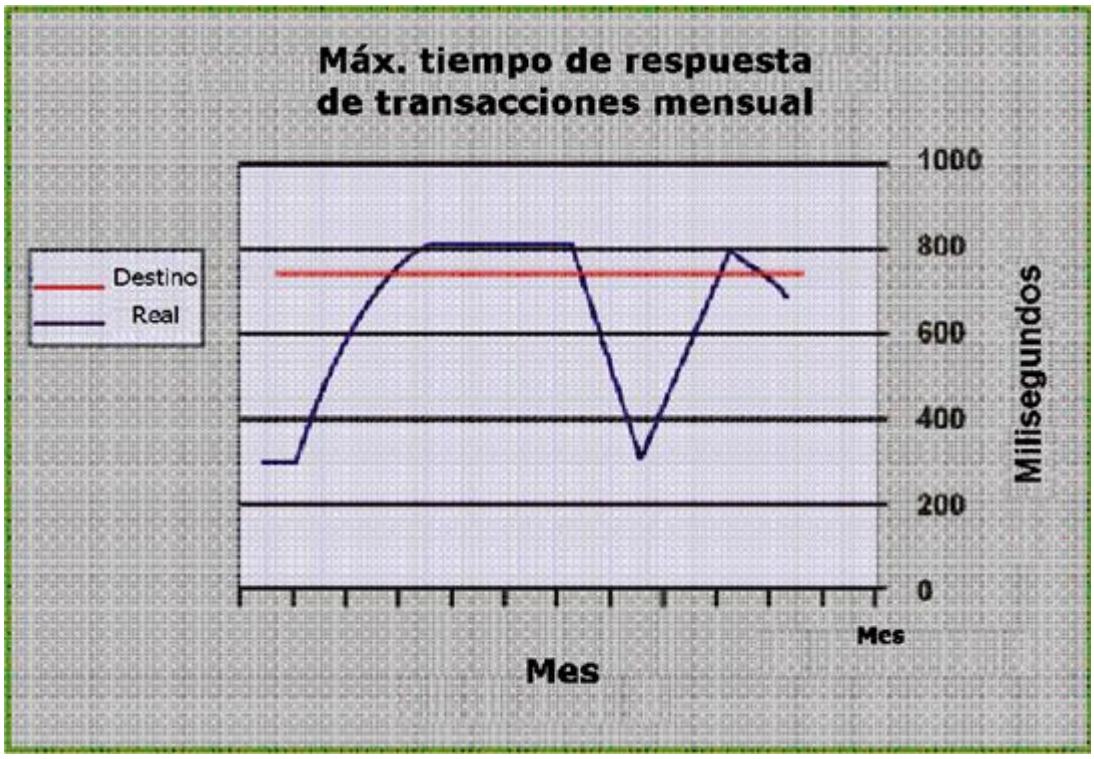
En base a la matriz superior, ¿cómo se calcula el nivel de servicio real?

En base a la definición de la categoría del dominio, parece que el nivel de servicio real se calcula como un valor máximo. Esto implica que, para todas las transacciones realizadas durante un mes, se captura la transacción con el valor máximo y este valor se compara con el destino.

Nota: El cálculo del nivel de servicio se basa en una agregación de datos sin procesar en un período de tiempo dado. Para cada período de tiempo, la métrica proporciona un solo resultado. El destino de una métrica no se compara con una sola transacción, sino que se compara con un resultado mensual que corresponde a la agregación periódica de todas las transacciones dentro de ese período. El gestor de contratos debe asegurarse de que este resultado refleja el contrato por un lado y la calidad del servicio por otro.

Tenga en cuenta que medir el tiempo de respuesta como valor máximo es una obligación muy estricta y muy difícil de lograr en la práctica. Medir un nivel máximo significa que una sola transacción de 751 ms entre un millón de transacciones llevadas a cabo durante el curso de un mes es suficiente como para generar una infracción del contrato. Todas las barras en los informes aparecerán por lo tanto en rojo y no reflejarán la calidad real del servicio que se ha proporcionado.

La ilustración siguiente muestra un informe típico en estas circunstancias.



Cualquier transacción que supere el destino se considerará una infracción del contrato, pero como base para entender que la calidad del servicio real proporcionada es muy pobre, ya que solamente refleja una sola transacción y no se sabe nada respecto al resto de las transacciones, como por ejemplo si era un único error o una tendencia. Si no era un incidente aislado, entonces ¿cuántos errores hubo? ¿o cuál es la tasa de transacciones erróneas con respecto al número total de transacciones realizadas durante el mes? Esos resultados pueden incluir varios meses y, por consiguiente, una infracción del contrato, ¿pero cuál es la tendencia? ¿Está mejorando o empeorando? Todas estas son preguntas que el gestor de nivel de servicio podría realizar, y el informe deberá ser capaz de proporcionar las respuestas.

Nota: Al definir la métrica y el esquema de cálculo asociado, es muy importante imaginar cómo aparecerán los resultados en un informe. Este informe debe proporcionar dos elementos cruciales:

- Si se ha producido una infracción del contrato.
- Debe proporcionar a los gestores suficiente transparencia en los datos y capacidad para investigar la causa raíz del error, así como debe darles las herramientas necesarias para entender completamente sus componentes de servicio.

Solución de modelado sugerida B	**Promedio de tiempo de respuesta**
Tiempo de respuesta en transacciones del sistema CNP	No debe superar los 750 milisegundos por mes
Métrica	Promedio de tiempo de respuesta en transacciones
Destino	750
Período de seguimiento.	1 mes
Unidad de medida	Milisegundos
Categoría del dominio	Promedio de tiempo de respuesta en transacciones

Calcular el tiempo de respuesta medio da una idea mejor de la calidad de servicio mensual, al tiempo que puede reflejar esos meses con tiempos de respuesta extremos o fuera de contrato.

Solución de modelado sugerida C	**Porcentaje de transacciones correctamente finalizadas bajo el umbral.**
Tiempo de respuesta en transacciones del sistema CNP	No debe superar los 750 milisegundos por mes
Métrica	Tiempo de respuesta de transacciones correctas
Destino	100
Período de seguimiento.	1 mes
Unidad de medida	% de éxito
Parámetro de métrica	750 ms
Servicio	CNP System
Dominio del servicio	Rendimiento
Categoría del dominio	Tiempo de respuesta de transacciones correctas
Ranura de tiempo	Siempre

Mediante este método, el cálculo será el porcentaje de transacciones correctamente completadas bajo el umbral de 750 ms durante el período especificado según la fórmula:

*((Número de transacciones bajo 750 ms.)/(Total de transacciones))*100*

Expresar la garantía como una tasa de éxito proporciona la capacidad de conservar una garantía estricta (destino 100 %), mientras que también permite indicar el valor real que representa lo bueno o malo que fue el servicio.

Mediante este método, el destino no es ahora un límite superior de 750 ms, sino la tasa que debe mantenerse. En casos en los que la garantía deba ser estricta, el destino deberá ser 100 %, lo cual no deja lugar para siquiera un solo error. En este caso se ha introducido una variable adicional, el parámetro de métrica. Este parámetro se deberá implementar como un parámetro de métrica para permitir hacer modificaciones sencillas, si fuera necesario.

Un modelo alternativo a este método puede ser imponer un modelo de tipo escalación:

Las soluciones siguientes definen tres métricas en lugar de una sola métrica, como ocurría en las soluciones anteriores.

Métrica	Tiempo de respuesta de transacciones correctas
Destino	95
Período de seguimiento.	1 mes
Unidad de medida	% de éxito
Parámetro de métrica	750 ms
Métrica	Tiempo de respuesta de transacciones correctas
Destino	99
Período de seguimiento.	1 mes
Unidad de medida	% de éxito
Parámetro de métrica	850 ms
Métrica	Tiempo de respuesta de transacciones correctas
Destino	100

Período de seguimiento.	1 mes
Unidad de medida	% de éxito
Parámetro de métrica	1000 ms

En caso de que sea necesario comunicar la obligación contractual, así como el número de transacciones que superen el umbral de 750 ms, es necesario definir una métrica adicional para contar el número de transacciones erróneas.

Nota: Cada métrica genera un solo resultado en un período de tiempo dado. Si se establece para calcular el porcentaje de transacciones, no puede proporcionar también el número de transacciones.

La única forma de producir informes adicionales desde una métrica es utilizar las salidas de la lógica de negocios. (Consulte Salidas: tablas de usuario (en la página 175) para obtener más información acerca de los resultados de la lógica de negocios).

Métrica	Número de transacciones erróneas
Destino	Ningún destino
Período de seguimiento.	1 mes
Unidad de medida	Número de transacciones
Parámetro de métrica	750 ms
Servicio	CNP System
Dominio del servicio	Rendimiento
Categoría del dominio	Número de transacciones
Ranura de tiempo	Siempre

Caso práctico 4: Rendimiento del servicio de asistencia

Caso práctico que ilustra una situación de servicio de asistencia

El servicio de asistencia debe lograr el 100 % del éxito en todos los aspectos siguientes:

Tipo de ticket	Tiempo de resolución
Prioridad 1	1 hora
Prioridad 2	2 horas

Prioridad 3	4 horas

Modelado sugerido, solución A:

Métrica	**Tiempo de resolución de prioridad 1**
Destino	100
Período de seguimiento.	1 mes
Unidad de medida	% de éxito
Parámetro de contrato	Matriz de hora de resolución
Servicio	Servicio de asistencia
Dominio del servicio	Rendimiento del servicio de asistencia
Categoría del dominio	Tiempo de resolución del ticket
Ranura de tiempo	Siempre

La matriz anterior se aplica a tres métricas. Para cada prioridad, se define una métrica separada con todas las prioridades dentro de las mismas categorías.

Modelado sugerido, solución B:

La definición de la métrica sigue siendo la misma que en la solución A.

Opción 1:

Servicio	Servicio de asistencia
Dominio del servicio	Prioridad de gestión del ticket 3
Categoría del dominio	Tiempo de resolución del ticket
Categoría del dominio	Tiempo de respuesta del ticket
Ranura de tiempo	Siempre

Opción 2:

Servicio	Servicio de asistencia
Dominio del servicio	Gestión de tickets
Categoría del dominio	Tiempo de resolución del ticket de prioridad 3
Categoría del dominio	Tiempo de respuesta del ticket
Ranura de tiempo	Siempre

Caso práctico 5: Copia de seguridad del sistema

Se realiza una copia de seguridad de la siguiente manera:

Marco temporal	Número de copias de seguridad
semanalmente	6
mensualmente	27
anualmente	350

Soluciones sugeridas:

Métrica	Número de copias de seguridad semanal
Destino	6
Período de seguimiento.	1 semana
Unidad de medida	Copias de seguridad
Servicio	Copia de seguridad
Dominio del servicio	Copia de seguridad
Categoría del dominio	Número de copias de seguridad por semana
Ranura de tiempo	Siempre
Métrica	**Número de copias de seguridad mensual**
Destino	27
Período de seguimiento.	1 mes
Unidad de medida	Copias de seguridad
Servicio	Copia de seguridad
Dominio del servicio	Copia de seguridad
Categoría del dominio	Número de copias de seguridad por semana
Ranura de tiempo	Siempre
Métrica	**Números de copia de seguridad anual**
Destino	350

Período de seguimiento.	1 año
Unidad de medida	Copias de seguridad
Servicio	Copia de seguridad
Dominio del servicio	Copia de seguridad
Categoría del dominio	Número de copias de seguridad por semana
Ranura de tiempo	Siempre

Ejemplo de modelado de métrica financiera

El caso práctico siguiente presenta un ejemplo de modelado financiero.

Caso práctico 6: Modelado de las condiciones financieras de modelado de un contrato/servicio

Hay tres tipos generales de métricas utilizadas para el modelado de las condiciones financieras de un servicio o contrato. Estas son:

- Costes de precios fijos
- Costes de consumo
- Cargos por sanciones/incentivos

Tenga en cuenta el siguiente ejemplo:

Una empresa nueva está empezando y requiere que se proporcione un servicio de correo electrónico junto con la instalación y mantenimiento de los buzones de correo. A medida que se contrate nuevo personal, el número de buzones de correo aumentará obviamente. La provisión de un servicio de correo electrónico para un contrato incurre en un coste de precio fijo de 1.000 dólares y, además, hay un coste adicional por buzón de correo que se cargará cada mes. Este coste por buzón de correo es un modelo de fijación de planes de precios de la siguiente manera:

Número de buzones de correo	**Coste por buzón de correo**
1-1.000	1,00 $
1.001 - 5.000	0,80 $
5.001+	0,50 $

Así, mientras más buzones de correo se agregan, más bajo será el coste adicional. (Por ejemplo: 1500 buzones de correo costarán (1000 x 1 $) + (500 x 0.80 $) = 1400 $.) Mediante este modelo, se pueden producir dos métricas para reflejar esto en el contrato.

- Coste de servicio de correo electrónico (fijo)
- Coste del uso del buzón de voz (variable/basado en el consumo)

Además hay una estimación realizada por el equipo de gestión del número de miembros del personal a lo largo del año (2007), que se describe a continuación. La tendencia viene marcada por el crecimiento inicial de la empresa mediante el empleo y por la consiguiente apertura de oficinas nuevas en otras regiones:

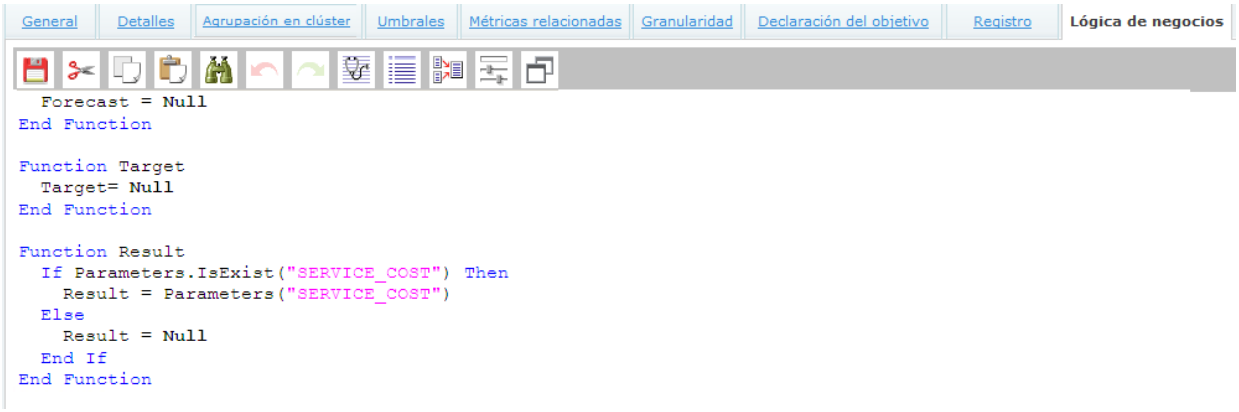
Ene	Feb	Mar	Abr	Mayo	Jun	Jul	Ago	Sep	Oct	Nov	Dic
50	100	500	900	1600	1700	1800	2500	2600	3500	3600	5800

Para modelar estas métricas, haga lo siguiente:

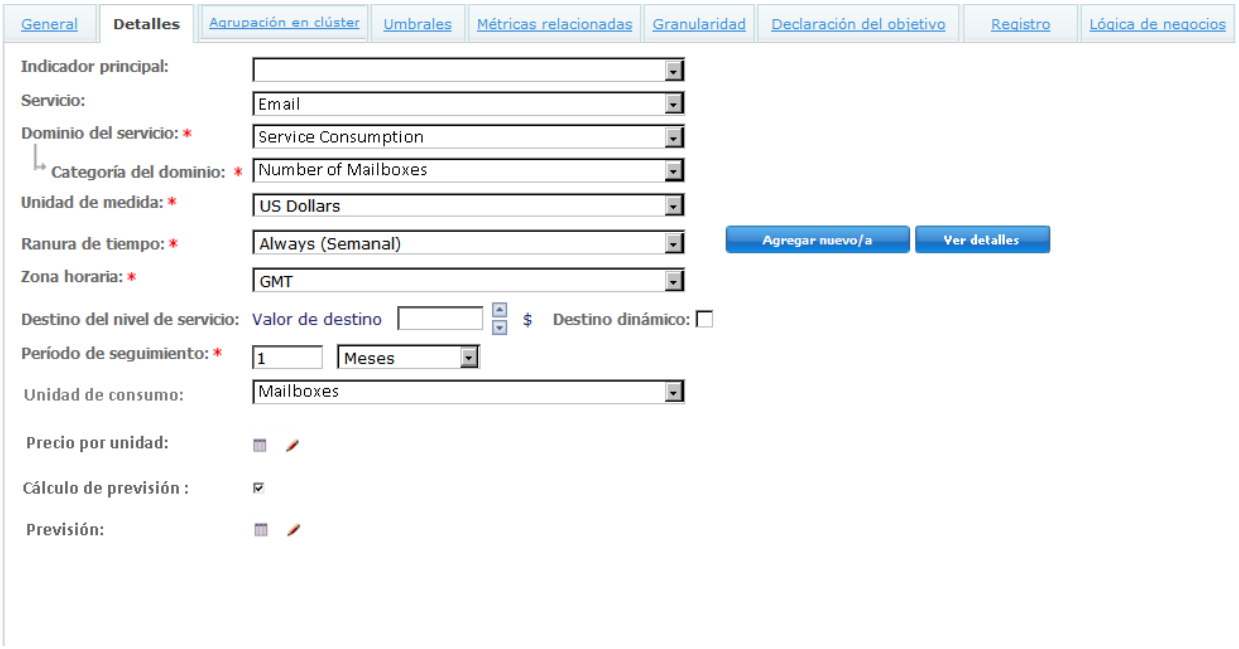
Cree la métrica de coste fijo (utilice el tipo Elemento del precio) en el contrato, mediante la información siguiente:

Para especificar el coste fijo del contrato de este servicio, implemente esto como un parámetro en la lógica de negocios (donde se deberá devolver el coste fijo de la función Resultado). Este parámetro se puede exponer mediante la declaración del objetivo de la métrica, como se muestra a continuación:

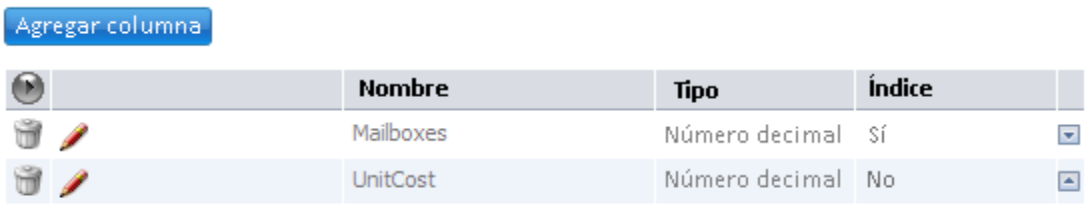
La devolución del valor de parámetro para esta métrica es tan solo un caso de devolver el valor del coste del servicio mediante la función Resultado.



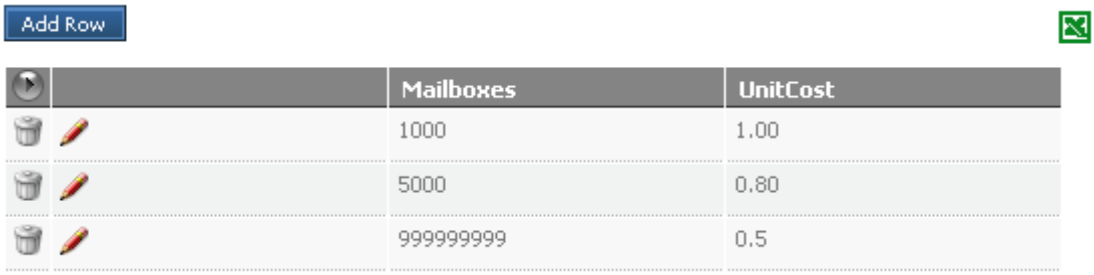
A continuación, cree la métrica de fijación de precios variables (de nuevo, utilice el tipo Elemento del precio) para determinar los costes de consumo del número de buzones de correo utilizados. Nombre a esta métrica "*Coste del consumo del buzón de correo*" y créela con la información siguiente:



En este caso, necesita introducir los parámetros de consumo en los detalles de la métrica. Estos se introducirán en la tabla de precio por unidad. Para modelar la tabla anterior para el número de buzones de correo frente al coste, cree una columna para el límite superior de buzones de correo y otra para el precio por unidad:



A continuación introduzca los valores para cada plan. En este caso, el límite superior de buzones de correo determina el coste asociado. Como hay 3 planes, se agregan a la tabla de esta manera:

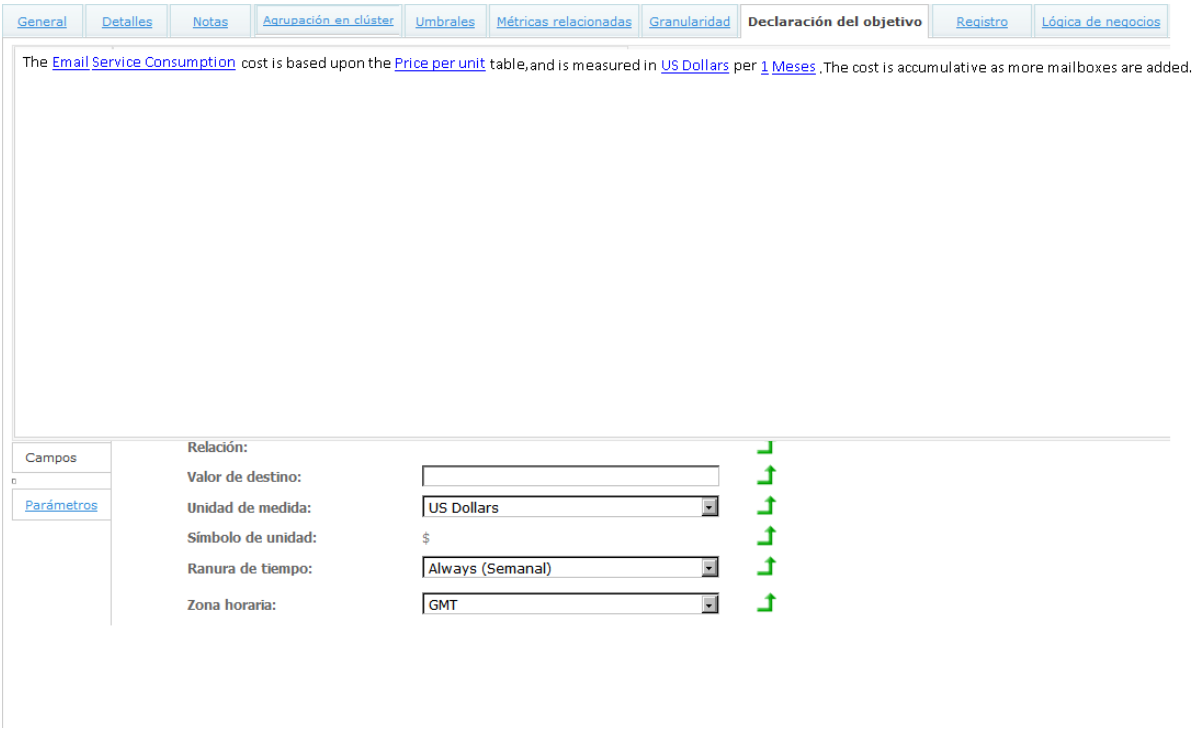


Además de esto, implemente la función de previsión sobre el consumo de buzones de correo. Para ello, cree la tabla de previsión con el diseño mensual preestablecido.

Esta se rellena a continuación con los valores de las tablas proporcionadas en la descripción del escenario.

Consumo	Ene-01 2007	Feb-01 2007	Mar-01 2007	Abr-01 2007	May-01 2007	Jun-01 2007	Jul-01 2007	Ago-01 2007	Sep-01 2007	Oct-01 2007	Nov-01 2007	Dic-01 2007
1	50	100	500	900	1600	1700	1800	2500	2600	3500	3600	5800

Ahora puede agregar la declaración del objetivo para la métrica. En este caso, no se necesitan valores de parámetros, ya que se obtienen de las tablas de precio por unidad y previsión.



Finalmente, complete la lógica de negocios de la siguiente manera:

```
Opción explícita

Dim PPUmap1, PPUmap2, PPUmap3, PPUkey, FCmap, periodFC, TierPPU
Dim currentMonth, TotalMailboxes, MailboxesThisPeriod, TotalPrice

Sub OnRegistration(dispatcher)
   'sólo registro de muestra
   dispatcher.RegisterByMetric "OnMailboxAddedEvent", "NewMailboxEventType", _
      "MailboxResource", "MONTH", "MetricName", "MetricContract", _
      "MetricContractParty"
End Sub

Sub OnLoad(TIME)
   'Inicializar las asignaciones de planes de precios y de previsión
   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: AÑADIR código aquí PARA tratar el evento de INICIO de período
   currentMonth = GetMonth (time)
   If Context.IsInForecast Then
      periodFC = getForecastValue (currentMonth)
   End If
   MailboxesThisPeriod = 0
   TotalPrice = 0
End Sub

Sub OnPeriodEnd(TIME, isComplete)
   ' Calcular el precio actual de todos los buzones de correo utilizando
   ' el modelo de planes de precios
   ' Utiliza un enfoque acumulativo a medida que pasa por cada plan para
   ' determinar el coste 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
```

```
Function Forecast
   Forecast = getMailboxCost (periodFC)
End Function

Function Target
   Target = Null
End Function

Resultado de la función
   result = TotalPrice
End Function

Function getforecastvalue(q)
   getforecastvalue = FCmap (q)
End Function

Function getmonth(time)
   'esta función recupera el mes
   Dim lTime
   lTime = Tools.GetLocaleTime(time)
   getmonth = monthname (datepart ("m", lTime), True) & _
      "-0" & datepart ("d", lTime) & "-" & datepart ("yyyy", lTime)
End Function

Function getMailboxCost(num_boxes)
   'Function calcula el coste de los buzones de correo utilizando el modelo de
planes de precio
   Dim returnValue
   If num_boxes <= PPUmap1 ("Mailboxes") Then
      'Primer plan
      returnValue = num_boxes * PPUmap1 ("UnitCost")
      'Out.Log "Tier1: " & num_boxes
   Else If num_boxes > PPUmap1 ("Mailboxes") And num_boxes <= PPUmap2
("Mailboxes") Then
      'segundo plan solamente
      returnValue = (PPUmap1 ("Mailboxes") * PPUmap1 ("UnitCost")) + _
         ((num_boxes - PPUmap1 ("Mailboxes")) * PPUmap2 ("UnitCost"))
      'Out.Log "Tier2: " & num_boxes
   Else If num_boxes > PPUmap2 ("Mailboxes") Then
      'tercer plan
      returnValue = (PPUmap1 ("Mailboxes") * PPUmap1 ("UnitCost")) + _
         ((PPUmap2 ("Mailboxes") - PPUmap1 ("Mailboxes")) * PPUmap2 ("UnitCost"))
+ _
         ((num_boxes - PPUmap2 ("Mailboxes")) * PPUmap3 ("UnitCost"))
      'Out.Log "Tier3: " & num_boxes
   End If
   getMailboxCost = returnValue
   'Out.Log "Cost is: " & returnValue
```

```
End Function
```

Nota: Este script de lógica de negocios domina tanto el cálculo de previsión (mediante la tabla de previsión) como los resultados del coste de consumo financiero. Ambos usan la misma fórmula getMailboxCost() que calcula el plan de precio fijo basado en la tabla de precio por unidad definida para esta métrica.

Ejemplos de modelado de datos

Los dos casos siguientes ilustran el proceso de modelado de datos básico, así como algunos de los puntos más importantes que pueden estar implicados.

Como el proceso de modelado de datos se hace después del proceso de modelado del contrato, los requisitos del cálculo derivados de las garantías del contrato son ya conocidos, pues se identificaron y definieron como parte del proceso de modelado del contrato.

El modelo de datos debe tener en cuenta todos los requisitos del cálculo.

En los dos casos prácticos siguientes, se elige un único requisito o se seleccionan varios para demostrar el proceso de modelado de datos.

Caso práctico 7: Rendimiento del servidor

Es un caso práctico típico sobre el rendimiento del servidor.

Tenemos la estructura de origen de datos siguiente:

Indicación	Servidor	Medida	Marca de tiempo
disponibilidad	Appserv01	1	03/01/2001 07:44
tiempo de respuesta	Appserv01	354.6	03/01/2001 09:58
carga de la CPU	Dbserv02	83 %	03/01/2001 12:12
disponibilidad	Appserv01	0	03/01/2001 14:26
carga de la CPU	Dbserv02	94,30 %	03/01/2001 16:40
capacidad	Firewall01	10 %	03/01/2001 18:54
tiempo de respuesta	Dbserv02	476,89	03/01/2001 21:08
disponibilidad	Appserv02	1	03/01/2001 21:24

Indicación	Servidor	Medida	Marca de tiempo
tiempo de respuesta	Appserv01	774,4	03/01/2001 21:48
carga de la CPU	Dbserv01	83 %	03/01/2001 21:52

Además de lo anterior, tenemos los requisitos de cálculo siguientes:

Calcule el % de disponibilidad de cada servidor de aplicaciones.

