

IBM WebSphere Business Integration Adapters



Guía del usuario de Adapter for Manugistics

V 11.x

IBM WebSphere Business Integration Adapters



Guía del usuario de Adapter for Manugistics

V 11.x

Nota

Antes de utilizar la información de este manual y el producto al que sirve de complemento, lea la información contenida en la sección "Avisos" en la página 139.

25 de junio de 2004

Esta edición del presente documento es aplicable al producto IBM WebSphere Business Integration Adapter for Manugistics (5724-H28), versión 1.1.x y a todos los releases y modificaciones posteriores hasta que se indique lo contrario en nuevas ediciones.

Este manual es la traducción del original inglés *Adapter for Manugistics User Guide*.

Para enviarnos comentarios sobre este documento, puede enviar un correo electrónico a HOJADOC@es.ibm.com. Nos interesa conocer su opinión.

Cuando envía información a IBM, otorga a IBM un derecho no exclusivo para utilizar o distribuir la información en cualquier forma que IBM considere apropiada, sin contraer ninguna obligación con el remitente de la información.

© Copyright International Business Machines Corporation 2003, 2004. Reservados todos los derechos.

Contenido

Acerca de este documento	v
A quién va dirigido este manual.	v
Documentos afines	v
Convenios tipográficos.	vi
Novedades en este release	vii
Novedades en el release 1.1.x	vii
Novedades en el release 1.0	vii
Capítulo 1. Visión general del conector	1
Componentes del conector.	1
Cómo actúa el conector.	2
Capítulo 2. Instalación y configuración del conector	9
Entorno del adaptador	9
Prerrequisitos.	10
Instalación del adaptador y de los archivos asociados	11
Estructura de archivos instalada	11
Habilitación de Manugistics para el conector	13
Habilitación del soporte para varios controladores	17
Habilitación de la clase gestora personalizada de objetos comerciales	17
Configuración del conector	18
Creación de varias instancias de un conector	32
Inicio del conector	33
Detención del conector	34
Capítulo 3. Descripción de los objetos comerciales en su relación con el conector.	37
Convenios de denominación para objetos comerciales y atributos	37
Estructura de los objetos comerciales	37
Procesos de verbos de objeto comercial	42
Propiedades de atributos de objetos comerciales	60
Información específica de la aplicación para objetos comerciales.	62
Capítulo 4. Creación de definiciones de objetos comerciales utilizando IBM ODA for Manugistics	75
Instalación y utilización	75
Utilización de ODA for Manugistics en el Diseñador de objetos comerciales.	79
Contenido de la definición generada	89
Archivo de definición de objeto comercial de ejemplo	92
Inserción de atributos que contienen objetos comerciales hijo	93
Adición de información a la definición de objeto comercial	93
Capítulo 5. Resolución y manejo de errores.	95
Problemas de inicio.	95
Proceso de sucesos	95
Correlación (solamente para el intermediario de integración InterChange Server)	95
Manejo y registro de errores.	97
Pérdida de la conexión con la aplicación.	98
Error de búsqueda fuera de secuencia	98
Error de recurso ocupado.	99
Funcionamiento incorrecto de ODA for Manugistics debido a un controlador JDBC no compatible	99
Manejo de errores cuando se utiliza IGP.	99
Apéndice A. Propiedades de configuración estándar para conectores	101

Propiedades nuevas y propiedades suprimidas	101
Configuración de propiedades estándar de conector	101
Resumen de las propiedades estándar	103
Propiedades de configuración estándar	107
Apéndice B. Configurador de conectores	121
Visión general del Configurador de conectores	121
Inicio del Configurador de conectores	122
Ejecución del Configurador desde el Gestor del sistema	123
Creación de una plantilla de propiedades específica del conector	123
Creación de un archivo de configuración nuevo	126
Utilización de un archivo existente	127
Finalización de un archivo de configuración	128
Definición de propiedades en el archivo de configuración	129
Guardar el archivo de configuración.	135
Modificación de un archivo de configuración	135
Finalización de la configuración	136
Uso del Configurador de conectores en un entorno globalizado	136
Apéndice C. Utilización de valores nulos y valores en blanco	137
Situaciones aprobadas y de error	137
Funcionalidad	138
Avisos	139
Información de la interfaz de programación	140
Marcas registradas y marcas de servicio	141

Acerca de este documento

La oferta de productos de IBM^(R) WebSphere^(R) Business Integration Adapter proporciona conectividad de integración para tecnologías avanzadas de e-business, aplicaciones empresariales, sistemas heredados y grandes sistemas. La familia de productos incluye herramientas y modelos para personalizar, crear y gestionar componentes a fin de integrar procesos empresariales.

Este documento describe la instalación, configuración, y desarrollo de objetos de negocio para el Adaptador de Manugistics.

A quién va dirigido este manual

Este documento está pensado para asesores, desarrolladores y administradores de sistemas que hacen uso del conector en sitios Web del cliente.

Documentos afines

El conjunto completo de documentación existente para este producto describe las funciones y componentes que son comunes a todas las instalaciones de WebSphere Business Integration Adapters, e incluye información de consulta sobre componentes determinados.

Puede instalar la documentación pertinente desde los sitios Web siguientes:

- Para obtener información general sobre adaptadores; para utilizar adaptadores con intermediarios de mensajes de WebSphere (WebSphere MQ Integrator, WebSphere MQ Integrator Broker, WebSphere Business Integration Message Broker); y para utilizar adaptadores con WebSphere Application Server:
<http://www.ibm.com/websphere/integration/wbiadapters/infocenter>
- Para utilizar adaptadores con InterChange Server:
<http://www.ibm.com/websphere/integration/wicserver/infocenter>
<http://www.ibm.com/websphere/integration/wbicollaborations/infocenter>
- Para obtener más información sobre intermediarios de mensajes (WebSphere MQ Integrator Broker, WebSphere MQ Integrator y WebSphere Business Integration Message Broker):
<http://www.ibm.com/software/integration/mqfamily/library/manualsa/>
- Para obtener más información sobre WebSphere Application Server:
<http://www.ibm.com/software/webservers/appserv/library.html>

Estos sitios Web contienen instrucciones sencillas para descargar, instalar y visualizar la documentación.

Nota: Puede encontrar información importante sobre el producto en Notas de soporte técnico y Comunicados emitidos después de la publicación del presente documento. Estos documentos adicionales se hallan en el sitio Web de soporte técnico de WebSphere Business Integration:
<http://www.ibm.com/software/integration/websphere/support/>.

Convenios tipográficos

Este documento utiliza los convenios siguientes:

font courier	Indica un valor literal, tal como un nombre de mandato, un nombre de archivo, información que debe escribir el usuario o información que el sistema muestra en la pantalla.
negrita	Indica un término nuevo, cuando aparece por primera vez.
<i>cursiva,</i>	Indica un nombre variable o una referencia cruzada.
<i>línea de contorno azul</i>	Sólo es visible cuando se visualiza el manual en la pantalla e indica un hiperenlace de referencia cruzada. Pulse con el ratón dentro de la línea de contorno para ir hacia el objeto al que apunta la referencia.
<i>DirProducto</i>	Representa el directorio donde está instalado el producto. Para el entorno de IBM WebSphere InterChange Server, el directorio por omisión del producto es "IBM\WebSphereICS". Para el entorno de IBM WebSphere Business Integration Adapters, el directorio por omisión del producto es "WebSphereAdapters".
{ }	En una línea de sintaxis, se utilizan llaves para delimitar un conjunto de opciones de las cuales el usuario debe elegir una sola.
	En una línea de sintaxis, se utiliza una barra vertical para separar un conjunto de opciones de las cuales el usuario debe elegir una sola.
[]	En una línea de sintaxis, los corchetes rectos delimitan un parámetro opcional.
...	En una línea de sintaxis, los puntos suspensivos denotan la repetición del parámetro anterior. Por ejemplo, opción[,...] significa que puede especificar varias opciones, separadas por comas.
< >	Los corchetes angulares delimitan elementos individuales de un nombre para distinguirlos entre sí, tal como en <nombre_servidor><nombre_conector>tmp.log.
/, \	En el presente documento, las barras inclinadas invertidas (\) se utilizan como convenio para representar las vías de acceso de directorios. Para las instalaciones en UNIX ^(R) , utilice barras inclinadas (/) en lugar de barras inclinadas invertidas. Todas las vías de acceso del producto están referidas al directorio donde reside el conector de JDBC en el sistema del usuario.
UNIX/Windows: ^(TM)	Los párrafos que comienzan con cualquiera de estas dos especificaciones indican notas que describen diferencias específicas del sistema operativo.
%texto% y \$texto	En Windows, el texto delimitado por símbolos de porcentaje (%) indica el valor de la variable del sistema o variable de usuario texto. La notación equivalente en un entorno UNIX es \$texto, que denota el valor de la variable de entorno texto de UNIX.

Novidades en este release

Novidades en el release 1.1.x

Actualizado en junio de 2004. El release de este documento para la versión 1.1.x del adaptador contiene la siguiente información nueva o corregida:

- El adaptador admite la utilización de Manugistics NetWORKS Collaborate, versión 7.2. Debido a esta nueva funcionalidad, el nombre del adaptador se ha cambiado a Adapter for Manugistics.
- Se han añadido nuevas propiedades de configuración específicas del conector: QueryTimeout, ReturnDummyBOForSP, SelectiveCommitForPoll y UseDefaultsWhenPolling (que sustituye a UseDefaultsForRetrieve). Se ha documentado una propiedad existente: DriverSupportForLong.
- El adaptador admite la utilización de un nuevo parámetro para los atributos de tipo DATE en la información específica de la aplicación.
- El conector reconoce estos nombres de atributos de procedimiento almacenado para el verbo DeltaUpdate: BeforeDeltaUpdateSP, AfterDeltaUpdateSP, DeltaUpdateSP.
- El Árbol de esquema contiene el nodo adicional sinónimos/apodos para identificar objetos de base de datos que se deben asociar con la definición de objeto comercial creada.
- El Árbol de esquema contiene el nombre de esquema ALL SCHEMAS para ayudar a recuperar objetos de bases de datos que no tienen esquemas asociados con los objetos.
- En la versión 1.1.x, Adapter for Manugistics no se puede utilizar en Solaris 7.0, por lo que las referencias a esa plataforma se han eliminado de la presente guía.

Novidades en el release 1.0

Actualizado en febrero de 2004. El release de este documento contiene la siguiente información nueva o corregida: se ha añadido el parámetro de nombre-valor [PH=true|false] a la "Información específica de la aplicación para atributos simples" en el Capítulo 3.

Diciembre de 2003

La versión 1.0 es el primer release del manual *Adapter for Manugistics Demand and Fulfillment Management, Guía del usuario*.

Capítulo 1. Visión general del conector

Los conectores constan de dos partes: la **infraestructura del conector** y el **componente específico de la aplicación**. La infraestructura del conector, cuyo código es común a todos los conectores, actúa como intermediario entre el intermediario de integración y el componente específico de la aplicación. El componente específico de la aplicación contiene código que está adaptado a una determinada aplicación o tecnología (en este caso, JDBC). La infraestructura del conector proporciona los servicios siguientes entre el intermediario de integración y el componente específico de la aplicación:

- Recibe y envía objetos comerciales
- Gestiona el intercambio de mensajes de inicio y mensajes administrativos

Este capítulo describe el componente conector de IBM WebSphere Business Integration Adapter for Manugistics. Observe que el presente documento contiene información tanto sobre la infraestructura del conector como sobre el componente específico de la aplicación. Se utiliza el término conector para hacer referencia a ambos elementos. Para obtener más información sobre la relación del intermediario de integración con el conector, consulte el manual *IBM WebSphere InterChange Server System Administration Guide* o *IBM WebSphere Business Integration Adapters Implementation Guide for WebSphere MQ Integrator Broker*.

Este capítulo contiene las secciones siguientes:

- “Componentes del conector”
- “Cómo actúa el conector” en la página 2

Componentes del conector

El conector de Manugistics permite que el intermediario de integración intercambie objetos comerciales con una aplicación basada en cualquier base de datos compatible con un controlador que se ajusta a la especificación JDBC 2.0 o posterior. Esta sección proporciona una descripción general de la arquitectura del conector y del uso que hace de diversos controladores JDBC.

Para especificar el controlador que debe utilizar el conector para conectar con la base de datos, consulte “Habilitación del soporte para varios controladores” en la página 17.

El conector conecta con la base de datos de la aplicación utilizando el mecanismo de JDBC Connect. Un parámetro de configuración específico del conector (DatabaseURL) le permite especificar el nombre del servidor de bases de datos al que se debe conectar el conector. Para obtener información sobre los parámetros de configuración, consulte “Configuración del conector” en la página 18.

Cuando se inicia el conector, establece un conjunto de conexiones con la base de datos. El conector utiliza conexiones de este conjunto para el proceso de todas las transacciones con la base de datos. Cuando finaliza la ejecución del conector, se cierran todas las conexiones del conjunto.

Arquitectura del conector

La Figura 1 muestra los componentes del conector y sus relaciones con el sistema de integración empresarial.

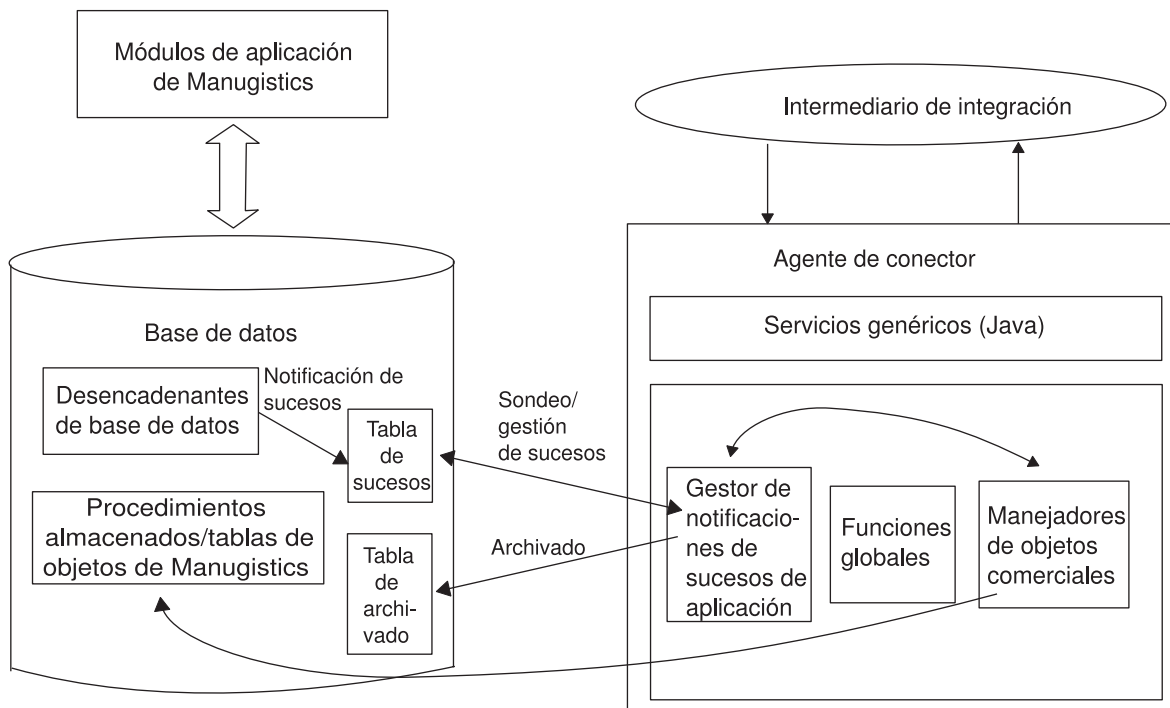


Figura 1. Arquitectura de las peticiones de objetos comerciales

Cómo actúa el conector

Esta sección describe cómo los metadatos mejoran la flexibilidad del conector, y proporciona una descripción general del proceso de los objetos comerciales y de la notificación de sucesos.

El conector y los metadatos

El conector está gobernado por metadatos. En el entorno de IBM WebSphere Business Integration Adapter, los **metadatos** son datos específicos de la aplicación que se almacenan en objetos comerciales y que ayudan al conector en su interacción con la aplicación. Un conector gobernado por metadatos maneja cada objeto comercial compatible de acuerdo con metadatos que están codificados en la definición del objeto comercial, en lugar de hacerlo basándose en instrucciones codificadas en el conector.

Los metadatos de objeto comercial incluyen la estructura de un objeto comercial, los valores de sus propiedades de atributos y el contenido de su información específica de la aplicación. Debido a que el conector está gobernado por metadatos, puede manejar objetos comerciales nuevos o modificados sin necesitar modificaciones en el código del conector.

El conector ejecuta sentencias de SQL o procedimientos almacenados para recuperar o cambiar datos en la base de datos/aplicación. Para crear sentencias de SQL dinámico o procedimientos almacenados, el conector utiliza metadatos específicos de la aplicación. Estas sentencias de SQL y procedimientos almacenados realizan las acciones necesarias de recuperación o cambio en la base de

datos/aplicación para el objeto comercial y para el verbo que el conector está procesando. Para obtener información sobre el uso de información específica de la aplicación, consulte el Capítulo 3, “Descripción de los objetos comerciales en su relación con el conector”, en la página 37.

Proceso de objetos comerciales

Esta sección proporciona una visión general de cómo el conector procesa peticiones de objetos comerciales y sucesos de aplicación. Para obtener información más detallada, consulte “Procesos de verbos de objeto comercial” en la página 42.

Proceso de peticiones de objetos comerciales

Cuando el conector recibe una petición para efectuar una operación de aplicación, el conector procesa recursivamente objetos comerciales jerárquicos; es decir, efectúa los mismos pasos para cada objeto comercial hijo hasta que ha procesado todos los objetos comerciales individuales. El orden en el que el conector procesa los objetos comerciales hijo y el objeto comercial de nivel superior depende de si los objetos comerciales hijo están contenidos con o sin propietario y de si están contenidos con cardinalidad simple o cardinalidad múltiple.

Nota: El término objeto comercial **jerárquico** hace referencia a un objeto comercial completo, incluidos todos los objetos comerciales hijo que el objeto contiene a cualquier nivel. El término objeto comercial **individual** hace referencia a un sólo objeto comercial, con independencia de los objetos comerciales hijo que pueda contener o en los que esté contenido. El término objeto comercial de **nivel superior** hace referencia al objeto comercial individual situado en lo más alto de la jerarquía y que no tiene ningún objeto comercial padre.

Recuperación de objetos comerciales: Cuando un intermediario de integración solicita al conector que recupere un objeto comercial jerárquico de la base de datos, el conector intenta devolver un objeto comercial que coincida exactamente con la representación actual de ese objeto comercial en la base de datos. Es decir, todos los atributos simples de cada objeto comercial individual devuelto al intermediario de integración coinciden con el valor del campo correspondiente de la base de datos. Además, el número de objetos comerciales individuales en cada matriz contenida por el objeto comercial devuelto coincide con el número de hijos en la base de datos para esa matriz.

Para realizar una recuperación, el conector utiliza los valores de clave primaria del objeto comercial de nivel superior para descender recursivamente por los datos correspondientes de la base de datos.

Recuperación de objetos comerciales por contenido: Cuando un intermediario de integración solicita al conector que recupere un objeto comercial jerárquico basándose en valores de atributos no de clave del objeto comercial de nivel superior, el conector utiliza el valor de todos los atributos no nulos como criterio para recuperar los datos.

Creación de objetos comerciales: Cuando un intermediario de integración solicita al conector que cree un objeto comercial jerárquico en la base de datos, el conector efectúa estos pasos:

1. Crea recursivamente cada objeto comercial hijo de cardinalidad simple contenido con propietario en la base de datos.
2. Procesa cada objeto comercial hijo de cardinalidad simple contenido sin propietario.
3. Crea el objeto comercial de nivel superior en la base de datos.

4. Crea los objetos comerciales hijo de cardinalidad simple que guardan la relación padre/hijo en el hijo.
5. Crea cada objeto comercial hijo de cardinalidad múltiple.

Modificación de objetos comerciales: Cuando un intermediario de integración solicita al conector que actualice un objeto comercial jerárquico en la base de datos, el conector efectúa estos pasos:

1. Utiliza los valores de clave primaria del objeto comercial de origen para recuperar la entidad correspondiente de la base de datos.
2. Actualiza recursivamente todos los hijos de cardinalidad simple del objeto comercial de nivel superior.
3. Para los objetos comerciales hijo de cardinalidad simple que guardan la relación padre/hijo en el padre, el conector establece cada valor de clave foránea del padre en el valor de la clave primaria del correspondiente objeto comercial hijo de cardinalidad simple.
4. Actualiza todos los atributos simples del objeto comercial recuperado, excepto aquéllos cuyo atributo correspondiente en el objeto comercial de origen contiene el valor CxIgnore.
5. Establece todos los valores de clave foránea de cada hijo que guarda la relación padre/hijo en el hijo (tanto de cardinalidad simple como múltiple) en el valor de clave primaria del correspondiente objeto comercial padre.
6. Procesa todas las matrices del objeto comercial recuperado.

Supresión de objetos comerciales: Cuando un intermediario de integración solicita al conector que suprima un objeto comercial jerárquico de la base de datos, el conector efectúa estos pasos:

1. Suprime los hijos de cardinalidad simple.
2. Suprime los hijos de cardinalidad múltiple.
3. Suprime el objeto comercial de nivel superior.

Proceso de sucesos de aplicación

El conector maneja los sucesos Crear, Actualizar y Suprimir generados por la aplicación de la manera descrita a continuación.

Notificación para Crear: Cuando el conector encuentra un suceso Crear en la tabla de sucesos, crea un objeto comercial del tipo especificado por el suceso, establece los valores de clave para el objeto comercial (utilizando las claves especificadas en la tabla de sucesos) y recupera el objeto comercial de la base de datos. Después de recuperar el objeto comercial, el conector lo envía junto con el verbo Crear al intermediario de integración.

Notificación para Actualizar: Cuando el conector encuentra un suceso Actualizar en la tabla de sucesos, crea un objeto comercial del tipo especificado por el suceso, establece los valores de clave para el objeto comercial (utilizando las claves especificadas en la tabla de sucesos) y recupera el objeto comercial de la base de datos. Después de recuperar el objeto comercial, el conector lo envía junto con el verbo Actualizar al intermediario de integración.

Notificación para Suprimir: Cuando el conector encuentra un suceso Suprimir en la tabla de sucesos, crea un objeto comercial del tipo especificado por el suceso, establece los valores de clave para el objeto comercial (utilizando las claves especificadas en la tabla de sucesos) y envía el objeto comercial junto con el verbo Suprimir al intermediario de integración. Todos los valores excepto los valores de

clave se establecen en CxIgnore. Si alguno de los campos que no son de clave son importantes en su sitio, modifique el valor de los campos según sea necesario.

El conector maneja las operaciones lógicas y físicas de Suprimir que son desencadenadas por la aplicación del conector. En el caso de las supresiones físicas, el mecanismo SmartFiltering elimina todos los sucesos no procesados del objeto comercial (tales como Crear o Actualizar) antes de insertar el suceso Suprimir en la tabla de sucesos. En el caso de las supresiones lógicas, el conector inserta un suceso Suprimir en la tabla de sucesos sin eliminar otros sucesos para el objeto comercial.

Recuperación de objetos comerciales para el proceso de sucesos: La Recuperación de un objeto comercial para el proceso de sucesos puede realizarse de dos maneras. La primera es la Recuperación basada en atributos de clave de un objeto comercial. La segunda es la Recuperación basada en atributos de clave y atributos no de clave. En este caso, el objeto comercial necesita ser compatible con el verbo RetrieveByContent y debe utilizar el par nombre_valor para las claves de objeto.

Nota: Si la clave de objeto no utiliza el par nombre_valor, las claves contenidas en el campo de claves de objeto deben seguir el mismo orden que las claves contenidas en el objeto comercial.

Notificación de sucesos

El mecanismo de detección de sucesos del conector utiliza una tabla de sucesos, una tabla de archivado, procedimientos almacenados y desencadenantes de base de datos. Debido a la existencia de posibles puntos de error asociados al proceso de sucesos, el proceso de gestión de sucesos no suprime un suceso de la tabla de sucesos hasta que se ha insertado en la tabla de archivado.

Los desencadenantes de base de datos añaden datos a una tabla de sucesos cada vez que se produce un suceso de interés en la base de datos. El conector sondea esta tabla según un intervalo regular configurable, recupera los sucesos y los procesa primero por orden de prioridad y luego secuencialmente. Cuando el conector ha procesado un suceso, se actualiza el estado del suceso.

Nota: El usuario debe añadir los desencadenantes a la base de datos como parte del procedimiento de instalación.

El valor de la propiedad ArchiveProcessed del conector determina si el conector archiva un suceso en la tabla de archivado después de actualizar el estado del suceso. Para obtener más información sobre la propiedad ArchiveProcessed, consulte "Configuración del conector" en la página 18.

La Tabla 1 muestra cómo se realiza el archivado dependiendo del valor de la propiedad ArchiveProcessed.

Tabla 1. Maneras de realizarse el archivado

Valor de ArchiveProcessed	Razón para suprimir de la tabla de sucesos	Acción del conector
true o sin valor	Procesado con éxito	Archivado con estado de Sent to InterChange
	Procesado sin éxito	Archivado con estado de Error
	No existe ninguna suscripción para el objeto comercial	Archivado con estado de Unsubscribed

Tabla 1. Maneras de realizarse el archivado (continuación)

Valor de ArchiveProcessed	Razón para suprimir de la tabla de sucesos	Acción del conector
false	Procesado con éxito	No se archiva y se suprime de la tabla de sucesos
	Procesado sin éxito	Permanece en la tabla de sucesos con el estado de Error
	No existe ninguna suscripción para el objeto comercial	Permanece en la tabla de sucesos con el estado de Unsubscribed

SmartFiltering es un mecanismo dentro de los desencadenantes de base de datos que minimiza el volumen de proceso que deben realizar el intermediario de integración y el conector. Por ejemplo, si una aplicación ha actualizado 15 veces el objeto comercial Contrato desde la última vez que el conector comprobó la existencia de sucesos, SmartFiltering guarda esos cambios como un único suceso de Actualizar.

Manejo de conexiones perdidas de base de datos

Existen muchas razones para perder una conexión de base de datos. Si se produce esta pérdida, el conector concluye su ejecución. La especificación JDBC no proporciona un mecanismo para detectar conexiones perdidas. Debido a que el conector puede trabajar con bases de datos diferentes, no existe una única definición de código de error para una conexión perdida con una base de datos.

Se utiliza la propiedad PingQuery para gestionar esta detección. Si se produce un error durante una petición de llamada de servicio, el conector ejecuta PingQuery para verificar que el error no se debió a una conexión perdida con una base de datos. Si la operación PingQuery falla y la propiedad AutoCommit tiene el valor "false", el conector intentará crear una nueva conexión con la base de datos. Si el conector logra establecer una nueva conexión con la base de datos, continuará el proceso; en caso contrario, el conector devuelve el código de retorno APPRESPONSETIMEOUT, que hace que concluya la ejecución del conector.

Se ejecuta PingQuery si se produce un error al acceder a una base de datos para cualquier tipo de transacción. Por ejemplo:

- Al acceder a las tablas de sucesos y de archivado
- Al recuperar el objeto comercial que está asociado al suceso
- Al crear o actualizar un registro relativo a un objeto comercial

Proceso de datos dependientes del entorno local

El conector se ha internacionalizado para que pueda trabajar con juegos de caracteres de doble byte y emitir mensajes en el idioma especificado. Cuando el conector transfiere datos desde una ubicación a otra que hace uso de un juego de códigos de caracteres diferente que la ubicación de origen, el conector realiza una conversión de caracteres para preservar el significado de los datos.

El entorno de ejecución Java^(TM) de la Máquina Virtual Java (Java Virtual Machine, JVM) representa los datos utilizando el juego de códigos de caracteres Unicode. Unicode contiene codificaciones de caracteres para la mayoría de juegos de códigos de caracteres conocidos (tanto de un sólo byte como de varios bytes). La mayoría de los componentes del sistema WebSphere Business Integration están escritos en Java. Por tanto, para la mayoría de los componentes del sistema WebSphere

Business Integration, no hay necesidad de realizar una conversión de caracteres cuando se transfieren datos entre los componentes.

Para registrar mensajes de error e informativos utilizando el idioma adecuado y para el país o territorio apropiado, configure la propiedad de configuración estándar Locale para su entorno. Para obtener más información sobre estas propiedades, consulte el Apéndice A, “Propiedades de configuración estándar para conectores”, en la página 101.

Nota: Para habilitar juegos de códigos de caracteres diferentes, consulte la documentación de Manugistics.

Capítulo 2. Instalación y configuración del conector

Este capítulo describe la instalación y configuración de IBM WebSphere Business Integration Adapter for Manugistics y la configuración de aplicaciones para trabajar con el conector. Este capítulo contiene las secciones siguientes:

- “Entorno del adaptador”
- “Prerrequisitos” en la página 10
- “Estructura de archivos instalada” en la página 11
- “Habilitación de Manugistics para el conector” en la página 13
- “Habilitación del soporte para varios controladores” en la página 17
- “Habilitación de la clase gestora personalizada de objetos comerciales” en la página 17
- “Configuración del conector” en la página 18
- “Creación de varias instancias de un conector” en la página 32
- “Inicio del conector” en la página 33

Entorno del adaptador

Antes de instalar, configurar y utilizar el adaptador, debe conocer los requisitos del adaptador referentes al entorno. Estos requisitos están listados en la sección siguiente.

- “Compatibilidad con el intermediario”
- “Plataformas del adaptador” en la página 10
- “Dependencias del adaptador” en la página 10
- “Globalización” en la página 10

Compatibilidad con el intermediario

La infraestructura de adaptador utilizada por un adaptador debe ser compatible con la versión del intermediario (o intermediarios) de integración con los cuales se comunica el adaptador. La infraestructura del adaptador se instala al instalar el intermediario de integración o al instalar el adaptador, dependiendo del intermediario de integración que se utilice. La versión 1.1.x de Adapter for Manugistics se puede utilizar con la infraestructura de adaptador y los intermediarios de integración siguientes:

- **Infraestructura de adaptador:**
WebSphere Business Integration Adapter Framework, versiones 2.3.1 y 2.4.
- **Intermediarios de integración:**
 - WebSphere InterChange Server, versiones 4.1.1, 4.2, 4.2.1 y 4.2.2
 - WebSphere MQ Integrator, versión 2.1.0
 - WebSphere MQ Integrator Broker, versión 2.1.0
 - WebSphere Business Integration Message Broker, versión 5.0
 - WebSphere Application Server Enterprise, versión 5.0.2, con WebSphere Studio Application Developer Integration Edition, versión 5.0.1

Consulte las *Notas de release* para conocer posibles excepciones.

Nota: Para obtener instrucciones sobre la instalación del intermediario de integración y sus prerrequisitos, consulte las guías siguientes.

Para WebSphere InterChange Server (ICS), consulte *IBM WebSphere InterChange Server System Installation Guide for UNIX* o la correspondiente guía para Windows (*for Windows*).

Para intermediarios de mensajes de WebSphere, consulte *Implementing Adapters with WebSphere Message Brokers*.

Para WebSphere Application Server, consulte *Implementing Adapters with WebSphere Application Server*.

Plataformas del adaptador

El adaptador se puede utilizar en las plataformas siguientes:

Sistemas operativos:

- AIX^(R) 5.1, AIX 5.2
- Solaris 8.0
- HP UX 11i
- Windows 2000

Bases de datos:

- Oracle 9i

Software de terceros:

- Manugistics Demand and Fulfillment Management, versión 7.1.
- Manugistics NetWORKS Collaborate, versión 7.2

Dependencias del adaptador

Si está utilizando una base de datos Oracle, debe instalar las bibliotecas de OracleOCI.

Globalización

El adaptador está habilitado para juegos de caracteres de doble byte (DBCS) y está traducido.

Prerrequisitos

Antes de utilizar el conector, debe seguir estos pasos:

- Instale el Adapter Development Kit sólo si piensa ampliar o modificar la funcionalidad del adaptador.

Si el conector se ejecuta en una máquina diferente que el intermediario de integración, instale el Adapter Development Kit que sea compatible con la versión del intermediario de integración que está utilizando.

- Instale el controlador JDBC que se utilizará.
- Verifique que se haya instalado todo el software específico del proveedor, incluidos los requisitos referentes al controlador JDBC.

Por ejemplo, si está utilizando el controlador JDBC de Tipo 2 para una base de datos Oracle, debe instalar las bibliotecas de OracleOCI.

- Compruebe la existencia de una cuenta de usuario en la aplicación.

El conector procesa datos de las bases de datos Manugistics que sean compatibles con controladores que cumplen la especificación JDBC. Para que el conector procese los datos de la base de datos, con la que se comunica directamente, debe tener acceso a una cuenta de usuario y a una contraseña que sea válida para la aplicación. La cuenta de usuario debe tener los privilegios

necesarios para recuperar, insertar, actualizar y suprimir datos de la base de datos de la aplicación. Si todavía no dispone de una cuenta así, debe crear una.

- Compruebe el juego de códigos de caracteres de la base de datos conectada.

El entorno de ejecución Java de la Máquina Virtual Java (Java Virtual Machine, JVM) representa los datos utilizando el juego de códigos de caracteres Unicode. Unicode contiene codificaciones de caracteres para la mayoría de juegos de caracteres conocidos (tanto de un sólo byte como de varios bytes). Como el conector está escrito en Java, reconoce el juego de códigos Unicode.

Instalación del adaptador y de los archivos asociados

Para obtener información sobre la instalación de los productos del adaptador WebSphere Business Integration, consulte el documento *Installation Guide for WebSphere Business Integration Adapters*, que se encuentra en el centro de información de WebSphere Business Integration Adapters, en el sitio Web siguiente:

<http://www.ibm.com/websphere/integration/wbiadapters/infocenter>

Estructura de archivos instalada

Las subsecciones siguientes describen la estructura de archivos instalada del adaptador en un sistema UNIX o Windows.

Nota: A menos que se indique otra cosa, las secciones restantes de este capítulo son relevantes para la instalación del adaptador tanto en UNIX como en Windows.

Instalación en un sistema UNIX

Para obtener instrucciones sobre la instalación de Adapter for Manugistics en un sistema UNIX, consulte el manual *Installation Guide for WebSphere Business Integration Adapters*.

La Tabla 2 en la página 12 describe la estructura de archivos UNIX utilizada por el conector.

Tabla 2. Estructura de archivos UNIX instalada para el conector

Subdirectorio de <i>DirProducto</i>	Descripción
connectors/Manugistics	Contiene los archivos BIA_Manugistics.jar y start_Manugistics.sh del conector. El archivo start_Manugistics.sh es un script de inicio del sistema utilizado por el conector. Este script se invoca desde el script del gestor de conectores genéricos. Cuando el usuario selecciona Instalar, en el Configurador de conectores (WebSphere MQ Integrator Broker como intermediario de integración) o en la pantalla Configuración de conectores del Gestor del sistema (InterChange Server como intermediario de integración), el programa de instalación de IBM WebSphere Business Integration Adapter crea un reiniciador personalizado para este script del gestor de conectores. Cuando use el conector con InterChange Server, utilice este reiniciador personalizado para iniciar y detener el conector. Cuando use el conector con WebSphere MQ Integrator Broker, utilice este reiniciador personalizado solamente para iniciar el conector. Utilice el mandato mqsi remotestopadapter para detener el conector.
connectors/Manugistics/dependencies	Contiene los scripts de SQL mediante los cuales se crean las tablas de sucesos, las tablas de archivado y las tablas de identificadores exclusivos.
connectors/messages	Contiene el archivo BIA_ManugisticsAdapter.txt.
repository/Manugistics	Contiene el archivo BIA_CN_Manugistics.txt.
/lib	Contiene el archivo WBIA.jar
/bin	Contiene el archivo CWConnEnv.sh.

Para obtener más información sobre la instalación del componente conector, consulte una de las guías siguientes, en función del intermediario de integración que esté utilizando:

- *System Installation Guide for UNIX* (cuando se utiliza InterChange Server como intermediario de integración)
- *IBM WebSphere Business Integration Adapters Implementation Guide for WebSphere MQ Integrator Broker* (cuando se utiliza WebSphere MQ Integrator Broker como intermediario de integración)

Estructura de archivos instalada en un sistema Windows

Para obtener instrucciones sobre la instalación de Adapter for Manugistics en un sistema Windows, consulte el manual *Installation Guide for WebSphere Business Integration Adapters*. La Tabla 3 describe la estructura de archivos Windows utilizada por el conector.

Tabla 3. Estructura de archivos Windows instalada para el conector

Subdirectorio de <i>DirProducto</i>	Descripción
connectors\Manugistics	Contiene los archivos BIA_Manugistics.jar y start_Manugistics.bat del conector.
connectors\Manugistics\dependencies	Contiene los scripts de SQL mediante los cuales se crean las tablas de sucesos, las tablas de archivado y las tablas de identificadores exclusivos.
connectors\messages	Contiene el archivo BIA_ManugisticsAdapter.txt.
repository\Manugistics	Contiene el archivo BIA_CN_Manugistics.txt.

Tabla 3. Estructura de archivos Windows instalada para el conector (continuación)

Subdirectorio de <i>DirProducto</i>	Descripción
\lib	Contiene el archivo WBIA.jar
\bin	Contiene el archivo CWConnEnv.bat.

El programa de instalación añade un icono para el archivo del conector en el menú de IBM WebSphere Business Integration Adapters. Para disponer de una forma rápida de iniciar el conector, cree un acceso directo para este archivo en el escritorio.

Para obtener más información sobre la instalación del componente conector, consulte una de las guías siguientes, en función del intermediario de integración que esté utilizando:

- *System Installation Guide for Windows* (cuando se utiliza InterChange Server como intermediario de integración)
- *IBM WebSphere Business Integration Adapters Implementation Guide for WebSphere MQ Integrator Broker* (cuando se utiliza WebSphere MQ Integrator Broker como intermediario de integración)

Habilitación de Manugistics para el conector

Esta sección trata los temas siguientes:

- “Configuración de las tablas de generación de interfaz de Manugistics”
- “Configuración del proceso de sucesos y de archivado” en la página 14
- “Tablas de sucesos y de archivado” en la página 15
- “Scripts de SQL para instalar las tablas de sucesos y de archivado” en la página 15
- “Esquema de las tablas de sucesos y de archivado” en la página 16

Configuración de las tablas de generación de interfaz de Manugistics

Adapter for Manugistics se integrará, utilizando IGP (Interface Generation Program) y sus tablas de interfaz, con objetos comerciales de la conexión JDBC y de WebSphere Business Integration Adapters que crean/actualizan/suprimen y recuperan datos de la base de datos. IGP es un proceso en el que se utilizan tablas intermedias para propagar datos hacia tablas activas de base de datos, mediante la ejecución de diversos procedimientos específicos de la aplicación.

Para que las peticiones de llamada de servicio creen y actualicen valores de datos en la base de datos, el adaptador utilizará las tablas de interfaz de IGP como ubicación intermedia para los valores. Este proceso supone colocar datos en las tablas de interfaz, los cuales se procesan mediante procedimientos almacenados y se trasladan a tablas activas. Estos procedimientos almacenados permiten realizar comprobaciones de integridad de los datos.

Necesitará ejecutar y configurar IGP (Interface Generation Program) de Manugistics para crear Inserciones, Actualizaciones y Actualizaciones/Inserciones para los datos deseados de la tabla de integración. Consulte el manual *Manugistics WebWorks Guide* para conocer detalles.

Debe configurar el mecanismo de notificación de sucesos en la base de datos para que el conector pueda procesar la entrega de sucesos. Para ello, debe realizar estas tareas:

- Cree las tablas de sucesos y de archivado en la base de datos.
- Instale desencadenantes de base de datos en las tablas de la aplicación para poder utilizar los objetos comerciales necesarios. Se supone que el usuario desarrolla sus propios desencadenantes de base de datos.
- Opcionalmente, instale una tabla contador. Efectúe este paso sólo si solicita al conector que genere un ID exclusivo al crear un objeto comercial. Para obtener más información sobre la generación de los ID exclusivos, consulte la explicación sobre el parámetro `UID=CW.uidcolumnname[=UseIfMissing]`.

Las secciones que siguen proporcionan información sobre la creación y configuración de las tablas de sucesos y de archivado.

Configuración del proceso de sucesos y de archivado

Para configurar el proceso de sucesos y de archivado, debe utilizar propiedades de configuración para especificar la información siguiente:

- El nombre de la tabla de sucesos (`EventTableName`). No necesita especificar un valor para esta propiedad si utiliza el conector solamente para procesar peticiones de objetos comerciales.
- La frecuencia de intervalo ("`PollFrequency`" en la página 115).
- El número de sucesos para cada intervalo de sondeo (`PollQuantity`).
- El nombre de la tabla de archivado (`ArchiveTableName`).
- Indicación de si el conector archiva los sucesos no suscritos y no procesados (`ArchiveProcessed`). Para obtener información sobre la suscripción que sea específica del intermediario de integración utilizado, consulte la guía de implementación del intermediario.
- El ID exclusivo del conector, el cual es importante cuando varios conectores sondean la misma tabla (`ConnectorID`).

Puede también especificar un valor para la propiedad `EventOrderBy` para indicar el orden de los sucesos que se deben procesar. Para obtener información sobre estas y otras propiedades de configuración, consulte el Apéndice A, "Propiedades de configuración estándar para conectores", en la página 101 y la Tabla 6 en la página 19.

Nota: La creación de las tablas de sucesos y de archivado es opcional. Sin embargo, si especifica un valor para `EventTableName`, pero no utiliza el conector para sondear sucesos y no crea una tabla de sucesos, se produce un tiempo de espera excedido para el conector. Para evitar este tiempo de espera excedido, establezca el valor de `EventTableName` en `null` (como serie de caracteres).

Por omisión, el nombre de la tabla de cola de sucesos es `xworlds_events`, y el nombre de la tabla de cola de archivado es `xworlds_archive_events`.

Para utilizar el conector solamente para el proceso de peticiones, utilice la opción `-fno` al iniciar el conector y establezca el valor de `EventTableName` en `null` (como serie de caracteres).

Si el controlador utilizado no es compatible con la clase `Java DatabaseMetaData`, y no desea que el conector compruebe la existencia de tablas de sucesos y de

archivado, inhabilita la propiedad `CheckForEventTableOnInit` estableciendo su valor en `false`. Por omisión, su valor es `true`. No es recomendable establecer el valor en `false`.

Nota: Si su sitio no archivará sucesos en la tabla de archivado, establezca el valor de `ArchiveProcessed` en `false`.

Tablas de sucesos y de archivado

El conector utiliza la tabla de sucesos para poner en cola sucesos que se deben registrar. Si ha establecido el valor de la propiedad `ArchiveProcessed` en `"true"` o no especifica ningún valor, el conector utiliza la tabla de archivado para almacenar sucesos después de actualizar su estado en la tabla de sucesos.

Para cada suceso, el conector obtiene el nombre del objeto comercial, el verbo y la clave contenidos en la tabla de sucesos. El conector utiliza esta información para recuperar la entidad completa de la aplicación. Si la entidad se ha cambiado después de registrar el suceso por primera vez, el conector obtiene el suceso inicial y todos los cambios subsiguientes. Es decir, si una entidad se crea y actualiza antes de que el conector la obtenga de la tabla de sucesos, el conector obtiene ambos cambios de datos en el intervalo individual.

