

WebSphere Business Integration Server  
Express e Express Plus



# Guida per l'utente dell'adattatore per JDBC

*Versione 44*



WebSphere Business Integration Server  
Express e Express Plus



# Guida per l'utente dell'adattatore per JDBC

*Versione 44*

**Nota!**

Prima di utilizzare queste informazioni ed il prodotto supportato, consultare "Informazioni particolari" a pagina 151.

**22 aprile 2005**

Questa edizione della documentazione si applica all'adattatore Adapter per JDBC (5724-H05), versione 2.6.x.

Per inviare commenti sulla documentazione relativa a WebSphere Business Integration Server Express, inviare un e-mail al seguente indirizzo: [doc-comments@us.ibm.com](mailto:doc-comments@us.ibm.com). Siamo in attesa di ricevere i vostri commenti.

L'IBM può utilizzare o divulgare le informazioni ricevute dagli utenti secondo le modalità ritenute appropriate, senza nessun obbligo nei loro confronti.

© Copyright International Business Machines Corporation 2004, 2005. Tutti i diritti riservati.

---

# Indice

<b>Figure</b>	<b>v</b>
<b>Informazioni</b>	<b>vii</b>
A chi è diretto questo manuale	vii
Documentazione correlata	vii
Convenzioni tipografiche	vii
<b>Novità di questo rilascio</b>	<b>ix</b>
Novità nel rilascio 2.6.x	ix
<b>Capitolo 1. Panoramica del connettore</b>	<b>1</b>
Componenti del connettore	1
Funzionamento del connettore	2
<b>Capitolo 2. Installazione del connettore</b>	<b>9</b>
Ambiente dell'adattatore	9
Prerequisiti	9
Installazione dell'adattatore e dei file correlati	10
Struttura dei file installati	10
Struttura dei file installati su Linux	10
Struttura dei file installati per Windows	11
Struttura dei file installati per i5/OS	11
<b>Capitolo 3. Generazione delle definizioni dell'oggetto business utilizzando JDBC ODA</b>	<b>13</b>
Installazione ed esecuzione di JDBC ODA	13
Utilizzo di JDBC ODA in Business Object Designer Express	17
Contenuti della definizione generata	27
File di definizione dell'oggetto business di esempio	30
Inserimento di attributi che contengono oggetti business child	31
Aggiunta di informazioni alla definizione dell'oggetto business	31
<b>Capitolo 4. Comprensione degli oggetti aziendali</b>	<b>33</b>
Convenzioni di denominazione degli oggetti business e degli attributi	33
Struttura oggetto business	33
Elaborazione dell'espressione oggetto business	38
Proprietà degli attributi di un oggetto business	55
Informazioni specifiche dell'applicazione dell'oggetto business	57
<b>Capitolo 5. Configurazione del connettore</b>	<b>69</b>
Abilitazione del supporto multi-driver	69
Abilitazione della classe del gestore oggetti business personalizzati	69
Impostazione delle proprietà di configurazione del connettore	70
Utilizzo di Connector Configurator Express	71
Panoramica su Connector Configurator Express	71
Avvio di Connector Configurator Express	72
Esecuzione di Connector Configurator Express da System Manager	73
Creazione di una maschera delle proprietà specifiche per un connettore	73
Creazione di un nuovo file di configurazione	76
Utilizzo di un file esistente	77
Completare un file di configurazione	78
Impostazione delle proprietà del file di configurazione	78
Salvataggio del file di configurazione	84
Modifica di un file di configurazione	84
Completamento della configurazione	84

Utilizzando Connector Configurator Express in un ambiente globalizzato. . . . .	85
<b>Capitolo 6. Configurazione dell'applicazione . . . . .</b>	<b>87</b>
Abilitazione dell'applicazione per il connettore . . . . .	87
<b>Capitolo 7. Esecuzione del connettore . . . . .</b>	<b>91</b>
Avvio del connettore . . . . .	91
Arresto del connettore. . . . .	92
Creazione di più istanze del connettore . . . . .	93
<b>Capitolo 8. Risoluzione dei problemi e gestione degli errori . . . . .</b>	<b>97</b>
Avvio dei problemi. . . . .	97
Elaborazione degli eventi. . . . .	97
Mappatura di InterChange Server Express . . . . .	97
Gestione degli errori e registrazione . . . . .	98
Perdita della connessione all'applicazione . . . . .	100
Impossibilità di individuare le tabelle di archivio o di eventi quando viene utilizzato DB2. . . . .	100
Abilitazione del connettore per effettuare operazioni con un database DB2 . . . . .	101
JDBC ODA si comporta in modo non corretto a causa di un driver JDBC non supportato . . . . .	102
<b>Appendice A. Proprietà di configurazione standard per i connettori . . . . .</b>	<b>103</b>
Nuove proprietà . . . . .	103
Panoramica sulle proprietà del connettore standard . . . . .	103
Guida di riferimento rapido per le proprietà standard. . . . .	105
Proprietà standard. . . . .	110
<b>Appendice B. Proprietà di configurazione specifiche dei connettori . . . . .</b>	<b>127</b>
Dettagli proprietà . . . . .	128
<b>Appendice C. Esempi di oggetti business . . . . .</b>	<b>143</b>
AfterUpdateSPSampleBO.txt . . . . .	143
BeforeCreateSPSampleBO.txt . . . . .	143
BOwithDifferentParameterOrder.txt . . . . .	143
BOwithIOandOPParams.txt. . . . .	144
BOwithFewerSPParamsthanBOAttribs.txt . . . . .	144
CreateSPUpdateSPSampleBO.txt . . . . .	144
<b>Appendice D. Supporto per valori vuoti o nulli . . . . .</b>	<b>145</b>
Scenari riusciti o non riusciti . . . . .	145
Funzionalità . . . . .	146
<b>Appendice E. Mainframe Data Access . . . . .</b>	<b>147</b>
Supporto del connettore JDBC per mainframe data access . . . . .	147
Utilizzo di mainframe data access . . . . .	147
Componenti di mainframe data access . . . . .	148
<b>Indice analitico. . . . .</b>	<b>149</b>
<b>Informazioni particolari . . . . .</b>	<b>151</b>
Informazioni sull'interfaccia di programmazione . . . . .	152
Marchi e marchi di servizio. . . . .	153

---

## Figure

1. Architettura del connettore per le richieste dell'oggetto business . . . . .	2	8. Selezione degli attributi della procedura memorizzata . . . . .	25
2. Selezione di un ODA . . . . .	19	9. Associazione delle procedure memorizzate ai relativi attributi . . . . .	26
3. Configurazione delle proprietà di inizializzazione dell'agente . . . . .	20	10. Relazione con cardinalità singola tipica . . . . .	35
4. Struttura ad albero dello schema con nodi estesi. . . . .	21	11. Relazione di oggetti business con più cardinalità . . . . .	37
5. Estensione di ALL SCHEMAS per richiamare gli oggetti . . . . .	22	12. Oggetto business con cardinalità singola con la relazione memorizzata nel child. . . . .	37
6. Conferma della sezione di oggetti del database . . . . .	23	13. Esempio di relazioni tra gli oggetti business . . . . .	63
7. Ulteriori informazioni sugli oggetti del database . . . . .	24	14. Supporto mainframe data access . . . . .	147





---

## Informazioni

I prodotti IBM<sup>(R)</sup> WebSphere<sup>(R)</sup> Business Integration Server Express e IBM WebSphere Business Integration Server Express Plus sono costituiti dai seguenti componenti: InterChange Server Express, Toolset Express associato, CollaborationFoundation e una serie di adattatori di integrazione software. Gli strumenti di Toolset Express consentono di creare, modificare e gestire i processi di business. E' possibile scegliere tra diversi adattatori preconfigurati per i processi di business che utilizzano le applicazioni. La maschera dei processi standard, CollaborationFoundation, consente di creare rapidamente processi personalizzati.

In questo documento vengono descritte le attività di installazione, di configurazione delle proprietà dei connettori, dello sviluppo degli oggetti di business e della risoluzione dei problemi relativi a WebSphere Business Integration Server Express Adapter for JDBC.

Se non diversamente indicato, tutte le informazioni riportate in questo manuale sono valide per IBM WebSphere Business Integration Server Express e IBM WebSphere Business Integration Server Express Plus. Il termine "WebSphere Business Integration Server Express" e le relative varianti fa riferimento a entrambi i prodotti.

---

## A chi è diretto questo manuale

Questo documento è per consulenti, sviluppatori e amministratori di sistema che utilizzano l'adattatore per i siti cliente.

---

## Documentazione correlata

La serie completa di manuali disponibile con questo prodotto descrive le funzioni e i componenti comuni a tutte le installazioni di adattatori e comprende i materiali di riferimento per componenti specifici.

E' possibile scaricare, installare e visualizzare la documentazione al seguente indirizzo: <http://www.ibm.com/websphere/wbiserverexpress/infocenter>.

---

## Convenzioni tipografiche

Questo manuale utilizza le seguenti convenzioni:

Font courier	Indica un valore letterale, come il nome di un comando, un nome file, informazioni immesse dall'utente o le informazioni visualizzate sullo schermo.
<b>grassetto</b>	Indica un termine nuovo la prima volta che viene visualizzato.
<i>Corsivo</i>	Indica un nome variabile o un riferimento incrociato.
<i>riquadro blu</i>	Un riquadro blu, disponibile quando si visualizza un manuale online, indica un collegamento ipertestuale con riferimenti incrociati. Fare clic all'interno del riquadro per passare all'oggetto del riferimento.

---

---

<i>ProductDir</i>	Rappresenta la directory in cui è installato il prodotto WebSphere Business Integration Server Express. Il valore predefinito per ciascuna piattaforma è: Windows: IBM\WebSphereServer\ProductDir i5/OS: /QIBM/ProdData/WBIServer44/ProductDir Linux: /home/\${username}/IBM/WebSphereServer/ProductDir
{ }	Su una riga della sintassi, le parentesi graffe racchiudono una serie di opzioni di cui ne va selezionata solo una.
	In una riga della sintassi, una barra separa una serie di opzioni tra cui è possibile sceglierne una e solo una.
[ ]	Su una riga della sintassi, le parentesi racchiudono un parametro facoltativo.
...	In una riga di sintassi, le ellissi indicano una ripetizione del parametro precedente. Ad esempio, <code>option[,...]</code> significa che è possibile inserire più opzioni separate da virgola.
< >	Le parentesi angolari racchiudono i singoli elementi di un nome per distinguere gli uni dagli altri, come ad esempio <code>&lt;server_name&gt;&lt;connector_name&gt;tmp.log</code> .
/, \	In questo documento, le barre inverse (\) vengono utilizzate come convenzione per i percorsi di directory per Windows. Per le installazioni i5/OS e Linux <sup>(R)</sup> , sostituire le barre (/) alle barre inverse. Tutti i nomi di percorso del prodotto sono relativi alla directory in cui è installato il connettore per JDBC sul sistema.
<b>Linux/Windows:</b> <sup>(TM)</sup>	I paragrafi che iniziano con questi, indicano note che elencano differenze tra sistemi operativi.
<i>%testo% e \$testo</i>	Un testo all'interno dei segni percentuale (%) indica il valore della variabile di sistema <code>text</code> o della variabile utente. La notazione equivalente in ambiente Linux è il simbolo \$, <i>che indica</i> il valore del testo della variabile di ambiente Linux.

---

---

## Novità di questo rilascio

---

### Novità nel rilascio 2.6.x

Il rilascio di questo documento per l'adattatore versione 2.6.x contiene le seguenti informazioni nuove o corrette.

Questo rilascio dell'adattatore supporta l'elaborazione di dati di script bidirezionale. Per ulteriori informazioni, consultare "Dati dipendenti dalla locale" a pagina 6.

Questo rilascio dell'adattatore aggiunge supporto per le seguenti piattaforme o aggiornamenti:

- Microsoft Windows 2003 (Professional, Server o Advanced Server) con Service Pack 4
- Microsoft Windows 2003 (Standard Edition o Enterprise Edition)
- IBM i5/OS V5R3 e OS/400 V5R2

**Nota:** A meno che non diversamente specificato, i5/OS fa riferimento a OS/400 e i5/OS.

- Linux:
  - RedHat 2.1 Enterprise Linux WS/AS/ES 3.0 Update 2, Intel (IA32)
  - SuSE Linux ES 81., Intel (IA32)
  - SuSE Linux ES 9.0, Intel (IA32)

Questo rilascio aggiunge supporto per Prepared Statement Caching per l'ottimizzazione delle prestazioni. Comprende una nuova proprietà di configurazione specifica del connettore, PreparedStatement CacheSize. La memorizzazione nella cache delle istruzioni preparate è abilitata a livello di oggetto business mediante ASI per Prepared Statement Caching: PSC=true|false.

Questo rilascio aggiunge RetVal ad ASI per gli attributi della procedura memorizzata.

Nel Capitolo 4, in "Prima di utilizzare JDBC ODA" a pagina 14, gli esempi DRIVERLIB nella *Tabella 6. Variabili di configurazione dei file shell e batch* sono stati modificati per rendere più chiaro il modo in cui trovare tutti i file .dll o .so. Le informazioni sono state aggiunte dopo la tabella e indicano che è necessario aggiungere il driver del database a DRIVERPATH nello script di avvio.



---

## Capitolo 1. Panoramica del connettore

Questo capitolo descrive il componente del connettore di Server Express Adapter for JDBC. Nel sistema WebSphere Business Integration Server Express, l'adattatore fornisce i componenti per supportare la comunicazione tra il broker di integrazione e un'applicazione o una tecnologia (come ad esempio JDBC). Un adattatore comprende sempre un connettore, file di messaggi e strumenti di configurazione. Inoltre, può includere un ODA (Object Discovery Agent) o un gestore dati.

Questo capitolo contiene le sezioni seguenti:

- "Componenti del connettore"
- "Funzionamento del connettore" a pagina 2

---

### Componenti del connettore

Il **connettore** è il componente di un adattatore che utilizza oggetti aziendali per inviare informazioni su un evento ad un broker di integrazione (notifica evento) o ricevere informazioni su una richiesta da un broker di integrazione (elaborazione della richiesta). I connettori sono costituiti da due parti: la **struttura del connettore** e il **componente specifico dell'applicazione**.

La struttura del connettore, il cui codice è comune a tutti i connettori, gestisce le interazioni tra il broker di integrazione e il componente specifico dell'applicazione.

Il componente specifico dell'applicazione contiene il codice adatto ad una determinata applicazione o tecnologia (in questo caso, JDBC). La struttura del connettore fornisce i seguenti servizi tra il broker di integrazione e il componente specifico dell'applicazione:

- Riceve ed invia oggetti di business
- Gestisce lo scambio di messaggi di avvio e di gestione

Il connettore per JDBC abilita il broker di integrazione allo scambio degli oggetti aziendali con un'applicazione creata in un qualunque database supportato da un driver che segue JDBC 2.0 o una specifica successiva. Questa sezione presenta una descrizione dettagliata dell'architettura del connettore.

Per specificare il driver che verrà utilizzato dal connettore per connettersi al database, consultare "Abilitazione del supporto multi-driver" a pagina 69.

Il connettore si collega al database dell'applicazione utilizzando il meccanismo di connessione JDBC. Un parametro di configurazione specifico del connettore (DatabaseURL) consente di specificare il nome del server del database cui il connettore deve connettersi. Per informazioni sui parametri di configurazione, consultare "Impostazione delle proprietà di configurazione del connettore" a pagina 70.

Una volta avviato il connettore, viene stabilito un pool di connessione con il database. Utilizza le connessioni da questo pool per tutte le transazioni in elaborazione con il database. Una volta arrestato il connettore, tutte le connessioni nel pool vengono chiuse.

## Architettura del connettore

Figura 1 illustra i componenti del connettore e le relative relazioni tra il sistema di integrazione aziendale.

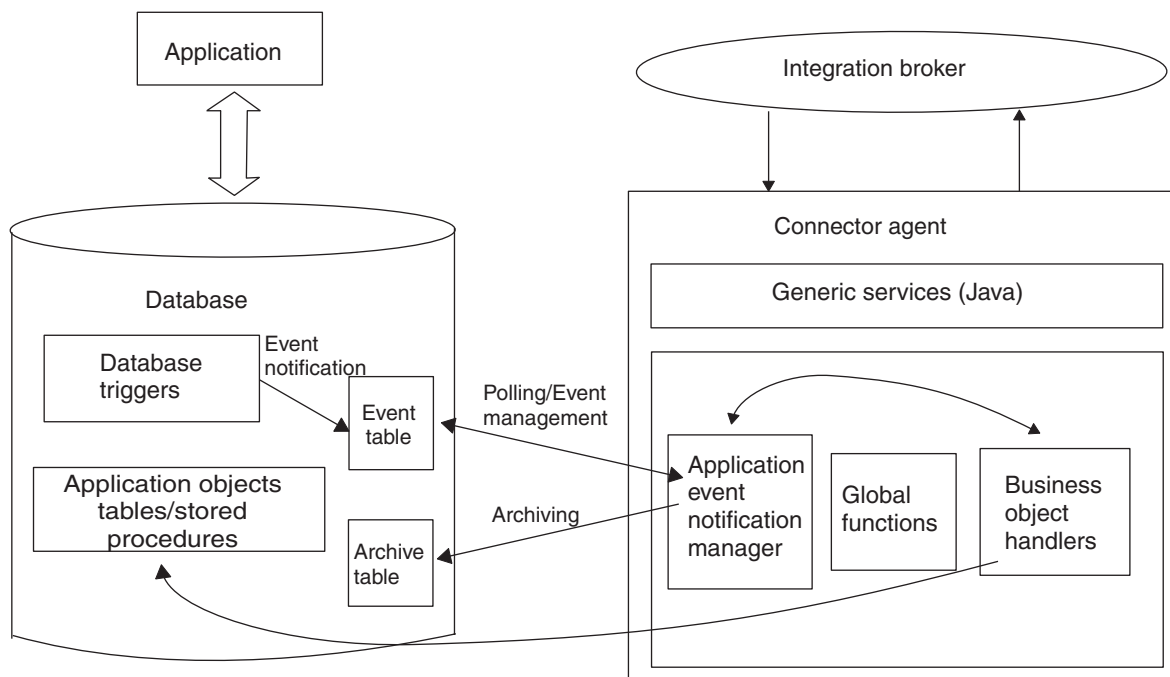


Figura 1. Architettura del connettore per le richieste dell'oggetto business

## Funzionamento del connettore

Questa sezione descrive il modo in cui i metadati migliorano la flessibilità del connettore, oltre a fornire una descrizione dettagliata dell'elaborazione e della notifica degli eventi dell'oggetto business.

### Connettore e metadati

Il connettore è basato sui metadati. **Metadati**, nell'ambiente WebSphere Business Integration Server Express Adapter, i dati specifici dell'applicazione vengono memorizzati negli oggetti aziendali e assistono il connettore nelle interazioni con l'applicazione. Un connettore basato sui metadati gestisce ciascun oggetto business supportato basato sui metadati codificati nella definizione dell'oggetto business piuttosto che sulle istruzioni cifrate nel connettore.

I metadati degli oggetti business include la struttura di un oggetto business, le impostazioni delle relative proprietà dell'attributo e il contenuto delle informazioni specifiche all'applicazione. Poiché il connettore è basato sui metadati, può gestire oggetti business nuovi o modificati senza richiedere modifiche al codice del connettore.

Il connettore esegue le istruzioni SQL o procedure memorizzate per richiamare o modificare i dati nel database o applicazione. Per creare istruzioni SQL dinamiche o procedure memorizzate, il connettore utilizza metadati specifici dell'applicazione. Queste istruzioni SQL e procedure memorizzate eseguono il recupero richiesto o le modifiche al database dell'applicazione per l'oggetto business e per l'espressione

che il connettore sta elaborando. Per informazioni sull'utilizzo delle informazioni specifiche all'applicazione, consultare Capitolo 4, "Comprensione degli oggetti aziendali", a pagina 33.

## Elaborazione dell'oggetto business

Questa sezione fornisce una panoramica del modo in cui il connettore elabora le richieste dell'oggetto business e gli eventi dell'applicazione. Per ulteriori informazioni, consultare "Elaborazione dell'espressione oggetto business" a pagina 38.

### Elaborazione delle richieste dell'oggetto business

Quando il connettore riceve una richiesta per eseguire un'operazione dell'applicazione, il connettore elabora gli oggetti business gerarchicamente in modo ricorrente, ovvero esegue gli stessi passi per ciascun oggetto business child fino a quando non ha elaborato tutti i singoli oggetti business. L'ordine in cui il connettore elabora gli oggetti business child e gli oggetti business di livello superiore dipende dagli oggetti business contenuti con o senza proprietà e se contenuti con una cardinalità singola o multipla.

**Nota:** Il termine oggetto business **gerarchico** fa riferimento ad un oggetto business completo, compresi tutti gli oggetti aziendali business child contenuti a qualunque livello. Il termine oggetto business **singolo** fa riferimento ad un oggetto business singolo, indipendentemente dagli oggetti business child che potrebbe contenere o che contiene. Il termine oggetto business **di livello superiore** fa riferimento all'oggetto business singolo all'inizio della gerarchia, che non dispone di un oggetto business parent.

**Richiamo di oggetti business:** Quando il broker di integrazione richiede al connettore di richiamare un oggetto business gerarchico dal database, il connettore tenta di restituire un oggetto business che corrisponde esattamente alla rappresentazione del database corrente di quell'oggetto business. In altre parole, tutti gli attributi semplici di ciascun oggetto business singolo restituito al broker di integrazione corrisponde al valore del campo corrispondente nel database. Inoltre, il numero degli oggetti business singoli in ciascuna serie contenuti dall'oggetto business restituito corrisponde al numero di child che si trovano nel database per questa serie.

Per eseguire tale recupero, il connettore utilizza i valori della chiave principale nell'oggetto business di livello superiore per passare in modo ricorrente ai dati in ordine decrescente corrispondenti nel database.

**Oggetto business RetrievalByContent:** Quando il broker di integrazione richiede al connettore di richiamare un oggetto business gerarchico basato sui valori negli attributi non chiave nell'oggetto business di livello superiore, il connettore utilizza il valore di tutti gli attributi non null come criteri per il richiamo dei dati.

**Creazione di un oggetto business:** Quando il broker di integrazione richiede al connettore di creare un oggetto business gerarchico, il connettore esegue i passi riportati:

1. Crea in modo ricorrente ciascun oggetto business child contenuto con la proprietà nel database.
2. Elaboro ciascun oggetto business child con cardinalità singola contenuto senza proprietà.
3. Crea l'oggetto business di livello superiore nel database.

4. Crea ciascun oggetto business child con cardinalità singola che memorizza la relazione parent-child nel child.
5. Crea ciascun oggetto business con cardinalità multipla.