La disponibilidad de cada servidor se debe calcular por separado. Por lo tanto, para calcular la disponibilidad de un único servidor es necesario recibir eventos solamente para ese servidor específico. Además, los orígenes de datos contienen otros indicadores de rendimiento que no son relevantes para los cálculos de disponibilidad (tiempo de respuesta, capacidad, entre otros); de esta manera, los indicadores de disponibilidad y el servidor que son relevantes deben filtrarse.

Como los criterios de filtrado de CA Business Service Insight son tipo de evento y recursos, traduzca los criterios de filtrado de los valores de origen de datos a una definición de recurso y tipo de evento.

En este caso, el indicador es un valor ideal para la traducción como un tipo de evento de CA Business Service Insight, pues describe lógicamente el tipo de evento. Hay un número limitado de tipos, como, disponibilidad, tiempo de respuesta, capacidad y carga de la CPU. Esto significa que para la métrica que calcula la disponibilidad del servidor, el registro es para el tipo de evento de disponibilidad.

En este caso, cuando hay un gran número de servidores y es necesario calcular la disponibilidad de cada servidor, cada servidor debe definirse como un recurso. A continuación tiene que agruparse dentro de un grupo de recursos y la métrica se agrupará en ese grupo de recursos.

Modelado sugerido:

Nombre del evento Evento de disponibilidad.

Comportamiento Comunicado como cambio de estado de 0 o 1.

Campo Marca de tiempo Marca de tiempo (el único campo de hora en el origen de datos).

Campo Recurso Servidor (todos los servidores que aparecen en el origen de datos se traducen a un recurso de CA Business Service Insight).

Campo Tipo de evento Indicador (cada uno de los valores de este campo se traducen en un tipo de evento de CA Business Service Insight. Hay cuatro tipos de eventos).

Campos de datos	La medida es 0 o 1 (solamente para registros de disponibilidad).

Se deben definir las asignaciones de recursos siguientes:

Atributo Tipo de recurso	Servidor de aplicaciones
Asignación a parte contratante	Cada servidor de aplicaciones se asigna a la parte contratante, donde el servidor relevante ejecuta su aplicación. Esto permite el registro por la parte contratante que recupera todos los servidores correspondientemente.
Asignación a servidor	Igual que la anterior.
Asignación a grupo de recursos	Opcional. Normalmente es necesario agrupar los recursos cuando se requiere el agrupamiento en clúster.

Y por último, en función de todas las definiciones anteriores:

Registro de	Para la métrica en clúster, que calcula la disponibilidad de cada servidor individualmente, el registro se realiza por recurso.

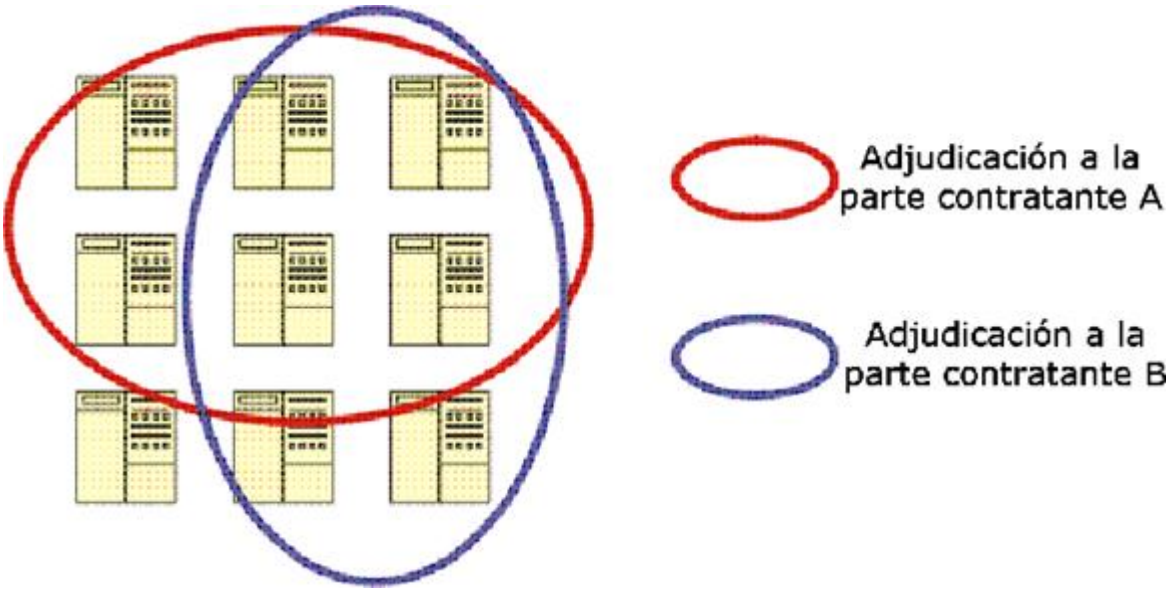
Para poder cumplir el requisito anterior, se agrega el siguiente criterio:

Calcule el tiempo de respuesta medio de los servidores de la aplicación para cada parte contratante de forma separada.

Para este requisito, es necesario recibir eventos de tiempos de respuesta para todos los servidores de aplicaciones que forman parte del grupo de servidores que ejecutan aplicaciones para la parte contratante específica. La recepción de los eventos de tiempo de respuesta se lleva a cabo registrando el tipo de evento creado a partir del campo de indicación con el valor "tiempo de respuesta". Esto garantiza que se reciban solamente los eventos que informan sobre los tiempos de respuesta de los servidores.

Para recibir solamente los eventos que informan de los servidores relevantes para una parte contratante específica, los recursos tienen que registrarse a través de la asignación de la parte contratante.

Un recurso se puede asignar a más de una sola parte contratante, servicio o grupo. Por lo tanto, puede suceder que un evento que se envió para el cálculo de tiempo de respuesta como parte del contrato de la parte contratante A sea también una parte de los cálculos de la parte contratante B.



Caso práctico 8: Rendimiento del servicio de asistencia

Es un caso práctico típico sobre el rendimiento del servicio de asistencia.

Tenemos el origen de datos mostrado a continuación:

Nº de ticket	Prioridad del ticket	Creado el	Cerrado el	Resuelto el	Ref cliente	Ubicación
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	Ipswitch
38084199	2	07/01/2004 11:20	14/01/2004 09:10	09/01/2004 12:00	CM2	Ipswitch
38188329	3	21/01/2004 09:27	27/01/2004 09:09	24/01/2004 12:00	CM3	Leeds
37964069	3	12/12/2003 14:04	05/01/2004 11:35	19/12/2003 12:00	CM286	Birmingham
3796448	1	12/12/2003 14:18	05/01/2004 11:39	19/12/2003 12:00	CM263	Luton
37965039	2	12/12/2003 14:57	14/01/2004 15:05	18/12/2003 12:00	CM264	Leeds
37970699	2	15/12/2003 09:26	05/01/2004 11:22	22/12/2003 12:00	CM288	Londres
37997259	1	17/12/2003 15:58	05/01/2004 11:27	23/12/2003 12:00	CM302	Ipswitch
38000259	1	18/12/2003 09:11	06/01/2004 14:44	29/12/2003 12:00	CM340	Londres
38021049	1	22/12/2003 09:32	06/01/2004 14:28	31/12/2003 12:00	CM341	Londres

El origen de los datos mostrado anteriormente indica los detalles de los tickets del servicio de asistencia que se gestionan para cada cliente en las diversas ubicaciones de servicios de atención al cliente. Una sola ubicación se puede compartir también entre clientes.

Los tres requisitos siguientes son obligatorios para crear los informes al utilizar este origen de datos:

- % de tickets de prioridad 1 resueltos en cuatro horas para clientes CM3.
- % de tickets de prioridad 1 resueltos en cuatro horas para clientes CM3 en cada ubicación.
- % de tickets de prioridad 1 cerrados en un día para clientes CM3 en cada ubicación.

Los requisitos anteriores indican que los eventos se deben filtrar por:

- prioridad
- cliente
- ubicación

Es necesario especificar cuáles de estos criterios de filtrado se traducen como un tipo de evento y cuál será el recurso relevante.

¿Cómo selecciono un tipo de evento?

De los tres criterios de filtrado necesarios, la prioridad del ticket es la más apropiada para que se traduzca al tipo de evento por las siguientes razones:

- Describe el tipo de evento que se está tratando (tickets de prioridad 1).
- El número de las distintas prioridades que existen es relativamente pequeño (prioridad 1, 2, 3).
- El tipo de evento en sí mismo es relativamente constante (el servicio de asistencia raramente cambia las prioridades con las que trata los tickets).

¿Cómo se selecciona un recurso?

Los otros dos criterios de filtrado obligatorios son el cliente y la ubicación y estos son las entidades más pequeñas que requieren la creación de informes. Por lo tanto, la combinación del cliente y la ubicación es el recurso.

El cliente y la ubicación son entidades relativamente fijas y tienen un ciclo de vida definido, por lo que se pueden agregar clientes nuevos o sitios nuevos. Además, la relación entre un sitio y un cliente puede cambiar.

Por motivos de traducción, es posible utilizar más de un campo del origen de datos. Mientras que en el anterior caso práctico el campo de servidor se traducía a un recurso de CA Business Service Insight, en este caso el recurso se construye mediante la combinación de dos campos. Cada permutación, por lo tanto, produce un nuevo recurso.

La lista de recursos se muestra a continuación:

Campo Cliente	Campo Sitio	Recurso de salida
CM3	Londres	CM3@London
CM1	Ipswitch	CM1@Ipswitch
CM2	Ipswitch	CM2@Ipswitch
CM3	Leeds	CM3@Leeds
CM286	Birmingham	CM286@Birmigham
CM263	Luton	CM263@Luton
CM264	Leeds	CM264@Leeds
CM288	Londres	CM288@London
CM302	Ipswitch	CM302@Ipswitch
CM340	Londres	CM340@London
CM341	Londres	CM341@London

El recurso de salida es una combinación del cliente y los campos de sitio utilizando el símbolo '+' para combinarlos. Es importante tener en cuenta el nombre del recurso, ya que se extrajo del origen de datos y aparece posteriormente en los informes. Las capacidades de creación de informes tienen que cubrir unas expectativas.

Nota: Uno de los errores más comunes durante el modelado de un servicio de asistencia o cualquier ticket o sistema de incidencia es definir un único incidente como un recurso.

Los siguientes supuestos incorrectos conducen a menudo a errores:

1. El único incidente es aquel del que se informa. No establezca la entidad que se tiene que comunicar como la entidad para la cual se generan los cálculos, de forma que la única incidencia no es la base de los cálculos del sitio del cliente. En general, el acuerdo de nivel de servicio está basado en el rendimiento global, y no en el rendimiento del tratamiento del ticket individual.
2. La garantía es por nivel de ticket. Esto es un error porque las garantías son periódicas, lo que se calcula es una agregación con el tiempo.

Cómo establecer asignaciones para los recursos

Para el primer requisito del cálculo:

1. % de tickets de prioridad 1 resueltos en cuatro horas para clientes CM3.
   - Es necesario recibir los eventos de los tickets que se pueden atribuir a un cliente específico. El cliente forma parte de este recurso, que también indica el sitio del cliente. Por lo tanto, para capturar todos los eventos relacionados con recursos y que se pueden atribuir a ese cliente, hay dos opciones:
   - En los casos en los que el cliente en el origen de datos representa una parte contratante, los recursos pueden estar adjuntos a la parte contratante relevante. Esto permite realizar el registro por parte contratante. Siempre que es posible, esto es siempre preferible.
   - Cree un grupo de recursos para cada uno de los clientes del origen de datos y agrupe todos los recursos relevantes dentro de este. Mediante este método, el registro se realiza por grupo de recursos.

Para los dos requisitos de cálculo siguientes:

2. % de tickets de prioridad 1 resueltos en cuatro horas para clientes CM3 en cada ubicación.
3. % de tickets de prioridad 1 cerrados en un día para clientes CM3 en cada ubicación.

Para estos requisitos, recopile los recursos en grupos de recursos. Esto es así porque las métricas tienen que estar en clúster ya que el cálculo es necesario para cada sitio individualmente.

Nota: Aunque las asignaciones de recursos para el modelo actual y los requisitos no sean necesarios, es importante crearlos teniendo en cuenta los requisitos futuros. Hacer esto evita las sobrecargas en el futuro desarrollo del sistema.

Selección del campo Marca de tiempo

Como se ha mencionado previamente, la marca de tiempo es muy importante para la forma en la que el motor de correlación trata los Eventos. Las métricas siempre calculan el nivel de servicio por período de tiempo y los eventos que se procesan en ese período de tiempo son aquellos cuya marca de tiempo está dentro de dicho período.

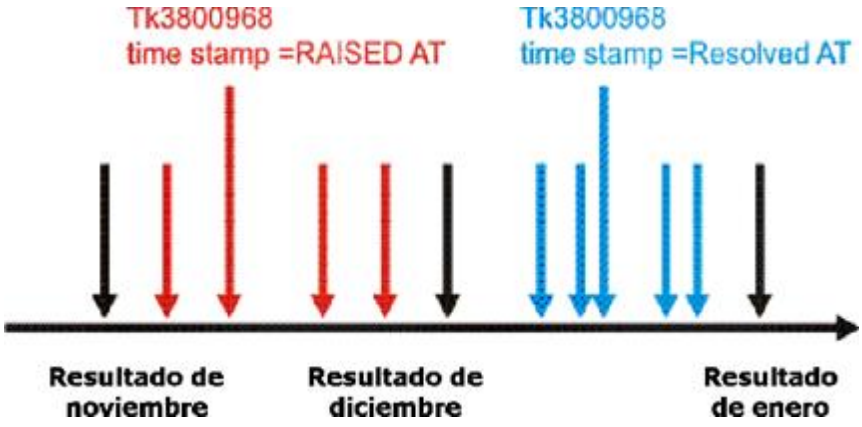
En el primer caso práctico, el origen de datos solamente tiene un campo de tiempo. Sin embargo en este caso hay tres posibles campos que se pueden establecer como la marca de tiempo. Vamos a examinar los dos primeros registros:

Nº de ticket	Prioridad del ticket	Creado el	Cerrado el	Resuelto el	Ref cliente	Ubicación
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	Ipswitch

Para calcular el tiempo de resolución, se necesitan el campo Creado el y Resuelto el. Para los objetivos del evento, es posible adjuntarle solamente una marca de tiempo. Así la otra se puede utilizar como un valor dentro de los campos de valor.

Si el valor de Creado el se utiliza como la marca de tiempo, el ticket se incluye en los resultados de diciembre. Si el valor de Resuelto el se utiliza como la marca de tiempo, el ticket se incluye en los resultados de enero. Las dos opciones son válidas. La selección sólo tiene que cumplir las expectativas en cuanto al lugar en el que deberían aparecer los tickets en los informes.

Es un punto muy importante que hay que considerar durante la implementación, ya que determina cuándo se utilizan los eventos para los cálculos. Por ejemplo, si un ticket se crea en noviembre, pero no se cierra hasta diciembre, ¿cuándo debería influir en el resultado del acuerdo de nivel de servicio? ¿Va a los datos de noviembre o de diciembre?



En este caso, como el ticket se comunica al origen de datos solamente después de cerrarse, los datos se pueden capturar sólo después de que el ticket se cierre. Normalmente la fecha de cierre es posterior a las fechas de creación y resolución. En un caso en el que la fecha de creación se elige como la marca de tiempo, debería procesarse como parte de los resultados de diciembre. La fecha de cierre fue en enero, lo que significa que diciembre ya había pasado cuando este ticket se comunicó. Por consiguiente, los resultados de diciembre ya se habían publicado. El motor de correlación se da cuenta de que el evento ocurrió con posterioridad porque la marca de tiempo pertenece a diciembre y activa el recálculo. Por lo tanto, los resultados de diciembre cambian retroactivamente.

Estas consecuencias deben entenderse completamente para poder definir qué campo de tiempo se deberá elegir como una marca de tiempo. Normalmente, la fecha de cierre se elige para tener informes finales que no cambien retroactivamente.

Por otra parte, al utilizar la fecha de cierre como la marca de tiempo se retrasa la introducción del ticket en los cálculos. Un ticket que se ha resuelto se puede cerrar solamente en un tiempo sustancial más tarde.

Sea consciente sin embargo, que este uso de la fecha de cierre podría activar también un proceso en la organización que acelerase el tiempo de cierre de los tickets.

Por tanto, la sugerencia final es:

Nombre del evento	Ticket de prioridad 1 (se puede definir también para otras prioridades si es obligatorio)
Comportamiento	Comunicar cuando el ticket se cierra
Campo Marca de tiempo	Cerrado el
Campo Recurso	campo cliente+campo de sitio
Campo Tipo de evento	Prioridad
Campos de datos	Todo
Atributo Tipo de recurso	Sitio de la parte contratante
Asignación a parte contratante	Cada sitio se asigna a la parte contratante relevante
Asignación a servidor	Igual que la anterior
Asignación a grupo de recursos	Se crea grupo de recursos para que cada parte contratante permita el agrupamiento en clúster

Registro de	Para las métricas en clúster, el registro se realiza por recurso y métrica y está adjunto a un grupo de recursos llamado sitios de cliente CM3. Para las métricas de tiempo cerradas, el registro es por parte contratante y servicio.

Uso del ejemplo de atributos personalizados

El siguiente caso práctico presenta un ejemplo de varios destinos dinámicos.

Caso práctico 9: Varios estilos dinámicos

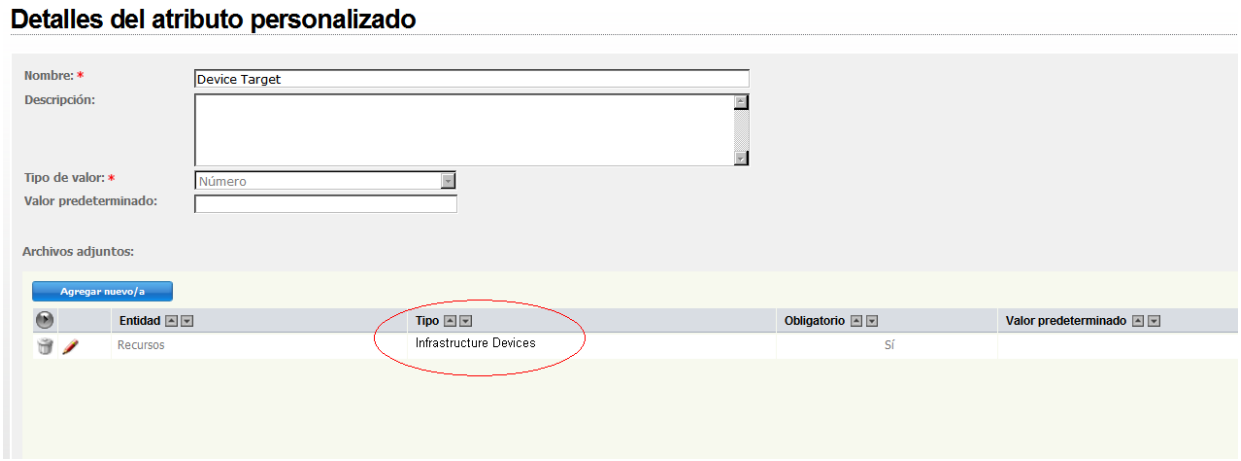
Piense en un escenario de ejemplo donde todos los dispositivos de infraestructura de hardware del entorno de un cliente tienen destinos individuales asignados para sus requisitos de disponibilidad. Mediante el enfoque de modelado estándar esto sería algo bastante difícil de conseguir e implicaría el tener que realizar mucha agrupación lógica para los dispositivos y una gestión mediante el modelo de recurso. Como complejidad añadida, los destinos para estos dispositivos pueden cambiar con el tiempo. Un script de traducción actualiza estos valores de destino de CA Business Service Insight a medida que los detalles se almacenan en un CMDB externo (consulte Ejemplos de prácticas recomendadas de scripts de traducción (en la página 302) para ver un ejemplo de script de traducción).

En este caso la métrica podría ser la siguiente:

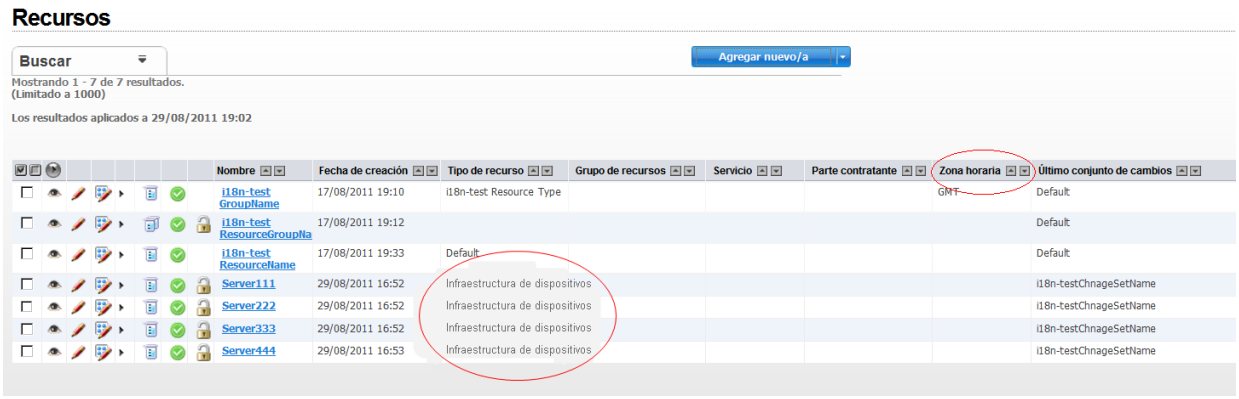
% de disponibilidad por dispositivo de hardware.

Una forma de modelar eficazmente esto es utilizar la función Atributos personalizados junto con una de las otras funciones clave, Destino dinámico. Las dos se pueden utilizar con una métrica en clúster para lograr los resultados deseados. Al agregar el destino del nivel de servicio directamente al recurso se permite que la lógica de negocios compare el nivel de servicio de cada recurso (dispositivo de hardware) con su propio destino. Una métrica en clúster proporciona el cumplimiento del servicio individual para cada pieza de hardware mediante una sola métrica.

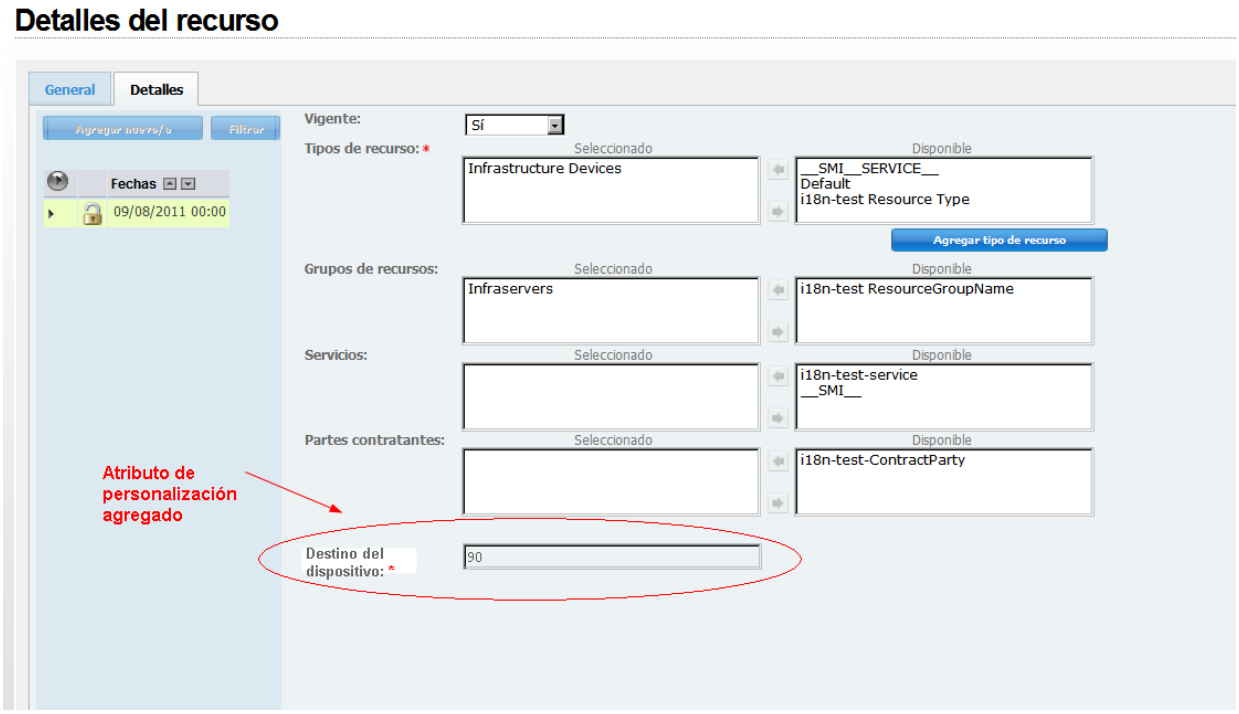
Por lo tanto, es necesario crear primero el atributo personalizado agregándolo al tipo de recurso de estos dispositivos (donde todos los dispositivos son un recurso del tipo Dispositivo de infraestructura). El atributo personalizado creado se llama destino del dispositivo y se puede agregar del menú en *Catálogo de servicios > Atributos personalizados*. Tenga en cuenta que al crear el atributo personalizado, debe conectarlo con los tipos de recurso que lo necesitan.



Ahora, al ver los recursos en el sistema, puede ver que el nuevo atributo personalizado está disponible para el tipo de recurso con el que se vinculó.



Y los recursos individuales tienen un nuevo campo que se puede actualizar.



En este ejemplo, este campo se insertaría/actualizaría normalmente mediante el script de traducción.

Ahora que cada uno de los recursos tiene un destino especificado, puede desarrollar la lógica para realizar el cálculo obligatorio (después de confirmar los cambios del recurso). El siguiente código de muestra ilustra cómo extraer el valor del atributo personalizado del recurso (en negrita).

```
Opción explícita

Dim TotalTime
Dim OutageTime
Dim PeriodStart

Sub OnRegistration(dispatcher)
   dispatcher.RegisterByResource "OnDeviceOutageEvent", "DeviceOutageEvent", _
      Context.ClusterItem
End Sub

Sub OnLoad(TIME)
   TotalTime = 0
   OutageTime = 0
End Sub

Sub OnPeriodStart(TIME)
   TotalTime = 0
   OutageTime = 0
   PeriodStart = TIME
End Sub

Sub OnPeriodEnd(TIME, isComplete)
   TotalTime = Tools.NetTime(PeriodStart, TIME)
End Sub

Sub OnDeviceOutageEvent(eventDetails)
   OutageTime = OutageTime + Tools.NetTime (eventDetails ("OutageStartTime"), _
      eventDetails ("OutageEndTime"))
End Sub

Function Target
   Target = eventDetails.CustomAttribute ("DeviceTarget")
End Function

Resultado de la función
   If TotalTime > 0 Then
      Result = (TotalTime - OutageTime) / TotalTime
   Else
      Result = Null
   End If
End Function
```

Ejemplos de scripts de traducción

Los casos prácticos siguientes incluyen un ejemplo de traducción automática básica y actualizaciones de modelo de recurso.

Caso práctico 10: Traducción automática básica

El ejemplo de script de traducción que aquí se proporciona es un caso bastante simplista de procesamiento de entradas pendientes en la pantalla Entradas de traducción.

El controlador OnTranslationEvent realiza una sencilla comprobación en el primer carácter del recurso y efectúa una acción según el valor: si es "a", la entrada de traducción se establece como "ignore"; si es "b", se suprime; si es "c" se traducirá, o de lo contrario no se modifica para que se traduzca manualmente. Fíjese en que los contadores en el código siguen las acciones que se realizan durante la ejecución del script. Esto es muy útil para la depuración o documentación de las ejecuciones de script cada vez que se ejecuta, especialmente cuando el script se automatiza. Es muy importante recordar el comando *Tools.Commit* al final de la función, ya que sin él ninguno de los cambios realizados por el script se guardarán en la base de datos.

La función TranslateResource() convocada simplemente realiza una comprobación para ver si existe en el sistema un recurso con el mismo nombre que el que ha pasado a él mediante la entrada de traducción pendiente (junto con el prefijo de E2E). Si no existe, el script agrega este recurso y, a continuación, realiza la traducción. Si ya existe, se crea una entrada de traducción de la cadena del recurso al recurso de CA Business Service Insight existente.