Son posibles los tres resultados siguientes para cada suceso procesado por un conector:

- El suceso se procesó con éxito
- El suceso no se procesó con éxito
- El suceso no se suscribió (para obtener información sobre la suscripción que sea específica del intermediario de integración utilizado, consulte la guía de implementación del intermediario)

Si los sucesos no se suprimen de la tabla de sucesos después de ser recogidos por el conector, malgastan espacio en la tabla. Pero si se suprimen, los sucesos no procesados se pierden y no es posible hacer un análisis retrospectivo del proceso de los sucesos. Por tanto, es recomendable que también cree una tabla de archivado y mantenga asignado el valor `"true"` a la propiedad `ArchiveProcessed`. Cada vez que se suprime un suceso de la tabla de sucesos, primero el conector lo inserta en la tabla de archivado.

Nota: Si se producen problemas al acceder a la base de datos de la aplicación que impiden que el conector suprima un suceso de la tabla de sucesos o inserte un suceso en la tabla de archivado, el conector devuelve el código de retorno `APPRESPONSETIMEOUT`.

Scripts de SQL para instalar las tablas de sucesos y de archivado

Estos scripts se proporcionan sólo como modelo para ayudarle a crear las tablas necesarias para el conector. Para otras bases de datos, cree sus scripts utilizando como guía estos scripts proporcionados.

Los scripts para instalar las tablas de sucesos, de archivado y de identificadores exclusivos para una base de datos Oracle son:

- `event_table_oracle.sql`
- `event_package_oracle.sql`
- `archive_table_oracle.sql`

- uid_table_oracle.sql

Estos archivos están situados en los directorios siguientes:

UNIX:

connectors/Manugistics/dependencies/

Windows:

connectors\Manugistics\dependencies\

Nota: El orden y tipo de datos en las columnas de las tablas es muy importante. Consulte “Esquema de las tablas de sucesos y de archivado” para ver el orden y tipo correctos.

Es recomendable que el administrador de bases de datos o la persona encargada de implementar el conector modifique estos scripts de acuerdo con los requisitos específicos de la instalación y las necesidades de optimización de las consultas. Por ejemplo, estos scripts no crean índices en las tablas. La persona encargada de implementar el conector debe crear índices para mejorar el rendimiento con el optimizador de consultas.

Esquema de las tablas de sucesos y de archivado

La Tabla 4 describe las columnas de las tablas de sucesos y de archivado.

Tabla 4. Esquema de las tablas de sucesos y de archivado

Nombre	Descripción	Tipo	Restricción
id_suceso	Identificador interno del suceso	NUMBER	Clave primaria
id_conector	ID exclusivo del conector para el cual está destinado el suceso. Este valor es importante cuando varios conectores sondan la misma tabla.	VARCHAR	
clave_objeto	Clave primaria del objeto comercial. La clave se puede representar en forma de un par nombre_valor, o como un conjunto de claves delimitadas por un signo de dos puntos u otro delimitador configurable (por ejemplo, 1000065:10056:2333). Consulte la explicación de la propiedad “EventKeyDel” en la página 24 para obtener más información.	VARCHAR	No null
nombre_objeto	Nombre del objeto comercial	VARCHAR	No null
verbo_objeto	Verbo asociado al suceso	VARCHAR	No null
prioridad_suceso	Prioridad del suceso (0 es la más alta, <i>n</i> es la más baja), que el conector utiliza para obtener sucesos en función de la prioridad. El conector no utiliza este valor para aumentar o disminuir prioridades.	NUMBER	No null
tiempo_suceso	Fecha y hora en que se produjo el suceso	DATETIME	Fecha/hora actual por omisión (para la tabla de archivado, la hora real del suceso)
tiempo_archivado	Fecha y hora en que se archivó el suceso (sólo aplicable a la tabla de archivado)	DATETIME	Fecha/hora de archivado
estado_suceso	-2 (Error al enviar suceso al intermediario de integración) -1 (Error al procesar suceso) 0 (Preparado para sondeo) 1 (Enviado al intermediario de integración)	NUMBER	No null

Tabla 4. Esquema de las tablas de sucesos y de archivado (continuación)

Nombre	Descripción	Tipo	Restricción
	2 (No existen suscripciones para el objeto comercial) 3 (En proceso). Este estado sólo se utiliza en la tabla de sucesos; no se utiliza en la tabla de archivado.		
comentario_suceso	Descripción del suceso u error	VARCHAR	

Habilitación del soporte para varios controladores

Puede especificar el controlador siguiendo estos pasos:

1. Instale el controlador en su máquina.
2. Coloque en el subdirectorio `connectors/Manugistics` del directorio del producto, todas las bibliotecas dinámicas que el conector necesita durante la ejecución.
3. Edite el archivo de inicio del conector para incluir en la variable `JDBC_DRIVER_PATH` todas las vías de acceso de clases pertinentes (incluida la información sobre licencias, si es necesario).

En UNIX, el archivo de inicio es:

```
DirProducto/connectors/Manugistics/start_Manugistics.sh
```

En Windows, el archivo de inicio es:

```
DirProducto\connectors\Manugistics\start_Manugistics.bat
```

4. Especifique un valor para la propiedad de configuración `JDBC_DRIVER_CLASS`.

Nota: Para todas las características con las que es compatible, el conector puede trabajar con cualquier controlador que cumpla la especificación JDBC 2.0 o superior. Si el controlador no puede utilizar una característica determinada, el conector no funciona debidamente. Por ejemplo, si el controlador no puede utilizar todas las llamadas de método utilizadas por ODA for Manugistics, el archivo de anotaciones cronológicas de ODA denota el proceso que el controlador no puede utilizar. En ese caso, debe utilizar un controlador diferente.

Habilitación de la clase gestora personalizada de objetos comerciales

El conector es compatible con la clase gestora personalizada de objetos comerciales, `CustomBOH`. Esta clase implementa la interfaz `JDBCBOHandlerInterface`. La sintaxis de esta interfaz es:

```
public interface JDBCBOHandlerInterface{
    public int doVerbForCustom(CWConnectorBusObj busObj) throws
        VerbProcessingFailedException, ConnectionFailureException;
}
```

Cuando implemente el método `doVerbForCustom`, asegúrese de que emita las dos excepciones, pero que no las reciba. Defina también el estado y el mensaje de cada excepción antes de emitirla.

- `VerbProcessingFailedException` — Se emite cuando falla la operación especificada por el verbo.
- `ConnectionFailureException` — Se emite cuando el conector no puede establecer conexión con la aplicación.

Para permitir que el conector pueda trabajar con este gestor de objetos comerciales:

- Especifique el nombre de clase CustomBOH en la información sobre el verbo que es específica de la aplicación.
El conector busca en esa información para obtener el nombre de la clase gestora personalizada de objetos comerciales. Utilice esta sintaxis:
`CustomBOH=nombreClaseGestoraPersonalizadaObjetosCom`
Por ejemplo, suponga que la información sobre el verbo que es específica de la aplicación está especificada de esta manera:
`CustomBOH=JDBCBOhandlerForOverrideSQL`
En este caso, JDBCBOhandlerForOverrideSQL es el nombre de la clase gestora personalizada de objetos comerciales.
- Asegúrese de que CustomBOH pertenece a `com.crossworlds.connectors.JDBC`
Si el conector encuentra "CustomBOH=" en la información sobre el verbo específica de la aplicación y encuentra la clase en el paquete `com.crossworlds.connectors.JDBC`, el conector ejecuta el gestor personalizado de objetos comerciales. Si el conector no encuentra CustomBOH, emite un error para indicar que no pudo encontrar la clase.

Configuración del conector

Debe definir las propiedades de configuración del conector antes de ejecutarlo, tanto las propiedades estándar como las específicas del conector. Utilice la herramienta Configurador de conectores para definir las propiedades de configuración de un conector:

- Si el intermediario de integración es InterChange Server, acceda a esa herramienta desde el Gestor del sistema.
- Si el intermediario de integración es WebSphere MQ Integrator Broker, acceda a esa herramienta desde la carpeta del programa IBM WebSphere Business Integration Adapter.

Para obtener más información sobre el Configurador de conectores, consulte el Apéndice B, "Configurador de conectores", en la página 121.

Propiedades de configuración estándar

Las propiedades de configuración estándar proporcionan información que utilizan todos los conectores. Consulte el Apéndice A, "Propiedades de configuración estándar para conectores", en la página 101 para obtener información detallada sobre estas propiedades.

Importante: Como el conector para productos JDBC puede trabajar con ambos intermediarios de integración, InterChange Server y WebSphere MQ Integrator Broker, las propiedades de configuración de ambos intermediarios son relevantes para el conector.

Además, consulte la Tabla 5 para obtener información de configuración específica del producto IBM WebSphere Business Integration Adapter for JDBC. La información de esta tabla complementa la información del apéndice.

Tabla 5. Información sobre propiedades específica del conector

Propiedades	Notas
CharacterEncoding	Este conector no utiliza la propiedad CharacterEncoding.

Tabla 5. Información sobre propiedades específica del conector (continuación)

Propiedades	Notas
Locale	Como el conector se ha internacionalizado, se puede cambiar el valor de la propiedad Locale. Nota: Si como intermediario utiliza WebSphere MQ Integrator Broker, debe utilizar el mismo entorno local (locale) para el adaptador, el intermediario y las aplicaciones.

Tenga en cuenta que debe proporcionar un valor para la propiedad de configuración ApplicationName antes de ejecutar el conector.

Propiedades específicas del conector

Las propiedades de configuración específicas del conector proporcionan información que el conector necesita durante la ejecución. Las propiedades específicas del conector también proporcionan una manera de cambiar información estática o lógica contenida en el conector, sin tener que volver a codificar y reconstruir el conector.

La Tabla 6 lista las propiedades de configuración del conector que son específicas de él. Consulte las secciones siguientes para obtener explicaciones de las propiedades.

Tabla 6. Propiedades de configuración específicas del conector

Nombre	Valores posibles	Valor por omisión	Obligatorio
ApplicationPassword	Contraseña para la cuenta de usuario del conector		Sí*
ApplicationUserName	Nombre de la cuenta de usuario del conector		Sí*
ArchiveProcessed	true o false	true	No
ArchiveTableName	Nombre de la tabla de la cola de archivado	xworlds_archive_events	Sí si Archive Processed es true
AutoCommit	true o false	false	No
CheckforEventTableInInit	true o false	true	No
ChildUpdatePhyDelete	true o false	false	No
CloseDBConnection	true o false	false	No
ConnectorID	ID exclusivo del conector	null	No
DatabaseURL	Nombre del servidor de bases de datos		Sí
DateFormat	Serie de caracteres para denotar una fecha y una hora	MM/dd/yyyy HH:mm:ss	No
DriverConnectionProperties	Propiedades adicionales de la conexión del controlador JDBC		No
DriverSupportForLong	true o false	true	No
EventKeyDel	Carácter o caracteres para delimitar las columnas de clave de objeto en la tabla de sucesos	punto y coma (;)	No
EventOrderBy	none, NombreColumna, NombreColumna, ...]		No
EventQueryType	Fixed o Dynamic	Fixed	No

Tabla 6. Propiedades de configuración específicas del conector (continuación)

Nombre	Valores posibles	Valor por omisión	Obligatorio
EventTableName	Nombre de la tabla de la cola de sucesos	xworlds_events	Sí, si es necesario el sondeo de la tabla de sucesos; null (como serie de caracteres) si el sondeo no es necesario
JDBCDriverClass	nombre de clase del controlador		Sí
MaximumDatabaseConnections	Número de conexiones simultáneas de base de datos	5	Sí
PingQuery	SELECT 1 FROM <nombre_tabla>		No
PollQuantity	Valores entre 1 y 500	1	No
PreserveUIDSeq	true o false	true	No
QueryTimeOut	Valor entero, expresado en segundos		No
RDBMS.initsession	Sentencia de SQL mediante la que se inicializa cada sesión de base de datos		No
RDBMSVendor	Oracle, Others		Sí
ReplaceAllStr	true o false	false	No
ReplaceStrList	Conjunto formado por un carácter individual, un delimitador de caracteres y la serie de caracteres de sustitución del carácter. También, varios conjuntos así con un delimitador de terminación entre ellos.	Q,DSQ Nota: en la herramienta de configuración de conectores, estos caracteres representan una comilla simple, seguida de una coma y dos comillas simples.	No
RetryCountAndInterval	Cuenta, intervalo en segundos	3,20	No
ReturnDummyBOForSP	true o false	false	No
SchemaName	Esquema en el que residen los sucesos		No
SelectiveCommitForPoll	true o false	false	No
SPBeforePollCall	Nombre del procedimiento almacenado que se debe ejecutar para cada llamada de sondeo		No
StrDelimiter	Delimitadores de caracteres y de terminación utilizados en la propiedad ReplaceStrList	,:	No
TimingStats	0, 1, 2	0	No
UniqueIDTableName	Nombre de la tabla utilizada para crear los ID	xworlds_uid	No
UseDefaults	true o false	false	Sí
UseDefaultsForCreatingChildBOs	true o false	false	No
UseDefaultsWhenPolling	true o false	true	No

* No es necesario utilizar ApplicationPassword ni ApplicationUserName si se utiliza la autenticación fiable.

ApplicationPassword

Contraseña para la cuenta de usuario del conector.

No existe valor por omisión.

ApplicationUserName

Nombre de la cuenta de usuario del conector.

No existe valor por omisión.

ArchiveProcessed

Especifica si el conector archiva sucesos para los cuales no existen suscripciones actuales.

Asigne el valor `true` a esta propiedad si desea que los sucesos se inserten en la tabla de archivado antes de ser suprimidos de la tabla de sucesos.

Asigne el valor `false` a esta propiedad si desea que el conector no realice el proceso de archivado. En este caso, el conector no comprueba el valor de la propiedad `ArchiveTableName`. Si `ArchiveProcessed` tiene el valor `false`, el conector realiza estas acciones:

- Si el suceso se procesa satisfactoriamente, el conector lo suprime de la tabla de sucesos y no lo archiva.
- Si el conector no se suscribe al objeto comercial del suceso, el conector deja el suceso en la tabla de sucesos y cambia su estado a `Unsubscribed`. Para obtener información sobre la suscripción que sea específica del intermediario de integración utilizado, consulte la guía de implementación del intermediario.
- Si el objeto comercial encuentra un problema mientras se procesa, el conector deja el suceso en la tabla de sucesos y el estado del suceso se establece en `Error`.

Si el valor de esta propiedad se establece en `false` y el volumen de sondeo (`PollQuantity`) es bajo, el conector parece estar sondeando la tabla de sucesos, pero en realidad está simplemente captando los mismos sucesos repetidamente.

Si esta propiedad no tiene ningún valor asignado, el conector considera que el valor es `true`. Si la propiedad `ArchiveTableName` tampoco tiene ningún valor asignado, el conector considera que el nombre de la tabla de archivado es `xworlds_archive_events`.

El valor por omisión es `true`.

ArchiveTableName

Nombre de la tabla de la cola de archivado.

Si el valor de la propiedad `ArchiveProcessed` está establecido en `false`, no es necesario especificar un valor para la propiedad `ArchiveTableName`.

El nombre por omisión es `xworlds_archive_events`.

AutoCommit

Esta propiedad permite configurar el valor de `AutoCommit`. Cuando el valor es `true`, todas las transacciones se confirman automáticamente. Algunas bases de datos (tales como Sybase) necesitan que `AutoCommit` se establezca en `true`. Si se establece en `false`, los procedimientos almacenados de Sybase fallarán.

Si se pierde la conexión con la base de datos, el conector intentará crear una nueva conexión para reiniciar el proceso completo siempre que `AutoCommit` tenga el valor `false`. Si la nueva conexión no es válida o si `AutoCommit` tiene el valor `true`, el conector devuelve el código de retorno `APPRESPONSETIMEOUT`, que hace que concluya la ejecución del conector.

El valor por omisión es `false`.

CheckforEventTableInInit

Si esta propiedad del conector se establece en `false`, se impide que el conector compruebe la existencia de las tablas de sucesos y de archivado durante la inicialización del conector. Es recomendable que establezca siempre esta propiedad en `true` a menos que el controlador JDBC utilizado no sea compatible con la clase `DatabaseMetaData` de JDBC.

Cuando el valor de esta propiedad es `false`, aunque el conector no comprueba la existencia de las tablas de sucesos y de archivado, estas tablas deben existir siempre, pues el conector las utiliza durante el proceso de inicialización. Para impedir que el conector utilice las tablas de sucesos y de archivado durante la inicialización, establezca la propiedad `EventTableName` en `null`.

El valor por omisión es `true`.

ChildUpdatePhyDelete

Durante una operación de actualización, esta propiedad especifica cómo el conector maneja los datos representados por un objeto comercial hijo que falta en el objeto comercial entrante, pero que existe en la base de datos.

Establezca esta propiedad en `true` si desea que el conector suprima físicamente el registro de datos de la base de datos.

Establezca esta propiedad en `false` si desea que el conector realice una supresión lógica del registro de datos en la base de datos asignando el valor apropiado a la columna de estado. La información específica de la aplicación obtiene el nombre de la columna de estado y su valor a partir del parámetro `StatusColumnValue` (SCN) especificado en la información específica de la aplicación a nivel de objeto comercial. Para obtener más información, consulte "Información específica de la aplicación a nivel de objeto comercial" en la página 63.

El valor por omisión es `false`.

CloseDBConnection

Esta propiedad permite configurar el cierre de la conexión de base de datos. Cuando su valor es `true`, la conexión de base de datos se cierra para cada petición de llamada de servicio y llamada de sondeo. No es aconsejable asignar el valor `true` a esta propiedad, pues ello disminuye el rendimiento.

El valor por omisión es `false`.

ConnectorID

Es un ID exclusivo para el conector. Este ID es útil para recuperar sucesos para una instancia determinada del conector.

El valor por omisión es `null`.

DatabaseURL

Nombre del servidor de bases de datos al cual se debe conectar el conector.

Si utiliza el controlador SQLServer de WebSphere Business Integration designado por el sistema, el URL recomendado es:

```
jdbc:ibm-crossworlds:sqlserver://nombre_máquina:número_puerto;DatabaseName=nombre_base_datos
```

Importante

Si AutoCommit se establece en false, debe definir un parámetro adicional, SelectMethod :jdbc:ibm-crossworlds:sqlserver://nombre_máquina:número_puerto;DatabaseName=nombre_base_datos;SelectMethod=cursor

Por omisión, SelectMethod se establece en direct. Para obtener más información, consulte "AutoCommit" en la página 21.

Debe asignar un valor a esta propiedad para que el conector procese satisfactoriamente.

DateFormat

Especifica el formato de fecha que el conector espera recibir y devolver. Esta propiedad es compatible con cualquier formato que esté basado en la sintaxis que se describe en la Tabla 7.

La Tabla 7 define la sintaxis de DateFormat utilizando una serie de caracteres para denotar una fecha y una hora. En esta serie de caracteres, todas las letras ASCII son letras reservadas.

Tabla 7. Sintaxis de DateFormat

Símbolo	Significado	Presentación	Ejemplo
G	designador de la era	(Texto)	AD
y	año	(Número)	1996
M	mes del año	(Texto & Número)	Julio & 07
d	día del mes	(Número)	10
h	hora en notación am/pm (1-12)	(Número)	12
H	hora del día (0-23)	(Número)	0
m	minuto dentro de la hora	(Número)	30
s	segundo dentro del minuto	(Número)	55
S	milisegundo	(Número)	978
E	día de la semana	(Texto)	Martes
D	día del año	(Número)	189
F	día de la semana dentro del mes	(Número)	2 (2 miércoles de julio)
w	semana dentro del año	(Número)	27
W	semana dentro del mes	(Número)	2
a	marcador am/pm	(Texto)	PM
k	hora del día (1-24)	(Número)	24
K	hora en notación am/pm (0-11)	(Número)	0
z	zona horaria	(Texto)	Hora estándar del Pacífico
'	carácter de escape para texto	(Delimitador)	
''	comilla simple	(Literal)	'

Tabla 8. Ejemplos para el entorno local de Estados Unidos

Patrón de formato	Resultado
"aaaa.MM.dd G 'at' hh:mm:ss z"	1996.07.10 AD at 15:08:56 PDT
"EEE, MMM d, ''aa"	Wed, July 10, '96
"h:mm a"	12:08 PM
"hh 'o''clock' a, zzzz"	12 o'clock PM, Pacific Daylight Time
"K:mm a, z"	0:00 PM, PST
"aaaa.MMMMM.dd GGG hh:mm aaa"	1996.July.10 AD 12:08 PM

DriverConnectionProperties

Además del nombre de usuario y la contraseña, puede ser necesario especificar otras propiedades o información para un controlador JDBC. La propiedad de conectar `DriverConnectionProperties` acepta propiedades adicionales que pueden ser necesarias para un controlador JDBC, especificadas en forma de pares nombre-valor. Las propiedades se deben especificar de esta manera:

```
propiedad1=valor1[;propiedad2=valor2...]
```

Las propiedades se deben proporcionar en forma de pares nombre-valor, separados por signos de punto y coma. La propiedad está separada de su valor por un signo de igualdad (sin espacios en blanco adicionales).

Por ejemplo, suponga que es necesario especificar la información sobre licencias y el número de puerto para el controlador JDBC. El nombre de propiedad que el controlador espera para la información sobre licencias es `MyLicense` y el valor es `ab23jk5`. El nombre de propiedad que el controlador espera para el número de puerto es `PortNumber` y el valor es `1200`. La propiedad `DriverConnectionProperties` se debe establecer en este valor: `MyLicense=ab23jk5;PortNumber=1200`.

DriverSupportForLong

Especifica cómo se manejan los parámetros enteros para sentencias preparadas. Cuando el valor de esta propiedad es "true", se utiliza `setLong` para definir el parámetro entero. Cuando el valor de la propiedad es "false", se utiliza `setInt` para definir el parámetro.

El valor por omisión es "true".

EventKeyDel

Especifica el delimitador cuando la columna `object_key` de la tabla de sucesos contiene varios valores de atributos.

Existen dos formas de recuperar el objeto comercial que se ha creado, actualizado o suprimido en la aplicación desencadenante.

- El primer método es llenar la columna `object_key` con valores de atributos que son claves en un objeto comercial. Establezca la propiedad de configuración `EventKeyDel` en un carácter individual que no forme parte del campo de clave. Por **ejemplo**, si el delimitador se especifica como ";", entonces `object_key` será: `xxx;123`
- El segundo método es llenar la columna `object_key` con valores de cualquier atributo de un objeto comercial. Estos valores se deben representar en forma de pares nombre-valor. El primer delimitador será para el nombre-valor y el segundo es para las claves. Por ejemplo, si el delimitador se especifica como "=", entonces `object_key` será: `CustomerName=xxx;CustomerId=123;`

Si el delimitador se especifica como "=", entonces `object_key` será:
`CustomerName=xxx:CustomerId=123:`

Nota: El orden en el que se definen los valores de clave debe ser el mismo orden que los atributos de clave en un objeto comercial.

Importante: Si utiliza datos para un atributo de fecha, evite utilizar el signo de dos puntos (:) como delimitador, pues puede estar incluido en los datos del atributo.

El valor por omisión es un signo de punto y coma (;), el cual está basado en claves, no en pares nombre-valor.

EventOrderBy

Especifica si se debe desactivar la ordenación de sucesos, o especifica un orden para el proceso de sucesos que es diferente del orden por omisión.

Por omisión, en cada sondeo el conector recupera sólo el número de sucesos especificado en la propiedad de conector `Pol1Quantity`, y ordena el proceso de sucesos de acuerdo con los valores de las columnas `event_time` y `event_priority` de la tabla de sucesos.

Para hacer que el conector no ordene los sucesos, establezca esta propiedad en `none`.

Para hacer que el conector ordene los sucesos de acuerdo con columnas diferentes de la tabla de sucesos, especifique los nombres de esas columnas. Separe los nombres de las columnas con una coma (,). El valor especificado para esta propiedad prevalece sobre el comportamiento definido por omisión para el conector.

No existe valor por omisión para esta propiedad.

EventQueryType

La propiedad `EventQueryType` se utiliza para indicar si el conector debe generar dinámicamente una consulta para recuperar sucesos de la tabla de sucesos o debe utilizar la consulta definida internamente en el conector. Para la consulta generada dinámicamente, el conector correlaciona su estructura de sucesos con las columnas de la tabla de sucesos. El orden de los datos en las columnas de la tabla es muy importante. Consulte "Esquema de las tablas de sucesos y de archivado" en la página 16 para ver el orden correcto.

Si el valor de `EventQueryType` es `Fixed` (como serie de caracteres), se ejecuta la consulta definida por omisión. Si el valor es `Dynamic` (como serie de caracteres), se crea una nueva consulta obteniendo los nombres de las columnas de la tabla que está especificada en la propiedad "`EventTableName`" en la página 26.

Los nombres de las columnas de la tabla de sucesos pueden cambiar, pero el orden y tipo de datos de las columnas deben seguir siendo los especificados en la sección referente a la creación de la tabla de sucesos. "`EventOrderBy`" se añadirá a la consulta definida por omisión o a la consulta generada dinámicamente.

Si no se añade la propiedad `EventQueryType` o no contiene ningún valor, su valor se establece en `Fixed` por omisión.

El valor por omisión es `Fixed` (como serie de caracteres).

EventTableName

Es el nombre de la tabla de la cola de sucesos, que es utilizado por el mecanismo de sondeo del conector.

El nombre por omisión es `xworlds_events`.

Establezca esta propiedad en `null` (como serie de caracteres) cuando el sondeo esté desactivado para el conector. De esta forma se evita que se compruebe la existencia de las tablas de sucesos y de archivado.

Para una tabla de sucesos definida por el usuario, asegúrese de que `id_suceso` se correlaciona con uno de los tipos siguientes de JDBC: `INTEGER`, `BIGINT`, `NUMERIC`, `VARCHAR`.

JDBCDriverClass

Especifica el nombre de clase de un controlador. Para utilizar un controlador JDBC determinado, especifique el nombre de clase del controlador en esta propiedad de configuración. Por ejemplo, para especificar el controlador `thin` de Oracle, establezca el valor de esta propiedad en: `oracle.jdbc.driver.OracleDriver`.

Para obtener más información, consulte “Habilitación del soporte para varios controladores” en la página 17 y “UseDefaultsForCreatingChildBOs” en la página 31.

No se proporciona ningún valor por omisión.

MaximumDatabaseConnections

Especifica el número máximo de conexiones simultáneas que se permiten establecer con la base de datos. Durante la ejecución, el número de conexiones abiertas con la base de datos es la suma de este valor más 1.

Si el valor de la propiedad “`PreserveUIDSeq`” en la página 27 se establece en “`false`”, el número de conexiones de base de datos abiertas durante la conexión es la suma del valor `MaximumDatabaseConnections` más 2.

El valor por omisión es 5.

PingQuery

Especifica la sentencia de SQL o procedimiento almacenado que el conector ejecuta para comprobar (sondear) la conectividad con la base de datos.

Lo siguiente es un ejemplo de sentencia de SQL utilizada como consulta de sondeo:

```
SELECT 1 FROM <nombre_tabla>
```

Lo siguiente es un ejemplo de una llamada de procedimiento almacenado (`sampleSP`) utilizada como consulta de sondeo con una base de datos Oracle:

```
call sampleSP( )
```

Observe que las llamadas de procedimiento almacenado no pueden tener parámetros de salida. Si es necesario un parámetro de entrada para la base de datos, el valor de entrada se debe especificar como parte de la consulta de sondeo. Por ejemplo:

```
Call checkproc(2)
```

No existe valor por omisión. Para obtener más información, consulte “Manejo de conexiones perdidas de base de datos” en la página 6 y “Pérdida de la conexión con la aplicación” en la página 98.

PollQuantity

Es el número de filas de la tabla de base de datos que el conector recupera en cada intervalo de sondeo. Los valores permitidos están comprendidos entre 1 y 500.

El valor por omisión es 1.

PreserveUIDSeq

Especifica si la secuencia identificadora exclusiva entrante se conservará o no en la tabla de identificadores exclusivos.

Si el valor de esta propiedad es true, el ID exclusivo no se confirma hasta que se procesa satisfactoriamente el objeto comercial en la aplicación de destino. Todos los demás procesos que intenten acceder a la tabla de identificadores exclusivos deberán esperar hasta que se confirme la transacción.

Si el valor de esta propiedad es false, el ID exclusivo se confirma cuando lo solicita el objeto comercial. El proceso del objeto comercial y el proceso del ID exclusivo tienen su propio bloque de transacciones (interno respecto al conector). Esto sólo es posible si la transacción referente a la tabla de identificadores exclusivos tiene su propia conexión.

Nota: Si esta propiedad no se añade a la configuración del conector, el comportamiento por omisión del conector es el mismo que si la propiedad se añadiera y su valor se estableciera en true. Además, si el valor de “AutoCommit” en la página 21 es true, el conector tiene el mismo comportamiento que si PreserveUIDSeq se estableciera en false.

Si el valor de la propiedad “PreserveUIDSeq” se establece en “false”, el número de conexiones de base de datos abiertas durante la conexión es la suma de este valor más 2.

El valor por omisión es true.

QueryTimeOut

El valor de esta propiedad es un número entero de segundos que define el tiempo de espera de las consultas (QueryTimeOut) para todas las acciones de recuperación. Si no se especifica este valor, se supone que no existe ningún tiempo de espera definido para la consulta. Si la consulta emplea más tiempo que el número de segundos especificado, la base de datos emite una excepción de SQL que es capturada. El mensaje asociado se registra en el archivo de anotaciones cronológicas.

No se proporciona ningún valor por omisión.

RDBMS.initsession

Es la sentencia de SQL por la que se inicializa cada sesión con la base de datos. El conector recibe una consulta y la ejecuta durante el proceso de inicio. Normalmente esta consulta no produce ningún valor de retorno. Es necesario el nombre de la propiedad, pero no es necesario un valor.

No existe valor por omisión.

RDBMSVendor

Especifica el sistema de gestión de bases de datos relacionales (RDBMS) que el conector utiliza para procesos especiales. Establece el valor de esta propiedad en `Oracle` para la base de datos Oracle.

Si utiliza una base de datos diferente de la base de datos por omisión, asegúrese de que se cargue el controlador apropiado. Si el valor de esta propiedad es `Others`, el conector localiza el controlador para determinar la base de datos que se debe utilizar.

Es necesario especificar un valor para que el conector se ejecute satisfactoriamente.

No se proporciona ningún valor por omisión.

ReplaceAllStr

Especifica si el conector sustituye todas las apariciones de cada carácter identificado en la propiedad `ReplaceStrList` por la serie de caracteres de sustitución especificada en esa propiedad. El conector evalúa `ReplaceAllStr` sólo si el parámetro `ESC=[true|false]` de la propiedad `AppSpecificInfo` de cada atributo no contiene un valor. Es decir, si se ha especificado el parámetro `ESC`, su valor tiene prioridad sobre el valor definido para la propiedad `ReplaceAllStr`. Para hacer que el conector utilice el valor de `ReplaceAllStr`, verifique que no se haya especificado el parámetro `ESC`.

El valor por omisión de `ReplaceAllStr` es `false`.

Nota: El parámetro `ESC` y las propiedades `ReplaceAllStr` y `ReplaceStrList` son compatibles con la utilización de caracteres de escape para una base de datos (por ejemplo, comillas simples utilizadas como carácter de escape). Como esta misma funcionalidad está también disponible utilizando las sentencias preparadas proporcionadas por el controlador JDBC, estas propiedades se eliminarán en versiones futuras del conector. El conector es actualmente compatible con el uso de las sentencias preparadas de JDBC.

ReplaceStrList

Especifica uno o más conjuntos de sustitución, cada uno formado por un carácter individual a sustituir, un delimitador de caracteres y una serie de caracteres de sustitución. El conector efectúa esta sustitución para un valor de atributo sólo si se ha especificado un valor para el parámetro `ESC=[true|false]` de la propiedad `AppSpecificInfo` del atributo o para la propiedad `ReplaceAllStr` del conector.

Nota: El parámetro `ESC` y las propiedades `ReplaceAllStr` y `ReplaceStrList` son compatibles con la utilización de caracteres de escape para una base de datos (por ejemplo, comillas simples utilizadas como carácter de escape). Como esta misma funcionalidad está también disponible utilizando las sentencias preparadas proporcionadas por el controlador JDBC, estas propiedades se eliminarán en versiones futuras del conector. El conector es actualmente compatible con el uso de las sentencias preparadas de JDBC.

La sintaxis de este atributo es:

```
carácter_individual1,serie_sustitución1[:carácter_individual2,serie_sustitución2[:...]]
```

donde:

carácter_individual Es un carácter a sustituir.

<i>serie_sustitución</i>	Es la serie de caracteres de sustitución que el conector utiliza para sustituir el carácter.
,	Es el delimitador de caracteres, el cual separa el carácter a sustituir respecto de la serie de caracteres que sustituirá al carácter. El delimitador de caracteres por omisión es una coma (.). Puede configurar este delimitador estableciendo el primer delimitador en la propiedad <code>StrDelimiter</code> .
:	Es el delimitador de terminación, el cual separa conjuntos de sustitución (cada uno de los cuales está formado por el carácter a sustituir, un delimitador de caracteres y la serie de caracteres de sustitución). El delimitador de terminación por omisión es un signo de dos puntos (:). Puede configurar este delimitador estableciendo el segundo delimitador en la propiedad <code>StrDelimiter</code> .

Por ejemplo, suponga que desea sustituir un signo individual de porcentaje (%) por dos signos de porcentaje (%%), y sustituir un signo de intercalación (^) por una barra inclinada invertida y un signo de intercalación (\^). Por omisión, `StrDelimiter` especifica una coma (,) como delimitador de caracteres, y un signo de dos puntos (:) como delimitador de terminación. Si conserva los delimitadores por omisión, utilice la serie de caracteres siguiente como valor de `ReplaceStrList`:

```
%,%:^\,\^
```

Nota: Una restricción de la herramienta de configuración de conectores impide especificar comillas simples. Por tanto, debe utilizar el carácter Q para representar una comilla simple, y los caracteres DSQ para representar dos comillas simples. En el ejemplo anterior, si también desea sustituir una comilla simple (') por dos comillas simples (' '), utilice esta notación: `Q,DSQ:%,%:^\,\^`

RetryCountAndInterval

Especifica el número de intentos y el intervalo en segundos que el conector debe utilizar cuando no puede bloquear datos al realizar una operación de actualización.

Antes de realizar una actualización, el conector bloquea las filas afectadas por la actualización e intenta recuperar datos actuales. Si el conector no puede bloquear las filas, intenta de nuevo realizar el bloqueo de acuerdo con los valores de cuenta e intervalo especificados en esta propiedad de configuración. Finalmente se produce un tiempo de espera excedido para el conector si no se puede obtener el bloqueo dentro de los valores especificados en esta propiedad.

Especifique el valor utilizando este formato: cuenta, intervalo en segundos. Por ejemplo, el valor 3,20 especifica tres reintentos con un intervalo de 20 segundos entre ellos.

El valor por omisión es 3,20.

ReturnDummyBOForSP

Esta propiedad se utiliza para devolver parámetros de salida aún cuando el conjunto de resultados esté vacío.

En el caso de RetrieveSP, se devuelve un conjunto de resultados. Si el conjunto de resultados está vacío, no se crean objetos comerciales, y los parámetros de salida devueltos por la llamada de procedimiento no se pueden recuperar. Sin embargo, si el valor de ReturnDummyBOForSP es true, se devuelve un objeto comercial ficticio cuyos valores de atributo se han obtenido de parámetros de salida y parámetros de entrada/salida.

El valor por omisión es "false".

SchemaName

Esta propiedad restringe la búsqueda de las tablas de sucesos y de archivado al esquema especificado por esta propiedad. Si esta propiedad no se añade o se deja vacía, el conector buscará en todos los esquemas a los que tenga acceso el usuario. El nombre de esquema especificado en esta propiedad también se utiliza al crear las consultas para acceder a las tablas de sucesos y de archivado.

La base de datos Oracle es compatible con la utilización de nombres de esquema. Consulte la documentación del controlador JDBC para obtener información específica.

No se proporciona ningún valor por omisión.

SelectiveCommitForPoll

Especifica cuándo tendrán lugar las operaciones de confirmación para la base de datos. Si el valor es true, se producirá una sola confirmación, una vez procesado el suceso. Si el valor es false, se produce el comportamiento estándar; es decir, las confirmaciones se producen en cada etapa del proceso de los sucesos.

El valor por omisión es "false".

SPBeforePollCall

Esta propiedad especifica el procedimiento almacenado que se ejecuta para cada llamada de sondeo. Si la propiedad SPBeforePollCall tiene un valor (el nombre de un procedimiento almacenado), entonces, al inicio de cada llamada de sondeo, el conector invoca el procedimiento almacenado, pasándole los valores de las propiedades de conector ConnectorID y PollQuantity. El procedimiento actualizará el número de filas especificado por PollQuantity, y establecerá la columna del ID de conector en ConnectorID, donde status=0 y el ID de conector es null. Esto permite el equilibrado de la carga de trabajo en el conector.

Nota: En el caso en que una llamada de sondeo falla prematuramente (la base de datos está inactiva o se pierde la conexión), el ID de conector permanece definido. Esto puede hacer que se pasen por alto registros durante el sondeo. Por tanto, es recomendable restablecer periódicamente el ID de conector en el valor null para todos los registros de la tabla de sucesos cuyo estado sea 0.

StrDelimiter

Especifica los delimitadores de caracteres y de terminación que se deben utilizar en la propiedad ReplaceStrList.

- El delimitador de caracteres separa el carácter a sustituir respecto de la serie de caracteres que sustituirá al carácter. El delimitador de caracteres ocupa la primera posición (por la izquierda) en los valores de esta propiedad y su valor por omisión es una coma (,).
- El delimitador de terminación separa conjuntos de sustitución (cada uno de los cuales está formado por el carácter a sustituir, un delimitador de caracteres y la

serie de caracteres de sustitución). El delimitador de terminación ocupa la segunda posición (por la derecha) en los valores de esta propiedad y su valor por omisión es un signo de dos puntos (:).

El usuario puede especificar un valor propio para cualquiera de estos delimitadores o para ambos. En este caso, no especifique un espacio en blanco ni ningún otro carácter entre los caracteres delimitadores.

El valor por omisión es una coma seguida inmediatamente por un signo de dos puntos (,:)

TimingStats

Esta propiedad permite calcular la duración de cada operación de verbo del conector para detectar problemas. Los valores posibles son:

- 0 No se realizan estadísticas de tiempo
- 1 Se muestra la hora al comienzo y final de la operación de verbo para un objeto comercial jerárquico completo.
- 2 Se muestra la hora al comienzo y final de cada operación de verbo para cada objeto comercial individual de un objeto comercial jerárquico.

Los mensajes de cronometraje son mensajes de anotaciones cronológicas más que mensajes de rastreo. Se pueden activar y desactivar, con independencia de los niveles de rastreo.

El valor por omisión es 0.

UniqueIDTableName

Especifica la tabla donde reside el último valor utilizado para la creación de un ID exclusivo. Por omisión, la tabla tiene una sola columna (id). Puede personalizar la tabla para añadir una columna para cada atributo que necesite la creación de un UID.

El valor por omisión es `xworlds_uid`.

UseDefaults

Si `UseDefaults` tiene el valor `true` o no está definido, el conector comprueba si se proporciona un valor válido o valor por omisión para cada atributo necesario de objeto comercial. Si se proporciona una valor, la operación de creación se ejecuta satisfactoriamente; de lo contrario, la operación falla.

Si `UseDefaults` tiene el valor `false`, el conector sólo comprueba si se proporciona un valor válido para cada atributo necesario de objeto comercial; la operación de creación falla si no se proporciona un valor válido.

El valor por omisión es `false`.

UseDefaultsForCreatingChildBOs

Si `UseDefaultsForCreatingChildBOs` tiene el valor `true` o no está definido, el conector comprueba si se proporciona un valor válido o valor por omisión para cada atributo necesario de objeto comercial. Si se proporciona una valor, la operación de creación se ejecuta satisfactoriamente; de lo contrario, la operación falla.

Si `UseDefaultsForCreatingChildBOs` tiene el valor `false`, el conector sólo comprueba si se proporciona un valor válido para cada atributo necesario de objeto comercial; la operación de creación falla si no se proporciona un valor válido.

El valor por omisión es `"false"`.

UseDefaultsWhenPolling

Si `UseDefaultsWhenPolling` tiene el valor `true`, los valores por omisión se establecerán en el objeto comercial antes de recuperarlo de la base de datos y enviarlo al servidor.

Si `UseDefaultsWhenPolling` tiene el valor `false`, los valores por omisión no se establecerán en el objeto comercial antes de recuperarlo de la base de datos y enviarlo al servidor.

El valor por omisión es `true`.

Nota: Esta propiedad de configuración específica del conector sustituye a `UseDefaultsForRetrieve`.

Creación de varias instancias de un conector

Crear varias instancias de un conector es en muchos aspectos lo mismo que crear un conector personalizado. Puede configurar el sistema para crear y ejecutar varias instancias de un conector siguiendo los pasos siguientes. Debe:

- Crear un nuevo directorio para la instancia de conector
- Asegurarse de que tiene las definiciones necesarias de objeto comercial
- Crear un nuevo archivo de definición de conector
- Crear un nuevo script de inicio

Crear un nuevo directorio

Debe crear un directorio de conector para cada instancia de conector. Este directorio de conector se debe llamar:

```
DirProducto\connectors\instanciaConector
```

donde `instanciaConector` identifica de forma exclusiva la instancia de conector.

Si el conector tiene algún metaobjeto específico del conector, debe crear un metaobjeto para la instancia de conector. Si guarda el metaobjeto como archivo, cree este directorio y guarde el archivo en él:

```
DirProducto\Repository\instanciaConector
```

Crear definiciones de objeto comercial

Si dentro del proyecto no existen todavía definiciones de objeto comercial para cada instancia de conector, debe crearlas.

1. Si necesita modificar definiciones de objeto comercial que están asociadas al conector inicial, copie los archivos apropiados y utilice el Diseñador de objetos comerciales para importarlos. Puede copiar cualquiera de los archivos correspondientes al conector inicial. Simplemente cambie el nombre de los archivos si realiza cambios en ellos.
2. Los archivos correspondientes al conector inicial deben residir en este directorio:

```
DirProducto\Repository\instanciaConectorInicial
```

Cualquier archivo adicional que cree debe estar en el subdirectorio apropiado `instanciaConector` de `DirProducto\repository`.

Crear una definición de conector

Debe crear un archivo de configuración (definición de conector) para la instancia de conector en el Configurador de conectores. Para ello siga estos pasos:

1. Copie el archivo de configuración (definición de conector) del conector inicial y cambie el nombre del archivo.
2. Asegúrese de que cada instancia de conector lista correctamente los objetos comerciales con los que puede trabajar (y todos los metaobjetos asociados).
3. Personalice las propiedades del conector según sea necesario.

Crear un script de inicio

Para crear un script de inicio:

1. Copie el script de inicio del conector inicial y cambie su nombre para que incluya el nombre del directorio del conector:
`dirname`
2. Coloque este script de inicio en el directorio que ha creado en “Crear un nuevo directorio” en la página 32.
3. Cree un acceso directo para el script de inicio (Windows solamente).
4. Copie el texto del acceso directo del conector inicial y cambie el nombre del conector inicial (en la línea de mandatos) para que coincida con el nombre de la nueva instancia de conector.

Ahora puede ejecutar al mismo tiempo ambas instancias del conector en el servidor de integración.

Para obtener más información sobre la creación de conectores personalizados, consulte el manual *Connector Development Guide for C++* o *Connector Development Guide for Java*.