**Modifica dell'oggetto business:** Quando il broker di integrazione richiede al connettore di aggiornare un oggetto business gerarchico, il connettore esegue i passi riportati:

1. Utilizza i valori della chiave principale dell'oggetto business per richiamare l'entità corrispondente dal database.
2. Aggiorna in modo ricorrente tutti i child con cardinalità singola dell'oggetto business di livello superiore.
3. Per una cardinalità singola con la proprietà che memorizza la relazione nel parent, viene impostato ciascun valore della chiave esterna nel parent al valore della chiave principale nell'oggetto business child con cardinalità singola.
4. Aggiorna tutti gli attributi semplici dell'oggetto business eccetto quali i cui attributi corrispondenti nell'oggetto business di origine contengono il valore CxIgnore.
5. Imposta tutti i valori delle chiavi esterne in ciascun child che memorizza la relazione parent-child nel child (cardinalità multipla e singola) con il valore della chiave principale dell'oggetto business parent corrispondente.
6. Elabora tutte le serie dell'oggetto business richiamato.

**Eliminazione dell'oggetto business:** Quando il broker di integrazione richiede al connettore di eliminare un oggetto business gerarchico dal database, il connettore procede nel modo seguente:

1. Elimina il child con cardinalità singola.
2. Elimina il child con cardinalità multipla.
3. Elimina l'oggetto business di livello superiore.

### **Elaborazione degli eventi dell'applicazione**

Il connettore gestisce gli eventi Crea, Aggiorna ed Elimina generati dall'applicazione nel modo descritto di seguito.

**Creazione della notifica:** Quando il connettore incontra un'operazione Crea evento nella tabella eventi, crea un oggetto business del tipo specificato dall'evento, imposta i valori chiave per l'oggetto business (utilizzando le chiavi specificate nella tabella eventi) e richiama l'oggetto business dal database. Una volta richiamato l'oggetto business, il connettore lo invia con l'operazione Crea espressione al broker di integrazione.

**Aggiornamento della notifica:** Quando il connettore incontra un'operazione Aggiorna evento nella tabella eventi, crea un oggetto business del tipo specificato dall'evento, imposta i valori chiave per l'oggetto business (utilizzando le chiavi specificate nella tabella eventi) e richiama l'oggetto business dal database. Una volta richiamato l'oggetto business, il connettore lo invia con l'operazione Aggiorna espressione al broker di integrazione.

**Notifica di eliminazione:** Quando il connettore incontra un'operazione Elimina evento nella tabella eventi, crea un oggetto business del tipo specificato dall'evento, imposta i valori chiave per l'oggetto business (utilizzando le chiavi specificate nella tabella eventi) e lo invia con il verbo Elimina al broker di integrazione. Tutti i valori diversi dai valori chiave sono impostati su CxIgnore. Se uno dei campi non chiave è significativo per la collocazione, il valore di tali campi può essere modificato in base alla necessità.



Il connettore gestisce le operazioni Elimina logiche e fisiche di cui viene eseguito il trigger dalla relativa applicazione. Nel caso di eliminazioni fisiche, il meccanismo SmartFiltering rimuove tutti gli eventi non elaborati dell'oggetto business (come ad esempio Crea o Aggiorna) prima di inserire l'evento Elimina nella tabella eventi. SmartFiltering è un meccanismo all'interno dei trigger del database che riduce la quantità di elaborazione che deve essere eseguita dal broker di integrazione e dal connettore. In caso di eliminazioni logiche, il connettore inserisce un evento Elimina nella tabella eventi senza la rimozione di altri eventi per l'oggetto business.

**Impostazione dell'ID eventi dell'oggetto:** L'ID eventi è un ID univoco utilizzato per evitare la registrazione di un di eventi duplicati dal broker dell'applicazione. Ad esempio, un evento è in corso e viene inviato al broker di integrazione, quindi l'adattatore non ha esito positivo. Quando la scheda viene avviata, elabora di nuovo e invia gli eventi. Il broker di integrazione confronta gli ID eventi e ed elimina gli eventuali eventi duplicati, poiché ciascun ID evento è univoco.

**Richiamo degli oggetti business per l'elaborazione degli eventi:** Un'operazione di richiamo può essere effettuata in due modi su un oggetto business per l'elaborazione dell'evento. Il primo si basa sugli attributi chiave in un oggetto business. Il secondo si basa sugli attributi chiave e non chiave. Nel secondo caso, l'oggetto business deve supportare il verbo RetrieveByContent e deve utilizzare una coppia nome-valore per le chiavi dell'oggetto.

**Nota:** Se la chiave dell'oggetto non utilizza la coppia nome-valore, le chiavi nel campo della chiave dell'oggetto devono seguire lo stesso ordine delle chiavi nell'oggetto business.

## Notifica eventi

Il meccanismo di rilevamento dell'evento del connettore utilizza una tabella di eventi, una tabella di archivio, delle procedure memorizzate e i trigger del database. Poiché i punti di errore potenziali sono associati all'elaborazione degli eventi, il processo di gestione eventi non elimina un evento dalla tabella eventi fino a quando non è stato inserito nella tabella di archivio.

I trigger del database, che vengono aggiunti al database come parte della procedura di installazione dell'adattatore, popolano una tabella eventi quando si verifica un evento di interesse nel database. Il connettore esegue il polling di questa tabella ad intervalli regolari configurabili, richiama gli eventi ed elabora gli eventi prima per priorità e poi in sequenza. Quando il connettore ha elaborato un evento, viene aggiornato lo stato degli eventi.

**Nota:** E' necessario aggiungere i trigger al database come parte della procedura di installazione.

Il valore della proprietà ArchiveProcessed determina se il connettore archivia un evento nella tabella di archivio dopo averne aggiornato lo stato. Per ulteriori informazioni sulla proprietà ArchiveProcessed, consultare "Impostazione delle proprietà di configurazione del connettore" a pagina 70.

Tabella 1 a pagina 6 illustra il comportamento di archiviazione in base al valore della proprietà ArchiveProcessed.

Tabella 1. Comportamento di archiviazione in base alla proprietà `ArchiveProcessed`

Valore <code>ArchiveProcessed</code>	Motivo eliminato dalla tabella eventi	Comportamento del connettore
true o nessun valore	Elaborato correttamente	Archiviato con lo stato Inviato a InterChange
	Elaborato in modo non corretto	Archiviato con lo stato di Errore
false	Nessuna sottoscrizione per l'oggetto business	Archiviato con lo stato di Non sottoscritto
	Elaborato correttamente	Non archiviato, eliminato dalla tabella eventi
	Elaborato in modo non corretto	Resta nella tabella eventi con lo stato di Errore
	Nessuna sottoscrizione per l'oggetto business	Resta nella tabella eventi con lo stato di Non sottoscritto

SmartFiltering è un meccanismo all'interno dei trigger del database che riduce la quantità di elaborazione che deve essere eseguita dal broker di integrazione e dal connettore. Ad esempio, se un'applicazione ha aggiornato l'oggetto business Contratto 15 volte da quando il connettore ha effettuato l'ultima volta il polling degli eventi, il meccanismo SmartFiltering memorizza queste modifiche come un evento Aggiorna.

## Connessioni al database perse

Esistono vari motivi per la perdita di una connessione al database. Se ciò si verifica, il connettore termina. La specifica JDBC non fornisce un meccanismo per il rilevamento delle connessioni perse. Poiché il connettore supporta più di un database, non è presente una singola definizione del codice errore per una connessione persa ad un database.

La proprietà "PingQuery" a pagina 135 viene fornita per gestire questo rilevamento. Se si verifica un errore durante una richiesta di richiamo di un servizio, il connettore esegue PingQuery per confermare che l'errore non è causato da una connessione perduta ad un database. Se PingQuery non riesce e la proprietà "AutoCommit" a pagina 129 viene impostata su false, il connettore tenta di creare una nuova connessione al database. Se il connettore riesce a creare una nuova connessione al database, l'elaborazione continua, se la connessione non riesce, viene restituito APPRESPONSETIMEOUT, e il connettore viene terminato.

PingQuery viene eseguito se si verifica un errore durante l'accesso del connettore al database per qualunque tipo di transazione. Ad esempio:

- Durante l'accesso delle tabelle di eventi e di archivio
- Durante il richiamo dell'oggetto business relativo all'evento
- Durante la creazione o l'aggiornamento di un record relativo ad un oggetto business

## Dati dipendenti dalla locale

Il connettore è stato internazionalizzato, pertanto supporta le serie di caratteri a due byte e recapita il testo dei messaggi nella lingua specificata. Quando il connettore trasferisce dati da una posizione che utilizza una serie di codici carattere ad una posizione che utilizza una serie di codici differente, effettua la conversione carattere per conservare il significato dei dati.

Tale adattatore supporta l'elaborazione di dati di script bidirezionali (bi-di) per lingue come Arabo, Ebraico, Urdu, Farsi e Yiddish. Per utilizzare la proprietà bidirezionale, è necessario configurare le proprietà standard bidirezionali. Per ulteriori informazioni, fare riferimento alle proprietà di configurazione standard per i connettori nell'Appendice A.

Se RDBMS (relational database management system) utilizza un formato bidirezionale diverso dal formato standard di Windows, tutte le proprietà con supporto bidirezionale vengono trasformate dal formato standard di Windows al formato bidirezionale di RDBMS di destinazione.

L'ambiente di runtime Java<sup>(TM)</sup> all'interno di JVM (Java Virtual Machine) rappresenta i dati nella serie del codice di caratteri Unicode. Unicode contiene i formati di codifica per i caratteri nelle più comuni serie di codici carattere (sia single-byte che multibyte). Gran parte dei componenti nel sistema WebSphere Business Integration Server Express sono scritti in Java. Pertanto, quando i dati vengono trasferiti tra gran parte dei componenti del sistema WebSphere Business Integration Server Express, non è necessaria la conversione dei caratteri.

Per registrare errori e messaggi di informazione nel linguaggio appropriato e per il paese o territorio appropriati, configurare la proprietà di configurazione standard Locale per il proprio ambiente. Per ulteriori informazioni sulla proprietà consultare Appendice A, "Proprietà di configurazione standard per i connettori", a pagina 103.



---

## Capitolo 2. Installazione del connettore

Questo capitolo descrive l'installazione di WebSphere Business Integration Server Express Adapter per JDBC. Questo capitolo contiene le sezioni seguenti:

- "Ambiente dell'adattatore"
- "Prerequisiti"
- "Installazione dell'adattatore e dei file correlati" a pagina 10

---

### Ambiente dell'adattatore

Prima di installare, configurare e utilizzare l'adattatore, è necessario comprendere i requisiti dell'ambiente. Vengono riportati nelle seguenti sezioni.

- "Piattaforme dell'adattatore"
- "Dipendenze dell'adattatore"

### Piattaforme dell'adattatore

Oltre a un broker, questo adattatore richiede uno dei seguenti sistemi operativi ed è supportato sui database e sui software elencati.

#### Sistemi operativi:

- Microsoft Windows 2003
- IBM i5/OS V5R3 e OS/400 V5R2
- Linux:
  - RedHat 2.1 Enterprise Linux WS/AS/ES 3.0 Update 2, Intel (IA32)
  - SuSE Linux ES 8.1, Intel (IA32)
  - SuSE Linux ES 9.0, Intel (IA32)

#### Database:

Qualunque database per i driver JDBC disponibili.

#### Software non IBM:

Driver JDBC

### Dipendenze dell'adattatore

L'adattatore per JDBC richiede i file del driver JDBC.

---

### Prerequisiti

Prima di utilizzare il connettore, è necessario procedere nel modo seguente:

- Installare il driver JDBC che verrà utilizzato.
- Verificare che tutto il software specifico del vendor richiesto, compresi i requisiti del driver JDBC, sia stato installato.
- Verificare l'esistenza di un account utente nell'applicazione.

Il connettore elabora dati in qualunque applicazione creata in un qualunque database supportato da un driver che segue una specifica JDBC. Affinché il connettore elabori i dati nel database con cui comunica direttamente, è necessario che abbia l'accesso ad un account utente e password validi per

l'applicazione. L'account utente deve disporre di privilegi per richiamare, inserire, aggiornare ed eliminare i dati dal database dell'applicazione. Se non si dispone già di tale account, è necessario crearlo.

- Verificare la serie del codice del carattere del database connesso.

L'ambiente di runtime di Java all'interno di JVM (Java Virtual Machine) rappresenta i dati nella serie del codice del carattere Unicode. Unicode contiene codifiche per i caratteri delle più comuni serie di codici di carattere (a un byte o a più byte). Poiché il connettore viene scritto in Java, comprende Unicode.

---

## Installazione dell'adattatore e dei file correlati

Per informazioni sull'installazione dei prodotti degli adattatori WebSphere Business Integration Server Express, fare riferimento a WebSphere Business Integration Server Express - Guida all'installazione per Windows, per Linux o per i5/OS. La guida è disponibile presso l'infocenter di WebSphere Business Integration Server Express Adapters al seguente sito:  
<http://www.ibm.com/websphere/wbiserverexpress/infocenter>.

---

## Struttura dei file installati

Le sezioni secondarie di seguito riportate descrivono la struttura dei file installati dell'adattatore su un sistema Linux, Windows, e i5/OS.

---

## Struttura dei file installati su Linux

Per ulteriori informazioni sull'installazione del componente del connettore, fare riferimento al manuale *Guida all'installazione di WebSphere Business Integration Server Express per Linux*.

Tabella 2 descrive la struttura del file Linux utilizzata dal connettore.

Tabella 2. Struttura dei file Linux installato per il connettore

Sottodirectory di <i>ProductDir</i>	Descrizione
connettori/JDBC	Contiene i file JDBC.jar e start_JDBC.sh del connettore. Il file start_JDBC.sh è uno script di avvio del sistema per il connettore. Viene richiamato dallo script del gestore connettore generico. Quando si fa clic su Installa Connector Configurator Express, lo schermo di Connector Configuration di System Manager (InterChange Server Express come broker di integrazione) e Installer for WebSphere Business Integration Server Express Adapter crea come separatore personalizzato per questo script del gestore connessioni. Quando il connettore funziona con InterChange Server Express, utilizzare il separatore personalizzato per avviare e arrestare il connettore.
connectors/JDBC/dependencies	Contiene gli script SQL che creano gli eventi, gli archivi e le tabelle degli identificativi univoci.
connettori/messaggi	Contiene il file JDBCConnector.txt.
connectors/JDBC/Samples	Contiene i file di esempio per la creazione di procedure memorizzate diverse e oggetti business. Questi esempi vengono descritti nell'appendice C.
/lib	Contiene il file WBIA.jar.
/bin	Contiene il file CWConnEnv.sh.

---

## Struttura dei file installati per Windows

Per istruzioni sull'installazione dell'adattatore JDBC su un sistema operativo Windows, consultare la *Guida all'installazione di WebSphere Business Integration Server Express per Windows*. Tabella 3 descrive la struttura dei file di Windows utilizzata dal connettore.

Tabella 3. Struttura del file Windows installato per il connettore

Sottodirectory di ProductDir	Descrizione
connectors\JDBC	Contiene lo script di avvio del file bat CWJDBC.jar e start_JDBC.bat del connettore start_JDBC_service.bat.
connectors\JDBC\dependencies	Contiene gli script SQL che creano gli eventi, gli archivi e le tabelle degli identificativi univoci.
connectors\messages	Contiene il file JDBCCconnector.txt.
connectors\JDBC\Samples	Contiene i file di esempio per la creazione delle procedure memorizzate diversi e degli oggetti business. Questi esempi vengono descritti nell'appendice C.
\lib	Contiene il file the WBIA.jar.
\bin	Contiene il file CWConnEnv.bat.

---

Il programma di installazione di WebSphere Business Integration Server Express Adapter per JDBC aggiunge un'icona per il file del connettore al menu degli adattatori di WebSphere Business Integration. Per avviare più rapidamente il connettore, creare un collegamento a questo file sul desktop.

---

## Struttura dei file installati per i5/OS

Per istruzioni sull'installazione dell'adattatore JDBC su un sistema i5/OS, consultare la *Guida all'installazione di WebSphere Business Integration Server Express per i5/OS*. Tabella 4 descrive la struttura dei file i5/OS utilizzata dal connettore.

Tabella 4. Struttura file i5/OS installata per il connettore

Sottodirectory di ProductDir	Descrizione
connectors/JDBC	Contiene i file JDBC.jar e start_JDBC.sh del connettore.
connectors/JDBC/dependencies	Contiene gli script SQL che creano gli eventi, gli archivi e le tabelle degli identificativi univoci.
connectors/messages	Contiene il file JDBCCconnector.txt.
repository/JDBC	Contiene il file CN_JDBC.txt.
connectors/JDBC/Samples	Contiene i file README_Samples.txt e di esempio per la creazione di procedure memorizzate e oggetti business diversi.
/lib	Contiene il file WBIA.jar.
/bin	Contiene il file CWConnEnv.sh.

---

## Console di WebSphere Business Integration Server Express

Per i5/OS, una funzione denominata Console di WebSphere Business Integration Server Express viene eseguita su Windows. Lo strumento Console può essere utilizzato come un modo semplice per eseguire funzioni di gestione sull'adattatore e sul relativo ODA. Per informazioni sulla console, fare riferimento alla guida online fornita con la Console.





---

## Capitolo 3. Generazione delle definizioni dell'oggetto business utilizzando JDBC ODA

Questo capitolo descrive il modo in cui generare le definizioni di oggetti business per il connettore per JDBC con JDBC ODA, un object discovery agent (ODA). JDBC ODA utilizza tabelle di database, viste, procedure memorizzate e sinonimi/nomi brevi per scoprire le richieste degli oggetti specifiche alla relativa origine dati JDBC.

**Nota:** Una conoscenza dei concetti del database e dei driver JDBC (a scopi di configurazione) consente di comprendere il funzionamento di JDBC ODA.

Questo capitolo contiene le sezioni seguenti:

- "Installazione ed esecuzione di JDBC ODA"
- "Utilizzo di JDBC ODA in Business Object Designer Express" a pagina 17
- "Contenuti della definizione generata" a pagina 27
- "File di definizione dell'oggetto business di esempio" a pagina 30
- "Inserimento di attributi che contengono oggetti business child" a pagina 31
- "Aggiunta di informazioni alla definizione dell'oggetto business" a pagina 31

---

### Installazione ed esecuzione di JDBC ODA

Questa sezione discute dei seguenti argomenti:

- "Installazione di JDBC ODA"
- "Prima di utilizzare JDBC ODA" a pagina 14
- "Avvio di JDBC ODA" a pagina 15
- "Esecuzione di più istanze di JDBC ODA" a pagina 16
- "Come lavorare con i file di messaggi di errore e di traccia" a pagina 16

### Installazione di JDBC ODA

Per installare JDBC ODA, utilizzare il programma di installazione per WebSphere Business Integration Server Express Adapter for JDBC. Seguire le istruzioni contenute nella guida all'installazione di *WebSphere Business Integration Server Express Windows, per Linux, o for i5/OS*. Una volta completata l'installazione, i seguenti file sono installati nella directory sul sistema una volta installato il prodotto:

- ODA\JDBC\JDBCODA.jar
- ODA\messages\JDBCODAAgent.txt
- file ODA\message\JDBCODAAgent\_\_ll\_TT.txt specifico alla lingua (ll) e un paese o una regione (TT).
- ODA\JDBC\start\_JDBCODA.bat (solo Windows)
- ODA\JDBC/start\_JDBCODA.sh (solo Linux e i5/OS)
- bin\CWODAEEnv.bat (solo Windows)
- bin\CWODAEEnv.sh (solo Linux e i5/OS)

**Nota:** Eccetto se specificato diversamente, questo documento utilizza le barre inverse (\) come convenzione per i precorsi di directory. Per le installazioni

Linux e i5/OS, sostituire le barre (/) alle barre inverse. Tutti i nomi di percorso del prodotto sono relativi alla directory dove il prodotto è installato sul proprio sistema.

## Prima di utilizzare JDBC ODA

Prima di utilizzare JDBC ODA, è necessario:

1. Installare il driver JDBC appropriato. Seguire le istruzioni contenute nella guida all'installazione di *WebSphere Business Integration Server Express Windows o per Linux*.
2. Per i5/OS: JDBC ODA può essere connesso a qualunque database utilizzando un driver JDBC che supporta JDBC 2.0 o versione successiva. Sono presenti due driver JDBC che possono essere utilizzati su i5/OS:
  - Il driver di origine denominato JT400NATIVE\_JAR. Il nome del percorso è: /QIBM/ProdData/HTTP/Public /jt400/lib/jt400Native.jar.
  - Il Toolbox per il driver Java denominato JT400\_JAR. Il nome del percorso è: /QIBM/ProdData/HTTP/Public /jt400/lib/jt400Native.jar.
3. Poiché JDBC ODA genera i nomi dell'oggetto business e i nomi dell'attributo dai nomi delle tabelle e delle colonne di database corrispondente e poiché i nomi degli oggetti business e i nomi degli attributi devono essere in ISO Latin-1, verificare che i componenti del database appropriato dispongano di nomi Latin-1. Altrimenti, sono possibili le seguenti scelte:
  - Creare manualmente una definizione di oggetto business in Business Object Designer Express.
  - Modificare la definizione generata da JDBC ODA, in modo che tutti i nomi degli oggetti business e i nomi degli attributi siano in Latin-1.
4. Aprire per la modifica il file shell Linux (start\_JDBCODA.sh) o il file batch di Windows (start\_JDBCODA.bat) e configurare i valori descritti in Tabella 5.

Tabella 5. Variabili di configurazione dei file di shell e batch

Variabile	Spiegazione	Esempio
AGENTNAME	Nome di ODA	<b>Linux:</b> AGENTNAME=JDBCODA <b>Windows:</b> impostare AGENTNAME=JDBCODA
AGENT	Nome del file jar di ODA	<b>Linux:</b> AGENT=\$CROSSWORLDS/ODA/JDBC/JDBCODA.jar <b>Windows:</b> impostare AGENT= %CROSSWORLDS%\ODA\JDBC\JDBCODA.jar
DRIVERPATH	Percorso della libreria del driver JDBC: JDBC ODA utilizza le classi del driver per stabilire una connessione ad un database specificato	<b>Linux:</b> DRIVERPATH=\$CROSSWORLDS/lib/ \xwutil.jar:\$CROSSWORLDS/lib/ \xwbase.jar:\$CROSSWORLDS/lib/ \xsqlserver.jar:\$CROSSWORLDS/lib/ \spy/lib/spy.jar <b>Windows:</b> set DRIVERPATH=%CROSSWORLDS% / lib\xwutil.jar;%CROSSWORLDS%\lib\ /xwbase.jar;%CROSSWORLDS%\lib\ /xsqlserver.jar;%CROSSWORLDS%\lib\ /spy\lib\spy.jar

Tabella 5. Variabili di configurazione dei file di shell e batch (Continua)

Variabile	Spiegazione	Esempio
DRIVERLIB	Percorso delle librerie native utilizzate dal driver JDBC	<p><b>Linux:</b> Dopo aver copiato db2jdbc.so in \$CROSSWORLDS/lib*</p> <p>DRIVERLIB=\$CROSSWORLDS/lib/db2jdbc.so</p> <p><b>Windows:</b> Dopo aver copiato db2jdbc.dll in %CROSSWORLDS%\bin*</p> <p>DRIVERLIB=%CROSSWORLDS%\bin\db2jdbc.dll</p> <p>*In alternativa, è possibile indicare le cartelle in cui si trovano questi file.</p>

Aggiungere il driver del database a DRIVERPATH nello script di avvio. Ad esempio, se si utilizza DB2, aggiungere db2java.zip.