La función final de resultado del script envía una descripción de las tareas realizadas por el script. El código es el siguiente:

```
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


Resultado de la función
   Result = translated & "entries were translated, "& _
      ignored & "entries were ignored," & _
      deleted & "entries were deleted and "& _
      manually & "entries were set to manually update!"
End Function
```

Caso práctico 11: Actualizaciones del modelo de recurso

Otro uso de scripts de traducción es actualizar el modelo de recurso de CA Business Service Insight con valores tomados de un origen de datos externo. Aunque ya no está estrictamente relacionado con la traducción, esta muestra es una funcionalidad muy valiosa para las actualizaciones automáticas del sistema.

En la sección anterior sobre atributos personalizados se describió un escenario donde los dispositivos de infraestructura de hardware de una organización se almacenan en un CMDB externo, junto con un destino de disponibilidad esperado para cada dispositivo. Esta información tiene que replicarse en el modelo de recurso de CA Business Service Insight para mantener la asignación de la infraestructura (y los destinos del dispositivo) actualizados.

En este caso, el script es obligatorio para realizar las tareas siguientes:

- Agregar algunos dispositivos de hardware nuevos que no existen actualmente en el sistema.
- Actualizar el destino de disponibilidad esperado de los dispositivos que ya existen (si el destino cambia).

El script es el siguiente:

```
Opción explícita

'*********************************************************************
'Variables globales y constantes
'*********************************************************************

Dim added
Dim updated
Dim ChangeSetName

added = 0
updated = 0

Const RESOURCE_TYPE = "Infrastructure Devices"
Const RESOURCE_GROUP = "InfraServers"
Const CHANGESET_NAME = "Infrastructure Devices"
Const CHANGESET_EFFECTIVE_DATE = "01/01/2007"

'*********************************************************************
'Sub OnLoad :
'Preparación de las entidades de infraestructura de la fundación
'*********************************************************************

Sub OnLoad()
   Tools.log "Translation Script : In OnLoad procedure", "D"
```

```
   'Buscar la versión de recurso preliminar existente
   Dim ChangeSetMap
   Set ChangeSetMap=Tools.SearchChangeSets(CHANGESET_EFFECTIVE_DATE,
CHANGESET_NAME)

   'Si no hay ninguna versión existente, crear una nueva versión
   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")
      'Agregar recursos a la última versión si todavía no existe
      If Not Tools.IsResourceExists(resource) Then
         resource_id = Tools.AddResource(resource, CHANGESET_NAME,
"Infrastructure Device", RESOURCE_TYPE, RESOURCE_GROUP, Null, Null)
         Tools.UpdateResourcesCustomAttribute resource, CHANGESET_NAME,
"DeviceTarget", deviceTarget
         Tools.Log "AddingResource: " & resource & " with target: " &
deviceTarget & " ; assigned ID= " & resource_id
         added = added + 1
      Else
         Set resMap = Tools.GetResourceDetails(resource,CHANGESET_NAME, False)
         Set custAttrib = resMap("CustomAttributes")
         custAttribValue =
CDbl(custAttrib("DeviceTarget")("CustomAttributeValue"))
         If CDbl(deviceTarget) <> custAttribValue Then
            Tools.UpdateResourcesCustomAttribute resource, CHANGESET_NAME,
"DeviceTarget", deviceTarget
            Tools.Log "Updating Resource target for : " & resource & " from: " &
deviceTarget & " to " & custAttribValue
            updated = updated + 1
```

```
            End If
         End If

         Tools.Commit
         rs.MoveNext
      Bucle

      rs.Close
      conn.Close
      Set rs = Nothing
      Set conn = Nothing
End Sub

Resultado de la función
      ' Confirmar la transacción
      Tools.CommitChangeSets CHANGESET_NAME
      Result = added & " resources added and " & updated & " resources updated."
End Function

'*******************************************************************************
```

Ejemplos de scripting de lógica de negocios

A continuación se presentan varias directrices generales para el scripting de lógica de negocios:

Variables globales

- Asegúrese de inicializar los valores globales que declare. El mecanismo de estado de PSL no puede guardar variables con valores nulos.

Codificación general.

- Antes de utilizar uno de los objetos clasificados a continuación, verifique que existe (mediante un método de IsExist apropiado):
  - Todos los tipos de parámetros (p. ej., tabla, lista, etc.)
  - Atributo personalizado
  - Recurso
- Asegúrese de que proporciona un módulo de lógica de negocios con todos los parámetros que requiere.
- Antes de cambiar el nombre de un recurso verifique la métrica que lo está utilizando y actualícelo como corresponda.

Asignaciones

- Uso de asignaciones en métricas en clúster: las asignaciones requieren mayor esfuerzo de cálculo por parte del motor; hay que tener en cuenta que cuando se tiene un agrupamiento en clúster en una métrica se multiplica el esfuerzo por el número de elementos de clúster.
- Cuando vaya a utilizar asignaciones globales grandes en la lógica de negocios para métricas en clúster, hágalo con sumo cuidado. Mientras el motor calcula una métrica en clúster, está ocupado cargando las variables globales desde los estados anteriores para cada elemento en el clúster por separado.
- Asegúrese de borrar las asignaciones y los vectores después de haber finalizado con ellos
- Si es necesario utilizar asignaciones grandes, asegúrese de que gestiona la asignación de forma eficaz dividiéndola en intervalos lógicos.

Caso práctico 12: Uso de la lógica de contador para calcular el número de errores

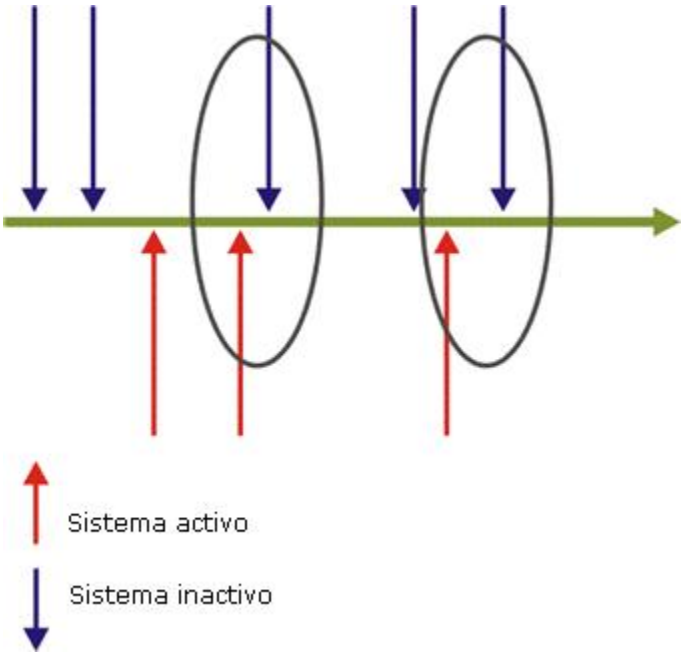
El ejemplo siguiente calcula el número de errores que ocurrieron dentro de un período de cálculo determinado. Esta fórmula y los métodos que se utilizan para implementarla se pueden tomar como ejemplo de una fórmula que es necesaria siempre que se debe realizar un cálculo.

Se utilizan los siguientes supuestos de cálculo:

- Eventos de entrada:
  - Evento de disponibilidad, estado (0/1)
  - El evento de disponibilidad llega cada pocos minutos. No se puede suponer la frecuencia de los eventos (el evento puede ocurrir cada cinco minutos o una vez cada hora) y puede haber también eventos redundantes (por ejemplo: 0, 0, 0, 1, 1, 0, etc.).
- Ranura de tiempo: horarios comerciales; no se consideran los errores ocurridos fuera de la ranura de tiempo.
- El resultado obligatorio es el recuento de los errores que han ocurrido durante el período.

Para concordar los errores que se han producido durante el período de cálculo es necesario almacenar una variable de contador periódica y también una variable que guarda el estado del sistema. Como se puede recibir información redundante del evento (es decir, un evento "activo" seguido por otro evento "activo"), también es necesario contar el número de ubicaciones en las que hubo un cambio del estado del sistema de "activo" a "inactivo" y no hay que contar todas las veces que un evento "inactivo" se recibe, pues puede ser un evento redundante que representa un error que ya se ha contado.

La siguiente figura describe gráficamente los tiempos activos e inactivos del sistema de recuento.



Hay que tener en cuenta algunos puntos importantes:

- **Constantes**: se recomienda el uso de definiciones constantes en lugar de valores constantes dentro del código. De esta manera, si el valor necesita cambiar, es fácil cambiar el valor en la definición constante solamente y no tener que buscarlo en todo el código. Además, es fácil cambiarlo a un uso de parámetro si es necesario. En el ejemplo anterior, los valores que representan el estado del sistema recibido en el evento se definen como constantes, para hacer que el código sea más comprensible y, además, para distinguir cuándo el número cero se utiliza como un número para fines de recuento y cuándo representa un estado del sistema.
- **Variables globales**:
  - **Contador**: la definición de la variable de contador es una definición global. En la fórmula se declara en la parte superior del código y fuera de subrutinas o funciones. En este caso, es importante definir la variable del contador como una variable global. Esto permite que se pueda utilizar en varias subrutinas/funciones dentro del código y también permite que el sistema conserve su valor en todo el período de cálculo y que proporcione su resultado al final del período.

En este ejemplo la variable de contador tiene que utilizarse en tres lugares en el código:

- Para que sea inicializada al inicio de un período.
- Para que se adelante en caso de un error en el evento en el controlador de eventos.
- Para que se envíe por la función de resultado al final del período.

– **Estado previo del sistema**: esta variable conserva el estado anterior del sistema y se actualiza siempre que se recibe un nuevo evento con el estado del sistema. Esta variable también tiene que ser global porque se utiliza en varias subrutinas/funciones en el código, como:

- Para inicializarse al principio del cálculo.
- Para actualizarse cuando se recibe un nuevo evento.

- **Consideraciones de la ranura de tiempo**: para verificar si un error se produjo dentro del período de una ranura de tiempo, es posible utilizar el valor de la propiedad *context.IsWithinTimeSlot.* El contexto es un objeto global que se puede utilizar desde cualquier sitio en el código, y en este caso, se utiliza para indicar cuándo se recibe un error y si está dentro o fuera del período de ranura de tiempo. Si en la marca de tiempo del evento el indicador devuelve TRUE, entones el evento ocurre dentro de un período de ranura de tiempo; sin embargo si devuelve FALSE, indica que el evento tuvo lugar fuera de la ranura de tiempo. Según la lógica obligatoria, cualquier error que ocurre fuera de la ranura de tiempo se ignora y, por tanto, no se tiene en cuenta.
- **Variables de inicialización**:

– **Contador variable**: conserva un valor periódico y, por tanto, se inicializa en el controlador *OnPeriodstart* para asegurarse de que cada período de cálculo cuenta sus errores de forma separada. Cada período empieza con cero y envía solamente el número de errores dentro de ese período.

En los casos en los que es necesario calcular los errores acumulados dentro de cada período, (lo que significa que el resultado de cada período son todos los errores que han ocurrido hasta el final de dicho período y se incluyen todos los períodos antes de este) es necesario inicializar el contador solamente en el controlador *OnLoad* y eliminarlo del controlador *OnPeriodStart*. De esta forma, el contador continúa contando y acumulando entre los períodos como se muestra a continuación:

```
Sub OnLoad(time)
    FingerprInted=0
End Sub
```

- **Variable del estado del sistema**: debería inicializarse una vez al inicio del cálculo y, desde entonces, actualizarse en cada evento de estado. Esta variable conserva su valor en todo el cálculo y no se encuentra limitada a un determinado período de cálculo. Debe conservar también su valor entre los períodos de cálculo. Como se desconoce el estado del sistema antes de recibir el primer evento, hay que suponerlo. El supuesto más común es que el sistema está "activo". Este supuesto debe aceptarse y comprobarse, ya que podría llevar a lo siguiente:

  Si cuando el cálculo se inició el estado del sistema era de hecho "inactivo" y el primer evento llega para indicar este estado "inactivo", esto se contará como un error porque el estado supuesto fue "activo". Este error se contará hacia el primer período de cálculo aunque no ocurriera necesariamente durante dicho período.

```
Opción explícita

'Definiciones de constantes
Const UP=1
Const DOWN=0

'Variable global para errores de recuento
Dim FingerprInted
Dim SystemStatus

Sub OnRegistration(dispatcher)
   ' Los parámetros del método son: <nombre del procedimiento>, <Tipo de evento>
   dispatcher.RegisterByContractPartyAndService
"OnAvailabilityEvent","AvailabilityEvent"
End Sub

Sub OnLoad(time)
   SystemStatus = UP 'asume el primer estado del sistema
End Sub

Sub OnPeriodStart(time)
   FingerprInted = 0
End Sub

Sub OnAvailabilityEvent(eventDetails)
   ' Si es un error y está dentro de la ranura de tiempo, se contará
   If context.IsWithinTimeSlot and SystemStatus=UP and _
         eventDetails("Status")=DOWN Then
      FingerprInted = FingerprInted + 1
   End If
   ' actualiza el estado del sistema para la comparación siguiente
   SystemStatus = eventDetails("Status")
End Sub
```

```
Resultado de la función
   Result = FingerprInted
End Function
```

Caso práctico 13: Tratamiento de un grupo de componente dinámico

A menudo, es necesario almacenar valores para un grupo de recursos en los que los miembros de este grupo pueden ser dinámicos y cambiar durante el período de cálculo. En el cálculo del ejemplo de requisito siguiente, es necesario realizar un cálculo intermedio en cada uno de los recursos para alcanzar el resultado global final.

A continuación, se presentan unos cuantos ejemplos:

- **Incidentes en el sitio**: calcula el porcentaje de sitios con incidentes que tardan más de X veces en resolverse o cuenta los sitios que tuvieron incidentes con tiempos medios de resolución superiores a X.

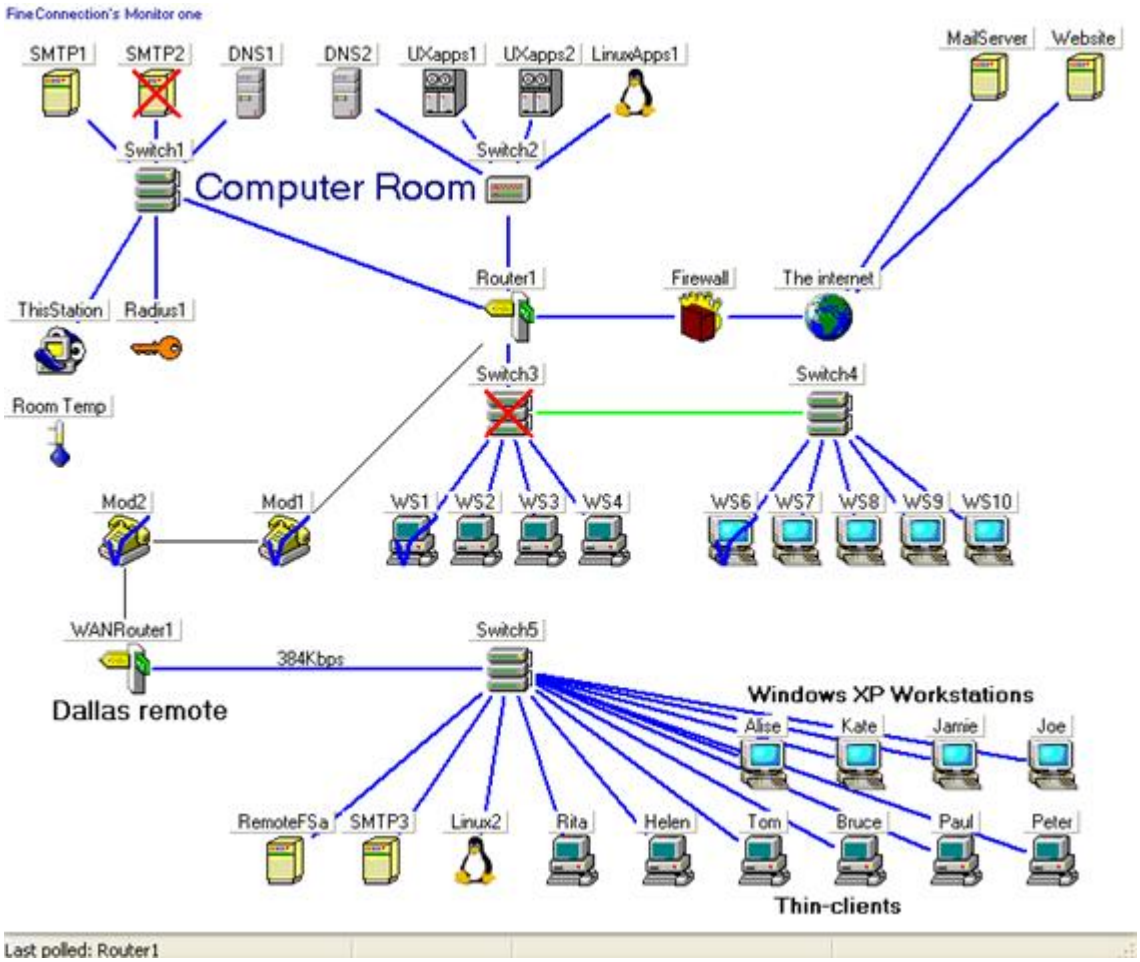
  En estos ejemplos, los sitios son recursos que tienen incidentes asociados con ellos.

- **Disponibilidad del servidor**: cuenta el número de servidores cuyo tiempo de disponibilidad fue superior a X%.

  Un servidor es un recurso para el cual hay que evaluar el porcentaje de disponibilidad.

- **Tipos de transacciones**: calcula el porcentaje de los tipos de transacciones que han producido un error más de X veces.

  En este caso, un tipo de transacción es un recurso que tiene eventos de error asociados. Para cada tipo de transacción, se guarda un contador de error como resultado intermedio y se cuentan el número de tipos de transacciones diferentes que tuvieron más de X errores.


FineConnection's Monitor one
SMTP1
SMTP2
DNS1
DNS2
UXapps1
UXapps2
LinuxApps1
MailServer
Website
Switch1
Switch2
Computer Room
Router1
Firewall
The internet
ThisStation
Radius1
Switch3
Switch4
Room Temp
Mod2
Mod1
WS1
WS2
WS3
WS4
WS6
WS7
WS8
WS9
WS10
WANRouter1
384Kbps
Switch5
Dallas remote
Windows XP Workstations
Alise
Kate
Jamie
Joe
RemoteFSa
SMTP3
Linux2
Rita
Helen
Tom
Bruce
Paul
Peter
Thin-clients
Last polled: Router1

Ejemplo:

Para el siguiente requisito de cálculo:

Calcule el porcentaje de disponibilidad de un sistema que se compone de un clúster de servidores. El sistema se considera que está disponible solamente cuando todos los servidores subyacentes están disponibles.

El diseño de la solución se implementará de la siguiente manera:

Los recursos en clúster subyacentes evalúan la disponibilidad del sistema. En este caso, es necesario gestionar y almacenar el estado de todos los recursos agrupados en clúster para evaluar el estado del sistema en cada momento.

Para ello, es preciso que la fórmula tenga una asignación (la asignación ServerAvailabilityIndicators en el código de ejemplo siguiente) que cuente con una entrada para cada uno de los recursos controlados. Como todos los objetos de asignación necesitan un campo clave para referenciar el valor asociado, para esta asignación la clave será el indicador de recurso (que es el ID del recurso) y el valor, el estado del componente. Cuando el estado de un componente cambia, la entrada relevante de dicha asignación debería cambiarse. Con cada evento de disponibilidad, la fórmula actualizará el estado del recurso relevante en la asignación y volverá a evaluar el estado del sistema como corresponda.

Dado que los recursos controlados pueden cambiar (algunos podrían eliminarse y otros podrían agregarse durante el período de cálculo), hay que gestionar esto dentro de la fórmula; para ello, se debe agregar una función que identifique el cambio y actualizar la asignación del componente controlado agregando una nueva entrada para un nuevo componente o suprimiendo una entrada si se eliminó un componente.

OnRegistration es el controlador de eventos que gestiona un evento de registro que se activa cuando se produce un cambio en los recursos controlados. Dicho cambio puede ocurrir como resultado de la confirmación de un recurso (o conjunto de cambios), que puede provocar cambios en los recursos incluidos en el cálculo, según el método de registro de la fórmula.

Durante cada registro es necesario actualizar la asignación que conserva los estados de los recursos con los cambios obligatorios. Esto quiere decir que hay que comparar la asignación utilizada para conservar los estados del recurso con la asignación que contiene los recursos en el momento de ejecutar el registro (basado en el método de registro), e incluir todos los recursos que se agregaron o suprimir los recursos que se eliminaron.

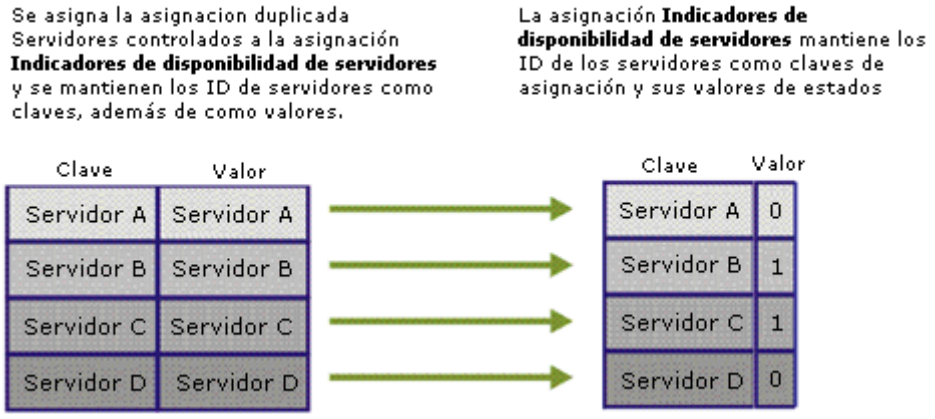
El procedimiento de OnRegistration debe, por tanto, ejecutar una función que compara los recursos controlados con los nuevos recursos adjudicados para estructurar los recursos controlados como corresponda.

El objeto de contexto tiene un conjunto de métodos que se asemejan a los métodos de registro. Estos devuelven una asignación con todos los recursos que son parte de los recursos según el método de registro.

En el ejemplo siguiente el registro de la fórmula es ByContractParty y, por consiguiente, Context.ResourcesOfContractParty utiliza el mismo método. Devuelve una asignación con todos los recursos que forman parte de este conjunto en el momento del registro.

Para comparar las dos asignaciones, es necesario iterar las asignaciones simultáneamente. Para iterar las asignaciones se utiliza la declaración "For each". Esta declaración permite procesar una iteración en los valores de una asignación y, por lo tanto, se necesita otra asignación como duplicado de la asignación de estados para poder procesar una iteración en los recursos y no en los valores de estado. Esta declaración itera los valores de una asignación y no sus claves. Por consiguiente, se necesita una asignación extra que tenga tanto los ID como una clave y un valor. Además, hay que mantener la asignación del duplicado para reflejar continuamente la asignación de estados, y que así se tenga siempre el mismo conjunto de recursos. Esto significa que cuando la asignación de estados se actualiza, la asignación del duplicado se deberá actualizar también.

La siguiente figura muestra el concepto de las asignaciones de duplicación.



Ejemplo:

```
Dim ServerAvailabilityIndicators
Dim MonitoredServers
Set ServerAvailabilityIndicators=CreateObject("SlalomMap.Map")
Set MonitoredServers=CreateObject("SlalomMap.Map")

Sub OnRegistration(dispatcher)
   dispatcher.RegisterByContractParty "OnAvailabilityEvent",_
      "Availability Event", "SAP Production Server"
   Dim AllocatedServers
   Set AllocatedServers = Context.ResourcesOfContractParty("SAP Production
Server")
   UpdateMonitoredServers AllocatedServers
End Sub

Sub UpdateMonitoredServers(allocatedServers)
   Dim Server
   For Each Server In allocatedServers
      If Not MonitoredServers.Exist(Server) Then
         MonitoredServers(Server) = Server
         ServerAvailabilityIndicators(Server)=True
      End If
   Next

   For Each Server In MonitoredServers
      If Not allocatedServers.Exist(Server) Then
         MonitoredServers.Erase Server
         ServerAvailabilityIndicators.Erase Server
      End If
   Next
End Sub
```

Ejemplo:

La siguiente función prueba cómo la asignación del duplicado se usa para procesar una iteración en la asignación de estados cuando es necesario evaluar todo el sistema según el estado de cada uno de los recursos controlados.

En este ejemplo, se considera que el sistema está disponible si todos los recursos están disponibles. Un solo componente inactivo ya es suficiente para considerar el sistema inactivo:

```
Function SystemAvailability
   Dim Server
   For Each Server In MonitoredServers
      If ServerAvailabilityIndicators(Server) = DOWN then
         SystemAvailability=DOWN
         Exit Function
      End if
   Next
```

```
End Function
```

En el siguiente ejemplo de patrón de diseño se describe un ejemplo de lógica de negocios completa con tratamiento de recursos dinámicos.

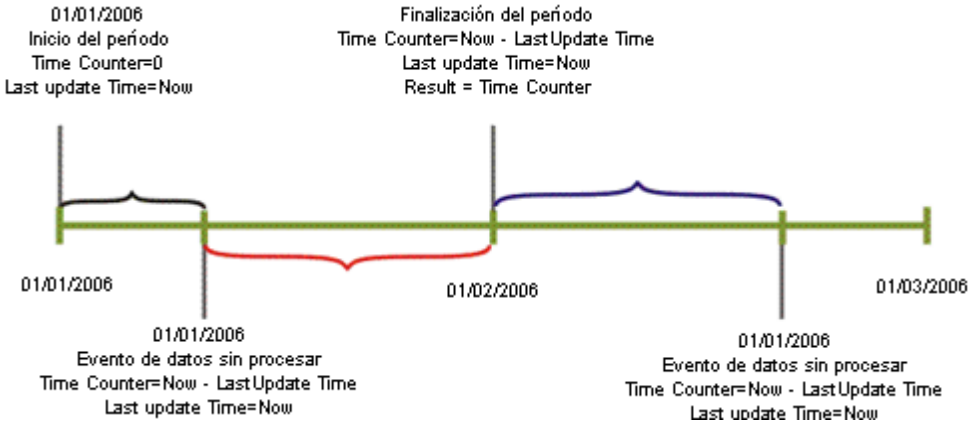
Caso práctico 14: Tratamiento del reloj de acumulación de tiempo

El patrón de diseño descrito en esta sección es apropiado cuando el resultado obligatorio es una función de un período de tiempo que ha pasado entre eventos, por ejemplo:

- Tiempo disponible: calculado como el tiempo transcurrido entre un error y otro.
- Tiempo de resolución: calculado como el tiempo entre la abertura del incidente y el momento de cierre.

Para acumular tiempo es necesario asignar una variable en la que se pueda acumular tiempo (en segundos) e implementar una función que comprueba las dos condiciones y el tiempo acumulado desde que la última vez que se actualizó la hora. Esta función se ejecuta para todos los eventos recibidos para la fórmula.

La siguiente figura describe el tratamiento del reloj de acumulación de tiempo.



La variable LastUpdateTime se almacena la última vez que se realizó la actualización, independientemente de si el contador de tiempo se actualizó o no. La función mantiene la condición que determina si el tiempo debe actualizarse y acumularse. Por ejemplo, el tiempo no se debería tener en cuenta si se supera el período de ranura de tiempo, el estado del sistema era inactivo o el incidente tuvo un estado pendiente.

Aunque la situación detallada aquí a menudo utiliza la función Tools.NetTime() para calcular duraciones, puede haber casos en los que es preferible utilizar la función estándar VB DateDiff().

La función Tools.NetTime incurre en una sobrecarga de comprobación de la ranura de tiempo cada vez que se utiliza. Se recomienda evitar utilizar NetTime en los procedimientos de eventos de datos, ya que estos procedimientos se llaman para cualquier nuevo evento entrante y, por lo tanto, invocan la llamada NetTime. Si la ranura de tiempo es 24/7, es recomendable utilizar la función DateDiff para evitar la sobrecarga de comprobación de la ranura de tiempo.

***Ejemplo 1*:**

La siguiente rutina de "contadores de actualización" acumula el período total de tiempo en la variable PeriodNetTime. En AvailabilityTime la rutina acumula el tiempo que el estado del sistema estuvo activo, lo que significa que el sistema estaba disponible.

El objeto Tools contiene el método NetTime, NetTime(beginTime, endTime)0. Este método devuelve el número de segundos entre beginTime y endTime que están dentro de la ranura de tiempo de la métrica actual. Cualquier momento entre estas dos variables forma parte de una ranura de tiempo que se excluye, por consiguiente, se utiliza muy a menudo para cálculos de duración en los que se utiliza una ranura de tiempo. (Por ejemplo: para los tickets de prioridad 1 resueltos en cuatro horas comerciales, donde incluso un ticket creado al final de un día laboral que no se puede resolver hasta la mañana siguiente está dentro del acuerdo de nivel de servicio debido a las horas de la ranura de tiempo que se excluyen.)

```
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
```

***Ejemplo 2*:**

El ejemplo siguiente calcula la disponibilidad de la aplicación al tratar los eventos de actividad e inactividad de varios componentes esenciales, así como los eventos de mantenimiento de dichos componentes. Si se está realizando el mantenimiento de todos los componentes, el tiempo no se considera como el tiempo previsto de disponibilidad.