Inicio del conector

Un conector se debe iniciar explícitamente utilizando su **script de inicio de conector**. El script de inicio debe residir en el directorio de ejecución del conector:

`DirProducto\connectors\nombreConector`

donde `nombreConector` identifica el conector. El nombre del script de inicio depende de la plataforma de sistema operativo, tal como muestra la Tabla 9.

Tabla 9. Scripts de inicio para un conector

Sistema operativo	Script de inicio
Sistemas basados en UNIX	<code>connector_manager_nombreConector</code>
Windows	<code>start_nombreConector.bat</code>

Puede invocar el script de inicio del conector de cualquiera de las maneras siguientes:

- En los sistemas Windows, desde el menú **Inicio**
Seleccione **Programas > IBM WebSphere Business Integration Adapters > Adapters > Connectors**. Por omisión, el nombre del programa es “IBM

WebSphere Business Integration Adapters". Pero el nombre se puede personalizar. Como alternativa, puede crear un acceso directo en el escritorio para el conector.

- Desde la línea de mandatos:
 - En los sistemas Windows:
`start_nombreConector nombreConector nombreIntermediario [-carchivoConfiguración]`
 - En los sistemas basados en UNIX:
`connector_manager_nombreConector -start`donde *nombreConector* es el nombre del conector y *nombreIntermediario* identifica el intermediario de integración, de la manera siguiente:
 - Para WebSphere InterChange Server, especifique para *nombreIntermediario* el nombre de la instancia de ICS.
 - Para los intermediarios de mensajes WebSphere (WebSphere MQ Integrator, WebSphere MQ Integrator Broker o WebSphere Business Integration Message Broker) o WebSphere Application Server, especifique para *nombreIntermediario* una serie de caracteres que identifique al intermediario.

Nota: Para un intermediario de mensajes WebSphere o WebSphere Application Server en un sistema Windows, *debe* incluir la opción `-c` seguida del nombre del archivo de configuración del conector. Para ICS, especificar `-c` es opcional.

- Desde Adapter Monitor (sólo para el producto WebSphere Business Integration Adapters), que se inicia al iniciar el Gestor del sistema
Puede utilizar esa herramienta para cargar, activar, desactivar, pausar, concluir o suprimir un conector.
- Desde System Monitor (sólo para el producto WebSphere InterChange Server)
Puede utilizar esa herramienta para cargar, activar, desactivar, pausar, concluir o suprimir un conector.
- En los sistemas Windows, puede configurar el conector para que se inicie como servicio de Windows. En este caso, el conector se inicia cuando arranca el sistema Windows (para un servicio configurado como Auto) o cuando el usuario inicia el servicio desde la ventana Servicios de Windows (para un servicio configurado como Manual).

Para obtener más información sobre cómo iniciar un conector, incluidas las opciones de inicio desde la línea de mandatos, consulte uno de los documentos siguientes:

- Para WebSphere InterChange Server, consulte el manual *System Administration Guide*.
- Para los intermediarios de mensajes WebSphere, consulte el manual *Implementing Adapters with WebSphere Message Brokers*.
- Para WebSphere Application Server, consulte el manual *Implementing Adapters with WebSphere Application Server*.

Detención del conector

La forma de detener un conector depende de la forma en que se inició el conector, de la manera siguiente:

- Si inició el conector desde la línea de mandatos, mediante el script de inicio del conector:

- En los sistemas Windows, la invocación del script de inicio crea una ventana de "consola" separada para el conector. En esta ventana, escriba "Q" y pulse Intro para detener el conector.
- En los sistemas basados en UNIX, los conectores se ejecutan en segundo plano, por lo que no tienen una ventana separada. En su lugar, utilice el mandato siguiente para detener el conector:

```
connector_manager_nombreConector -stop
```

donde *nombreConector* es el nombre del conector.
- Desde Adapter Monitor (sólo para el producto WebSphere Business Integration Adapters), que se inicia al iniciar el Gestor del sistema
Puede utilizar esa herramienta para cargar, activar, desactivar, pausar, concluir o suprimir un conector.
- Desde System Monitor (sólo para el producto WebSphere InterChange Server)
Puede utilizar esa herramienta para cargar, activar, desactivar, pausar, concluir o suprimir un conector.
- En los sistemas Windows, puede configurar el conector para que se inicie como servicio de Windows. En este caso, el conector se detiene cuando concluye el sistema Windows.

Capítulo 3. Descripción de los objetos comerciales en su relación con el conector

Este capítulo describe cómo el conector de JDBC procesa objetos comerciales y describe los supuestos sobre los que trabaja el conector al recuperar y modificar datos. Este capítulo contiene las secciones siguientes:

- “Convenios de denominación para objetos comerciales y atributos”
- “Estructura de los objetos comerciales”
- “Procesos de verbos de objeto comercial” en la página 42
- “Propiedades de atributos de objetos comerciales” en la página 60
- “Información específica de la aplicación para objetos comerciales” en la página 62

Puede utilizar esta información como guía para modificar objetos comerciales existentes o implementar unos nuevos. Para obtener información sobre el programa de utilidad que automatiza la creación de archivos de definición de objetos comerciales de WebSphere Business Integration Adapter, consulte el Capítulo 4, “Creación de definiciones de objetos comerciales utilizando IBM ODA for Manugistics”, en la página 75.

El conector actúa sobre la base de determinadas premisas referentes a la estructura de los objetos comerciales con los que trabaja, las relaciones entre los objetos comerciales padre y hijo, el formato de la información específica de la aplicación y la representación del objeto comercial en la base de datos. Por tanto, cuando cree o modifique un objeto comercial que será procesado por el conector, las modificaciones que realice deben ajustarse a las normas que está previsto que el conector siga. De lo contrario, el conector no puede procesar correctamente los objetos comerciales nuevos o modificados.

Convenios de denominación para objetos comerciales y atributos

El nombre de un objeto comercial utilizado por el conector sólo puede estar formado por caracteres alfanuméricos y el carácter de subrayado. Los nombres de atributos de objetos comerciales también sólo pueden constar de caracteres alfanuméricos y el carácter de subrayado.

Estructura de los objetos comerciales

En la mayoría de los casos, el conector supone que cada objeto comercial individual está representado por una sola tabla o vista de base de datos, y que cada **atributo simple** (es decir, un atributo que representa un valor individual, tal como String, Integer o Date) contenido en el objeto está representado por una columna de esa tabla o vista. Por tanto, los atributos contenidos en el mismo objeto comercial individual no se pueden guardar en tablas de base de datos diferentes. Sin embargo, pueden darse las situaciones siguientes:

- La tabla de base de datos puede tener más columnas que el número de atributos simples contenidos en el correspondiente objeto comercial individual (es decir, algunas columnas de la base de datos no están representadas en el objeto comercial). El usuario debe incluir en su diseño solamente aquellas columnas necesarias para el proceso de objetos comerciales.

- El objeto comercial individual puede tener más atributos simples que el número de columnas contenidas en la correspondiente tabla de base de datos (es decir, algunos atributos del objeto comercial no están representados en la base de datos). Los atributos que no están representados en la base de datos carecen de información específica de la aplicación o bien se han definido con un valor por omisión o especifican procedimientos almacenados.
- El objeto comercial individual puede representar una vista que abarca varias tablas de base de datos. El conector puede utilizar un objeto comercial de esa clase al procesar sucesos Crear, Recuperar, Actualizar y Suprimir que se han desencadenado en la aplicación. Sin embargo, al procesar peticiones de objeto comercial, el conector puede utilizar un objeto comercial de esa clase solamente para peticiones de Recuperar.
- El objeto comercial individual puede representar un objeto envoltorio que se utiliza como contenedor para objetos comerciales que no guardan relación entre sí. El objeto envoltorio no está representado por una tabla ni una vista de base de datos. Los objetos envoltorio no se pueden utilizar como hijos de otros objetos.

Nota: Si un objeto comercial está basado en un procedimiento almacenado, cada atributo simple (salvo los atributos especiales de procedimiento almacenado) pueden tener o no tener información específica de la aplicación. Para obtener más información, consulte “Procedimientos almacenados” en la página 51.

Los objetos comerciales de WebSphere Business Integration Adapter pueden ser planos o jerárquicos. Todos los atributos de un objeto comercial **plano** son simples y representan un valor individual. El término objeto comercial **jerárquico** hace referencia a un objeto comercial completo, incluidos todos los objetos comerciales hijo que el objeto contiene a cualquier nivel. El término objeto comercial **individual** hace referencia a un sólo objeto comercial, con independencia de los objetos comerciales hijo que pueda contener o en los que esté contenido. El término objeto comercial de **nivel superior** hace referencia al objeto comercial individual situado en lo más alto de la jerarquía y que no tiene ningún objeto comercial padre.

Un objeto comercial jerárquico tiene atributos que representan un objeto comercial hijo, una matriz de objetos comerciales hijo o una combinación de ambos. A su vez, cada objeto comercial hijo puede contener un objeto comercial hijo o una matriz de objetos comerciales, y así sucesivamente. Existe una **relación de cardinalidad simple** cuando un atributo de un objeto comercial padre representa un objeto comercial hijo individual. En este caso, el atributo es del mismo tipo que el objeto comercial hijo.

Existe una **relación de cardinalidad múltiple** cuando un atributo de un objeto comercial padre representa una matriz de objetos comerciales hijo. En este caso, el atributo es una matriz del mismo tipo que los objetos comerciales hijo.

El conector es compatible con los tipos siguientes de relaciones entre objetos comerciales:

- “Relaciones de cardinalidad simple” en la página 39
- “Relaciones de cardinalidad simple y datos sin propietario” en la página 39
- “Relaciones de cardinalidad múltiple” en la página 40
- “Relaciones de cardinalidad simple que almacenan la relación en el hijo” en la página 41
- “Objetos envoltorio” en la página 42

En cada tipo de cardinalidad, la relación entre los objetos comerciales padre e hijo está descrita por la información específica de la aplicación del atributo de clave correspondiente al objeto comercial donde está contenida la relación. Para obtener más información sobre esta información específica de la aplicación, consulte "FK=[nombre_objeto_cf.]nombre_atributo_cf" en la página 65.

Relaciones de cardinalidad simple

Normalmente, un objeto comercial que contiene un objeto comercial hijo de cardinalidad simple tiene como mínimo dos atributos que representan la relación. El tipo de uno de los atributos es el mismo que el tipo del hijo. El otro atributo es un atributo simple que contiene la clave primaria del hijo como clave foránea del padre. El padre tiene tantos atributos de clave foránea como el hijo tiene atributos de clave primaria.

Debido a que las claves foráneas que establecen la relación están almacenadas en el padre, cada padre solamente puede contener un sólo hijo de cardinalidad simple de un tipo dado.

La Figura 2 muestra una relación típica de cardinalidad simple. En el ejemplo, fk1 es el atributo simple donde reside la clave primaria del hijo e hijo[1] es el atributo que representa al objeto comercial hijo.

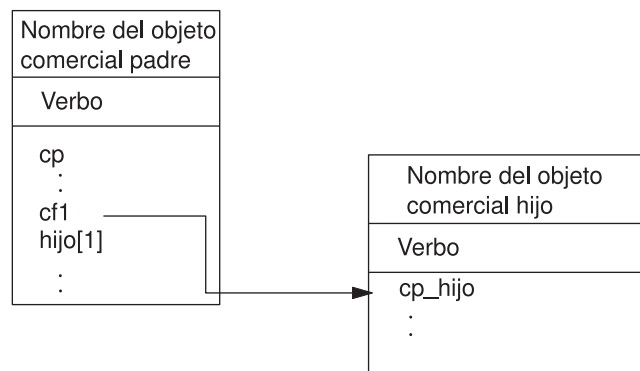


Figura 2. Relación típica de cardinalidad simple

Relaciones de cardinalidad simple y datos sin propietario

Normalmente, cada objeto comercial padre es el **propietario** de los datos del objeto comercial hijo contenido en el objeto padre. Por ejemplo, si cada objeto comercial Cliente contiene un objeto comercial individual denominado Dirección, cuando se crea un nuevo cliente, se añade una nueva fila tanto a la tabla de clientes como a la tabla de direcciones. La nueva dirección es exclusiva del nuevo cliente. Del mismo modo, cuando se suprime un cliente de la tabla de clientes, también se elimina la dirección del cliente en la tabla de direcciones.

Sin embargo, existen situaciones en las que varios objetos comerciales jerárquicos contienen los mismos datos, los cuales no pertenecen a ninguno de los objetos comerciales. Por ejemplo, suponga que el objeto comercial Dirección tiene el atributo EstadoProvincia[1] que representa la tabla de búsqueda EstadoProvincia con cardinalidad simple. Debido a que la tabla de búsqueda raramente se actualiza y su mantenimiento se realiza con independencia de los datos de la dirección, la creación o modificación de los datos de la dirección no afecta a los datos de la

tabla de búsqueda. El conector encuentra un nombre de estado o provincia existente o bien no logra encontrarlo. No añade ni modifica valores en la tabla de búsqueda.

Cuando varios objetos comerciales contienen el mismo objeto comercial hijo de cardinalidad simple, el atributo de clave foránea de cada objeto comercial padre debe especificar la relación como NO_OWNERSHIP (sin propietario). Cuando un intermediario de integración envía al conector un objeto comercial jerárquico con una petición Crear, Suprimir o Actualizar, el conector pasa por alto los hijos de cardinalidad simple contenidos sin propietario. Para estos objetos comerciales el conector sólo realiza acciones de recuperación. Si el conector no logra recuperar un objeto comercial de cardinalidad simple de esa clase, devuelve un error y detiene el proceso.

Para obtener información sobre cómo especificar la relación sin propietario, consulte “Atributos que representan un objeto comercial hijo de cardinalidad simple” en la página 71. Para obtener más información sobre la especificación de relaciones de clave foránea, consulte “Especificación de la clave foránea de un atributo” en la página 67.

Datos desnormalizados y datos sin propietario

Además de facilitar el uso de tablas de búsqueda estáticas, la inclusión sin propietario proporciona otra capacidad: la sincronización de datos normalizados y desnormalizados.

Sincronización desde datos normalizados a datos desnormalizados: Especificar una relación como NO_OWNERSHIP le permite crear o modificar datos cuando sincroniza desde una aplicación normalizada a otra desnormalizada. Por ejemplo, suponga que su aplicación normalizada de origen almacena datos en dos tablas, A y B. Suponga también que su aplicación desnormalizada de destino almacena todos los datos en una tabla individual de forma que cada entidad A almacena datos de B de forma redundante.

En este ejemplo, para sincronizar un cambio en los datos de la tabla B desde la aplicación de origen a la aplicación de destino, debe activar un suceso de la tabla A cada vez que cambien los datos de la tabla B. Además, debido a que los datos de la tabla B se almacenan de forma redundante en la tabla A, debe enviar un objeto comercial para cada fila de la tabla A que contenga los datos cambiados de la tabla B.

Sincronización desde datos desnormalizados a datos normalizados: Cuando sincroniza datos desde una aplicación desnormalizada de origen a una aplicación normalizada de destino, el conector no crea, suprime ni actualiza datos contenidos sin propietario en la aplicación normalizada.

Cuando sincroniza datos a una aplicación normalizada, el conector pasa por alto todos los objetos hijo de cardinalidad simple contenidos sin propietario. Para poder crear, eliminar o modificar tales datos hijo, el usuario debe procesar los datos manualmente.

Relaciones de cardinalidad múltiple

Normalmente, un objeto comercial que contenga una matriz de objetos comerciales hijo tiene un sólo atributo para representar la relación. El atributo es una matriz del mismo tipo que los objetos comerciales hijo. Para que un objeto padre contenga más de un objeto hijo, las claves foráneas que establecen la relación se almacenan en cada hijo.

Por tanto, cada hijo tiene como mínimo un atributo simple que contiene la clave primaria del padre como clave foránea. El hijo tiene tantos atributos de clave foránea como atributos de clave primaria tiene el padre.

Debido a que las claves foráneas que establecen la relación se almacenan en el hijo, cada padre puede tener de cero a varios hijos.

La Figura 3 muestra una relación de cardinalidad múltiple. En el ejemplo, IDpadre es el atributo simple donde reside la clave primaria del padre y hijo[n] es el atributo que representa la matriz de objetos comerciales hijo.

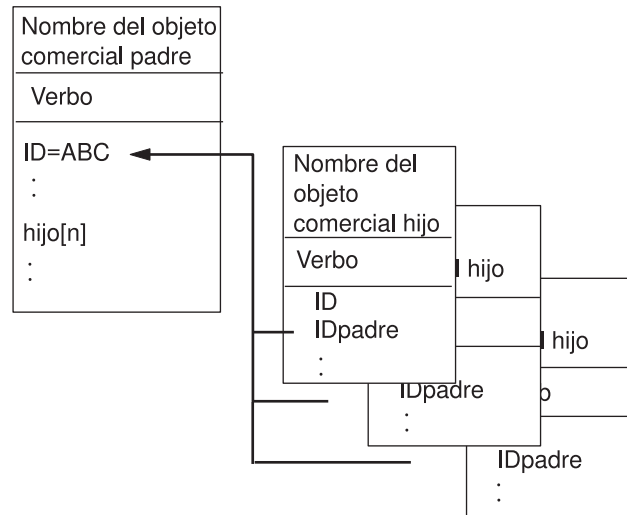


Figura 3. Relación de objetos comerciales de cardinalidad múltiple

Relaciones de cardinalidad simple que almacenan la relación en el hijo

Algunas aplicaciones almacenan una entidad hija individual de forma que la relación se almacena en el hijo en lugar de hacerlo en el padre. Es decir, el hijo contiene una clave foránea cuyo valor es idéntico al valor almacenado en la clave primaria del padre.

La Figura 4 muestra este tipo especial de relación de cardinalidad simple.

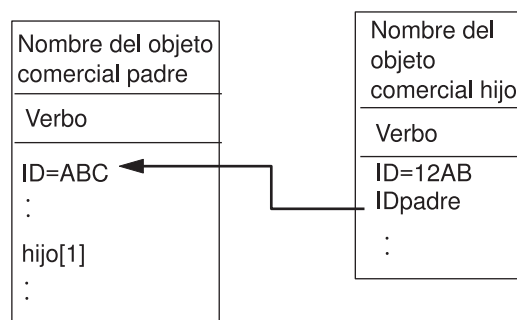


Figura 4. Objeto comercial de cardinalidad simple con la relación almacenada en el hijo

Las aplicaciones utilizan este tipo de relación de cardinalidad simple cuando los datos del hijo no existen con independencia del padre respectivo y sólo se puede acceder a ellos a través del padre. Tales datos del hijo no son nunca propiedad de

más de un padre, y necesitan que el padre y su valor de clave primaria existan para poder crear el hijo y su valor de clave foránea.

Para tener en cuenta las aplicaciones de esta clase, el conector también puede trabajar con objetos comerciales jerárquicos que contienen un hijo con cardinalidad simple, pero que almacenan la relación en el hijo en lugar de hacerlo en el padre.

Para especificar que un objeto comercial padre contiene un hijo de cardinalidad simple de esta forma especial, cuando especifique la información específica de la aplicación del atributo donde está contenido el hijo, no incluya el parámetro CONTAINMENT. Para obtener más información, consulte “Atributos que representan un objeto comercial hijo de cardinalidad simple” en la página 71.

Objetos envoltorio

El **objeto envoltorio** es un objeto comercial de nivel superior que no corresponde a ninguna tabla ni vista de base de datos. El objeto envoltorio se denota mediante la propiedad de objeto comercial de nivel superior WRAPPER, que tiene el valor true. El objeto envoltorio es un padre ficticio que se utiliza como contenedor para hijos que no guardan relación entre sí. Al procesar el objeto envoltorio, el conector pasa por alto el objeto comercial de nivel superior y sólo procesa los hijos. El objeto envoltorio puede contener entidades de cardinalidad N o de cardinalidad N-1 o ambas cosas.

Una entidad de cardinalidad N debe tener como mínimo un atributo exclusivo designado como clave primaria y al menos un atributo designado como clave foránea. Esta clave foránea se añadirá luego como clave primaria al objeto envoltorio. La clave foránea de la entidad hará referencia a la clave primaria del objeto envoltorio que se acaba de añadir.

En el caso de una entidad de cardinalidad N-1, la clave primaria debe estar designada como clave primaria y clave foránea, y hacer referencia a la clave primaria del envoltorio, que es la misma que la clave primaria contenida en la entidad N-1.

Procesos de verbos de objeto comercial

Esta sección describe los aspectos siguientes del proceso de los verbos de un objeto comercial:

- “Determinación del verbo” en la página 43, que describe cómo el conector determina el verbo que se debe utilizar para cada objeto comercial de origen.
- “Imágenes posteriores y deltas” en la página 43, que define los términos y describe cómo el conector trabaja con imágenes posteriores
- “Proceso de verbos” en la página 44, que describe los pasos que el conector emprende al crear, recuperar, actualizar o suprimir un objeto comercial
- “Sentencias de SQL” en la página 51, que describe cómo el conector utiliza sentencias simples de SQL para operaciones de selección, actualización, recuperación o supresión
- “Procedimientos almacenados” en la página 51, que describe cómo el conector utiliza procedimientos almacenados
- “Confirmación y retrotracción de transacciones” en la página 59, que describe brevemente cómo el conector utiliza bloques de transacciones

Determinación del verbo

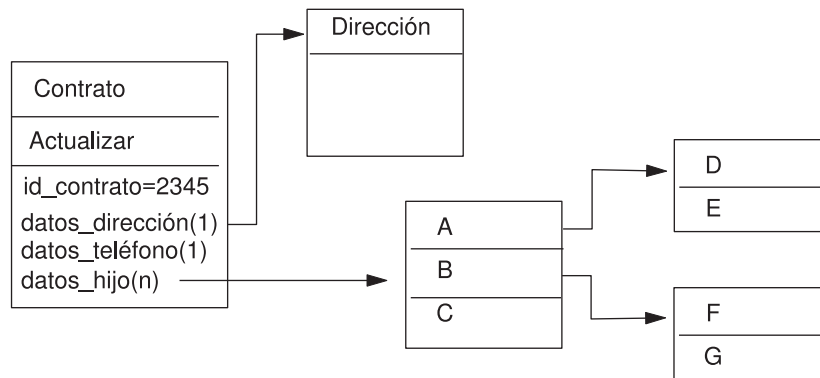
Un objeto comercial de nivel superior y cada uno de sus objetos comerciales hijo puede contener sus propios verbos. Por tanto, un intermediario de integración puede pasar al conector un objeto comercial que tiene verbos diferentes para los objetos comerciales padre y los objetos comerciales hijo. Cuando se da esta condición, el conector utiliza el verbo del objeto comercial padre de nivel superior para determinar cómo procesar el objeto comercial completo. Para obtener más información, consulte “Proceso de verbos” en la página 44.

Imágenes posteriores y deltas

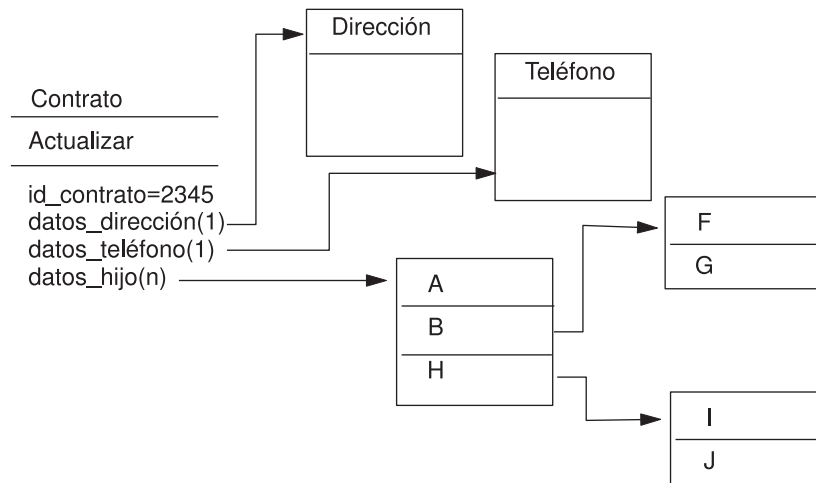
Una **imagen posterior** es el estado de un objeto comercial después de realizar todos los cambios en el objeto. Un **delta** es un objeto comercial utilizado en una operación de actualización que sólo contiene valores de clave y los datos que se deben cambiar. Puesto que el conector sólo puede trabajar con imágenes posteriores, cuando recibe un objeto comercial para actualizar, el conector supone que el objeto comercial representa el estado deseado de los datos tras la actualización.

Por tanto, cuando un intermediario de integración envía al conector un objeto comercial con el verbo Actualizar, el conector cambia la representación actual del objeto comercial en la base de datos para que coincida exactamente con el objeto comercial de origen. Para hacer esto, el conector cambia valores de atributos simples y añade o elimina objetos comerciales hijo.

Por ejemplo, suponga que el estado actual de Contrato 2345 en la base de datos es el siguiente:



Suponga también que el intermediario de integración pasa el objeto comercial siguiente al conector:



Para procesar la actualización, el conector aplica los cambios siguientes a la base de datos:

- Actualiza los atributos simples en los objetos comerciales de nivel superior Contrato y Dirección
- Crea el objeto comercial Teléfono
- Actualiza los atributos simples en los objetos comerciales hijo A, B, F y G
- Suprime los objetos comerciales hijo C, D y E
- Crea los objetos comerciales hijo H, I y J

Debido a que el conector supone que cada objeto comercial que recibe del intermediario de integración representa una imagen posterior, es importante asegurarse de que cada objeto comercial enviado para actualizar a ese conector contenga objetos comerciales hijo existentes válidos. Aunque no haya cambiado ninguno de los atributos simples de un objeto comercial hijo, el objeto comercial hijo debe estar incluido en el objeto comercial de origen.

Sin embargo, existe una forma de impedir que algunos conectores supriman objetos comerciales hijo faltantes durante una operación de actualización. Puede utilizar la información específica de la aplicación correspondiente al atributo representativo del hijo o matriz de hijos para indicar al conector que conserve los objetos comerciales hijo que no estén incluidos en el objeto comercial de origen. Para hacer esto, establezca `KEEP_RELATIONSHIP` en `true`. Para obtener más información, consulte “Especificación de la clave foránea de un atributo” en la página 67.

Proceso de verbos

Esta sección describe los pasos que el conector emprende al crear, recuperar, actualizar o suprimir un objeto comercial que ha recibido de un intermediario de integración. El conector procesa los objetos comerciales jerárquicos de forma recursiva, es decir, ejecuta los mismos pasos para cada objeto comercial hijo hasta que ha procesado todos los objetos comerciales individuales.

Nota: Un objeto comercial de nivel superior que sea un objeto envoltorio admite la utilización de los verbos de crear, recuperar, actualizar y suprimir. La única

diferencia en el proceso de un objeto envoltorio es que el objeto envoltorio no se procesa, sólo se procesan los objetos que contiene.

Comparación de objetos comerciales

En diversos momentos del proceso descrito más abajo, el conector compara dos objetos comerciales para ver si son idénticos. Por ejemplo, durante una operación de actualización, el conector determina si un objeto comercial determinado existe en una matriz de objetos comerciales. Para realizar la comprobación, el conector compara el objeto comercial con cada objeto comercial contenido en la matriz. Para que dos objetos comerciales sean idénticos, se deben cumplir estas dos condiciones:

- El tipo de los objetos comerciales sujetos a comparación debe ser el mismo. Por ejemplo, un objeto comercial Cliente no se considera nunca que es idéntico a un objeto comercial Contacto aunque todos sus atributos sean exactamente los mismos.
- Todos los atributos de clave correspondientes de los dos objetos comerciales deben contener valores idénticos. Si un atributo de clave tiene el valor CxIgnore en ambos objetos comerciales, el conector considera que los objetos son idénticos. En cambio, si un atributo de clave tiene el valor CxIgnore en uno de los objetos comerciales, pero no en el otro, los objetos comerciales no son idénticos.

Operaciones de creación

Cuando crea un objeto comercial, el conector devuelve el código de estado VALCHANGE si la operación se ha realizado satisfactoriamente (con independencia de si la operación ha causado o no cambios en el objeto comercial), o devuelve FAIL si la operación ha fallado.

El conector ejecuta los pasos siguientes al crear un objeto comercial jerárquico:

1. Inserta recursivamente cada objeto comercial hijo de cardinalidad simple contenido con propietario en la base de datos. Es decir, el conector crea el objeto hijo y todos los objetos hijo contenidos en el hijo y en sus hijos.
Si la definición del objeto comercial especifica que un atributo representa un objeto comercial hijo con cardinalidad simple y que el atributo está vacío, el conector pasa por alto el atributo. Sin embargo, si la definición del objeto comercial necesita que un atributo represente un hijo y no lo hace, el conector devuelve un error y detiene el proceso.
2. Procesa cada objeto comercial hijo de cardinalidad simple contenido sin propietario, de la manera siguiente:
 - a. Intenta recursivamente recuperar el hijo de la base de datos utilizando los valores de clave pasados por el intermediario de integración.
 - b. Si la recuperación es infructuosa, lo que indica que el hijo no existe actualmente en la base de datos, el conector devuelve un error y detiene el proceso. Si la recuperación se realiza satisfactoriamente, el conector actualiza recursivamente el objeto comercial hijo.

Nota: Para que esta manera de proceder sea efectiva cuando el objeto comercial hijo ya existe en la base de datos de la aplicación, asegúrese de que las referencias cruzadas de los atributos de clave primaria de los objetos comerciales hijo sean correctas en las operaciones de creación. Si el objeto comercial hijo no existe ya en la base de datos de la aplicación, establezca los atributos de clave primaria en CxBlank.

3. Inserta el objeto comercial de nivel superior en la base de datos, de esta manera:

- a. Establece los valores de clave foránea del objeto de nivel superior en los valores de clave primaria del correspondiente objeto comercial hijo representado con cardinalidad simple. Debido a que los valores contenidos en los objetos comerciales hijo pueden ser establecidos por secuencias o contadores de base de datos o por la propia base de datos durante la creación del hijo, este paso asegura que los valores de clave foránea del padre sean correctos antes de que el conector inserte el padre en la base de datos.
- b. Crea un nuevo valor de ID exclusivo para cada atributo definido automáticamente por la base de datos. El nombre de la secuencia o contador de base de datos se guarda en la información específica de la aplicación correspondiente al atributo. Si un atributo tiene una secuencia o contador de base de datos asociado, el valor creado por el conector sobrescribe cualquier valor pasado por el intermediario de integración. Para obtener más información sobre la especificación de una secuencia o contador de base de datos, consulte UID=AUTO en “Información específica de la aplicación para atributos simples” en la página 65.
- c. Copia el valor de un atributo en el valor de otro atributo tal como especifica el parámetro CA (CopyAttribute) de la información específica de la aplicación correspondiente al atributo. Para obtener más información sobre la utilización del parámetro CA, consulte CA=set_attr_name en “Información específica de la aplicación para atributos simples” en la página 65.
- d. Inserta el objeto comercial de nivel superior en la base de datos.

Nota: Si el objeto comercial de nivel superior es un objeto envoltorio, no se insertará en la base de datos.

4. Procesa cada uno de los objetos comerciales hijo de cardinalidad simple del objeto de nivel superior en los que la relación padre/hijo se guarda en el hijo, de la manera siguiente:
 - a. Establece los valores de clave foránea del hijo para hacer referencia al valor de los correspondientes atributos de clave primaria contenidos en el padre. Debido a que los valores de clave primaria del padre se pueden haber generado durante la creación del padre, esto asegura que sean correctos los valores de clave foránea de cada hijo antes de que el conector inserte el hijo en la base de datos.
 - b. Inserta el hijo en la base de datos.
5. Procesa cada uno de los objetos comerciales hijo de cardinalidad múltiple contenidos en el objeto de nivel superior, de esta manera:
 - a. Establece los valores de clave foránea de cada hijo para hacer referencia al valor de los correspondientes atributos de clave primaria contenidos en el padre. Debido a que los valores de clave primaria del padre se pueden haber generado durante la creación del padre, esto asegura que sean correctos los valores de clave foránea de cada hijo antes de que el conector inserte el hijo en la base de datos.
 - b. Inserta cada objeto comercial hijo de cardinalidad múltiple en la base de datos.

Operaciones Retrieve (recuperación)

El conector ejecuta los pasos siguientes al recuperar objeto comercial jerárquico:

1. Elimina todos los objetos comerciales hijo del objeto comercial de nivel superior que ha recibido del intermediario de integración.
2. Recupera el objeto comercial de nivel superior de la base de datos.
 - Si la recuperación devuelve una sola fila, el conector continúa el proceso.

- Si la recuperación no devuelve ninguna fila, lo que indica que el objeto comercial de nivel superior no existe en la base de datos, el conector devuelve BO_DOES_NOT_EXIST.
- Si la recuperación devuelve más de una fila, el conector devuelve FAIL.

Notas:

- a. Un objeto comercial puede tener atributos que no corresponden a ninguna columna de base de datos, como los atributos marcadores de posición. Durante la recuperación, el conector no cambia tales atributos en el objeto comercial de nivel superior; estos atributos permanecen establecidos en los valores que han recibido del intermediario de integración. En los objetos comerciales hijo, el conector establece tales atributos en sus valores por omisión durante la recuperación.
 - b. Un objeto comercial de nivel superior que sea un objeto envoltorio debe contener todos los valores de atributo de los objetos situados en el nivel inmediatamente por debajo del objeto envoltorio, los cuales serían necesarios para recuperar los objetos, incluidos los atributos de clave y los atributos marcadores de posición. El objeto envoltorio debe tener definidos todos los atributos de clave y atributos marcadores de posición. Los atributos simples del objeto envoltorio que se utilizarán como claves foráneas en los objetos situados en el nivel inmediatamente por debajo del envoltorio deben estar marcados como claves en el objeto envoltorio.
3. El conector recupera recursivamente todos los objetos comerciales hijo de cardinalidad múltiple.

Nota: El conector no exige la unicidad de los valores al llenar con datos una matriz de objetos comerciales. Corresponde a la base de datos garantizar la unicidad de los valores. Si la base de datos devuelve objetos comerciales hijo duplicados, el conector devuelve hijos duplicados.

4. El conector recupera recursivamente cada hijo de cardinalidad simple, con independencia de si el objeto comercial hijo está contenido con o sin propietario.

Nota: Todos los objetos comerciales hijo de cardinalidad simple se procesan de acuerdo con su aparición en el objeto comercial y antes de que se procese el objeto comercial padre. La posesión y no-posesión del objeto hijo no determinan la secuencia de proceso, pero sí el tipo de proceso.

Operaciones RetrieveByContent

El verbo RetrieveByContent sólo es aplicable al objeto comercial de nivel superior, pues el conector realiza operaciones de recuperación basándose en atributos que sólo existen en el objeto comercial de nivel superior.

Si un objeto comercial de nivel superior utiliza el verbo RetrieveByContent, se utilizan como criterios de recuperación todos los atributos no nulos (incluidos los atributos que no son de clave).

Si se devuelve más de una fila, el conector utiliza la primera fila como fila resultante y devuelve MULTIPLE_HITS.

Nota: El verbo RetrieveByContent no es aplicable a un objeto comercial de nivel superior que sea un objeto envoltorio.

Operaciones Update (actualización)

Cuando actualiza un objeto comercial, el conector devuelve el código de estado VALCHANGE si la operación se ha realizado satisfactoriamente (con independencia de si la operación ha causado o no cambios en el objeto comercial), o devuelve FAIL si la operación ha fallado. Cuando trabaja con una base de datos Oracle, el conector bloquea los datos mientras los recupera para asegurar su integridad.

El conector ejecuta los pasos siguientes al actualizar un objeto comercial jerárquico:

1. Utiliza los valores de clave primaria del objeto comercial de origen para recuperar la entidad correspondiente de la base de datos. El objeto comercial recuperado es una representación exacta del estado actual de los datos en la base de datos.

- Si la recuperación falla, lo que indica que el objeto comercial de nivel superior no existe en la base de datos, el conector devuelve BO_DOES_NOT_EXIST y la actualización falla.

Nota: Si un objeto comercial de nivel superior es un objeto envoltorio, no es necesario que exista en la base de datos. Sin embargo, el objeto comercial de nivel superior debe contener todos los valores de atributo de los objetos situados en el nivel inmediatamente por debajo del objeto envoltorio, los cuales serían necesarios para recuperar los objetos, incluidos los atributos de clave y atributos marcadores de posición. El objeto envoltorio debe tener definidos todos los atributos de clave y atributos marcadores de posición. Los atributos simples del objeto envoltorio que se utilizarán como claves foráneas en los objetos situados en el nivel inmediatamente por debajo del envoltorio deben estar marcados como claves en el objeto envoltorio.

- Si la recuperación tiene éxito, el conector compara el objeto comercial recuperado con el objeto comercial de origen para determinar qué objetos comerciales hijo necesitan cambios en la base de datos. Sin embargo, el conector no compara los valores de atributos simples del objeto comercial de origen con los contenidos en el objeto comercial recuperado. El conector actualiza el valor de todos los atributos simples que no son de clave. Si todos los atributos simples del objeto comercial de nivel superior representan claves, el conector no puede generar una consulta de actualización para el objeto comercial de nivel superior. En este caso, el conector registra un aviso y continúa en el paso 2.

2. El conector actualiza recursivamente todos los hijos de cardinalidad simple del objeto comercial de nivel superior.

Si la definición de objeto comercial exige que un atributo represente un objeto comercial hijo, el hijo debe existir tanto en el objeto comercial de origen como en el objeto comercial recuperado. De lo contrario, la actualización falla y el conector devuelve un error.

El conector maneja los hijos de cardinalidad simple contenidos con propietario de una de las maneras siguientes:

- Si el hijo existe tanto en el objeto comercial de origen como en el objeto recuperado, en lugar de actualizar el hijo ya existente en la base de datos, el conector suprime el hijo existente y crea un nuevo hijo.
- Si el hijo existe en el objeto comercial de origen, pero no en el objeto comercial recuperado, el conector lo crea recursivamente en la base de datos.
- Si el hijo existe en el objeto comercial recuperado, pero no en el objeto comercial de origen, el conector lo suprime recursivamente de la base de

datos. El tipo de supresión, física o lógica, depende del valor de la propiedad `ChildUpdatePhyDelete` de la operación de supresión.

Para los hijos de cardinalidad simple contenidos sin propietario, el conector intenta recuperar cada hijo de la base de datos que esté presente en el objeto comercial de origen. Si recupera el hijo con éxito, el conector llena con datos el objeto comercial hijo, pero no lo actualiza, pues el conector nunca modifica los hijos de cardinalidad simple contenidos sin propietario.

3. Para los objetos comerciales hijo de cardinalidad simple que guardan la relación padre/hijo en el padre, el conector establece cada valor de clave foránea del padre en el valor de la clave primaria existente en el correspondiente objeto comercial hijo de cardinalidad simple. Este paso es necesario porque los hijos de cardinalidad simple se pueden haber añadido a la base de datos en pasos anteriores, dando lugar a la creación de nuevos ID exclusivos.
4. El conector actualiza todos los atributos simples del objeto comercial recuperado, excepto aquéllos cuyo atributo correspondiente en el objeto comercial de origen contiene el valor `CxIgnore`.

Debido a que el objeto comercial que se está actualizando debe ser exclusivo, el conector verifica que como resultado se procese una sola fila. El conector devuelve un error si se devuelve más de una fila.

5. El conector asigna el valor de clave primaria del correspondiente objeto comercial padre a todos los valores de clave foránea de cada hijo que guarda la relación padre/hijo en el hijo (tanto de cardinalidad simple como de cardinalidad múltiple). (Cuando se utiliza `InterChange Server` como intermediario de integración, normalmente se han establecido referencias cruzadas entre estos valores durante la correlación de datos). Sin embargo, este paso es importante para asegurarse de que sean correctos los valores de clave foránea de los nuevos hijos que guardan la relación padre/hijo en el hijo, antes de que el conector actualice esos hijos.
6. El conector procesa cada hijo de cardinalidad múltiple del objeto comercial recuperado de una de las maneras siguientes:
 - Si el hijo existe tanto en las matrices de los objetos comerciales de origen como en los objetos comerciales recuperados, el conector actualiza recursivamente el hijo en la base de datos.
 - Si el hijo existe en la matriz de origen, pero no en la matriz del objeto comercial recuperado, el conector crea recursivamente el hijo en la base de datos.
 - Si el hijo existe en la matriz del objeto comercial recuperado, pero no en la matriz de origen, el conector suprime recursivamente el hijo de la base de datos, a menos que la información específica de aplicación correspondiente al atributo que representa al hijo en el padre tenga el parámetro `KEEP_RELATIONSHIP` establecido en `true`. En este caso, el conector no suprime el hijo de la base de datos. Para obtener más información, consulte “Especificación de la clave foránea de un atributo” en la página 67. El tipo de supresión, física o lógica, depende del valor de la propiedad `ChildUpdatePhyDelete` de la operación de supresión.

Nota: El intermediario de integración debe asegurarse de que los objetos comerciales contenidos con cardinalidad múltiple en el objeto comercial de origen sean exclusivos (es decir, que una matriz no contenga dos o más ejemplares del mismo objeto comercial). Si el conector recibe duplicados de un objeto comercial en una matriz de origen, procesará el objeto comercial dos veces, lo cual puede producir resultados imprevisibles.

Operaciones DeltaUpdate

El proceso del verbo DeltaUpdate es diferente al proceso del verbo Update en los aspectos siguientes:

1. En un DeltaUpdate no se realiza ninguna recuperación antes de la actualización, a diferencia de lo que ocurre en el proceso del verbo Update.
2. No se realizan comparaciones entre el objeto comercial de entrada y el objeto comercial existente en la base de datos.
3. Todos los hijos se procesan de acuerdo con el verbo definido en cada objeto hijo. Si un hijo no tiene un verbo definido en él, el conector devolverá un error.

Cuando realiza una actualización delta de un objeto comercial, el conector devuelve el código de estado VALCHANGE si la operación se ha realizado satisfactoriamente (con independencia de si la operación ha causado o no cambios en el objeto comercial), o devuelve FAIL si la operación ha fallado.

El conector ejecuta los pasos siguientes al efectuar una actualización delta de un objeto comercial jerárquico:

1. Procesa recursivamente todos los hijos de cardinalidad simple del objeto padre. Si un hijo está marcado como IsRequired en la especificación del objeto comercial, el hijo debe estar presente en el objeto de entrada. De lo contrario, la actualización delta fallará y el conector devolverá un error.
2. El conector establece todos los valores de clave foránea del padre que hacen referencia a atributos de hijos de cardinalidad simple en los valores correspondientes del objeto hijo. Esto es necesario porque los hijos de cardinalidad simple se pueden haber añadido a la base de datos en pasos anteriores, dando lugar a la creación de nuevos valores de secuencia.
3. El conector actualiza mediante una sentencia UPDATE de SQL o un procedimiento almacenado el objeto actual que se está procesando. Se actualizan todos los atributos simples del objeto comercial individual, excepto aquellos atributos que tienen asignado el valor IsIgnore en el objeto comercial de entrada. El conector no compara a nivel de atributo el objeto de entrada con el objeto actual para determinar qué atributos es necesario añadir a la sentencia de actualización; se actualizan todos los atributos. Debido a que el objeto que se está actualizando debe ser exclusivo, el conector verifica que solamente se procese una sola fila como resultado. Se devuelve un error si se procesa más de una fila.
4. El conector asigna los correspondientes valores del objeto padre a todos los valores de clave foránea en todos los hijos de cardinalidad N del objeto actual que hacen referencia a atributos del padre. Normalmente ya se habrán establecido referencias cruzadas entre estos valores durante la correlación de datos; sin embargo, esto puede no ser así para hijos nuevos existentes en contenedores de cardinalidad N. Este paso garantiza que los valores de clave foránea sean correctos en todos los hijos de cardinalidad N antes de que se actualicen esos hijos.
5. El conector actualiza todos los contenedores de cardinalidad N del objeto actual.