Una volta installato il driver JDBC e impostati i valori di configurazione nel file batch o shell, è necessario procedere nel modo seguente per generare gli oggetti business:

1. Avviare ODA.
2. Avviare Business Object Designer Express.
3. Seguire il processo costituito da sei passi in Business Object Designer Express per configurare ed eseguire ODA.

Le sezioni di seguito riportate descrivono dettagliatamente questi passi.

## Avvio di JDBC ODA

E' possibile avviare JDBC ODA con lo script di avvio appropriato al sistema operativo di cui si dispone.

### Avvio di ODA su Linux:

start\_JDBCODA.sh

### Avvio di su Windows:

start\_JDBCODA.bat

### Avvio di ODA su i5/OS:

Per avviare i5/OS, selezionare uno dei seguenti metodi:

1. Dal sistema Windows in cui è installata la console di WebSphere Business Integration Server Express, selezionare **Programmi >IBM WebSphere Business Integration Express >Toolset Express >Console di gestione**. Quindi, specificare il nome del sistema OS/400 o i5/OS o indirizzo IP e un profilo utente con relativa password con un'autorità speciale \*JOBCTL. Selezionare ODA dall'elenco di ODA, quindi selezionare il pulsante **Avvia ODA**.
2. Dall'immissione del comando i5/OS, eseguire il comando QSH CL e dall'ambiente QSHELL eseguire lo script /QIBM/ProdData/WBIServer44/bin/submit\_oda.sh con i seguenti parametri per  
pathToODAScript jobDescriptionName  
dove *pathToODAScript* è il percorso completo per lo script di avvio ODA e *jobDescriptionName* è il nome della descrizione del lavoro da utilizzare la libreria QWIBSVR44.

3. Dall'immissione del comando OS/400, eseguire il comando QSH CL e dall'immissione del comando QSHELL, eseguire direttamente lo script di avvio ODA:  
start\_ODAName.sh.

#### Arresto di ODA in i5/OS

Il modo in cui arrestare ODA per i5/OS dipende dal modo in cui è stato avviato. Se viene utilizzata l'opzione 1 o 2 in "Avvio di ODA in i5/OS", quindi:

1. Dal sistema Windows in cui è installata la console di WebSphere Business Integration Server Express, selezionare **IBM Websphere Business Integration Server Express >Toolset Express >Console di gestione**. Quindi, specificare il nome del sistema OS/400 o i5/OS oppure l'indirizzo IP e un profilo utente e la password che abbia autorizzazioni speciali \*JOBCTL. Selezionare JDBC ODA dall'elenco degli ODA, quindi fare clic sul pulsante Arresta ODA.
2. Dall'immissione del comando i5/OS, eseguire il comando CL: WRKACTJOB SBS (QWBISVR44). Sullo schermo vengono visualizzati tutti i lavori in esecuzione nel sottosistema.
3. Scorrere l'elenco fino a trovare il nome del lavoro che corrisponde alla descrizione del lavoro per ODA. Per ODAName JDBCODA, è QWBIJDBODA.
4. Selezionare l'opzione 4 per questo lavoro, quindi premere **F4** per richiedere il comando ENDJOB. Specificare IMMED per il parametro OPTION.
5. Premere **Invio**.

Se è stata utilizzata l'opzione 3 in "Avvio di ODA su i5/OS" per avviare ODA, premere **F3** dall'immissione del comando QSHELL in cui è stato eseguito lo script start\_ODAName.sh.

**Linux:**

start\_JDBCODA.sh

Configurare ed eseguire JDBC ODA utilizzando Business Object Designer Express. Business Object Designer Express individua ciascun ODA mediante il nome specificato nella variabile AGENTNAME di ciascuno script o file batch. Il nome ODA predefinito per questo connettore è JDBCODA.

## Esecuzione di più istanze di JDBC ODA

Si consiglia di modificare il nome di ODA quando si eseguono più istanze. Per creare ulteriori istanze unicamente denominate di JDBC ODA:

- Creare un file di batch o script a parte per ciascuna istanza.
- Specificare un nome univoco nella variabile AGENTNAME di ciascun file di batch o script.

Si consiglia di apporre un prefisso a ciascun nome costituito dal nome della macchina host quando si eseguono istanze ODA su macchine diverse.

Figura 2 a pagina 19 illustra la finestra Business Object Designer Express da cui selezionare l'ODA da eseguire.

## Come lavorare con i file di messaggi di errore e di traccia

I file di errori e messaggi di tracce (il valore predefinito è JDBCDAAgent.txt) si trovano in \ODA\messages\, ovvero nella directory del prodotto. Questi file usano le seguenti convenzioni di assegnazione dei nomi:

*AgentNameAgent.txt*

Se si creano più istanze dello script ODA o del file di batch e si fornisce un nome univoco per ciascun ODA rappresentato, è possibile ottenere un file di messaggi per ciascuna istanza ODA. In alternativa, si possono avere ODA con nomi differenti che usano lo stesso file di messaggio. Ci sono due modi per specificare un file di messaggio valido:

- Se si modifica il nome di un ODA senza creare un file di messaggi per questo, è necessario modificare il nome del file di messaggi in Business Object Designer Express come parte della configurazione ODA. Business Object Designer Express fornisce un nome per il file di messaggi, ma al momento non crea il file. Se il file visualizzato come parte della configurazione ODA non esiste, cambiare il valore in modo che punti ad un file esistente.
- E' possibile copiare il file di messaggio esistente per un ODA specifico e modificarlo nel modo richiesto. Business Object Designer Express presume che ciascun file sia denominato in base alla convenzione prestabilita. Ad esempio, se la variabile AGENTNAME specifica JDBCODA1, lo strumento prevede che il nome del file di messaggi associati è JDBCODA1Agent.txt. Pertanto, quando Business Object Designer Express fornisce il nome del file per la verifica come parte della configurazione ODA, tale nome file si basa sul nome ODA. Verificare che il file di messaggi predefinito abbia il nome corretto e, se necessario, correggerlo.

**Importante:** L'indicazione non corretta del nome del file di messaggio quando si configura ODA comporta la sua esecuzione senza messaggi. Per ulteriori informazioni sul nome file di messaggi, consultare "Configurazione delle proprietà di inizializzazione" a pagina 19.

Durante il processo di configurazione, specificare:

- Il nome del file in cui JDBC ODA scrive le informazioni di traccia e gli errori
- Il livello di traccia, che varia da 0 a 5.

Tabella 6 descrive questi valori.

Tabella 6. Livelli di traccia

Livello di traccia	Descrizione
0	Registra tutti gli errori
1	Esegue la traccia di tutti i messaggi in entrata e in uscita per il metodo
2	Esegue la traccia delle proprietà ODA e dei relativi valori
3	Esegue la traccia dei nomi di tutti gli oggetti business
4	Esegue la traccia dei dettagli di tutti i thread
5	<ul style="list-style-type: none"> <li>• Indica i valori di inizializzazione ODA per tutte le proprietà</li> <li>• Esegue la traccia di uno stato dettagliato di ciascun thread di JDBC ODA</li> <li>• Esegue la traccia del dump della definizione dell'oggetto business</li> </ul>

Per informazioni sulla configurazione dei valori, consultare "Configurazione delle proprietà di inizializzazione" a pagina 19.

---

## Utilizzo di JDBC ODA in Business Object Designer Express

Questa sezione descrive il modo in cui utilizzare JDBC ODA in Business Object Designer Express per generare le definizioni dell'oggetto business. Per informazioni sull'avvio di Business Object Designer Express, consultare il manuale *Business Object Development Guide*. E' possibile effettuare il download e visualizzare

la documentazione al seguente sito:  
<http://www.ibm.com/websphere/wbiserverexpress/infocenter>.

Una volta avviato un ODA, è necessario avviare Business Object Designer Express per configurarlo ed eseguirlo. Per generare una definizione di oggetto business utilizzando un ODA, è necessario effettuare sei passi in Business Object Designer Express. Business Object Designer Express dispone di una procedura guidata che consente di eseguire questi passi.

Una volta avviato ODA, procedere nel modo seguente per avviare la procedura guidata:

1. Aprire Business Object Designer Express.
2. Dal menu File, selezionare il menu secondario Nuovo utilizzando ODA.  
Business Object Designer Express visualizza la prima finestra della procedura guidata, denominata Seleziona agente. Figura 2 a pagina 19 illustra questa finestra.

Per selezionare, configurare ed eseguire ODA, procedere nel modo seguente:

1. "Selezione di ODA"
2. "Configurazione delle proprietà di inizializzazione" a pagina 19
3. "Espansione dei nodi e selezione degli oggetti del database" a pagina 21
4. "Conferma delle selezioni dell'oggetto del database" a pagina 22
5. "Generazione delle definizioni" a pagina 23 e facoltativamente "Come fornire informazioni aggiuntive" a pagina 23
6. "Salvataggio delle definizioni" a pagina 27

## **Selezione di ODA**

Figura 2 illustra la prima finestra di dialogo nella procedura guidata a sei passi di Business Object Designer Express. Da questa finestra, selezionare l'ODA da eseguire.

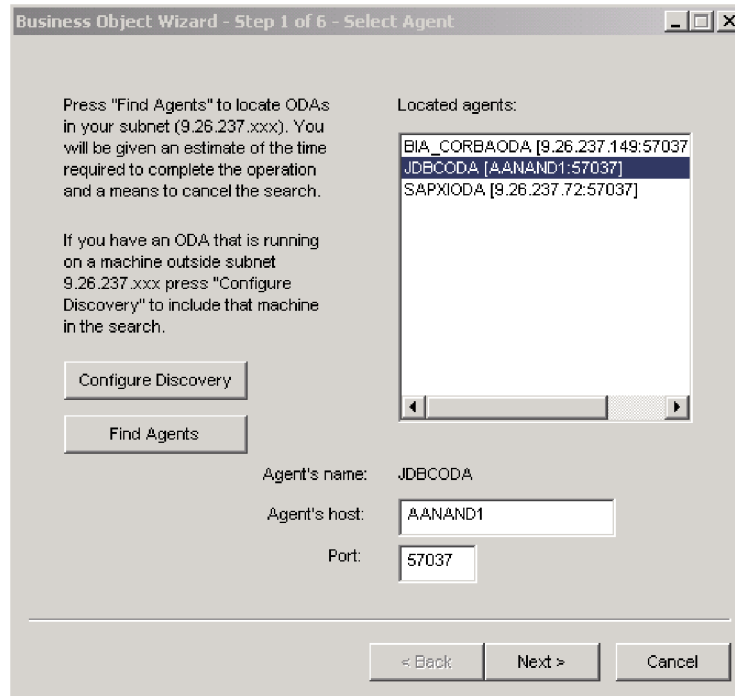


Figura 2. Selezione di un ODA

Per selezionare ODA:

1. Fare clic sul pulsante Trova agenti per visualizzare tutti gli ODA registrati o al momento in esecuzione nel campo Agenti trovati.

**Nota:** Se Business Object Designer Express non trova l'ODA desiderato, verificare la configurazione dell'ODA.

2. Selezionare l'ODA desiderato dall'elenco visualizzato.

Business Object Designer Express visualizza la selezione nel campo del nome Agente.

3. Fare clic su Avanti.

## Configurazione delle proprietà di inizializzazione

La prima volta che Business Object Designer Express comunica con JDBC ODA, viene richiesto di immettere una serie di proprietà di inizializzazione come illustrato in Figura 3. E' possibile salvare queste proprietà in un profilo denominato, in modo che non è necessario immetterle di nuovo ogni volta che viene utilizzato JDBC ODA. Per informazioni sulla specifica del profilo ODA, consultare il manuale *Business Object Development Guide*. E' possibile effettuare il download e visualizzare la documentazione al seguente sito: <http://www.ibm.com/websphere/wbiserverexpress/infocenter>.

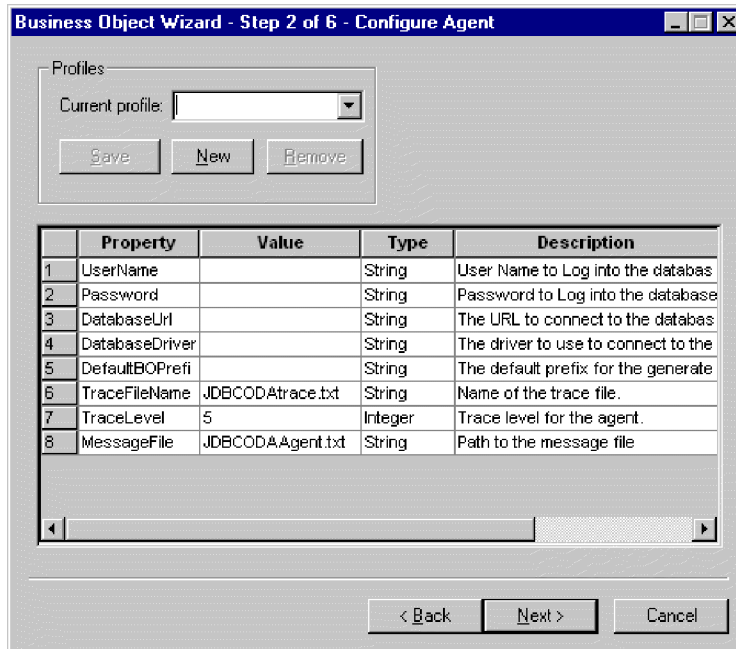


Figura 3. Configurazione delle proprietà di inizializzazione dell'agente

Configurare le proprietà JDBC ODA descritte in Tabella 7.

Tabella 7. Proprietà JDBC ODA

Numero riga	Nome proprietà	Tipo proprietà	Descrizione
1	UserName	Stringa	Nome dell'utente con l'autorizzazione per collegarsi al database
2	Password	Stringa	Password dell'utente con l'autorizzazione per collegarsi al database
5	DefaultBOPrefix	Stringa	Testo aggiunto al nome dell'oggetto business per renderlo univoco. E' possibile modificarlo in seguito, se necessario, quando Business Object Designer Express richiede le proprietà dell'oggetto business. Per ulteriori informazioni, consultare "Come fornire informazioni aggiuntive" a pagina 23.
6	TraceFileName	Stringa	File in cui JDBC ODA scrive le informazioni sulla traccia. Se il file non esiste, JDBC ODA lo crea nella directory \ODA\JDBC. Se il file già esiste, JDBC ODA lo appone ad esso. JDBC ODA denomina il file in base alla convenzione di denominazione. Ad esempio, se l'agente è denominato JDBCODA, genera un file di traccia denominato JDBCODATrace.txt. Utilizzare questa proprietà per specificare un nome diverso per il file.
7	TraceLevel	Intero	Livello di traccia abilitato per JDBC ODA. Consultare Tabella 6 a pagina 17.
8	MessageFile	Stringa	Nome dell'errore e del file di messaggi. JDBC ODA visualizza il nome del file in base alla convenzione di denominazione. Ad esempio, se l'agent è denominato JDBCODA, il valore del file property del messaggio viene visualizzato come JBCODAAgent.txt. <b>Importante:</b> il file di messaggio e di errore si trova nella directory \ODA\messages. Usare questa proprietà per verificare o specificare un file esistente.



### Importante

Correggere il nome del file di messaggi se il valore predefinito visualizzato in Business Object Designer Express rappresenta un file non esistente. Se il nome non è corretto quando ci si sposta all'esterno della finestra di dialogo, Business Object Designer Express visualizza un messaggio di errore nella finestra da cui è stato avviato ODA. Questo messaggio non compare in Business Object Designer Express. Se non si riesce a specificare un file di messaggi validi ODA viene eseguito senza messaggi.

## Espansione dei nodi e selezione degli oggetti del database

Una volta configurate le proprietà di inizializzazione per JDBC ODA, Business Object Designer Express si connette al database specificato e visualizza una struttura ad albero con tutti i nomi dello schema nel database. Questi nomi, che sono presentati come nodi nella struttura, sono espandibili. Fare clic su tali nodi per visualizzare tutte le tabelle, le viste, le procedure memorizzate e i sinonimi/nomi brevi in ciascuno schema. Figura 4 illustra questa finestra di dialogo con alcuni schemi estesi.

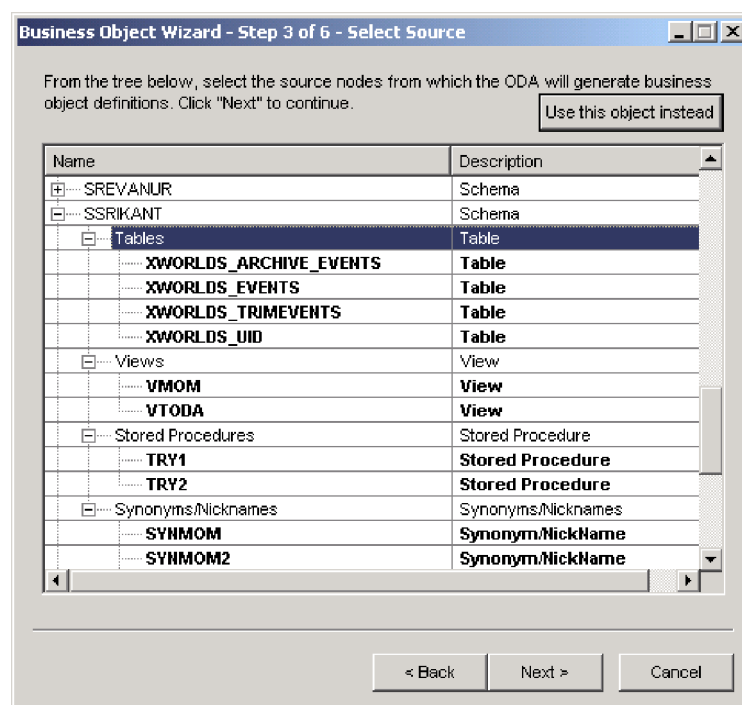


Figura 4. Struttura ad albero dello schema con nodi estesi

Per identificare tutti gli oggetti del database che memorizzano i dati per la definizione dell'oggetto business generato, selezionare tutte le tabelle, le viste, le procedure memorizzate e i sinonimi/nomi brevi richiesti, quindi fare clic su Avanti. Per informazioni sul modo in cui filtrare gli oggetti restituiti, consultare il manuale *Business Object Development Guide*. E' possibile effettuare il download di tale manuale dall'Infocenter di WebSphere Business Integration Server Express.

Il nome dello schema ALL SCHEMAS viene utilizzato per semplificare il richiamo di oggetti quali tabelle e viste del database che non dispongono di schemi associati agli oggetti. Quando si espande ALL SCHEMAS, viene visualizzata una struttura

ad albero con tabelle, viste, procedure memorizzate e sinonimi/nomi brevi. Quando ciascuno di questi nodi viene esteso, vengono visualizzati tutti gli oggetti rilevanti del database, indipendentemente della schema cui appartengono gli oggetti.

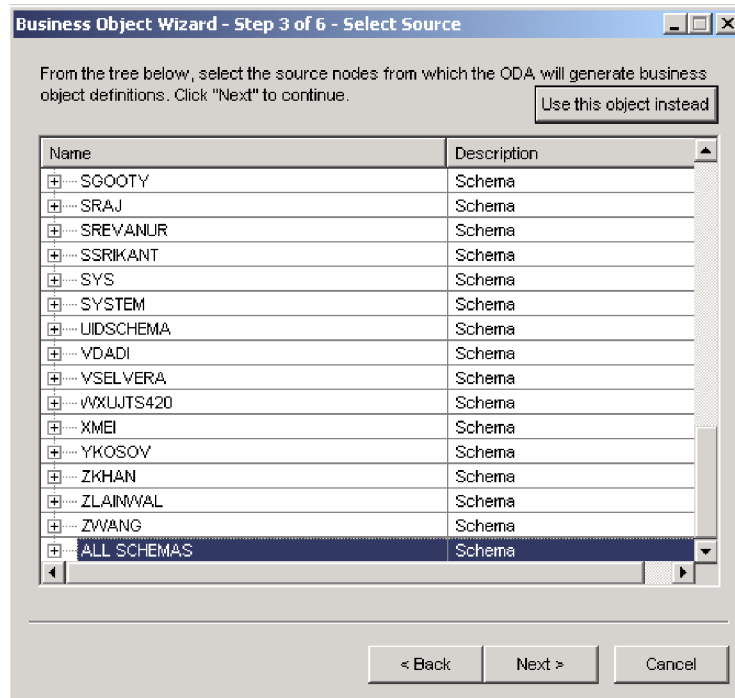


Figura 5. Estensione di ALL SCHEMAS per richiamare gli oggetti

## Conferma delle selezioni dell'oggetto del database

Una volta identificati tutti gli oggetti del database da associare alla definizione dell'oggetto business generato, Business Object Designer Express visualizza la finestra di dialogo solo con le tabelle, le viste, le procedure memorizzate e i sinonimi/nomi brevi selezionati. Figura 6 illustra questa casella di dialogo.

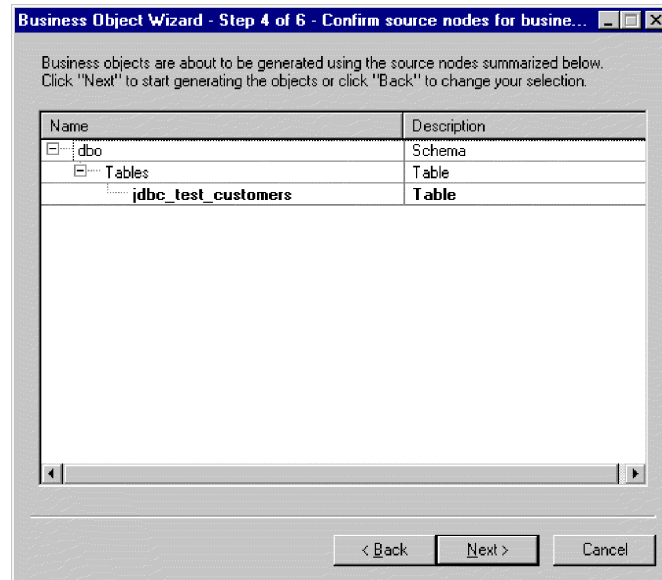


Figura 6. Conferma della sezione di oggetti del database

Questa finestra fornisce le seguenti opzioni:

- Per confermare la selezione, fare clic su Avanti.
- Se la selezione non è corretta, fare clic su Indietro per tornare alla finestra precedente ed effettuare le modifiche necessarie. Quando la selezione è corretta, fare clic su Avanti.

## Generazione delle definizioni

Una volta confermati gli oggetti del database, una finestra di dialogo informa che Business Object Designer Express sta generando le definizioni.

## Come fornire informazioni aggiuntive

Se sono necessarie ulteriori informazioni per JDBC ODA, Business Object Designer Express visualizza la finestra Proprietà BO, illustrata in Figura 7 a pagina 24, che richiede le informazioni.