La subrutina UpdateCounters adelanta los contadores tiempo cuando es necesario y se ejecuta con todos los eventos recibidos para la fórmula (evento de datos sin procesar/evento de motor). También actualiza el tiempo disponible previsto en los casos en los que el tiempo está dentro del período de ranura de tiempo y los componentes no están dentro de un período de tiempo de inactividad esperado. La fórmula actualiza el tiempo de disponibilidad real solamente cuando también tiene un estado "activo" del sistema.

DateDiff es una función de VB estándar que devuelve el tiempo entre dos fechas, pero no excluye ninguna información de ranura de tiempo.

```
'Imponer la declaración de la variable
Opción explícita

'Variables globales
Dim ExpectedAvailabilityTime
Dim ActualAvailabilityTime
Dim LastUpdateTime
Dim AvailabilityIndicators
Dim MonitoredComponents
Dim DowntimeStatuses

'Creación de objetos de asignación
Set AvailabilityIndicators=CreateObject("SlalomMap.Map")
Set MonitoredComponents=CreateObject("SlalomMap.Map")
Set DowntimeStatuses=CreateObject("SlalomMap.Map")

'Después de haberse cargado y siempre que ocurre un cambio de infraestructura
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")
```

```
   'Hay que asegurarse de que la fórmula se monitoriza solamente cuando es
relevante y todos los recursos son relevantes
   UpdateMonitoredComponents AllocatedComponents
End Sub

Sub OnLoad(time)
   'Cuando el sistema está activo por primera vez: asumir que la disponibilidad
es correcta
   LastUpdateTime = time
End Sub

Sub OnPeriodStart(time)
   'inicialización de contadores para renovar el cálculo periódico
   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
   'escribir el estado de disponibilidad de los recursos sobre los que se informa
   AvailabilityIndicators(eventDetails.ResourceId) = _
      AvailabilityIndicators(eventDetails.ResourceId)+1
End Sub

Sub OnComponentUpEvent(eventDetails)
   UpdateCounters eventDetails.Time
   'escribir el estado de disponibilidad de los recursos sobre los que se informa
   AvailabilityIndicators(eventDetails.ResourceId)= _
      AvailabilityIndicators(eventDetails.ResourceId)-1
End Sub

Sub OnMaintenanceStartEvent(eventDetails)
   UpdateCounters eventDetails.Time
   'escribir el estado de disponibilidad de los recursos sobre los que se informa
   DowntimeStatuses(eventDetails.ResourceId)= _
      DowntimeStatuses(eventDetails.ResourceId)+1
End Sub

Sub OnMaintenanceEndEvent(eventDetails)
   UpdateCounters eventDetails.Time
   'escribir el estado de disponibilidad de los recursos sobre los que se informa
```

```
   DowntimeStatuses(eventDetails.ResourceId)= _
      DowntimeStatuses(eventDetails.ResourceId)-1
End Sub

Sub OnPeriodEnd(time,isComplete)
   UpdateCounters time
End Sub

Resultado de la función
   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
      'actualizar contador de segundos en el período (cuando se espera la
disponibilidad)
      ExpectedAvailabilityTime = ExpectedAvailabilityTime +
DateDiff("s",LastUpdateTime,time)
      If SystemIsAvailable Then
         'actualizar contador de segundos de disponibilidad
         ActualAvailabilityTime = ActualAvailabilityTime +
DateDiff("s",LastUpdateTime,time)
      End If
   End If
   LastUpdateTime=time
End Sub

Sub UpdateMonitoredComponents(allocatedComponents)
   Dim Component
   'añadir a la asignación de componentes controlados todos los nuevos
componentes que hay que monitorizar
   For Each Component In allocatedComponents
      If Not MonitoredComponents.Exist(Component) Then
         MonitoredComponents(Component) = Component
         AvailabilityIndicators(Component) = 0
         DowntimeStatuses(Component) = 0
      End If
   Next

   'eliminar de la asignación de componentes controlados todos los componentes
que ya no son relevantes
   For Each Component In MonitoredComponents
      If Not allocatedComponents.Exist(Component) Then
         MonitoredComponents.Erase Component
         AvailabilityIndicators.Erase Component
```

```
         DowntimeStatuses.Erase Component
      End If
   Next
End Sub

Function SystemIsAvailable
   Dim SystemAvailability
   SystemAvailability = True

   Dim Component
   Dim ComponentAvailability
   For Each Component In MonitoredComponents
      ComponentAvailability = AvailabilityIndicators(Component) = 0 _
         Or DowntimeStatuses(Component) > 0
      'la disponibilidad del sistema se evalúa con la disponibilidad
      SystemAvailability = SystemAvailability And ComponentAvailability
   Next
   SystemIsAvailable = SystemAvailability
End Function

Function AllComponentsAreInPlannedDowntime
   Dim ComponentsInPlannedDowntime
   ComponentsInPlannedDowntime = 0
   Dim Component
   For Each Component In MonitoredComponents
      If DowntimeStatuses(Component) > 0 Then
         ComponentsInPlannedDowntime = ComponentsInPlannedDowntime + 1
      End If
   Next

   If ComponentsInPlannedDowntime = MonitoredComponents.Count Then
      AllComponentsAreInPlannedDowntime = True
   Else
      AllComponentsAreInPlannedDowntime = False
   End If
End Function
```

Ejemplos de escritura de lógica de negocios vigente

En los temas siguientes se ofrecen recomendaciones para escribir de forma eficaz lógicas de negocios:

- **Métricas en clúster**
  - Al evaluar el volumen de sistema, cuente una métrica en clúster como el número de elementos en la métrica y recuerde que todo se multiplica.
  - El recálculo de un elemento de clúster implica el recálculo de todo el clúster. Por tanto, recuerde tenerlo en cuenta cuando planifique el agrupamiento en clúster, la forma en la que se configuran los adaptadores y al cambiar la estructura de recursos.
  - Los mismos eventos de datos sin procesar que van a muchos elementos de clúster tienen un alto coste de rendimiento (conmutación de contexto).
- **Variables globales**
  - Obtenga valores de atributos y parámetros en cada lugar del código que necesitan. Evite guardarlos en una variable global, especialmente en métricas en clúster (aumenta el tamaño de los "estados").
  - Evite la lógica que procesa grandes asignaciones. En cambio, procese cada evento en el método OnXXEvent
  - Elimine elementos de asignaciones tan pronto como sea posible. Por ejemplo, cuando se cierra un ticket y al final del período.
- **Patrones de diseño**

  El paquete de contenido predeterminado contiene varios patrones de diseño para escenarios comunes. Utilizar estos patrones de diseño puede mejorar el rendimiento.
- **Funcionalidad incorporada**

  ACE tiene herramientas y funcionalidad incorporada para diversas finalidades, como las siguientes:
  - Funcionalidad de ranura de tiempo
    - NetTime
    - IsWithinTimeslot
  - Tiempo de eventos
    - TimeOfLastEvent
    - TimeOfLastEventHandler

  - Context Object
    - Contiene muchos métodos sensibles al entorno
    - Utilice estos métodos y evite "ODBC seguro".

- **Resultados de lógica de negocios**

  Guarde la estructura en T_SLALOM_OUTPUTS. Esto significa que si tiene varias tablas de lógicas en T_SLALOM_OUTPUTS que tengan una estructura similar, es muy útil colocar campos lógicos similares en el mismo campo físico. Esto facilita la indexación fácil para un rendimiento de informes mejorado.

- **Reutilización de evento**

  Para utilizar cuando:

  - Varias métricas están realizando un cálculo de primera fase, que es necesario para el contrato, y envían eventos a una métrica de resumen que calcula el resultado (p.ej. cálculo financiero, indicador de clave de rendimiento de alto nivel).
  - Una única métrica está realizando una agregación preliminar en datos sin procesar y envía eventos a otras métricas. La métrica envía considerablemente menos eventos que los que obtiene o realiza cálculos significativos que, de otra forma, realizaría muchas veces.

  Para evitar cuando:

  - Aumenta de forma significativa el número de métricas.
  - Se implementan más de tres niveles.
  - La complejidad de la implementación aumenta y el mantenimiento se vuelve más difícil.

- **Recálculos**
  - Evite recálculos masivos como parte del funcionamiento normal del sistema.
  - Los motivos de los recálculos son:
    - Datos sin procesar con una marca de tiempo anterior.
    - Singularidad del evento que cambia datos sin procesar en el pasado.
    - Correcciones
    - Excepciones
    - Cambios en los módulos de lógica de negocios
    - Cambios en un contrato

        - Eventos de reutilización de eventos con una marca de tiempo anteriormente
        - Cambios en la estructura de recursos
    - Considere usar la funcionalidad de bloqueo de datos calculado
- **Módulos de lógica de negocios**
    - Los módulos de lógica de negocios se deberían escribir de forma que, una vez que se hayan revisado por completo, no tengan que modificarse.
    - La directriz es: un módulo es igual a una lógica genérica.
    - Los módulos de lógica de negocios que son muy específicos no pueden servir muchas métricas y no promueven la reutilización de código y lógica.
    - Los módulos de lógica de negocios que son demasiado genéricos son difíciles de mantener. Además, si un módulo de lógica de negocios implementa muchas lógicas complejas, una corrección en un flujo (utilizado por parte de la métrica), provoca el recálculo de todas las métricas
- **Registro**
    - Realice toda la filtración de eventos utilizando el registro. El dejar la filtración al código tiene un gran impacto en el rendimiento.
    - Simplificación
    - Para el código que no es el propio registro, utilice los métodos dispatcher.IsRunTimeMode y OnResourceStructureChange, sobre todo cuando hay muchos cambios de recursos.
    - Evite utilizar el método RegisterByEventType.
    - En los módulos de lógica de negocios, utilice un formulario genérico (por parte contratante, servicio, tipo de recurso), utilice parámetros o deje el registro que hay que realizar mediante la interfaz de usuario (opción preferida para la reutilización de eventos).

Caso práctico 15: Métricas en clúster

Normalmente, cuando se describe un trozo determinado de software, esta descripción se puede dividir en dos partes: el QUÉ y el CÓMO. Con el QUÉ queremos hacer referencia a la descripción de lo que hace ese trozo de código. El CÓMO es cómo lo hace. Hay una tendencia a concentrarse en el QUÉ e ignorar parte del CÓMO. El motivo es sencillo y, en muchos casos, está justificado. Con ello, se reduce el acoplamiento entre los componentes y no hay que preocuparse con información que muchas veces es irrelevante. Sin embargo, en muchos casos, el coste de ignorar el CÓMO supone un mal rendimiento.

Este caso práctico trata la forma en la que el motor está calculando métricas en clúster (la respuesta a la parte del CÓMO) y describe el coste de rendimiento que implican determinadas implementaciones. También trata las diversas formas de reducir este coste mediante el cambio de la implementación.

¿Qué son las métricas en clúster?

Las métricas en clúster son métricas que incluyen en su definición un determinado grupo de recursos. Este grupo se conoce como el clúster de la métrica y cada uno de los recursos de ese grupo se conoce como un elemento de clúster. Al calcular una métrica en clúster, se realiza un cálculo separado para cada uno de esos elementos de clúster. Los cálculos para cada uno de esos elementos de clúster son similares entre ellos, excepto:

- Context.ClusterItem: el valor de la propiedad Context.ClusterItem que es igual al elemento de clúster que se está calculando.
- Registrations by Cluster item: cuando una métrica se registra en los eventos por elemento de clúster, cada elemento de clúster recibe eventos de datos sin procesar brutos y eventos de reutilización que se envían a este elemento de clúster.

¿Cómo se calculan las métricas en clúster?

Lo más importante que hay que entender sobre el cálculo de una métrica en clúster es que todos los elementos de clúster se calculan en paralelo. Por paralelo no se quiere decir que se calculan por distintos subprocesos, sino que mientras se procesan los distintos eventos que los elementos de clúster deben manejar, los eventos se están procesando de forma secuencial y para cada evento los elementos de clúster relevantes se convocan y procesan este evento. Por ejemplo, hay muchos eventos que deberían tratar varios elementos de clúster. Hay dos formas de hacerlo:

Ejemplo: Opción 1

```
Para cada elemento de clúster C
     Para cada evento E que C debería manejar
          Dejar que C maneje a E
```

Ejemplo: Opción 2

```
Para cada evento E
     Para cada elemento de clúster C que maneja a E
          Dejar que C maneje a E
```

El motor maneja métricas en clúster mediante la Opción 2.

Otro punto importante que hay que entender es que un componente llamado Script Control realiza la ejecución de VBScript dentro de PslWriter. Hay solamente una sola instancia de este componente por cada métrica y esta instancia se reutiliza para el cálculo de los diversos elementos de clúster. Como los elementos de clúster se calculan paralelamente, como se ha mencionado antes, y el componente Script Control puede contener solamente los datos de un solo elemento de clúster en cada momento, tenemos que cambiar frecuentemente los datos dentro del componente Script Control.

Para explicar esto, se presenta a continuación un código falso más detallado que realiza el cálculo.

```
1-     Para cada métrica M
2-          Permitir que X sea el primer evento que no se maneja todavía por M
3-          Permitir que T sea la marca de tiempo del último estado antes de X
4-          Permitir que L sea la lista de todos los eventos registrados por M
(todos los elementos de clúster) empezando por la marca de tiempo T hasta la hora
actual
5-          Para cada evento E en L
6-               Para cada elemento de clúster C que maneja a E
7-                    Permitir que C sea el elemento de clúster que está cargado
actualmente en Script Control
8-                    Tomar los valores de las variables globales de Script
Control y almacénelas aparte para C
9-                    Tomar los valores de las variables globales guardadas
aparte para C y cárguelos en Script Control
10-                    Manejar evento E
```

Este proceso completo de encontrar algo de tiempo de un evento que todavía no se ha tenido cuenta y realizar, a continuación, el cálculo a partir de ese punto se llama recálculo. El proceso de sustituir los valores de las variables globales (pasos 8 y 9 en el código anterior) se llama conmutación de contexto.

Los dos problemas principales que se pueden ver fácilmente en el código anterior son:

- El recálculo se hace para todos los elementos de clúster juntos. A partir del momento en el que el tiempo T (paso 3 en el código anterior) se encuentra una vez, todos los elementos de clúster realizan un recálculo basándose en ese punto. Esto significa que siempre que un solo elemento de clúster tenga algún nuevo evento, todos los elementos de clúster se recalculan.
- La conmutación de contexto se realiza muy a menudo. Esto se puede ver fácilmente, ya que la conmutación de contexto (pasos 8 y 9 en el código anterior) se encuentra dentro del bucle interior.

Recálculo de métricas en clúster

- Descripción del problema

  Como ya se ha explicado, todos los elementos de clúster en una métrica en clúster se recalculan en conjunto. Esto significa que si tenemos una métrica agrupada en más de 1000 elementos de clúster y uno de ellos necesita un recálculo del último año, debido a algún nuevo evento que ha llegado, todos los 1000 elementos de clúster deben recalcularse para el último año.

- Posibles soluciones

  Las siguientes sugerencias de soluciones pueden reducir las consecuencias de este problema, pero no son siempre aplicables y cada una tiene sus propias desventajas. Lo más importante es entender el problema y su coste estimado y comparar este coste con el coste de la solución propuesta.

  - Cuando el número de elementos de clúster es pequeño, podemos considerar la opción de definir cada uno de ellos como una métrica separada. La desventaja de este enfoque es, por supuesto, el coste de mantenimiento de varias métricas, así como el hecho de que no podemos realizar un informe de la métrica completa y, a continuación, entrar en más detalle en un elemento de clúster específico
  - Cuando el número de elementos de clúster es grande pero solamente uno (o unos pocos) de ellos se recalcula frecuentemente, podemos considerar el hecho de poner este elemento de clúster en una métrica aparte y dejar todos los demás elementos de clúster en la otra métrica
  - Utilice con frecuencia un cálculo fijo para el contrato/métrica relevante para que esta métrica nunca tenga recálculos largos.
  - Realice algunos cambios en los adaptadores y los orígenes de datos para que no haya recálculos largos (es decir, no envíe eventos cuya marca de tiempo sea anterior a un mes)

Conmutación de contexto

- Descripción del problema

Como ya se ha explicado, la conmutación de contexto se hace en el bucle más interior. Dicho de otra manera, para cada evento que debería manipularse por cada elemento de clúster. Cuando una métrica recibe muchos eventos y cuando cada evento se manipula por muchos elementos de clúster, esta cantidad puede ser muy alta. Hay que añadir que la operación de conmutación de contexto es relativamente cara (en relación a la manipulación del propio evento en la lógica de negocios), con lo que tendríamos un problema.

El coste de la operación de conmutación de contexto es proporcional al tamaño de los datos que se "cambian". Los datos que cambiamos durante la conmutación de contexto son los valores de todas las variables globales en nuestra lógica de negocios (también llamada "el estado"). De esta forma, mientras más variables globales se tenga y mayor sea el tamaño de esas variables globales, más cara será la operación de conmutación de contexto.

En particular, no se recomienda utilizar las asignaciones de lógica de negocios en métricas en clúster, sobre todo si el tamaño de esas asignaciones puede ser grande.

- Posibles soluciones
  - Reducción del tiempo de cada una de las conmutaciones de contexto

    La idea es reducir el tamaño del estado (variables globales). Este enfoque se puede hacer reescribiendo la lógica de negocios para que no contenga asignaciones. Por supuesto esto no es siempre posible, pero cuando sí lo es, se recomienda.
  - Reducción de la cantidad de los conmutadores de contexto

    Cuando el clúster es pequeño es posible crear una métrica separada para cada elemento de clúster.

    Evite métricas en clúster con muchos elementos agrupados en clúster que se registran a los mismos eventos. La idea aquí es la siguiente:

    Si cada evento está manejado por un solo elemento de clúster, la cantidad de la conmutación de contexto es proporcional al número de eventos

    Si cada evento está manejado por todos los elementos de clúster, la cantidad de la conmutación de contexto es proporcional al número de eventos por el número de elementos de clúster.

Cree una métrica que no esté agrupada en clúster que calcule los resultados de todos los elementos de clúster original (que son ahora recursos sencillos y no elementos de clúster). Haga que esta métrica envíe el resultado de cada uno de los elementos de clúster como un evento. Cree otra métrica que esté agrupada en clúster y que reciba los eventos de la primera métrica e informe del valor recibido en esos eventos como resultado. La idea aquí es que una métrica que no esté agrupada en clúster manejará la gran cantidad de eventos de datos sin procesar y que la métrica en clúster manejará un solo evento por período por elemento de clúster.

Caso práctico 16: Patrones de diseño de lógica de negocios

Hay varios "patrones de diseño" que se pueden utilizar en muchas lógicas de negocios. Esos patrones de diseño se prueban y, si se usan cuando corresponde, pueden ahorrar mucho tiempo, además de, en la mayor parte de los casos, crear una lógica de negocios más eficaz. Este caso práctico se centra en un patrón de diseño.

Patrón de diseño de contadores de actualización

Este patrón de diseño es útil en casi todas las lógicas de negocios, que están pensadas para medir el tiempo entre determinados eventos. Los ejemplos de dichas lógicas de negocios son las lógicas de negocios para medir la disponibilidad, el tiempo de inactividad, el tiempo medio entre errores, el tiempo medio de restauración, el tiempo medio de respuesta, el tiempo medio de resolución, el porcentaje de componentes con una disponibilidad menor que X, el número de casos no resueltos puntualmente, etc.

El elemento común de todas esas lógicas de negocios es que su resultado depende de la marca de tiempo de los diversos eventos que reciben.

Una regla general para decidir si su lógica de negocios se puede beneficiar de este patrón de diseño sería: si las lógicas de diseño dependen de la marca de tiempo de los diversos eventos que la reciben, lo más probable es que deba utilizar este patrón de diseño.

Esqueleto de este patrón de diseño

El código de la lógica de negocios que utiliza este patrón se puede dividir en dos partes: un marco y una implementación. El marco contiene el código que en la mayoría de los caso es fijo y no cambia para las diversas lógicas de negocios. Esta parte es la misma para el cálculo de disponibilidad y el número de tickets no resueltos a tiempo. La implementación contiene código que es específico de cada lógica de negocios.

Se recomienda poner estas dos partes del código en módulos de lógica de negocios separados y reutilizar el módulo del marco en una métrica diferente.

Este es el código del marco:

```
Dim g_PrevEventTimestamp

Sub OnLoad(time)
     g_PrevEventTimestamp = time
     InitializeStatusVariables
End Sub

Sub OnRegistration(Dispatcher)
     ' Si hay una copia separada de las variables de estado
     ' para cada recurso registrado depende del registro
     ' recursos para los que debería fijar valores iniciales
     ' variables de estado de los recursos recientemente agregados aquí
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
```

A continuación, se presenta un esqueleto del módulo de implementación:

```
' Definir las variables de estado aquí. Puede ser una
' variable global simple o varias variables globales complejas
' dependiendo de la lógica de negocios
Dim g_StatusVar_1, g_StatusVar_2, ... ,g_StatusVar_n

' Definir los contadores aquí.
' Puede ser una variable global simple o varias
' variables globales complejas dependiendo de la lógica de negocios
Dim g_Counter_1, g_Counter_2, ... , g_Counter_n

Sub InitializeStatusVariables ()
     ' Establecer los valores iniciales a las diversas variables de estado
End Sub

Sub InitializeCounters ()
     ' Establecer los valores iniciales a los diversos contadores
     g_Counter_1 = 0
     g_Counter_2 = 0
     '…
     g_Counter_n = 0
End Sub

Function CalculateResult ()
     ' Calcular el resultado. El resultado debería depender
     ' del valor de los contadores. No debería depender
     ' del valor de las variables de estado. No se debería
     ' cambiar los valores de los contadores o
     ' las variables de estado
End Function

Sub UpdateStatus(method, eventDetails)
     ' Actualizar el valor de las variables de estado basándose en
     ' el parámetros (y posiblemente en el valor anterior de
     ' las variables de estado)
End Sub

Sub UpdateCounters(diff)
     ' Actualizar los valores de los contadores basándose en su
     ' anterior valor, en el valor de las variables de estado
```

```
     ' y en el valor del parámetro diff.
     ' En muchos casos este cálculo está basado también en
     ' el 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 explicar mejor este patrón de diseño, a continuación se presenta un ejemplo de la implementación de este patrón para calcular la disponibilidad. Este ejemplo supone que hay eventos de actividad e inactividad que se manejan por dos controladores de eventos separados en la lógica de negocios. La disponibilidad se define como el porcentaje de tiempo durante el cual el sistema está activo a partir del tiempo total en la ranura de tiempo. Se supone que el estado del sistema es el estado del último evento recibido (activo o inactivo).

Aquí está el código de la implementación (el código del marco no se cambia):

```
' Estado variable
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
```

Hay varias variaciones de este patrón. Una de las variaciones más comunes es cundo un contador de tiempo separado se debería mantener para entidades diferentes. Por ejemplo, cuando se mide el tiempo de solución, debería mantenerse un contador separado para cada ticket abierto. En este caso cuando se maneja un evento que es relevante solamente para un ticket es más eficaz actualizar solamente el contador de ese ticket. Cuando un evento común se maneja (como OnPeriodEnd u OnTimeslotEnter) los contadores de todos los tickets se deberían actualizar.

Nota: Esta variación del patrón requiere mantener una copia separada de la variable global g_PrevEventTimestamp para cada ticket.

Algunos buenos ejemplos del uso de este patrón se pueden ver en nuestro contenido predeterminado. Tenga en cuenta que este patrón se utiliza de forma un poco diferente en el contenido predeterminado y que la separación entre el marco y la implementación no es tan obvia allí.

Caso práctico 17: Funcionalidad incorporada

ACE tiene herramientas y funcionalidad incorporada para diversas finalidades. Es preferible utilizar esta funcionalidad a escribir en VBS. Como VBS es un lenguaje interpretado, la reproducción en VBS perjudica el rendimiento.

A continuación se presenta una lista de las funciones incorporadas y la forma apropiada de utilizarlas:

IsWithinTimeslot

Es la más sencilla de las funciones incorporadas. Su objetivo es permitir que la lógica de negocios diga si el sistema está actualmente dentro o no de una ranura de tiempo. Esto elimina el tener que gestionar una variable en la funciones de entrada de ranura de tiempo y salida de ranura de tiempo para hacer lo mismo. Por ejemplo, en lugar de ejecutar el código siguiente:

```
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
```

Se puede ejecutar este código mucho más sencillo:

```
Sub OnEvent(eventDetails)
     If context.IsWithinTimeslot Then
          count = count + 1
     End if
End sub
```

Si se desea utilizar o conservar información sobre la marca de tiempo de la salida y entrada de la ranura de tiempo, esta funcionalidad no cubriría sus necesidades. Pero normalmente esto no es necesario y este código resulta suficiente.

TimeOfLastEvent

Esta función proporciona la marca de tiempo de los últimos eventos de datos intermedios o eventos de datos sin procesar que se manejaron. Esto significa que no es necesario guardar esta información en el controlador de eventos, ya que está directamente disponible mediante esta función. Por ejemplo:

```
Resultado de la función
     Dim LastEventTimestamp
     LastEventTimestamp = Context.TimeOfLastEvent
End function
```

TimeOfLastEventHandler

Esta función devuelve la marca de tiempo del último controlador de eventos convocado por ACE. Esto no sólo incluye los controladores de eventos de datos sin procesar e intermedios, sino también algunos eventos del sistema que también se llamaron. Es especialmente útil en controladores de eventos que no reciben el tiempo para, p.ej., la función de resultado. Por ejemplo:

```
Resultado de la función
     Dim LastEventHandlerTimestamp
     LastEventHandlerTimestamp= Context.TimeOfLastEventHandler
End function
```

NetTime

Esta función le permite especificar dos marcas de tiempo y recibir el tiempo neto (en segundos) que el sistema estuvo dentro de la ranura de tiempo para la regla actual, entre esas dos marcas de tiempo. Se trata de una funcionalidad algo complicada y no se debería implementar en VBS. La implementación de esta en VBS implicaría mantener una lista de todas las salidas y entradas de la ranura de tiempo o calcular la diferencia cada vez que se introduce la salida de la ranura de tiempo directamente, para conocer el intervalo de tiempo entre ellas. En condiciones extremas, esto podría suceder un gran número de veces y no sería bueno para el rendimiento de cálculo. La función interna hace lo mismo después de una significativa optimización, y así lo es mucho más eficaz. Por ejemplo:

```
Resultado de la función
     Dim MyNetTime
     MyNetTime = Tools.NetTime(MyBeginTimestamp, MyEndTimestamp)
End function
```

El objeto de contexto

El objeto de contexto cuenta con una variedad de parámetros que proporcionan información sobre:

- La métrica actual.
- El contrato en el que está.
- El estado de cálculo actual.
- Dominio del servicio, categoría del dominio y sus valores relacionados (p.ej., umbral).

- Información de agrupamiento en clúster: el clúster en general y el elemento de clúster específico que se están manejando.
- Funciones que devuelven listas de recursos según los requisitos del usuario.
- Funciones que le permiten convertir nombres de recurso en Id. de recursos y viceversa.

Acceder a esta información directamente de la base de datos mediante ODBC seguro es muy ineficaz y no tiene sentido, ya que la información está fácilmente disponible a partir del objeto de contexto. Si es posible, utilice siempre la funcionalidad incorporada como una forma de obtener información.

Caso práctico 18: Registro

La lógica de negocios se escribe a menudo de forma que conserva una asignación de la estructura del recurso de la métrica para usarla durante los cálculos. Como la estructura cambia con el tiempo, dicha lógica de negocios tiene que actualizar la estructura en la asignación cuando la estructura del recurso cambia.

El método OnRegistration se convoca cuando la estructura del recurso cambia, ya que es responsable de gestionar cómo debe comportarse el motor con los cambios en los registros y el agrupamiento en clúster provocados por los cambios en la estructura de recursos. El hecho de que este método se convoque para cada cambio de estructura de recurso hace sea un lugar conveniente para actualizar la asignación mencionada anteriormente. Sin embargo, rellenar la asignación no es pertinente para los procesos de registro. Esto significa que, al rellenar la asignación, se degrada el rendimiento de la función OnRegistration. Esto no es importante durante el tiempo de ejecución, ya que no suele suceder muy a menudo. Sin embargo, el método OnRegistration se convoca también durante el proceso de procesamiento de infraestructura del motor, mientras el sistema averigua si los cambios en la estructura de recursos son relevantes para el registro de cada métrica específica de la cual es responsable la instancia. Durante este proceso, el método OnRegistration se convoca para todos los cambios en la estructura de recursos, aunque el cambio de estructura no sea relevante para la métrica actual. Esto significa que el método se puede convocar un gran número de veces por métrica.

Si dicha lógica se implementa en el método OnRegistration, una pequeña degradación del rendimiento durante el tiempo de ejecución podría convertirse en una degradación muy significativa del rendimiento durante el procesamiento de la infraestructura.

Para resolver este problema, las asignaciones de relleno o cualquier otra inicialización que necesite ejecutarse cuando se produce un cambio en la estructura de recursos, que no es relevante para el registro, se pueden llevar a cabo de dos maneras:

Mediante la propiedad IsRunTimeMode en el objeto distribuidor

Esta propiedad permite al usuario averiguar si la ejecución actual es o no una ejecución de cálculo y encerrar la lógica que no sea relevante para el registro en una declaración "if", que asegurará que se ejecutará solamente durante el tiempo de ejecución.

En el ejemplo a continuación, la parte marcada en azul es la parte de la lógica de negocios que es relevante para el registro y siempre tiene que ejecutarse. La parte marcada en verde no es relevante para el registro y se puede encerrar en la nueva declaración "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)
     For Each resource In ThisResourceMap
          GlobalResourceVector.Add resource
     Next
End Sub
```

Este código se puede mejorar cambiándolo de la siguiente manera:

```
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)
          For Each resource In ThisResourceMap
               GlobalResourceVector.Add resource
          Next
```

```
     End If
End Sub
```

Uso del método OnResourceStructureChanged

Este método se ejecuta justo después del método OnRegistration y, de esta manera, ofrece la misma funcionalidad que la metodología original, pero solamente se ejecuta durante el tiempo de ejecución. Este método no se convoca durante el procesamiento de infraestructuras y por ello el rendimiento no se ve perjudicado.