Cuando se procesan los objetos hijo, se obtiene el verbo de cada hijo y se realiza la operación apropiada. Los verbos permitidos en una operación DeltaUpdate para un hijo son Create, Delete y DeltaUpdate.

- Si se encuentra un verbo Create en el hijo, se crea el hijo en la base de datos si es un hijo de un propietario. Los hijos sin propietario se recuperan para comprobar su existencia en la base de datos.

- Si se encuentra un verbo Delete en el hijo, se suprime ese hijo.
- Si se encuentra un verbo DeltaUpdate en el hijo, se actualiza el hijo en la base de datos.

Operaciones Delete (supresión)

Cuando suprime un objeto comercial, el conector devuelve el estado SUCCESS si la operación ha sido satisfactoria, o FAIL si la operación ha fallado. Primero se recupera el objeto comercial padre y luego el adaptador suprime recursivamente todos los hijos de cardinalidad simple que sean propiedad del padre; a continuación suprime el propio padre y finalmente todos los hijos de cardinalidad N. Los hijos sin propietario, de cardinalidad simple, no se suprimen nunca. Si el objeto comercial no existe, el conector devuelve el estado FAIL.

El conector puede realizar supresiones lógicas y físicas, dependiendo del valor del Nombre de la columna de estado (valor SCN) existente en la información específica de la aplicación correspondiente al objeto. Si el valor SCN está definido, el conector realiza una supresión lógica. Si el valor SCN no está definido, el conector realiza una supresión física.

Supresiones físicas: El conector ejecuta los pasos siguientes cuando realiza la supresión física de un objeto comercial jerárquico:

1. Suprime recursivamente todos los objetos comerciales hijo de cardinalidad simple contenidos con propietario.
2. Suprime el objeto comercial de nivel superior.
3. Suprime recursivamente todos los objetos comerciales hijo de cardinalidad múltiple.

Nota: Un objeto comercial de nivel superior que sea un objeto envoltorio carece de la tabla de base de datos correspondiente, por lo que no se suprime de la base de datos. Los valores de atributos simples de un objeto envoltorio no se tendrán en cuenta.

Supresiones lógicas: El conector ejecuta los pasos siguientes cuando realiza la supresión lógica de un objeto comercial:

1. Emite una sentencia UPDATE que asigna al atributo de estado del objeto comercial el valor especificado por la información específica de la aplicación correspondiente al objeto comercial. El conector verifica que como resultado se actualice una sola fila de la base de datos y devuelve un error si se actualiza más de una fila.
2. Realiza una supresión lógica recursiva de todos los hijos de cardinalidad simple contenidos con propietario y de todos los hijos de cardinalidad múltiple. El conector no suprime los hijos de cardinalidad simple contenidos sin propietario.

Sentencias de SQL

El conector puede utilizar sentencias simples de SQL para operaciones de selección, actualización, recuperación o supresión. Los nombres de columna utilizados para las sentencias de SQL se obtienen de la propiedad AppSpecificInfo de un atributo. Cada consulta abarca una sola tabla, a menos que se emita para una vista.

Procedimientos almacenados

Un procedimiento almacenado es un grupo de sentencias de SQL que forman una unidad lógica y ejecutan una tarea determinada. Un procedimiento almacenado

abarca un conjunto de operaciones o consultas para que el conector las ejecute sobre un objeto contenido en un servidor de bases de datos.

El conector invoca un procedimiento almacenado en las circunstancias siguientes:

- Antes de procesar un objeto comercial, para realizar procesos operacionales preparatorios
- Después de procesar un objeto comercial, para realizar procesos post-operacionales
- Para ejecutar un conjunto de operaciones sobre un objeto comercial, en lugar de utilizar una sentencia simple INSERT, RETRIEVE, UPDATE o DELETE

Cuando procesa un objeto comercial jerárquico, el conector puede utilizar un procedimiento almacenado para procesar el objeto comercial de nivel superior o cualquiera de sus objetos comerciales hijo. Pero cada objeto comercial o matriz de objetos comerciales debe tener su propio procedimiento almacenado.

Especificación de un procedimiento almacenado

Esta sección describe los pasos que debe seguir para hacer que el conector utilice un procedimiento almacenado para un objeto comercial. Se tratan los temas siguientes:

- “Adición de atributos al objeto comercial”
- “Sintaxis de un procedimiento almacenado” en la página 53
- “Ejemplos de procedimientos almacenados que no devuelven conjuntos de resultados” en la página 54
- “Especificación del procedimiento almacenado” en la página 55

Adición de atributos al objeto comercial: Debe añadir una clase especial de atributo al objeto comercial para cada tipo de procedimiento almacenado que procese el conector. Estos atributos representan sólo el tipo de procedimiento almacenado y la información específica de la aplicación que lo define. Estos atributos no utilizan los parámetros de información específica de la aplicación que se pueden utilizar para un atributo simple estándar.

Designa el atributo de acuerdo con el tipo de procedimiento almacenado que se debe utilizar. Por ejemplo, para hacer que el conector utilice los procedimientos almacenados AfterUpdate y BeforeRetrieve, añade los atributos AfterUpdateSP y BeforeRetrieveSP.

El conector reconoce los nombres siguientes de atributos de objeto comercial:

```
BeforeCreateSP
AfterCreateSP
CreateSP
BeforeUpdateSP
AfterUpdateSP
UpdateSP
BeforeDeleteSP
AfterDeleteSP
DeleteSP
BeforeRetrieveSP
AfterRetrieveSP
RetrieveSP
BeforeRetrieveByContentSP
AfterRetrieveByContentSP
RetrieveByContentSP
BeforeRetrieveUpdateSP
AfterRetrieveUpdateSP
```


RetrieveUpdateSP
BeforeDeltaUpdateSP
AfterDeltaUpdateSP
DeltaUpdateSP

Nota: Cree un atributo sólo para aquellos procedimientos almacenados que desee que sean ejecutados por el conector. Utilice la información específica de la aplicación o la correlación de datos (sólo si se utiliza InterChange Server como intermediario de integración) para especificar valores para estos atributos antes de que el objeto comercial se envíe al conector. Es necesario reiniciar el conector para que reconozca los cambios hechos en estos valores para las llamadas subsiguientes realizadas para un objeto comercial.

Sintaxis de un procedimiento almacenado: La sintaxis para especificar un procedimiento almacenado es la siguiente:

```
SPN=nombre_procedimiento_almacenado;RS=true|false[;IP=nombre_atributo1  
[:nombre_atributo2 [...]]] [;OP=nombre_atributo1 | RS[:nombre_atributo2 | RS [...]]]  
[;IO=nombre_atributo1[:nombre_atributo2[:...]]]
```

donde:

nombre_procedimiento_almacenado

Es el nombre del procedimiento almacenado.

RS

Es true si el procedimiento almacenado devuelve un conjunto de resultados, o false en caso contrario. El valor por omisión es false. Si el valor es true, la propiedad ColumnName contenida en la información específica de la aplicación de un atributo apunta a la columna apropiada del conjunto de resultados. Si RS forma parte de la lista de parámetros de salida, ese parámetro determinado devuelve un conjunto de resultados. Solamente se permite un sólo parámetro OUT como conjunto de resultados. Si se devuelve más de un conjunto de resultados como parámetro OUT, sólo se devuelve el primer conjunto de resultados y los demás se pasan por alto. Actualmente esta característica está disponible para Oracle 8i y versiones posteriores, para procedimientos almacenados que hacen uso del Controlador JDBC de Oracle. Para el procedimiento almacenado contenido en la base de datos, el parámetro correspondiente debe devolver un tipo REF_CURSOR.

IP

Parámetros de entrada: es la lista de atributos de objeto comercial cuyos valores el conector debe utilizar como valores de entrada al ejecutar el procedimiento almacenado.

OP

Parámetros de salida: es la lista de atributos de objeto comercial a los cuales el conector debe devolver valores tras ejecutar el procedimiento almacenado. Consulte RS para obtener una descripción del conjunto de resultados.

IO

Parámetros de entrada/salida: es la lista de atributos de objeto comercial cuyos valores el conector debe utilizar como valores de entrada y a

los cuales el conector debe devolver valores tras ejecutar el procedimiento almacenado.

El orden de los parámetros nombre_procedimiento_almacenado y RS es importante, no así el orden de los parámetros entre ellos. Es decir, no es importante para el conector si el procedimiento almacenado agrupa todos los parámetros de cada tipo o si mezcla los tipos de parámetros. Cuando varios parámetros del mismo tipo estén agrupados juntos, separe los valores utilizando un signo de dos puntos como delimitador; no es necesario que repita el tipo de parámetro para cada valor. Separe los parámetros de tipos diferentes utilizando un signo de punto y coma como delimitador. Cuando especifique valores de parámetros, no coloque un espacio en blanco en ninguno de los dos lados del signo de igualdad (=).

Ejemplos de procedimientos almacenados que devuelven conjuntos de resultados: Los ejemplos siguientes hacen uso de los procedimientos almacenados denominados CustomerAddressRetrieve y CustomerAddressRetrieveForOracleDB para devolver un conjunto de resultados que contiene varias direcciones y se utiliza para crear un objeto comercial hijo de cardinalidad N.

Nota: Los conjuntos de resultados se procesan para el atributo RetrieveSP solamente y se utilizan para crear un objeto comercial hijo de cardinalidad N.

Para las bases de datos Oracle, el conjunto de resultados se devuelve como parámetro de salida y es procesado por el adaptador según convenga. Para otras bases de datos, el conjunto de resultados es un valor de retorno devuelto por el procedimiento almacenado.

- CustomerAddressRetrieve (para bases de datos que no sean Oracle)
Attribute : RetrieveSP
ASI : SPN=CustomerAddressRetrieve;RS=true;IP=CustomerName:IP=CustomerId;
OP=ErrorStatus;OP=ErrorMsg
- CustomerAddressRetrieveForOracleDB (para bases de datos Oracle)
Attribute : RetrieveSP
ASI : SPN=CustomerAddressRetrieveForOracleDB;RS=true;IP=CustomerName:
IP=CustomerId;OP=RS;OP=ErrorStatus;OP=ErrorMsg

(OP=RS significa que el primer parámetro de salida es un conjunto de resultados)

Ejemplos de procedimientos almacenados que no devuelven conjuntos de resultados: Los ejemplos siguientes hacen uso de los procedimientos almacenados denominados CustomerInsert y VendorInsert que obtienen valores a partir de dos atributos de entrada, y devuelven valores para cuatro atributos de salida. Los ejemplos muestran diferentes estructuras para procedimientos almacenados.

- Los parámetros del mismo tipo están agrupados juntos (IP, IP, OP, OP, OP, OP, IO):
SPN=CustomerInsert;RS=false;IP=LastName:FirstName;OP=CustomerName:
CustomerID:ErrorStatus:ErrorMessage;IO=VendorID
- Los parámetros del mismo tipo están mezclados (IP, OP, OP, OP, IP, IO, OP):
SPN=VendorInsert;RS=false;IP=LastName;OP=CustomerName:
CustomerID:ErrorStatus;IP=FirstName;IO=VendorID;OP=ErrorMessage

El conector admite solamente los tipos de datos simples que son compatibles con el controlador JDBC.

Especificación del procedimiento almacenado: Existen dos formas de especificar el nombre del procedimiento almacenado y los valores de sus parámetros:

- Propiedad AppSpecificInfo del atributo

Si la longitud del texto por el que se especifica el procedimiento almacenado es menor o igual que 4000 bytes, puede especificar el valor en la propiedad AppSpecificInfo del atributo. Puede utilizar esta propiedad para especificar el procedimiento almacenado con independencia de si el conector ha hecho sondeos para el objeto comercial (es decir, el objeto comercial representa un suceso de aplicación) o si ha recibido el objeto comercial como petición de un intermediario de integración.

El ejemplo siguiente muestra la especificación del procedimiento almacenado en la información específica de la aplicación. En este caso, el valor especificado para la propiedad MaxLength no es importante para el procedimiento almacenado.

```
[Attribute]
Name = BeforeCreateSP
Type = String
MaxLength = 15
IsKey = false
IsRequired = false
AppSpecificInfo = SPN=ContactInsert;IP=LastName:FirstName;OP=CustomerName:
CustomerID:ErrorStatus:ErrorMessage
```

[End]

- Valor del atributo (sólo aplicable si se utiliza InterChange Server como intermediario de integración)

Si la longitud del texto por el que se especifica el procedimiento almacenado tiene más de 4000 bytes, debe utilizar la correlación de valores para especificar el procedimiento almacenado. Puede utilizar la correlación de valores para especificar el procedimiento almacenado sólo si el objeto comercial representa una petición de intermediario de integración. Es decir, no puede utilizar el valor de un atributo para especificar un procedimiento almacenado cuando el conector está sondeando la presencia de sucesos.

Si el texto del procedimiento almacenado tiene más de 4000 bytes y utiliza la correlación de valores para especificar el procedimiento almacenado, debe aumentar el valor de la propiedad MaxLength para dar cabida al texto completo.

Nota: Si se ejecuta un procedimiento almacenado para una operación de creación, actualización o supresión sobre un objeto comercial jerárquico que contiene una matriz de objetos comerciales hijo, el conector procesa cada objeto comercial hijo individualmente. Por ejemplo, si el conector ejecuta un procedimiento almacenado BeforeCreate, no procesa la matriz como una unidad, sino que procesa cada miembro de la matriz. Cuando el conector procesa un procedimiento almacenado BeforeRetrieve, el conector actúa sobre un objeto comercial individual. Cuando el conector procesa un procedimiento almacenado AfterRetrieve, el conector actúa sobre todos los objetos comerciales devueltos por la operación de recuperación.

Proceso de objetos comerciales utilizando procedimientos almacenados o sentencias simples de SQL

Los temas siguientes describen cómo el conector procesa los procedimientos almacenados:

- “Operaciones de creación de objetos comerciales” en la página 56
- “Operaciones de actualización de objetos comerciales” en la página 56
- “Operaciones de supresión de objetos comerciales” en la página 57
- “Operaciones de recuperación de objetos comerciales” en la página 57

- “Operaciones RetrieveByContent para objetos comerciales” en la página 58
- “Operaciones Retrieve-for-Update para objetos comerciales” en la página 59

Operaciones de creación de objetos comerciales: Normalmente un procedimiento almacenado Create devuelve valores que el conector utiliza para cumplimentar los atributos simples del objeto comercial de nivel superior. El conector ejecuta los pasos siguientes cuando procesa los procedimientos almacenados Create (BeforeCreate, Create, AfterCreate):

1. Comprueba si el objeto comercial contiene un atributo BeforeCreateSP. Si el objeto comercial contiene el atributo, el conector invoca el procedimiento almacenado BeforeCreate.
2. Si el procedimiento almacenado devuelve valores mediante parámetros de salida, el conector utiliza los valores para definir el valor de atributos simples del objeto comercial.
3. El conector crea los objetos comerciales hijo de cardinalidad simple.
4. Establece los valores de clave foránea del objeto comercial de nivel superior en el valor de clave primaria de cada objeto comercial hijo de cardinalidad simple.
5. Comprueba si el objeto comercial contiene un atributo CreateSP. Si el objeto comercial contiene el atributo, el conector invoca el procedimiento almacenado Create para crear el objeto comercial de nivel superior. Si el objeto comercial no contiene el atributo, el conector crea y ejecuta una sentencia INSERT para crear el objeto comercial de nivel superior.
6. Si el procedimiento almacenado Create devuelve valores mediante parámetros de salida, el conector utiliza los valores para definir el valor de atributos simples del objeto comercial.
7. El conector establece el valor de clave foránea de cada hijo de cardinalidad múltiple en el valor del atributo de clave primaria del padre respectivo del objeto hijo.
8. El conector crea los objetos comerciales hijo de cardinalidad múltiple.
9. Comprueba si el objeto comercial contiene un atributo AfterCreateSP. Si el objeto comercial contiene el atributo, el conector invoca el procedimiento almacenado AfterCreate.
10. Si el procedimiento almacenado devuelve valores mediante parámetros de salida, el conector utiliza los valores para definir el valor de atributos simples del objeto comercial.

El conector puede utilizar los valores devueltos en el paso 10 para cambiar el valor de un objeto comercial que creó en los pasos 3 o 5.

Operaciones de actualización de objetos comerciales: Normalmente un procedimiento almacenado Update devuelve valores que el conector utiliza para cumplimentar los atributos simples del objeto comercial de nivel superior. El conector ejecuta los pasos siguientes cuando procesa los procedimientos almacenados Update (BeforeUpdate, Update, AfterUpdate):

1. Comprueba si el objeto comercial contiene un atributo BeforeUpdateSP. Si el objeto comercial contiene el atributo, el conector invoca el procedimiento almacenado BeforeUpdate.
2. Si el procedimiento almacenado BeforeUpdate devuelve valores mediante parámetros de salida, el conector utiliza los valores para definir el valor de atributos simples del objeto comercial.
3. El conector actualiza los objetos comerciales hijo de cardinalidad simple.

4. Establece los valores de clave foránea del objeto comercial de nivel superior en el valor de clave primaria de cada objeto comercial hijo contenido con cardinalidad simple.
5. Comprueba si el objeto comercial contiene un atributo UpdateSP. Si el objeto comercial contiene el atributo, el conector invoca el procedimiento almacenado Update para actualizar el objeto comercial de nivel superior. Si el objeto comercial no contiene el atributo, el conector crea y ejecuta una sentencia UPDATE para actualizar el objeto comercial de nivel superior.
6. Si el procedimiento almacenado Update devuelve valores mediante parámetros de salida, el conector utiliza los valores para definir el valor de atributos simples del objeto comercial.
7. Establece los valores de clave foránea de los hijos de cardinalidad múltiple para hacer referencia al valor de los correspondientes atributos de clave primaria contenidos en el padre.
8. El conector actualiza los objetos comerciales hijo de cardinalidad múltiple.
9. Comprueba si el objeto comercial contiene un atributo AfterUpdateSP. Si el objeto comercial contiene el atributo, el conector invoca el procedimiento almacenado AfterUpdate.
10. Si el procedimiento almacenado devuelve valores mediante parámetros de salida, el conector utiliza los valores para definir el valor de atributos simples del objeto comercial.

Operaciones de supresión de objetos comerciales: Un procedimiento almacenado Delete no devuelve valores al conector. El conector ejecuta los pasos siguientes cuando procesa los procedimientos almacenados Delete (BeforeDelete, Delete, AfterDelete):

1. Comprueba si el objeto comercial contiene un atributo BeforeDeleteSP. Si el objeto comercial contiene el atributo, el conector invoca el procedimiento almacenado BeforeUpdate.
2. El conector suprime los objetos comerciales hijo de cardinalidad simple.
3. El conector suprime los objetos comerciales hijo de cardinalidad múltiple.
4. Comprueba si el objeto comercial contiene un atributo DeleteSP. Si el objeto comercial contiene el atributo, el conector invoca el procedimiento almacenado Delete para suprimir el objeto comercial de nivel superior. Si el objeto comercial no contiene el atributo, el conector crea y ejecuta una sentencia DELETE.
5. Comprueba si el objeto comercial contiene un atributo AfterDeleteSP. Si el objeto comercial contiene el atributo, el conector invoca el procedimiento almacenado AfterDelete.

Operaciones de recuperación de objetos comerciales: Para las operaciones de recuperación simples, se pueden utilizar procedimientos almacenados para el objeto comercial de nivel superior, los hijos de cardinalidad simple así como para los hijos de cardinalidad múltiple. El orden de los procedimientos es el siguiente:

- BeforeRetrieve
- Retrieve
- AfterRetrieve

El conector crea un objeto temporal para recuperar un objeto comercial hijo de cardinalidad simple o de cardinalidad múltiple. El conector aplica el procedimiento almacenado BeforeRetrieve al objeto comercial temporal. El procedimiento almacenado AfterRetrieve se aplica a cada objeto hijo recuperado del contenedor.

El conector ejecuta el procedimiento almacenado `AfterRetrieve` después de ejecutar sobre el objeto comercial una consulta `Retrieve` generada dinámicamente a partir de los metadatos del objeto comercial o un procedimiento almacenado.

De acuerdo con la especificación JDBC, existen tres tipos de llamadas de procedimiento almacenado (`StoredProcedure`):

- `{call <nombre_procedimiento_almacenado>(?,?,?)}`
- `{call <nombre_procedimiento_almacenado>}`
- `{?= call <nombre_procedimiento_almacenado>(?,?,?)}`

donde `nombre_procedimiento_almacenado` es el nombre del procedimiento almacenado.

El conector admite la utilización de los dos primeros tipos. El conector procesará el conjunto de resultados (`ResultSet`) devuelto por el procedimiento almacenado (`StoredProcedure`).

En la sintaxis del procedimiento almacenado, si `RS=true`, se procesa el conjunto de resultados del procedimiento almacenado. Si `RS=false`, no se procesa el conjunto de resultados. El valor por omisión de `RS` es `false`. Una vez procesados los valores del conjunto de resultados, se procesan las variables de salida del procedimiento almacenado. Si `RS=true`, los hijos de cardinalidad múltiple no pueden especificar las variables de salida en el procedimiento almacenado asociado.

Nota: El proceso del conjunto de resultados sólo se permite para las operaciones del verbo `Retrieve` y para `RetrieveSP`.

Proceso del conjunto de resultados devuelto por un procedimiento almacenado de recuperación (`RetrieveSP`): Para el conjunto de resultados devuelto por el procedimiento almacenado se obtienen metadatos del conjunto de resultados (`ResultSetMetaData`). Se obtienen los valores de todas las columnas del conjunto de resultados y esos valores se asignan al atributo correspondiente del objeto comercial. La propiedad `ColumnName` de la información específica de la aplicación correspondiente a un atributo debe contener el nombre de columna de `ResultSet` para asociar el atributo a la columna.

Para los objetos de cardinalidad simple, el correspondiente conjunto de resultados debe constar de una sola fila. Si se devuelven varias filas en el conjunto de resultados, se notifica un error.

Para los hijos de cardinalidad múltiple, se pueden devolver varias filas a través del conjunto de resultados. Para cada fila devuelta, se crea un nuevo objeto y se añade al contenedor. Luego el contenedor se añade al objeto padre en el índice de atributo necesario.

El hijo de cardinalidad N de un objeto comercial envoltorio tiene atributos de procedimiento almacenado, y los atributos representan los parámetros de entrada y las columnas del conjunto de resultados. `WRAPPER=true` se establece a nivel de la información específica de la aplicación correspondiente al objeto comercial. La información específica de la aplicación correspondiente al objeto comercial hijo tendrá `TN=dummy`.

Operaciones `RetrieveByContent` para objetos comerciales: Para las operaciones `RetrieveByContent` simples, se pueden utilizar procedimientos almacenados sólo para el objeto comercial de nivel superior y sus hijos de cardinalidad simple; es

decir, no se pueden utilizar para devolver un conjunto de resultados ni varias filas. El orden de los procedimientos es el siguiente:

- BeforeRetrieveByContent
- RetrieveByContent
- AfterRetrieveByContent

El conector crea un objeto temporal para recuperar un objeto comercial hijo de cardinalidad simple o de cardinalidad múltiple. Para los objetos comerciales de cardinalidad múltiple, el conector aplica el procedimiento almacenado BeforeRetrieveByContent al objeto comercial temporal. El procedimiento almacenado AfterRetrieveByContent se aplica a cada objeto hijo recuperado del contenedor.

El conector ejecuta el procedimiento almacenado AfterRetrieveByContent después de ejecutar sobre el objeto comercial una consulta RetrieveByContent generada dinámicamente a partir de los metadatos del objeto comercial o un procedimiento almacenado. En este caso, aunque la recuperación de un objeto comercial jerárquico también recupera sus objetos comerciales hijo, el conector ejecuta el procedimiento almacenado AfterRetrieveByContent para cada objeto comercial presente en la matriz.

Operaciones Retrieve-for-Update para objetos comerciales: Los procedimientos almacenados siguientes se invocan para el objeto comercial de nivel superior y recuperan todos los objetos comerciales hijo de la misma manera que la operación Retrieve simple.

El orden de los procedimientos es el siguiente:

- BeforeRetrieveUpdate
- RetrieveUpdate
- AfterRetrieveUpdate

Estos procedimientos almacenados ejecutan las mismas operaciones que BeforeRetrieve y AfterRetrieve. Los procedimientos tienen nombres distintivos para que el usuario pueda crear atributos diferentes que hagan que el conector ejecute tanto operaciones BeforeRetrieve como BeforeRetrieveUpdate, así como operaciones AfterRetrieve y AfterRetrieveUpdate.

El conector crea un objeto temporal para recuperar un objeto comercial hijo de cardinalidad simple o de cardinalidad múltiple. Para los objetos comerciales de cardinalidad múltiple, el conector aplica el procedimiento almacenado BeforeRetrieveUpdate al objeto comercial temporal. El procedimiento almacenado AfterRetrieveUpdate se aplica a cada objeto hijo recuperado del contenedor.

El conector ejecuta el procedimiento almacenado AfterRetrieveUpdate después de ejecutar sobre el objeto comercial una consulta RETRIEVE generada dinámicamente a partir de los metadatos del objeto comercial o un procedimiento almacenado. En este caso, aunque la recuperación de un objeto comercial jerárquico también recupera sus objetos comerciales hijo, el conector ejecuta el procedimiento almacenado AfterRetrieveUpdate para cada objeto comercial presente en la matriz.

Confirmación y retrotracción de transacciones

Cada vez que el conector recibe un objeto comercial para su proceso, el conector inicia un bloque de transacciones. Todas las sentencias de SQL que el conector ejecuta mientras procesa ese objeto comercial están encapsuladas dentro del bloque

de transacciones. Cuando el conector finaliza el proceso del objeto comercial, confirma el bloque de transacciones si el proceso se realizó satisfactoriamente, o retrotrae la transacción si el conector encontró un error.

Propiedades de atributos de objetos comerciales

La arquitectura de los objetos comerciales define diversas propiedades referentes a los atributos. Esta sección describe cómo el conector interpreta varias de estas propiedades y cómo definir las al modificar un objeto comercial.

La propiedad Name

Cada atributo de objeto comercial debe tener un nombre exclusivo.

La propiedad Type

Cada atributo de objeto comercial debe tener un tipo, tal como Integer, String o el tipo de un objeto comercial hijo. Cuando el conector encuentra un atributo de tipo Date, Long Text o String, el conector encierra el valor entre comillas y trata el valor como datos de tipo carácter.

La propiedad Cardinality

Cada atributo de objeto comercial que representa un objeto comercial hijo o matriz de objetos comerciales hijo tiene el valor 1 o n, respectivamente, en este atributo. Todos los atributos que representan objetos comerciales hijo tienen una propiedad ContainedObjectVersion (que especifica el número de versión del hijo) y una propiedad Relationship (que especifica el valor Containment).

La propiedad MaxLength

Si el atributo es de tipo String, esta propiedad especifica la longitud máxima permitida para el valor del atributo.

La propiedad Key

Como mínimo un atributo simple de cada objeto comercial se debe especificar como clave. Para definir un atributo como clave, establezca esta propiedad en Sí. Si el atributo del objeto comercial es de tipo String, es recomendable que el tipo de datos en la base de datos sea Varchar en lugar de char.

Nota: El conector no admite especificar como atributo de clave un atributo que representa un objeto comercial hijo o una matriz de objetos comerciales hijo.

Si la propiedad key se establece en true para un atributo simple, cuando el conector procesa el objeto comercial, añade ese atributo a la cláusula WHERE de las sentencias de SQL SELECT, UPDATE, RETRIEVE y DELETE generadas por el conector.

Si la propiedad key se establece en true para un atributo de un hijo que guarda la relación padre/hijo en el hijo (tanto para la cardinalidad simple como múltiple), el conector utiliza las claves primarias del padre en la cláusula WHERE de la sentencia SELECT, y no utiliza la propiedad Key. Para obtener información sobre la especificación del nombre de atributos de objetos comerciales cuyos valores se utilizan para definir los atributos de clave foránea del hijo, consulte "Información específica de la aplicación a nivel de atributo" en la página 64.

La propiedad IsForeignKey

El conector utiliza esta propiedad para determinar si un atributo es una clave foránea.

La propiedad IsRequired

La propiedad IsRequired especifica si un atributo debe contener un valor.

Si se especifica esta propiedad para un atributo que representa un objeto comercial hijo de cardinalidad simple, el conector exige que el objeto comercial padre contenga un objeto comercial hijo para este atributo.

Cuando el conector recibe un objeto comercial con una petición Create, el conector hace que la operación Create falle si se cumplen las dos condiciones siguientes:

- El objeto comercial no tiene un valor válido ni un valor por omisión para un atributo necesario.
- La información específica de la aplicación no especifica que el conector genere un ID exclusivo.

Cuando el conector recibe un objeto comercial con una petición Retrieve, el conector hace que la operación de recuperación falle si el objeto comercial no tiene un valor válido ni un valor por omisión para un atributo necesario.

El conector no utiliza esta propiedad para atributos que contienen una matriz de objetos comerciales hijo.

Nota: Si el atributo de clave utiliza una secuencia o contador o es cumplimentado por la base de datos (UID=AUTO), no se debe marcar como atributo necesario ("Required").

La propiedad AppSpecificInfo

Para obtener información sobre esta propiedad, consulte "Información específica de la aplicación a nivel de atributo" en la página 64.

La propiedad DefaultValue

Esta propiedad especifica un valor por omisión que el conector utiliza para asignar un valor a un atributo simple si el atributo no tiene un valor obtenido de la tabla de base de datos. El conector no evalúa esta propiedad para atributos que representan un objeto comercial hijo o una matriz de objetos comerciales hijo.

El conector evalúa esta propiedad sólo si la propiedad de configuración UseDefaults tiene el valor true. Para obtener más información, consulte la Tabla 6 en la página 19.

Valores especiales de atributos

Los atributos simples de objetos comerciales pueden tener el valor especial CxIgnore. Cuando el conector recibe un objeto comercial desde el intermediario de integración, el conector pasa por alto todos los atributos cuyo valor sea CxIgnore. Es como si esos atributos fueran invisibles para el conector.

Cuando el conector recupera datos de la base de datos y la sentencia SELECT devuelve un valor null para un atributo, el conector establece el valor de ese atributo en CxIgnore por omisión. Si se ha especificado un valor para el parámetro

UNVL de la información específica de la aplicación correspondiente al atributo, el conector utiliza ese valor para representar el valor null.

Debido a que el conector exige que cada objeto comercial tenga como mínimo un atributo de clave primaria, los desarrolladores deben asegurarse de que los objetos comerciales de WebSphere Business Integration Adapter pasados al conector tengan como mínimo una clave primaria cuyo valor no sea CxIgnore. La única excepción a este requisito es un objeto comercial cuya clave primaria deba ser generada por el conector utilizando una secuencia o contador, o que se generada por la base de datos.

Cuando el conector inserta datos en la base de datos y un atributo de objeto comercial no tiene ningún valor especificado, el conector utiliza el valor especificado por la propiedad UseNullValue del atributo. Para obtener más información sobre UseNullValue, consulte UNVL=value en la Tabla 11 en la página 65.

Información específica de la aplicación para objetos comerciales

Esta sección describe, a nivel de objeto, el formato de la información específica de la aplicación referente a los atributos y verbos para los objetos comerciales compatibles con el conector.

La información específica de la aplicación contenida en las definiciones de objetos comerciales proporciona al conector instrucciones dependientes de la aplicación sobre cómo procesar objetos comerciales. El conector analiza la información específica de la aplicación a partir de los atributos o verbo de un objeto comercial o a partir del propio objeto comercial para generar consultas para operaciones de creación, actualización, recuperación y supresión.

El conector guarda en la antememoria parte de la información específica de la aplicación del objeto comercial y utiliza esta información para crear consultas para todos los verbos.

Si amplía o modifica un objeto comercial específico de la aplicación, asegúrese de que la información específica de la aplicación contenida en la definición del objeto comercial se ajusta a la sintaxis que el conector espera.

Nota: En un entorno de InterChange Server, el rendimiento es mejor cuando los objetos comerciales específicos de la aplicación son menores que 1 MB. Nunca deben sobrepasar los 5 MB. Los objetos comerciales mayores producen problemas de rendimiento debidos a limitaciones de la Máquina Virtual Java en la que se ejecuta InterChange Server.

La Tabla 10 en la página 63 proporciona una visión general de la funcionalidad disponible en la información específica de la aplicación del objeto comercial.

Tabla 10. Visión general de la información específica de la aplicación en los objetos comerciales admitidos

Ámbito de la información específica de la aplicación	Funcionalidad
Objeto comercial completo	<p>Especifica:</p> <ul style="list-style-type: none"> • El nombre de la correspondiente tabla de base de datos. • Define la columna cuyo valor el conector utiliza en la cláusula WHERE para realizar una supresión lógica. • Que el objeto comercial de nivel superior es un objeto envoltorio.
Atributos simples	<p>Especifica:</p> <ul style="list-style-type: none"> • El nombre de la columna de base de datos correspondiente a un atributo. • La relación de clave foránea entre un atributo del objeto comercial actual y un objeto comercial padre o hijo. • La generación automática de valores de identificador exclusivos. • El nombre de otro atributo dentro del mismo objeto comercial cuyo valor el conector debe utilizar para definir el valor del atributo actual. • La indicación de si se debe utilizar el atributo actual al clasificar una recuperación. • El valor que se debe utilizar cuando el valor del atributo actual es null. • La forma de realizarse la sustitución de series de caracteres. • La indicación de si se debe utilizar el operador LIKE o el operador = al comparar series de caracteres. • El valor que se debe utilizar como posición comodín cuando se utiliza el operador LIKE.
Atributos que contienen un hijo o una matriz de objetos comerciales hijo	Especifica si un hijo de cardinalidad simple es propiedad del padre. Especifica si el conector suprime datos del hijo durante una operación de actualización si los datos no están representados en el objeto comercial de origen.
Verbo de objeto comercial	Se utiliza solamente para el verbo Retrieve. Este texto especifica los atributos que se deben incluir en la cláusula WHERE para una recuperación. El usuario puede también especificar operadores y valores de atributos.

Las secciones siguientes describen esta funcionalidad con más detalle.

Información específica de la aplicación a nivel de objeto comercial

La información específica de la aplicación a nivel de objeto comercial le permite:

- Especificar el nombre de la correspondiente tabla de base de datos.
- Proporcionar la información necesaria para realizar una supresión física o lógica.
- Especificar que el objeto comercial de nivel superior es un objeto envoltorio.

A nivel de objeto comercial, la información específica de la aplicación consta de parámetros separados por un signo de dos puntos (:) o un signo de punto y coma (;) que actúan como delimitadores:

TN=nombreTabla; SCN=nombreColumnaEstado:valorEstado; WRAPPER=true|false

donde `nombreTabla` identifica la tabla de base de datos, `nombreColumnaEstado` es el nombre de la columna de base de datos utilizada para realizar supresiones lógicas y `valorEstado` es el valor que denota que un objeto comercial está inactivo o suprimido.

Por ejemplo, suponga que el objeto comercial "Customer" (Cliente) tiene el valor siguiente especificado para su información específica de la aplicación a nivel de objeto comercial:

```
TN=CUSTOMER; SCN=CUSTSTATUS:DELETED
```

Suponga también que el conector recibe una petición para suprimir el cliente. Tal valor hace que el conector emita la siguiente sentencia de SQL:

```
UPDATE CUSTOMER SET CUSTSTATUS = 'DELETED' WHERE CUSTOMER_ID = 2345
```

Si no se incluye el parámetro SCN o no se especifica ningún valor para él, el conector suprime físicamente el objeto comercial de la base de datos. O de otro modo, si el objeto comercial con el verbo Delete incluye el parámetro SCN en su información específica de la aplicación, el conector realiza una supresión lógica. Si el objeto comercial con el verbo Delete no incluye el parámetro SCN en su información específica de la aplicación, el conector realiza una supresión física.

Tanto la operación de actualización como la de supresión pueden utilizar el valor de la propiedad SCN:

- Cuando el conector realiza una actualización, utiliza el valor de su propiedad `ChildUpdatePhyDelete` para determinar si debe realizar una supresión física o lógica de los datos faltantes del hijo. Si realiza una supresión lógica de los datos del hijo, el conector utiliza el valor de su parámetro SCN para obtener el nombre de la columna de estado y el texto del valor de estado. Para obtener más información, consulte "Operaciones Update (actualización)" en la página 48.
- Cuando el conector realiza una supresión, utiliza el valor de su parámetro SCN para determinar si debe realizar una supresión física o lógica del objeto comercial completo. Si el parámetro SCN contiene un valor, el conector realiza una supresión lógica. Si el parámetro SCN no contiene un valor, el conector realiza una supresión física. Para obtener más información, consulte "Operaciones Delete (supresión)" en la página 51.

A nivel de objeto comercial, la información específica de la aplicación se puede utilizar para especificar un objeto envoltorio:

```
WRAPPER=true|false
```

si el parámetro WRAPPER tiene el valor "true", el objeto comercial de nivel superior es un objeto envoltorio. El objeto envoltorio no está representado por una tabla ni una vista de base de datos. Un objeto envoltorio se utiliza como contenedor para objetos comerciales que no guardan relación entre sí. El conector pasa por alto el objeto de nivel superior y procesa solamente los hijos. El objeto envoltorio puede contener entidades de cardinalidad N o de cardinalidad N-1 o ambas cosas.

Información específica de la aplicación a nivel de atributo

La información específica de la aplicación para atributos difiere según si el atributo es un atributo simple o un atributo que representa n hijo o una matriz de objetos comerciales hijo. La información específica de la aplicación para un atributo que representa un hijo también difiere según si la relación padre/hijo se guarda en el hijo o en el padre. Para obtener información sobre la información específica de la

aplicación para atributos que representan un hijo o una matriz de objetos comerciales hijo, consulte “Especificación de la clave foránea de un atributo” en la página 67.

Información específica de la aplicación para atributos simples

Para los atributos simples, el formato de la información específica de la aplicación consta de once parámetros en forma de pares nombre-valor, cada uno de los cuales incluye el nombre del parámetro y su valor. Cada parámetro definido está separado del siguiente por un signo de dos puntos (:) que actúa de delimitador.

El formato de la información específica de la aplicación para atributos se muestra a continuación. Los corchetes rectos ([]) delimitan parámetros opcionales. Una barra vertical (|) separa los miembros de un conjunto de opciones. El signo de dos puntos (:) está reservado para utilizarlo como delimitador.

```
CN=n_col: [FK=[n_objeto_cf.]n_atributo_cf]: [UID=[AUTO|n_idu| n_esquema.n_idu
[=UseIfMissing]|CW.n_columna_idu [=UseIfMissing]]]:
[PH=true|false]: [CA=n_atributo_definido| ..n_atributo_definido]:
[OB=[ASC|DESC]]: [UNVL= valor]:
[ESC=true|false]:
[FIXEDCHAR=true|false]:
[BYTEARRAY=true|false]: [USE_LIKE=true|false]:
[WILDCARD_POSITION=número no negativo|NONE|BEGIN|END|BOTH]:
[CLOB=true]:
{TS=true|false}]
```

El único parámetro necesario para un atributo simple que desee que procese el conector es el nombre de columna. Por ejemplo, para especificar solamente el nombre de columna, utilice este formato:

```
CN=id_cliente
```

La Tabla 11 describe cada parámetro formado por un par nombre-valor.

Tabla 11. Parámetros de la forma nombre-valor en la información específica de la aplicación para atributos

Parámetro	Descripción
CN= <i>nombre_col</i>	El nombre de la columna de base de datos para este atributo.
FK=[<i>nombre_objeto_cf</i> .] <i>nombre_atributo_cf</i>	El valor de esta propiedad depende de si la relación padre/hijo se guarda en el objeto comercial padre o en el hijo. Si un atributo no es una clave foránea, no incluya este parámetro en la información específica de la aplicación. Para obtener más información, consulte “Especificación de la clave foránea de un atributo” en la página 67.
UID=AUTO	El conector utiliza este parámetro para generar el ID exclusivo del objeto comercial. Si un atributo no necesita que se cree un ID exclusivo, no incluya este parámetro en la información específica de la aplicación.
UID= <i>nombre_idu</i> <i>nombre_esquema.nombre_idu</i> [=UseIfMissing]	Consulte la descripción de la propiedad PreserveUIDSeq para conocer detalles sobre cómo preservar el ID exclusivo durante el proceso de objetos comerciales. Para obtener más información, consulte “Generación de un identificador exclusivo de un objeto comercial” en la página 70.
UID=CW. <i>nombre_columna_idu</i> [=UseIfMissing]	Nota: CW es una palabra clave utilizada para representar el tipo de ID de usuario y no representa el nombre de tabla.
PH=true false	Si PH=true , el atributo simple correspondiente es un atributo marcador de posición. Un atributo simple es también un marcador de posición si su información específica de la aplicación está en blanco o es nula.

Tabla 11. Parámetros de la forma nombre-valor en la información específica de la aplicación para atributos (continuación)

Parámetro	Descripción
CA= <i>n_atributo_def</i> <i>.n_atributo_def</i>	Si el valor de <code>nombre_atributo_definido</code> es el nombre de otro atributo contenido dentro del objeto comercial individual actual, el conector utiliza el valor del atributo especificado para establecer el valor de este atributo antes de que añada el objeto comercial a la base de datos durante la operación Create. El valor de <code>set_attr_name</code> no puede hacer referencia a un atributo de un objeto comercial hijo, pero puede hacer referencia a un atributo del objeto comercial padre si <code>set_attr_name</code> está precedido por dos puntos (.). Si no incluye este parámetro en la información específica de la aplicación, el conector utiliza el valor del atributo actual sin copiar el valor del atributo (CA) desde otro atributo.
OB=[ASC DESC]	Si se especifica un valor para este parámetro y el atributo reside en un objeto comercial hijo, el conector utiliza el valor del atributo en la cláusula ORDER BY de las consultas de recuperación. El conector puede recuperar objetos comerciales hijo en orden ascendente o descendente. Utilice ASC para especificar un orden ascendente de recuperación. Utilice DESC para especificar un orden descendente de recuperación. Si no incluye este parámetro en la información específica de la aplicación, el conector no utiliza este atributo al especificar un orden de recuperación.
UNVL= <i>valor</i>	Especifica el valor que el conector utiliza para representar un valor nulo cuando recupera un objeto comercial que tiene atributos cuyos valores son nulos. Si no incluye este parámetro en la información específica de la aplicación, el conector inserta CxIgnore como valor del atributo.
ESC=[true false]	Determina si el conector sustituye todas las apariciones de cada carácter identificado en la propiedad ReplaceAllStr por las series de caracteres de sustitución también especificadas en la propiedad ReplaceStrList. Si este parámetro no contiene un valor, el conector utiliza el valor de la propiedad ReplaceStrList para realizar esa determinación. Nota: El parámetro ESC y las propiedades ReplaceAllStr y ReplaceStrList son compatibles con la utilización de caracteres de escape para una base de datos (por ejemplo, comillas simples utilizadas como carácter de escape). Debido a que esta misma funcionalidad está también disponible utilizando las sentencias preparadas proporcionadas por el controlador JDBC, estas propiedades se eliminarán en versiones futuras del conector. El conector es actualmente compatible con el uso de las sentencias preparadas de JDBC.
FIXEDCHAR=true false	Especifica si el atributo es de longitud fija cuando las columnas de la tabla son de tipo CHAR, no VARCHAR. Por ejemplo, si un atributo determinado está asociado a una columna de tipo CHAR, el conector espera FIXEDCHAR para la longitud; para la información específica de la aplicación correspondiente a ese atributo, especifique FIXEDCHAR=true. Asegúrese de que la propiedad MaxLength del atributo tiene longitud CHAR, que se especifica en la base de datos. Por omisión, FIXEDCHAR=false.
BYTEARRAY=true false	Si BYTEARRAY=true, el conector lee y escribe datos binarios en la base de datos y envía esos datos en forma de serie de caracteres al intermediario InterChange Server o WebSphere MQ Integrator. BYTEARRAY=false es el valor por omisión. Para obtener más información, consulte "Utilización de datos binarios" en la página 71.
USE_LIKE=true false	Especifica si el conector compara series de caracteres utilizando el operador = o el operador LIKE. Si USE_LIKE tiene el valor "true", se pueden ejecutar consultas de comodín definiendo WILDCARD_POSITION. Si USE_LIKE tiene el valor "false", se utilizará el operador =.