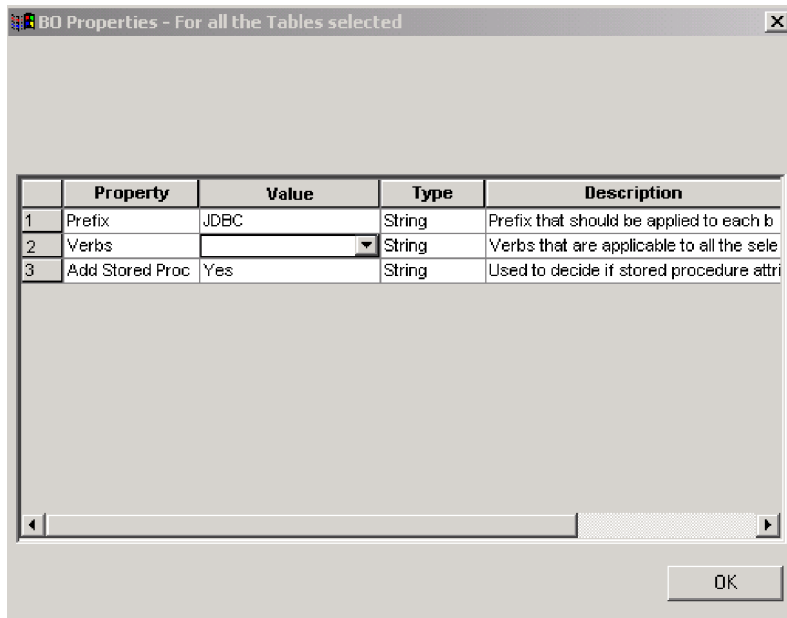


Figura 7. Ulteriori informazioni sugli oggetti del database

Nella finestra Proprietà BO, immettere o modificare le seguenti informazioni:

- *Prefixo*—Il testo che è aggiunto al nome dell’oggetto business per renderlo univoco. Se il valore viene immesso è corretto per la proprietà *DefaultBOPrefix* nella finestra Configura Agent (Figura 3), non è necessario modificare qui il valore.
  - *Verbi*— Fare clic nel campo *Valore* e selezionare uno o più verbi dal menu a comparsa. Si tratta di verbi supportati dall’oggetto business.
  - *Aggiungi procedura memorizzata*—Fare clic su Sì o No nel campo valore:
    - Se si seleziona Sì e si fa clic su OK, JDBC ODA visualizza una finestra con un elenco di tutti gli attributi relativi alla procedura memorizzata, come illustrato in Figura 8 a pagina 25. Selezionare gli attributi della procedura memorizzata da aggiungere all’oggetto business.
    - Selezionare No per verificare che non è stato aggiunta alcun attributo della procedura memorizzata alla definizione dell’oggetto business generato.
- Il valore predefinito è Sì.

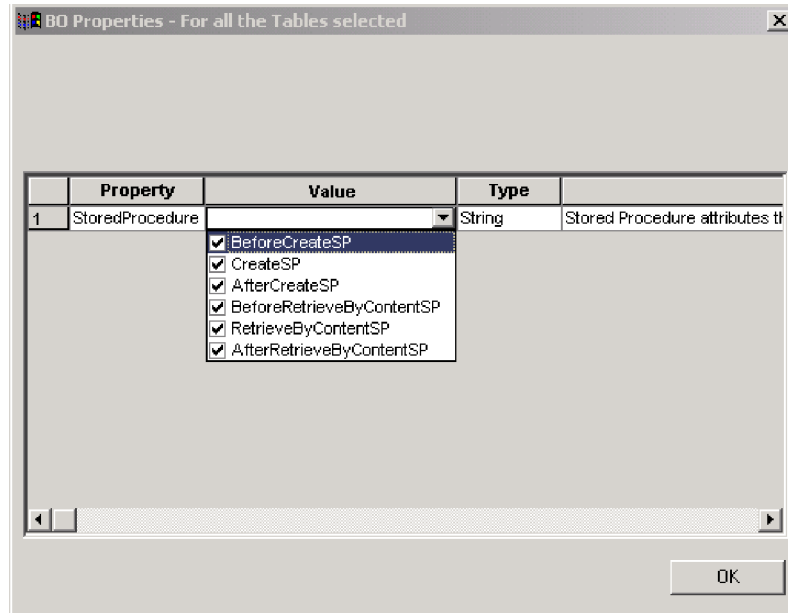


Figura 8. Selezione degli attributi della procedura memorizzata

**Nota:** Se un campo nella finestra di dialogo Proprietà BO ha più valori, quando si apre per la prima volta sembra vuota. Fare clic nel campo per visualizzare un elenco a discesa dei valori.

Gli attributi della procedura memorizzata da aggiungere all'oggetto business possono essere associati ad una delle procedure memorizzate nel database in quello schema. Come illustrato in Figura 9 a pagina 26, è possibile scegliere una procedura memorizzata da un elenco a discesa di tutte le procedure memorizzate nel database in quello schema, per ciascun attributo della procedura memorizzata. Queste informazioni generano le informazioni specifiche dell'applicazione necessarie per l'attributo.

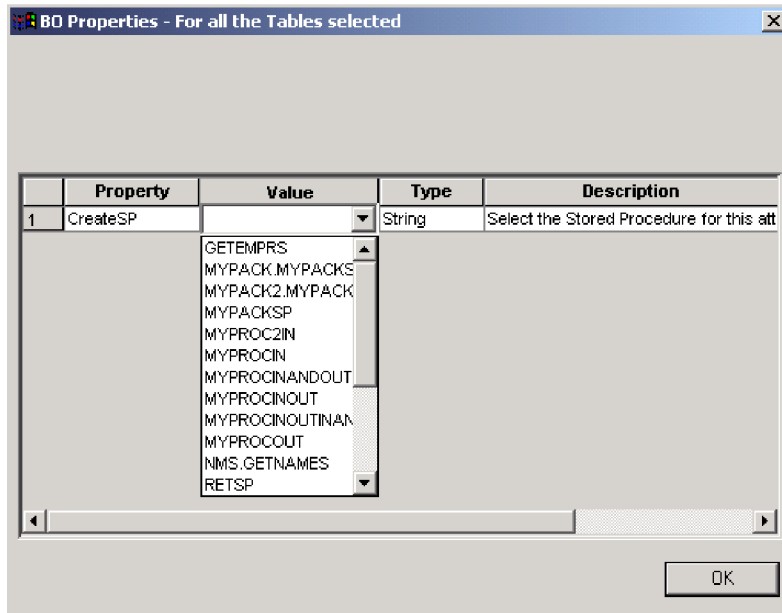


Figura 9. Associazione delle procedure memorizzate ai relativi attributi

Le informazioni specifiche dell'applicazione (ASI) per il livello di oggetto sarà  
 TN=tableName

E per il livello di attributo, l'ASI sarà CN=ColumnName

Se un oggetto business viene generato da una procedura memorizzata e se sono associati gli attributi della procedura memorizzata del connettore JDBC (come ad esempio SPForCreate), ODA fornisce un elenco di tutti i nomi delle procedure memorizzate nello schema con gli attributi delle procedure memorizzate. ODA consente di associare la procedura memorizzata richiesta all'oggetto business. Ciò genera l'ASI per l'attributo della procedura memorizzata del connettore JDBC nel modo seguente:

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

Dove IN significa che il parametro della procedura memorizzata è di tipo INPUT, OUT significa che il parametro è di tipo OUTPUT e IO significa che è di tipo INPUT/OUTPUT. ODA non imporrà RS su true o false in ASI, quindi è necessario impostarlo manualmente.

I verbi aggiunti all'oggetto business sono verbi standard, in genere Richiama, RetrieveByContent, Crea, Aggiorna e Elimina.

Se il parametro di ritorno della procedura memorizzata è ResultSet Type, ODA analizza la serie di risultati e crea un oggetto business, rendendo le colonne degli attributi della serie di risultati dell'oggetto business. L'ASI per le colonne della procedura memorizzata viene impostato come CN=StoredProcedureColumnName. ODA imposta gli attributi chiave in base alle informazioni sui metadati JDBC restituiti dal driver. Se non viene restituito nulla, ODA non contrassegna gli attributi per impostazione predefinita come chiavi. Tutti gli altri attributi, come ad esempio la lunghezza e il tipo, vengono impostati come per gli attributi generati dalle tabelle.

## Salvataggio delle definizioni

Dopo aver fornito le informazioni richieste nella finestra di dialogo Proprietà BO e aver fatto clic su OK, Business Object Designer Express visualizza la finestra di dialogo finale nella procedura guidata. Qui, è possibile salvare la definizione nel server o in un file oppure è possibile aprire la definizione per la modifica in Business Object Designer Express. Per ulteriori informazioni e per ulteriori modifiche, consultare il manuale *Business Object Development Guide*. E' possibile effettuare il download di tale manuale dall'Infocenter di WebSphere Business Integration Server Express.

---

## Contenuti della definizione generata

La definizione dell'oggetto business che JDBC ODA genera contiene:

- Un attributo per ciascuna colonna nelle tabelle di database, nelle viste, nelle procedure memorizzate e nei sinonimi/nomi brevi specificati.
- I verbi specificati nella finestra Proprietà BO
- Informazioni specifiche dell'applicazione:
  - A livello di oggetto business
  - Per ciascun attributo
  - Per ciascun verbo

Tale sezione descrive:

- "Proprietà a livello di oggetto business"
- "Proprietà degli attributi" a pagina 28
- "Verbi" a pagina 30

## Proprietà a livello di oggetto business

JDBC ODA genera le seguenti informazioni a livello di oggetto business:

- Nome dell'oggetto di business
- Valori predefiniti della versione su 1.0.0
- Informazioni specifiche sull'applicazione

Le informazioni specifiche dell'applicazione a livello di oggetto business consentono di:

- Specificare il nome della tabella di database corrispondente
- Fornire le informazioni necessarie per eseguire un'eliminazione logica o fisica

A livello di oggetto business, il formato di informazioni specifiche dell'applicazione è costituito da parametri separati da delimitatori punto e virgola (;). Il nome del parametro ed i relativi valori sono separati da un delimitatore due punti (:). La sintassi è:

```
TN=TableName; SCN=StatusColumnName:StatusValue
```

dove *TableName* identifica la tabella di database, *StatusColumnName* è il nome della colonna di database utilizzata per eseguire le eliminazioni logiche e *StatusValue* è il valore che significa che un oggetto business non è attivo o eliminato.

*AppSpecificInfo* generato da JDBC ODA a questo livello contiene un valore solo per il nome della tabella di database, della vista, della procedura memorizzata o

del sintomo/nome breve. Per informazioni sulla specifica di un valore per la colonna di stato, consultare "Informazioni specifiche dell'applicazione a livello di oggetto business" a pagina 58.

## Proprietà degli attributi

Questa sezione descrive le proprietà generate da JDBC ODA per ciascun attributo. Per ulteriori informazioni sugli attributi, consultare la sezione "Proprietà degli attributi di un oggetto business" a pagina 55.

### Proprietà del nome

JDBC ODA ottiene il valore del nome dell'attributo dal nome della colonna nella tabella del database, nella vista, nella procedura memorizzata o sintomo/nome breve.

### Proprietà tipo di dati

Quando si imposta il tipo di un attributo, JDBC ODA converte il tipo di dati di una colonna di una tabella o vista in un oggetto business corrispondente. Questa conversione viene effettuata in due fasi. In primo luogo, il tipo di dati nel database viene convertito in un tipo JDBC. In seguito, il tipo JDBC viene convertito in un tipo di oggetto business. La prima conversione viene effettuata dal driver JSBC che si sta utilizzando. Fare riferimento alla specifica (2.0 e versione successiva) per dettagli sulla mappatura del tipo di database singolo ad un tipo JDBC. Tabella 8 illustra la conversione dal tipo JDBC al tipo di oggetto business corrispondente.

Tabella 8. Corrispondenza dei tipi di dati

Tipo JDBC	Tipo di oggetto business di WebSphere Business Integration Server Express 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
BINARY	STRING, aggiungere BYTEARRAY=TRUE  a AppSpecificInfo



Tabella 8. Corrispondenza dei tipi di dati (Continua)

Tipo JDBC	Tipo di oggetto business di WebSphere Business Integration Server Express Adapter
VARBINARY	STRING,  aggiungere BYTEARRAY=TRUE  a AppSpecificInfo

**Nota:** Se un tipo di dati della colonna non è del tipo illustrato in Tabella 8 a pagina 28, l'ODA JDBC ignora la colonna e visualizza un messaggio indicante che non è possibile elaborare la colonna.

### Proprietà di cardinalità

JDBC ODA imposta la cardinalità di tutti gli attributi semplici su 1.

### Proprietà MaxLength

JDBC ODA ottiene la lunghezza di una stringa dalla lunghezza specificata per i tipi di dati varchar, char o text.

### Proprietà IsKey

Se la colonna è una chiave principale nella tabella, JDBC ODA la contrassegna come attributo chiave. Tuttavia, se viene selezionata una vista, una procedura memorizzata o un sinonimo/nome breve, invece di una tabella come nodo di origine per generare gli oggetti business, JDBC ODA non contrassegna la colonna come attributo chiave. In questo caso, l'attributo chiave deve essere impostato manualmente.

### Proprietà IsForeignKey

JDBC ODA non imposta la proprietà IsForeignKey. E' possibile impostarla in Business Object Designer Express.

### Proprietà IsRequired

Se un campo è designato come non null nella tabella, nella vista, nella procedura memorizzata o nel sinonimo/nome breve, JDBC ODA lo contrassegna come un attributo richiesto. Tuttavia, JDBC ODA non contrassegna il campo chiave come richiesto, poiché potrebbe essere presente una sequenza associata o potrebbe trattarsi di una colonna di identità.

### Proprietà AppSpecificInfo

JDBC ODA include due parametri per la proprietà AppSpecificInfo a livello di attributo. La sintassi dei parametri specificati è:

- `CN=ColumnName`

dove ColumnName è il nome della colonna nella tabella di database, vista, procedura memorizzata o sinonimo/nome breve associato all'attributo specifico.

- `BYTEARRAY=true|false`

JDBC ODA riconosce le colonne con i dati binari e crea un attributo del tipo Stringa con una proprietà AppSpecificInfo BYTEARRAY=true.

**Nota:** E' possibile impostare parametri AppSpecificInfo aggiuntivi in Business Object Designer Express. Per informazioni su questi parametri, consultare "Informazioni specifiche dell'applicazione a livello di attributo" a pagina 60.

## Verbi

JDBC ODA genera i verbi specificati nella finestra Proprietà BO. Crea una proprietà AppSpecificInfo per ciascun verbo ma non la popola. Per ulteriori informazioni, consultare "Informazioni specifiche dell'applicazione formato per i verbi" a pagina 67.

---

## File di definizione dell'oggetto business di esempio

Di seguito è riportata una definizione di oggetto business di esempio:

```
[BusinessObjectDefinition]
Nome = CUSTOMER
Versione = 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]

    *****Other attributes *****

[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]
```

---

## **Inserimento di attributi che contengono oggetti business child**

Utilizzare Business Object Designer Express per inserire gli attributi che rappresentano oggetti business child con cardinalità multipla o singola. Per ulteriori informazioni, consultare il manuale *Business Object Development Guide*. E' possibile effettuare il download di tale manuale dall'Infocenter di WebSphere Business Integration Server Express.

---

## **Aggiunta di informazioni alla definizione dell'oggetto business**

Poiché le viste, tabelle di database, le procedure memorizzate e i sinonimi/nomi brevi potrebbero non disporre di tutte le informazioni richieste dalla definizione di un oggetto business, potrebbe essere necessario aggiungere informazioni alla definizione dell'oggetto business creato da JDBC ODA. Per ulteriori informazioni, consultare Capitolo 4, "Comprensione degli oggetti aziendali", a pagina 33.

Per esaminare la definizione dell'oggetto business o aggiungere informazioni, è possibile utilizzare Business Object Designer Express o un editor di testo. Per ricaricare una definizione esaminata nell'archivio di WebSphere Business Integration Server Express Adapter, è possibile utilizzare Business Object Designer Express con InterChange Server Express come broker di integrazione, utilizzando il comando `repos_copy`.



---

## Capitolo 4. Comprensione degli oggetti aziendali

Questo capitolo descrive il connettore per gli oggetti aziendali dei processi JDBC e descrive le ipotesi effettuate dal connettore durante il richiamo e la modifica dei dati. Questo capitolo contiene le sezioni seguenti:

- “Convenzioni di denominazione degli oggetti business e degli attributi”
- “Struttura oggetto business”
- “Elaborazione dell’espressione oggetto business” a pagina 38
- “Proprietà degli attributi di un oggetto business” a pagina 55
- “Informazioni specifiche dell’applicazione dell’oggetto business” a pagina 57

E’ possibile utilizzare queste informazioni come guida per la modifica di oggetti aziendali esistenti o per l’implementazione di oggetti aziendali nuovi. Per informazioni sull’utilità che automatizza la creazione dei file di definizione degli oggetti aziendali dalle tabelle del database, consultare Capitolo 3, “Generazione delle definizioni dell’oggetto business utilizzando JDBC ODA”, a pagina 13. Gli oggetti business devono essere generati prima di poter eseguire il connettore, sebbene sia possibile utilizzare gli esempi dell’oggetto business in Appendice C, “Esempi di oggetti business”, a pagina 143 se si preferisce provare a configurare ed eseguire il connettore prima della creazione degli oggetti aziendali.

Il connettore suppone la struttura degli oggetti business supportati, le relazioni tra gli oggetti business parent e child, il formato delle informazioni specifiche dell’applicazione e la rappresentazione del database dell’oggetto business. Pertanto, quando si crea o si modifica un oggetto business che verrà elaborato dal connettore, le modifiche devono essere conformi alle regole progettate per il connettore. Se non lo sono, il connettore non è in grado di elaborare correttamente oggetti aziendali nuovi o modificati.

---

### Convenzioni di denominazione degli oggetti business e degli attributi

Il nome dell’oggetto business utilizzato dal connettore può essere costituito solo da caratteri alfanumerici e dal carattere trattino basso. Anche i nomi degli attributi dell’oggetto business possono essere costituiti solo da caratteri alfanumerici e dal carattere trattino basso.

---

### Struttura oggetto business

In gran parte dei casi, il connettore suppone che ogni oggetto business è rappresentato da una tabella o vista di database e che ciascun **attributo semplice** (ovvero, un attributo che rappresenta un valore univoco, come Stringa o Intero o Date) all’interno dell’oggetto è rappresentato da una colonna della tabella o nella vista. Pertanto, gli attributi all’interno dello stesso oggetto business singolo non possono essere memorizzati in tabelle di database diverse. Tuttavia, sono possibili le seguenti situazioni:

- La tabella del database potrebbe disporre di più colonne rispetto agli attributi semplici di un oggetto business singolo (ovvero, alcune colonne nel database non sono rappresentate nell’oggetto business). Dovrebbero essere incluse solo le colonne necessarie per l’elaborazione dell’oggetto business nel progetto.
- L’oggetto business singolo potrebbe disporre di più attributi semplici rispetto alle colonne della tabella del database (ovvero, alcuni attributi nell’oggetto

business non sono rappresentati nel database). Gli attributi senza una rappresentazione nel database non dispongono nelle informazioni specifiche dell'applicazione vengono impostati con un valore predefinito o specificano le procedure memorizzate.

- L'oggetto business singolo può rappresentare una visualizzazione che separa più tabelle di database. Il connettore può utilizzare tale oggetto business quando viene eseguito il trigger degli eventi Crea, Richiama, Aggiorna ed Elimina. Tuttavia, quando si elaborano le richieste dell'oggetto business, il connettore può utilizzare tale oggetto business solo per le richieste Richiama.
- L'oggetto business singolo può rappresentare un oggetto wrapper che viene utilizzato come contenitore per gli oggetti business non correlati. L'oggetto wrapper non è rappresentato da una vista o tabella di database. Gli oggetti wrapper non possono essere utilizzati come child di altri oggetti.

**Nota:** Se un oggetto business si basa su una procedura memorizzata (SP) ciascun attributo semplice (diverso dagli attributi SP speciali) potrebbe o meno disporre informazioni specifiche dell'applicazione. Per ulteriori informazioni, consultare "Procedure memorizzate" a pagina 47.

Gli oggetti business possono essere flat o gerarchici. Tutti gli attributi di un oggetto business flat sono semplici e rappresentano un valore. Il termine **gerarchico** fa riferimento ad un oggetto business completo, compresi tutti gli oggetti aziendali business child contenuti a qualunque livello. Il termine oggetto business **singolo** fa riferimento ad un oggetto business indipendentemente dagli oggetti business child che potrebbe contenere o in cui è contenuto. Il termine oggetto business **di livello superiore** fa riferimento ad un oggetto business singolo di livello superiore nella gerarchia che non dispone di un oggetto business parent.

Un oggetto business gerarchico dispone di attributi che rappresentano un oggetto business child, una serie di oggetti business o una combinazione di entrambi. Ciascun oggetto business può contenere un oggetto business child o una serie di oggetti business e così via. Una **relazione con cardinalità singola** si verifica quando un attributo in un oggetto business parent rappresenta un attributo business child. In questo caso, l'attributo è dello stesso tipo dell'oggetto business child.

Una **relazione con cardinalità multipla** si verifica quando l'attributo nell'oggetto business parent rappresenta una serie di oggetti business child. In questo caso, l'attributo è una serie dello stesso tipo degli oggetti business child.

Il connettore supporta le relazioni di seguito riportate tra gli oggetti:

- "Relazioni con cardinalità singola" a pagina 35
- "Relazioni con cardinalità singola e dati senza proprietà" a pagina 35
- "Relazioni con cardinalità multipla" a pagina 36
- "Relazioni con cardinalità singola che memorizzano la relazione nel child" a pagina 37
- "Oggetti wrapper" a pagina 38

In ciascun tipo di cardinalità, la relazione tra gli oggetti business child e parent viene descritta dalle informazioni specifiche dell'applicazione dell'attributo chiave dell'oggetto business memorizzando la relazione. Per ulteriori dettagli sulle informazioni specifiche dell'applicazione, consultare "FK=[fk\_object\_name.]fk\_attribute\_name" a pagina 61.

## Relazioni con cardinalità singola

In genere, un oggetto business che contiene un oggetto business child con cardinalità singola dispone di almeno due attributi che rappresentano la relazione. Il tipo di un attributo è uguale al tipo di child. L'altro attributo è di tipo semplice che contiene la chiave principale del child come chiave esterna nel parent. Il parent dispone di tanti attributi della chiave esterna quanti sono gli attributi della chiave principale.

Poiché le chiavi esterne che stabiliscono la relazione vengono memorizzate nel parent, ciascun parent può contenere solo un child con cardinalità singola di un determinato tipo.

Figura 10 illustra una relazione con cardinalità singola. Nell'esempio, fk1 nella casella ParentBOName è presente l'attributo semplice che contiene la chiave principale del child e child[1], anche nella casella ParentBOName è presente l'attributo che rappresenta l'oggetto business child.