En el ejemplo a continuación, la parte marcada en azul es la parte de la lógica de negocios que es relevante para el registro y siempre tiene que estar en el método OnRegistration. La parte marcada en verde no es relevante para el registro y se puede colocar en la nueva función.

```
Sub OnRegistration (dispatcher)
     Dim MyResource
     MyResource = Context.ClusterItem
     Dispatcher.RegisterByResource “OnEvent”, “My Event Type”, MyResource
     Dim ThisResourceMap
     Set GlobalResourceVector= CreateObject("SlalomVector.Vector")
     Dim resource
     Set ThisResourceMap = Context.ResourcesOfResourceGroup(Context.ClusterItem)
     For Each resource In ThisResourceMap
          GlobalResourceVector.Add resource
     Next
End Sub
```

Este código se puede mejorar cambiándolo de la siguiente manera:

```
Sub OnRegistration (dispatcher)
     Dim MyResource
     MyResource = Context.ClusterItem
     Dispatcher.RegisterByResource “OnEvent”, “My Event Type”, MyResource
End Sub

Sub OnResourceStructureChanged(time)
     Dim ThisResourceMap
     Set GlobalResourceVector= CreateObject("SlalomVector.Vector")
     Dim resource
     Set ThisResourceMap = Context.ResourcesOfResourceGroup(Context.ClusterItem)
     For Each resource In ThisResourceMap
          GlobalResourceVector.Add resource
     Next
End Sub
```

Caso práctico 19: Asistente del adaptador para origen de datos basado en archivo

Este caso práctico examina los métodos de prácticas recomendadas para la integración con un origen de datos basado en archivo. El escenario de ejemplo maneja un archivo de datos CSV que el sistema de origen produce. Con la mayor parte de las integraciones basadas en archivos, CA recomienda seguir algunas directrices básicas para garantizar el mínimo riesgo a la integración. Se presentan a continuación:

- Cuando la opción está disponible, solicitaremos que los datos se proporcionen al sistema de archivos del servidor de la aplicación CA Business Service Insight. Esto asegura que el mecanismo de entrega no depende del adaptador que intenta buscar datos de un almacén remoto (minimiza los problemas de permiso con cuentas de usuario, las incidencias de sincronización, etc.).
- La convención de denominación de archivos es importante, ya que el adaptador ordena los archivos según la denominación alfanumérica. Si podemos controlar esto, CA recomienda solicitar dos partes:
  - Una convención de denominación sensata basada en el contenido del archivo de origen (sobre todo si más de un archivo viene del origen).
  - Una marca de tiempo de orden inverso para asegurar que los archivos se ordenan con el archivo más reciente en último lugar. (esto es, *<nombre_archivo>_yyyymmdd_hhmiss.csv*). La profundidad de la marca de tiempo usada dependerá de la frecuencia de los datos proporcionados.

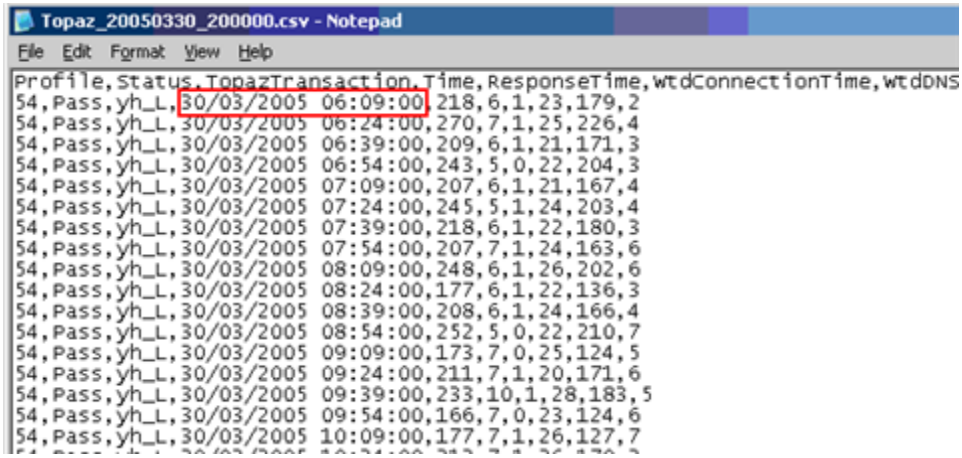
En este escenario, los archivos de origen son de un origen de datos Topaz (ahora conocido como Mercury Global Monitor, propiedad de HP). Esto se creó mediante una API del producto para incluir los archivos obligatorios para los KPI específicos obligatorios. Un proceso automático entrega directamente los archivos al servidor de la aplicación CA Business Service Insight. Los archivos de origen se denominan: *Topaz_yyyymmdd_hhmiss.csv*.

Nota: La marca de tiempo del archivo es la hora en la que se creó, por tanto todas las entradas dentro del archivo llevan a esta hora.

A continuación, se puede ver una muestra de los datos de dentro del archivo.

	A	B	C	D	E	F	G	H	I	J	K
1	Profile	Status	TopazTransaction	Time	Responcetime	WtdConnectionTime	WtdDNSTime	WtdSSLTime	WtdNetworkTime	WtdDownloadTime	
2	54	Pass	yh_L	30/03/2005 06:09:0	218	6	1	23	179	2	
3	54	Pass	yh_L	30/03/2005 06:24:0	270	7	1	25	226	4	
4	54	Pass	yh_L	30/03/2005 06:39:0	209	6	1	21	171	3	
5	54	Pass	yh_L	30/03/2005 06:54:0	243	5	0	22	204	3	
6	54	Pass	yh_L	30/03/2005 07:09:0	207	6	1	21	167	4	
7	54	Pass	yh_L	30/03/2005 07:24:0	245	5	1	24	203	4	
8	54	Pass	yh_L	30/03/2005 07:39:0	218	6	1	22	180	3	
9	54	Pass	yh_L	30/03/2005 07:54:0	207	7	1	24	163	6	
10	54	Pass	yh_L	30/03/2005 08:09:0	248	7	1	26	202	6	
11	54	Pass	yh_L	30/03/2005 08:24:0	177	7	1	22	136	3	
12	54	Pass	yh_L	30/03/2005 08:39:0	208	7	1	24	166	4	
13	54	Pass	yh_L	30/03/2005 08:09:0	252	5	0	22	210	7	
14	54	Pass	yh_L	30/03/2005 09:09:0	173	7	0	25	124	5	
15	54	Pass	yh_L	30/03/2005 09:24:0	211	7	1	20	171	6	
16	54	Pass	yh_L	30/03/2005 09:39:0	166	7	0	24	124	6	

Nota: CA recomienda comprobar los archivos de CSV en el Bloc de Notas (en lugar de únicamente en Excel) para garantizar que el formato de fecha es el previsto. Excel tiene tendencia a aplicar formato a las fechas según los valores de configuración regionales del equipo y puede que no coincida con el formato real que ve el adaptador.



Antes de comenzar cualquier proceso de creación del adaptador, es esencial que haya realizado todo el análisis y la investigación necesarios sobre el origen de datos y los KPI relacionados para asegurar que se conoce lo siguiente:

- Qué campos necesita la lógica de negocios.
- Cuál es el formato de las fechas en el archivo.
- Cuál es la zona horaria de los valores de fecha/hora en el archivo de origen (estas zonas horarias se deberían crear en el sistema antes de comenzar el proceso de creación del adaptador).
- Si hay campos de fecha que pueden tener entradas en blanco o con valor nulo.
- Cuál es el comportamiento del origen de datos (todos los registros se agregan o algunos registros son una actualización a un evento anterior).
- Qué detalles se necesitan para los KPI (esto puede afectar a su elección de "recurso").
- Qué "tipos de eventos" se necesitan para filtrar de forma eficaz los eventos en la lógica de negocios.

Cuando se conozcan estos puntos, puede comenzar el proceso de creación.

Ahora, podemos llevar a cabo el proceso de creación del adaptador basándonos en este escenario, mediante el Asistente del adaptador.

En nuestro escenario utilizamos "TopazTransaction" como el recurso, "Profile" como el tipo de evento y el campo "hora" como la marca de tiempo. También traemos los campos "Status", "ResponseTime" y "WtdDownloadTime" a la estructura del evento para utilizarlos con la lógica de negocios.

Creación del adaptador

Primero, asegúrese de que el sistema está preparado para crear el adaptador nuevo e implementarlo en el servidor correctamente, comprobando los servicios siguientes que se inician en el servidor de aplicaciones.

- Servicio de implementación del adaptador
- Servicio de escucha del adaptador

A continuación, vaya a la página de Adaptadores y cree un adaptador nuevo. Debería elegir la opción Agregar nuevo/a > Adaptador de archivos de texto de la página Adaptadores.

Paso General

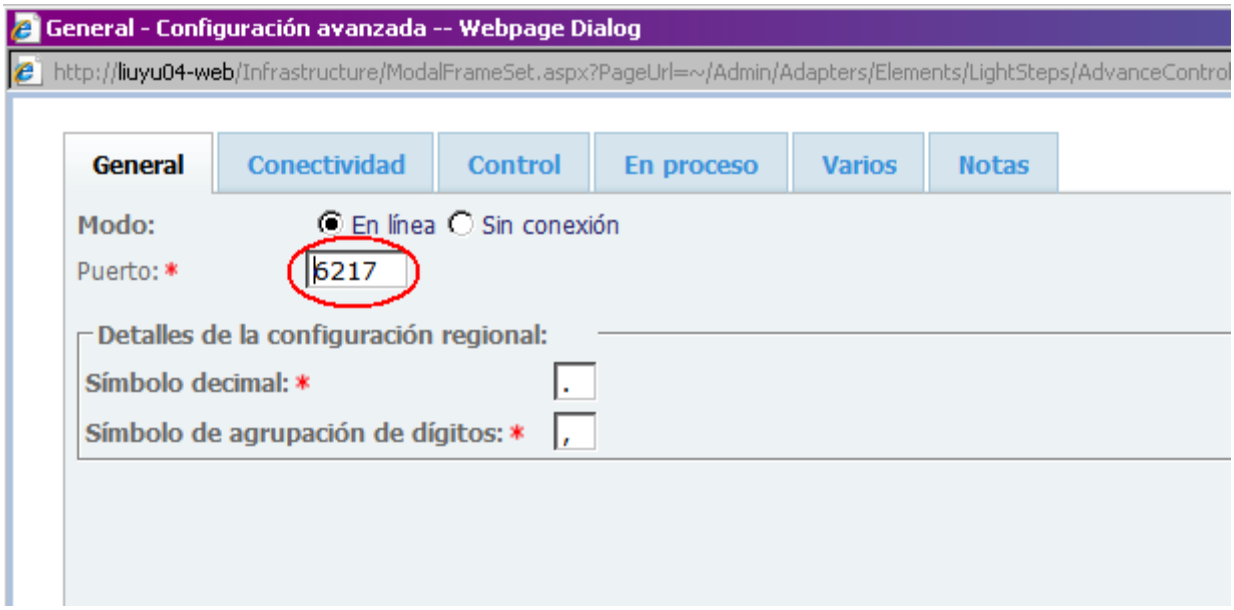
En el paso General del Asistente del adaptador introduzca los campos siguientes:

- *Nombre*: proporcione un nombre apropiado para el adaptador.
- *Dirección de adaptador*: LOCALHOST es la opción predeterminada (para la implementación del servidor de aplicaciones), pero se pueden introducir otras direcciones mediante el botón si es necesario.
- *Formato de hora*: es el formato de hora predeterminado utilizado en los campos de fecha/hora dentro del origen de datos. Si se elige esto correctamente se asegura que los campos se detecten automáticamente de forma adecuada más tarde en el proceso del asistente. En este momento se pueden introducir un Nuevo formato de hora si es obligatorio, mediante el botón de este campo.
- *Zona horaria*: es la zona horaria en la que se registran los datos de registro. Esto es necesario para que el adaptador pueda volver a cambiar correctamente el campo event_timestamp (y otros campos de fecha/hora) a UTC para un desplazamiento interno de hora correcto. Se debería haber introducido previamente según la lista de comprobación del origen de datos de la sección anterior.

Notas:

Hay también un botón "Opciones avanzadas" en esta página que proporciona acceso a varios parámetros de configuración del adaptador. La mayor parte de estos se pueden dejar como los valores predeterminados, a menos que sea obligatorio modificarlos.

El Adaptador "Puerto" se asigna automáticamente al adaptador a partir de CA Business Service Insight hacia adelante, pero se puede anular aquí si es necesario. Otros parámetros que se pueden cambiar son: detalles de la configuración regional, modo En línea/Sin conexión, detalles de conexión, opciones de control e inicio de sesión, valores de configuración Ejecutar una vez/Siempre, Límites de error, nombres de archivo y comentarios.



La descripción de cada uno de estos valores de configuración vas más allá del ámbito de este caso práctico, pero puede encontrar más información sobre ellos en Especificaciones de configuración del adaptador (en la página 369).

Haga clic en Siguiente para continuar con el siguiente paso del Asistente. El paso siguiente proporciona acceso a la Interfaz de origen de datos del adaptador.

Paso Interfaz de origen de datos

En el paso de Interfaz de origen de datos del Asistente del adaptador introduzca los campos siguientes:

- *Nombre del origen de datos*: un nombre para este archivo de origen particular (se pueden tener varios archivos de origen en un adaptador).
- *Ruta de archivo*: la ruta en el servidor de aplicaciones (u otro servidor) que contiene los archivos de datos de origen. Para un servidor distinto al servidor de aplicaciones, utilice una notación de UNC (es decir, //server01/sourcefolder)
- *Patrón del nombre*: lo puede usar con un carácter comodín para filtrar los archivos encontrados en la "Ruta de archivo" que están cargados por el adaptador.

La ficha Definición de análisis también ayuda a definir la estructura del archivo que se está importando. Los campos se pueden utilizar de la siguiente manera:

- *Título*: casilla de verificación de valores booleanos para especificar si hay o no una fila de "título" en el archivo de datos (es decir, la primera fila en el CSV es un nombre de título, seguido por los valores de las siguientes filas).
- *Delimitadores*: especifique el delimitador en el archivo que separa los campos individuales.

Nota: Hay también dos botones Opciones avanzada que proporcionan opciones de configuración adicionales. Un botón está en la ficha Detalles y el otro está en la ficha Definición de análisis. El botón Opciones avanzadas en la ficha Detalles proporciona acceso a los parámetros siguientes:

El origen de datos está activo: casilla de verificación booleana que le permite activar/desactivar esta sección particular de origen del adaptador.

Después del procesamiento: le permite especificar si el archivo se suprime o se mantiene después del procesamiento.

Nombre de archivo inicial: establece el nombre del archivo inicial para iniciar el procesamiento (en caso de que no se desee cargar todos los archivos en un directorio particular).

Comprobar datos nuevos cada: define el intervalo de tiempo entre comprobaciones de nuevos archivos cuando el adaptador está establecido para ejecutarse continuamente.

El botón Opciones avanzadas en la ficha Definición de análisis proporciona acceso para definir opciones de anulación de registros, como registros de varias líneas, etc.

Una vez que se establecen los detalles y las definiciones de análisis, es posible cargar una muestra del archivo de origen en el asistente para probar las opciones de configuración y obtener una vista previa del contenido del archivo.

Si hace clic en el botón Explorar se puede cargar un archivo de muestra en la ventana de vista previa a continuación. Vaya al archivo de muestra y, a continuación, pulse el botón Archivo de prueba. Este paso es opcional.

Nota: la función Explorar abre el equipo local en el que está trabajando. Si este no es el servidor de aplicaciones, debe tener una copia de los archivos de origen que estén en el servidor para que esto funcione. No se puede explorar en el servidor directamente mediante esta función.

Una vez que el archivo está cargado debería poder ver el contenido mostrado en la ventana de vista previa, como a continuación.

Nota: El nombre "Campo" se muestra en el encabezado, junto con el tipo de campo detectado (entero, cadena, hora, etc.). Hay que comprobar que estos se han detectado correctamente y de acuerdo con sus requisitos antes de continuar. Hay que asegurarse de que:

- La marca de tiempo se ha detectado como "hora"; esto se hace según el "Formato de hora" especificado en el primer paso del Asistente.
- El recurso se ha detectado como una "cadena".

Cuando esté satisfecho con la vista previa de archivo, seleccionar Siguiente. Se muestra el paso Asignación.

Paso Asignación

El siguiente (y último) paso del Asistente del adaptador realiza las tareas de traducción y le permite asignar los campos de entrada del origen de datos a los campos de salida, que conforman el evento de CA Business Service Insight. Hay dos opciones para continuar, dependiendo de si el tipo de evento ya está creado o no en el sistema. También hay varias opciones que puede decidir configurar y asegurarse de que han seguido los estándares de denominación. Aunque algunas de estas son opcionales, facilitan el proceso y reducen los pasos obligatorios. A continuación, se indican los pasos obligatorios.

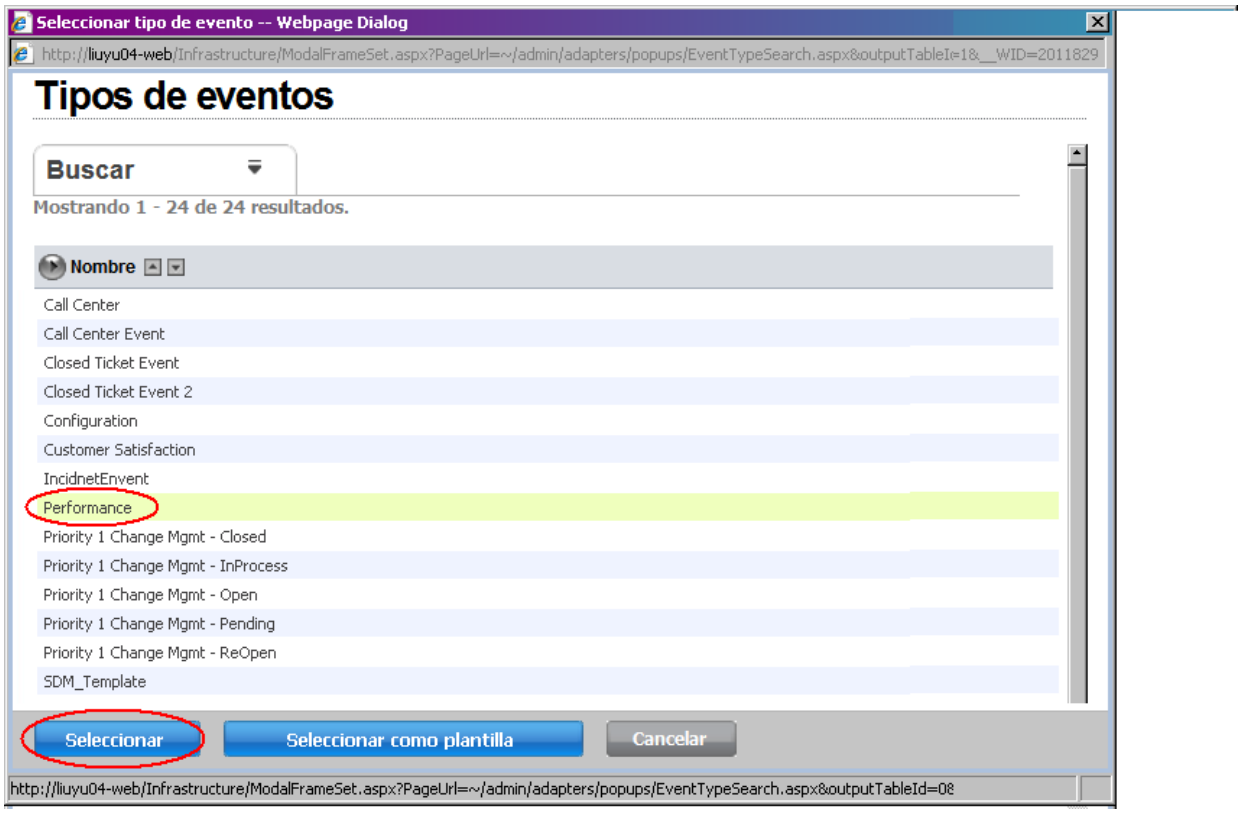
Los pasos para configurar la fase de asignación son los siguientes (incluye pasos opcionales):

1. Proporcione un nombre al formato de entrada (útil para adaptadores con varias entradas).
2. Agregue algunos campos adicionales obligatorios (como Valor constante, Origen de datos, Título, Nombre de archivo o tipos de campos compuestos).
3. Cree algunos criterios obligatorios de Validación de entrada.
4. Seleccione un tipo de evento existente o cree un nuevo tipo de evento (obligatorio).
5. Realice la asignación de campos de entrada a salida (obligatorio).
6. Proporcione un nombre para el formato de salida.
7. Realice la asignación para el Id. del recurso, la marca de tiempo y el tipo de evento.
8. Cree algunos criterios obligatorios de Validación de salida.
9. Especifique el valor de configuración de "OnDuplication" para los eventos.

Si decide crear un nuevo tipo de evento, aparece una nueva ventana que se rellena previamente, según el formato de entrada de la pantalla principal. Debe introducir el nombre de tipo de evento y asignar también un tipo de recurso al tipo de evento.

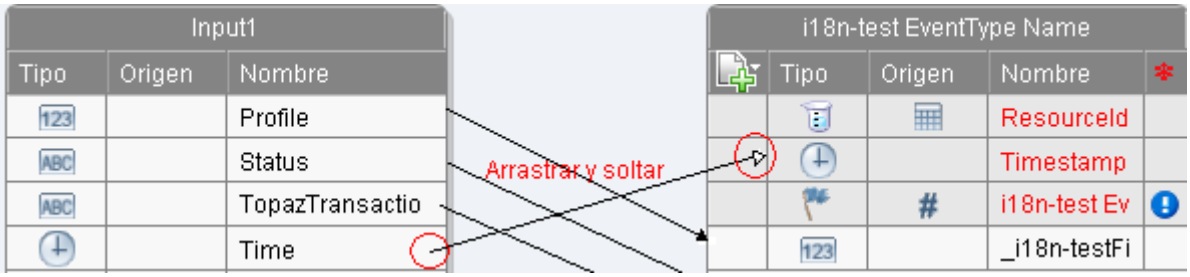
Cuando haya completado estos pasos, puede "Guardar y cerrar" y la asignación finaliza automáticamente.

Si elige la opción "Seleccionar tipo de evento", aparece una lista de los tipos de eventos existentes en el sistema y entre los que puede elegir. Al continuar, sin embargo, se vinculan sólo automáticamente los campos de la entrada con nombres coincidentes y los tipos de datos del tipo de evento. Los campos restantes tienen que asignarse manualmente.

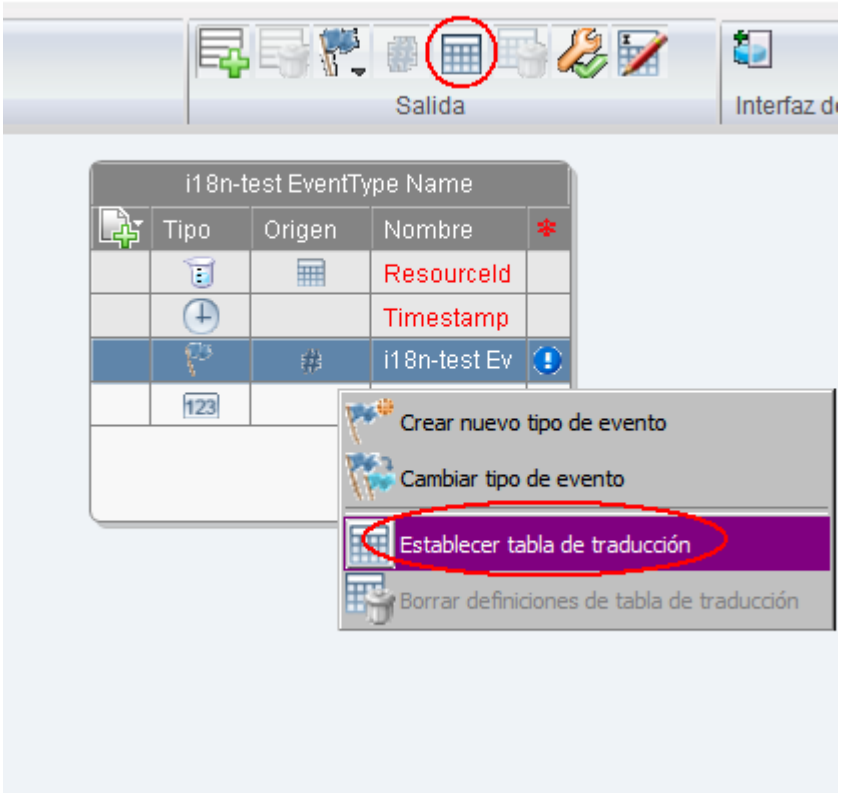


El siguiente paso es configurar el Id. del recursos, la marca de tiempo y el tipo de evento. Si los campos ya existen (en caso contrario, se deberían crear en pasos anteriores) se pueden vincular a estos campos según sea necesario.

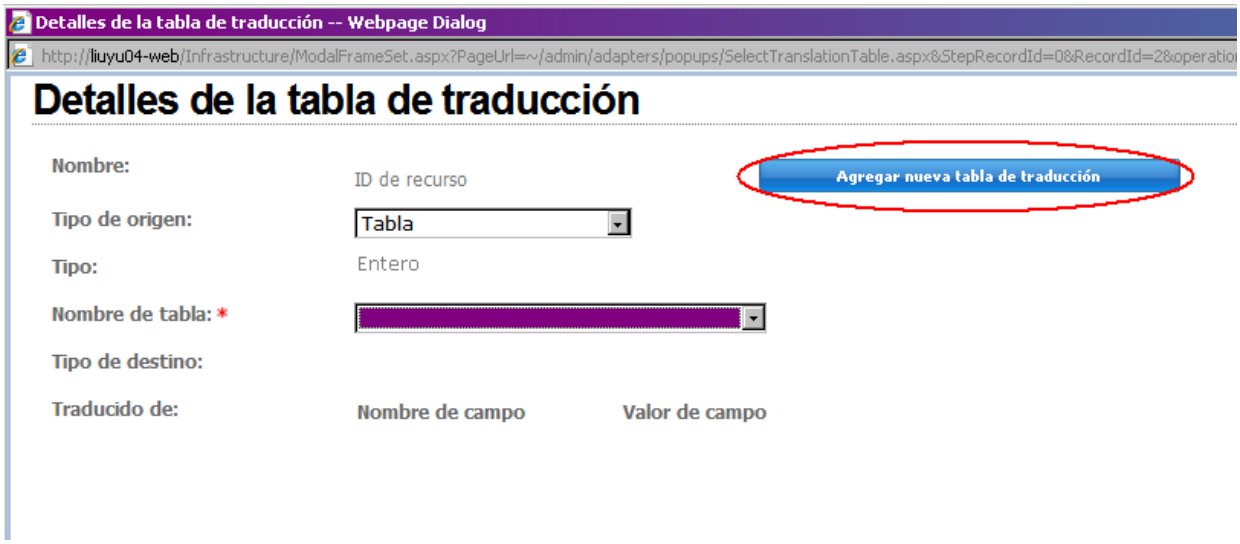
La interfaz es compatible con el método de asociación de arrastrar y soltar. Si la asociación no permanece después de haberla configurado, asegúrese de que los tipos coinciden (es decir, hora/cadena/entero, etc.).