Tabla 11. Parámetros de la forma nombre-valor en la información específica de la aplicación para atributos (continuación)

Parámetro	Descripción
WILDCARD_POSITION=número no negativo NONE BEGIN END BOTH	Si USE_LIKE tiene el valor "true", se utiliza WILDCARD_POSITION para especificar la posición del carácter comodín. Este valor puede un número no negativo cualquiera, NONE, BEGIN, END o BOTH. Por ejemplo, si se utiliza BEGIN, el carácter comodín se colocará en la primera posición de la serie de caracteres (%serie). Si se utiliza END, el carácter comodín se colocará en la última posición de la serie de caracteres (serie%). Si se utiliza BOTH, se colocarán caracteres comodines en la primera y última posición de la serie de caracteres (%serie%).
CLOB=true	sólo es aplicable al tipo de atributo String. Especifica que el tipo de datos de la columna de base de datos correspondiente a este atributo es CLOB. Nota: El tipo de datos CLOB se define de la manera siguiente: <ul style="list-style-type: none"> • El atributo CLOB tiene el tipo de datos String, cuya longitud se utiliza para definir la longitud del CLOB • El atributo CLOB tiene ASI=CN=xyz; CLOB=true • Cualquier otro tipo de atributo que haga referencia a CLOB en la información específica de la aplicación dará un error como resultado. • CLOB=false daría lugar a un error como resultado <p>Un tipo String regular sería el mismo y sin ninguna referencia a CLOB en la información específica de la aplicación. Los tipos de datos CLOB de 4k y mayores se pueden insertar o actualizar. Pero sólo se pueden utilizar con Oracle y necesitan la última versión del controlador "thin" que tenga la capacidad de trabajar con los CLOB. La utilización de cualquier otro controlador puede causar errores.</p>
TS=true false	Para los atributos de tipo DATE, cuando se especifica TS=false en la información específica de la aplicación correspondiente al atributo, el atributo se trata como si fuera de tipo DATE. Cuando TS=true, el atributo se trata como si fuera de tipo TIMESTAMP. El valor por omisión de TS es "true".

Nota: Si la información específica de la aplicación contenida en cualquiera de los atributos de un objeto comercial no hace que el conector cree ni ejecute una consulta, el conector escribe un aviso en el archivo de anotaciones cronológicas y prosigue su proceso. El conector no emite una excepción ni devuelve un error.

Especificación de la clave foránea de un atributo: El valor de esta propiedad depende de si la relación padre/hijo se guarda en el objeto comercial padre o en el hijo:

- Si la relación se guarda en el padre: asigne como valor de esta propiedad el tipo del objeto comercial hijo y el nombre del atributo contenido en el hijo para que se utilice como clave foránea.
- Si la relación se guarda en el hijo: asigne como valor de esta propiedad solamente el nombre del atributo contenido en el padre para que se utilice como clave foránea.

Si el valor de nombre_objeto_cf no coincide con el tipo del objeto comercial hijo, y el valor de nombre_atributo_cf no coincide con el nombre del atributo contenido en el padre o en el hijo (según corresponda), el conector no puede procesar este atributo como clave foránea. El uso de mayúsculas o minúsculas en el nombre del objeto comercial y en el nombre del atributo tiene implicaciones importantes.

Por ejemplo, suponga que el objeto comercial Cliente contiene el atributo Direc[1], que representa el objeto comercial hijo Dirección, y el atributo AID, que contiene la clave primaria del objeto comercial hijo en forma de clave foránea. En este caso, la información específica de la aplicación correspondiente al atributo de clave foránea del padre debe contener el tipo del objeto comercial hijo (Dirección) así como el nombre de su atributo de clave primaria (ID). En este ejemplo, la información específica de la aplicación correspondiente al atributo AID incluiría FK=Dirección.ID.

Designación de un atributo de clave foránea: Varios objetos comerciales padre pueden contener el mismo objeto comercial hijo, con independencia de si el hijo está contenido con cardinalidad simple o múltiple y de si la relación padre/hijo se guarda en el padre o en el hijo. Sin embargo, todos los objetos comerciales padre que guardan la relación padre/hijo deben utilizar atributos de igual nombre para contener la clave primaria del hijo. Además, todos los objetos comerciales hijo que guardan la relación padre/hijo deben utilizar atributos del mismo nombre para contener la clave primaria del padre. La Figura 5 muestra estas relaciones.

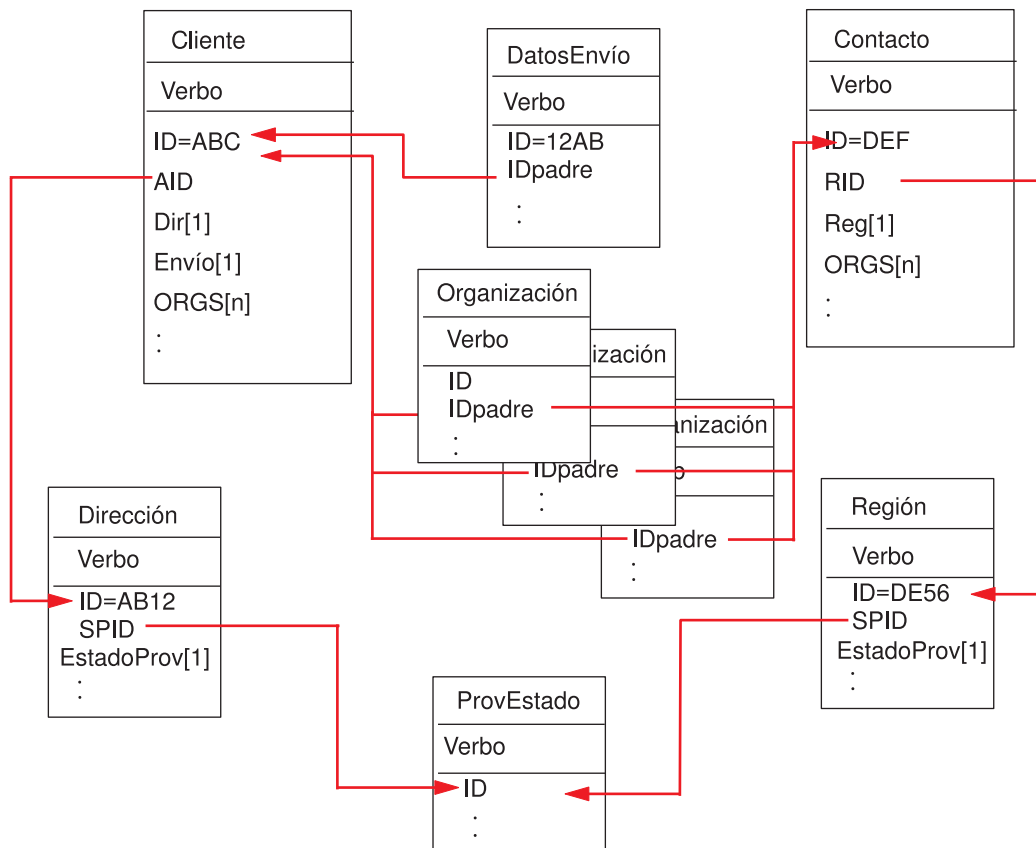


Figura 5. Ejemplo de relaciones existentes entre objetos comerciales

La Figura 5 muestra las relaciones siguientes:

- El atributo ORGS[n] del Cliente ABC y del Contacto DEF representa una matriz de objetos comerciales Organización. El valor de clave foránea para cada objeto comercial de la matriz de Organizaciones corresponde al valor de clave primaria contenido en el atributo ID de los objetos comerciales Cliente y Contacto. En este caso, cada objeto comercial de la matriz está contenido en varios padres.

La información específica de la aplicación para el atributo ORGS podría ser:

KEEP_RELATIONSHIP=true

Para obtener más información sobre el parámetro KEEP_RELATIONSHIP, consulte “Información específica de la aplicación para atributos que representan hijos” en la página 70.

La información específica de la aplicación para el atributo IPpadre de cada hijo de la matriz de Organizaciones contiene el nombre de la columna de base de datos que corresponde al atributo actual, y especifica la clave foránea del atributo actual incluyendo el nombre del atributo de clave primaria del padre; por ejemplo:

CN=ORG_ID:FK=ID

Nota: Para que varios objetos comerciales contengan el mismo hijo (cuando la relación padre/hijo se guarda en el hijo), todos los objetos comerciales padre deben utilizar un atributo de igual nombre para contener la clave foránea del hijo. El parámetro de clave primaria de la información específica de la aplicación correspondiente a ese hijo identifica sólo el nombre del atributo, y no el tipo del objeto comercial padre. El conector supone que el padre directo es el propietario de cada hijo.

- El atributo Direc[1] de Cliente representa el objeto comercial Dirección con propietario. El atributo AID de Cliente identifica la clave primaria del objeto comercial Dirección como clave foránea contenida en el padre. En este caso, el atributo de clave foránea del padre debe contener el tipo del objeto comercial hijo así como el nombre de su atributo de clave primaria. El hijo de cardinalidad simple, Dirección, está contenido solamente en un único padre.

La información específica de la aplicación para el atributo Direc es:

CONTAINMENT=OWNERSHIP

La información específica de la aplicación para el atributo AID contiene el nombre de la columna de base de datos que corresponde al atributo actual, y especifica la clave foránea del atributo actual incluyendo el tipo del objeto comercial hijo y el nombre de su atributo de clave primaria; por ejemplo:

CN=FK_AD:FK=Dirección.ID

La información específica de la aplicación para el atributo del clave primaria del hijo es:

CN=pk

- Los atributos EstadoProv[1] de los objetos comerciales Dirección y Región representan el objeto comercial EstadoProvincia sin propietario. Los atributos SPID de los objetos comerciales Dirección y Región contienen el tipo del objeto comercial hijo (EstadoProvincia) y el nombre de su atributo de clave primaria, que actúa como clave foránea del padre. Un mismo hijo de cardinalidad simple, EstadoProvincia, está contenido en varios padres.

La información específica de la aplicación para el atributo SPID es:

CONTAINMENT=NO_OWNERSHIP

Para obtener más información sobre el parámetro CONTAINMENT, consulte “Información específica de la aplicación para atributos que representan hijos” en la página 70.

La información específica de la aplicación para el atributo SPID de Dirección contiene el nombre de la columna de base de datos que corresponde al atributo actual, y especifica la clave foránea del atributo actual incluyendo el tipo del objeto comercial hijo y el nombre de su atributo de clave primaria; por ejemplo:

CN=FK_SP:FK=EstadoProvincia.ID

La información específica de la aplicación para el atributo del clave primaria del hijo es:

CN=SP_ID

Nota: Para que varios objetos comerciales (que guardan la relación padre/hijo en el padre) contengan el mismo hijo, todos los objetos comerciales hijo deben utilizar un atributo de igual nombre para contener la clave foránea del padre.

- El atributo Envío[1] de Cliente representa el objeto comercial DatosEnvío, donde está contenida la información de envío del cliente. El atributo ID de Cliente actúa como clave foránea para los datos de envío. En este caso, debido a que DatosEnvío no puede existir con independencia de su padre y se crea solamente después de haberse creado su padre, la relación padre/hijo se guarda en el hijo. La información específica de la aplicación para el atributo IPpadre del hijo contiene el nombre de la columna de base de datos que corresponde al atributo actual, y especifica la clave foránea del atributo actual incluyendo el nombre del atributo de clave primaria del padre; por ejemplo:

CN=SD_ID:FK=ID

Generación de un identificador exclusivo de un objeto comercial: El conector utiliza el parámetro UID para generar el ID exclusivo del objeto comercial. El conector genera los ID exclusivos utilizando secuencias (tal como hace Oracle), o contadores (que tienen estructura de tabla), y luego emite la sentencia INSERT.

El conector utiliza una secuencia o contador para generar el valor de ID y luego emite la sentencia INSERT:

- Si UID = *nombre_idu*, el valor de *nombre_idu* proporciona el nombre de la secuencia de Oracle que el conector utiliza para generar un ID exclusivo para el atributo. Después de obtener el valor de secuencia, el conector cumplimenta el atributo de clave y emite la sentencia INSERT.
- Si UID = *nombre_idu=UseIfMissing* y el valor del atributo no es CxIgnore, el conector utiliza el valor del atributo en lugar de generar un ID exclusivo. El parámetro =UseIfMissing no puede contener espacios y distingue entre mayúsculas y minúsculas.
- Si UID=CW.*nombre_columna_idu*, el conector utiliza una tabla contador para generar un ID exclusivo para el atributo. La tabla creada tiene una sola columna, denominada id, y su nombre se puede configurar. Puede personalizar la tabla para añadir una columna para cada atributo que necesite la creación de un UID. Utilice el parámetro *nombre_columna_idu* para especificar el nombre de la columna que el conector debe utilizar al crear el ID exclusivo. Observe que el conector sólo admite el tipo de datos numérico para las columnas que requieren la creación de un ID de usuario.

Para obtener información sobre la configuración del nombre de la tabla, consulte UniqueIDTableName. El script para instalar esta tabla es:

```
\connectors\Manugistics\dependencias\BIA_uid_table_oracle.sql
```

- Si UID =CW.*nombre_columna_idu=UseIfMissing* y el valor del atributo no es CxIgnore, el conector utiliza el valor del atributo en lugar de generar un ID exclusivo. El parámetro =UseIfMissing no puede contener espacios y no distingue entre mayúsculas y minúsculas.

Consulte la propiedad "PreserveUIDSeq" en la página 27 para obtener información sobre la preservación de la secuencia del ID exclusivo durante el proceso.

Información específica de la aplicación para atributos que representan hijos

Los atributos que representan un objeto comercial hijo de cardinalidad simple pueden especificar si el hijo es propiedad del padre o si es compartido por varios padres.

Los atributos que representan un hijo de cardinalidad simple o una matriz de objetos comerciales hijo pueden especificar la forma en que el conector actualiza el objeto padre y un subconjunto de los hijos.

Atributos que representan un objeto comercial hijo de cardinalidad simple: El formato de la información específica de la aplicación para atributos que representan un objeto comercial hijo de cardinalidad simple es el siguiente:

```
CONTAINMENT= [OWNERSHIP|NO_OWNERSHIP]
```

Establezca CONTAINMENT en OWNERSHIP para representar una relación de cardinalidad simple en la que el objeto comercial hijo es propiedad del padre. Establezca CONTAINMENT en NO_OWNERSHIP para representar una relación de cardinalidad simple en la que el objeto comercial hijo es compartido por el padre. No incluya el parámetro CONTAINMENT cuando desee representar una relación de cardinalidad simple en la que la relación padre/hijo se guarda en el hijo, y no en el padre.

Para obtener más información, consulte “Relaciones de cardinalidad simple y datos sin propietario” en la página 39 y “Relaciones de cardinalidad simple que almacenan la relación en el hijo” en la página 41.

Atributos que representan un hijo en el que se guarda la clave del padre: Para las operaciones de actualización sobre una matriz de objetos comerciales que guardan la relación padre/hijo en el hijo, existe un valor especial para el atributo que representa al hijo: puede establecer KEEP_RELATIONSHIP en true para impedir que el conector suprima los datos existentes del hijo que no están representados en el objeto comercial de origen.

Por ejemplo, suponga que un contrato está asociado a una ubicación, tal como Nueva York. Suponga también que el conector recibe una petición para actualizar el objeto comercial Contrato, el cual contiene un objeto comercial hijo simple que establece una asociación con San Francisco como ubicación. Si la evaluación de KEEP_RELATIONSHIP da como resultado true para el atributo que representa los datos de ubicación, el conector actualiza el contrato para añadir su asociación con San Francisco y no suprime su asociación con Nueva York.

Sin embargo, si la evaluación de KEEP_RELATIONSHIP da como resultado false, el conector suprime todos los datos existentes del hijo que no estén contenidos en el objeto comercial de origen. En este caso, el contrato se asocia con San Francisco solamente.

El formato de esta información específica de la aplicación es:

```
KEEP_RELATIONSHIP=[true|false]
```

No se distingue entre letras mayúsculas y minúsculas al examinar esta información específica de la aplicación.

Utilización de datos binarios: Si BYTEARRAY=true, el conector lee y escribe datos binarios en la base de datos. Debido a que la versión actual del sistema WebSphere Business Integration no permite trabajar con datos binarios, éstos se convierten en una serie de caracteres (String) y luego se envían al intermediario de integración. El formato de esta serie de caracteres es un número hexadecimal con 2 caracteres por byte. Por ejemplo, si los datos binarios de la base de datos son 3 bytes que tienen los valores decimales (1, 65, 255), la serie de caracteres será "0141ff".

Formato de la información específica de la aplicación para verbos

El conector utiliza información específica de la aplicación para los verbos Retrieve y RetrieveByContent. Este texto le permite especificar los atributos que se deben incluir en la cláusula WHERE para una recuperación. Puede también especificar operadores y valores de atributos.

La sintaxis de la información específica de la aplicación para los verbos Retrieve y RetrieveByContent es:

```
[variable_condición operador_condicional @ [...]:[..]nombre_atributo [, ...]]
```

donde:

<i>variable_condición</i>	Es el nombre de la columna de base de datos.
<i>operador_condicional</i>	Es el operador compatible con la base de datos, por ejemplo =, >, OR, AND e IN (<i>valor1</i> , <i>valor2</i>).
@	Es una variable que se sustituye por el valor recuperado por <code>getAttribute(nombre_atributo)</code> . La sustitución es posicional; es decir, el conector sustituye el primer signo @ por el valor de la primera variable <i>nombre_atributo</i> especificada a continuación del carácter delimitador <code>:</code> .
..	El atributo especificado en la variable <i>nombre_atributo</i> pertenece al objeto comercial padre inmediato; si falta este valor, el atributo pertenece al objeto comercial actual.
<i>nombre_atributo</i>	Es el nombre del atributo cuyo valor el conector utiliza como sustituto de @.

Para comprender la sintaxis de esta propiedad, suponga que el objeto comercial Artículo tiene un atributo `id_artículo` cuyo valor es XY45 y un atributo `Color` cuyo valor es ROJO. Suponga también que especifica la propiedad `AppSpecificInfo` del verbo Retrieve de la manera siguiente:

```
Color='ROJO'
```

El valor anterior perteneciente a la información específica de la aplicación hace que el conector genere la siguiente cláusula WHERE para una operación de recuperación:

```
where id_artículo=XY45 and Color = 'ROJO'
```

Veamos un ejemplo más complejo. Suponga que el objeto comercial Cliente tiene un atributo `id_cliente` cuyo valor es 1234 y un atributo `fecha_creación` cuyo valor es 01/01/90. Suponga también que el padre de este objeto comercial tiene un atributo `cantidad` cuyo valor es 20.

Suponga también que especifica la propiedad `AppSpecificInfo` del verbo Retrieve de la manera siguiente:

```
fecha_creación > @ OR cantidad = @ AND estado_cliente IN ('GOLD', 'PLATINUM') : fecha_creación, ..cantidad
```

El valor anterior perteneciente a la información específica de la aplicación hace que el conector genere la siguiente cláusula WHERE para una operación de recuperación:

```
where id_cliente=1234 and fecha_creación > '01/01/90'  
OR cantidad = 20 AND estado_cliente IN ('GOLD', 'PLATINUM')
```

El conector obtiene el valor de fecha ('01/01/90') a partir del atributo fecha_creación del objeto comercial actual. El conector obtiene el valor de cantidad (20) a partir del atributo de cantidad del objeto comercial padre (tal como está especificado por cantidad en la información específica de la aplicación).

Después de analizar la información específica de la aplicación para el verbo Retrieve, el conector añade el texto a la cláusula WHERE de la sentencia RETRIEVE que genera a partir de las claves primarias o foráneas del objeto comercial. El conector añade el operador inicial AND a la cláusula WHERE. El valor de la información específica de la aplicación debe ser una sintaxis válida de SQL. En el caso de RetrieveByContent, la información específica de la aplicación se añade a la cláusula WHERE de la sentencia RETRIEVE que el conector genera a partir de los atributos del objeto comercial cuyos valores están cumplimentados.

La cláusula WHERE puede también especificar atributos marcadores de posición en lugar de los atributos propiamente dichos en el objeto comercial padre. Estos marcadores de posición no contienen ninguna información específica de la aplicación. Un atributo puede ser un marcador de posición si cumple una de las condiciones siguientes con respecto a su información específica de la aplicación:

1. Atributo simple con ASI=null o ''
2. Atributo simple con ASI=PH=TRUE

Por ejemplo: el objeto comercial Pedido contiene el objeto comercial Línea de detalle, de cardinalidad múltiple, y es necesario recuperar solamente determinadas líneas de detalle. Esta recuperación se puede realizar mediante un atributo marcador de posición en el objeto comercial Pedido. Este marcador de posición es necesario en el objeto comercial padre porque todos los objetos hijo son eliminados. Durante la ejecución, el atributo marcador de posición puede ser cumplimentado por el intermediario de integración con una lista de líneas de detalle determinadas, separadas por una coma (,).

Para los fines de este ejemplo, añadiría la información siguiente a la cláusula WHERE del verbo Retrieve para el objeto comercial hijo Línea de detalle:

```
id_línea_detalle in(@,@,):..marcador_posición1,..marcador_posición2,..marcador_posición3
```

Donde id_línea_detalle in es el ID contenido en el objeto comercial hijo, marcador_posición es el atributo contenido en el padre. Si marcador_posición contiene los valores 12,13,14, la consulta seleccionaría lo siguiente de la cláusula WHERE:

```
id_línea_detalle in(12,13,14)
```

Donde SELECT:..FROM:..WHERE x in (1,2,3) es la sintaxis estándar de SQL para bases de datos.

En el verbo RetrieveByContent, si la longitud de la cláusula WHERE es 0, el conector utiliza la información específica de la aplicación en la cláusula WHERE de la sentencia RETRIEVE. Mediante esta característica, el usuario puede enviar un objeto comercial sin ningún valor de atributo cumplimentado y especificar información específica de la aplicación para el verbo RetrieveByContent, y el conector creará la cláusula WHERE basándose solamente en la información específica de la aplicación especificada para el verbo.

Capítulo 4. Creación de definiciones de objetos comerciales utilizando IBM ODA for Manugistics

Este capítulo describe IBM ODA for Manugistics, un agente de descubrimiento de objetos (Object Discovery Agent, ODA), que crea definiciones de objetos comerciales para el conector de JDBC. Debido a que el conector trabaja con objetos que están basados en tablas o en vistas, ODA for Manugistics utiliza tablas de base de datos, vistas, procedimientos almacenados y sinónimos/apodos para descubrir requisitos de objetos comerciales que son específicos de la fuente de datos JDBC del objeto.

Nota: Para comprender cómo trabaja ODA for Manugistics, puede ser útil tener conocimientos sobre conceptos de bases de datos y controladores JDBC.

Este capítulo contiene las secciones siguientes:

- “Instalación y utilización”
- “Utilización de ODA for Manugistics en el Diseñador de objetos comerciales” en la página 79
- “Contenido de la definición generada” en la página 89
- “Archivo de definición de objeto comercial de ejemplo” en la página 92
- “Inserción de atributos que contienen objetos comerciales hijo” en la página 93
- “Adición de información a la definición de objeto comercial” en la página 93

Instalación y utilización

Esta sección trata los temas siguientes:

- “Instalación de ODA for Manugistics”
- “Antes de utilizar ODA for Manugistics” en la página 76
- “Inicio de ODA for Manugistics” en la página 77
- “Ejecución de varias instancias de ODA for Manugistics” en la página 77
- “Utilización de archivos de mensajes error y de rastreo” en la página 78

Instalación de ODA for Manugistics

Para instalar ODA for Manugistics, utilice el programa de instalación de IBM WebSphere Business Integration Adapter for Manugistics. Siga las instrucciones contenidas en el manual *System Installation Guide for UNIX* o el manual *System Installation Guide for Windows*. Cuando finaliza la instalación, los archivos siguientes están instalados en el directorio del sistema donde instaló el producto:

- ODA\Manugistics\BIA_ManugisticsODA.jar
- ODA\messages\BIA_ManugisticsODAAgent.txt
- ODA\messages\BIA_ManugisticsODAAgent_11_11.txt (archivos de mensajes específicos de un idioma (*11*) y de un país o territorio (*11*)).
- ODA\Manugistics\start_ManugisticsODA.bat (Windows solamente)
- ODA\Manugistics/start_ManugisticsODA.sh (UNIX solamente)
- bin\CWODAEV.bat (Windows solamente)
- bin\CWODAEV.sh (UNIX solamente)

Nota: A menos que se indique otra cosa, el presente documento utiliza barras inclinadas invertidas (\) como convenio para representar las vías de acceso de directorios. Para las instalaciones en UNIX, utilice barras inclinadas (/) en lugar de barras inclinadas invertidas. Todas las vías de acceso del producto están referidas al directorio donde está instalado el producto en el sistema.

Antes de utilizar ODA for Manugistics

Antes de poder ejecutar ODA for Manugistics, debe seguir estos pasos:

1. Instale el controlador JDBC apropiado. Siga las instrucciones contenidas en el manual *System Installation Guide for UNIX* o el manual *System Installation Guide for Windows*.

Importante: ODA for Manugistics se puede conectar a una base de datos cualquiera utilizando un controlador JDBC que sea compatible con JDBC 2.0 o posterior.

2. Debido a que ODA for Manugistics crea nombres de objetos comerciales y nombres de atributos a partir de los nombres de las correspondientes tablas y columnas de bases de datos, y debido a que esos nombres deben estar especificados utilizando ISO Latin-1, compruebe que los componentes de base de datos apropiados tengan nombres expresados mediante Latin-1. Si no es así, tiene las opciones siguientes:
 - Cree manualmente la definición del objeto comercial en el Diseñador de objetos comerciales.
 - Edite la definición creada por ODA for Manugistics para que todos los nombres de objetos comerciales y nombres de atributos estén expresados utilizando Latin-1.
3. Abra en la modalidad de edición el shell de UNIX o el archivo de proceso por lotes de Windows y configure los valores descritos en la Tabla 12.

Tabla 12. Variables de configuración del shell y del archivo de proceso por lotes

Variable	Explicación	Ejemplo
AGENTNAME	Nombre del ODA	UNIX: AGENTNAME=ManugisticsODA Windows: set AGENTNAME=ManugisticsODA
AGENT	Nombre del archivo jar del ODA	UNIX: AGENT=\$CROSSWORLDS/ODA/Manugistics/BIA_ManugisticsODA.jar Windows: set AGENT=%CROSSWORLDS%\ODA\Manugistics\BIA_ManugisticsODA.jar
DRIVERPATH	Vía de acceso de la biblioteca del controlador JDBC; ODA for Manugistics utiliza las clases de controlador para establecer una conexión con una base de datos especificada.	UNIX: DRIVERPATH=\$CROSSWORLDS/lib/ \xwutil.jar:\$CROSSWORLDS/lib/ \xwbase.jar:\$CROSSWORLDS/lib/ \xwsqserver.jar:\$CROSSWORLDS/lib/ \spy/lib/spy.jar Windows: set DRIVERPATH=%CROSSWORLDS%\lib\xwutil.jar;%CROSSWORLDS%\lib\ /xwbase.jar;%CROSSWORLDS%\lib\ /xwsqserver.jar;%CROSSWORLDS%\lib\ /spy\lib\spy.jar

Tabla 12. Variables de configuración del shell y del archivo de proceso por lotes (continuación)

Variable	Explicación	Ejemplo
DRIVERLIB	Vía de acceso de las bibliotecas nativas utilizadas por el controlador JDBC	UNIX: DRIVERLIB=\$CROSSWORLDS/lib/db2jdbc.so Windows: DRIVERLIB=%CROSSWORLDS%\bin\db2jdbc.dll

Después de instalar el controlador JDBC y definir valores de configuración en el shell o archivo de proceso por lotes, debe seguir los pasos siguientes para crear objetos comerciales:

1. Inicie el ODA.
2. Inicie el Diseñador de objetos comerciales.
3. En el Diseñador de objetos comerciales, siga el proceso necesario para configurar y ejecutar el ODA.

Este proceso consta de seis pasos, que se describen con detalle en las secciones siguientes.

Inicio de ODA for Manugistics

Puede iniciar ODA for Manugistics con el script de inicio correspondiente al sistema operativo que utilice.

UNIX:

```
start_ManugisticsODA.sh
```

Windows:

```
start_ManugisticsODA.bat
```

Puede configurar y ejecutar ODA for Manugistics utilizando el Diseñador de objetos comerciales. El Diseñador de objetos comerciales localiza cada ODA de acuerdo con el nombre especificado en la variable AGENTNAME de cada script o archivo de proceso por lotes. El nombre de ODA por omisión para este conector es ManugisticsODA.

Ejecución de varias instancias de ODA for Manugistics

Es recomendable que cambie el nombre del ODA cuando ejecute varias instancias de él. Para crear instancias adicionales de ODA for Manugistics con un nombre exclusivo:

- Cree un script o archivo de proceso por lotes separado para cada instancia.
- Especifique un nombre exclusivo en la variable AGENTNAME de cada script o archivo de proceso por lotes.

Si ejecuta instancias de ODA en máquinas diferentes, es recomendable que a cada nombre de ODA anteponga como prefijo el nombre de la máquina de sistema principal.

La Figura 6 en la página 80 muestra la ventana del Diseñador de objetos comerciales desde la que selecciona el ODA para ejecutar.

Utilización de archivos de mensajes error y de rastreo

Los archivos de mensajes de error y de rastreo (el archivo por omisión es BIA_ManugisticsODAAgent.txt) están situados en \ODA\messages\, dentro del directorio del producto. Estos archivos utilizan el convenio de denominación siguiente:

nombreAgenteAgent.txt

Si crea varias instancias del script o archivo de proceso por lotes de ODA y proporciona un nombre exclusivo para cada ODA representado, puede disponer de un archivo de mensajes para cada instancia de ODA. Como alternativa, puede tener varios ODA con nombres diferentes que utilizan el mismo archivo de mensajes. Existen dos maneras de especificar un archivo de mensajes válido:

- Si cambia el nombre de un ODA y no crea un archivo de mensajes para él, debe cambiar el nombre del archivo de mensajes en el Diseñador de objetos comerciales como parte de la configuración de ODA. El Diseñador de objetos comerciales proporciona un nombre para el archivo de mensajes, pero no crea realmente el archivo. Si el archivo mostrado como parte de la configuración de ODA no existe, cambie el valor para que haga referencia a un archivo existente.
- Puede copiar el archivo de mensajes existente para un ODA determinado, y modificarlo según sea necesario. El Diseñador de objetos comerciales da por supuesto que el usuario asigna un nombre a cada archivo de acuerdo con el convenio de denominación. Por ejemplo, si la variable AGENTNAME especifica ManugisticsODA1, la herramienta supone que el nombre del archivo de mensajes asociado es ManugisticsODA1Agent.txt. Por tanto, cuando el Diseñador de objetos comerciales proporciona el nombre de archivo con fines de verificación como parte de la configuración de ODA, el nombre de archivo está basado en el nombre de ODA. Verifique que el archivo de mensajes por omisión tiene el nombre correcto y corríjalo si fuera necesario.

Importante: Si no especifica correctamente el nombre del archivo de mensajes cuando configura el ODA, el ODA se ejecutará sin emitir mensajes. Para obtener más información sobre la especificación del nombre del archivo de mensajes, consulte “Configurar propiedades de inicialización” en la página 80.

Durante el proceso de configuración, debe especificar:

- El nombre del archivo en el que ODA for Manugistics escribe información sobre errores y de rastreo.
- El nivel de rastreo, que está comprendido entre 0 y 5.

La Tabla 13 describe estos valores.

Tabla 13. Niveles de rastreo

Nivel de rastreo	Descripción
0	Registra todos los errores
1	Rastrea todos los mensajes entrantes y salientes emitidos para el método
2	Rastrea las propiedades de ODA y sus valores
3	Rastrea los nombres de todos los objetos comerciales
4	Rastrea detalles de todas las hebras iniciadas

Tabla 13. Niveles de rastreo (continuación)

Nivel de rastreo	Descripción
5	<ul style="list-style-type: none">• Indica los valores de inicialización del ODA para todas las propiedades de éste• Rastrea un estado detallado de cada hebra iniciada por ODA for Manugistics• Rastrea el vuelco de la definición de objeto comercial

Para obtener información sobre dónde configurar estos valores, consulte “Configurar propiedades de inicialización” en la página 80.

Utilización de ODA for Manugistics en el Diseñador de objetos comerciales

Esta sección describe cómo utilizar ODA for Manugistics en el Diseñador de objetos comerciales para crear definiciones de objetos comerciales. Para obtener información sobre el inicio del Diseñador de objetos comerciales, consulte el manual *Business Object Development Guide*. Puede descargar este manual desde el centro de información de IBM WebSphere Business Integration Adapters: <http://www.ibm.com/websphere/integration/wbiadapters/infocenter>

Después de iniciar un ODA, debe iniciar el Diseñador de objetos comerciales para configurar y ejecutar el ODA. Debe realizar seis pasos en el Diseñador de objetos comerciales para crear una definición de objeto comercial utilizando un ODA. El Diseñador de objetos comerciales proporciona un asistente para guiar al usuario en la ejecución de esos pasos.

Después de iniciar el ODA, realice lo siguiente para iniciar el asistente:

1. Abra el Diseñador de objetos comerciales.
2. En el menú Archivo, seleccione el submenú Nuevo utilizando ODA....
El Diseñador de objetos comerciales mostrará la primera ventana del asistente, denominada Seleccionar agente. La Figura 6 en la página 80 muestra esta ventana.

Para seleccionar, configurar y ejecutar el ODA, siga estos pasos:

1. “Seleccionar el ODA”
2. “Configurar propiedades de inicialización” en la página 80
3. “Expandir nodos y seleccionar objetos de base de datos” en la página 82
4. “Confirmar las selecciones de objetos de base de datos” en la página 84
5. “Generar definiciones” en la página 85 y, opcionalmente, “Proporcionar información adicional” en la página 85
6. “Guardar definiciones” en la página 89

Seleccionar el ODA

La Figura 6 muestra el primer recuadro de diálogo que se visualiza en el asistente de seis pasos del Diseñador de objetos comerciales. Desde esta ventana, seleccione el ODA para ejecutar.

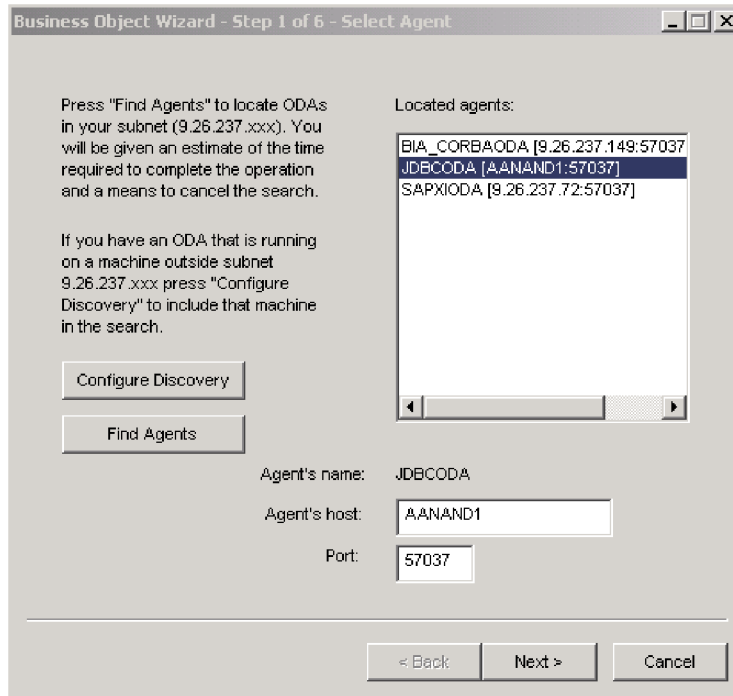


Figura 6. Selección del ODA

Para seleccionar el ODA:

1. Pulse el botón Buscar agentes (Find agents) para visualizar en el campo Agentes localizados todos los ODA registrados o que están en ejecución actualmente.

Nota: Si el Diseñador de objetos comerciales no localiza el ODA deseado, compruebe la configuración del ODA.

2. Seleccione el ODA deseado en la lista mostrada.

El Diseñador de objetos comerciales mostrará en el campo Nombre del agente (Agent's name) la selección hecha por el usuario.

3. Pulse Siguiente (Next).

Configurar propiedades de inicialización

Cuando el Diseñador de objetos comerciales se comunica por primera vez con ODA for Manugistics, solicita al usuario que especifique un conjunto de propiedades de inicialización, tal como muestra la Figura 7. Puede guardar estas propiedades en un perfil designado por un nombre para que no tenga que especificarlas de nuevo cada vez que utilice ODA for Manugistics. Para obtener información sobre la especificación de un perfil de ODA, consulte el manual *Business Object Development Guide*.

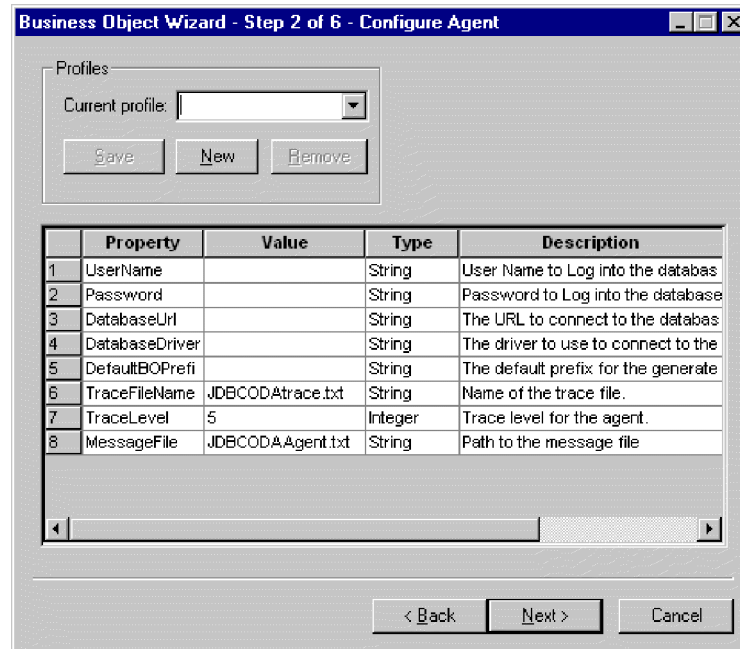


Figura 7. Configuración de propiedades de inicialización del agente

Configure las propiedades de ODA for Manugistics descritas en la Tabla 14.

Tabla 14. Propiedades de ODA for Manugistics

Número de fila	Nombre de propiedad	Tipo de propiedad	Descripción
1	UserName	String	Nombre del usuario con autorización para conectar con la base de datos
2	Password	String	Contraseña del usuario con autorización para conectar con la base de datos
3	DatabaseUrl	String	URL que permite establecer una conexión con la base de datos. Por ejemplo: jdbc:oracle:thin:@MACHINE:1521:SIDNAME
4	DatabaseDriver	String	Nombre del controlador utilizado para establecer la conexión. Por ejemplo: oracle.jdbc.driver.OracleDriver
5	DefaultBOPrefi	String	Texto que se añade como prefijo al nombre del objeto comercial para que sea un nombre exclusivo. Si es necesario, puede cambiar esto más tarde, cuando el Diseñador de objetos comerciales le solicite especificar las propiedades del objeto comercial. Para obtener más información, consulte "Proporcionar información adicional" en la página 85.
6	TraceFileName	String	Archivo en el que ODA for Manugistics escribe información de rastreo. Si el archivo no existe, ODA for Manugistics lo crea en el directorio \ODA\Manugistics. Si el archivo ya existe, ODA for Manugistics añade datos al final del archivo. ODA for Manugistics asigna un nombre al archivo de acuerdo con el convenio de denominación. Por ejemplo, si el agente se denomina ManugisticsODA, ODA for Manugistics crea un archivo de rastreo llamado ManugisticsODATrace.txt. Utilice esta propiedad para especificar un nombre diferente para este archivo.
7	TraceLevel	Integer	Nivel de rastreo habilitado para ODA for Manugistics. Consulte la Tabla 13 en la página 78.

Tabla 14. Propiedades de ODA for Manugistics (continuación)

Número de fila	Nombre de propiedad	Tipo de propiedad	Descripción
8	MessageFile	String	Nombre del archivo de errores y de mensajes. ODA for Manugistics muestra el nombre de archivo de acuerdo con el convenio de denominación. Por ejemplo, si el agente se denomina ManugisticsODA, el valor mostrado en la propiedad correspondiente al archivo de mensajes es ManugisticsODAAgent.txt. Importante: el archivo de errores y de mensajes debe estar situado en el directorio \ODA\messages. Utilice esta propiedad para verificar o especificar un archivo existente.

Importante

Corrija el nombre del archivo de mensajes si el valor por omisión mostrado en el Diseñador de objetos comerciales representa un archivo inexistente. Si el nombre no es correcto cuando salga de este recuadro de diálogo para continuar, el Diseñador de objetos comerciales mostrará un mensaje de error en la ventana desde la que se inició el ODA. Este mensaje no aparece en el Diseñador de objetos comerciales. Si no especifica un archivo de mensajes válido, el ODA se ejecutará sin emitir mensajes.

Expandir nodos y seleccionar objetos de base de datos

Después de que el usuario ha configurado todas las propiedades de inicialización de ODA for Manugistics, el Diseñador de objetos comerciales se conecta a la base de datos especificada y muestra un árbol con todos los nombres de esquema existentes en la base de datos. Estos nombres, que se presentan en forma de nodos del árbol, se pueden expandir. Pulse sobre ellos para visualizar todas las tablas, vistas, procedimientos almacenados y sinónimos/apodos de cada esquema. La Figura 8 muestra este recuadro de diálogo con algún esquema expandido.