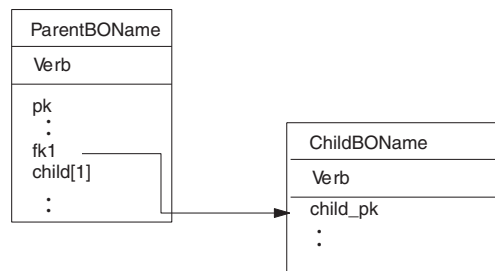


Figura 10. Relazione con cardinalità singola tipica

## Relazioni con cardinalità singola e dati senza proprietà

In genere, ciascun oggetto business parent **dispone** di dati all'interno dell'oggetto business child che lo contiene. Ad esempio, se ciascun oggetto business Cliente contiene un oggetto business Indirizzo, quando viene creato un nuovo cliente, viene inserita una nuova riga sia nella tabella Indirizzo sia nella tabella Cliente. Il nuovo indirizzo è univoco per il nuovo cliente. In modo simile, quando si elimina un cliente dalla tabella Clienti, l'indirizzo del cliente viene eliminato dalla tabella degli indirizzi.

Tuttavia, si verificano situazioni in cui più oggetti business gerarchici contengono gli stessi dati, che non appartengono a nessuno di essi. Ad esempio, si presume che un oggetto business Indirizzo disponga di un attributo StateProvince[1] che rappresenta la tabella di ricerca StateProvince con cardinalità singola. Poiché la tabella di ricerca viene aggiornata raramente e viene conservata indipendentemente dai dati di indirizzo, la creazione o la modifica dei dati di indirizzo non condizionano i dati della tabella di ricerca. Il connettore trova un nome di provincia o di stato esistente oppure non riesce. Non aggiunge o modifica i valori nella tabella di ricerca.

Quando più oggetti business contengono lo stesso oggetto business con cardinalità singola, l'attributo della chiave esterna in ciascun oggetto business parent deve specificare la relazione come NO\_OWNERSHIP. Quando il broker di integrazione invia il connettore come oggetto business gerarchico con una richiesta Crea, Elimina o Aggiorna, il connettore ignora il child con cardinalità singola contenuto senza proprietà. Il connettore esegue solo le operazioni di richiamo su

questi oggetti business. Se il connettore non riesce a richiamare tale oggetto business con cardinalità singola, restituisce un errore e termina l'elaborazione.

Per informazioni sul modo in cui specificare la relazione senza proprietà, consultare "Attributi che rappresentano un oggetto business child con cardinalità singola" a pagina 66. Per ulteriori informazioni sulla specifica delle relazioni relative alla chiave esterna, consultare "Specifica di una chiave esterna di un attributo" a pagina 62.

### **Dati denormalizzati e dati senza proprietà**

Oltre a facilitare l'utilizzo di tabelle di ricerca statiche, il contenimento senza proprietà dispone di un'altra funzione: la sincronizzazione di dati normalizzati e non normalizzati.

**Sincronizzazione da dati normalizzati a dati denormalizzati:** La specifica di una relazione come NO\_OWNERSHIP consente la creazione o la modifica dei dati quando si effettua la sincronizzazione da un'applicazione normalizzata ad una denormalizzata. Ad esempio, si supponga che l'applicazione di un'origine normalizzata memorizzi i dati nella due tabelle, A e B. Si supponga che l'applicazione di destinazione denormalizzata memorizzi tutti i dati in una tabella, in modo che ciascuna entità A memorizzi in modo ridondante i dati B.

In questo esempio, per sincronizzare una modifica dei dati contenuti nella tabella B dall'applicazione di origine all'applicazione di destinazione, è necessario eseguire il trigger di un evento della tabella A ogni volta che vengono modificati i dati della tabella B. Inoltre, poiché i dati della tabella B vengono memorizzati in modo ridondante nella tabella A, è necessario inviare un oggetto business per ciascuna riga nella tabella A che contiene i dati modificati della tabella B.

**Sincronizzazione da dati denormalizzati a dati normalizzati:** Se si effettua la sincronizzazione dati da un'applicazione di origine denormalizzata ad un'applicazione di destinazione normalizzata, il connettore non crea, elimina o aggiorna i dati contenuti senza proprietà nell'applicazione normalizzata.

Quando si sincronizzano i dati con un'applicazione normalizzata, il connettore ignora tutti i child con cardinalità singola senza proprietà. Per creare, rimuovere o modificare tali dati child, è necessario elaborare manualmente i dati.

## **Relazioni con cardinalità multipla**

In genere, un oggetto business che contiene una serie di oggetti business child dispone di un solo attributo che rappresenta la relazione. Il tipo di attributo è una serie dello stesso tipo degli oggetti business child. Affinché un parent contenga più di un child, le chiavi esterne che stabiliscono la relazione vengono memorizzate in ciascun child.

Pertanto, ciascun child dispone di almeno un attributo semplice che contiene la chiave principale esterna del parent come chiave esterna. Il child dispone di tanti attributi della chiave esterna quanti sono gli attributi della chiave principale.

Poiché le chiavi esterne che stabiliscono la relazione vengono memorizzate nel child, ciascun parent può contenere zero o più child.

Figura 11 illustra una relazione con cardinalità multipla. Nell'esempio parentId nelle tre caselle ChildBOName è presente l'attributo semplice che contiene la



chiave principale del parent e child[n] nella casella ParentBOName box è l'attributo che rappresenta la serie di oggetti business child.

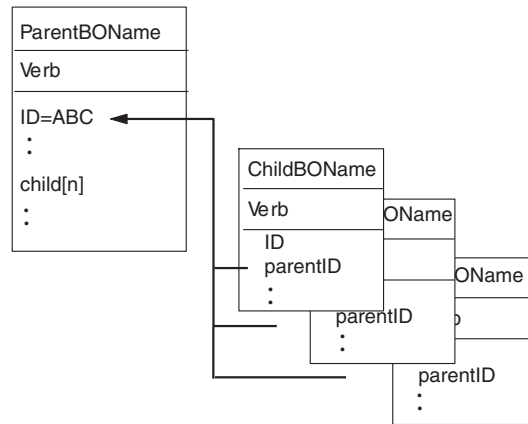


Figura 11. Relazione di oggetti business con più cardinalità

## Relazioni con cardinalità singola che memorizzano la relazione nel child

Alcune applicazioni memorizzano un'entità child in modo che la relazione parent-child sia memorizzata nel child piuttosto che nel parent. In altre parole, il child contiene una chiave esterna il cui valore è identico al valore memorizzato nella chiave principale del parent.

Figura 12 illustra questo tipo particolare di relazione con cardinalità singola.

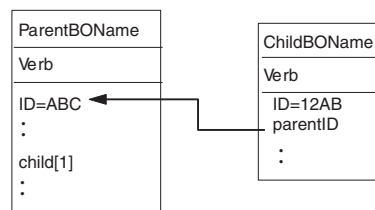


Figura 12. Oggetto business con cardinalità singola con la relazione memorizzata nel child

Le applicazioni utilizzando questo tipo di relazione con cardinalità singola quando i dati non esistono indipendentemente dal parent ed è possibile accedervi solo mediante il relativo parent. Tali dati child non appartengono mai a più di un parent e richiedono che il parent e il relativo valore della chiave principale esista prima della creazione del child e del valore della chiave esterna.

Per la conformità di tali applicazioni, il connettore supporta anche gli oggetti business gerarchici contenenti un child con cardinalità singola, ma memorizzano la relazione nel child piuttosto che nel parent.

Per specificare che un oggetto parent contiene un child con cardinalità singola in questo modo particolare, quando si specificano le informazioni specifiche all'applicazione dell'attributo che contiene il child, non includere il parametro CONTAINMENT. Per ulteriori informazioni, consultare "Attributi che rappresentano un oggetto business child con cardinalità singola" a pagina 66.

## Oggetti wrapper

Un **oggetto wrapper** è un oggetto di livello superiore che non corrisponde ad alcuna vista o tabella di database. L'oggetto wrapper è denotato dalla proprietà dell'oggetto business di livello superiore di WRAPPER con un valore true. L'oggetto wrapper è un parent fittizio utilizzato come contenitore per i child non correlati. Elaborando l'oggetto wrapper, il connettore ignora l'oggetto business di livello superiore ed elabora solo il child. L'oggetto wrapper può contenere n e/o n-1 entità di cardinalità.

Un'entità di cardinalità deve disporre di almeno un attributo univoco contrassegnato come chiave principale ed almeno un attributo contrassegnato come chiave esterna. Questa chiave esterna viene aggiunta come chiave principale nell'oggetto wrapper. La chiave esterna dell'entità fa riferimento alla chiave principale dell'oggetto wrapper appena aggiunta.

Nel caso di un'entità di cardinalità n-1, la chiave principale deve essere contrassegnata come chiave principale e come chiave esterna, facendo riferimento alla chiave principale nel wrapper, che è lo stesso della chiave principale nell'entità n-1.

---

## Elaborazione dell'espressione oggetto business

Questa sezione descrive gli aspetti di seguito riportati dell'elaborazione dei verbi dell'oggetto business:

- "Determinazione del verbo", che illustra il modo in cui il connettore determina il verbo da utilizzare per ciascun oggetto business di origine
- "Delta e dopo immagini", che definisce i termini e illustra il modo in cui funziona il connettore con il dopo immagini
- "Elaborazione dei verbi" a pagina 40, che illustra i passi effettuati dal connettore durante la creazione, il richiamo, l'aggiornamento o l'eliminazione di un oggetto business
- "Istruzioni SQL" a pagina 46, che illustra il modo in cui il connettore utilizza le istruzioni SQL semplici per la selezione, l'aggiornamento, il richiamo o l'eliminazione degli oggetti business.
- "Procedure memorizzate" a pagina 47, che illustra il modo in cui il connettore utilizza le procedure memorizzate
- "Commit e rollback delle transazioni" a pagina 55, che illustra brevemente il modo in cui il connettore utilizza i blocchi delle transazioni

## Determinazione del verbo

Un oggetto business di livello superiore e ciascuno dei singoli oggetti business child può contenere i relativi verbi. Pertanto, il broker di integrazione può inoltrare un oggetto business con diversi verbi parent e oggetti business child al connettore. Quando ciò si verifica, il connettore utilizza il verbo di un oggetto business parent di livello superiore per determinare il modo in cui elaborare tutto l'oggetto business. Per ulteriori informazioni, consultare "Elaborazione dei verbi" a pagina 40.

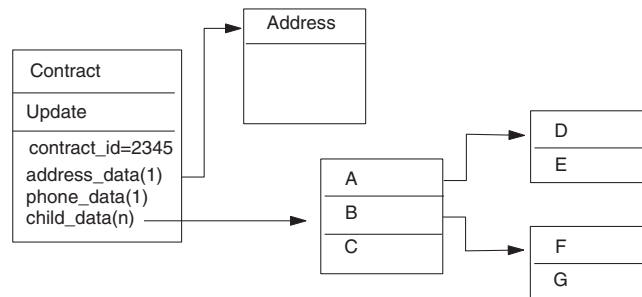
## Delta e dopo immagini

Un **dopo immagine** è lo stato di un oggetto business una volta effettuate tutte le modifiche. Un **delta** è un oggetto business utilizzato in un'operazione di aggiornamento che contiene solo i valori chiave e i dati da modificare. Poiché il connettore supporta solo dopo immagini, quando riceve un oggetto business per

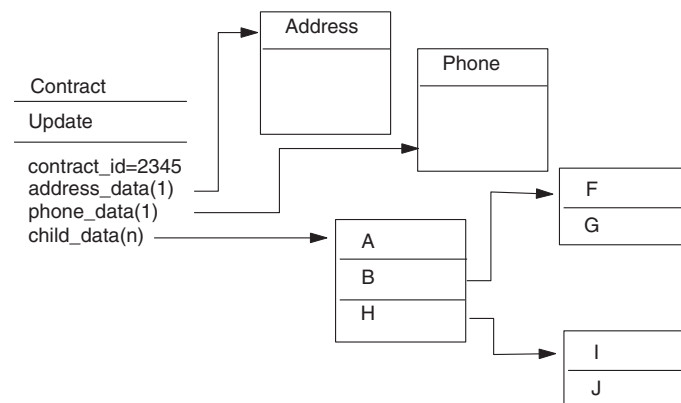
l'aggiornamento, il connettore presume che tale oggetto business rappresenti lo stato desiderato dei dati dopo l'aggiornamento.

Pertanto, quando il broker di integrazione invia un oggetto business con il verbo Aggiorna al connettore, quest'ultimo modifica la rappresentazione corrente dell'oggetto business nel database in modo che corrisponda esattamente all'oggetto business di origine. Per procedere nel modo seguente, il connettore modifica i valori dell'attributo semplice e aggiunge o rimuovi gli oggetti business child.

Ad esempio, si supponga che lo stato corrente del Contratto 2345 nel database viene visualizzato nel seguente diagramma che dispone di oggetti business Contratto e Indirizzo di livello superiore, attributi contract\_id, address\_data, phone\_data e child\_data e oggetti business child dalla A alla G.:



Inoltre, si supponga che il broker di integrazione inoltri l'oggetto business al connettore, che causa le modifiche descritte in seguito:



Per elaborare l'operazione di aggiornamento, il connettore applica le seguenti modifiche al database:

- Aggiorna gli attributi semplici negli oggetti business Contratto e Indirizzo di livello superiore
- Crea l'oggetto business Telefono
- Aggiorna gli attributi semplici negli oggetti business child A, B, F e G
- Elimina gli oggetti business child C, D e E
- Crea gli oggetti business child H, I e J

Poiché il connettore suppone che ciascun oggetto business ricevuto dal broker di integrazione rappresenti un dopo immagine, è importante assicurarsi che ciascun oggetto business inviato a tale connettore per l'aggiornamento contenga oggetti

business child esistenti validi. Anche se non è stato modificato alcuno degli attributi semplici di un oggetto business child, l'oggetto business child deve essere incluso nell'oggetto business di origine.

Tuttavia, esiste un modo per impedire ad alcuni connettori di eliminare gli oggetti business child mancanti durante un'operazione di aggiornamento. E' possibile utilizzare le informazioni specifiche dell'applicazione per l'attributo che rappresentano il child o la serie di child per istruire il connettore affinché conservi gli oggetti business child non inclusi nell'oggetto business di origine. Questa operazione viene effettuata impostando `KEEP_RELATIONSHIP` su `true`. Per ulteriori informazioni, consultare "Specifica di una chiave estesa di un attributo" a pagina 62.

## Elaborazione dei verbi

Questa sezione illustra i passi effettuati dal connettore durante la creazione, il richiamo, l'aggiornamento o l'eliminazione di un oggetto business che riceve dal broker di integrazione. Il connettore elabora oggetti business gerarchici in modo ricorrente, ovvero esegue gli stessi passi per ciascun oggetto business child fino a quando non sono stati elaborati tutti i singoli oggetti business.

**Nota:** Un oggetto business wrapper di livello superiore supporta i verbi crea, richiama, aggiorna ed elimina. Un oggetto wrapper non viene elaborato; vengono elaborati solo gli oggetti contenuti.

### Confronto oggetto business

In vari punti dell'elaborazione del verbo dell'oggetto business illustrati di seguito, il connettore confronta i due oggetti business per visualizzare se sono uguali. Ad esempio, durante un'operazione di aggiornamento, il connettore determina se un determinato oggetto business esiste in una serie di oggetti business. Per eseguire la verifica, il connettore confronta l'oggetto business a ciascun oggetto business all'interno della serie. Affinché due oggetti business siano identici, devono essere soddisfatte le due condizioni di seguito riportate:

- Il tipo di oggetti business confrontati devono essere uguali. Ad esempio, un oggetto business Cliente non è mai considerato identico ad un oggetto business Contatto anche se tutti i relativi attributi sono esattamente uguali.
- Tutti gli attributi chiave corrispondenti nei due oggetti business devono contenere valori identici. Se un attributo chiave viene impostato su `CxIgnore` in entrambi gli oggetti business, il connettore li considera identici. Tuttavia, se un attributo chiave viene impostato su `CxIgnore` in un oggetto business ma non in un altro, gli oggetti business non sono identici.

### Creazione delle operazioni

Quando si crea un oggetto business, il connettore restituisce uno stato `VALCHANGE`, se l'operazione è stata completata correttamente (se l'operazione ha causato delle modifiche o meno all'oggetto business) o `FAIL`, se l'operazione non è stata completata correttamente.

Di seguito sono riportati i passi effettuati dal connettore durante la creazione di un oggetto business gerarchico:

1. Inserisce in modo ricorrente ciascun oggetto business child con cardinalità singola contenuto con l'appartenenza nel database. In altre parole, il connettore crea il child e tutti gli oggetti business child contenuti dal child e i relativi child.

Se la definizione dell'oggetto business specifica che un attributo rappresenta un oggetto business child con cardinalità singola e che l'attributo è vuoto, il

connettore ignora l'attributo. Tuttavia, se la definizione dell'oggetto business richiede che l'attributo rappresenti un child e ciò non si verifica, il connettore restituisce un errore ed arresta l'elaborazione.

2. Elabora ciascun oggetto business child con cardinalità singola con appartenenza contenuto senza appartenenza come segue:
  - a. Tenta in modo ricorrente di richiamare il child dal database utilizzando i valori della chiave inoltrati dal broker di integrazione.
  - b. Se l'operazione di richiamo non viene completata correttamente, indicando che il child non esiste nel database, il connettore restituisce un errore ed arresta l'elaborazione. Se l'operazione di richiamo viene eseguita correttamente, il connettore aggiorna in modo ricorrente l'oggetto business child.

**Nota:** Affinché questo approccio funzioni correttamente quando l'oggetto business child esiste nel database dell'applicazione, gli attributi della chiave principale negli oggetti business child devono disporre di riferimenti incrociati corretti con le operazioni di creazione. Se un oggetto business child non esiste nel database dell'applicazione, gli attributi della chiave principale devono essere impostati su CxBlank.

3. Inserisce l'oggetto business di livello superiore nel database nel modo seguente:
  - a. Imposta ciascun valore della chiave esterna sui valori della chiave principale dell'oggetto business corrispondente rappresentato con cardinalità singola. Poiché i valori degli oggetti business child possono essere impostati dalle sequenze di database, dai conteggi o dal database stesso durante la creazione del child, questo passo assicura che i valori della chiave esterna nel parent sono corretti prima che il connettore inserisca il parent nel database.
  - b. Genera nuovo un valore di ID univoco per ciascun attributo che viene impostato automaticamente dal database. Il nome del contatore o della sequenza del database viene memorizzato nelle informazioni specifiche dell'applicazione dell'attributo. Se un attributo dispone di un contatore o di una sequenza del database associato, il valore generato dal connettore sovrascrive qualunque valore inoltrato dal broker di integrazione. Per ulteriori informazioni sulla specifica di una sequenza del database o contatore, consultare UID=AUTO in "Informazioni specifiche dell'applicazione per gli attributi semplici" a pagina 60.
  - c. Il valore di un attributo viene copiato nel valore di un altro attributo, come specificato dal parametro CA (CopyAttribute) delle informazioni specifiche dell'applicazione dell'attributo. Per ulteriori informazioni sull'utilizzo del parametro CA, consultare CA=set\_attr\_name in "Informazioni specifiche dell'applicazione per gli attributi semplici" a pagina 60.
  - d. Inserisce l'oggetto business di livello superiore nel database.

**Nota:** Se un oggetto business di livello superiore è un wrapper, non viene inserito nel database.

4. Elabora ciascuna delle cardinalità singole con appartenenza che memorizza le relazioni parent-child nel child come segue:
  - a. Imposta i valori della chiave esterna nel child per fare riferimento al valore negli attributi della chiave principale corrispondente nel parent. Poiché i valori della chiave principale del parent possono essere stati generati durante la creazione del parent, ciò assicura che i valori della chiave esterna in ciascun child siano corretti prima che il connettore inserisca il child nel database.

- b. Inserisce il child nel database.
- 5. Elabora ciascuno degli oggetti business child con cardinalità multipla come segue:
  - a. Imposta i valori della chiave esterna in ciascun child per fare riferimento al valore negli attributi della chiave principale corrispondente nel parent. Poiché i valori della chiave principale del parent possono essere stati generati durante la creazione del parent, ciò assicura che i valori della chiave esterna in ciascun child siano corretti prima che il connettore inserisca il child nel database.
  - b. Inserisce ciascuno degli oggetti business child con cardinalità multipla nel database.

## Operazioni di richiamo

Di seguito sono riportati i passi effettuati dal connettore durante il richiamo di un oggetto business gerarchico:

1. Rimuove tutti gli oggetti business child dall'oggetto business di livello superiore ricevuti dal broker di integrazione.
2. Richiama l'oggetto business di livello superiore dal database.
  - Se il richiamo restituisce una riga, il connettore continua l'elaborazione.
  - Se un richiamo non restituisce alcuna riga, significa che l'oggetto business di livello superiore non esiste nel database e il connettore restituisce `BO_DOES_NOT_EXIST`.
  - Se il richiamo restituisce più di una riga, il connettore restituisce `FAIL`.

### Note:

- a. Un oggetto business può disporre di attributi che non corrispondono ad alcuna colonna di database, come ad esempio gli attributi segnalibro. Durante il richiamo, il connettore non modifica tali attributi nell'oggetto business di livello superiore, restano impostati sui valori ricevuti dal broker di integrazione. Negli oggetti business child, il connettore imposta tali attributi ai valori predefiniti durante il richiamo.
- b. Un oggetto business wrapper di livello superiore deve contenere qualunque valore di attributo dagli oggetti ad un livello immediatamente inferiore rispetto all'oggetto wrapper necessario per il richiamo degli oggetti, compresi gli attributi dei segnalibri e delle chiavi. L'oggetto wrapper deve disporre di tutti gli attributi dei segnalibri e delle chiavi popolati. Gli attributi semplici nell'oggetto wrapper utilizzati come chiavi esterne negli oggetti di un livello inferiore al wrapper devono essere contrassegnati come chiavi nell'oggetto wrapper.
3. Richiama in modo ricorrente tutti gli oggetti business child con cardinalità multipla.

**Nota:** Il connettore non impone l'univocità quando viene popolata una serie di oggetti business. E' responsabilità del database assicurare l'univocità. Se il database restituisce oggetti business child duplicati, il connettore restituisce i child duplicati.

4. Richiama in modo ricorrente ciascun child con cardinalità singola indipendentemente dall'oggetto business child contenuto con o senza appartenenza.

**Nota:** Tutte le cardinalità singole con appartenenza vengono elaborate in base alla ricorrenza nell'oggetto business e prima che sia elaborato l'oggetto

business parent. L'appartenenza o la non appartenenza dell'oggetto child non determinano la sequenza di elaborazione, ma determinano il tipo di elaborazione.

## Operazioni RetrieveByContent

Un verbo RetrieveByContent è applicabile solo all'oggetto business di livello superiore, poiché il connettore esegue un richiamo in base agli attributi solo nell'oggetto business di livello superiore.

Se un oggetto business di livello superiore utilizza il verbo RetrieveByContent, tutti gli attributi (compresi gli attributi non chiave) che non sono null, vengono utilizzati come criteri di richiamo.