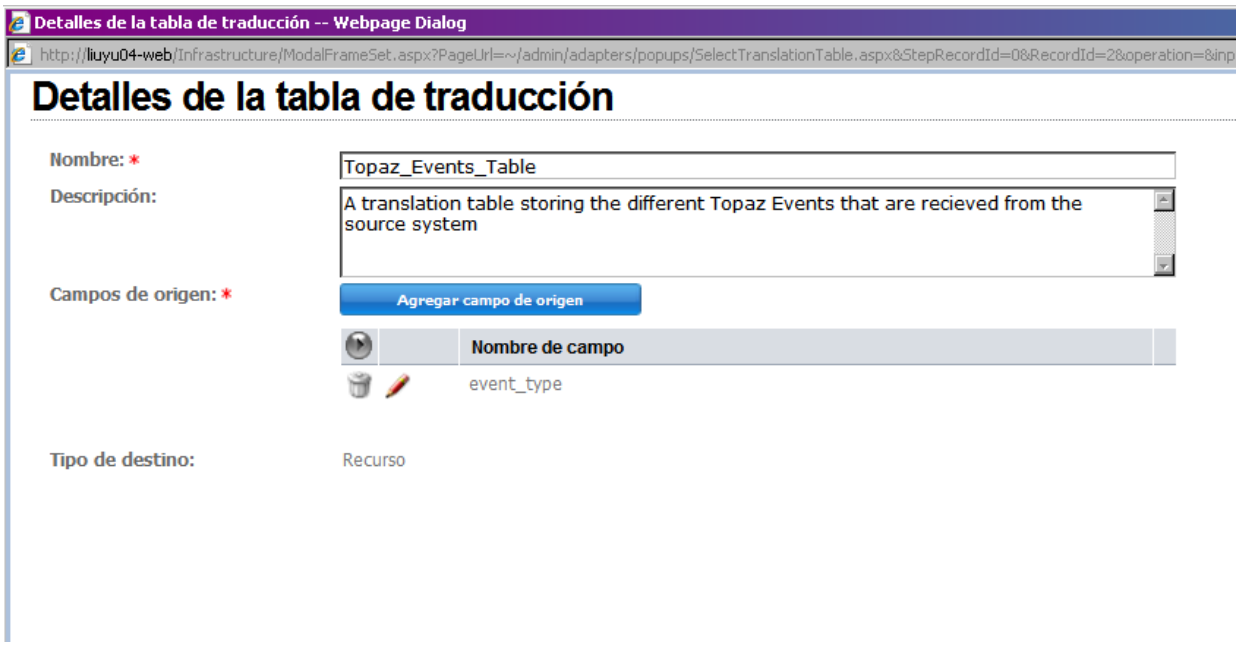
- El Id. del recurso se debería asignar según los requisitos (identificados durante el análisis) a partir de un valor de cadena en la entrada.
- La marca de tiempo se debería establecer como la variable de hora que define la hora en la que el evento tuvo lugar y se utilizará para fines de cálculo.
- El tipo de evento tiene como valor predeterminado un valor constante según el campo "Nombre del origen de datos" de la pantalla anterior. Esto se puede anular si es necesario. Esto sería necesario si se deben enviar varios eventos desde este adaptador, según el valor de un campo específico (es decir, desea enviar un evento diferente dependiendo del valor de un campo de entrada). Para hacer esto, haga clic con el botón secundario del ratón en la fila tipo de evento (o haciendo clic en el botón anterior, como se indica en el cuadro más abajo) y seleccione "Establecer tabla de traducción".



Después de esto, aparece una ventana emergente.



Si la tabla de traducción para este campo de valor ya existe, se puede elegir aquí o también se puede configurar una nueva tabla de traducción. Se puede utilizar el botón de la pantalla anterior para ello.



Una vez que la tabla se ha creado, es necesario especificar qué campos de entrada se asignan a los "Campos de origen" indicados anteriormente.

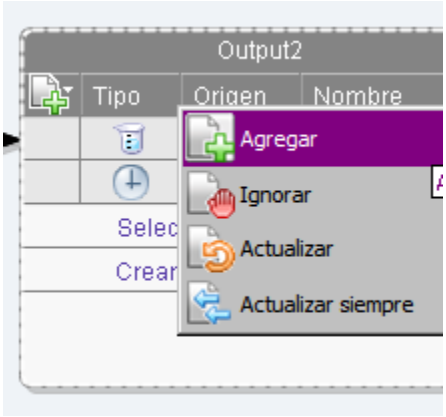
Si desea especificar una tabla de traducción diferente para los recursos del adaptador, debería hacerlo en este momento. Esto se hace mediante un proceso similar al descrito para el tipo de evento.

Nota: De forma predeterminada, el Asistente del adaptador ha adjudicado todos los recursos a una tabla de traducción preexistente llamada "Default_Translation_Table", si no se especificó de otra manera. Esto puede estar bien para implementaciones sencillas, pero para implementaciones más complejas (y con fines de separación de datos) CA recomienda utilizar una tabla diferente. También es obligatorio cuando los "Campos de origen" de la sección de asignación del adaptador son diferentes o contienen más que un valor.

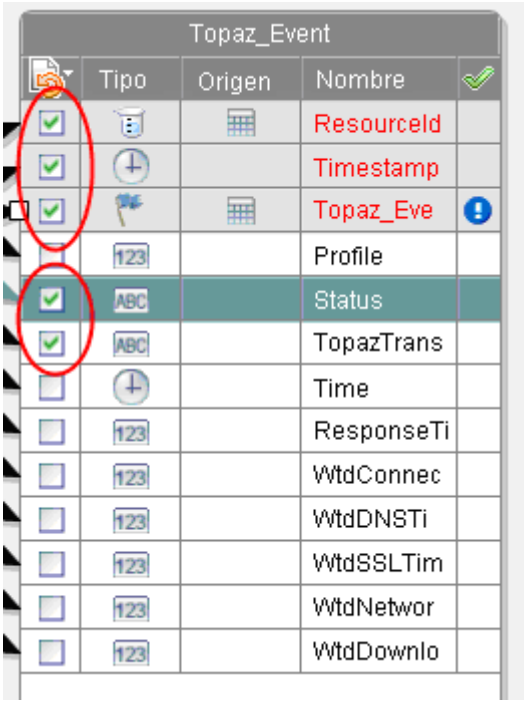
El último paso de la fase de Asignación es configurar los valores de configuración de "OnDuplication" para el adaptador. Esta configuración describe la acción llevada a cabo cuando se recibe un segundo evento, con valores para coincidentes para todos los campos "clave". Esta "clave" única se puede definir para cada salida del adaptador (consulte a continuación para obtener más información sobre esto). De forma predeterminada este valor de "OnDuplication" está definido para "Agregar", así que sólo debe modificarse si se necesita una acción diferente. Los valores disponibles son:

- *Agregar*: el nuevo evento se agrega como un nuevo evento separado.
- *Ignorar*: el adaptador ignora (elimina) el nuevo evento.
- *Actualizar*: el nuevo evento sustituye el evento previamente cargado solamente si los campos "Valor" de los eventos son diferentes
- *Actualizar siempre*: el nuevo evento sustituye el evento previamente cargado.

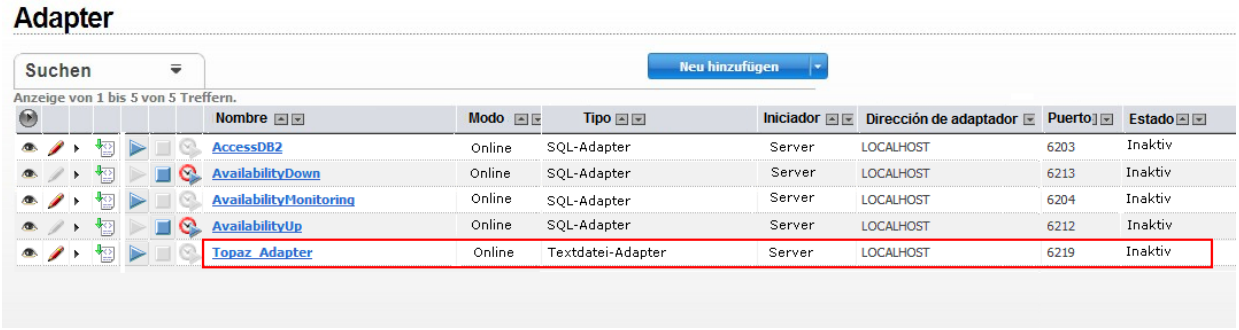
El uso de las opciones distintas a "Agregar" puede afectar al rendimiento del adaptador y debería considerarse con cuidado antes de implementarse, sobre todo cuando los volúmenes de datos son muy grandes.



Si el valor se establece como un valor distinto a "Agregar", la estructura del resultado muestra una serie de casillas de verificación que se usan para definir la "clave" única. La "clave" está formada por cada uno de los elementos con una comprobación de ellos. La elección de la clave debe decidirse según los requisitos después de realizar un análisis detallado.



Una vez que la configuración de la sección de asignación se ha completado, haga clic en el botón Finalizar en la parte derecha inferior de la pantalla. Se vuelve a la lista de adaptadores del sistema. Ahora debe poder ver el adaptador que ha creado con un estado de "Inactivo".



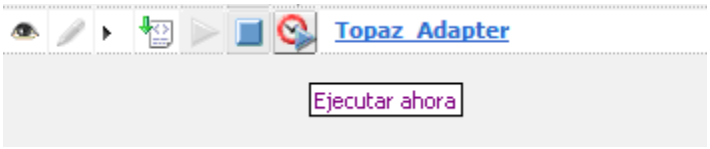
Implementación del adaptador

Ahora que ya ha configurado el adaptador, es necesario implementarlo en el servidor de aplicaciones. Si hace clic en el botón "Iniciar adaptador", se hace que al Servicio de implementación del adaptador implemente el adaptador en el sistema de archivos. Esto incluye lo siguiente:

- Creación de las carpetas obligatorias para ejecutar el adaptador.
- Copia de la configuración XML a la carpeta.
- Creación de un acceso directo para ejecutar el adaptador.

Una vez que esto se ha hecho, los cambios se reflejan en el servidor.

A partir de ahora, es posible ejecutar el adaptador desde la interfaz gráfica de usuario; para ello, haga clic en el botón Ejecutar, que estará activo. Esto hace que el Servicio de implementación del adaptador ejecute el adaptador en el servidor y comience la recopilación de datos.

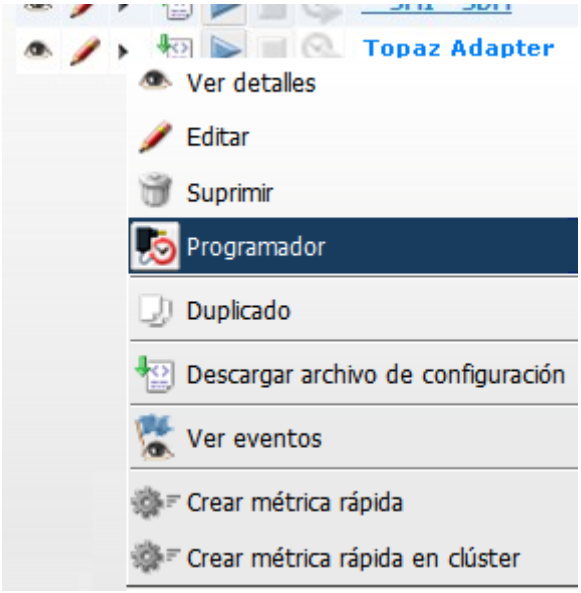


Una vez que el adaptador se ha ejecutado debe ser ahora capaz de ver los eventos y recursos pendientes en la sección de traducciones pendientes del sistema.

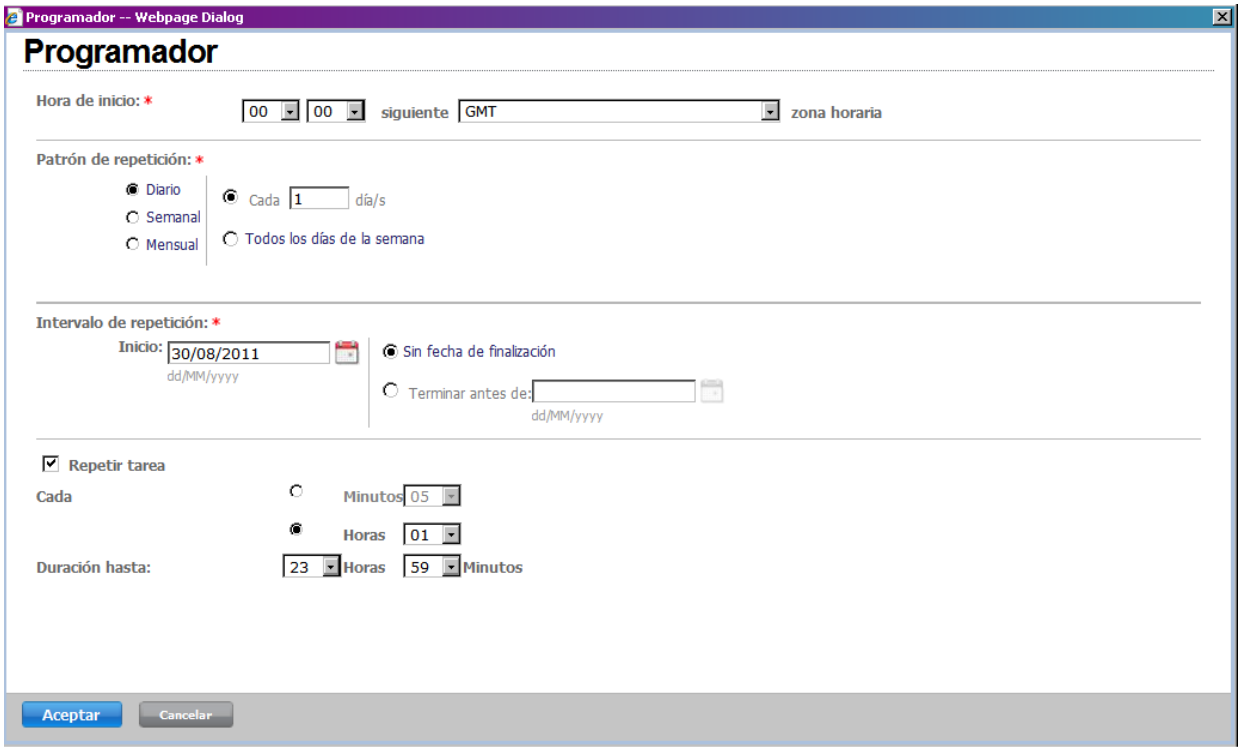
Las entradas pendientes se pueden traducir, a continuación, según la configuración normal del sistema. Una vez que se ha finalizado la traducción, el adaptador debe volver a ejecutarse otra vez para cargar los datos sin procesar en el sistema.

Programación del adaptador

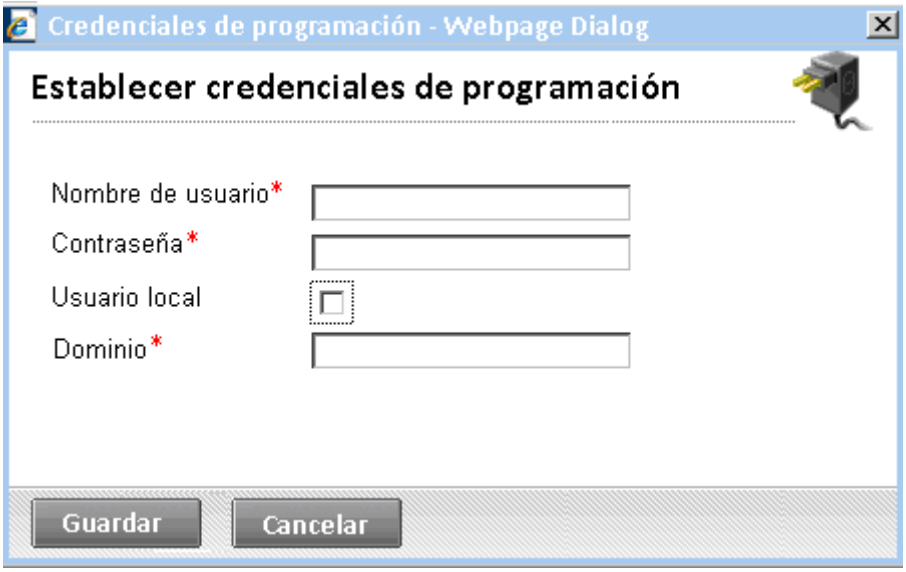
Además de ejecutar el adaptador, también es posible establecer una programación para el adaptador desde la interfaz gráfica de usuario. Sin embargo, para hacer esto el adaptador tiene que estar en el estado "Detenido". Una vez está detenido, puede establecer la programación en el menú contextual del adaptador, seleccionando "Programador".



Las opciones de programación son las mismas que las proporcionadas por el Programador de tareas de Windows. En realidad, el Servicio de implementación del adaptador crea "entre bastidores" esta programación como un elemento del Programador de tareas.



Después de agregar una programación, y cuando el adaptador se implementa a continuación, se pide al usuario que introduzca las credenciales de la cuenta en el servidor que realiza la tarea. Hay que introducirlas, ya que se necesita la cuenta de servicio creada para ejecutar el sistema CA Business Service Insight (OblicoreSrv de forma predeterminada), u otra cuenta administrativa.



Este programador integrado es una función muy útil, ya que permite al usuario de la interfaz gráfica controlar la programación del adaptador sin tener que acceder directamente al escritorio del servidor de aplicaciones (a la espera de los permisos de usuario apropiados, por supuesto).

Caso práctico 21: Ejemplo de integración de LDAP

Requisitos de la organización

Use los valores definidos por el usuario que ya están presentes en el servidor LDAP de la organización. Además, el portal de la organización se usa para el inicio de sesión y acceso de CA Business Service Insight mediante las capacidades de inicio de sesión silencioso de CA Business Service Insight para los portales de inició único de sesión (SSO).

Defina un script de traducción de Visual Basic (VB) para la creación automática de usuarios en el sistema de CA Business Service Insight (sincronización de LDAP). El script de traducción se utiliza para conectarse al servidor LDAP de la organización y extraer la lista de usuarios de allí. Los métodos de paquetes de herramientas de CA Business Service Insight se utilizan para la creación de usuarios, grupos y roles.

Ejemplo de código de VB de conexión de LDAP

```
Option Explicit On
Imports System.DirectoryServices
    Public Function GetLDAPUsers(ByVal ldapServerName As String, ByVal pFindWhat
As String) As ArrayList
        Dim oSearcher As New DirectorySearcher
        Dim oResults As SearchResultCollection
        Dim oResult As SearchResult
        Dim RetArray As New ArrayList
        Dim mCount As Integer
        Dim mIdx As Integer
        Dim mLDAPRecord As String
        Dim ResultFields() As String = {"securityEquals", "cn"}
        Try
            With oSearcher
                .SearchRoot = New DirectoryEntry("LDAP://" & ldapServerName & _
              "/dc=lippogeneral,dc=com")
                .PropertiesToLoad.AddRange(ResultFields)
                .Filter = "cn=" & pFindWhat & "*"
                oResults = .FindAll()
            End With
            mCount = oResults.Count
            If mCount > 0 Then
                For Each oResult In oResults
                    mLDAPRecord =
oResult.GetDirectoryEntry().Properties("cn").Value & "  " &
oResult.GetDirectoryEntry().Properties("mail").Value
                    RetArray.Add(mLDAPRecord)
                Next
            End If
```

```
        Catch e As Exception
            MsgBox("Error is " & e.Message)
            Return RetArray
        End Try

        Return RetArray
    End Function

Sub CheckAddUser
Dim map
Set map = Tools.GetUserDetails("acme@Test")
' Comprobar si ya existe el usuario
' Asignación de Tools.AddUserByMap
' Consultar con el duplicado
map("UserName") = "acme2"
map("UserPassword") = "acme2"
map("UserPasswordExpirationInterval") = "50"
map("UserDescription") = "New description"
map("UserStatus") = "INACTIVE"
Tools.AddUserByMap map
Tools.Commit
End Sub
```

Métodos de script de traducción de VB de CA Business Service Insight

- Métodos de organización

  AddOrganization / IsOrganizationExists

- Métodos de roles

  IsRoleExists / SearchRoles

- Métodos de usuarios

  AddUserByMap / GetUserName

  GetOrganizationName / IsUserExists

  GetUserDetails / SearchUsers

  GetUserFullName / UpdateUserByMap

- Métodos de grupos de usuarios

  AddUserGroupByMap / IsUserGroupExists

  DeleteUserGroup / SearchUserGroups

  GetUserGroupDetails / UpdateUserGroupByMap

Cree un código de "inicio de sesión silencioso" e intégrelo en el portal de la organización que se va a usar para el inicio de sesión de CA Business Service Insight.

Ejemplo de código C# de la puerta de enlace de CA Business Service Insight (se integrará en el portal de la organización)

```
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>
      /// Esta página de muestra es una puerta de enlace de muestra a la interfaz
de la aplicación Oblicore Guarantee(tm)
      /// La página se debería convocar antes de navegar a cualquier página en el
sitio web de Oblicore Guarantee
      /// o cualquier página utilizando los servicios web proporcionados por
Oblicore
      /// La página de OblicoreGateway debería realizar las acciones siguientes:
      ///         1) Convocar el servicio web de autenticación de Oblicore para
autenticar el usuario actual
      ///         2) Convocar la página SilentLogin.asp en el sitio web de
Oblicore para iniciar sesión silenciosamente al sitio web de Oblicore
      ///            y crear el contexto de sesión de usuario
      ///         3) Redirigir a la página deseada
      /// </summary>
      public partial class _Default : System.Web.UI.Page
      {

            /// <summary>
            /// credenciales de usuario de Oblicore
            /// </summary>
            struct UserCredentials
            {
                  public string UserName;
                  public string Organization;
            }

            private void Page_Load(object sender, System.EventArgs e)
            {
```

```
                if (Request["OGSESSIONID"]!=null)
                {
                      //SilentLogin.asp ha vuelto a dirigir a esta página
después de la autenticación.
                      //Guardar OGSESSIONID en la cookie para utilizar más
adelante
                      HttpCookie SessionCookie = new
HttpCookie("OGSESSIONID",Request["OGSESSIONID"]);
                      Response.Cookies.Add(SessionCookie);
                      //Redirigir a la página deseada
                      Response.Redirect("/");
                }
                else
                {
                      //Primera vez que se entra en la página.
                      //Realizar la autenticación.

                      string sAuthToken = string.Empty;

                      // Obtener nombre de usuario de OG y organizaciones del
directorio de usuarios del portal
                      UserCredentials ucOblicoreUser =
GetOblicoreUserCredentials();

                      //Inicializar el servicio web de autenticación de
Oblicore
                      //El proyecto debería incluir la referencia web al
servicio
                      //El servicio está localizado en el sitio web de Oblicore
Guarantee en /WebServices/OblicoreAuth.asmx
                      OblicoreAuth oAuthService = new OblicoreAuth();
                   //     oAuthService.ClientCertificates.Add(x509);
                      oAuthService.Url = "https://" + "localhost" +
"/WebServices/OblicoreAuth.asmx";
                      try
                      {
                            //Invocar servicio web de autenticación.
                            //El método AuthenticateUser devuelve un token
cifrado, que debería pasar a
                            //la página SilentLogin.asp, situada en la carpeta
raíz del sitio web de Oblicore Guarantee
                            sAuthToken =
oAuthService.AuthenticateUser(ucOblicoreUser.UserName,ucOblicoreUser.Organization
);
                      }
                      catch (Exception ex)
                      {
                            //Continuar con error de autenticación si los hay
```

```
                              Response.Write("The error has occurs during
Oblicore authentication: " + ex.Message);
                              Response.End() ;
                        }

                        //Convocar la página Call SilentLogin.asp junto con el
pase a la carpeta de la autenticación
                        //La página SilentLogin.asp está situada en la carpeta
raíz del sitio web de Oblicore Guarantee
                        //Después del inicio de sesión, la página SilentLogin.asp
volverá a redirigir a la página actual junto con el parámetro OGSESSIONID de pase

//Response.Redirect(ConfigurationSettings.AppSettings["OGURL"].ToString() +
"/SilentLogin.asp?AuthToken="+Server.UrlEncode(sAuthToken)+"&DesiredPage="+GetCur
rentPageURL());
                        Response.Redirect("https://vit-
05/SilentLogin.asp?AuthToken=" + Server.UrlEncode(sAuthToken) +
"&DesiredPage=/Oblicore.asp"); // + GetCurrentPageURL());
                  }
            }


            /// <summary>
            /// Obtener nombre de usuario y organización de Oblicore Guarantee
del directorio de usuarios del portal
            /// El método se supone que convoca ActiveDirectory u otro
repositorio mediante el portal de la API
            /// para obtener el nombre de usuario y organización actual en
términos de Oblicore Guarantee
            /// </summary>
            /// <returns>estructura de credenciales de usuario de Oblicore
Guarantee</returns>
            private UserCredentials GetOblicoreUserCredentials()
            {
                  UserCredentials ucOblicoreUser = new UserCredentials();
                  //actualmente siempre se supone que el usuario es administrador
y la organización es Oblicore (valor predeterminado)
                  ucOblicoreUser.UserName = "sadmin";
                  ucOblicoreUser.Organization = "Oblicore";
                  return ucOblicoreUser;
            }



            /// <summary>
            /// Recupera URL de página actual
            /// </summary>
            /// <returns>URL completa de la página actual</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, código generado por el diseñador de formularios web
            override protected void OnInit(EventArgs e)
            {
                  //
                  // CODEGEN: el diseñador de formularios web de ASP.NET requiere
esta llamada.
                  //
                  InitializeComponent();
                  base.OnInit(e);
            }

            /// <summary>
            /// Método obligatorio por motivos de compatibilidad del diseñador;
no modificar
            /// el contenido de este método con el editor de código.
            /// </summary>
            private void InitializeComponent()
            {
                  this.Load += new System.EventHandler(this.Page_Load);
            }
            #endregion
      }
}
```

- Diagrama de flujo

  El diagrama siguiente muestra el proceso de sincronización y el flujo de acceso del portal a los usuarios. El script de traducción se configura para ejecutarse periódicamente. El script mantiene la lista de usuarios de LDAP hasta la fecha y añade/elimina los usuarios según sea necesario.

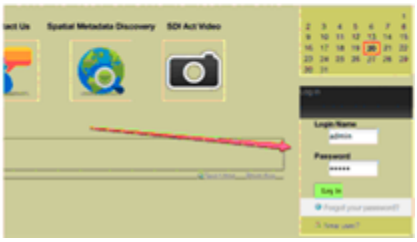
  Los usuarios realizan un inicio de sesión al portal de la organización. El portal se puede configurar para reenviarlos al servidor de CA Business Service Insight o mostrar una lista de otras aplicaciones disponibles. El servidor de CA Business Service Insight utiliza las credenciales que se proporcionaron durante el inicio de sesión del portal inicial.

Lista de usuarios LDAP:

John
Albert
Sandy
Allison

Usuarios de Insight incluyendo los usuarios LDAP sincronizados:

Sadmin
John
Albert
Sandy
Allison



Caso práctico 22: Generación de informes mediante PERL

El ejemplo siguiente muestra cómo utilizar scripts de PERL para conectarse al servicio web de informe de CA Business Service Insight y transferir el parámetro de xml de criterios en una envoltura SOAP mediante un flujo de HTTP.

Nota: El código XML que se transfiere en la envoltura SOAP no puede contener los símbolos "<" o ">", sino el código HTML para estos símbolos (es decir, <=> >=<)

```
#!/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;
```

Apéndice C: Especificaciones de configuración del adaptador

Esta sección contiene los siguientes temas:

Estructura global (en la página 369)
Sección General (en la página 370)
Sección Interface de CA Business Service Insight (en la página 376)
Sección DataSourceInterface (en la página 379)
Sección de la interfaz de SQL (en la página 382)
Sección InputFormatCollection (en la página 387)
Sección TranslationTableCollection (en la página 392)
Sección TranslatorCollection (en la página 394)

Estructura global

La estructura general del archivo XML de configuración es la siguiente:

```
< AdapterConfiguration>
    <General…>
    <OblicoreInterface…>
    <DataSourceInterface…>
    <InputFormatCollection…>
    <TranslatorCollection…>
    <TranslationTableCollection…>
</ AdapterConfiguration>
```

Cada uno de los nodos se describe en las secciones siguientes.

Sección General

La sección General incluye atributos y subnodos de la siguiente forma:

Estructura 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 el número de versión mayor.
- MinorVersion: especifica el número de versión menor.
- WorkingDirectoryName: (opcional).

  Especifica la ruta predeterminada para los archivos de salida del adaptador (control de origen de datos, traducción y envíos).

El valor predeterminado es el directorio de salida (carpeta) en el directorio donde está el archivo de configuración.

- DataSourceControlFileName: (opcional).