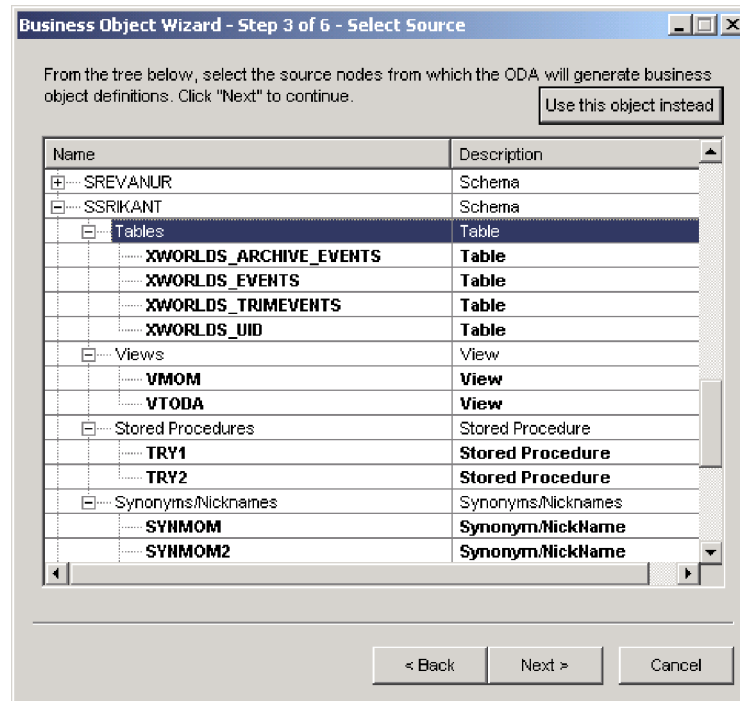


Figura 8. Árbol de esquema con nodos expandidos

Para identificar todos los objetos de base de datos que guardan datos para la definición de objeto comercial generada, seleccione todas las tablas, vistas, procedimientos almacenados y sinónimos/apodos necesarios, y pulse Next (Siguiente). Para obtener información sobre cómo filtrar los objetos devueltos, consulte el manual *Business Object Development Guide*.

El nombre de esquema ALL SCHEMAS se utiliza para facilitar la recuperación de objetos, como tablas y vistas de bases de datos, que no tienen esquemas asociados a los objetos. Cuando expande ALL SCHEMAS, se muestra un árbol con tablas, vistas, procedimientos almacenados y sinónimos/apodos. Cuando se expande cada uno de estos nodos, aparecen todos los objetos pertinentes de la base de datos, para cualquiera que sea el esquema al que pertenezca el objeto.

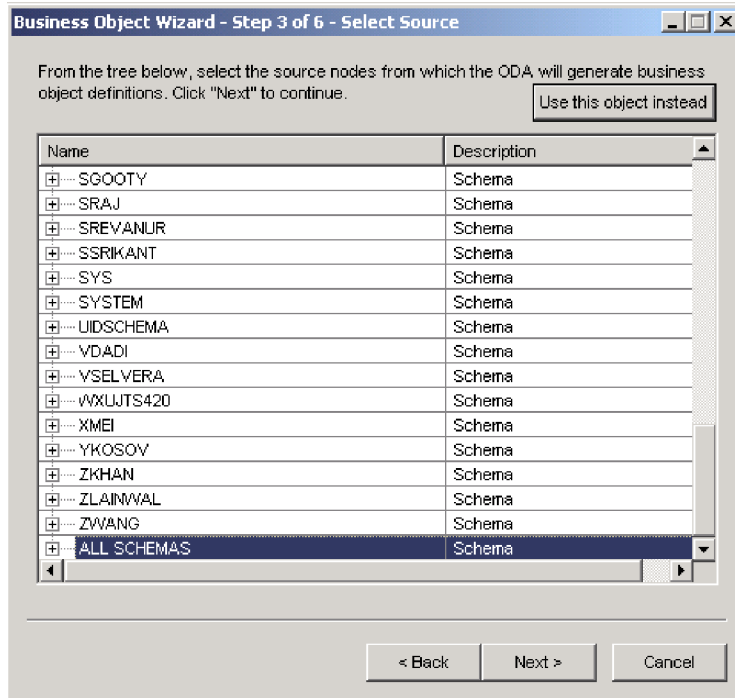


Figura 9. Expansión de ALL SCHEMAS para recuperar objetos

Confirmar las selecciones de objetos de base de datos

Después de que el usuario ha identificado todos los objetos de base de datos que se deben asociar a la definición de objeto comercial generada, el Diseñador de objetos comerciales muestra un recuadro de diálogo con solamente las tablas, vistas, procedimientos almacenados y sinónimos/apodos seleccionados. La Figura 10 muestra este recuadro de diálogo.

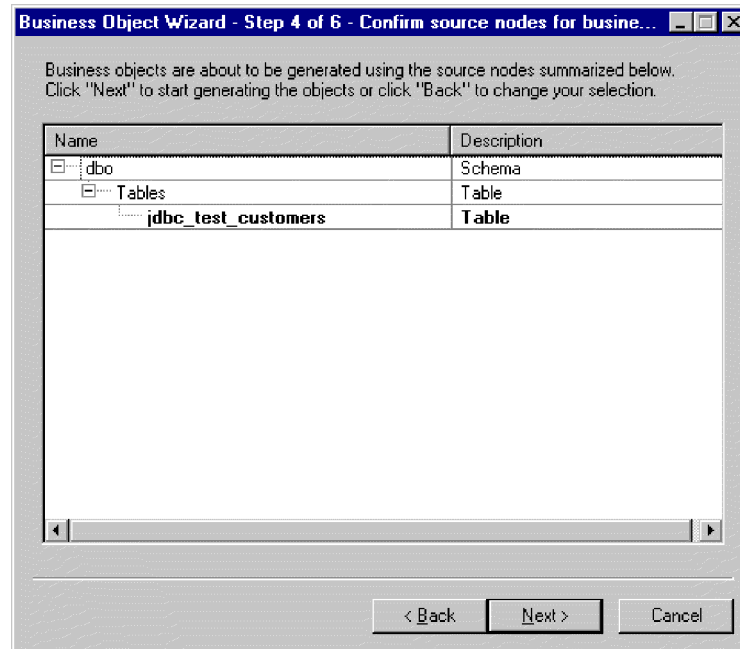


Figura 10. Confirmación de la selección de objetos de base de datos

Esta ventana proporciona las opciones siguientes:

- Para confirmar la selección, pulse Siguiente (Next).
- Si la selección no es correcta, pulse Atrás (Back) para volver a la ventana anterior y hacer los cambios necesarios. Cuando la selección sea correcta, pulse Siguiente (Next).

Generar definiciones

Después de que el usuario ha confirmado los objetos de base de datos, un recuadro de diálogo le informa de que el Diseñador de objetos comerciales está creando las definiciones.

Proporcionar información adicional

Si ODA for Manugistics necesita información adicional, el Diseñador de objetos comerciales muestra la ventana Propiedades de objeto comercial (BO Properties), que solicita la información al usuario.

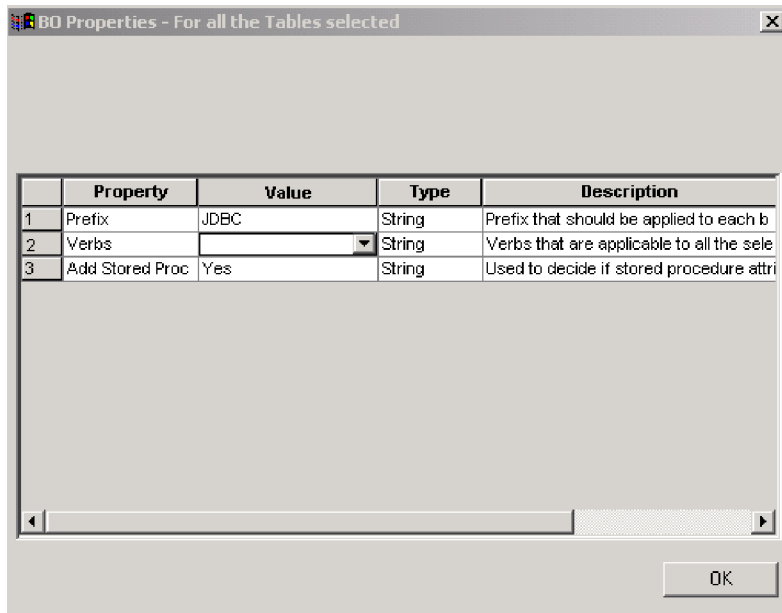


Figura 11. Suministro de información adicional sobre objetos de base de datos

En la ventana Propiedades de objeto comercial, escriba o modifique la información siguiente:

- *Prefijo (Prefix)*: texto que se añade como prefijo al nombre del objeto comercial para hacer que el nombre sea exclusivo. Si está satisfecho con el valor que especificó para la propiedad *DefaultBOPrefix* en la ventana Configurar agente (Figura 7), no es necesario que cambie el valor de este campo.
 - *Verbos (Verbs)*: pulse en el campo *Valor* y seleccione uno o más verbos en el menú emergente. Estos son los verbos admitidos por el objeto comercial.
 - *Añadir procedimiento almacenado (Add Stored Procedure)*: pulse Sí (Yes) o No (No) en el campo Valor:
 - Si selecciona Sí y pulsa Aceptar, ODA for Manugistics muestra una ventana con una lista de todos los atributos de procedimiento almacenado. Seleccione los atributos de procedimiento almacenado que desee añadir al objeto comercial.
 - Seleccione No para asegurarse de que no se añada ningún atributo de procedimiento almacenado a la definición de objeto comercial creada.
- El valor por omisión es Sí.

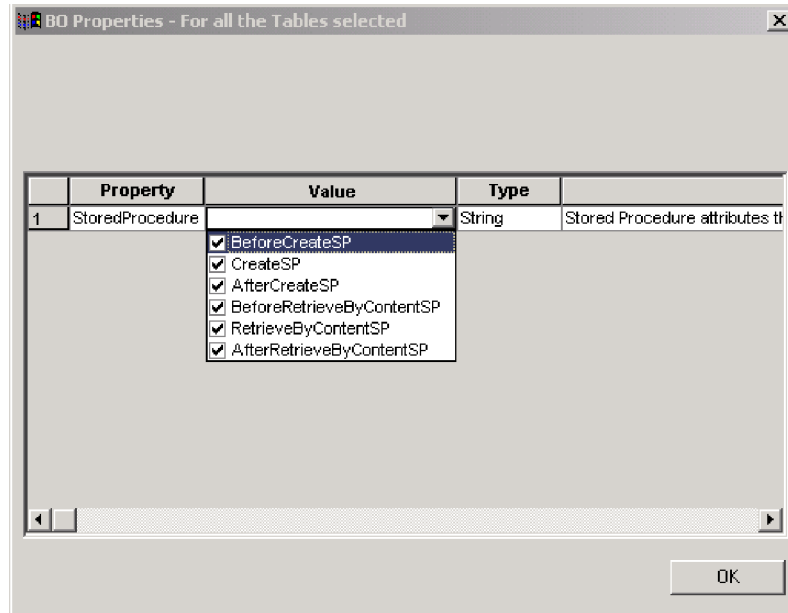


Figura 12. Selección de atributos de procedimiento almacenado

Nota: Si un campo del recuadro de diálogo Propiedades de objeto comercial admite varios valores, el campo aparece en blanco cuando se abre por primera vez el recuadro de diálogo. Pulse en el campo para mostrar una lista desplegable de los valores del campo.

Los atributos de procedimiento almacenado que se deben añadir al objeto comercial se puede asociar a uno de los procedimientos almacenados existentes en ese esquema de la base de datos. Puede seleccionar un procedimiento almacenado en una lista desplegable que muestra todos los procedimientos almacenados de la base de datos pertenecientes a ese esquema, junto a cada atributo de procedimiento almacenado. Esta información producirá la información específica de la aplicación necesaria para ese atributo.

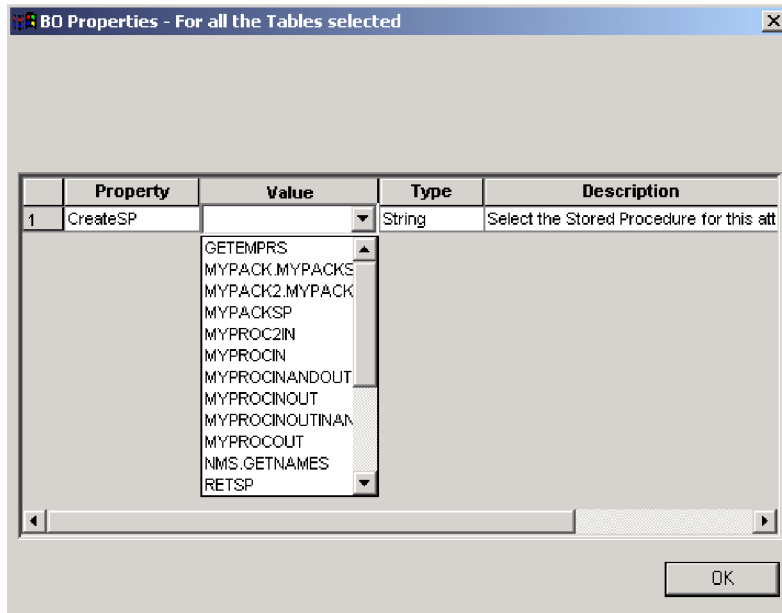


Figura 13. Asociación de procedimientos almacenados con atributos de procedimiento almacenado

La información específica de la aplicación para el nivel de objeto tiene este aspecto: TN=nombreTabla

La información específica de la aplicación para el nivel de atributo tiene este aspecto: CN=nombreColumna

Si se crea un objeto comercial a partir de un procedimiento almacenado y los atributos de procedimiento almacenado del ODA for Manugistics Adapter (tales como SPForCreate) están asociados al objeto, el ODA proporciona una lista de todos los procedimientos almacenados de ese esquema, junto con los atributos del procedimiento almacenado. Este ODA le permite asociar el procedimiento almacenado necesario con el objeto comercial. Esto genera la información específica de la aplicación para el atributo de procedimiento almacenado del ODA for Manugistics Adapter, del modo siguiente:

SPN=stored procedure Name; IN=a1:a2; OUT=b1:b2; IO=c1:c2

Donde IN significa que el parámetro del procedimiento almacenado es de tipo INPUT, OUT significa que el parámetro es de tipo OUTPUT, y IO significa que es de tipo INPUT/OUTPUT. El ODA no establece RS en "true" o "false" en la información específica de la aplicación, por lo que el usuario debe definir manualmente ese valor.

Los verbos añadidos al objeto comercial son los verbos estándar, esencialmente Retrieve, RetrieveByContent, Create, Update y Delete.

Si el parámetro de retorno del procedimiento almacenado es de tipo ResultSet, el ODA analiza el conjunto de resultados y crea un objeto comercial, convirtiendo las columnas del conjunto de resultados en atributos del objeto comercial. La información específica de la aplicación para las columnas del procedimiento almacenado se definirá como CN=StoredProcedureColumnName. El ODA define los atributos de clave basándose en la información de metadatos JDBC devuelta por el controlador. Si no se devuelve ninguna información, el ODA no marca por omisión

ningún atributo como claves. Todos los demás atributos, como la longitud y el tipo, se definen de la misma manera que para los atributos generados a partir de tablas.

Guardar definiciones

Después de que el usuario proporcione toda la información necesaria en el recuadro de diálogo Propiedades de objeto comercial y pulse Aceptar (OK), el Diseñador de objetos comerciales muestra el recuadro de diálogo final del asistente. En este recuadro de diálogo puede guardar la definición en el servidor o en un archivo, o puede abrir la definición para editarla en el Diseñador de objetos comerciales. Para obtener más información y hacer modificaciones adicionales, consulte el manual *Business Object Development Guide*.

Contenido de la definición generada

La definición de objeto comercial creada por ODA for Manugistics contiene:

- Un atributo para cada columna contenida en las tablas de base de datos, vistas, procedimientos almacenados y sinónimos/apodos especificados.
- Los verbos especificados en la ventana Propiedades de objeto comercial
- Información específica de la aplicación:
 - A nivel de objeto comercial
 - Para cada atributo
 - Para cada verbo

Esta sección trata los temas siguientes:

- “Propiedades a nivel de objeto comercial”
- “Propiedades de atributo” en la página 90
- “Verbos” en la página 92

Propiedades a nivel de objeto comercial

ODA for Manugistics crea la información siguiente a nivel de objeto comercial:

- Nombre del objeto comercial
- Versión: toma el valor 1.0.0 por omisión
- Información específica de la aplicación

La información específica de la aplicación a nivel de objeto comercial le permite:

- Especificar el nombre de la correspondiente tabla de base de datos
- Proporcionar la información necesaria para realizar una supresión física o lógica

A nivel de objeto comercial, la información específica de la aplicación consta de parámetros separados por un signo de punto y coma (;) que actúa como delimitador. El nombre del parámetro y su valor están separados por un signo de dos puntos (:) que actúa como delimitador. La sintaxis es:

TN=nombreTabla; SCN=nombreColumnaEstado:valorEstado

donde *nombreTabla* identifica la tabla de base de datos, *nombreColumnaEstado* es el nombre de la columna de base de datos utilizada para realizar supresiones lógicas y *valorEstado* es el valor que denota que un objeto comercial está inactivo o suprimido.

La información específica de la aplicación (AppSpecificInfo) que ODA for Manugistics crea a este nivel contiene un valor solamente para el nombre de la tabla de base de datos, vista, procedimiento almacenado o sinónimo/apodo. Para obtener información sobre la especificación de un valor para la columna de estado, consulte “Información específica de la aplicación a nivel de objeto comercial” en la página 63.

Propiedades de atributo

Esta sección describe las propiedades que ODA for Manugistics crea para cada atributo. Para obtener más información sobre los atributos, consulte “Propiedades de atributos de objetos comerciales” en la página 60.

La propiedad Name

ODA for Manugistics obtiene el valor del nombre del atributo a partir del nombre de columna contenido en la tabla de base de datos, vista, procedimiento almacenado o sinónimo/apodo.

La propiedad Type (tipo de datos)

Cuando ODA for Manugistics define el tipo de un atributo, convierte el tipo de datos de una columna contenida en la tabla o vista en el correspondiente tipo de objeto comercial de IBM WebSphere Business Integration Adapter. Esta conversión se realiza en dos pasos. Primero, el tipo de datos contenido en la base de datos se convierte en un tipo JDBC. A continuación, el tipo JDBC se convierte en un tipo de objeto comercial de IBM WebSphere Business Integration Adapter. La primera conversión es realizada por el controlador JDBC que está utilizando. Consulte la especificación de JDBC (2.0 y superior) para conocer detalles sobre la asignación de tipos de base de datos a un tipo JDBC. La Tabla 14 muestra la conversión del tipo JDBC al correspondiente tipo de objeto comercial de IBM WebSphere Business Integration Adapter.

Tabla 15. Correspondencia de tipos de datos

Tipo JDBC	Tipo de objeto comercial de WebSphere Business Integration Adapter
BIT	BOOLEAN
CHAR	STRING
VARCHAR	STRING
LONGVARCHAR	STRING
INTEGER	INTEGER
NUMERIC	INTEGER
SMALLINT	INTEGER
TINYINT	INTEGER
BIGINT	INTEGER
DATE	DATE
TIME	DATE
TIMESTAMP	DATE
DECIMAL	STRING
DOUBLE	DOUBLE
FLOAT	DOUBLE
REAL	FLOAT

Tabla 15. Correspondencia de tipos de datos (continuación)

Tipo JDBC	Tipo de objeto comercial de WebSphere Business Integration Adapter
BINARY	STRING, añadir BYTEARRAY=TRUE a AppSpecificInfo
VARBINARY	STRING, añadir BYTEARRAY=TRUE a AppSpecificInfo

Nota: Si el tipo de datos de una columna no es ninguno de los mostrados en la Tabla 15 en la página 90, ODA for Manugistics pasa por alto la columna y muestra un mensaje para indicar que la columna no se puede procesar.

La propiedad Cardinality

ODA for Manugistics establece en 1 la cardinalidad de todos los atributos simples.

La propiedad MaxLength

ODA for Manugistics obtiene la longitud de una serie de caracteres a partir de la longitud especificada para el tipo de datos varchar, char o text.

La propiedad IsKey

Si la columna es una clave primaria en la tabla, ODA for Manugistics la marca como atributo de clave. En cambio, si como nodo de origen para generar objetos comerciales se selecciona una vista, procedimiento almacenado o sinónimo/apodo en lugar de una tabla, ODA for Manugistics no marca la columna como atributo de clave. En este caso, el atributo de clave se debe definir manualmente.

La propiedad IsForeignKey

ODA for Manugistics no define la propiedad IsForeignKey. El usuario puede definirla en el Diseñador de objetos comerciales.

La propiedad IsRequired

Si un campo está identificado como not null en la tabla, vista, procedimiento almacenado o sinónimo/apodo, ODA for Manugistics lo marca como atributo necesario. En cambio, ODA for Manugistics no marca el campo de clave como necesario, pues podría existir una secuencia asociada a él, o podría ser una columna de identidad.

La propiedad AppSpecificInfo

ODA for Manugistics incluye dos parámetros para la propiedad AppSpecificInfo a nivel de atributo. La sintaxis de los parámetros especificados es:

- *CN=nombreColumna*

donde nombreColumna es el nombre de la columna de la tabla de base de datos, vista, procedimiento almacenado o sinónimo/apodo correspondiente al atributo específico.

- BYTEARRAY=*true|false*

ODA for Manugistics reconoce las columnas con datos binarios y crea un atributo de tipo String con una propiedad BYTEARRAY=true de AppSpecificInfo.

Nota: El usuario puede definir parámetros AppSpecificInfo adicionales en el Diseñador de objetos comerciales. Para obtener información sobre estos parámetros, consulte “Información específica de la aplicación a nivel de atributo” en la página 64.

Verbos

ODA for Manugistics genera los verbos especificados en la ventana Propiedades de objeto comercial. Crea una propiedad AppSpecificInfo para cada verbo, pero no asigna un valor a la propiedad. Para obtener más información, consulte “Formato de la información específica de la aplicación para verbos” en la página 72.

Archivo de definición de objeto comercial de ejemplo

A continuación sigue una definición de objeto comercial de ejemplo:

```
[BusinessObjectDefinition]
Name = CUSTOMER
Version = 1.0.0
AppSpecificInfo = TN=ra_customers;SCN=

    [Attribute]
    Name = customer_id
    Type = Integer
    Cardinality = 1
    MaxLength = 0
    IsKey = true
    IsForeignKey = false
    IsRequired = false
    AppSpecificInfo = CN=customer_id
    DefaultValue =
    [End]

    ***** Otros atributos *****

[Attribute]
Name = ObjectEventId
Type = String
Cardinality = 1
MaxLength = 0
IsKey = false
IsForeignKey = false
IsRequired = false
AppSpecificInfo =
DefaultValue =
[End]

[Verb]
Name = Delete
AppSpecificInfo =
[End]

[Verb]
Name = Update
AppSpecificInfo =
[End]
```

```
[Verb]
Name = Create
AppSpecificInfo =
[End]

[Verb]
Name = Retrieve
AppSpecificInfo =
[End]

[End]
```

Inserción de atributos que contienen objetos comerciales hijo

Utilice el Diseñador de objetos comerciales para insertar atributos que representan objetos comerciales hijo de cardinalidad simple o múltiple. Para obtener más información, consulte el manual *Business Object Development Guide*.

Adición de información a la definición de objeto comercial

Debido a que las tablas de base de datos, vistas, procedimientos almacenados y sinónimos/apodos pueden no tener toda la información que una definición de objeto comercial necesita, puede ser necesario añadir información a la definición de objeto comercial creada por ODA for Manugistics. Para obtener más información, consulte el Capítulo 3, “Descripción de los objetos comerciales en su relación con el conector”, en la página 37.

Para examinar la definición de objeto comercial o añadir información, puede utilizar el Diseñador de objetos comerciales o un editor de texto. Para volver a cargar una definición revisada en el depósito de IBM WebSphere Business Integration Adapter, puede utilizar el Diseñador de objetos comerciales o, si el intermediario de integración es InterChange Server, el mandato `repos_copy`.

Capítulo 5. Resolución y manejo de errores

Este capítulo describe los problemas que el usuario puede encontrar al iniciar o ejecutar el conector para JDBC. Este capítulo contiene las secciones siguientes:

- “Problemas de inicio”
- “Proceso de sucesos”
- “Correlación (solamente para el intermediario de integración InterChange Server)”
- “Manejo y registro de errores” en la página 97
- “Pérdida de la conexión con la aplicación” en la página 98
- “Error de recurso ocupado” en la página 99
- “Funcionamiento incorrecto de ODA for Manugistics debido a un controlador JDBC no compatible” en la página 99

Problemas de inicio

Si encuentra dificultades al intentar iniciar el conector, compruebe que el intermediario de integración esté iniciado y en ejecución.

Proceso de sucesos

Si la tabla de sucesos contiene sucesos que no son procesados mientras el conector está en ejecución, asegúrese de que se cumplan estas condiciones:

- El proceso comercial pertinente está en ejecución.
- El nombre del objeto comercial contenido en la tabla de sucesos coincide con el nombre del objeto comercial especificado para el puerto de proceso comercial.

Correlación (solamente para el intermediario de integración InterChange Server)

Esta sección trata los temas siguientes:

- “Problemas de correlación”
- “Conversión de fechas”

Problemas de correlación

Si no se produce la correlación de objetos comerciales o no se invoca el proceso de correlación, asegúrese de que se hayan instalado las correlaciones en el directorio correcto.

Conversión de fechas

Nota: Este procedimiento de conversión de fechas solo es aplicable a las versiones del conector anteriores a la versión 1.5.0.

Utilice correlaciones para convertir los datos almacenados en la base de datos con formato Date al formato String utilizado por los objetos comerciales de WebSphere Business Integration Adapter.

Por ejemplo, la fecha siguiente está almacenada en una base de datos Oracle:

Sun Jan 01 00:00:00 CEST 1999

y desea convertirla a la siguiente serie de caracteres (String), la cual se procesa en un objeto comercial de WebSphere Business Integration Adapter para JDBC:

Jan 01 1999 00:00:00

Para efectuar esta conversión, utilice los constructores `DtpDate()` y `DtpSplitString()` definidos para la transformación de datos en la correlación. Para conocer la sintaxis y una descripción de estos constructores y las clases cuyos objetos son construidos por ellos, consulte el manual *Map Development Guide*.

Para utilizar una correlación para convertir el valor `Date` en un valor `String`, siga estos pasos:

1. Utilice `DtpSplitString()` con un espacio como delimitador para descomponer la serie de caracteres en seis fragmentos y redistribirlos según un orden que `DtpDate` pueda utilizar. Para convertir la fecha de ejemplo, utilice:

```
DtpSplitString OurSplitString = new DtpSplitString
("Sun Jan 01 00:00:00 CEST 1999", " ");
```

En la sentencia anterior, `OurSplitString` es una variable definida por el usuario, de tipo `DtpSplitString`, y se ha especificado un espacio en blanco como delimitador.

2. Utilice el método `nextElement()` de la clase `DtpSplitString` para ejecutar un proceso iterativo sobre la recién creada variable `OurSplitString`, colocando cada uno de los seis elementos de la variable en una matriz cuyos elementos son de tipo `String`. El ejemplo siguiente especifica `OurStringPieces` como matriz de salida:

```
String[] OurStringPieces = new String[6];
for (i=0;i<=5;i=i+1){
    OurStringPieces[i]=OurSplitString.nextElement();
}
```

Este proceso iterativo produce los elementos de matriz siguientes:

```
OurStringPieces[0] = Sun
OurStringPieces[1] = Jan
OurStringPieces[2] = 01
OurStringPieces[3] = 00:00:00
OurStringPieces[4] = CEST
OurStringPieces[5] = 1999
```

3. Concatene los fragmentos de la serie de caracteres necesarios para utilizarlos como entrada de `DtpDate`. La conversión del ejemplo utiliza "M D A h:m:s" como formato de entrada para `DtpDate`, lo cual necesita que la serie de caracteres convertida tenga este aspecto: "Jan 01 1999 00:00:00". En este ejemplo, `String` utiliza los elementos 1, 2, 5 y 3 de la matriz `OurStringPieces`:

```
OurConcatenatedString =
OurStringPieces[1]+OurStringPieces[2]+OurStringPieces[5]+OurStringPieces[3];
```

4. Utilice la nueva serie de caracteres concatenada como entrada para `DtpDate`:
- ```
DtpDate OurDtpDate = new DtpDate(OurConcatenatedString,"M D A h:m:s");
```

Una vez colocado el valor `Date` en el formato `DtpDate`, está preparado para trabajar con la fecha en la correlación.

---

## Manejo y registro de errores

El conector registra un mensaje de error cada vez que encuentra una condición que hace que falle el proceso que el conector ejecuta actualmente para un objeto comercial y un verbo. Cuando se produce un error de esta clase, el conector también escribe una representación textual del objeto comercial fallido tal como se recibió el objeto. El conector escribe el texto en el archivo de anotaciones cronológicas del conector o en la corriente de salida estándar, dependiendo de la configuración del conector. El usuario puede utilizar este texto como ayuda para determinar la fuente del error.

### Tipos de error

La Tabla 16 describe los tipos de mensajes de rastreo que el conector emite para cada nivel de rastreo. Estos mensajes complementan a los mensajes de rastreo emitidos por la arquitectura IBM WebSphere Business Integration Adapter, tal como el envoltorio de ejecución del conector Java y la interfaz de mensajes de WebSphere MQ.

*Tabla 16. Mensajes de rastreo del conector*

| Nivel de rastreo | Mensajes de rastreo                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nivel 0          | Mensaje que identifica la versión del conector. No se realiza ningún otro rastreo a este nivel. Este es el valor por omisión.                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Nivel 1          | <ul style="list-style-type: none"><li>• Mensajes de estado</li><li>• Mensajes que proporcionan información (clave) de identificación para cada objeto comercial procesado</li><li>• Mensajes que se entregan cada vez que se ejecuta el método <code>pollForEvents</code></li></ul>                                                                                                                                                                                                                                                                                                   |
| Nivel 2          | <ul style="list-style-type: none"><li>• Mensajes del gestor de objetos comerciales que contienen información tal como las matrices y objetos comerciales hijo que el conector encuentra o recupera durante el proceso de un objeto comercial</li><li>• Mensajes que se registran cada vez que se envía un objeto comercial al intermediario de integración, ya sea desde <code>gotApplEvent()</code> o desde <code>executeCollaboration()</code></li><li>• Mensajes que indican que se ha recibido un objeto comercial en forma de petición de intermediario de integración</li></ul> |
| Nivel 3          | <ul style="list-style-type: none"><li>• Mensajes de proceso de claves foráneas, que contienen información tal como la indicación de cuándo el conector ha encontrado o definido una clave foránea en un objeto comercial</li><li>• Mensajes que proporcionan información sobre el proceso de objetos comerciales. Por ejemplo, estos mensajes se entregan cuando el conector detecta una coincidencia entre objetos comerciales, o cuando encuentra un objeto comercial en una matriz de objetos comerciales hijo</li></ul>                                                           |
| Nivel 4          | <ul style="list-style-type: none"><li>• Mensajes de la información específica de la aplicación, como los mensajes que muestran los valores devueltos por las funciones que analizan los campos de la información específica de la aplicación del objeto comercial</li><li>• Mensajes que indican cuándo el conector entra o sale de una función, lo cual ayuda a determinar el flujo de proceso del conector</li><li>• Todos los mensajes específicos de las hebras. Si el conector crea varias hebras, aparece un mensaje para la creación de cada nueva hebra</li></ul>             |

Tabla 16. Mensajes de rastreo del conector (continuación)

| Nivel de rastreo | Mensajes de rastreo                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nivel 5          | <ul style="list-style-type: none"><li>• Mensajes que denotan la inicialización del conector, como los mensajes que muestran el valor de cada propiedad de configuración obtenida del intermediario de integración</li><li>• Mensajes que incluyen sentencias ejecutadas en la aplicación. A este nivel de rastreo, el archivo de anotaciones cronológicas del conector contiene todas las sentencias ejecutadas en la aplicación de destino y el valor de todas las variables que se sustituyen.</li><li>• Mensajes que comprenden una representación de un objeto comercial antes de que el conector comience a procesarlo (mostrando el estado que tiene el objeto cuando lo recibe el conector) y la representación del objeto después de que el conector ha terminado su proceso (mostrando el estado que tiene el objeto cuando lo devuelve el conector)</li><li>• Mensajes que comprenden un vuelco de objeto comercial</li><li>• Mensajes que indican el estado de cada hebra creada por el conector mientras está en ejecución</li></ul> |

## Mensajes de error

### Archivo de mensajes del conector

Todos los mensajes de error creados por el conector se guardan en un archivo de mensajes llamado `JDBCCConnector.txt` o `JDBCCConnector_II_TT.txt` (donde *II* especifica un idioma, y *TT* especifica un país o territorio). Cada error tiene un número de error seguido por el mensaje de error. Por ejemplo:

```
20017
Connector Infrastructure version does not match.
20018
Connection from {1} to the Application is lost! Please enter 'q'
to stop the connector, then restart it after the problem is
fixed.
20019
Error: ev_id is NULL in pollForEvent().
```

## Pérdida de la conexión con la aplicación

Si el conector no logra establecer conexión, envía el código FAIL al intermediario de integración y concluye su ejecución.

Cuando `AutoCommit` tiene el valor `false` y falla la operación `PingQuery`, el conector intenta crear una nueva conexión con la base de datos. Si el conector logra establecer una nueva conexión con la base de datos, continuará el proceso; en caso contrario, el conector devuelve el código `APPRESPONSETIMEOUT`, que hace que concluya la ejecución del conector.

## Error de búsqueda fuera de secuencia

La propiedad `AutoCommit` debe tener el valor `"false"` cuando se utiliza la base de datos Oracle versión 8.0 y 8.1 con Sun Solaris, u Oracle 8.1 con Windows 2000. De lo contrario, obtendrá mensajes de error `ORA-01002` (búsqueda fuera de secuencia). Este error no se produce en versiones anteriores de bases de datos Oracle. Establecer `AutoCommit` en `"false"` mejora el rendimiento del sistema.

---

## Error de recurso ocupado

**Nota:** Este conector solamente encuentra este error cuando se ejecuta en una base de datos Oracle.

Algunas veces el conector encuentra un error como el siguiente al recuperar o cambiar datos en una aplicación:

```
[Time: 2001/05/29 16:30:07.356] [System: ConnectorAgent] [SS: SOVTConnector]
[Type: Trace] [Mesg: Select CLIENT,COUNTRY,STRT_CODE,CITY_CODE,CITYP_CODE,
STRTYPEAB,COMMU_CODE,REGIOGROUP,TAXJURCODE from ADRSTREET where CLIENT='100'
and COUNTRY='DE' and STRT_CODE='000001114136' FOR UPDATE NOWAIT]
[Time: 2001/05/29 16:30:07.526] [System: ConnectorAgent] [SS: SOVTConnector]
[Type: Trace] [Mesg: :logMsg]
[Time: 2001/05/29 16:30:07.536] [System: ConnectorAgent] [SS: SOVTConnector]
[Type: Error] [MsgID: 37002]
[Mesg: Execution of Retrieve statement failed : java.
sql.SQLException: ORA-00054: Versuch, mit NOWAIT eine bereits
belegte Ressourcenzufordern.]
```

Este error se produce cuando el conector intenta actualizar un registro que está bloqueado actualmente. El registro puede estar bloqueado por otro proceso o, debido a que el conector es de ejecución paralela, el registro puede estar bloqueado por el propio conector.

Observe que los registros deben estar bloqueados durante el proceso de actualización. El conector intenta recuperar una imagen posterior del objeto recibido por el intermediario de integración y, como consecuencia, bloquea el objeto completo en la base de datos para preservar la integridad de los datos.

Para resolver este problema, puede detener el proceso que está impidiendo que el conector obtenga un bloqueo sobre el registro, o puede ajustar la propiedad de configuración `RetryCountInterval` del conector.

---

## Funcionamiento incorrecto de ODA for Manugistics debido a un controlador JDBC no compatible

Si el controlador JDBC no es compatible con una función de ODA for Manugistics, el agente de descubrimiento de objetos (ODA) no funciona debidamente. Por ejemplo, si el controlador no puede trabajar con todas las llamadas de método utilizadas por ODA for Manugistics, el archivo de anotaciones cronológicas de ODA denota el proceso que ha fallado. Lo siguiente es un ejemplo del contenido del archivo de anotaciones cronológicas:

```
[Time: 2002/05/15 17:00:55.147] [System: Object Discovery Agent] [SS: null]
[Type: 6] [Mesg: A SQL Error occurred in getting Schema Names from Database.
Reason [nombreProducto] [ODBC nombreProducto Driver] Optional feature not
implemented]
```

En este caso, debe utilizar un controlador JDBC diferente.

---

## Manejo de errores cuando se utiliza IGP

Cuando el procedimiento almacenado definido en la tabla de interfaz ejecuta una operación de Inserción/Actualización, los errores se colocan en la correspondiente tabla de errores. No es posible la notificación directa de errores para las operaciones de inserción y actualización realizadas mediante las tablas de interfaz.

El ingeniero de integración deberá crear colaboraciones o flujos de trabajo para conocer el resultado de estos tipos de operaciones mediante el examen de las tablas de errores (ERR) apropiadas. Estos errores también se podrían gestionar habilitando la notificación de sucesos para las tablas de errores.

---

## Apéndice A. Propiedades de configuración estándar para conectores

Este apéndice describe las propiedades de configuración estándar del componente conector de los adaptadores WebSphere Business Integration. La información de este apéndice trata sobre conectores que se ejecutan en los intermediarios de integración siguientes:

- WebSphere InterChange Server (ICS)
- WebSphere MQ Integrator, WebSphere MQ Integrator Broker y WebSphere Business Integration Message Broker se designan colectivamente como WebSphere Message Brokers (WMQI)
- WebSphere Application Server (WAS)

No todos los conectores hacen uso de todas estas propiedades estándar. Cuando selecciona un intermediario de integración desde el Configurador de conectores, obtiene una lista de las propiedades estándar que es necesario configurar para el adaptador que se ejecuta con ese intermediario.

Para obtener información sobre propiedades específicas del conector, consulte la correspondiente guía del usuario del adaptador.

**Nota:** En el presente documento, las barras inclinadas invertidas (\) se utilizan como convenio para representar las vías de acceso de directorios. Para las instalaciones en UNIX, utilice barras inclinadas (/) en lugar de barras inclinadas invertidas y siga los convenios de cada sistema operativo.

---

### Propiedades nuevas y propiedades suprimidas

En el presente release se han añadido las propiedades estándar siguientes.

#### Propiedades nuevas

- XMLNameSpaceFormat

#### Propiedades suprimidas

- RestartCount

---

### Configuración de propiedades estándar de conector

Los conectores de adaptador tienen dos tipos de propiedades de configuración:

- Propiedades de configuración estándar
- Propiedades de configuración específicas del conector

Esta sección describe las propiedades de configuración estándar. Para obtener información sobre propiedades específicas de un conector, consulte la guía del usuario del adaptador respectivo.

## Utilización del Configurador de conectores

Para configurar propiedades de conector utilice el Configurador de conectores, al cual accede desde el Gestor del sistema. Para obtener más información sobre la utilización del Configurador de conectores, consulte el apéndice referente al Configurador de conectores.

**Nota:** El Configurador de conectores y el Gestor del sistema se ejecutan en el sistema Windows solamente. Si está ejecutando el conector en un sistema UNIX, debe tener una máquina Windows con esas herramientas instaladas. Para configurar propiedades de conector para un conector que se ejecuta en UNIX, debe iniciar el Gestor del sistema en la máquina Windows, conectar con el intermediario de integración de UNIX e iniciar el Configurador de conectores para el conector.

## Definición y actualización de valores de propiedades

La longitud por omisión de un campo de propiedad es 255 caracteres.

El conector utiliza el orden siguiente para determinar el valor de una propiedad (donde el número más alto prevalece sobre los demás valores):

1. Valor por omisión
2. Depósito (sólo si el intermediario de integración es WebSphere InterChange Server)
3. Archivo de configuración local
4. Línea de mandatos

Un conector obtiene sus valores de configuración cuando se inicia. Si cambia el valor de una o más propiedades de conector durante una sesión de ejecución, el **Método de actualización** de la propiedad determina cómo entra en vigor el cambio. Existen cuatro métodos de actualización diferentes para propiedades de conector estándar:

- **Dinámico**  
El cambio entra en vigor inmediatamente después de guardarlo en el Gestor del sistema. Si el conector está trabajando en la modalidad autónoma (independientemente del Gestor del sistema), por ejemplo con uno de los intermediarios de mensajes WebSphere, el usuario sólo puede cambiar propiedades mediante el archivo de configuración. En este caso no es posible una actualización dinámica.
- **Reinicio de agente (solamente para ICS)**  
El cambio entra en vigor sólo después de detener y reiniciar el componente específico de la aplicación.
- **Reinicio de componente**  
El cambio entra en vigor sólo después de detener y luego reiniciar el conector desde el Gestor del sistema. No es necesario detener y reiniciar el componente específico de la aplicación ni el intermediario de integración.
- **Reinicio de servidor**  
El cambio entra en vigor sólo después de detener y reiniciar el componente específico de la aplicación y el intermediario de integración.

Para determinar cómo se actualiza una propiedad determinada, consulte la columna **Método de actualización** en la ventana del Configurador de conectores, o consulte la columna Método de actualización de la Tabla 17 en la página 103, más abajo.



## Resumen de las propiedades estándar

La Tabla 17 proporciona una guía de consulta rápida para las propiedades de configuración estándar de un conector. No todos los conectores hacen uso de todas estas propiedades, y los valores de las propiedades pueden diferir según el intermediario de integración utilizado, pues las dependencias de las propiedades estándar están basadas en `RepositoryDirectory`.

Debe definir los valores de algunas de estas propiedades antes de ejecutar el conector. Consulte la sección siguiente para obtener una explicación de cada propiedad.

**Nota:** En la columna "Notas" de la Tabla 17, la frase "El directorio de depósito es REMOTE" indica que el intermediario es InterChange Server. Cuando el intermediario es WMQI o WAS, el directorio de depósito está definido como LOCAL.