Se viene restituita più di una riga, il connettore utilizza la prima riga come riga del risultato e restituisce un messaggio MULTIPLE\_HITS.

**Nota:** Un verbo RetrieveByContent non è applicabile per un oggetto business wrapper di livello superiore.

## Operazioni di aggiornamento

Quando si aggiorna un oggetto business, il connettore restituisce uno stato VALCHANGE, se l'operazione è stata completata correttamente (se l'operazione ha causato delle modifiche o meno all'oggetto business) o FAIL, se l'operazione non è stata completata correttamente.

Di seguito sono riportati i passi effettuati dal connettore durante l'aggiornamento di un oggetto business gerarchico:

1. Utilizza i valori della chiave principale dell'oggetto business per richiamare l'entità corrispondente dal database. L'oggetto business richiamato è una rappresentazione precisa dello stato corrente dei dati nel database.
  - Se il richiamo non ha esito positivo, significa che l'oggetto business di livello superiore non esiste nel database, il connettore restituisce BO\_DOES\_NOT\_EXIST e l'aggiornamento non riesce.

**Nota:** Un oggetto business wrapper di livello superiore non deve esistere nel database. Tuttavia, deve contenere qualunque valore di attributo dagli oggetti ad un livello immediatamente inferiore rispetto all'oggetto wrapper necessario per il richiamo degli oggetti, compresi gli attributi dei segnalibri e delle chiavi. L'oggetto wrapper deve disporre di tutti gli attributi dei segnalibri e delle chiavi popolati. Gli attributi semplici nell'oggetto wrapper utilizzati come chiavi esterne negli oggetti di un livello inferiore al wrapper devono essere contrassegnati come chiavi nell'oggetto wrapper.

- Se il richiamo viene eseguito correttamente, il connettore confronta l'oggetto business richiamato all'oggetto business di origine per determinare quali oggetti business child richiedono modifiche nel database. Tuttavia, il connettore non confronta i valori negli attributi semplici dell'oggetto business di origine a quelli nell'oggetto business richiamato. Il connettore aggiorna i valori degli attributi semplici non chiave.

Se tutti gli attributi semplici in un oggetto business di livello superiore rappresentano chiavi, il connettore non è in grado di generare una query di aggiornamento per l'oggetto business di livello superiore. In questo caso, il connettore registra un'avvertenza e continua con il passo 2.

2. Aggiorna in modo ricorrente tutti i child con cardinalità singola dell'oggetto business di livello superiore.

Se la definizione dell'oggetto business richiede che un attributo rappresenti un oggetto business child, il child deve esistere sia nell'oggetto business di origine sia nell'oggetto business richiamato. Altrimenti, l'operazione di aggiornamento non riesce e il connettore restituisce un errore.

Il connettore gestisce child con cardinalità singola contenuti con l'appartenenza in uno dei seguenti modi:

- Se il child è presente sia negli oggetti business di origine sia in quelli richiamati, invece di aggiornare il child esistente nel database, il connettore elimina il child esistente e crea il nuovo child.
- Se il child è presente nell'oggetto business di origine ma non in quello richiamato, il connettore lo crea in modo ricorrente nel database.
- Se il child è presente nell'oggetto business richiamato ma non in quello di origine, il connettore lo elimina in modo ricorrente dal database. Il tipo di operazione di eliminazione, fisica o logica, dipende dal valore della relativa proprietà `ChildUpdatePhyDelete`.

Per i child con cardinalità singola contenuti senza appartenenza, il connettore tenta di richiamare ogni child dal database presente nell'oggetto business di origine. Se il child viene richiamato correttamente, il connettore popola l'oggetto business child ma non lo aggiorna, poiché i child con cardinalità singola contenuti senza appartenenza non vengono mai modificati dal connettore.

3. Per una cardinalità singola con la proprietà che memorizza la relazione nel parent, viene impostato ciascun valore della chiave esterna nel parent al valore della chiave principale nell'oggetto business child con cardinalità singola. Questo passo è necessario poiché i child con cardinalità singola possono essere stati aggiunti al database durante i passi precedenti e ciò causa la generazione di nuovi ID univoci.
4. Aggiorna tutti gli attributi semplici dell'oggetto business eccetto quali i cui attributi corrispondenti nell'oggetto business di origine contengono il valore `CxIgnore`.

Poiché l'oggetto business che si sta aggiornando deve essere univoco, il connettore verifica che sia elaborata una sola riga come risultato. Se vengono restituite più righe, si verifica un errore.

5. Imposta tutti i valori delle chiavi esterne in ciascun child che memorizza la relazione parent-child nel child (cardinalità multipla e singola) con il valore della chiave principale dell'oggetto business parent corrispondente. (Con `InterChange Server Express` come broker di integrazione, questi valori sono in genere incrociati con riferimenti durante la mappatura.) Questo passo è importante per assicurare che i valori della chiave esterna dei nuovi child che memorizzano la relazione nel child siano corretti prima che il connettore li aggiorni.
6. Elabora ciascuna child con cardinalità multipla dell'oggetto business richiamato in uno dei seguenti modi:
  - Se il child esiste sia nella serie dell'oggetto business di origine sia in quello richiamato, il connettore lo aggiorna in modo ricorrente nel database.
  - Se il child esiste nella serie di origine ma non nella serie dell'oggetto business richiamato, il connettore lo crea in modo ricorrente nel database.
  - Se il child esiste nella serie di oggetti business richiamati ma non nella serie di origine, il connettore lo elimina in modo ricorrente dal database, se le informazioni specifiche dell'applicazione per l'attributo che rappresente il child nel parent non ha il valore `KEEP_RELATIONSHIP` impostato su `true`. In questo caso il connettore non elimina il child dal database. Per ulteriori informazioni, consultare "Specifica di una chiave estesa di un attributo" a pagina 62



pagina 62. Il tipo di operazione di eliminazione, fisica o logica, dipende dal valore della relativa `ChildUpdatePhyDelete` proprietà.

**Nota:** Il broker di integrazione deve verificare che gli oggetti business contenuti con cardinalità multipla nell'oggetto business di origine siano univoci (ovvero, che una serie non contiene due o più copie dello stesso oggetto business). Se il connettore riceve duplicati di un oggetto business in una serie di origine, elabora l'oggetto business due volte, con risultati imprevedibili.

## Operazioni DeltaUpdate

L'elaborazione del verbo `DeltaUpdate` è diversa dall'elaborazione del verbo `Aggiorna`:

- In un'operazione `DeltaUpdate`, non si verifica alcuna operazione di richiamo prima dell'aggiornamento, dal momento che viene effettuata in un'elaborazione del verbo `Aggiorna`.
- Non sono effettuati confronti tra l'oggetto business in entrata e quello nel database.
- Tutti i child vengono elaborati in base al verbo impostato in ciascun oggetto child. Se un child non dispone di un verbo impostato, il connettore restituisce un errore.

Quando si aggiorna un oggetto business `DeltaUpdate`, il connettore restituisce uno stato `VALCHANGE`, se l'operazione è stata completata correttamente (se l'operazione ha causato delle modifiche o meno all'oggetto business) o `FAIL`, se l'operazione non è stata completata correttamente.

Di seguito sono riportati i passi effettuati dal connettore durante l'aggiornamento di un oggetto business con `DeltaUpdate`:

1. Elaboro in modo ricorrente tutti i singoli child con cardinalità singola dell'oggetto parent. SE un child è contrassegnato come `IsRequired` nella specifica dell'oggetto business, deve essere presente nell'oggetto in entrata. Altrimenti, l'operazione `DeltaUpdate` non riesce e il connettore restituisce un errore.
2. Imposto tutti i valori delle chiavi esterne nel parent che si riferiscono agli attributi nei child con cardinalità singola sui valori child corrispondenti. Questo passo è necessario poiché i child con cardinalità singola possono essere stati aggiunti al database durante i passi precedenti e ciò causa la generazione di nuovi valori di sequenza.
3. Aggiorno l'oggetto corrente elaborato utilizzando un'istruzione `UPDATE` o una procedura memorizzata. Tutti gli attributi semplici dell'oggetto business singolo sono aggiornati, escluso quelli impostati su `Ignore` nell'oggetto business in entrata. Il connettore non confronta l'oggetto in entrata all'oggetto corrente con un livello di attributo per determinare gli attributi necessari da aggiungere per aggiornare l'istruzione, vengono aggiornati tutti. Poiché l'oggetto che si sta aggiornando deve essere univoco, il connettore verifica che sia elaborata una sola riga come risultato. Se viene elaborata più di una riga, si verifica un errore.
4. Imposto tutti i valori della chiave esterna in tutti i child N con cardinalità dell'oggetto corrente che correla gli attributi parent ai valori parent corrispondenti. In genere, questi valori sono già correlati durante la mappatura dei dati; tuttavia non è il caso dei nuovi child nei contenitori N con cardinalità. Questo passo assicura che i valori della chiave esterna in tutti i child N con cardinalità siano corretti prima dell'aggiornamento di tali child.
5. Aggiorno tutti i contenitori N con cardinalità dell'oggetto corrente.

Quando gli oggetti child vengono elaborati, viene ottenuto ciascun verbo del child, quindi viene effettuata l'operazione appropriata. I verbi consentiti in un child nelle operazioni DeltaUpdate sono Crea, Elimina e DeltaUpdate.

- Se viene trovato un verbo Crea nel child, quest'ultimo viene creato nel database se è un child di appartenenza. I child non di appartenenza vengono richiamati per convalidarne l'esistenza nel database.
- Se viene trovato un verbo Elimina nel child, quest'ultimo viene eliminato.
- Se viene trovato un verbo DeltaUpdate nel child, quest'ultimo viene aggiornato nel database.

### **Operazioni di eliminazione**

Quando elimina un oggetto business, il connettore restituisce uno stato SUCCESS se l'operazione è stata completata correttamente o FAIL se l'operazione non è riuscita. L'oggetto business parent viene richiamato, quindi l'adattatore elimina in modo ricorrente tutti i child con cardinalità singola che dispongono della relazione di appartenenza nel parent, quindi l'oggetto business parent stesso ed infine tutti i child N con cardinalità. I child senza appartenenza con cardinalità singola non vengono mai eliminati. Se l'oggetto business non esiste, il connettore restituisce FAIL.

Il connettore supporta le operazioni di eliminazione logica e fisica, in base al valore SCN (Status Column Name) nelle informazioni specifiche dell'applicazione dell'oggetto. Se il valore SCN è definito, il connettore esegue un'operazione di eliminazione logica. Se il valore SCN non è definito, il connettore esegue un'operazione di eliminazione fisica.

**Operazioni di eliminazione fisica:** Di seguito sono riportati i passi effettuati dal connettore durante l'eliminazione di un oggetto business gerarchico:

1. Elimina in modo ricorrente tutti gli oggetti business con cardinalità singola con appartenenza.
2. Elimina l'oggetto business di livello superiore.
3. Elimina in modo ricorrente tutti gli oggetti business child con cardinalità multipla.

**Nota:** Un oggetto business wrapper di livello superiore non dispone di una tabella di database corrispondente, quindi non viene eliminato dal database. Qualunque valore per l'attributo semplice per un wrapper viene ignorato.

**Operazioni di eliminazione logica:** Di seguito sono riportati i passi eseguiti dal connettore durante l'eliminazione logica di un oggetto business:

1. Inoltra un'operazione Aggiorna che imposta l'attributo di stato dell'oggetto business specificato dalle informazioni specifiche dell'applicazione dell'oggetto business. Il connettore assicura che sia aggiornata solo una riga nel database, altrimenti restituisce un errore.
2. Elimina logicamente in modo ricorrente tutti i child con cardinalità singola contenuti con tutti i child con cardinalità multipla di appartenenza. Il connettore non elimina i child con cardinalità singola contenuti senza l'appartenenza.

## **Istruzioni SQL**

Il connettore può utilizzare istruzioni SQL semplici per le operazioni di selezione, aggiornamento, richiamo o eliminazione. I nomi delle colonne per le istruzioni SQL derivano da una proprietà dell'attributo AppSpecificInfo. Ciascuna query attraversa solo una tabella, se non viene inviata ad una vista.

## Procedure memorizzate

Una **procedura memorizzata** è rappresentata da un gruppo di istruzioni SQL che formano un'unità logica ed eseguono un'attività particolare. Una procedura memorizzata comprende una serie di operazioni o query per consentire l'esecuzione del connettore su un oggetto in un server di database.

Il connettore richiama le procedure memorizzate nei seguenti casi:

- Prima di elaborare un oggetto business, per eseguire processi operativi preparatori
- Dopo l'elaborazione di un oggetto business, per eseguire processi post-operativi
- Per eseguire una serie di operazioni su un oggetto business, invece di utilizzare un'istruzione INSERT, RETRIEVE, UPDATE o DELETE semplice

Quando elabora un oggetto business gerarchico, il connettore può utilizzare una procedura memorizzata per elaborare l'oggetto business di livello superiore o uno qualunque dei relativi oggetti business child. Tuttavia, ciascun oggetto business o serie di oggetti business deve disporre di una proprietà di procedura memorizzata.

### Sintassi ed esempi della procedura memorizzata

Questa sezione descrive la sintassi delle procedure memorizzate per gli oggetti business. Inoltre, comprende degli esempi. Questo capitolo contiene le sezioni seguenti:

- "Aggiunta di attributi all'oggetto business"
- "Sintassi di una procedura memorizzata" a pagina 48
- "Esempi di procedure memorizzate che non restituiscono serie di risultati" a pagina 50
- "Specifiche della procedura memorizzata" a pagina 50

**Aggiunta di attributi all'oggetto business:** Un tipo speciale di attributo deve essere aggiunto all'oggetto business per ciascun tipo di procedura memorizzata elaborata dal connettore. Questi attributi rappresentano solo il tipo di procedura memorizzata e le informazioni specifiche dell'applicazione definite. Questi attributi non utilizzano i parametri delle informazioni specifiche dell'applicazione disponibili per un attributo semplice standard.

L'attributo è denominato in base al tipo di procedura memorizzata da utilizzare. Ad esempio, per configurare il connettore per utilizzare le procedure memorizzate AfterUpdate e BeforeRetrieve, aggiungere gli attributi AfterUpdateSP e BeforeRetrieveSP.

Il connettore riconosce i nomi dell'attributo dell'oggetto business di seguito riportati:

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

RetrieveByContentSP  
 BeforeRetrieveUpdateSP  
 AfterRetrieveUpdateSP  
 RetrieveUpdateSP  
 BeforeDeltaUpdateSP  
 AfterDeltaUpdateSP  
 DeltaUpdateSP

**Nota:** Creare un attributo solo per le procedure memorizzate che si desidera siano eseguite dal connettore. Utilizzare le informazioni specifiche dell'applicazione o di mappatura (solo se viene utilizzato InterChange Server Express come broker di integrazione) per specificare i valori per questi attributi prima che l'oggetto venga inviato al connettore. Riavviare il connettore, in modo che riconosca le modifiche a questi valori per i richiami successivi ad un oggetto business.

Gli attributi dell'oggetto business vengono abilitati per l'utilizzo con i linguaggi bidirezionali. Il formato di questi attributi sarà trasformato in base agli attributi impostati per la proprietà di configurazione standard BiDi.Metadata. Per ulteriori informazioni su questa proprietà, consultare Appendice A, "Proprietà di configurazione standard per i connettori", a pagina 103

**Sintassi di una procedura memorizzata:** Di seguito viene riportata la sintassi per la specifica di una procedura memorizzata:

```
SPN=StoredProcedureName;[RetVal=Attribute_Name|RS:]RS=true|false
[;IP=Attribute_Name1[:Attribute_Name2[:...]]]
[;OP=Attribute_Name1|RS[:Attribute_Name2]RS[:...]]
[;IO=Attribute_Name1[:Attribute_Name2[:...]]]
```

dove:

<i>StoredProcedureName</i>	Il nome della procedura memorizzata.
RetVal	L'esistenza di questo ASI implica che si tratta di un richiamo funzionale e non di un richiamo di procedura, poiché sta per essere restituito un valore dalla funzione. Se il valore restituito è RS, quest'ultimo è costituito da una serie di regole e verrà utilizzato per creare un contenitore con cardinalità N corrispondente a questo oggetto business. Se il valore restituito è un attributo, verrà assegnato a quel determinato attributo nell'oggetto business. Se l'attributo è un altro oggetto business child, viene restituito un errore.
RS	E' true se la procedura memorizzata restituisce una serie di risultati, altrimenti è false. Il valore predefinito è false. Se il valore è true, la proprietà ColumnName nelle informazioni specifiche dell'applicazione di un attributo indica la colonna appropriata nella serie di risultati. Se RS è parte di un elenco di parametri di output, quel determinato parametro restituisce una serie di risultati. E' supportata solo un parametro di serie di risultati OUT. Se viene restituita più di una serie di risultati come parametro di OUT, viene restituita solo la prima serie di risultati e tutte le altre serie vengono ignorate. Per la procedura memorizzata nel

database, il parametro corrispondente dovrebbe restituire un tipo REFCURSOR.

IP	Parametri di input: l'elenco degli attributi dell'oggetto business i cui valori sono utilizzati dal connettore come valori di input durante l'esecuzione della procedura memorizzata.
OP	Parametri di output: l'elenco degli attributi dell'oggetto business cui il connettore restituisce i valori dopo l'esecuzione della procedura memorizzata. Per una descrizione della serie di risultati, si veda RS.
IO	Parametri InputOutput: l'elenco degli attributi dell'oggetto business i cui valori sono utilizzati dal connettore come valori di input cui il connettore restituisce i valori dopo l'esecuzione della procedura memorizzata.

La proprietà `ReturnDummyBOForSP` restituisce i parametri di output anche quando la serie di risultati è true ma vuota. Nel caso di `RetrieveSP`, viene restituita una serie di risultati. Se la serie di risultati è vuota, non vengono creati oggetti business e i parametri di output restituiti dal richiamo della procedura non possono essere richiamati in alcuna modo dal richiamo della procedura. Se `ReturnDummyBOForSP` è true, viene restituito un oggetto business fittizio con valori dai parametri di output e input/output popolati negli attributi corrispondenti. Il valore predefinito di questa proprietà è false.

L'ordine di `StoredProcedureName`, `RS` e dei parametri è importante; tuttavia, l'ordine tra i parametri non è importante. In altre parole, per il connettore non fa differenza se la procedura memorizzata raggruppa tutti i parametri di ciascun tipo o divide i tipi di parametri. Quando vengono raggruppati più parametri dello stesso tipo, i valori sono separati da un delimitatore con due punti e il nome del parametro per ciascun valore non viene ripetuto. I parametri di tipi diversi sono separati da un delimitatore con punto e virgola. I valori del parametro non contengono spazi a destra e a sinistra del segno uguale (=).

**Esempi di procedure memorizzate che restituiscono serie di risultati:** L'esempio di seguito riportato utilizza una procedura memorizzata denominata `CustomerAddressRetrieve`, che restituisce una serie di risultati contenente più indirizzi ed è utilizzata per creare un oggetto business child con cardinalità n.

**Nota:** Le serie di risultati vengono elaborate solo per l'attributo `RetrieveSP` e vengono utilizzate per creare un oggetto business child con cardinalità n.

Per i database DB2 e SQL Server 2000, la serie di risultati è un valore restituito dalla procedura memorizzata.

`CustomerAddressRetrieve` (per i database diversi da Oracle):

```
Attributo: RetrieveSP
ASI:SPN-CustomerAddressRetrieve; RS=true;
ip-CustomerName:IP-CustomerId;
OPErrorStatus;OP-ErrorMsg
```

### Esempi di procedure memorizzate che non restituiscono serie di risultati:

L'esempio di seguito riportato utilizza procedure memorizzate denominate CustomerInsert e VendorInsert che ottengono valori da due attributi di input e restituiscono valori a quattro attributi di output. Gli esempi illustrano strutture diverse per le procedure memorizzate.

- I parametri dello stesso tipo vengono raggruppati insieme (IP, IP, OP, OP, OP, OP, IO):

```
SPN=CustomerInsert;RS=false;IP=LastName:FirstName;OP=CustomerName:
CustomerID:ErrorStatus:ErrorMessage;IO=VendorID
```

- I parametri dello stesso tipo sono separati (IP, OP, OP, OP, IP, IO, OP):

```
SPN=VendorInsert;RS=false;IP=LastName;OP=CustomerName:
CustomerID:ErrorStatus;IP=FirstName;IO=VendorID;OP=ErrorMessage
```

Il connettore supporta solo i tipi di dati semplici supportati dal driver JDBC.

**Specifica della procedura memorizzata:** Esistono due motivi per specificare il nome della procedura memorizzata e i relativi valori del parametro:

- Proprietà AppSpecificInfo dell'attributo

Se il testo che specifica la procedura memorizzata è inferiore o uguale ai 4000 byte, il valore può essere specificato nella proprietà AppSpecificInfo dell'attributo. Questa proprietà può essere utilizzata per specificare la procedura memorizzata indipendentemente dal polling effettuato dal connettore per l'oggetto business (ovvero, l'oggetto business rappresenta un evento dell'applicazione) o se ha ricevuto l'oggetto business come richiesta del broker di integrazione.

L'esempio di seguito riportato illustra la specifica della procedura memorizzata nelle informazioni specifiche dell'applicazione. In questo caso, il valore specificato per la proprietà MaxLength non è importante per la procedura memorizzata.

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

[End]

- Il valore dell'attributo (rilevante solo se viene utilizzato InterChange Server Express come broker di integrazione)

Se la lunghezza del testo che specifica la procedura memorizzata è superiore ai 4000 byte, è necessario utilizzare la mappatura per specificare la procedura memorizzata. La mappatura viene utilizzata per specificare la procedura memorizzata solo se l'oggetto business rappresenta la richiesta del broker di integrazione. In altre parole, un valore di un attributo non può essere utilizzato per specificare una procedura memorizzata quando il connettore sta effettuando il polling degli eventi.

Se il testo della procedura memorizzata è superiore ai 4000 byte e viene utilizzata la mappatura per specificarla, è necessario estendere il valore della proprietà MaxLength affinché venga visualizzato il testo completo.

**Nota:** Se una procedura memorizzata che gestisce un'operazione di creazione, di aggiornamento o di eliminazione viene eseguita in un oggetto business gerarchico contenente una serie di oggetti business child, il connettore

elabora singolarmente ciascun oggetto business child. Ad esempio, se il connettore esegue una procedura memorizzata BeforeCreate, non elabora la serie come un'unità, ma ciascun membro della serie. Quando viene elaborata una procedura memorizzata BeforeRetrieve, il connettore funziona su un oggetto business. Quando viene elaborata una procedura memorizzata AfterRetrieve, il connettore opera su tutti gli oggetti business restituiti dal richiamo.