  Nombre del archivo que emplea el adaptador para hacer un seguimiento de los últimos registros de datos recuperados del origen de datos.

  El valor predeterminado es DataSourcecontrol.xml (si no se especifica lo contrario, se usa el valor predeterminado).

- SendFileName: (opcional).

  Nombre del archivo que incluye los eventos antes de enviarlos a CA Business Service Insight.

  El valor predeterminado es Value-Send.txt (si no se especifica lo contrario, se usa el valor predeterminado).

- SendControlFileName: (opcional).

  Nombre del archivo que emplea el adaptador para hacer un seguimiento del proceso de envío.

  El valor predeterminado es Value-SendControl.xml (si no se especifica lo contrario, se usa el valor predeterminado).

- DataSourceIdMaxLength.

  Longitud máxima para el campo DataSourceId en la tabla t_raw_data. Si el usuario inserta una cadena de mayor longitud que esta, recibe un error en el adaptador.

  Este valor debe ser menor que o igual al tamaño real de la tabla.

  Valor predeterminado: 60.

- SaveIllegalEvents: si se selecciona, este atributo indica que se deberán escribir en el archivo los eventos ilegales.

  Valor predeterminado: no.

- WriteEventErrorToLog: indica si los errores de datos se escriben en la tabla T_Log.

  Valores posibles: [yes/no]

  Valor predeterminado: yes.

- IllegalEventsDirectoryName: (no tiene valor predeterminado).

- SendFileName: (opcional).

  Nombre del archivo que incluye los registros antes de enviarlos a CA Business Service Insight.

El valor predeterminado es Value-Send.txt (si no se especifica lo contrario, se usa el valor predeterminado).

- §SendControlFileName: (opcional).

  Nombre del archivo que emplea el adaptador para hacer un seguimiento del proceso de envío.

  El valor predeterminado es Value-SendControl.xml (si no se especifica lo contrario, se usa el valor predeterminado).

- LogDebugMode: (opcional)

  Valores posibles: [yes/no]

  Cuando se establece en "yes", se envía lo siguiente al registro: la fila original del origen de datos, los resultados del análisis y los eventos unificados de CA Business Service Insight.

- ConsoleDebugMode: (opcional).

  Valores posibles: [yes/no]

  Cuando se establece en "yes", aparecen mensajes de depuración en la consola.

  - **Indicadores para leer y traducir registros**:
    - **.**: para un registro leído y traducido a un evento de salida.
    - **i**: para un registro ignorado por el analizador de expresiones regulares.
    - **I**: para un registro leído e ignorado por las tablas de traducción.
    - **R**: para un registro leído pero rechazado por la tabla de traducción (las tablas de traducción no lo pueden traducir).
    - **X**: para un registro cuyo análisis ha dado un problema. Se ignorará y se perderá, o se guardará en el directorio de eventos ilegales.

  Nota: Cuando el registro leído pasa por más de un traductor, la indicación del registro se encerrará entre corchetes. Por ejemplo: [...] o [RRI].

  - **Indicadores de estado del adaptador**:
    - **0**: el adaptador está activo y no lee registros en el último segundo.
    - **E**: estado de error.
    - **P**: estado de pausa.
    - **S**: comando de detención recibido de CA Business Service Insight.
    - **B**: estado de bloqueo, la tabla de eventos rechazada está llena.
    - **Indicadores de tablas de traducción**:

- **L**: esperando tablas de traducción.
- **T**: la tabla de traducción enviada por/recibida por CA Business Service Insight.
- **t**: cambios en la tabla de traducción enviada por/recibida por CA Business Service Insight.
- **Indicadores de conexión**:
- <Connecting CA Business Service Insight: intentando conectarse a CA Business Service Insight.
- <Connecting CA Business Service Insight>: conexión establecida.
- <Connecting DataSource: intentando conectarse al origen de datos.
- <Connecting DataSource>: conexión establecida.

- RunOnce: (opcional).

  Valores posibles: [yes/no]

  Cuando se establece en "no", el adaptador se ejecutará de forma continua.

  Cuando se establece en "yes", el adaptador de archivos de texto se ejecuta, lee registros y se detiene automáticamente. El adaptador de archivos lee los archivos enteros, espera unos segundos e intenta leer nuevos registros (dependiendo del valor de SleepTime). Si no hay ningún registro nuevo, se detiene.

  El adaptador de SQL ejecuta cada una de las consultas solamente una vez. Si se define RepeatUntilDry en "no", se detiene inmediatamente. Si RepeatUntilDry se establece en "yes", espera (dependiendo del valor de SleepTime), intenta ejecutar las consultas otra vez (dependiendo del valor SleepTime en la consulta) y, si no hay ningún nuevo registro, se detiene.

- RepeatUntilDry: (opcional).

  Valores posibles: [yes/no]. El valor predeterminado es "yes".

  Consulte el atributo RunOnce arriba.

- RuntimeErrorsLimit: (opcional).

  Determina el límite de errores (por ejemplo los errores en los eventos de entrada), antes de que se bloquee el adaptador. Si es igual a 0, el adaptador no se bloquea.

  Valor predeterminado: 1 (el adaptador se bloquea después del primer error).

- RetryRejectedEvents: (opcional).

  Valores posibles: [yes/no]. El valor predeterminado es "yes".

Si se establece en "yes", se guardan los registros que necesitan traducción en el archivo de eventos rechazado, a la espera de traducción.

No se recomienda establecer este atributo en "no", ya que existe la posibilidad de que se pierdan datos importantes.

- RejectedEventsFileName: (opcional).

  Nombre del archivo que el adaptador utiliza para hacer un seguimiento de los registros de datos recuperados del origen de datos y en espera de traducción.

  Valor predeterminado: Value-Send.txt (si no se especifica lo contrario, se usa el valor predeterminado).

- RejectedEventsUpperLimit: (opcional).

  Determina el límite de registros rechazados que se guardan en el archivo rejected.txt. Cuando el adaptador alcanza este límite superior, se detiene y cambia al estado de bloqueo hasta que se traducen estos registros.

  Valor predeterminado: 1000.

- RegExUnlimitedSearch: indica al adaptador que debe realizar una búsqueda completa según la expresión regular.

  Valores posibles: [yes/no]

  Valor predeterminado: no.

- HoldRejectedEventsSpan: (opcional).

  Determina el número de horas para las cuales guardar el archivo de eventos rechazados. Si no se menciona este atributo, el adaptador no borra ningún evento que se deba gestionar antes de que el adaptador se pueda iniciar otra vez.

  No se recomienda utilizar este atributo, ya que existe la posibilidad de que se pierdan datos importantes.

- GenerateStatistics: (opcional).

  Valores posibles: [yes/no]. El valor predeterminado es "yes".

  Cuando se establece en "yes", el adaptador crea un archivo de estadísticas con información de estadísticas cada minuto.

- StatisticsFileName: (opcional).

  Nombre del archivo de estadísticas.

  Valor predeterminado: statistics.txt (si no se especifica lo contrario, se usa el valor predeterminado).

- KeepHistoricState: (opcional).

Valores posibles: [yes/no]. El valor predeterminado es "no".

Cuando se establece en "yes", el adaptador guarda todos los archivos (excepto el archivo de registro), en un directorio nuevo llamado Historic state aaaammdd-hhmmss, donde aaaammdd y hhmmss corresponden al formato de fecha y hora de creación.

- DefaultTimeFormat: (opcional).

  El formato de hora predeterminado. Si se especifica este atributo, se utiliza como el formato de hora allí donde se haya omitido el atributo TimeFormat. Si no se especifica, el atributo TimeFormat en los otros elementos es obligatorio.

- DefaultDecimalSymbol="." (punto).

  El símbolo decimal predeterminado para los campos de números reales.

  Valor predeterminado: (si no se especifica lo contrario, se usa el valor predeterminado).

- DefaultDigitGroupingSymbol = "," (coma).

  El símbolo de agrupación de dígitos predeterminado para campos de números enteros y reales.

  Valor predeterminado: (si no se especifica lo contrario, se usa el valor predeterminado).

- DataSourceDifferenceFromUTC: indica la diferencia de hora entre UTC y la zona horaria del equipo donde se encuentra el origen de datos.
  - DefaultOffset: variación con respecto la hora UTC fuera del horario de verano.
  - TimeFormat: especifica el formato por el cual se deberán analizar los detalles de horario de verano (descritos a continuación).
  - DayLightSaving: período de horario de verano de la zona horaria del origen de datos. Este elemento es opcional (si no se selecciona, hay horario de verano) y puede existir más que una vez. Cuando hay varios elementos, se deben ordenar por tiempo, y los períodos no se deben solapar.
    - From: fecha de inicio del período.
    - To: fecha de finalización del período.
    - Shift: el cambio de hora añadido a DefaultOffset dentro del período de horario de verano.

Sección Interface de CA Business Service Insight

La sección Interface de CA Business Service Insight está formada por atributos que especifican la conexión a CA Business Service Insight. Hay dos modos: en línea y sin conexión. En el modo en línea, el adaptador se conecta a CA Business Service Insight, obtiene las tablas de traducción y el comando de control de CA Business Service Insight, y envía eventos, estados y solicitudes de traducción a CA Business Service Insight. En el modo sin conexión, el adaptador trabaja con un archivo de tablas de traducción local y escribe los eventos en un archivo de salida.

Estructura XML para el modo en línea:

```
<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: número de puerto TCP/IP que el adaptador utiliza para comunicarse con el servidor de CA Business Service Insight.
- ConnectionInitiator: (opcional).

  Valores posibles: "server" o "adapter". Valor predeterminado: "server".

  Define el iniciador de la conexión, el lado del adaptador o la escucha del adaptador en el lado de CA Business Service Insight.

  El iniciador desempeña el rol de "esclavo" o de cliente, mientras que el otro desempeña el rol de "maestro" o de servidor.
- Address: dirección de red del servidor, obligatoria cuando el iniciador es el adaptador.
- SecurityLevel: define el nivel de autenticación entre el adaptador y el servidor de CA Business Service Insight (protocolo de enlace). Opciones de nivel:
  - none: no se realiza ninguna autenticación.

- high: se realiza la autenticación. Debe indicarse el valor SecurityKey.

- SecurityKey: cadena de 32 dígitos. Debe ser la misma cadena que la definida para el adaptador en la base de datos de CA Business Service Insight.

  El flujo del proceso de "protocolo de enlace" es el siguiente:

  - El servidor de CA Business Service Insight se conecta al adaptador.
  - Se envía una cadena aleatoria desde el adaptador al servidor de CA Business Service Insight.
  - El servidor de CA Business Service Insight cifra la cadena con una clave preconfigurada y envía el resultado al adaptador.
  - El adaptador cifra la misma cadena aleatoria que se envió al servidor de CA Business Service Insight mediante la cadena SecurityKey y compara los resultados.
  - El servidor de CA Business Service Insight elige aleatoriamente una cadena y la envía al adaptador.
  - El adaptador cifra la cadena y la devuelve al servidor de CA Business Service Insight.
  - El servidor de CA Business Service Insight cifra la misma cadena y compara los resultados con lo que se ha recibido del adaptador.
  - Se establece la conexión.
  - Si se detecta cualquier error en alguna de las fases del flujo, no se establece la conexión.

- UseAcknowledgeProtocol: (opcional).

  Valores posibles: [yes/no]. El valor predeterminado es "yes".

  Cuando se establece en "yes", el adaptador utiliza el protocolo de reconocimiento. Cuando se establece en "no", el adaptador envía los mensajes/paquetes y no espera confirmación por parte de CA Business Service Insight.

  No se recomienda establecer este atributo en "no", ya que existe la posibilidad de que se pierdan eventos.

- PacketSize: (opcional).

  Valores posibles: de 1 a 1000.

  Valor predeterminado: 50.

  Número máximo de eventos en un paquete.

- PacketDeadline: (opcional).

  Valores posibles: de 1 a 3600.

Valor predeterminado: 60.

Número de segundos antes de enviar un paquete que no está lleno.

- PacketResendTimeout: (opcional).

  Valores posibles: de 1 a 3600.

  Valor predeterminado: 60.

  Cantidad de tiempo en segundos que se debe esperar a recibir un mensaje de confirmación antes de enviar el paquete de nuevo. Este atributo es aplicable solamente cuando el atributo UseAcknowledgeProtocol se establece en "yes".

- WindowSize: (opcional).

  Valores posibles: de 1 a 100.

  Valor predeterminado: 10.

  Número de paquetes en ventana. Este atributo es aplicable solamente cuando el atributo UseAcknowledgeProtocol se establece en "yes".

Estructura XML del modo sin conexión:

```
<OblicoreInterface
        Mode="offline">
        <OfflineInterface
                OutputFileName="outputEvents.txt"
                OutputFileType="tsv"
                OutputTimeFormat="%Y/%m/%d %H:%M:%S"
                OutputDecimalSymbol="."
        />
</OblicoreInterface>
```

- OutputFileName: (opcional).

  Nombre del archivo donde el adaptador escribe los eventos de salida.

  El valor predeterminado es Value-AdapterOutput.txt (si no se especifica lo contrario, se usa el valor predeterminado).

- OutputFileType: (opcional).

  Valores posibles: csv/tsv/xml.

  Define el formato del evento.

- OutputTimeFormat: define el formato de los campos de fecha. Este atributo se puede omitir si se ha definido el atributo DefaultTimeFormat en la sección General.

- OutputDecimalSymbol: (opcional).

  Define el símbolo decimal para campos de números reales.

Valor predeterminado: . (punto).

Sección DataSourceInterface

La sección DataSourceInterface está formada por atributos que especifican la conexión y el tipo de conexión entre el adaptador y el origen de datos (herramienta de medida, CRM, registro del sistema, etc.) y está dividido en dos tipos principales: interfaz de archivo e interfaz SQL.

Interfaz de archivo

El adaptador de archivos se puede usar para recuperar datos de archivos de registro, informes programados o cualquier otro texto basado en archivos, y DataSourceInterface define las reglas para analizar la información del origen de datos del archivo y extraerla en campos.

La sección DataSourceInterface también define cómo gestiona el adaptador el archivo de origen (si suprime el archivo original si sólo lo creó el adaptador o si deja los datos por si hacen falta para otros usos, etc.).

Estructura 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). Cuando se fija DeleteFileAfterProcessing como "no", el archivo se copia a este nombre; cuando se establece como "yes", se cambia el nombre del archivo a este nombre. Si no se especifica, se tomará el valor predeterminado ('WorkFile.txt').
  - Files: recopilación de los elementos de archivo (se puede definir más de un archivo en un adaptador).
  - File: especifica los atributos de archivo.
    - IsActive: (opcional) [yes/no]. Define si este archivo está activo (cuando se establece como "no" este archivo no se leerá).
    - InputFormat: el InputFormat asociado con este archivo. El adaptador utiliza el valor de InputFormat para extraer los datos del origen de datos.
    - Path: la ruta de la ubicación de los archivos de datos de origen.
    - NamePattern: especifica el nombre de archivo del origen de datos. Se pueden utilizar caracteres comodín si más de un archivo utiliza el mismo formato de entrada.
    - DeleteFileAfterProcessing [yes|no]: la forma en la que el adaptador maneja el archivo de origen. Cuando el archivo se creó sólo para el adaptador y pueda suprimirse después del procesamiento, hay que establecerlo como "yes". Se cambia el nombre al archivo, se procesa y, a continuación, se suprime. Cuando se establece como "no" el archivo se copia y el procesamiento tiene lugar en el archivo copiado. Si se añaden nuevos registros al final de este archivo, el adaptador copia estos nuevos registros al archivo de trabajo durante el ciclo siguiente. Si no se añaden registros nuevos al archivo, el adaptador busca el primer archivo con el mismo patrón y nombre (en orden lexicográfico) que el archivo actual. Si el adaptador encuentra dicho archivo, empieza a funcionar con este. El adaptador no vuelve al archivo anterior, ni siquiera si se añaden nuevos registros a este archivo. Defínalo en "no" cuando sea necesario mantener la integridad del archivo de origen.
    - InitialFileName: el primer nombre de archivo desde el que el adaptador empezará a buscar el archivo con el patrón determinado. Utilice este atributo cuando el campo NamePattern contiene caracteres comodín y no desea que el adaptador lea archivos antiguos.
- Delimiters: (opcional). Uno o más caracteres que sirven de delimitadores, según los cuales las filas datos sin procesar se analizarán en campos; si no se especifica, el valor predeterminado es "\t".

- IgnoreRedundantDelimiters: (opcional) [yes/no]. Cuando se establece como "yes" los delimitadores consecutivos se tratarán como uno, y se creará un campo vacío entre los delimitadores.
- RegExForParser: (opcional). Una expresión regular que se utiliza para extraer campos del origen de datos que sustituye a los valores de Delimiters especificados anteriormente. Por ejemplo:

```
<File
    ….
    RegExForParser="^(.{8}) (.{6}) (?:[ ])*([0-9]+) "
/>
```

  Consulte la documentación de las expresiones regulares para obtener más información.

- TitleExists: (opcional) [yes/no] especifica si la primera línea del archivo de origen de datos que no esté en blanco es una línea de título. El adaptador puede utilizar el título al analizar los datos.
- SleepTime: especifique el intervalo de datos de recuperación (en segundos); es decir, el número del intervalo en segundos entre la extracción de datos del adaptador desde el archivo de datos de origen.
- LogicLineDefinition: (opcional)

```
<File
    ….
    <LogicLineDefinition
                    FirstLine="Job server:.*"
                    NumberOfLines="5"
    />
/>
```

  Si el conjunto de datos se construye a partir del número de líneas y no por una línea, los atributos siguientes definen el punto de partida y punto final de la extracción y el número de líneas que componen los datos:

  - AllFile: (opcional) [yes/no]. Cuando se establece como "yes" todo el archivo se considera como un registro, una línea lógica.
  - FirstLine: (opcional). Expresión regular que define la primera línea de la línea lógica. Puede especificarse con o sin LastLine o NumberOfLines.
  - LastLine: (opcional). Expresión regular que define la última línea de la línea lógica. Se puede especificar con o sin FirstLine o NumberOfLines.
  - NumberOfLines: (opcional). Número de líneas en una línea lógica. Se puede especificar con o sin FirstLine o LastLine.
  - MatchCase: (opcional) [yes/no]. Define si la expresión regular distingue entre mayúsculas y minúsculas.

Sección de la interfaz de SQL

Se pueden utilizar los adaptadores de SQL para recuperar datos de bases de datos mediante una declaración de SQL.

La interfaz de SQL define la conexión con la base de datos y las consultas que se utilizan para recuperar los datos de la siguiente manera:

Estructura XML:

```
        < DataSourceInterface >
                <ConnectionString ConnectionTimeout="60" QueryTimeout="30">
                        <![CDATA[ Driver={Microsoft Access Driver
(*.mDataBase)};DataBaseq=d:\Oblicore\database1.mdatabase; ]]>
                        </ConnectionString>
                <QueryCollection>
                <Query QueryName="cases" InputFormat="cases" SleepTime="3600">
                        <SelectStatement AutoCompleteQuery="yes">
                        select
dateclosed,callid,dateopened,companyname,priority,closedmn,responsemn
                        from calls where dateclosed is not NULL
                        </SelectStatement>
                <QueryKeyFields>
                        <KeyField Name="dateclosed" Sort="asc"/>
                        <KeyField Name="callid" Sort="desc"/>
                        <SelectInitialValues>
                                Select min(dateclosed) , 'min date' from calls
                        </SelectInitialValues>
                </QueryKeyFields>
        </Query>
        <Query QueryName="contracts" InputFormat="contracts" SleepTime="3600">
        <ConnectionString>
                <Segment Type="text"
                Text=" Driver={Microsoft Excel Driver (*.xls)}; DriverId=790;
DataBaseq="/>
                <Segment Type="File">
                <File Path="d:\Oblicore " NamePattern="Availabilty_*.XLS>
                </Segment>
                <Segment Type="text" Text=";"/>
        </ConnectionString>
<SelectStatement AutoCompleteQuery="yes">….</SelectStatement>
<QueryKeyFields>…..</QueryKeyFields>
</Query>
</QueryCollection>
</DataSourceInterface>
```

- ConnectionString: cadena de conexión para acceder a la base de datos de origen.

ConnectionString se puede definir en el elemento DataSourceInterface o en los elementos Query. La definición de ConnectionString en el elemento DataSourceInterface es el valor predeterminado y se utiliza solamente en consultas donde no se ha definido ConnectionString.

La cadena de conexión puede definirse en una cadena o por segmentos. Cuando el elemento de ConnectionString no contiene elementos de segmento, se toma la cadena de conexión del texto de elemento de ConnectionString. Si contiene al menos un elemento de segmento, la cadena de conexión se concatena a partir de este.

Hay dos tipos de segmentos. El primero es texto y contiene texto que se concatena a la cadena de conexión tal cual. El segundo tipo es un archivo y contiene un nombre de archivo con o sin caracteres comodín. El segmento de archivo puede aparecer solamente una vez y contiene otros atributos que definen lo que hay que hacer con el archivo de lectura.

- ConnectionTimeout: (opcional). Un número entero positivo que indica, en segundos, el tiempo que se debe esperar para que se abra la conexión. El valor 0 indica que se espera de forma indefinida hasta que se abra la conexión. El valor predeterminado es 15 segundos.

Nota: algunos proveedores no admiten esta funcionalidad.

- QueryTimeout: (opcional). Un número entero positivo que indica, en segundos, el tiempo que se debe esperar para que se ejecute la consulta. El valor 0 indica que se espera de forma indefinida hasta que finalice la ejecución. El valor predeterminado es 30 segundos.

Nota: algunos proveedores no admiten esta funcionalidad.

- Segment: especifica los atributos del segmento.
- Type: (opcional) (text, file). Tipo de segmento.
- Text: el texto del segmento de texto.
- File: especifica los atributos del archivo
- Path: la ruta de la ubicación de los archivos de datos de origen.
- NamePattern: especifica el nombre del archivo de datos del origen, se pueden utilizar caracteres comodín.
- InitialFileName (opcional). Dice al usuario desde qué archivo debe iniciarse la búsqueda y el patrón que debe buscarse.
- UsePath: (opcional) [yes, no]. Si se establece como "yes", la ruta de archivo se concatena a la cadena de conexión.

- UseFileName: (opcional) [yes/no]. Si se establece como "yes", el nombre del archivo se concatena a la cadena de conexión (necesaria para los archivos de Excel).
- UpdateSchemaFile: (opcional) [yes/no] Si se establece en "yes", el adaptador actualiza el archivo schema.ini con el nombre de archivo actual.

Nota: Utilice este atributo solamente cuando desee que el adaptador cambie el archivo schema.ini (base de datos o archivos de texto).

- QueryCollection: recopilación de consultas (se puede definir más de una consulta en un adaptador).
  - PreliminaryCheck: (opcional [yes/no]). El adaptador comprueba las consultas cuando se conecta a la base de datos. Si este atributo se establece como "no" el adaptador comprueba las consultas ejecutándolos sin ningún cambio, el adaptador se ejecuta en los registros de retorno más tarde y no los lee otra vez. Cuando se establece como "yes", el adaptador sustituye cada cadena "where" en la declaración "where 1=2 and" y ejecuta las consultas en ese momento. Utilice esta opción si desea comprobar todas las consultas antes de que el adaptador empiece a ejecutar los registros y cuando esta agregación acorte en gran medida el tiempo de consulta. Nota: Algunos de los proveedores no optimizan el proceso de consulta y cuando la consulta es legal la ejecución tarda el mismo tiempo de lo que tardaría sin la agregación.
  - ExternalCommand: (opcional). El comando externo que se ejecuta antes de que el adaptador inicie un nuevo ciclo de lectura.
  - Command: nombre del archivo por lotes que se ejecutará.
  - SleepTime: número entero positivo que indica, en segundos, cuánto tiempo hay que esperar antes de ejecutar el comando otra vez.
- Query: especifica los atributos de la consulta.
  - QueryName: nombre de la consulta.
  - IsActive: (opcional) [yes/no]. Cuando se establece como "no" el adaptador no lee los registros de este archivo.
  - InputFormat: InputFormat está asociado con esta consulta. El adaptador utiliza el valor de InputFormat para extraer los datos del registro de origen.
  - SleepTime: tiempo en segundos durante los cuales el adaptador permanece inactivo a la espera de que lleguen nuevos datos.

- Critical: (opcional) [yes/no]. Si se establece como "yes" el adaptador se detiene inmediatamente cuando encuentra un error en la consulta. La opción "no" se utiliza cuando es un error conocido que se resuelve sin cambiar el archivo de configuración.
- TransactionMode: (opcional) [implicit/explicit]. Si se establece como "explicit", el adaptador inicia una transacción de la base de datos antes de la consulta. Esta opción resuelve varios problemas en consultas enormes y complicadas.
- RollbackSegementName: (opcional). Define los segmentos de reversión que utiliza el adaptador. De lo contrario, la base de datos elige el segmento de reversión.

- SelectStatement: contiene la declaración seleccionada para ejecutar en la base de datos de origen.
  - AutoCompleteQuery: (opcional) [yes/no]. Si se establece en "yes", el adaptador concatena automáticamente una declaración "where" a la consulta especificada, que hace lo siguiente:
  - Creación de una declaración "where" que recupera solamente valores nuevos basados en los campos clave.
  - Ordenando la declaración de resultados según los campos clave.
- QueryKeyFields: define los campos donde iniciar la siguiente recuperación de datos:
  - KeyField:
  - Name: nombre del campo que se toma de los campos de la consulta.
  - Sort: (opcional) [asc/desc]. Método de clasificación de datos (ascendente/descendente).
  - Function: (opcional). Utilice este atributo si debería operar una función especial en este campo, para combinar el valor del campo en el uso de la función (:fieldname). Por ejemplo utilizando la función de fecha de Oracle con un nombre de campo Ts-Function="to_date(':ts','dd/mm/yyy')"
  - ValueLeftWrapper: (opcional). Utilice este atributo para concatenar caracteres antes del valor del campo. El valor predeterminado para la cadena y los campos de fecha es "'" (apóstrofo).
  - ValueRightWrapper: (opcional). Utilice este atributo para concatenar caracteres después del valor del campo. El valor predeterminado para la cadena y los campos de fecha es "'" (apóstrofo).

- NameLeftWrapper: (opcional). Utilice este atributo para concatenar caracteres antes del nombre del campo. El valor predeterminado es una cadena vacía.
- NameRightWrapper: (opcional). Utilice este atributo para concatenar caracteres después del nombre del campo. El valor predeterminado es una cadena vacía.
- IsPrimaryKey: (opcional) [yes/no]. Si se establece como "no", el adaptador no utiliza esta clave en la declaración "where" automática de la consulta.
- DefaultInitialValue: (opcional). Si la consulta SelectInitialValues no devuelve registros, el adaptador toma de aquí el valor predeterminado. Si hay algunos campos clave, todos los campos clave deben tener este atributo.
- SelectInitialValues: una declaración de selección que proporciona los valores iniciales para la primera declaración "where" (cuando el archivo de control está vacío).

Nota: El orden de los valores debe coincidir con el orden de los elementos de Field para este elemento QueryKeyFields y devolver un resultado para cada campo.

Sección InputFormatCollection

Esta sección especifica la estructura de los datos recuperados desde el origen de datos, cómo se dividirán las filas de datos en campos y qué tipos de campos y formatos hay. En esta sección se puede hacer un filtrado y una manipulación inicial de los datos mediante los campos InputFormatSwitch y Compound respectivamente.

El flujo de trabajo general de esta sección es el siguiente:

- La fila de datos coincide con respecto a uno o más InputFormats.
- Los datos se diseccionan en campos siguiendo la especificación InputFormat correspondiente.
- Se asignan valores a los campos compuestos combinando y dividiendo los campos de datos.
- Los datos procesados se comprueban con las condiciones de TranslatorSwitch.
- Los datos procesados se envían al traductor coincidente o se ignoran.

El nodo InputFormatCollection puede contener uno o más nodos InputFormat.

Estructura XML:

```
<InputFormatCollection>
        <MyInputFormat InputFormatName="InputFormat">
        <InputFormatFields>
                <InputFormatField Name="sid_id" Type="string"/>
                <InputFormatField Name="content" Type="string"/>
                <InputFormatField Name="date" Type="time"
               TimeFormat="%d/%m/%Y %H:%M:%S"/>
                <InputFormatField Name="server" Type="string"
               Source="compound">
                <Compound>
                        <Segment SourceField="content"
                    RegularExpression=".*Job server: ([^\n]+).*" />
                </Compound>
                </InputFormatField>
        </InputFormatFields>
<TranslatorSwitch DefaultTranslator="GeoTranslator">
        <TranslatorCase  TranslatorName="NonGeoTranslator" Break="yes">
                <Condition SourceField="routing_info" Operator="EQ"
             Value="cnano"/>
        </TranslatorCase>
</TranslatorSwitch>
</InputFormat>
</InputFormatCollection>
```

- InputFormat:
  - InputFormatName: nombre para este formato al que hará referencia la sección DataSourceInterface.
- RequiredFields: (opcional) Número mínimo de los campos que se espera encontrar en una fila de datos. Se ignora la fila que contiene menos campos y se registra un error.
- InputFormatFields: InputFormatFields puede contener uno o más nodos de campos, según el número de campos de entrada en el origen de datos.
  - InputFormatField: especifica un campo de datos de la fila de datos de origen, o un campo compuesto.

    ```
    <InputFormatField Name="timestamp" Type="time"
    TimeFormat="%d/%m/%Y %H:%M:%S"/>
    ```
  - Name: un nombre para este campo, para que otros elementos puedan hacer referencia a él.
  - Type: tipo de datos del campo (cadena, entero, hora, real).
  - Source: (opcional). El valor predeterminado que se toma es "event", los posibles valores son:
  - event: el valor del campo se toma del evento que viene del origen de datos, los valores de los campos se toman en el mismo orden en el que venían del origen de datos.
  - compound: es un campo compuesto. Toma este valor después de hacer algún tipo de manipulación en otros valores de campos o constantes.
  - title: el valor del campo se toma de los nombres del campo de título. El campo de referencia ya debe haberse definido.
  - filename: el valor del campo se toma del nombre de archivo del origen de datos; solamente para adaptadores de archivos de texto.
  - constant: el valor del campo es constante y se toma de la propiedad ConstantValue, que deberá aparecer a continuación.
  - FieldTitleName: cuando el origen es un título, especifica el título del campo que se tiene que utilizar. El campo de origen se deberá definir antes.
  - ConstantValue: cuando un origen es constante, especifica el valor con el que tiene que coincidir. Cuando el tipo de campo es de tiempo, el valor constante es una cadena con formato según TimeFormat o "Now" o "NowUtc", donde "Now" es la hora en curso en la configuración regional del origen de datos y "NowUtc" es la hora en curso en UTC.
  - TimeFormat: por el cual se analiza el campo. Los siguientes códigos de carácter se pueden utilizar para el formato de fecha y hora:

Signo	Descripción
%d	Día 1-31
%a	Nombre abreviado del día de la semana (Lun., Mar., etc.)
%A	Nombre completo del día de la semana (Lunes, Martes, etc.)
%m	Mes 1-12
%b	Nombre abreviado del mes (Ene., Feb., etc.)
%B	Nombre completo del mes (Enero, Febrero, etc.)
%y	Año (2 dígitos)
%Y	Año (4 dígitos)
%H	Hora en formato de 24 horas (00- 23)
%I	Hora en formato de 12 horas (01- 12)
%p	Indicador AM/PM para el formato de 12 horas
%M	Minutos 0-59
%S	Segundos 0-59
%D	Fecha completa sin separadores (ddmmyyyy)
%T	Hora completa sin segundos ni separadores (hhmm)
%n	Número de segundos desde 1970
%E	Fecha y hora en formato hexadecimal
%%	Signo de porcentaje

  - DecimalSymbol: (opcional). Es el símbolo decimal para los campos. Si no se especifica, se toma DefaultDecimalSymbol de la sección General.
  - DigitGroupingSymbol: (opcional). El símbolo de agrupación de dígitos predeterminado para los campos de números enteros y reales. Si no se especifica, se toma DefaultDigitGroupingSymbol de la sección General.

- Compound: obligatorio cuando source=compound. Especifica las manipulaciones de campo que se tienen que recopilar en un campo compuesto.
  - Segment: especifica una manipulación de campo que se tiene que agregar al compuesto creado. Solamente es obligatorio el atributo SourceField.

- SourceField: el campo debe basarse en este campo. El campo de referencia ya debe haberse definido.
- RegularExpression: expresión regular que manipula.
- MatchCase: (opcional) [yes/no]. Defina si la expresión regular distingue entre mayúsculas y minúsculas.
- SelectionStart: posición de inicio de extracción del texto, empieza en 0.
- SelectionLength: tamaño de la extracción del texto.
- Prefix: una cadena que debe usarse como prefijo para el resultado de manipulación.
- Suffix: una cadena que debe usarse como sufijo para el resultado de manipulación.
- XpathExpression: expresión xpath que manipula.

- InputFormatSwitch: se utiliza para especificar los criterios de formato, cuando hay filas de datos con más de un formato.

  Nota: Al utilizar InputFormatSwitch, el orden de los nodos de InputFormat es importante. Se debe haber definido ya un InputFormat de referencia.

  DefaultInputFormat: especifica el nombre de InputFormat al que se tiene que enrutar, en caso de que no haya ningún criterio que coincida.
- InputFormatCase: especifica un criterio que hay que probar en las filas de datos para determinar a qué InputFormat se debería enrutar.
- InputFormatName: el InputFormat que se tiene que utilizar cuando los criterios coinciden.
- LogicOperator: (opcional) [and/or].
- and: todas las condiciones deben coincidir (valor predeterminado).
- or: como mínimo una condición debe coincidir.
  - Condition: condición que hay que probar en una fila de datos para determinar su formato.

    SourceField: campo que hay que probar.

    Operator: tipo de prueba, puede ser una de las opciones siguientes:
    - **EQ**: igual a
    - **NE**: no igual a
    - **GT**: mayor que
    - **LT**: menor que
    - **GE**: mayor o igual a

- **LE**: menor o igual a
- **MATCH**: una expresión regular debe coincidir
- **UNMATCH**: una expresión regular no debe coincidir

ValueType: (opcional) [constant/field/previousValue]

- constant: el contenido del atributo Value es una constante independientemente de los datos de origen.
- field: el contenido del atributo Value es el nombre de un campo del mismo registro.
- previousValue: el contenido del atributo Value es el nombre de un campo del registro anterior de la misma consulta con el mismo formato de entrada.

Value: valor con el que se debe coincidir o una expresión regular.

MatchCase: (opcional) [yes/no]. Define si las pruebas distinguen entre mayúsculas y minúsculas. Cuando no está establecido como "yes", los dos valores se traducen a minúsculas antes de la prueba.

- TranslatorSwitch: determina qué traductor se utiliza para traducir la fila de datos a un evento unificado de CA Business Service Insight.
  - DefaultTranslator: traductor que se debe utilizar cuando no hay criterios coincidentes. Si el valor se establece en "Ignore", no se usará ningún traductor, y la línea se ignorará.
  - TranslatorCase: especifica los criterios que hay que probar en los datos para determinar a qué traductor se debería enrutar.

    Break [yes|no]

    yes: (valor predeterminado). Si hay criterios coincidentes, no se comprueban los criterios siguientes.

    no: en cualquier caso, después de evaluar los criterios y hacer funcionar al traductor si hay coincidencias, se procede a los criterios siguientes.

    LogicOperator: (opcional) [and/or].

    - and: todas las condiciones deben coincidir (valor predeterminado).
    - or: como mínimo una condición debe coincidir.

    TranslatorName: el traductor al que hay que redireccionar si se cumplen las condiciones.

    Condition: la condición que hay que comprobar en los datos procesados para determinar el traductor relevante que se debe utilizar con ella. Es igual que Condition en InputFormatSwitch.

Sección TranslationTableCollection

La sección contiene tablas de asignación de los valores de origen de datos "in" a los campos de evento de CA Business Service Insight.

Estructura 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 el nombre del archivo en el que las tablas se almacenan localmente, si no se especifica se toma el valor predeterminado ("Translation.XML").
- LoadingMode: (opcional) [standalone, remote].

  Nota: El valor predeterminado para la interfaz en línea es "remote", y para la interfaz sin conexión es "standalone". El método de carga de las tablas de traducción se especifica como sigue:

- standalone: el adaptador carga las tablas de traducción localmente. No hay conexión con el servidor de CA Business Service Insight en cuanto a la traducción. Los cambios en las tablas de traducción se almacenan solamente en el archivo local.
- remote: el adaptador envía una solicitud para cargar todas las tablas desde el servidor de CA Business Service Insight. Los cambios realizados en las tablas de traducción se almacenan también localmente.
- LoadTimeout: (opcional). Cuando el modo de carga es "remote" se puede especificar aquí un tiempo de espera en segundos.
  - TranslationTable: vincula el valor del evento con la tabla de asignación.

- Name: nombre que utilizará el traductor y hará referencia a este. El nombre legal empieza con una letra o un guión bajo y contiene letras, dígitos y guiones bajos".
- DestinationType: [resource, event_type, contract_party, service, time_zone, value].
- ValueType: (opcional) [integer, real, string]. El tipo de valor que se devuelve de la tabla. La cadena y los valores "real" son legales solamente para tablas con DestinationType="value"
- TranslationField: nombre del campo que hay que traducir, el nombre del campo se toma de los campos de formato de entrada. Se pueden tener hasta 5 campos.

Sección TranslatorCollection

La sección TranslatorCollection describe cómo traducir el registro del origen de datos analizado y manipulado extraído en la sección anterior a un evento de CA Business Service Insight.

Cuando el modo de interfaz se ha establecido como en línea ("online"), el evento de CA Business Service Insight tiene una estructura unificada que contiene los campos siguientes:

- Timestamp: hora en la que se produjo el evento.
- ResourceId: el ID de recurso de CA Business Service Insight asociado al evento (el recurso que se midió, etc.).
- EventTypeId: el tipo de evento de CA Business Service Insight asociado al evento y que describe el tipo de evento (tipo de medición del recurso, tipo de acción de ticket, etc.).
- DataSourceId: (opcional). El nombre del origen de datos de entrada (nombre de archivo / nombre de tabla...).
- Value: el valor del evento (resultado de medida, número de ticket, etc.). Este campo puede aparecer más que una vez.

Cuando el modo de interfaz es "offline", no hay limitaciones en el número de los campos o en su nombre.

Estructura 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>
```

- Translator: describe cómo traducir el conjunto de campos recibidos en el evento de salida.
  - TranslatorName: nombre utilizado por InputFormat para enviar conjuntos de campos a este traductor.
  - OnDuplication: miembro que guarda el valor "ignore", "add", "update" o "updateAlways" para determinar lo que hacer con el evento de duplicación (consulte Singularidad del evento).
  - TranslatorFields: contiene una lista de elementos TranslatorField, cada uno de los cuales contiene los atributos siguientes:
  - Name: nombre del campo. En la interfaz en línea debe ser Timestamp, ResourceId, EventTypeId,Value o DataSourceId.
  - SourceType:

    field: el valor de este campo se toma del campo en el formato de entrada. El atributo SourceName contiene el nombre del campo.

    table: el valor del campo se toma de la tabla de traducción. El atributo SourceName contiene el nombre de tabla.

    lookup: el valor de este campo se toma de la tabla de traducción. El atributo SourceName contiene el nombre de tabla. El valor que se tiene que traducir se toma del atributo LookupValue y no del formato de entrada.

    constant: el valor del campo es constante, y su valor está en el atributo ConstantValue.
  - SourceName: contiene el nombre del campo o el nombre de la tabla de traducción.
  - Type: [integer/real/string/time]. Necesario solamente cuando el tipo del campo no se ha definido previamente (por nombre de campo o por tipo de origen). En la interfaz en línea se requiere solamente para el campo Value con SourceType=constant. En la interfaz sin conexión se requiere para cada campo con SourceType=constant.
  - IsKey: representa la clave única del evento. Esta clave se construye a partir de los campos marcados como TranslatorFields?IsKey = "yes".

    (Consulte Singularidad del evento).
  - LookupValue: contiene el valor de búsqueda cuando SourceType=lookup.

- ConstantValue: contiene el valor constante cuando SourceType=constant. Cuando el tipo de campo es de tiempo, el valor constante es una cadena con formato según TimeFormat o "Now" o "NowUtc", donde "Now" es la hora en curso en la configuración regional del origen de datos y "NowUtc" es la hora en curso en UTC.
- TimeFormat: contiene el formato de hora, necesario solamente para campos con SourceType=constant y Type=time.
- TimeShift: define la variación de tiempo en segundos, solamente para campos de hora.
- TimeShiftFieldName: (opcional). Contiene el nombre del campo, del formato de entrada, que contiene la variación de la hora en segundos. Los atributos TimeShift y TimeShiftFieldName pueden ir juntos.

Apéndice D: Definición de fórmulas de lógica de negocios (Experto en la lógica de negocios)

Esta sección contiene los siguientes temas:

Qué evitar al crear fórmulas de lógica de negocios (en la página 397)
Métricas en clúster y efectividad de recursos (en la página 397)

Qué evitar al crear fórmulas de lógica de negocios

Al crear fórmulas de lógica de negocios, asegúrese de tener en cuenta lo siguiente:

- Nunca asigne un valor nulo a una variable global. Al asignar un valor nulo, puede hacer que el PSLWriter genere un error al calcular la métrica. Si es necesario un valor no inicializado, utilice "empty" mejor.
- Evite utilizar objetos de vectores y de asignación en métricas en clúster. Si se deben utilizar estos objetos, haga que tengan el menor tamaño posible. Las métricas en clúster con grandes asignaciones y vectores requieren mucho tiempo para calcularse.

Métricas en clúster y efectividad de recursos

¿Qué es una métrica en clúster?

La métrica en clúster permite ejecutar una definición de métrica para cada miembro individual de un grupo de recursos con objeto de aplicar la misma definición y la misma lógica para un conjunto de elementos. Las agrupaciones en clúster se pueden definir de manera estática en un conjunto predeterminado de recursos o de forma dinámica en los miembros del grupo de recursos, mientras que el grupo se puede cambiar con el tiempo, así como incluir o excluir miembros del grupo.

Se puede excluir un recurso o grupo de recursos e incluirse en el grupo con el tiempo, e incluso se puede excluir del grupo y volver a incluirse una y otra vez en el mismo período de cálculo (día, mes, año, etc.).

¿Qué sucede en la lógica de negocios cuando un elemento de clúster se elimina del grupo base de clúster?

Se activan el método OnPeriodEnd y la función Result para el elemento de clúster. Si esto sucede en medio del período de cálculo, el resultado se escribe solamente en la base de datos cuando el período de cálculo original ha finalizado (p. ej., final del mes, final del año).

¿Qué sucede en la lógica de negocios cuando un elemento de clúster se une al grupo base de clúster?

Las variables globales se inician, se activan los métodos OnLoad, OnRegistration y OnPeriosStart para el elemento de clúster.

¿Qué sucede en la lógica de negocios cuando un elemento de clúster se une al grupo base de clúster después de que se haya eliminado del grupo en el mismo período de cálculo?

El resultado que se estableció para el período durante el cual el elemento de clúster formaba parte del grupo se anula con el nuevo resultado. Dicho de otra manera, el resultado al final del período de cálculo se refiere solamente al último período del período calculado en el que el elemento de clúster forma parte del grupo.

¿Cuál es el efecto del atributo vigente de un recurso en la lógica de negocios?

Durante el tiempo que un recurso no es vigente, no se recopila ningún dato sin procesar para el recurso que no es vigente.

¿Cuál es el efecto del atributo vigente de un recurso en el agrupamiento en clúster?

Si cambiamos un recurso para que no sea vigente, tiene el mismo impacto en el agrupamiento en clúster como la exclusión del recurso del grupo. El agrupamiento en clúster se comporta de la misma manera tanto para la efectividad de recursos como para la pertenencia a un grupo.

¿Cómo debería implementar las excepciones en un recurso? ¿Está utilizando la efectividad de recursos el método correcto?

Hay algunos casos de negocio en los que un período de excepción se debería establecer en un recurso individual; por ejemplo, es posible que un servidor se someta a mantenimiento y que se omita de los cálculos para el período de mantenimiento.

Puesto que la lógica de negocios ignora los eventos de datos sin procesar de un recurso no vigente, puede elegir implementar excepciones en un recurso mediante el mecanismo de vigencia. Puede funcionar en algunos casos. Sin embargo, si el recurso forma parte de una métrica en clúster y el recurso va a entrar en vigencia y dejar de estar vigente en el mismo período de cálculo, solamente se tendrá en cuenta en el resultado el último período en el que el recurso estuvo vigente, como se ha indicado anteriormente. En ese caso, se aconseja utilizar la función de atributos personalizados. Un atributo adicional para el recurso que indica que el estado del recurso se puede gestionar. La fórmula de lógica de negocios consulta entonces el estado del recurso siempre que sea relevante en el script.

Glosario

Adaptador

La interfaz entre CA Business Service Insight y fuentes de datos de terceros. Los adaptadores traducen datos obtenidos de estas fuentes de datos a un formato que se puede utilizar para realizar cálculos a nivel del servicio que se proporcionan a las partes contratantes.

Adjudicación de recursos

La asignación de un recurso a un servicio dirige el flujo de eventos del recurso a este servicio. Un recurso se puede adjudicar a un servicio, parte contratante, tipo o grupo.

Agente

Un objeto que representa una métrica en una unidad de tiempo dada.

Alerta

Notificaciones a usuarios sobre eventos que tienen lugar en el sistema. Las alertas permiten a los usuarios realizar acciones correctivas que eviten desviarse de contratos, incurriendo en sanciones entre otros.

Anotaciones de los eventos

Información adicional acerca de un evento específico. Las anotaciones de los eventos se establecen de forma automática o manual (dentro de los informes).

Asistente de informes

Interfaz gráfica de usuario (GUI) que se utiliza para definir los parámetros de los informes.

Asistente de métrica

Una interfaz fácil de usar desarrollada para permitir a los usuarios modificar fácilmente métricas la primera vez que las métricas se agregan al contrato a partir de una plantilla de nivel de servicio.

Autor del contrato

El usuario responsable de crear un determinado contrato.

Carpeta de informes

Carpeta de informes es un contenedor que se usa para agrupar informes relacionados entre sí de un modo u otro.

Catálogo de servicios

Una recopilación de toda la información acerca de servicios, dominios y unidades, la biblioteca de plantillas y las plantillas de nivel de servicio en CA Business Service Insight.

Categoría del dominio

Un aspecto concreto del nivel del servicio acordado en el contrato. Por ejemplo, podría tener un dominio del servicio llamado Help Desk y una categoría del dominio como Tiempo de respuesta del Help Desk, que describe ese aspecto particular de servicio de Help Desk. El administrador del sistema del proveedor de servicios normalmente define las categorías del dominio.
CA Business Service Insight también proporciona categorías del dominio predeterminadas.

Comentario de causa raíz

Un conjunto de comentario dentro de la lógica de negocios que explica los resultados a nivel del servicio. Los comentarios de causa raíz se asocian con métricas.

Comentario manual de causa raíz

Un comentario, creado de forma manual por quien visualiza el informe, para explicar o agregar información adicional a un resultado de nivel de servicio de métrica específica. Los comentarios manuales de causa raíz pueden compartirlos todos aquellos que visualizan los informes de la misma métrica.

Conjunto de cambios

Un conjunto de cambios a la asignación de recursos del proveedor de servicios.

Consumo

Un tipo de métrica que calcula el consumo. Se utiliza principalmente para el módulo financiero. Mediante este tipo de métrica, es posible consultar en los informes el consumo y los precios por separado. El consumo se puede calcular también en la métrica Elemento del precio.

Contrato archivado

Un contrato cuyos datos se han trasladado al archivo. Estos contratos no se incluyen en cálculos y no se pueden consultar en informes.

Contrato de soporte

Contrato de soporte. Un contrato con un proveedor externo que abarca la prestación de servicios que respalden la prestación de servicios por parte de la organización.

Cuadro de mandos

Una interfaz de usuario destinada a gestores de alto nivel que organiza y muestra información en tiempo real e informa en una manera fácil de leer.

Declaración del objetivo

La descripción lógica de una métrica que contiene los parámetros que definen la métrica. Las declaraciones de los objetivos se muestran en la interfaz gráfica de usuario de CA Business Service Insight para proporcionar captura efectiva de la esencia de la métrica.

Dispositivo de alerta

Dispositivos de la red que reciben mensajes de alerta mediante correo electrónico, elementos emergentes o SMS, o mediante un programa de terceros.

Elemento del precio

Una métrica que calcula los precios. Los precios pueden estar basados en el consumo o estar definidos como precios fijos.

Entrada de traducción

Representa una definición de valores de origen y destino utilizados por la tabla de traducción.

Evento de datos

Eventos generados por adaptadores de CA Business Service Insight

Evento de datos sin procesar

Un evento generado por datos sin procesar en CA Business Service Insight.

Evento de métrica

Un evento generado por una métrica en CA Business Service Insight.

Eventos recibidos

Una lista de eventos recibidos de otras métricas mostradas en la ventana Ámbito de lógica empresarial al seleccionar en la ficha Eventos recibidos.

Excepción

Período de tiempo que no cuenta para el cálculo de niveles de servicio. Por ejemplo, tiempo de inactividad.

Fecha de entrada del recurso

La fecha en la que se introdujo el recurso en CA Business Service Insight. No se debe confundir con la fecha de vigencia del recurso.

Fecha de vigencia del recurso

La fecha desde la que CA Business Service Insight puede utilizar el recurso. La fecha de vigencia del recurso se define mediante la versión del recurso que incluye el recurso. Ningún recurso que tenga una fecha de vigencia anterior a la versión del recurso en la que el creó el recurso podrá reconocer el recurso.

Folleto

Archivo basado en formato RTF que puede incluir datos de contratos e informes relacionados en un formato conveniente mediante plantillas de diseño predefinidas y configurables

Granularidad

La granularidad determina las unidades de tiempo que el autor de la métricas requiere para obtener el resultado para excluir el período de seguimiento de la métrica.

Grupo de contratantes

Grupo de contratantes (clientes) definido de forma lógica. Al hacer referencia a un grupo en lugar de a una parte contratante individual, se perfecciona la creación de contratos. Un parte contratante puede pertenecer a más de un grupo de contratantes.

Grupo de recursos

Una colección de recursos que se agrupan como una unidad lógica. Un grupo de recursos puede contener a uno o varios recursos, o un grupo de recursos.

Informe compuesto

Varios informes regulares que se muestran como varias series en un solo gráfico

Informe de formato libre

Un informe que se crea mediante una declaración SQL definida por el usuario, basada en la base de datos de CA Business Service Insight u otras bases de datos externas.

Informe de grupo de informes

Un informe que engloba varios informes regulares, colocados uno al lado del otro.

Informe del grupo

Un informe que coloca varios informes de forma simultánea en una página.

Informe simple

Un medio de analizar resultados calculados por CA Business Service Insight basado en un amplio conjunto de criterios.

KPI

Indicador clave de rendimiento. Una medida significativa que se utiliza sola, o en combinación con otros indicadores clave de rendimiento, para controlar en qué medida se están logrando los objetivos cuantificables de un negocio o un servicio.

KQI

Indicador clave de la calidad. Una medida significativa que se utiliza sola, o en combinación con otros indicadores clave de rendimiento, para controlar en qué medida se están logrando los objetivos de calidad de un negocio o un servicio.

Mapa del cuadro de mandos

El diseño del cuadro de mandos. La disposición de los widget en el cuadro de mandos.

Métrica

Una combinación de varios parámetros que definen el nivel de servicio de destino para un servicio concreto en un momento determinado. Cada métrica está adjunta a un solo dominio del servicio. Los campos de métricas y los atributos incluyen categoría del dominio, fórmula al nivel del servicio, servicio, ranura de tiempo, destino al nivel del servicio y período de seguimiento.

Métrica de consumo

Una métrica que permite de consultar consumo y precios por separado en un informe.

Métrica derivada

Una métrica creada a partir de una plantilla de contrato o una plantilla de nivel de servicio.

Métrica en clúster

Un tipo de métrica que se aplica a varios recursos o grupos de recursos.

Métrica informativa

Una métrica que realiza el cálculo informativo que únicamente se va a usar en aspectos relacionados con la elaboración de informes.

Métrica provisional

Una métrica que puede generar eventos con el fin de calcular eventos. Las métricas provisionales no se pueden calcular.

Métrica relacionada

Una métrica a la que se hace referencia como un destino en una relación.

Módulos de lógica de negocios

Biblioteca de funciones de script que se pueden utilizar en la lógica de negocios.

Motor de estado actual

Un proceso independiente que calcula las métricas de estado actual. No hay ningún límite al número de instancias que permite ejecutar en uno o más equipos.

OLA

Acuerdo de nivel operativo. Un OLA es un SLA en el que el proveedor es la organización interna y los clientes son unidades de negocio internas.

Opciones regionales

Detalles específicos para la localidad del parte contratante. Los parámetros incluyen idioma, moneda, diferencia horaria con respecto al horario GMT, horario de verano, formato de fecha y moneda.

Paquete

Una recolección de plantillas de nivel de servicio y plantillas empaquetadas juntas para instalarlas y desempaquetarlas en otra instancia de CA Business Service Insight.

Parámetro

Un valor que se puede establecer fuera de la lógica de negocio para que afecte a la fórmula de lógica de negocio. La lógica de negocio utiliza parámetros para determinar el resultado a nivel del servicio. Un parámetro puede ser de tipo texto, lista, número, fecha o tabla. Ejemplos de parámetros son umbrales y nombres del recurso.

Parámetro a nivel de contrato

Un parámetro de un contrato estándar, plantilla de contrato y plantilla de nivel de servicio que utilizan las métricas en la lógica de negocios.

Parámetros a nivel de métrica

Parámetros de una métrica.

Parte contratante

El cliente o proveedor con el que se firma el contrato. Un parte contratante también puede representar a una entidad interna dentro de una organización de mayor tamaño.

Perfil de alertas

Un conjunto de parámetros que define las condiciones en las que se activa un mensaje de alerta, los usuarios a los que se suministra y cómo se envía.

Período de seguimiento

El intervalo de tiempo durante el que CA Business Service Insight mide el nivel de servicio prestado en relación con el nivel del servicio especificado en el contrato. La medición realizada durante este período determina si existe una desviación con respecto al objetivo definido (como definido por la lógica de negocios) y si se ha incurrido en algún tipo de sanción o prima. El período de seguimiento también determina cómo se generan informes empresariales.

Pista de auditoría

Un registro cronológico de usuario de CA Business Service Insight y de actividades del sistema, que se guarda en el sistema. La auditoría se puede revisar más tarde para identificar acciones de usuarios en el sistema o procesos que se produjeron en el sistema.

Pista de auditoría del contrato

Un registro de todas las actividades de un contrato.

Plantilla de lógica de negocios

Fórmulas de lógica de negocios predeterminadas que se pueden utilizar para definir nuevas fórmulas de lógica de negocios.

Plantilla de nivel de servicio

Una definición del servicio es un conjunto predeterminado de servicios y métricas, que se utiliza para facilitar la creación y modificación de contratos según procesos empresariales habituales.

Plantilla del contrato

Contrato predeterminado que se utiliza para crear un nuevo contrato.

Recurso

Un solo elemento de la infraestructura total del proveedor de servicios. Los ejemplos de recursos incluyen servidores individuales, conmutadores, concentradores, enrutadores, un servicio de ayuda Help Desk o cualquier otro elemento mesurable. Se pueden definir varios recursos como parte de un grupo de recursos, que, a continuación, se trata como otro recurso en sí mismo.

Relación

Una entidad que describe una conexión de un cierto tipo establecida entre dos métricas y que tiene algunas propiedades.

Rol

Un conjunto de responsabilidades, actividades y autorizaciones. El rol de usuario define el modo de vista de CA Business Service Insight. Todos los usuarios de CA Business Service Insight tienen asignado uno o varios roles que determinan qué acciones puede realizar el usuario dentro de CA Business Service Insight. Las acciones que no puede realizar el rol se eliminan de la interfaz de usuario de CA Business Service Insight cuando el usuario en cuestión accede a la aplicación.

Sanción

La compensación debida a la parte contratante si no se cumple el objetivo a nivel del servicio en un período de seguimiento definido. Las sanciones se pueden activar y desactivar en CA Business Service Insight.

Script de traducción

Un script que se puede utilizar para simplificar el mantenimiento de traducciones y prevenir errores.

SLO

Objetivo de nivel de servicio. Se utiliza para medir acuerdos contractuales.

Tabla de traducción

Un mecanismo utilizado para traducir los valores de los datos que proceden de datos sin procesar a los datos definidos en el sistema.

Tasa

Un tipo de Unidad.

Tipo de métrica

Describe el motivo del cálculo de una cierta métrica y define la lista de campos y su comportamiento en términos de campos obligatorios y valores predeterminados que debería tener un tipo de métrica determinado.

Tipo de recurso

Una categoría de recursos incorporada. El recurso se debe adjudicar al menos a un tipo de recurso.

Tipos de evento

Un evento es una medición hecha en un recurso por una herramienta de medición de terceros y a continuación traducida por el adaptador a un formato utilizable por CA Business Service Insight. Los tipos de eventos determinan cómo se definen y se formatean estos eventos para CA Business Service Insight.

Traducciones

La conversión de datos recopilados por adaptadores de sus fuentes de datos a entidades definidas en CA Business Service Insight.

Unidad

Un catálogo de unidades mensurables. Entre los ejemplos se incluyen porcentaje y moneda.

Unidad de medida

Una unidad de medida para todas las métricas definidas para una categoría del dominio.