Tabla 17. Resumen de las propiedades de configuración estándar

| Nombre de propiedad           | Valores posibles                                                                                                                              | Valor por omisión                               | Método de actualización | Notas                                                                                                 |
|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------|-------------------------|-------------------------------------------------------------------------------------------------------|
| AdminInQueue                  | Nombre de cola válido de JMS                                                                                                                  | <code>NOMBRECONECTOR/ADMININQUEUE</code>        | Reinicio de componente  | El transporte de entrega es JMS                                                                       |
| AdminOutQueue                 | Nombre de cola válido de JMS                                                                                                                  | <code>NOMBRECONECTOR/ADMINOUTQUEUE</code>       | Reinicio de componente  | El transporte de entrega es JMS                                                                       |
| AgentConnections              | 1-4                                                                                                                                           | 1                                               | Reinicio de componente  | El transporte de entrega es MQ o IDL: El directorio de depósito es <REMOTE> (el intermediario es ICS) |
| AgentTraceLevel               | 0-5                                                                                                                                           | 0                                               | Dinámico                |                                                                                                       |
| ApplicationName               | Nombre de aplicación                                                                                                                          | Valor para el nombre de aplicación del conector | Reinicio de componente  |                                                                                                       |
| BrokerType                    | ICS, WMQI, WAS                                                                                                                                |                                                 | Reinicio de componente  |                                                                                                       |
| CharacterEncoding             | ascii7, ascii8, SJIS, Cp949, GBK, Big5, Cp297, Cp273, Cp280, Cp284, Cp037, Cp437<br><b>Nota:</b> Esto es un subconjunto de valores admitidos. | ascii7                                          | Reinicio de componente  |                                                                                                       |
| ConcurrentEventTriggeredFlows | De 1 a 32.767                                                                                                                                 | 1                                               | Reinicio de componente  | El directorio de depósito es <REMOTE> (el intermediario es ICS)                                       |
| ContainerManagedEvents        | Ningún valor o JMS                                                                                                                            | Ningún valor                                    | Reinicio de componente  | El transporte de entrega es JMS                                                                       |

Tabla 17. Resumen de las propiedades de configuración estándar (continuación)

| Nombre de propiedad           | Valores posibles                                                                                                        | Valor por omisión                         | Método de actualización | Notas                                                                     |
|-------------------------------|-------------------------------------------------------------------------------------------------------------------------|-------------------------------------------|-------------------------|---------------------------------------------------------------------------|
| ControllerStoreAndForwardMode | true o false                                                                                                            | true                                      | Dinámico                | El directorio de depósito es <REMOTE> (el intermediario es ICS)           |
| ControllerTraceLevel          | 0-5                                                                                                                     | 0                                         | Dinámico                | El directorio de depósito es <REMOTE> (el intermediario es ICS)           |
| DeliveryQueue                 |                                                                                                                         | NOMBRECONECTOR/DELIVERYQUEUE              | Reinicio de componente  | Transporte JMS solamente                                                  |
| DeliveryTransport             | MQ, IDL o JMS                                                                                                           | JMS                                       | Reinicio de componente  | Si el directorio de depósito es local, entonces el valor es JMS solamente |
| DuplicateEventElimination     | true o false                                                                                                            | false                                     | Reinicio de componente  | Transporte JMS solamente: CME debe ser <NONE>                             |
| FaultQueue                    |                                                                                                                         | NOMBRECONECTOR/FAULTQUEUE                 | Reinicio de componente  | Transporte JMS solamente                                                  |
| jms.FactoryClassName          | CxCommon.Messaging.jms.IBMMQSeriesFactory o CxCommon.Messaging.jms.SonicMQFactory o cualquier nombre de clase Java      | CxCommon.Messaging.jms.IBMMQSeriesFactory | Reinicio de componente  | Transporte JMS solamente                                                  |
| jms.MessageBrokerName         | Si FactoryClassName es IBM, utilice crossworlds.queue.manager.<br>Si FactoryClassName es Sonic, utilice localhost:2506. | crossworlds.queue.manager                 | Reinicio de componente  | Transporte JMS solamente                                                  |
| jms.NumConcurrentRequests     | Entero positivo                                                                                                         | 10                                        | Reinicio de componente  | Transporte JMS solamente                                                  |
| jms.Password                  | Cualquier contraseña válida                                                                                             |                                           | Reinicio de componente  | Transporte JMS solamente                                                  |
| jms.UserName                  | Cualquier nombre válido                                                                                                 |                                           | Reinicio de componente  | Transporte JMS solamente                                                  |
| JvmMaxHeapSize                | Tamaño de almacenamiento dinámico en megabytes                                                                          | 128m                                      | Reinicio de componente  | El directorio de depósito es <REMOTE> (el intermediario es ICS)           |
| JvmMaxNativeStackSize         | Tamaño de pila en kilobytes                                                                                             | 128k                                      | Reinicio de componente  | El directorio de depósito es <REMOTE> (el intermediario es ICS)           |

Tabla 17. Resumen de las propiedades de configuración estándar (continuación)

| Nombre de propiedad  | Valores posibles                                                                                                                               | Valor por omisión           | Método de actualización | Notas                                                                 |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------|-------------------------|-----------------------------------------------------------------------|
| JvmMinHeapSize       | Tamaño de almacenamiento dinámico en megabytes                                                                                                 | 1m                          | Reinicio de componente  | El directorio de depósito es <REMOTE> (el intermediario es ICS)       |
| ListenerConcurrency  | 1-100                                                                                                                                          | 1                           | Reinicio de componente  | El transporte de entrega será MQ                                      |
| Locale               | en_US, ja_JP, ko_KR, zh_CN, zh_TW, fr_FR, de_DE, it_IT, es_ES, pt_BR<br><b>Nota:</b> Esto es un subconjunto de los entornos locales admitidos. | en_US                       | Reinicio de componente  |                                                                       |
| LogAtInterchangeEnd  | true o false                                                                                                                                   | false                       | Reinicio de componente  | El directorio de depósito debe ser <REMOTE> (el intermediario es ICS) |
| MaxEventCapacity     | 1-2147483647                                                                                                                                   | 2147483647                  | Dinámico                | El directorio de depósito debe ser <REMOTE> (el intermediario es ICS) |
| MessageFileName      | Vía de acceso o nombre de archivo                                                                                                              | NOMBRECONECTORConnector.txt | Reinicio de componente  |                                                                       |
| MonitorQueue         | Cualquier nombre de cola válido                                                                                                                | NOMBRECONECTOR/MONITORQUEUE | Reinicio de componente  | Transporte JMS solamente: DuplicateEvent Elimination debe ser "true"  |
| OADAutoRestartAgent  | true o false                                                                                                                                   | false                       | Dinámico                | El directorio de depósito debe ser <REMOTE> (el intermediario es ICS) |
| OADMaxNumRetry       | Un número positivo                                                                                                                             | 1000                        | Dinámico                | El directorio de depósito debe ser <REMOTE> (el intermediario es ICS) |
| OADRetryTimeInterval | Un número positivo en minutos                                                                                                                  | 10                          | Dinámico                | El directorio de depósito debe ser <REMOTE> (el intermediario es ICS) |
| PollEndTime          | HH:MM                                                                                                                                          | HH:MM                       | Reinicio de componente  |                                                                       |

Tabla 17. Resumen de las propiedades de configuración estándar (continuación)

| Nombre de propiedad     | Valores posibles                                                                                                                                                                      | Valor por omisión                   | Método de actualización | Notas                                                                                                      |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------|-------------------------|------------------------------------------------------------------------------------------------------------|
| PollFrequency           | Un entero positivo en milisegundos<br><br>no (para inhabilitar sondeo)<br><br>key (para sondear sólo cuando se entre la letra p en la ventana del indicador de mandatos del conector) | 10000                               | Dinámico                |                                                                                                            |
| PollQuantity            | 1-500                                                                                                                                                                                 | 1                                   | Reinicio de agente      | Transporte JMS solamente: se especifica Container Managed Events                                           |
| PollStartTime           | HH:MM(HH es 0-23, MM es 0-59)                                                                                                                                                         | HH:MM                               | Reinicio de componente  |                                                                                                            |
| RepositoryDirectory     | Ubicación de depósito de metadatos                                                                                                                                                    |                                     | Reinicio de agente      | Para ICS: poner en <REMOTE><br>Para WebSphere MQ Message Brokers y WAS: poner en C:\crossworlds\repository |
| RequestQueue            | Nombre de cola válido de JMS                                                                                                                                                          | <i>NOMBRECONECTOR/REQUESTQUEUE</i>  | Reinicio de componente  | El transporte de entrega es JMS                                                                            |
| ResponseQueue           | Nombre de cola válido de JMS                                                                                                                                                          | <i>NOMBRECONECTOR/RESPONSEQUEUE</i> | Reinicio de componente  | El transporte de entrega es JMS: sólo necesario si el directorio de depósito es <REMOTE>                   |
| RestartRetryCount       | 0-99                                                                                                                                                                                  | 3                                   | Dinámico                |                                                                                                            |
| RestartRetryInterval    | Valor positivo apropiado en minutos: 1 - 2147483547                                                                                                                                   | 1                                   | Dinámico                |                                                                                                            |
| RHF2MessageDomain       | mrm, xml                                                                                                                                                                              | mrm                                 | Reinicio de componente  | sólo si el transporte de entrega es JMS y WireFormat es CwXML.                                             |
| SourceQueue             | Nombre válido de WebSphere MQ                                                                                                                                                         | <i>NOMBRECONECTOR/SOURCEQUEUE</i>   | Reinicio de agente      | sólo si el transporte de entrega es JMS y está especificado Container Managed Events                       |
| SynchronousRequestQueue |                                                                                                                                                                                       | <i>NCON/SYNCHRONOUSREQUESTQUEUE</i> | Reinicio de componente  | El transporte de entrega es JMS                                                                            |

Tabla 17. Resumen de las propiedades de configuración estándar (continuación)

| Nombre de propiedad           | Valores posibles                    | Valor por omisión                     | Método de actualización | Notas                                                                                            |
|-------------------------------|-------------------------------------|---------------------------------------|-------------------------|--------------------------------------------------------------------------------------------------|
| SynchronousRequestTimeout     | 0 - cualquier número (milisegundos) | 0                                     | Reinicio de componente  | El transporte de entrega es JMS                                                                  |
| SynchronousResponseQueue      |                                     | <i>NCON</i> /SYNCHRONOUSRESPONSEQUEUE | Reinicio de componente  | El transporte de entrega es JMS                                                                  |
| WireFormat                    | CwXML, CwBO                         | CwXML                                 | Reinicio de agente      | CwXML si el directorio de depósito no es <REMOTE>; CwBO si el directorio de depósito es <REMOTE> |
| WsifSynchronousRequestTimeout | 0 - cualquier número (milisegundos) | 0                                     | Reinicio de componente  | WAS solamente                                                                                    |
| XMLNamespaceFormat            | short, long                         | short                                 | Reinicio de agente      | WebSphere MQ Message Brokers y WAS solamente                                                     |

## Propiedades de configuración estándar

Esta sección lista y define cada una de las propiedades de configuración estándar de conector.

### AdminInQueue

Es la cola utilizada por el intermediario de integración para enviar mensajes administrativos al conector.

El valor por omisión es *NOMBRECONECTOR*/ADMININQUEUE.

### AdminOutQueue

Es la cola utilizada por el conector para enviar mensajes administrativos al intermediario de integración.

El valor por omisión es *NOMBRECONECTOR*/ADMINOUTQUEUE.

### AgentConnections

sólo es aplicable si *RepositoryDirectory* es <REMOTE>.

La propiedad *AgentConnections* controla el número de conexiones de ORB (Object Request Broker) abiertas por *orb.init[]*.

El valor por omisión de esta propiedad está establecido en 1. Puede cambiar este valor según convenga.

### AgentTraceLevel

Es nivel de los mensajes de rastreo para el componente específico de la aplicación. El valor por omisión es 0. El conector entrega todos los mensajes de rastreo aplicables con el nivel de rastreo establecido o un nivel menor.

## ApplicationName

Es el nombre que identifica de forma exclusiva la aplicación del conector. El administrador del sistema utiliza este nombre para supervisar el entorno del sistema WebSphere Business Integration. Debe asignar un valor a esta propiedad antes de ejecutar el conector.

## BrokerType

Identifica el tipo de intermediario de integración que el usuario está utilizando. Las opciones son ICS, intermediarios de mensajes WebSphere (WMQL, WMQIB o WBIMB) o WAS.

## CharacterEncoding

Especifica el conjunto de códigos de caracteres utilizado para correlacionar un carácter (letra del alfabeto, representación numérica o signo de puntuación) con un valor numérico.

**Nota:** Los conectores basados en Java no utilizan esta propiedad. Los conectores para C++ utilizan actualmente `asci i7` como valor de esta propiedad.

Por omisión, la lista desplegable sólo muestra un subconjunto de los juegos de caracteres admitidos. Para añadir otros valores admitidos a la lista desplegable, debe modificar manualmente el archivo `\Data\Std\stdConnProps.xml` en el directorio del producto. Para obtener más información, consulte el apéndice referente al Configurador de conectores.

## ConcurrentEventTriggeredFlows

sólo es aplicable si `RepositoryDirectory` es `<REMOTE>`.

Determina cuántos objetos comerciales pueden ser procesados simultáneamente por el conector para la entrega de sucesos. Asigne como valor de este atributo el número de objetos comerciales que desee que se correlacionen y entreguen a la vez. Por ejemplo, establezca el valor de esta propiedad en 5 para hacer que se procesen cinco objetos comerciales simultáneamente. El valor por omisión es 1.

Establecer esta propiedad en un valor mayor que 1 permite que el conector de una aplicación de origen correlacione al mismo tiempo varios objetos comerciales de sucesos y los entregue simultáneamente a varias instancias de colaboración. Esto acelera la entrega de objetos comerciales al intermediario de integración, especialmente si los objetos comerciales utilizan correlaciones complejas. El aumento del ritmo de llegada de objetos comerciales a las colaboraciones puede aumentar el rendimiento general del sistema.

Para implementar el proceso simultáneo para un flujo completo (desde una aplicación de origen a una aplicación de destino), debe seguir estos pasos:

- Configure la colaboración para que utilice varias hebras estableciendo la propiedad `Número máximo de sucesos simultáneos` en un valor lo suficientemente alto para utilizar varias hebras.
- Asegúrese de que el componente específico de aplicación de la aplicación de destino puede procesar peticiones simultáneamente. Es decir, debe ser de ejecución paralela, lo que equivale a ser capaz de utilizar el paralelismo de agente de conector y ser configurado para varios procesos. Establezca la propiedad de configuración `Grado de proceso paralelo` en un valor mayor que 1.

La propiedad `ConcurrentEventTriggeredFlows` no afecta al sondeo realizado por el conector, el cual tiene una sola hebra y se ejecuta secuencialmente.

## ContainerManagedEvents

Esta propiedad permite que un conector habilitado para JMS que hace uso de un almacén de sucesos JMS proporcione una entrega de sucesos garantizada, en la que un suceso se elimina de la cola de origen y se coloca en la cola de destino como transacción JMS individual.

No existe valor por omisión.

Cuando `ContainerManagedEvents` tiene el valor `JMS`, debe configurar las propiedades siguientes para habilitar la entrega de sucesos garantizada:

- `PollQuantity = 1 a 500`
- `SourceQueue = /SOURCEQUEUE`

Debe también configurar un gestor de datos con las propiedades `MimeType`, `DHClass` (clase de gestor de datos) y `DataHandlerConfigMOName` (nombre del metaobjeto, que es opcional). Para definir esos valores, utilice la pestaña **Gestor de datos** del Configurador de conectores.

Estas propiedades son específicas del adaptador, pero los valores de **ejemplo** son:

- `MimeType = text/xml`
- `DHClass = com.crossworlds.DataHandlers.text.xml`
- `DataHandlerConfigMOName = MO_DataHandler_Default`

Los campos para estos valores en la pestaña Gestor de datos sólo se muestran si ha establecido `ContainerManagedEvents` en `JMS`.

**Nota:** Cuando `ContainerManagedEvents` tiene el valor `JMS`, el conector *no* invoca su método `pollForEvents()`, por lo que se inhabilita la funcionalidad de ese método.

Esta propiedad sólo se muestra si la propiedad `DeliveryTransport` tiene el valor `JMS`.

## ControllerStoreAndForwardMode

sólo es aplicable si `RepositoryDirectory` es `<REMOTE>`.

Determina cómo actúa el controlador del conector después de que éste detecte que el componente específico de la aplicación de destino no está disponible.

Si esta propiedad tiene el valor `true` y el componente específico de la aplicación de destino no está disponible cuando un suceso llega a ICS, el controlador del conector bloquea la petición dirigida al componente específico de la aplicación. Cuando el componente específico de la aplicación pasa a ser funcional, el controlador le reenvía la petición.

Sin embargo, si el componente específico de la aplicación de destino pasa a estar no disponible **después** de que el controlador del conector le reenvíe una petición de llamada de servicio, el controlador del conector cancela la petición.

Si esta propiedad tiene el valor `false`, el controlador del conector comienza a cancelar todas las peticiones de llamada de servicio tan pronto como detecta que el componente específico de la aplicación de destino no está disponible.

El valor por omisión es `true`.

## ControllerTraceLevel

sólo es aplicable si `RepositoryDirectory` es `<REMOTE>`.

Es nivel de los mensajes de rastreo para el controlador del conector. El valor por omisión es `0`.

## DeliveryQueue

sólo es aplicable si `DeliveryTransport` es `JMS`.

Es la cola utilizada por el conector para enviar objetos comerciales al intermediario de integración.

El valor por omisión es `NOMBRECONECTOR/DELIVERYQUEUE`.

## DeliveryTransport

Especifica el mecanismo de transporte para la entrega de sucesos. Los valores posibles son `MQ` para WebSphere MQ, `IDL` para CORBA IIOP, o `JMS` para Java Messaging Service.

- Si `RepositoryDirectory` tiene el valor `REMOTE`, el valor de la propiedad `DeliveryTransport` puede ser `MQ`, `IDL` o `JMS`, y el valor por omisión es `IDL`.
- Si `RepositoryDirectory` es un directorio local, el valor solamente puede ser `JMS`.

El conector envía peticiones de llamada de servicio y mensajes administrativos a través de CORBA IIOP si el valor configurado para la propiedad `DeliveryTransport` es `MQ` o `IDL`.

### WebSphere MQ e IDL

Utilice WebSphere MQ en lugar de IDL para el transporte de la entrega de sucesos, a menos que tenga un sólo producto. WebSphere MQ proporciona las ventajas siguientes respecto a IDL:

- Comunicación asíncrona:  
WebSphere MQ permite que el componente específico de la aplicación sondee y guarde sucesos de forma permanente aunque el servidor no esté disponible.
- Rendimiento en el extremo del servidor:  
WebSphere MQ proporciona un mejor rendimiento en el extremo del servidor. En la modalidad optimizada, WebSphere MQ sólo guarda el puntero que apunta a un suceso contenido en la base de datos de depósito, mientras que el suceso propiamente dicho permanece en la cola de WebSphere MQ. Esto ahorra tener que escribir sucesos potencialmente grandes en la base de datos de depósito.
- Rendimiento en el extremo del agente:  
WebSphere MQ proporciona un mejor rendimiento en el extremo del componente específico de la aplicación. Cuando se utiliza WebSphere MQ, el proceso de sondeo del conector captura un suceso, lo coloca en la cola del conector y luego captura el suceso siguiente. Esto es más rápido que IDL, que necesita que el proceso de sondeo del conector capture un suceso, vaya por la red hacia el proceso servidor, guarde el suceso de forma permanente en la base de datos de depósito y luego capture el suceso siguiente.



## JMS

Permite la comunicación entre el conector y la infraestructura del conector cliente utilizando Java Messaging Service (JMS).

Si selecciona JMS como transporte de entrega, el Configurador de conectores muestra más propiedades de JMS, como `jms.MessageBrokerName`, `jms.FactoryClassName`, `jms.Password` y `jms.UserName`. Las primeras dos de estas propiedades son necesarias para este mecanismo de transporte.

**Importante:** Puede existir una limitación de memoria si utiliza el mecanismo de transporte JMS para un conector en el entorno siguiente:

- AIX 5.0
- WebSphere MQ 5.3.0.1
- Cuando ICS es el intermediario de integración

En este entorno, puede tener dificultades para iniciar el controlador del conector (en el extremo del servidor) y el conector (en el extremo del cliente) debido al uso de la memoria dentro del cliente WebSphere MQ. Si su instalación utiliza menos de 768M como tamaño del almacenamiento dinámico de proceso, IBM recomienda que siga estos pasos:

- Defina la variable de entorno `LDR_CNTRL` en el script `CWSharedEnv.sh`.  
Este script reside en el directorio `\bin`, dentro del directorio del producto. Mediante un editor de texto, añada la línea siguiente como primera línea del script `CWSharedEnv.sh`:  

```
export LDR_CNTRL=MAXDATA=0x30000000
```

  
Esta línea restringe el uso de la memoria a un máximo de 768 MB (3 segmentos de 256 MB). Si la memoria de proceso crece por encima de este límite, puede producirse un intercambio de páginas de memoria, lo cual puede afectar negativamente al rendimiento del sistema.
- Asigne el valor 11 o 12 a la propiedad `IPCCBaseAddress`. Para obtener más información sobre esta propiedad, consulte el manual *System Installation Guide for UNIX*.

## DuplicateEventElimination

Cuando establece esta propiedad en `true`, un conector habilitado para JMS puede garantizar que no se entreguen sucesos duplicados a la cola de entrega. Para utilizar esta característica, el conector debe tener definido en el código específico de la aplicación un identificador de sucesos exclusivo para el atributo `ObjectEventId` del objeto comercial. Esto se realiza durante el desarrollo del conector.

Esta propiedad también se puede establecer en `false`.

**Nota:** Cuando `DuplicateEventElimination` tiene el valor `true`, debe también configurar la propiedad `MonitorQueue` para habilitar la entrega garantizada de sucesos.

## FaultQueue

Si el conector encuentra un error al procesar un mensaje, el conector coloca el mensaje en la cola especificada por esta propiedad, junto con un indicador de estado y una descripción del problema.

El valor por omisión es `NOMBRECONECTOR/FAULTQUEUE`.

## JvmMaxHeapSize

Es el tamaño máximo del almacenamiento dinámico del agente (expresado en megabytes). Esta propiedad sólo es aplicable si el valor de RepositoryDirectory es <REMOTE>.

El valor por omisión es 128m.

## JvmMaxNativeStackSize

Es el tamaño máximo de la pila nativa del agente (expresado en kilobytes). Esta propiedad sólo es aplicable si el valor de RepositoryDirectory es <REMOTE>.

El valor por omisión es 128k.

## JvmMinHeapSize

Es el tamaño mínimo del almacenamiento dinámico del agente (expresado en megabytes). Esta propiedad sólo es aplicable si el valor de RepositoryDirectory es <REMOTE>.

El valor por omisión es 1m.

## jms.FactoryClassName

Especifica el nombre de clase para el que se debe crear una instancia para un proveedor JMS. *Debe* definir esta propiedad del conector si selecciona JMS como mecanismo de transporte de entrega (DeliveryTransport).

El valor por omisión es CxCommon.Messaging.jms.IBMMQSeriesFactory.

## jms.MessageBrokerName

Especifica el nombre del intermediario que se debe utilizar para el proveedor JMS. *Debe* definir esta propiedad del conector si selecciona JMS como mecanismo de transporte de entrega (DeliveryTransport).

El valor por omisión es crossworlds.queue.manager. Utilice el valor por omisión cuando se conecte a un intermediario de mensajes local.

Cuando se conecta a un intermediario de mensajes remoto, esta propiedad toma los valores (obligatorios) siguientes:

nombreGestorColas:<Canal>:<nombreSistemaPrincipal>:<númeroPuerto>,

donde las variables son:

nombreGestorColas: el nombre del gestor de colas.

Canal: el canal utilizado por el cliente.

nombreSistemaPrincipal: el nombre de la máquina donde debe residir el gestor de colas.

númeroPuerto: el número de puerto que debe utilizar el gestor de colas para estar a la escucha.

Por ejemplo:

```
jms.MessageBrokerName = WBIMB.Queue.Manager:CHANNEL1:RemoteMachine:1456
```

## **jms.NumConcurrentRequests**

Especifica el número máximo de peticiones simultáneas de llamada de servicio que se pueden enviar a un conector al mismo tiempo. Cuando se alcanza ese máximo, las nuevas llamadas de servicio se bloquean y esperan a que finalice otra petición antes de continuar.

El valor por omisión es 10.

## **jms.Password**

Especifica la contraseña del proveedor JMS. Es opcional especificar un valor para esta propiedad.

No existe valor por omisión.

## **jms.UserName**

Especifica el nombre de usuario del proveedor JMS. Es opcional especificar un valor para esta propiedad.

No existe valor por omisión.

## **ListenerConcurrency**

Esta propiedad permite la ejecución paralela en MQ Listener cuando el intermediario de integración es ICS. Esta propiedad hace posible la escritura por lotes de varios sucesos en la base de datos, con lo que mejora el rendimiento del sistema. El valor por omisión es 1.

Esta propiedad sólo es aplicable a los conectores que hacen uso del mecanismo de transporte MQ. La propiedad `DeliveryTransport` debe tener el valor MQ.

## **Locale**

Especifica el código de idioma, el país o territorio y, opcionalmente, el correspondiente juego de caracteres. El valor de esta propiedad determina convenios culturales tales como la secuencia de ordenación y orden de clasificación de los datos, los formatos de fecha y hora, y los símbolos de moneda.

El nombre especificado para "locale" (entorno local) tiene este formato:

*ll\_TT.juegoCódigos*

donde:

*ll* es un código de idioma formado por dos caracteres (normalmente en letras minúsculas)

*TT* es un código de país o territorio formado por dos letras (normalmente en letras mayúsculas)

*juegoCódigos* es el nombre del correspondiente juego de caracteres; esta porción del nombre es opcional.

Por omisión, la lista desplegable sólo muestra un subconjunto de los entornos locales admitidos. Para añadir otros valores admitidos a la lista desplegable, debe modificar manualmente el archivo `\Data\Std\stdConnProps.xml` en el directorio del producto. Para obtener más información, consulte el apéndice referente al Configurador de conectores.

El valor por omisión es en\_US. Si el conector no se ha globalizado, el único valor válido para esta propiedad es en\_US. Para determinar si un conector determinado se ha globalizado, consulte la lista de versiones de conector en estos sitios Web:

<http://www.ibm.com/software/websphere/wbiadapters/infocenter>, o  
<http://www.ibm.com/websphere/integration/wicserver/infocenter>

## LogAtInterchangeEnd

sólo es aplicable si RepositoryDirectory es <REMOTE>.

Especifica si se deben registrar errores en el archivo de anotaciones cronológicas del intermediario de integración de destino. El registro de anotaciones en el archivo de registro del intermediario de destino también activa la notificación por correo electrónico, la cual genera mensajes de correo electrónico para el receptor de mensajes (MESSAGE\_RECIPIENT) especificado en el archivo InterchangeSystem.cfg cuando se producen errores.

Por ejemplo, cuando un conector pierde la conexión con su aplicación y LogAtInterChangeEnd tiene el valor true, se envía un mensaje de correo electrónico al receptor de mensajes especificado. El valor por omisión es false.

## MaxEventCapacity

Es el número máximo de sucesos contenidos en el almacenamiento intermedio del controlador. Esta propiedad es utilizada por el control de flujo y sólo es aplicable si el valor de la propiedad RepositoryDirectory es <REMOTE>.

El valor puede ser un número entero positivo comprendido entre 1 y 2147483647. El valor por omisión es 2147483647.

## MessageFileName

Es el nombre del archivo de mensajes del conector. La ubicación estándar del archivo de mensajes es \connectors\messages, dentro del directorio del producto. Especifique el nombre del archivo de mensajes en forma de vía de acceso absoluta si el archivo de mensajes no está situado en la ubicación estándar.

Si no existe un archivo de mensajes de conector, el conector utiliza InterchangeSystem.txt como archivo de mensajes. Este archivo está situado en el directorio del producto.

**Nota:** Para determinar si un conector determinado tiene su propio archivo de mensajes, consulte la guía del usuario del adaptador respectivo.

## MonitorQueue

Es la cola lógica que el conector utiliza para supervisar sucesos duplicados. Esta propiedad sólo es aplicable si el valor de la propiedad DeliveryTransport es JMS y DuplicateEventElimination está establecido en TRUE.

El valor por omisión es NOMBRECONECTOR/MONITORQUEUE.

## OADAutoRestartAgent

sólo es aplicable si RepositoryDirectory es <REMOTE>.

Especifica si el conector utiliza el reinicio automático y remoto. Esta característica hace uso de OAD (Object Activation Daemon) desencadenado por MQ para reiniciar el conector después de una terminación anómala o para iniciar un conector remoto desde el Supervisor del sistema.

Esta propiedad debe tener el valor `true` para permitir el reinicio automático y remoto. Para obtener información sobre cómo configurar OAD desencadenado por MQ, consulte el manual *Installation Guide for Windows* o *Installation Guide for UNIX*.

El valor por omisión es `false`.

## **OADMaxNumRetry**

sólo es aplicable si `RepositoryDirectory` es `<REMOTE>`.

Especifica el número máximo de veces que OAD desencadenado por MQ intenta automáticamente reiniciar el conector después de una terminación anómala. La propiedad `OADAutoRestartAgent` debe tener el valor `true` para que esta propiedad sea aplicable.

El valor por omisión es `1000`.

## **OADRetryTimeInterval**

sólo es aplicable si `RepositoryDirectory` es `<REMOTE>`.

Especifica el número de minutos dentro del intervalo de tiempo de reintento para el daemon OAD desencadenado por MQ. Si el agente del conector no se reinicia dentro de este intervalo de tiempo de reintento, el controlador del conector solicita al daemon OAD que reinicie de nuevo el agente del conector. El daemon OAD repite este proceso de reintento tantas veces como especifique la propiedad `OADMaxNumRetry`. La propiedad `OADAutoRestartAgent` debe tener el valor `true` para que esta propiedad sea aplicable.

El valor por omisión es `10`.

## **PollEndTime**

Es la hora para detener el sondeo de la cola de sucesos. El formato es `HH:MM`, donde `HH` representa 0-23 horas, y `MM` representa 0-59 segundos.

Debe proporcionar un valor válido para esta propiedad. El valor por omisión es `HH:MM`, pero cambiarse.

## **PollFrequency**

Es el intervalo que transcurre entre el final del último sondeo y el inicio del sondeo siguiente. `PollFrequency` especifica la cantidad de tiempo (en milisegundos) que transcurre entre el final de una acción de sondeo y el inicio de la acción de sondeo siguiente. Esto no es el intervalo entre acciones de sondeo, sino que la lógica es la siguiente:

- Sondear para obtener el número de objetos especificado por el valor de `PollQuantity`.
- Procesar estos objetos. Para algunos adaptadores, esto se puede realizar parcialmente en hebras separadas, que se ejecutan de forma asíncrona hasta la acción de sondeo siguiente.
- Retardo para el intervalo especificado por `PollFrequency`.

- Repetir el ciclo.

Establezca `PollFrequency` en uno de los valores siguientes:

- El número de milisegundos entre acciones de sondeo (valor entero).
- La palabra `key`, que hace que el conector ejecute el sondeo sólo cuando el usuario escribe la letra `p` en la ventana del indicador de mandatos del conector. Escriba la palabra en minúsculas.
- La palabra `no`, que hace que el conector no ejecute el sondeo. Escriba la palabra en minúsculas.

El valor por omisión es `10000`.

**Importante:** Existen restricciones sobre el uso de esta propiedad para algunos conectores. Cuando existen estas restricciones, están documentadas en el capítulo referente a la instalación y configuración del adaptador.

## PollQuantity

Especifica el número de elementos de la aplicación para los que debe efectuar el sondeo el conector. Si el adaptador tiene una propiedad específica del conector para definir `PollQuantity`, el valor establecido en esa propiedad prevalece sobre el valor de la propiedad estándar.

FIX

Un mensaje de correo electrónico también se considera un suceso. El conector actúa del modo siguiente cuando se le sondea para correo electrónico.

Cuando se sondea una vez (PO), el conector va a recoger el cuerpo del mensaje, porque también se considera un anexo. Puesto que no se ha especificado ningún manejador de datos (DH) para este tipo mime, se pasará por alto el cuerpo. El conector procesa el primer anexo PO. DH se encuentra disponible para este tipo mime, por lo que envía el objeto comercial al Conector de prueba visual (VTC). Cuando se sondea por segunda vez, el conector procesa el segundo anexo PO. El DH se encuentra disponible para este tipo mime, por lo que envía el objeto comercial al Conector de prueba visual. Una vez aceptado, debería pasar el tercer anexo PO. Este es el comportamiento correcto.

## PollStartTime

Es la hora para iniciar el sondeo de la cola de sucesos. El formato es `HH:MM`, donde `HH` representa 0-23 horas, y `MM` representa 0-59 segundos.

Debe proporcionar un valor válido para esta propiedad. El valor por omisión es `HH:MM`, pero debe cambiarse.

## RequestQueue

Es la cola utilizada por el intermediario de integración para enviar objetos comerciales al conector.

El valor por omisión es `CONNECTOR/REQUESTQUEUE`.

## RepositoryDirectory

Es la ubicación del depósito donde el conector lee los documentos de esquema XML donde se guardan los metadatos para definiciones de objetos comerciales.

Cuando el intermediario de integración es ICS, esta propiedad debe tener el valor <REMOTE>, pues el conector obtiene esa información del depósito de InterChange Server.

Cuando el intermediario de integración es un intermediario de mensajes WebSphere o WAS, este valor debe ser el <directorio local>.

## ResponseQueue

Es aplicable sólo si DeliveryTransport es JMS y es necesario sólo si RepositoryDirectory es <REMOTE>.

Designa la cola de respuestas JMS, la cual entrega un mensaje de respuesta desde la infraestructura del conector al intermediario de integración. Cuando el intermediario de integración es ICS, el servidor envía la petición y espera a que aparezca un mensaje de respuesta en la cola de respuestas JMS.

## RestartRetryCount

Especifica el número de veces que el conector intenta reiniciarse a sí mismo. Cuando esta propiedad se utiliza para un conector paralelo, especifica el número de veces que el componente específico de la aplicación del conector maestro intenta reiniciar el componente específico de la aplicación del conector esclavo.

El valor por omisión es 3.

## RestartRetryInterval

Especifica el intervalo, en minutos, tras el cual el conector intenta reiniciarse a sí mismo. Cuando esta propiedad se utiliza para un conector paralelo, especifica el intervalo tras el cual el componente específico de la aplicación del conector maestro intenta reiniciar el componente específico de la aplicación del conector esclavo. Los valores posibles están comprendidos entre 1 y 2147483647.

El valor por omisión es 1.

## RHF2MessageDomain

sólo aplicable a los intermediarios de mensajes WebSphere y WAS.

Esta propiedad le permite configurar el valor del campo correspondiente al nombre de dominio en la cabecera JMS. Cuando se envían datos a WMQI a través del mecanismo de transporte JMS, la infraestructura del adaptador escribe información para la cabecera JMS, con un nombre de dominio y un valor fijo de mrm. Un nombre de dominio configurable permite a los usuarios hacer un seguimiento de cómo el intermediario WMQI procesa los datos de mensajes.

Esto es un ejemplo del aspecto que tendría una cabecera:

```
<mcd><Msd>mrm</Msd><Set>3</Set><Type>
Retek_POPhyDesc</Type><Fmt>CwXML</Fmt></mcd>
```

El valor por omisión es mrm, pero también se puede asignar el valor xml. Esta propiedad sólo se muestra cuando DeliveryTransport tiene el valor JMS y WireFormat tiene el valor CwXML.

## SourceQueue

Esta propiedad sólo es aplicable si `DeliveryTransport` es `JMS` y `ContainerManagedEvents` está especificado.

Esta propiedad designa la cola JMS de origen que la infraestructura del conector utiliza para permitir la entrega de sucesos garantizada mediante conectores habilitados para JMS que hacen uso de un almacén de sucesos JMS. Para obtener más información, consulte “`ContainerManagedEvents`” en la página 109.

El valor por omisión es `CONNECTOR/SOURCEQUEUE`.

## SynchronousRequestQueue

sólo es aplicable si `DeliveryTransport` es `JMS`.

La cola especificada por esta propiedad entrega mensajes de peticiones que necesitan una respuesta síncrona desde la infraestructura del conector al intermediario. Esta cola es necesaria sólo si el conector hace uso de la ejecución síncrona. Cuando se utiliza la ejecución síncrona, la infraestructura del conector envía un mensaje a la cola `SynchronousRequestQueue` y espera la respuesta del intermediario en la cola `SynchronousResponseQueue`. El mensaje de respuesta enviado al conector lleva un ID de correlación que coincide con el ID del mensaje original.

El valor por omisión es `NOMBRECONECTOR/SYNCHRONOUSREQUESTQUEUE`

## SynchronousResponseQueue

sólo es aplicable si `DeliveryTransport` es `JMS`.

La cola especificada por esta propiedad entrega mensajes de respuesta enviados en respuesta a una petición síncrona desde el intermediario a la infraestructura del conector. Esta cola es necesaria sólo si el conector hace uso de la ejecución síncrona.

El valor por omisión es `NOMBRECONECTOR/SYNCHRONOUSRESPONSEQUEUE`

## SynchronousRequestTimeout

sólo es aplicable si `DeliveryTransport` es `JMS`.

Especifica el tiempo, en minutos, que el conector espera una respuesta a una petición síncrona. Si la respuesta no se recibe dentro del tiempo especificado, el conector traslada el mensaje original de petición síncrona a la cola de errores, junto con un mensaje de error.

El valor por omisión es 0.

## WireFormat

Es el formato del mensaje en el mecanismo de transporte.

- Si `RepositoryDirectory` es un directorio local, el valor de esta propiedad es `CwXML`.
- Si el valor de `RepositoryDirectory` es `<REMOTE>`, el valor de esta propiedad es `CwB0`.



## **WsifSynchronousRequestTimeout**

sólo aplicable si el intermediario de integración es WAS.

Especifica el tiempo, en minutos, que el conector espera una respuesta a una petición síncrona. Si la respuesta no se recibe dentro del tiempo especificado, el conector traslada el mensaje original de petición síncrona a la cola de errores, junto con un mensaje de error.

El valor por omisión es 0.

## **XMLNamespaceFormat**

sólo aplicable a los intermediarios de mensajes WebSphere y al intermediario de integración WAS.

Esta poderosa propiedad permite al usuario especificar espacios de nombres cortos ("short") y largos ("long") en el formato XML de definiciones de objetos comerciales.

El valor por omisión es short.



---

## Apéndice B. Configurador de conectores

Este apéndice describe cómo utilizar el Configurador de conectores para definir valores de propiedades de configuración para el adaptador.

Utilice el Configurador de conectores para:

- Crear una plantilla de propiedades específica del conector para configurar el conector
- Crear un archivo de configuración
- Definir propiedades en un archivo de configuración

**Nota:**

En el presente documento, las barras inclinadas invertidas (\) se utilizan como convenio para representar las vías de acceso de directorios. Para las instalaciones en UNIX, utilice barras inclinadas (/) en lugar de barras inclinadas invertidas y siga los convenios de cada sistema operativo.

Los temas tratados en este apéndice son:

- “Visión general del Configurador de conectores” en la página 121
- “Inicio del Configurador de conectores” en la página 122
- “Creación de una plantilla de propiedades específica del conector” en la página 123
- “Creación de un archivo de configuración nuevo” en la página 126
- “Definición de propiedades en el archivo de configuración” en la página 129
- “Uso del Configurador de conectores en un entorno globalizado” en la página 136

---

### Visión general del Configurador de conectores

El Configurador de conectores le permite configurar el componente conector del adaptador para utilizarlo con estos intermediarios de integración:

- WebSphere InterChange Server (ICS)
- WebSphere MQ Integrator, WebSphere MQ Integrator Broker y WebSphere Business Integration Message Broker se designan colectivamente como WebSphere Message Brokers (WMQI)
- WebSphere Application Server (WAS)

Utilice el Configurador de conectores para:

- Crear una **plantilla de propiedades específica del conector** para configurar el conector.
- Crear un **archivo de configuración del conector**; debe crear un archivo de configuración para cada conector que instale.
- Definir propiedades en un archivo de configuración.  
Puede ser necesario que modifique los valores por omisión que están definidos para las propiedades en las plantillas de conector. Debe también designar definiciones de objetos comerciales admitidas y, con ICS, correlaciones para utilizar con colaboraciones así como especificar parámetros de mensajería, anotaciones cronológicas, rastreo y de gestor de datos, según sea necesario.

La forma en que ejecuta el Configurador de conectores y el tipo de archivo de configuración que utiliza pueden ser diferentes dependiendo del intermediario de integración que esté utilizando. Por ejemplo, si el intermediario utilizado es WMQI, el Configurador de conectores se ejecuta directamente, y no desde el Gestor del sistema (consulte “Ejecución del Configurador en la modalidad autónoma” en la página 122).

Las propiedades de configuración del conector comprenden propiedades de configuración estándar (comunes a todos los conectores) y propiedades específicas del conector (que el conector necesita para una determinada aplicación o tecnología).

Debido a que las **propiedades estándar** son utilizadas por todos los conectores, no es necesario que defina esas propiedades partiendo desde cero; el Configurador de conectores las incorpora al archivo de configuración tan pronto como el usuario crea el archivo. Sin embargo, es necesario que defina el valor de cada propiedad estándar en el Configurador de conectores.

El conjunto de propiedades estándar puede no ser el mismo para todos los intermediarios y configuraciones. Algunas propiedades sólo están disponibles si se asigna un valor determinado a otras propiedades. La ventana Propiedades estándar del Configurador de conectores muestra las propiedades que están disponibles para la configuración particular utilizada.

Sin embargo, para las **propiedades específicas del conector**, debe definir primero las propiedades y luego establecer sus valores. Para ello debe crear una plantilla de propiedades, específica del conector, para el adaptador que utilice. Puede que ya esté configurada una plantilla en su sistema, en cuyo caso, simplemente utilice esa plantilla. De lo contrario, siga los pasos descritos en “Creación de una plantilla nueva” en la página 123 para configurar una plantilla nueva.

**Nota:** El Configurador de conectores sólo se ejecuta en un entorno Windows. Si está ejecutando el conector en un entorno UNIX, utilice el Configurador de conectores en Windows para modificar el archivo de configuración y luego copie el archivo en su entorno UNIX.

---

## Inicio del Configurador de conectores

Puede iniciar y ejecutar el Configurador de conectores en una de las dos modalidades siguientes:

- De forma independiente, en la modalidad autónoma
- Desde el Gestor del sistema

### Ejecución del Configurador en la modalidad autónoma

Puede ejecutar el Configurador de conectores de forma independiente y trabajar con archivos de configuración de conector, para cualquiera que sea el intermediario de integración que utilice.

Para ello siga estos pasos:

- Seleccione **Inicio > Programas**, y luego pulse **IBM WebSphere InterChange Server > IBM WebSphere Business Integration Tools > Configurador de conectores**.
- Seleccione **Archivo > Nuevo > Configuración de conectores**.

- Cuando pulsa en el menú desplegable situado junto a **System Connectivity Integration Broker**, puede seleccionar ICS, WebSphere Message Brokers o WAS, dependiendo del intermediario que utilice.

Puede elegir ejecutar el Configurador de conectores de forma independiente para crear el archivo, y luego conectar con el Gestor del sistema para guardarlo en un proyecto del Gestor del sistema (consulte “Finalización de un archivo de configuración” en la página 128.)

---

## Ejecución del Configurador desde el Gestor del sistema

Puede ejecutar el Configurador de conectores desde el Gestor del sistema.

Para ejecutar el Configurador de conectores:

1. Abra el Gestor del sistema.
2. En la ventana del Gestor del sistema, abra el icono **Bibliotecas de componentes de integración** y resalte **Conectores**.
3. En la barra de menús del Gestor del sistema, pulse **Herramientas > Configurador de conectores**. Se abrirá la ventana del Configurador de conectores, que muestra el recuadro de diálogo **Conector nuevo**.
4. Cuando pulsa en el menú desplegable situado junto a **System Connectivity Integration Broker**, puede seleccionar ICS, WebSphere Message Brokers o WAS, dependiendo del intermediario que utilice.

Para editar un archivo de configuración existente:

- En la ventana del Gestor del sistema, seleccione cualquiera de los archivos de configuración listados en la carpeta Conector y pulse sobre él con el botón derecho del ratón. Se abrirá el Configurador de conectores, que muestra el archivo de configuración con el tipo de intermediario de integración y el nombre de archivo visualizados en la parte superior.
- Desde el Configurador de conectores, seleccione **Archivo > Abrir**. Seleccione el nombre del archivo de configuración de conector desde un proyecto o directorio donde esté contenido.
- Pulse la pestaña Propiedades estándar para ver las propiedades incluidas en el archivo de configuración.

---

## Creación de una plantilla de propiedades específica del conector

Para crear un archivo de configuración para el conector, necesita una plantilla de propiedades específica del conector y las propiedades estándar proporcionadas por el sistema.

Puede crear una plantilla nueva para las propiedades específicas del conector o bien utilizar como plantilla una definición de conector existente.

- Para crear una plantilla nueva, consulte “Creación de una plantilla nueva” en la página 123.
- Para utilizar un archivo existente, simplemente modifique una plantilla existente y guárdela con el nuevo nombre. Puede encontrar plantillas ya creadas en el directorio `\WebSphereAdapters\bin\Data\App`.

### Creación de una plantilla nueva

Esta sección describe cómo crear propiedades en la plantilla, definir características generales y valores para esas propiedades y especificar las dependencias necesarias

entre las propiedades. A continuación, guarde la plantilla y utilícela como base para crear un nuevo archivo de configuración de conector.