## **Elaborazione degli oggetti business con procedure memorizzate o istruzioni SQL semplici**

La sezione di seguito riportata illustra il modo in cui il connettore elabora le procedure memorizzate e le istruzioni SQL semplici:

- "Operazione di creazione di un oggetto business"
- "Operazioni di aggiornamento dell'oggetto business" a pagina 52
- "Operazioni di eliminazione di un oggetto business" a pagina 52
- "Operazioni di richiamo di un oggetto business" a pagina 52
- "Serie di risultati restituita dalle operazioni della procedura memorizzata Richiama (RetrieveSP)" a pagina 53
- "Operazioni RetrieveByContent di un oggetto business" a pagina 54
- "Operazioni di richiamo per aggiornamento di un oggetto business" a pagina 54

**Operazione di creazione di un oggetto business:** La creazione di una procedura memorizzata restituisce, in genere, valori che il connettore utilizza per popolare gli attributi semplici nell'oggetto business di livello superiore. Il connettore esegue i passi indicati durante l'elaborazione dell'operazione Crea procedure memorizzate (BeforeCreate, Create, AfterCreate): It c

1. verifica se l'oggetto business contiene un attributo BeforeCreateSP. Altrimenti, richiama la procedura memorizzata BeforeCreate.
2. Se la procedura memorizzata restituisce i valori attraverso i parametri di output, utilizza i valori per impostare il valore degli attributi semplici nell'oggetto business.
3. Crea oggetti business child con cardinalità multipla.
4. Imposta ciascuno dei valori della chiave esterna dell'oggetto business di livello superiore sul valore della chiave principale di ciascun oggetto business child con cardinalità singola.
5. Verifica se l'oggetto business contiene un attributo CreateSP. Altrimenti, richiama la procedura memorizzata Create per creare un oggetto business di livello superiore. Altrimenti, crea ed esegue un'istruzione INSERT per creare un oggetto business di livello superiore.
6. Se la procedura memorizzata Crea restituisce valori mediante i parametri di output, utilizza i valori per impostare il valore degli attributi semplici nell'oggetto business.
7. Imposta il valore della chiave esterna in ciascun child con cardinalità multipla sul valore dell'attributo della chiave principale parent.
8. Crea oggetti business child con cardinalità multipla.
9. Verifica se l'oggetto business contiene un attributo AfterCreateSP. Altrimenti, richiama la procedura memorizzata AfterCreate.
10. Se la procedura memorizzata restituisce i valori attraverso i parametri di output, utilizza i valori per impostare i valori degli attributi semplici nell'oggetto business.

Il connettore può utilizzare i valori restituiti al passo 10 per modificare i valori di un oggetto business creato al passo 3 o 5.

**Operazioni di aggiornamento dell'oggetto business:** L'aggiornamento di un procedura memorizzata restituisce, in genere, valori che il connettore utilizza per popolare gli attributi semplici nell'oggetto business di livello superiore. Il connettore esegue i passi indicati durante l'elaborazione dell'operazione Aggiorna procedure memorizzate (BeforeUpdate, Update, AfterUpdate): It c

1. Verifica se l'oggetto business contiene un attributo BeforeUpdateSP. Altrimenti, richiama la procedura memorizzata BeforeUpdate.
2. Se la procedura memorizzata BeforeUpdate restituisce valori mediante i parametri di output, utilizza i valori per impostare il valore degli attributi semplici nell'oggetto business.
3. Aggiorna oggetti business child con cardinalità multipla.
4. Imposta ciascuno dei valori della chiave esterna dell'oggetto business di livello superiore sul valore della chiave principale di ciascun oggetto business child con cardinalità singola.
5. Verifica se l'oggetto business contiene un attributo UpdateSP. Altrimenti, richiama la procedura memorizzata Update per aggiornare un oggetto business di livello superiore. Altrimenti, crea ed esegue un'istruzione UPDATE per aggiornare un oggetto business di livello superiore.
6. Se la procedura memorizzata Aggiorna restituisce valori mediante i parametri di output, utilizza i valori per impostare il valore degli attributi semplici nell'oggetto business.
7. Imposta i valori della chiave esterna in child con cardinalità multipla per fare riferimento al valore negli attributi della chiave principale corrispondente nel parent.
8. Aggiorna oggetti business child con cardinalità multipla.
9. Verifica se l'oggetto business contiene un attributo AfterUpdateSP. Altrimenti, richiama la procedura memorizzata AfterUpdate.
10. Se la procedura memorizzata restituisce i valori attraverso i parametri di output, utilizza i valori per impostare il valore degli attributi semplici nell'oggetto business.

**Operazioni di eliminazione di un oggetto business:** L'eliminazione di una procedura memorizzata non restituisce valori al connettore. Il connettore esegue i passi indicati durante l'elaborazione dell'operazione Elimina procedure memorizzate (BeforeDelete, Elimina, AfterDelete):

1. Verifica se l'oggetto business contiene un attributo BeforeDeleteSP. Altrimenti, richiama la procedura memorizzata BeforeDelete.
2. Elimina oggetti business child con cardinalità multipla.
3. Elimina oggetti business child con cardinalità multipla.
4. Verifica se l'oggetto business contiene un attributo DeleteSP. Altrimenti, richiama la procedura memorizzata elimina per eliminare l'oggetto business di livello superiore. Altrimenti, crea ed esegue un'istruzione DELETE.
5. Verifica se l'oggetto business contiene un attributo AfterDeleteSP. Altrimenti, richiama la procedura memorizzata AfterDelete.

**Operazioni di richiamo di un oggetto business:** Per le operazioni di richiamo semplici, è possibile utilizzare le procedure memorizzate per l'oggetto business di livello superiore, i child con cardinalità singola e più child con cardinalità multipla. L'ordine delle procedure è il seguente:



- BeforeRetrieve
- Retrieve
- AfterRetrieve

Il connettore crea un oggetto temporaneo per richiamare un oggetto business con cardinalità singola o con cardinalità multipla. Il connettore applica la procedura memorizzata BeforeRetrieve all'oggetto business temporaneo. La procedura memorizzata AfterRetrieve viene applicata a ciascuno degli oggetti child richiamati per il contenitore.

Il connettore esegue la procedura memorizzata AfterRetrieve dopo aver eseguito una query RETRIEVE generata in modo dinamico dai metadati dell'oggetto business o dalla procedura memorizzata sull'oggetto business.

Secondo le specifiche JDBC, sono presenti tre tipi di richiami StoredProcedure:

- {call <spName>(?,?,?)}
- {call <spName>}
- {?= call <spName>(?,?,?)}

dove spName è il nome della procedura memorizzata.

Il connettore supporta i primi due tipi. Elabora ResultSet restituito da StoredProcedure.

Nella sintassi della procedura memorizzata, se RS=true, viene elaborata la serie di risultati dalla procedura memorizzata. Se RS=false, il risultato impostato non viene elaborato. Per impostazione predefinita, il valore RS è false. Una volta elaborati i valori della serie di risultati, vengono elaborate le variabili di output della procedura memorizzata. Se RS=true, i child con cardinalità multipla non possono specificare le variabili di output nella relativa procedura memorizzata.

**Nota:** L'elaborazione della serie di risultati è supportata solo per le operazioni del verbo Richiama e per RetrieveSP.

#### **Serie di risultati restituita dalle operazioni della procedura memorizzata**

**Richiama (RetrieveSP):** ResultSetMetaData è ottenuto per la serie di risultati restituita dalla procedura memorizzata. I valori di tutte le colonne nella serie di risultati sono ottenuti e impostati sull'attributo corrispondente dell'oggetto business. La proprietà ColumnName delle informazioni specifiche dell'applicazione deve contenere il nome della colonna ResultSet affinché l'attributo corrisponda alla colonna.

Per oggetti con cardinalità singola, la serie di risultati corrispondenti deve essere costituita da una sola riga. Se sono restituite più righe nella serie di risultati, viene riportato un errore.

Per i child con cardinalità multipla, possono essere restituite più righe mediante la serie di risultati. Per ciascuna riga restituita, viene creato un nuovo oggetto e aggiunto al contenitore. Il contenitore viene quindi aggiunto all'oggetto parent nell'indice dell'attributo richiesto.

Il child con cardinalità n di un oggetto business wrapper dispone degli attributi della procedura memorizzata e questi ultimi rappresentano i parametri di input e le colonne della serie di risultati. WRAPPER=true è impostato a livello di

informazioni specifiche dell'applicazione dell'oggetto business. Le informazioni specifiche dell'applicazione dell'oggetto business child dispongono di TN=dummy.

**Operazioni RetrieveByContent di un oggetto business:** Per le operazioni RetrieveByContent semplici, le procedure memorizzate possono essere utilizzate solo per l'oggetto business di livello superiore e per il relativo child con cardinalità singola, ovvero non è possibile utilizzarle per restituire una serie di risultati o più righe. L'ordine delle procedure è il seguente:

- BeforeRetrieveByContent
- RetrieveByContent
- AfterRetrieveByContent

Il connettore crea un oggetto temporaneo per richiamare un oggetto business con cardinalità singola o un oggetto business child con cardinalità multipla. Per gli oggetti business con cardinalità multipla, il connettore applica la procedura memorizzata BeforeRetrieveByContent all'oggetto business temporaneo. La procedura memorizzata AfterRetrieveByContent viene applicata a ciascun oggetto child richiamato per il contenitore.

Il connettore esegue la procedura memorizzata AfterRetrieveByContent dopo aver eseguito una query RetrieveByContent generata in modo dinamico dai metadati dell'oggetto business o dalla procedura memorizzata sull'oggetto business. In questo caso, anche se il richiamo di un oggetto business gerarchico richiama anche i relativi oggetti business child, il connettore esegue la procedura memorizzata AfterRetrieveByContent su ogni oggetto business presente nella serie.

**Operazioni di richiamo per aggiornamento di un oggetto business:** Le seguenti procedure memorizzate vengono richiamate sull'oggetto business di livello superiore e richiamano tutti gli oggetti business child nello stesso modo dell'operazione di Richiamo semplice.

L'ordine delle procedure è il seguente:

- BeforeRetrieveUpdate
- RetrieveUpdate
- AfterRetrieveUpdate

Queste procedure memorizzate eseguono le stesse operazioni di BeforeRetrieve e AfterRetrieve. Dispongono di nomi diversi per consentire al connettore di eseguire le operazioni BeforeRetrieve e BeforeRetrieveUpdate, oltre alle operazioni AfterRetrieve e AfterRetrieveUpdate.

Il connettore crea un oggetto temporaneo per richiamare un oggetto business con cardinalità singola o con cardinalità multipla. Per gli oggetti business con cardinalità multipla, il connettore applica la procedura memorizzata BeforeRetrieveUpdate all'oggetto business temporaneo. La procedura memorizzata AfterRetrieveUpdate viene applicata a ciascuno degli oggetti child richiamati per il contenitore.

Il connettore esegue la procedura memorizzata AfterRetrieveUpdate dopo aver eseguito una query RETRIEVE generata in modo dinamico dai metadati dell'oggetto business o dalla procedura memorizzata sull'oggetto business. In questo caso, anche se il richiamo di un oggetto business gerarchico richiama anche i relativi oggetti business child, il connettore esegue la procedura memorizzata AfterRetrieveUpdate su ogni oggetto business presente nella serie.

## Commit e rollback delle transazioni

Ogni volta che il connettore riceve un oggetto business per l'elaborazione, inizia un blocco della transazione. Tutte le istruzioni SQL che il connettore eseguirà durante l'elaborazione dell'oggetto business vengono integrate all'interno del blocco della transazione. Quando il connettore termina l'elaborazione dell'oggetto business, esegue il commit del blocco della transazione se l'elaborazione è corretta oppure esegue il rollback della transazione se si verifica un errore.

---

## Proprietà degli attributi di un oggetto business

L'architettura di un oggetto business definisce varie proprietà che si applicano agli attributi. Questa sezione descrive il modo in cui il connettore interpreta varie proprietà e ne descrive le impostazioni.

### Proprietà del nome

Ciascun attributo dell'oggetto business deve avere un nome univoco.

### Proprietà del tipo

Ciascun attributo dell'oggetto business deve disporre di un tipo, come ad esempio Integer, String o il tipo di un oggetto business child. Quando il connettore rileva un attributo di tipo Date, Long Text o String, inserisce il valore tra gli apici e lo gestisce come dati del carattere.

### Proprietà di cardinalità

Ciascun attributo dell'oggetto business che rappresenta un child o una serie di oggetti business child ha un valore di 1 o n, rispettivamente, per questo attributo. Tutti gli attributi che rappresentano gli oggetti business child dispongono anche di una proprietà `ContainedObjectVersion` (che specifica il numero di versione del child) e una proprietà `Relationship` (che specifica il valore `Containment`).

### Proprietà di lunghezza massima

Se l'attributo è del tipo String, questa proprietà specifica la lunghezza massima consentita per il valore dell'attributo.

### Proprietà chiave

Almeno un attributo semplice in ogni oggetto business deve essere specificato come la chiave. Il valore `Si` definisce l'attributo come chiave. Se l'attributo dell'oggetto business è di tipo String, il tipo di dati nel database dovrebbe essere di tipo Varchar invece di char.

**Nota:** Il connettore non supporta la specifica di un attributo che rappresenta un oggetto business child o una serie di oggetti business child come attributo chiave.

Se la proprietà della chiave viene impostata su `true` per un attributo semplice, il connettore aggiunge l'attributo alla clause `WHERE` delle istruzioni SQL `SELECT`, `UPDATE`, `RETRIEVE` e `DELETE` che genera durante l'elaborazione dell'oggetto business.

Se la proprietà chiave è impostata su `true` per un attributo in un child che memorizza la relazione parent-child nel child (sia con cardinalità singola sia con cardinalità multipla), il connettore utilizza le chiavi principali del parent nella clause `WHERE` dell'istruzione `SELECT` e non utilizza la proprietà chiave. Per informazioni sulla specifica del nome degli attributi dell'oggetto business i cui

valori sono utilizzati per impostare gli attributi della chiave esterna del child, consultare “Informazioni specifiche dell’applicazione a livello di attributo” a pagina 60.

## Proprietà della chiave esterna

Il connettore utilizza questa proprietà per determinare se un attributo è una chiave esterna.

## Proprietà richiesta

La Proprietà richiesta specifica se un attributo deve contenere un valore.

Se questa proprietà viene specificata per un attributo che rappresenta un oggetto business con cardinalità singola, il connettore richiede che l’oggetto business parent contenga un oggetto business child per questo attributo.

Quando il connettore riceve un oggetto business con una richiesta Crea, tale operazione non riesce se si verificano entrambe le condizioni:

- L’oggetto business non dispone di un valore valido o di un valore predefinito per l’attributo richiesto.
- Le informazioni specifiche dell’applicazione non specificano che il connettore genera un ID univoco.

Quando il connettore riceve un oggetto business con una richiesta Richiama e l’oggetto business non dispone di un valore valido o di un valore predefinito per un attributo richiesto, l’operazione di richiamo non riesce.

Il connettore non utilizza questa proprietà per gli attributi che contengono una serie di oggetti business child.

**Nota:** Se l’attributo chiave utilizza una sequenza o un contatore oppure è popolato dal database (UID=AUTO), non deve essere contrassegnato come Richiesto.

## AppSpecificInfo

Per ulteriori informazioni su questa proprietà, consultare “Informazioni specifiche dell’applicazione a livello di attributo” a pagina 60

## Proprietà del valore predefinito

Questa proprietà specifica un valore predefinito che il connettore utilizza per popolare un attributo semplice se non è popolato con un valore dalla tabella di database.

Il connettore non valuta questa proprietà per gli attributi che rappresentano un oggetto business child o una serie di oggetti business child. Il connettore valuta questa proprietà solo se la proprietà di configurazione UseDefaults è impostata su true. Per ulteriori informazioni, consultare Tabella 16 a pagina 127 e “UseDefaults” a pagina 140.

## Valore speciale degli attributi

Gli attributi semplici negli oggetti business possono disporre di un valore speciale CxIgnore. Quando riceve un oggetto business dal broker di integrazione, il connettore ignora tutti gli attributi con un valore CxIgnore. E’ come se tali attributi fossero invisibili per il connettore.

Quando il connettore richiama i dati dal database e l'istruzione SELECT restituisce un valore null per un attributo, il connettore imposta il valore di quell'attributo su CxIgnore per impostazione predefinita. Se è stato specificato un valore per il parametro UNVL delle informazioni specifiche dell'applicazione dell'attributo, il connettore utilizza quel valore per rappresentare il valore null.

Poiché il connettore richiede che ogni oggetto business disponga di almeno un attributo della chiave principale, gli oggetti business inoltrati al connettore devono disporre di almeno una chiave principale non impostata su CxIgnore. La sola eccezione a questo requisito è costituita da un oggetto business la cui chiave principale viene generata dal connettore con un contatore o una sequenza oppure generata dal database.

Quando il connettore inserisce i dati nel database e un attributo di un oggetto business non dispone di alcun valore specificato, il connettore utilizza il valore specificato dalla proprietà UseNullValue dell'attributo. Per ulteriori informazioni su UseNullValue, consultare UNVL=value in Tabella 10 a pagina 61.

---

## Informazioni specifiche dell'applicazione dell'oggetto business

Questa sezione fornisce informazioni sull'attributo a livello di oggetto e sul formato delle informazioni specifiche dell'applicazione per gli oggetti business supportati dal connettore.

Le informazioni specifiche dell'applicazione forniscono al connettore istruzioni dipendenti dall'applicazione su come elaborare gli oggetti business. Il connettore analizza le informazioni specifiche dell'applicazione dagli attributi o dal verbo di un oggetto business o dall'oggetto stesso per generare query per creare, aggiornare, richiamare ed eliminare le operazioni.

Il connettore memorizza alcune delle informazioni specifiche dell'applicazione nella cache e utilizza queste informazioni per creare query per tutti i verbi.

In un oggetto specifico dell'applicazione esteso o modificato, le informazioni specifiche dell'applicazione nella definizione dell'oggetto business devono corrispondere alla sintassi prevista dal connettore.

**Nota:** In un ambiente InterChange Server Express, le prestazioni sono migliori quando gli oggetti specifici dell'applicazione sono inferiori a 1 MB e non dovrebbero mai superare i 5 MB. Gli oggetti business di dimensioni più grandi causano problemi nelle prestazioni a causa di limitazioni di Java Virtual Machine su cui viene eseguito InterChange Server Express.

Tabella 9 a pagina 58 fornisce una panoramica delle funzioni disponibili nelle informazioni specifiche dell'applicazione dell'oggetto business.

Tabella 9. Panoramica delle informazioni specifiche dell'applicazione negli oggetti business supportati

Ambito delle informazioni specifiche dell'applicazione	Funzionalità
Oggetto business completo	<ul style="list-style-type: none"> <li>• Specifica il nome della tabella di database corrispondente</li> <li>• Definisce la colonna il cui valore è utilizzato dal connettore nella clause WHERE per eseguire un'eliminazione logica (o soft)</li> <li>• Specifica che l'oggetto business di livello superiore è un wrapper</li> </ul>
Attributi semplici	<ul style="list-style-type: none"> <li>• Specifica il nome della colonna di database per un attributo</li> <li>• Specifica la relazione della chiave esterna tra un attributo nell'oggetto business corrente e in un oggetto business parent o child.</li> <li>• Specifica la generazione automatica dei valori di un identificativo univoco</li> <li>• Specifica il nome di un altro attributo nello stesso oggetto business il cui valore deve essere utilizzato dal connettore per impostare il valore dell'attributo corrente</li> <li>• Specifica se utilizzare l'attributo corrente durante l'ordinamento dei dati per il richiamo</li> <li>• Specifica il valore da utilizzare quando il valore dell'attributo corrente è null</li> <li>• Specifica il comportamento di sostituzione della stringa</li> <li>• Specifica se utilizzare l'operatore LIKE o = operatore durante il confronto delle stringhe</li> <li>• Specifica il valore da utilizzare come posizione del carattere globale quando viene utilizzato l'operatore LIKE</li> </ul>
Attributi che contengono un oggetto business child o una serie di oggetti business child	<p>Specifica se una child con cardinalità singola appartiene al parent e se il connettore elimina i dati child durante un'operazione di aggiornamento se i dati non vengono rappresentati nell'oggetto business di origine.</p>
Verbo oggetto business	<p>Utilizzato solo per il verbo Richiama, questo testo specifica gli attributi da includere nella clause WHERE per un richiamo, inoltre specifica i valori degli operatori e dell'attributo</p>

## Informazioni specifiche dell'applicazione a livello di oggetto business

Le informazioni specifiche sull'applicazione a livello di oggetto business sono utilizzate per le seguenti attività:

- Specifica il nome della tabella di database corrispondente.
- Fornisce le informazioni necessarie per eseguire un'operazione di eliminazione logica o fisica.
- Specifica che l'oggetto business di livello superiore è un oggetto wrapper.

A livello di oggetto business, il formato di informazioni specifiche dell'applicazione è costituito da parametri separati da delimitatori due punti (:) o punto e virgola (;):

TN=TableName; SCN=StatusColumnName:StatusValue; WRAPPER=true|false;  
PSC=true|false

dove *TableName* identifica la tabella di database, *StatusColumnName* è il nome della colonna di database utilizzato per eseguire le operazioni di eliminazione logica, *StatusValue* è il valore che significa che un oggetto business non è attivo o eliminato, *true|false* indica se l'oggetto business di livello superiore è un oggetto wrapper e il *true|false* finale indica se tutte le istruzioni preparate per l'oggetto business verranno memorizzate nella cache o meno.

Ad esempio, si supponga che un oggetto business Cliente disponga del seguente valore specificato per le relative informazioni specifiche dell'applicazione dell'oggetto business:

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

Si supponga, inoltre, che il connettore riceve una richiesta per eliminare il cliente. Una tale richiesta consente al connettore di inoltrare la seguente istruzione SQL:

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

Se il parametro SCN non è incluso o non è specificato alcun valore, il connettore elimina fisicamente l'oggetto business dal database. In altre parole, se l'oggetto business con il verbo Elimina comprende il parametro SCN nelle informazioni specifiche dell'applicazione, il connettore esegue un'operazione di eliminazione logica. Se l'oggetto business con il verbo Elimina non comprende il parametro SCN nelle informazioni specifiche dell'applicazione, il connettore esegue un'operazione di eliminazione fisica.

Entrambe le operazioni di eliminazione possono utilizzare il valore della proprietà SCN:

- Quando si esegue un'operazione di aggiornamento, il connettore utilizza il valore della relativa proprietà `ChildUpdatePhyDelete` per determinare se eliminare logicamente o fisicamente i dati del child mancanti. Per eliminare logicamente i dati child, il connettore utilizza il valore del parametro SCN per ottenere il nome della colonna di stato e il testo del valore di stato. Per ulteriori informazioni, consultare "Operazioni di aggiornamento" a pagina 43.
- Quando si esegue un'operazione di eliminazione, il connettore utilizza il valore del relativo parametro SCN per determinare se eliminare fisicamente o logicamente tutto l'oggetto business. Se il parametro SCN contiene un valore, il connettore esegue un'operazione di eliminazione logica. Se il parametro SCN non contiene un valore, il connettore esegue un'operazione di eliminazione fisica. Per ulteriori informazioni, consultare "Operazioni di eliminazione" a pagina 46.