Para crear una plantilla en el Configurador de conectores:

1. Pulse **Archivo > Nuevo > Plantilla de propiedades específicas del conector**.
2. Se abrirá el recuadro de diálogo **Plantilla de propiedades específicas del conector**.
  - Escriba un nombre para la nueva plantilla en el campo **Nombre**, debajo de **Entrar un nombre de plantilla nuevo**. Este nombre se mostrará de nuevo cuando abra el recuadro de diálogo para crear un nuevo archivo de configuración desde una plantilla.
  - Para ver las definiciones de propiedades específicas del conector contenidas en una plantilla cualquiera, seleccione el nombre de esa plantilla en la pantalla **Nombre de plantilla**. Se abrirá la pantalla **Presentación preliminar de la plantilla**, que muestra una lista de las definiciones de propiedades contenidas en la plantilla.
3. Como base inicial para su plantilla puede utilizar una plantilla existente cuyas definiciones de propiedades sean similares a las que necesita su conector. Si no encuentra ninguna plantilla que muestre las propiedades específicas de conector utilizadas por su conector, necesitará crear una plantilla.
  - Si piensa modificar una plantilla existente, seleccione el nombre de la plantilla en la lista que muestra la tabla **Nombre de plantilla**, debajo de **Seleccionar la plantilla existente a modificar: Encontrar plantilla**.
  - Esta tabla muestra los nombres de todas las plantillas existentes actualmente. Puede también buscar una plantilla.

## Especificación de características generales

Cuando pulsa **Siguiente** para seleccionar una plantilla, se abre el recuadro de diálogo **Propiedades - Plantilla de propiedades específicas del conector**. El recuadro de diálogo tiene la pestaña **General**, para las características generales de las propiedades definidas, y la pestaña **Valor**, para restricciones de valores. La pantalla **General** tiene los campos siguientes:

- **General:**
  - Tipo de propiedad
  - Método actualizado
  - Descripción
- **Distintivos**
  - Distintivos estándar
- **Distintivo personalizado**
  - Distintivo

Una vez seleccionadas las características generales de la propiedad, pulse la pestaña **Valor**.

## Especificación de valores

La pestaña **Valor** le permite establecer la longitud máxima, los valores múltiples máximos, un valor por omisión o un rango de valores para la propiedad. También permite editar valores. Para ello:

1. Pulse la pestaña **Valor**. El panel de pantalla de Valor pasa a sustituir al panel de pantalla de General.
2. Seleccione el nombre de la propiedad en la pantalla **Editar propiedades**.
3. Especifique los valores que desee en los campos **Longitud máxima** y **Valores múltiples máximos**.

Para crear un nuevo valor de propiedad:

1. Seleccione la propiedad en la lista **Editar propiedades** y pulse en ella con el botón derecho del ratón.
2. En el recuadro de diálogo, seleccione **Agregar**.
3. Escriba el nombre del nuevo valor de propiedad y pulse Aceptar. El valor se mostrará en el panel **Valor** del lado derecho.

El panel **Valor** muestra una tabla con tres columnas:

La columna **Valor** muestra el valor que especificó en el recuadro de diálogo **Valor de propiedad**, y los valores anteriores que haya creado.

La columna **Valor por omisión** le permite designar cualquiera de los valores como valor por omisión.

El **Rango de valores** muestra el rango que entró en el recuadro de diálogo **Valor de propiedad**.

Una vez creado un valor y mostrado en la tabla, se puede editar desde dentro de la tabla.

Para hacer un cambio en un valor contenido en la tabla, seleccione una fila completa pulsando en el número de fila. Luego pulse con el botón derecho del ratón en el campo **Valor** y pulse **Editar valor**.

## Definición de dependencias

Cuando haya hecho los cambios deseados en las pestañas **General** y **Valor**, pulse **Siguiente**. Se abrirá el recuadro de diálogo **Dependencias - Plantilla de propiedades específicas del conector**.

Una propiedad dependiente es una propiedad que se incluye en la plantilla y se utiliza en el archivo de configuración *solamente si* el valor de otra propiedad cumple una condición determinada. Por ejemplo, `PollQuantity` aparece en la plantilla sólo si `JMS` es el mecanismo de transporte y `DuplicateEventElimination` tiene el valor `True`.

Para designar una propiedad como dependiente y definir la condición de la que depende, siga estos pasos:

1. En la pantalla **Propiedades disponibles**, seleccione la propiedad que desea definir como dependiente.
2. En el campo **Seleccionar propiedad**, utilice el menú desplegable para seleccionar la propiedad donde estará contenido el valor condicional.
3. En el campo **Operador de condición**, seleccione una de estas opciones:
  - == (igual a)
  - != (no igual a)
  - > (mayor que)
  - < (menor que)
  - >= (mayor o igual que)
  - <=(menor o igual que)
4. En el campo **Valor condicional**, escriba el valor que se exige para que la propiedad dependiente se incluya en la plantilla.
5. En la pantalla **Propiedades disponibles**, resalte la propiedad dependiente y pulse sobre una flecha para trasladar la propiedad a la pantalla **Propiedad dependiente**.

6. Pulse **Finalizar**. El Configurador de conectores guarda, en forma de documento XML, la información entrada por el usuario, en el subdirectorio `\data\app` del directorio `\bin` donde se ha instalado el Configurador de conectores.

---

## Creación de un archivo de configuración nuevo

Cuando crea un nuevo archivo de configuración, debe asignarle un nombre y seleccionar un intermediario de integración.

- En la ventana del Gestor de tareas, pulse con el botón derecho del ratón en la carpeta **Conectores** y seleccione **Crear conector nuevo**. Se abrirá la ventana del Configurador de conectores, que muestra el recuadro de diálogo **Conector nuevo**.
- En la modalidad autónoma: desde el Configurador de conectores, seleccione **Archivo > Nuevo > Configuración de conectores**. En la ventana Conector nuevo, escriba el nombre del conector nuevo.

Puede también ser necesario que seleccione un intermediario de integración. El intermediario que seleccione determinará las propiedades que aparecerán en el archivo de configuración. Para seleccionar un intermediario:

- En el campo **Intermediario de integración**, seleccione conectividad ICS, WebSphere Message Brokers o WAS.
- Complete los campos restantes de la ventana **Conector nuevo**, tal como se describe más adelante en este capítulo.

## Creación de un archivo de configuración a partir de una plantilla específica del conector

Una vez creada una plantilla específica del conector, puede utilizarla para crear un archivo de configuración:

1. Pulse **Archivo > Nuevo > Configuración del conector**.
2. Se abrirá el recuadro de diálogo **Conector nuevo**, que muestra los campos siguientes:
  - **Nombre**  
Escriba el nombre del conector. Existe distinción entre mayúsculas y minúsculas para los nombres. El nombre que especifique debe ser exclusivo y ser coherente con el nombre de archivo correspondiente a un conector que esté instalado en el sistema.  
  
**Importante:** El Configurador de conectores no comprueba la ortografía del nombre que el usuario especifique. Debe asegurarse de que el nombre es correcto.
  - **Conectividad del sistema**  
Pulse una de estas opciones: ICS, WebSphere Message Brokers o WAS.
  - **Seleccionar plantilla de propiedades específicas del conector**  
Escriba el nombre de la plantilla que se ha diseñado para su conector. Las plantillas disponibles se muestran en la pantalla **Nombre de plantilla**. Cuando selecciona un nombre en la pantalla Nombre de plantilla, la pantalla **Presentación preliminar de la plantilla de propiedades** muestra las propiedades específicas del conector que se han definido en esa plantilla.  
Seleccione la plantilla que desee utilizar y pulse **Aceptar**.
3. Se abrirá una pantalla de configuración para el conector que está configurando. La barra del título muestra el nombre del intermediario de integración y del



conector. Puede llenar todos los campos de valores para completar la definición ahora, o puede guardar el archivo y completar los campos más tarde.

4. Para guardar el archivo, pulse **Archivo > Guardar > en Archivo** o **Archivo > Guardar > en Proyecto**. Para guardar en un proyecto, el Gestor del sistema debe estar en ejecución.

Si guarda en un archivo, se abre el recuadro de diálogo **Guardar archivo del conector**. Seleccione \*.cfg como tipo de archivo y compruebe que en el campo Nombre de archivo esté escrito correctamente el nombre y el uso de mayúsculas y minúsculas es correcto. Vaya al directorio donde desea colocar el archivo y pulse **Guardar**. El estado mostrado en el panel de mensajes del Configurador de conectores indica que el archivo de configuración se ha creado satisfactoriamente.

**Importante:** La vía de acceso y nombre de directorio que defina aquí debe coincidir con la vía de acceso y nombre del archivo de configuración de conector que ha proporcionado en el archivo de inicio del conector.

5. Para completar la definición del conector, especifique valores en los campos para cada una de las pestañas de la ventana del Configurador de conectores, tal como se describe más adelante en este capítulo.

---

## Utilización de un archivo existente

Puede estar disponible en su sistema un archivo con uno o más de los formatos siguientes:

- Un archivo de definición de conector.  
Es un archivo de texto con una lista de propiedades y valores por omisión aplicables para un conector determinado. Algunos conectores incluyen un archivo de esa clase en un directorio llamado `\repository` del paquete del producto (normalmente la extensión del archivo es `.txt`; por ejemplo, `CN_XML.txt` para el conector de XML).
- Un archivo de depósito de ICS.  
Las definiciones utilizadas en una implementación anterior del conector para ICS pueden estar disponibles en un archivo de depósito que se utilizó en la configuración de ese conector. Un archivo de esa clase tiene normalmente la extensión `.in` o `.out`.
- Un archivo de configuración anterior del conector.  
Normalmente este archivo tiene la extensión `.cfg`.

Aunque cualquiera de estas fuentes de archivos puede contener la mayoría o todas las propiedades específicas del conector, el archivo de configuración del conector no estará completo hasta que haya abierto el archivo y definido las propiedades, tal como se describe más adelante en este capítulo.

Para utilizar un archivo existente para configurar el conector, debe abrir el archivo en el Configurador de conectores, revisar la configuración y luego volver a guardar el archivo.

Siga estos pasos para abrir un archivo \*.txt, \*.cfg o \*.in situado en un directorio:

1. En el Configurador de conectores, pulse **Archivo > Abrir > Desde archivo**.
2. En el recuadro de diálogo **Abrir archivo del conector**, seleccione uno de los tipos de archivos siguientes para ver los archivos disponibles:
  - Configuración (\*.cfg)
  - Depósito de ICS (\*.in, \*.out)

Seleccione esta opción si se utilizó un archivo de depósito para configurar el conector en un entorno ICS. Un archivo de depósito puede contener varias definiciones de conector, todas las cuales se muestran al abrir el archivo.

- Todos los archivos (\*.\*)

Seleccione esta opción si el paquete del adaptador contenía un archivo \*.txt para el conector, o si existe un archivo de definición con otra extensión.

3. En la pantalla de directorios, vaya hasta el archivo de definición de conector apropiado, selecciónelo y pulse **Abrir**.

Siga estos pasos para abrir una configuración de conector desde un proyecto del Gestor del sistema:

1. Inicie el Gestor del sistema. Es necesario que el Gestor del sistema esté en ejecución para abrir una configuración desde el Gestor del sistema o guardarla en él.
2. Inicie el Configurador de conectores.
3. Pulse **Archivo > Ovrir > Desde proyecto**.

---

## Finalización de un archivo de configuración

Cuando abre un archivo de configuración o un conector desde un proyecto, la ventana del Configurador de conectores muestra la pantalla de configuración, con los atributos y valores actuales.

El título de la pantalla de configuración muestra el nombre del intermediario de integración y del conector, tal como están especificados en el archivo. Asegúrese de que tiene el intermediario correcto. Si no es correcto, cambie el valor del intermediario antes de configurar el conector. Para ello siga estos pasos:

1. En la pestaña **Propiedades estándar**, seleccione el campo de valor correspondiente a la propiedad BrokerType. En el menú desplegable, seleccione el valor ICS, WMQI o WAS.
2. La pestaña Propiedades estándar mostrará las propiedades asociadas al intermediario seleccionado. Puede guardar el archivo ahora o completar los campos de configuración restantes, tal como se describe en "Especificación de definiciones de objetos comerciales compatibles" en la página 131.
3. Cuando termine de realizar la configuración, pulse **Archivo > Guardar > En proyecto** o bien **Archivo > Guardar > En archivo**.

Si guarda la configuración en un archivo, seleccione \*.cfg como extensión, seleccione la ubicación correcta para el archivo y pulse **Guardar**.

Si están abiertas varias configuraciones de conector, pulse **Guardar todo en archivo** para guardar todas las configuraciones en un archivo, o pulse **Guardar todo en proyecto** para guardar todas las configuraciones de conector en un proyecto del Gestor del sistema.

Antes de guardar el archivo, el Configurador de conectores comprueba que se hayan definido valores para todas las propiedades estándar necesarias. Si falta un valor para una propiedad estándar necesaria, el Configurador de conectores muestra un mensaje para indicar que la validación ha fallado. El usuario debe proporcionar un valor para la propiedad a fin de guardar el archivo de configuración.

---

## Definición de propiedades en el archivo de configuración

Cuando el usuario crea y designa un nuevo archivo de configuración de conector, o cuando abre uno existente, el Configurador de conectores muestra una pantalla de configuración con pestañas correspondientes a las categorías de valores de configuración necesarios.

El Configurador de conectores solicita valores para las propiedades comprendidas en las categorías siguientes para los conectores que se ejecutan en todos los intermediarios:

- Propiedades estándar
- Propiedades específicas del conector
- Objetos comerciales compatibles
- Valores de archivo de rastreo/anotaciones cronológicas
- Gestor de datos (válido para conectores que hacen uso de la mensajería JMS con entrega de sucesos garantizada)

**Nota:** Para los conectores que hacen uso de la mensajería JMS, se puede visualizar una categoría adicional para configurar los gestores de datos que convierten datos en objetos comerciales.

Para los conectores que se ejecutan en ICS, son también necesarios valores para estas propiedades:

- Correlaciones asociadas
- Recursos
- Mensajería (cuando sea aplicable)

**Importante:** El Configurador de conectores acepta valores de propiedad expresados utilizando juegos de caracteres ingleses o no ingleses. Pero los nombres de las propiedades estándar, los nombres de las propiedades específicas del conector y los nombres de los objetos comerciales compatibles se deben expresar utilizando solamente el juego de caracteres inglés.

Las propiedades estándar difieren de las propiedades específicas del conector en estos aspectos:

- Las propiedades estándar son comunes al componente específico de la aplicación de un conector y a su componente intermediario. Todos los conectores tienen el mismo conjunto de propiedades estándar. Estas propiedades se describen en el Apéndice A de la guía respectiva del adaptador. Puede cambiar algunos de estos valores, pero no todos.
- Las propiedades específicas del conector son aplicables solamente al componente específico de la aplicación del conector, es decir, el componente que interacciona directamente con la aplicación. Cada conector tiene propiedades específicas de la aplicación que son exclusivas de la aplicación del conector. Algunas de estas propiedades proporcionan valores por omisión y algunas no; puede modificar algunos de los valores por omisión. En la guía de cada adaptador, los capítulos dedicados a la instalación y configuración describen las propiedades específicas de la aplicación y los valores recomendados.

Los campos para las **Propiedades estándar** y las **Propiedades específicas del conector** muestran colores que denotan qué propiedades son configurables:

- Un campo con un fondo gris identifica una propiedad estándar. Puede cambiar el valor, pero no puede cambiar el nombre ni eliminar la propiedad.
- Un campo con un fondo blanco identifica una propiedad específica de la aplicación. Estas propiedades varían de acuerdo con las necesidades específicas de la aplicación o conector. Puede cambiar el valor y suprimir estas propiedades.
- Los campos de valor son configurables.
- El campo **Método de actualización** se muestra para cada propiedad. Este campo indica si es necesario reiniciar un componente o agente para activar valores modificados. El usuario no puede configurar este valor.

## Definición de propiedades estándar de conector

Para cambiar el valor de una propiedad estándar:

1. Pulse el campo cuyo valor desee definir.
2. Escriba un valor, o seleccione uno en el menú desplegable, si hubiera.
3. Después de entrar todos los valores para las propiedades estándar, puede seguir una de estas opciones:
  - Para descartar los cambios, conservar los valores originales y salir del Configurador de conectores, pulse **Archivo > Salir** (o cierre la ventana) y pulse **No** cuando se le invite a guardar los cambios.
  - Para entrar valores para otras categorías del Configurador de conectores, seleccione la pestaña correspondiente a la categoría. Los valores que especifique para las **Propiedades estándar** (o cualquier otra categoría) se conservan cuando pasa a la categoría siguiente. Cuando cierra la ventana, se le solicita si desea guardar o descartar los valores que ha entrado en todas las categorías consideradas globalmente.
  - Para guardar los valores revisados, pulse **Archivo > Salir** (o cierre la ventana) y pulse **Sí** cuando se le invite a guardar los cambios. Como alternativa, pulse **Guardar > En archivo** en el menú Archivo o en la barra de herramientas.

## Definición de propiedades de configuración específicas de la aplicación

Para las propiedades de configuración específicas de la aplicación, puede añadir o cambiar nombres de propiedades, configurar valores, suprimir una propiedad y cifrar una propiedad. La longitud por omisión de una propiedad es 255 caracteres.

1. Pulse con el botón derecho del ratón sobre la parte superior izquierda de la tabla. Se mostrará una barra de menús emergentes. Pulse **Agregar** para añadir una propiedad. Para añadir una propiedad hija, pulse con el botón derecho del ratón sobre el número de fila del padre y pulse **Agregar hijo**.
2. Especifique un valor para la propiedad o propiedad hija.
3. Para cifrar una propiedad, seleccione la casilla **Cifrar**.
4. Elija guardar o descartar los cambios, tal como se describe en “Definición de propiedades estándar de conector”.

El Método de actualización mostrado para cada propiedad indica si es necesario reiniciar un componente o agente para activar valores modificados.

**Importante:** Cambiar el nombre predefinido de una propiedad de conector específica de la aplicación puede hacer que el conector falle. Determinados nombres de propiedad pueden ser necesarios para que el conector se conecte a una aplicación o se ejecute correctamente.

## Cifrado de propiedades de conector

Puede cifrar propiedades específicas de la aplicación seleccionando la casilla **Cifrar** en la ventana Propiedades específicas del conector. Para descifrar un valor, pulse en la casilla **Cifrar** para quitar la marca de selección, escriba el valor correcto en el recuadro de diálogo **Verificación** y pulse **Aceptar**. Si el valor entrado es correcto, el valor se descifra y se muestra.

La guía del usuario del adaptador correspondiente a cada conector contiene una lista y descripción de las propiedades y su valor por omisión.

Si una propiedad tiene varios valores, la casilla **Cifrar** se mostrará para el primer valor de la propiedad. Cuando selecciona **Cifrar**, se cifran todos los valores de la propiedad. Para descifrar varios valores de una propiedad, pulse en la casilla **Cifrar** para quitar la marca de selección para el primer valor de la propiedad, y luego especifique el nuevo valor en el recuadro de diálogo **Verificación**. Si el valor entrado es correcto, se descifran todos los valores.

## Método de actualización

Consulte las descripciones de métodos de actualización que se encuentran en la sección "Definición y actualización de valores de propiedades" en la página 102 del apéndice *Propiedades de configuración estándar para conectores*.

## Especificación de definiciones de objetos comerciales compatibles

Utilice la pestaña **Objetos comerciales compatibles** del Configurador de conectores para especificar los objetos comerciales que utilizará el conector. Debe especificar objetos comerciales genéricos y específicos de la aplicación, así como asociaciones para las correlaciones entre objetos comerciales.

**Nota:** Algunos conectores requieren que determinados objetos comerciales se especifiquen como compatibles para realizar la notificación de sucesos o tareas adicionales de configuración (utilizando metaobjetos) con sus aplicaciones. Para obtener más información, consulte los manuales *Connector Development Guide for C++* o *Connector Development Guide for Java*.

## Cuando el intermediario de integración es ICS

Para especificar que una definición de objeto comercial es compatible con el conector, o para cambiar los valores admitidos para una definición existente de objeto comercial, pulse la pestaña **Objetos comerciales compatibles** y utilice los campos siguientes.

**Nombre de objeto comercial:** Para indicar que una definición de objeto comercial es compatible con el conector, efectúe lo siguiente con el Gestor del sistema en ejecución:

1. Pulse un campo vacío de la lista **Nombre de objeto comercial**. Se abrirá una lista desplegable, que muestra todas las definiciones de objeto comercial existentes en el proyecto del Gestor del sistema.
2. Pulse un objeto comercial para añadirlo.
3. Defina el **Soporte de agente** (descrito más abajo) para el objeto comercial.
4. En el menú Archivo de la ventana del Configurador de conectores, pulse **Guardar en proyecto**. La definición de conector revisada, incluido el soporte designado para la definición de objeto comercial añadida, se guarda en un proyecto de ICL (Integration Component Library) en el Gestor del sistema.

Para suprimir un objeto comercial de la lista de objetos compatibles:

1. Para seleccionar un campo de objeto comercial, pulse el número situado a la izquierda del objeto comercial.
2. En el menú **Editar** de la ventana del Configurador de conectores, pulse **Suprimir fila**. El objeto comercial se suprime de la lista visualizada.
3. En el menú **Archivo**, pulse **Guardar en proyecto**.

Suprimir un objeto comercial de la lista de objetos compatibles cambia la definición de conector y hace que el objeto comercial suprimido no pueda ser utilizado en la implementación actual del conector. La operación no afecta al código del conector, ni elimina la propia definición del objeto comercial en el Gestor del sistema.

**Soporte de agente:** Si un objeto comercial tiene Soporte de agente, el sistema intentará utilizar ese objeto comercial para entregar datos a una aplicación a través del agente de conector.

Normalmente, los objetos comerciales específicos de la aplicación correspondientes a un conector son compatibles con el agente de ese conector, pero no así los objetos comerciales genéricos.

Para indicar que el objeto comercial es compatible con el agente de conector, seleccione la casilla **Soporte de agente**. La ventana del Configurador de conectores no valida las selecciones hechas por el usuario respecto al Soporte de agente.

**Nivel máximo de transacciones:** El nivel máximo de transacciones de un conector es el nivel más alto de transacciones que el conector admite.

Para la mayoría de los conectores, la única opción posible es Nivel óptimo.

Debe reiniciar el servidor para que los cambios en el nivel de transacciones entre en vigor.

### **Cuando el intermediario de integración es WebSphere Message Broker**

Si está trabajando en la modalidad autónoma (no conectado al Gestor del sistema), debe entrar manualmente el nombre del objeto comercial.

Si el Gestor del sistema está en ejecución, puede seleccionar la casilla vacía situada bajo la columna **Nombre de objeto comercial**, en la pestaña **Objetos comerciales compatibles**. Se abrirá un cuadro combinado que muestra una lista de los objetos comerciales disponibles en el proyecto de ICL (Integration Component Library) al que pertenece el conector. Seleccione el objeto comercial que desee de la lista.

El campo **ID de conjunto de mensajes** es opcional para WebSphere Business Integration Message Broker 5.0, y no es necesario que sea un valor exclusivo si se cumplimenta ese campo. En cambio, para WebSphere MQ Integrator e Integrator Broker 2.1, debe proporcionar un **ID** exclusivo.

### **Cuando el intermediario de integración es WAS**

Cuando se selecciona WebSphere Application Server como tipo de intermediario utilizado, el Configurador de conectores no necesita los ID de conjunto de mensajes. La pestaña **Objetos comerciales compatibles** muestra la columna **Nombre de objeto comercial** sólo para objetos comerciales compatibles.

Si está trabajando en la modalidad autónoma (no conectado al Gestor del sistema), debe entrar manualmente el nombre del objeto comercial.



Si el Gestor del sistema está en ejecución, puede seleccionar la casilla vacía situada bajo la columna Nombre de objeto comercial, en la pestaña Objetos comerciales compatibles. Se abrirá un cuadro combinado que muestra una lista de los objetos comerciales disponibles en el proyecto de ICL (Integration Component Library) al que pertenece el conector. Seleccione el objeto comercial que desee de la lista.

## Correlaciones asociadas (ICS solamente)

Cada conector admite una lista de definiciones de objetos comerciales y sus correlaciones asociadas que están activas actualmente en WebSphere InterChange Server. Esta lista se muestra cuando selecciona la pestaña **Correlaciones asociadas**.

La lista de objetos comerciales contiene el objeto comercial específico de la aplicación que es compatible con el conector y el correspondiente objeto genérico que el controlador envía a la colaboración suscriptora. La asociación de una correlación determina la correlación que se utilizará para transformar el objeto comercial específico de la aplicación en el objeto comercial genérico o viceversa.

Si está utilizando correlaciones que están definidas de forma exclusiva para determinados objetos comerciales de origen y de destino, las correlaciones ya estarán asociadas con sus objetos comerciales apropiados cuando abra la pantalla, y no necesitará (o no podrá) cambiarlas.

Si existe más de una correlación disponible para ser utilizada por un objeto comercial compatible, necesitará vincular explícitamente el objeto comercial con la correlación que el objeto debe utilizar.

La pestaña **Correlaciones asociadas** muestra los campos siguientes:

- **Nombre de objeto comercial**

Indica los objetos comerciales compatibles con el conector, tal como están designados en la pestaña **Objetos comerciales compatibles**. Si designa más objetos comerciales en la pestaña **Objetos comerciales compatibles**, aparecerán en esta lista después de guardar los cambios seleccionando **Guardar en proyecto** en el menú **Archivo** de la ventana del Configurador de conectores.

- **Correlaciones asociadas**

La pantalla muestra todas las correlaciones que se han instalado en el sistema para ser utilizadas con los objetos comerciales compatibles del conector. El objeto comercial de origen de cada correlación aparece a la izquierda del nombre de correlación, en la pantalla **Nombre de objeto comercial**.

- **Explícito**

En algunos casos puede ser necesario que vincule explícitamente una correlación asociada.

La vinculación explícita sólo es necesaria cuando existe más de una correlación para un determinado objeto comercial compatible. Cuando se inicia, ICS intenta vincular automáticamente una correlación con cada objeto comercial compatible para cada conector. Si un mismo objeto comercial es utilizado como entrada por más de una correlación, el servidor intenta localizar y vincular una sola correlación que incluya a las demás.

Si no existe una correlación que abarque a las demás, el servidor no podrá vincular el objeto comercial a una sola correlación, y deberá definir una vinculación explícita.

Para vincular una correlación explícitamente:

1. En la columna **Explícito**, marque la casilla correspondiente a la correlación que desee vincular.

2. Seleccione la correlación que desee asociar al objeto comercial.
3. En el menú **Archivo** de la ventana del Configurador de conectores, pulse **Guardar en proyecto**.
4. Despliegue el proyecto en ICS.
5. Reinicie el servidor para que los cambios entren en vigor.

## Recursos (ICS)

La pestaña **Recurso** le permite definir un valor que determina si, y en qué medida, el agente de conector gestionará varios procesos simultáneamente, utilizando el paralelismo de agente de conector.

No todos los conectores pueden utilizar esta característica. Si está ejecutando un agente de conector que se diseñó en Java para ser de ejecución paralela, no es aconsejable que utilice esta característica, pues a menudo es más eficiente utilizar varias hebras en lugar de varios procesos.

## Mensajería (ICS)

Las propiedades de mensajería sólo pueden utilizarse si ha definido MQ como valor de la propiedad estándar `DeliveryTransport` e ICS como tipo de intermediario de integración. Estas propiedades afectan a la forma en que el conector utilizará las colas.

## Definición de valores de archivo de rastreo/anotaciones cronológicas

Cuando el usuario abre un archivo de configuración o archivo de definición de un conector, el Configurador de conectores utiliza los valores de anotaciones cronológicas y de rastreo de ese archivo como valores por omisión. El usuario puede cambiar esos valores en el Configurador de conectores.

Para cambiar los valores de anotaciones cronológicas y de rastreo:

1. Pulse la pestaña **Archivos de rastreo/anotaciones cronológicas**.
2. Para las anotaciones cronológicas o el rastreo, puede elegir escribir mensajes en una o en ambas de las ubicaciones siguientes:

- En la consola (STDOUT):  
Los mensajes de anotaciones cronológicas o de rastreo se escriben en la pantalla STDOUT.

**Nota:** sólo puede utilizar la opción STDOUT desde la pestaña **Archivos de rastreo/anotaciones cronológicas** para los conectores que se ejecutan en la plataforma Windows.

- En archivo:  
Los mensajes de anotaciones cronológicas o de rastreo se escriben en un archivo especificado por el usuario. Para especificar el archivo, pulse el botón de directorio (puntos suspensivos), vaya a la ubicación preferida, proporcione un nombre de archivo y pulse **Guardar**. Los mensajes de anotaciones cronológicas o rastreo se escribirán en el archivo y ubicación que especifique.

**Nota:** Los archivos de anotaciones cronológicas y de rastreo son archivos de texto simple. Puede utilizar la extensión de archivo que prefiera cuando defina los nombres de esos archivos. Sin embargo, para los archivos de rastreo es aconsejable utilizar la extensión `.trace` en lugar de `.trc`, para evitar la confusión con otros archivos que puedan residir



en el sistema. Para los archivos de anotaciones cronológicas, son habituales las extensiones de archivo .log y .txt.

## Gestores de datos

La sección referente a los gestores de datos sólo se puede configurar si ha definido JMS como valor de DeliveryTransport y como valor de ContainerManagedEvents. No todos los adaptadores utilizan gestores de datos.

Consulte las descripciones del Apéndice A correspondientes ContainerManagedEvents, en Propiedades estándar, para conocer los valores que se deben utilizar para estas propiedades. Para obtener más información, consulte los manuales *Connector Development Guide for C++* o *Connector Development Guide for Java*.

---

## Guardar el archivo de configuración

Cuando termine de configurar el conector, guarde el archivo de configuración del conector. El Configurador de conectores guarda el archivo utilizando la modalidad de intermediario que el usuario seleccionó durante la configuración. La barra del título del Configurador de conectores muestra siempre la modalidad de intermediario (ICS, WMQI o WAS) que está utilizando actualmente.

El archivo se guarda como documento XML. Puede guardar el documento XML de tres maneras:

- Desde el Gestor del sistema, como archivo con extensión \*.con en ICL (Integration Component Library) o bien
- En un directorio especificado por el usuario.
- En la modalidad autónoma, como archivo con extensión \*.cfg en una carpeta de directorios. Por omisión, el archivo se guarda en \WebSphereAdapters\bin\Data\App.
- Puede también guardar el archivo en un proyecto de WebSphere Application Server, si ha configurado uno.

Para conocer detalles sobre el uso de proyectos en el Gestor del sistema, y para obtener más información sobre el despliegue, consulte las siguientes guías de implementación:

- Para ICS: *Implementation Guide for WebSphere InterChange Server*
- Para WebSphere Message Brokers: *Implementing Adapters with WebSphere Message Brokers*
- Para WAS: *Implementing Adapters with WebSphere Application Server*

---

## Modificación de un archivo de configuración

Puede cambiar el valor correspondiente al intermediario de integración para un archivo de configuración existente. Esto le permite utilizar el archivo como plantilla para crear un nuevo archivo de configuración, que puede ser utilizado con un intermediario diferente.

**Nota:** Si cambia de intermediario de integración, necesitará cambiar otras propiedades de configuración además de la propiedad referente a la modalidad de intermediario.

Para cambiar el intermediario seleccionado dentro de un archivo de configuración existente (opcional):

- Abra el archivo de configuración existente en el Configurador de conectores.
- Seleccione la pestaña **Propiedades estándar**.
- En el campo **BrokerType** de la pestaña Propiedades estándar, seleccione el valor correspondiente al intermediario.  
Cuando cambie el valor actual, las pestañas disponibles y selecciones de campos en la pantalla de propiedades cambiarán de inmediato y se mostrarán solamente las pestañas y campos pertenecientes al nuevo intermediario seleccionado.

---

## Finalización de la configuración

Una vez creado y modificado un archivo de configuración para un conector, compruebe que el conector puede localizar el archivo de configuración cuando se inicia el conector.

Para ello, abra el archivo de inicio del conector y verifique que el nombre de archivo y la ubicación utilizados para el archivo de configuración del conector coinciden exactamente con el nombre que ha dado al archivo y con el directorio o vía de acceso donde ha situado el archivo.

---

## Uso del Configurador de conectores en un entorno globalizado

El Configurador de conectores está globalizado y puede realizar la conversión de caracteres entre el archivo de configuración y el intermediario de integración. El Configurador de conectores utiliza la codificación nativa. Cuando escribe en el archivo de configuración, utiliza la codificación UTF-8.

El Configurador de conectores admite el uso de caracteres no ingleses en:

- Todos los campos de valores
- La vía de acceso del archivo de anotaciones cronológicas y del archivo de rastreo (especificada en la pestaña **Archivos de rastreo/anotaciones cronológicas**)

La lista desplegable para las propiedades de configuración estándar CharacterEncoding y Locale sólo muestra un subconjunto de valores admitidos. Para añadir otros valores a la lista desplegable, debe modificar manualmente el archivo `\Data\Std\stdConnProps.xml` en el directorio del producto.

Por ejemplo, para añadir el entorno local en\_GB a la lista de valores para la propiedad Locale, abra el archivo `stdConnProps.xml` y añada la línea que aparece más abajo en letra negrita:

```
<Property name="Locale"
isRequired="true"
updateMethod="component restart">
 <ValidType>String</ValidType>
 <ValidValues>
 <Value>ja_JP</Value>
 <Value>ko_KR</Value>
 <Value>zh_CN</Value>
 <Value>zh_TW</Value>
 <Value>fr_FR</Value>
 <Value>de_DE</Value>
 <Value>it_IT</Value>
 <Value>es_ES</Value>
 <Value>pt_BR</Value>
 <Value>en_US</Value>
 <Value>en_GB</Value>
 </ValidValues>
 <DefaultValue>en_US</DefaultValue>
</Property>
```

---

## Apéndice C. Utilización de valores nulos y valores en blanco

Este apéndice describe diversas situaciones aprobadas y de error en las que el valor de clave de un objeto comercial es un blanco o nulo. Este apéndice también describe los cambios funcionales necesarios para valores en blanco o nulos de objetos comerciales.

---

### Situaciones aprobadas y de error

Si un valor de clave de un objeto comercial está en blanco o tiene un valor nulo en la base de datos, utilice el tipo "is null" en lugar del tipo de operador "=" para crear la cláusula WHERE.

IBM recomienda que los objetos comerciales tengan al menos un atributo de clave que no tenga un valor en blanco.

El supuesto siguiente corresponde a un objeto padre con una clave que tiene un valor nulo. Esta situación produce un error en estas condiciones:

*Tabla 18. Customer*

| Atributo | Tipo          |
|----------|---------------|
| cid      | Integer (Key) |
| name     | String        |
| comments | String        |

El supuesto siguiente corresponde a un objeto padre con dos claves y una de ellas tiene un valor nulo. Esta situación no produce error en estas condiciones:

*Tabla 19. Customer*

| Atributo | Tipo          |
|----------|---------------|
| cid      | Integer (Key) |
| name     | String        |
| comments | String        |

En el segundo supuesto, cree la consulta de recuperación de datos seleccionando los atributos "cid", "name" y "comments" de "customer", donde cid=1000 y "name" tiene un valor nulo.

El supuesto siguiente corresponde a un objeto padre con un objeto hijo en un objeto contenedor con una referencia de clave foránea. Esta situación produce un error en estas condiciones:

*Tabla 20. Customer*

| Atributo | Tipo          |
|----------|---------------|
| cid      | Integer (Key) |
| name     | String (Key)  |
| comments | String        |

Tabla 20. Customer (continuación)

| Atributo | Tipo                     |
|----------|--------------------------|
| Address  | Address                  |
| Aid      | Integer (Key) ASI:FK=cid |
| Acity    | String                   |
| Azip     | String                   |

Si cid contiene un valor nulo, cree la consulta de recuperación de datos seleccionando Aid, Acity y Azip de "address". Establezca el valor de Aid en nulo.

El supuesto siguiente corresponde a un objeto padre con un objeto hijo en un objeto contenedor con dos referencias de clave. Esta situación no produce error en estas condiciones:

Tabla 21. Customer

| Atributo | Tipo                     |
|----------|--------------------------|
| cid      | Integer (Key)            |
| name     | String                   |
| comments | String                   |
| Address  | Address                  |
| Aid      | Integer (Key) ASI:FK=cid |
| Acity    | String (Key) ASI:FK=name |
| Azip     | String                   |

Si "name" tiene un valor nulo, cree la sentencia de recuperación de datos seleccionando Aid, Acity y Azip de "address", donde Aid=Cid y Acity tiene un valor nulo.

---

## Funcionalidad

Si el conector encuentra un valor en blanco en una clave, compara ese valor con el valor de UseNull contenido en el atributo. Si el valor es "true", el conector añade un valor nulo a la consulta. Esto afecta a las siguientes operaciones de verbo:

- Retrieve
- RetrieveBy Content
- Update
- Delete

---

## Avisos

IBM puede que no comercialice en todos los países los productos, servicios o características tratados en el presente documento. Consulte al representante local de IBM para obtener información sobre los productos y servicios que se pueden adquirir actualmente en su zona geográfica. Cualquier referencia a un producto, programa o servicio de IBM no pretende afirmar ni implicar que sólo se pueda utilizar ese producto, programa o servicio de IBM. En su lugar se puede utilizar cualquier producto, programa o servicio funcionalmente equivalente que no vulnere ningún derecho de propiedad de intelectual de IBM. Si embargo, corresponde al usuario evaluar y verificar el funcionamiento de cualquier producto, programa o servicio que no sea de IBM.

IBM puede tener patentes o solicitudes de patente en tramitación que abarquen temas descritos en el presente documento. El suministro de este documento no le otorga ninguna licencia sobre esas patentes. Puede enviar solicitudes de información sobre licencias, por escrito, a esta dirección:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
EE.UU.

El párrafo siguiente no es aplicable al Reino Unido ni a ningún otro país donde tales disposiciones sean incompatibles con la legislación local:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROPORCIONA ESTA PUBLICACIÓN "TAL CUAL", SIN GARANTÍA DE NINGUNA CLASE, YA SEA EXPRESA O IMPLÍCITA, INCLUIDAS, PERO SIN LIMITARSE A ELLAS, LAS GARANTÍAS IMPLÍCITAS DE NO VULNERACIÓN DE DERECHOS, COMERCIALIZABILIDAD O ADECUACIÓN PARA UN FIN DETERMINADO. Algunos estados no permiten la renuncia de garantías expresas ni implícitas en determinadas transacciones, por lo que esta declaración puede no ser aplicable a su caso.

La presente información puede contener inexactitudes técnicas o errores tipográficos. Periódicamente se realizan cambios en la información aquí contenida; estos cambios se incorporarán a las nuevas ediciones de la publicación. En cualquier momento y sin previo aviso, IBM puede hacer mejoras y/o cambios en los productos y/o programas descritos en esta publicación.

Cualquier referencia en esta publicación a sitios Web no pertenecientes a IBM se proporciona solamente para la comodidad del usuario, y en ninguna manera constituye un aval de esos sitios Web. La información de esos sitios Web no forma parte de la información del presente producto de IBM y la utilización de esos sitios Web se realiza bajo la responsabilidad del usuario.

IBM puede utilizar o distribuir cualquier información que el usuario le proporcione de la manera que IBM considere apropiada, sin contraer ninguna obligación con el usuario.

Los licenciarios de este programa que deseen tener información sobre él con el fin de permitir: (i) el intercambio de información entre programas creados

independientemente y otros programas (incluido el presente programa) y (ii) el uso recíproco de la información que se ha intercambiado, deben ponerse en contacto con:

IBM Burlingame Laboratory Director  
IBM Burlingame Laboratory  
577 Airport Blvd., Suite 800  
Burlingame, CA 94010  
EE.UU.

Dicha información puede estar disponible, sujeta a los términos y condiciones apropiados, incluido en algunos casos, el pago de una tarifa.

El programa bajo licencia descrito en este documento y toda la información bajo licencia existente para él son proporcionados por IBM bajo los términos del Contrato del Cliente IBM, el Acuerdo Internacional de Licencia de Programas de IBM o cualquier otro acuerdo equivalente existente entre las partes.

Los datos de rendimiento contenidos en este manual se determinaron en un entorno controlado. Por tanto, los resultados obtenidos en otros entornos operativos pueden variar significativamente. Algunas mediciones se pueden haber realizado en sistema a nivel de desarrollo y no es seguro que estas mediciones serán las mismas en los sistemas disponibles normalmente. Además, algunas mediciones se pueden haber calculado mediante extrapolación. Los resultados reales pueden variar. Los usuarios de este documento deben verificar los datos aplicables para su entorno específico.

La información referente a productos que no son de IBM se ha obtenido de los proveedores de esos productos, de sus anuncios publicados o de otras fuentes disponibles públicamente. IBM no ha probado necesariamente esos productos y no puede confirmar la exactitud del rendimiento, la compatibilidad ni otras afirmaciones referentes a productos que no son de IBM. Las preguntas sobre las prestaciones de productos que no son de IBM se deben dirigir a los proveedores de esos productos.

La presente información puede contener ejemplos de datos e informes que se utilizan en operaciones comerciales diarias. Para ilustrarlos de la forma más completa posible, los ejemplos pueden incluir nombres de personas, empresas, marcas y productos. Todos estos nombres son ficticios y cualquier similitud con los nombres y direcciones utilizados por una empresa real es totalmente fortuita.

Todas las declaraciones de intenciones de IBM están sujetas a cambio y anulación sin previo aviso, y representan metas y objetivos solamente.

---

## Información de la interfaz de programación

La información de la interfaz de programación, si se proporciona, está pensada para ayudarle a crear software de aplicación utilizando el presente programa.

Las interfaces de programación de uso general le permiten escribir software de aplicación que obtiene los servicios de las herramientas de este programa.

Sin embargo, esta información puede también contener información de diagnóstico, modificación y ajuste del rendimiento. La información de diagnóstico, modificación y ajuste del rendimiento se proporciona para ayudarle a depurar su software de aplicación.

**Aviso:** No utilice esta información de diagnóstico, modificación y ajuste del rendimiento como interfaz de programación, pues está sujeta a cambio.

---

## Marcas registradas y marcas de servicio

Los términos siguientes son marcas registradas o marcas comerciales de International Business Machines Corporation en los Estados Unidos o en otros países:

IBM  
el logotipo de IBM  
AIX  
CrossWorlds  
DB2  
DB2 Universal Database  
Domino  
Lotus  
Lotus Notes  
MQIntegrator  
MQSeries  
Tivoli  
WebSphere

Microsoft, Windows, Windows NT y el logotipo de Windows son marcas registradas de Microsoft Corporation en los Estados Unidos o en otros países.

MMX, Pentium y ProShare son marcas registradas o marcas comerciales de Intel Corporation en los Estados Unidos o en otros países.

Java y todas las marcas registradas basadas en Java son marcas registradas de Sun Microsystems, Inc. en los Estados Unidos o en otros países.

Otros nombres de empresas, productos o servicios pueden ser marcas registradas o marcas de servicio de terceros.

Adapter for Manugistics incluye software desarrollado por Eclipse Project (<http://www.eclipse.org>)

WebSphere Business Integration Adapter Framework V2.4.0.

IBM WebSphere InterChange Server V4.2.1, IBM WebSphere Business Integration Toolset V4.2.1, IBM WebSphere Business Integration Adapters V2.3.1, IBM WebSphere Business Integration Collaborations V4.2.











Printed in Denmark by IBM Danmark A/S