I parametri ASI abilitati per l'utilizzo dei linguaggi bidirezionali sono *TableName* e *StatusColumnName*. Il formato per questi parametri verrà trasformato in base agli attributi impostati per la proprietà `BiDi.Metadata`. Per ulteriori informazioni su questa proprietà, consultare Appendice A, "Proprietà di configurazione standard per i connettori", a pagina 103.

A livello di oggetto business, le informazioni specifiche dell'applicazione possono essere utilizzate per specificare un wrapper:

```
WRAPPER=true|false
```

Se il parametro WRAPPER è impostato su *true*, l'oggetto business di livello superiore è un oggetto wrapper. L'oggetto wrapper non è rappresentato da una vista o tabella di database. Un wrapper viene utilizzato come contenitore per gli

oggetti business non correlati. Il connettore ignora l'oggetto di livello superiore ed elabora solo i child. L'oggetto wrapper può contenere n e/o n-1 entità di cardinalità.

A livello di oggetto business, il parametro PSC delle informazioni specifiche dell'applicazione per Prepared Statement Caching abilita o disabilita la memorizzazione nella cache delle istruzioni preparate per l'oggetto business. Tutte le istruzioni dell'oggetto business verranno memorizzate nella cache se PSC è impostato su true e non verrà memorizzato nella cache se PSC è impostato su false.

## Informazioni specifiche dell'applicazione a livello di attributo

Le informazioni specifiche dell'applicazione per gli attributi differiscono se si tratta di un attributo semplice o di un attributo che rappresenta una serie di oggetti business child o un oggetto business child. Le informazioni specifiche dell'applicazione per un attributo che rappresenta un child sono diverse se la relazione parent-child è memorizzata nel child o nel parent. Per ulteriori dettagli sulle informazioni specifiche dell'applicazione per gli attributi che rappresentano una serie di oggetti business child o un oggetto business child, consultare "Specifiche di una chiave estesa di un attributo" a pagina 62.

### Informazioni specifiche dell'applicazione per gli attributi semplici

Per gli attributi semplici, il formato delle informazioni specifiche dell'applicazione è costituito da un numero di parametri nome-valore, ciascuno dei quali comprende il nome del parametro ed i relativi valori. Ciascun parametro impostato è separato da un delimitatore con il simbolo dei due punti (:).

Di seguito viene illustrato il formato delle informazioni specifiche dell'applicazione. Le parentesi quadrate ([ ]) racchiudono un parametro facoltativo. Una barra verticale (|) separa i membri di una serie di opzioni. I due punti costituiscono il delimitatore.

```
CN=col_name:[FK=[fk_object_name.]fk_attribute_name]:
[UID=[AUTO|uid_name|schema_name.uid_name[=UseIfMissing]|CW.uidcolumnname
[=UseIfMissing]]]:
[PH=true|false]:[CA=set_attr_name|..set_attr_name]:[OB=[ASC|DESC]]:[UNVL=value]:
[ESC=true|false]:[FIXEDCHAR=true|false]:
[BYTEARRAY=true|false]:[USE_LIKE=true|false]:
[WILDCARD_POSITION=numero non negativo|NONE|BEGIN|END|BOTH]:
[CLOB=true]
[TS=true|false]]
```

Il solo parametro richiesto affinché un attributo semplice sia elaborato dalla connettore è il nome della colonna. Ad esempio, di seguito è riportato il formato per specificare solo il nome della colonna:

```
CN=customer_id
```

I parametri ASI dell'attributo abilitati per l'utilizzo con i linguaggi bidirezionali sono nomi di colonne, nomi di ID univoci, nomi di colonne di ID univoci e valori. Il formato di questi parametri, escluso il valore, verrà trasformato in base agli attributi impostati per la proprietà BiDi.Metadata. I valori verranno trasformati in base agli attributi impostati per la proprietà BiDi.Application. Per ulteriori informazioni su tali proprietà, consultare Appendice A, "Proprietà di configurazione standard per i connettori", a pagina 103.

Tabella 10 a pagina 61 descrive ciascun parametro nome-valore.



Tabella 10. I parametri nome-valore nelle informazioni specifiche dell'applicazione

Parametro	Descrizione
CN= <i>col_name</i>	Il valore di questo parametro è il nome della colonna di database per questo attributo.
FK=[ <i>fk_object_name.</i> ] <i>fk_attribute_name</i>	Il valore di questo parametro dipende dalla relazione parent-child memorizzata nell'oggetto business parent o child. Se un attributo non è una chiave esterna, non includere questo parametro nelle informazioni specifiche dell'applicazione. Per ulteriori informazioni, consultare "Specifiche di una chiave estesa di un attributo" a pagina 62.
UID=AUTO	Il connettore utilizza questo parametro per generare l'ID univoco per l'oggetto business. Se un attributo non richiede un ID univoco, non includere questo parametro nelle informazioni specifiche dell'applicazione. Consultare la descrizione della proprietà
UID= <i>uid_name</i>   <i>schema_name.uid_name</i> [=UseIfMissing]	PreserveUIDSeq per i dettagli sul modo in cui preservare l'ID univoco durante l'elaborazione dell'oggetto business. Per ulteriori informazioni, consultare "Generazione di un identificativo univoco di un oggetto business" a pagina 65.
UID=CW. <i>uidcolumnname</i> [=UseIfMissing]	<b>Nota:</b> CW è una parola chiave utilizzata per rappresentare il tipo di UID e non rappresenta il nome della tabella.
PH=true   false	Se PH=true, l'attributo semplice corrispondente è un attributo segnalibro. Un attributo semplice è anche un segnalibro se le relative informazioni specifiche dell'applicazione sono vuote o null.
CA= <i>set_attr_name</i>   <i>..set_attr_name</i>	Se <i>set_attr_name</i> è impostato sul nome di un altro attributo all'interno dell'oggetto business individuale corrente, il connettore utilizza il valore dell'altro attributo per impostare il valore di questo attributo prima di aggiungere l'oggetto business al database durante un'operazione di creazione. Il valore di <i>set_attr_name</i> non può fare riferimento ad un attributo nell'oggetto business parent se <i>set_attr_name</i> viene preceduto da due punti. Se non si include questo parametro nelle informazioni specifiche dell'applicazione, il connettore utilizza il valore dell'attributo corrente senza copiare il valore CA da un altro attributo.
OB=[ASC   DESC]	Se un valore viene specificato per questo parametro e l'attributo è un oggetto business child, il connettore utilizza il valore dell'attributo nella clause ORDER BY delle query di richiamo. Il connettore può richiamare gli oggetti business child in ordine crescente (ASC) o decrescente (DESC). Se non si include questo parametro nelle informazioni specifiche dell'applicazione, il connettore non utilizza questo attributo quando si specifica l'ordine di richiamo.
UNVL= <i>value</i>	Questo parametro specifica il valore che usa il connettore per rappresentare un valore nullo quando richiama un oggetto business con attributi valutati nulli. Se non si include questo parametro nelle informazioni specifiche dell'applicazione, il connettore inserisce CxIgnore per il valore dell'attributo.
ESC=[true   false]	Questo parametro specifica se il connettore sostituisce tutte le istanze di ciascun carattere identificato nella proprietà ReplaceAllStr con le stringhe di sostituzione specificata nella proprietà ReplaceStrList. Se il parametro non contiene un valore, il connettore utilizza il valore della proprietà ReplaceStrList per rendere questa determinazione. <b>Nota:</b> Il parametro ESC e le proprietà ReplaceAllStr e ReplaceStrList forniscono supporto per le funzioni del carattere per l'uscita dal database (ad esempio, per uscire dagli apici singoli).
FIXEDCHAR=true   false	Questo parametro specifica se l'attributo è di lunghezza fissa quando le colonne nella tabella sono di tipo CHAR, non VARCHAR. Ad esempio, se un determinato attributo viene collegato ad una colonna di tipo CHAR, il connettore prevede FIXEDCHAR per la lunghezza, per le informazioni specifiche dell'applicazione dell'attributo, specificare true. Assicurarsi che la proprietà MaxLength dell'attributo sia di lunghezza CHAR specificata nel database. Per impostazione predefinita FIXEDCHAR=false.

Tabella 10. I parametri nome-valore nelle informazioni specifiche dell'applicazione (Continua)

Parametro	Descrizione
BYTEARRAY=true false	Se BYTEARRAY=true, il connettore legge e scrive i dati binari nel database e invia i dati come stringa a InterChange Server Express. BYTEARRAY=false Per ulteriori informazioni, consultare "Operazioni con i dati binari" a pagina 66.
USE_LIKE=true false	Specifica se il connettore confronta stringhe utilizzando l'operatore = o l'operatore LIKE. Se USE_LIKE viene impostato su true, le query di caratteri globali possono essere eseguite impostando WILDCARD_POSITION. Se USE_LIKE è impostato su false, verrà utilizzato l'operatore =.
WILDCARD_POSITION=non-negative number NONE BEGIN END BOTH	Se USE_LIKE è true, viene utilizzato WILDCARD_POSITION per specificare la posizione del carattere globale. Questo valore può essere un numero non negativo, NONE, BEGIN, END o BOTH. Ad esempio, se si utilizza BEGIN il carattere globale si trova nella prima posizione della stringa (%string). Utilizzando END il carattere globale viene posizionato alla fine della stringa (stringa%). Utilizzando BOTH i caratteri globali vengono posizionati all'inizio e alla fine della stringa (%stringa%).
CLOB=true	Applicabile solo per il tipo di attributo Stringa. Specifica che la colonna del database che corrisponde a questo attributo è un tipo di dati CLOB. <b>Nota:</b> Un tipo di dati CLOB viene definito come segue: <ul style="list-style-type: none"> <li>• L'attributo CLOB dispone di una stringa la cui lunghezza viene utilizzata per definire la lunghezza del CLOB</li> <li>• L'attributo CLOB dispone di ASI=CN=xyz; CLOB=true</li> <li>• Qualunque altro tipo di attributo con il riferimento su CLOB nelle informazioni specifiche all'applicazione restituisce una condizione di errore.</li> <li>• CLOB=false restituisce un errore</li> </ul>
TS=true false	Per gli attributi di tipo DATE, quando è specificato TS=false nelle informazioni specifiche all'applicazione dell'attributo, quest'ultimo viene trattato come un tipo DATE. Quando TS=true, l'attributo viene trattato come tipo TIMESTAMP. Il valore predefinito di TS è true.

**Nota:** Se nessuna delle informazioni specifiche dell'applicazione in qualunque attributo dell'oggetto business causa la creazione o l'esecuzione di una query da parte del connettore, il connettore registra un'avvertenza e continua il funzionamento. Non restituisce un'eccezione o un errore.

**Specifica di una chiave esterna di un attributo:** Il valore di questa proprietà dipende dalla relazione parent-child memorizzata nell'oggetto business parent o child:

- Memorizzato nella serie parent, il valore per includere il tipo di oggetto business child e il nome dell'attributo nel child da utilizzare come chiave esterna.
- Nella serie child, il valore per includere solo il nome dell'attributo nel parent da utilizzare come chiave esterna.

Se il valore `fk_object_name` non corrisponde al tipo di oggetto business child e il valore di `fk_attribute_name` non corrisponde al nome dell'attributo nel parent o nel child (come applicabile), il connettore non può elaborare questo attributo come chiave esterna. Il nome dell'oggetto business e dell'attributo sono sensibili al maiuscolo o al minuscolo.

Ad esempio, si supponga che l'oggetto business Cliente contiene l'attributo `Addr[1]`, che rappresenta l'oggetto business child Indirizzo e l'attributo `AID`, che memorizza la chiave principale dell'oggetto business child come chiave esterna. In questo caso, le informazioni specifiche dell'applicazione dell'attributo della chiave

esterna parent devono contenere il tipo di oggetto business child (Indirizzo) oltre al nome dell'attributo della chiave principale (ID). In questo esempio, le informazioni specifiche dell'applicazione dell'attributo AID comprendono FK=Address.ID.

**Denominazione di un attributo della chiave esterna:** Più oggetti business parent possono contenere lo stesso oggetto business child, indipendentemente dal child memorizzato con cardinalità singola o multipla e dalla relazione parent-child memorizzata sul parent o sul child. Tuttavia, tutti gli oggetti business parent che memorizzano la relazione parent-child devono utilizzare attributi denominati in modo identico per contenere la chiave principale del child. Inoltre, tutti gli oggetti business child che memorizzano la relazione parent-child devono utilizzare attributi denominati in modo identico per contenere la chiave principale del parent. Figura 13 illustra queste relazioni.

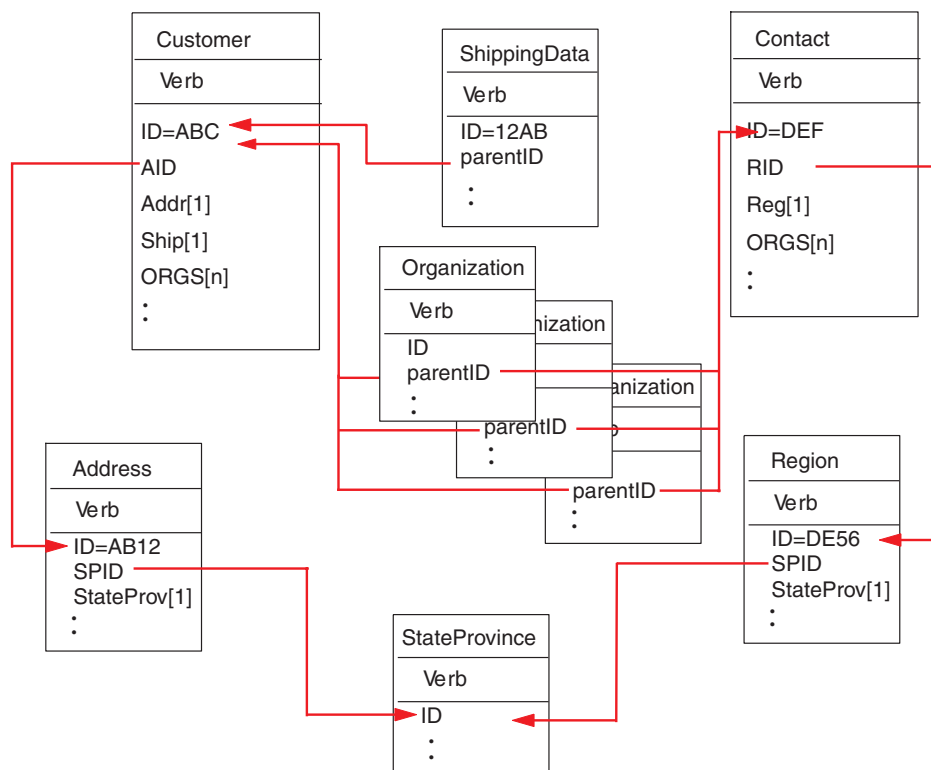


Figura 13. Esempio di relazioni tra gli oggetti business

Figura 13 illustra le seguenti relazioni:

- L'attributo ORGS[n] di Cliente ABC e Contratto DEF rappresenta una serie di oggetti business Organizzazione. Il valore della chiave esterna per ciascun oggetto business nella serie Organizzazione corrisponde al valore della chiave principale nell'attributo ID negli oggetti business Cliente e Contratto. In questo caso, ciascun oggetto business nella serie è contenuto da più parent.

Le informazioni specifiche all'applicazione per l'attributo ORGS potrebbero essere:

KEEP\_RELATIONSHIP=true

Per ulteriori informazioni sul parametro KEEP\_RELATIONSHIP, consultare la sezione "Informazioni specifiche dell'applicazione per gli attributi che rappresentano i child" a pagina 66.

Le informazioni specifiche all'applicazione per l'attributo parentID di ciascun child e nella serie Organizzazioni contengono il nome della colonna nel database

corrispondente all'attributo corrente e specificano la chiave esterna dell'attributo corrente contenendo il nome dell'attributo della chiave principale del parent, ad esempio:

CN=ORG\_ID:FK=ID

**Nota:** Affinché più oggetti business contengano lo stesso child (dove la relazione parent-child è memorizzata nel child), tutti gli oggetti business parent devono utilizzare un attributo denominato in modo identico per contenere la chiave esterna per il child. Il parametro della chiave esterna delle informazioni specifiche all'applicazione del child identificano solo il nome dell'attributo e non il tipo di oggetto business parent. Il connettore suppone che il parent diretto è il proprietario di ciascun child.

- L'attributo Addr[1] del Cliente rappresenta l'oggetto business Indirizzo con l'appartenenza. L'attributo AID Cliente identifica la chiave principale dell'oggetto business Indirizzo come chiave esterna nel parent. In questo caso, l'attributo della chiave esterna del parent deve contenere il tipo di oggetto business child, oltre al nome del relativo attributo della chiave principale. Il child con cardinalità singola, Indirizzo, è contenuto da un solo parent.

Le informazioni specifiche dell'applicazione per l'attributo Addr sono:

CONTAINMENT=OWNERSHIP

Le informazioni specifiche all'applicazione per l'attributo AID contengono il nome della colonna nel database corrispondente all'attributo corrente e specificano la chiave esterna dell'attributo corrente contenendo il nome dell'attributo della chiave principale del parent, ad esempio:

CN=FK\_AD:FK=Address.ID

Le informazioni specifiche dell'applicazione per l'attributo della chiave principale del child sono:

CN=pk

- Gli attributi StateProv[1] degli oggetti business Indirizzo e Regione rappresentano l'oggetto business StateProvince senza appartenenza. Gli attributi SPID degli oggetti business Indirizzo e Regione contengono il tipo di oggetto business (StateProvince) e il nome dell'attributo della chiave principale, che serve come chiave esterna del parent. Lo stesso child con cardinalità singola, StateProvince, è contenuto da più parent.

Le informazioni specifiche dell'applicazione per l'attributo SPID sono:

CONTAINMENT=NO\_OWNERSHIP

Per ulteriori informazioni sul parametro CONTAINMENT, consultare "Informazioni specifiche dell'applicazione per gli attributi che rappresentano i child" a pagina 66.

Le informazioni specifiche all'applicazione per l'attributo Indirizzo SPID contengono il nome della colonna nel database corrispondente all'attributo corrente e specificano la chiave esterna dell'attributo corrente contenendo il nome dell'attributo della chiave principale del parent, ad esempio:

CN=FK\_SP:FK=StateProvince.ID

Le informazioni specifiche dell'applicazione per l'attributo della chiave principale del child sono:

CN=SP\_ID

**Nota:** Affinché più oggetti business contengano lo stesso child (che memorizza la relazione parent-child nel parent), tutti gli oggetti business child devono utilizzare un attributo denominato in modo identico per contenere la chiave esterna per il parent.

- L'attributo Ship[1] Cliente rappresenta un oggetto business ShippingData che contiene le informazioni di inoltro del cliente. L'attributo ID delle funzioni Cliente come chiave esterna per i dati di inoltro. In questo caso, poiché ShippingData non può esistere indipendentemente dal relativo parent e viene creato solo dopo la creazione del parent, la relazione parent-child viene memorizzata nel child.

Le informazioni specifiche all'applicazione per l'attributo parentID del child contengono il nome della colonna nel database corrispondente all'attributo corrente e specificano la chiave esterna dell'attributo corrente contenendo il nome dell'attributo della chiave principale del parent, ad esempio:

CN=SD\_ID:FK=ID

**Generazione di un identificativo univoco di un oggetto business:** Il connettore utilizza il parametro UID per generare l'ID univoco per l'oggetto business. Il connettore genera ID univoci utilizzando sequenze come DB2 o i conteggi (strutturati come tabelle) e quindi inoltra l'istruzione INSERT.

IBM DB2 e Microsoft SQL Server non richiedono l'inoltro dell'ID in un'istruzione INSERT. Invece, generano l'ID al momento della creazione. Una volta completata correttamente la creazione dell'oggetto business, il connettore può richiamare ed utilizzare questo valore.

Il connettore utilizza una sequenza o un contatore per generare il valore dell'ID e quindi inoltra l'istruzione INSERT:

- Se UID = AUTO, il database genera l'ID e il connettore deve richiamarlo. Questa impostazione è disponibile solo per i database IBM DB2 e Microsoft SQL Server.
- Se UID =uid\_name, il valore di uid\_name fornisce il nome della sequenza DB2 che il connettore utilizza per generare un ID univoco per l'attributo. Dopo che il connettore ha aperto il valore della sequenza, popola l'attributo chiave e inoltra l'istruzione INSERT. Questa sintassi è al momento utilizzata solo per database DB2.
- Se UID =uid\_name=UseIfMissing e se il valore dell'attributo non è CxIgnore, il connettore utilizza il valore dell'attributo piuttosto che generare un ID univoco. Il parametro =UseIfMissing non può contenere spazi e non è sensibile al maiuscolo. Questa opzione è disponibile solo per DB2.
- Se UID=CW.uidcolumnname, il connettore utilizza una tabella del contatore per generare un ID univoco per l'attributo. La tabella, il cui nome è configurabile, viene creata con una singola colonna denominata id. E' possibile personalizzare la tabella per aggiungere una colonna per ciascun attributo che richiede la generazione di un UID. Utilizzare il parametro uidcolumnname per specificare il nome della colonna per il connettore da utilizzare per la generazione dell'ID univoco. Si noti che il connettore supporta solo il tipo di dati numerici per le colonne che richiedono la generazione di un UID.

Per informazioni sulla configurazione dei nomi delle tabelle, consultare UniqueIDTableName. Gli script per l'installazione di questa tabella sono:

```
\connectors\JDBC\dependencies\uid_table_mssqlserver.sql
```

```
\connectors\JDBC\dependencies\uid_table_db2.sql
```

- Se UID=CW.uidcolumnname=UseIfMissing e se il valore dell'attributo non è CxIgnore, il connettore utilizza il valore dell'attributo piuttosto che generare un ID univoco. Il parametro =UseIfMissing non può contenere spazi e non è sensibile al maiuscolo/minuscolo.

Per informazioni sulla conservazione della sequenza dell'ID univoco durante l'elaborazione, fare riferimento alla proprietà "PreserveUIDSeq" a pagina 136.

### **Informazioni specifiche dell'applicazione per gli attributi che rappresentano i child**

Gli attributi che rappresentano un oggetto business child con cardinalità singola possono specificare se il child appartiene al parent o è condiviso tra più parent.

Gli attributi che rappresentano un child con cardinalità singola o una serie di oggetti business child possono specificare il comportamento del connettore durante l'aggiornamento del parent e una serie secondaria di child.

**Attributi che rappresentano un oggetto business child con cardinalità singola:** Il formato delle informazioni specifiche dell'applicazione per gli attributi che rappresentano un oggetto business child con cardinalità singola è:

CONTAINMENT= [OWNERSHIP|NO\_OWNERSHIP]

Impostare CONTAINMENT su OWNERSHIP per rappresentare una relazione con cardinalità singola dove il parent appartiene all'oggetto business child. Impostare CONTAINMENT su NO\_OWNERSHIP per rappresentare una relazione con cardinalità singola dove il parent condivide l'oggetto business child. Non includere il parametro CONTAINMENT quando si rappresenta una relazione con cardinalità singola che memorizza la relazione nel child piuttosto che nel parent.

Per ulteriori informazioni consultare "Relazioni con cardinalità singola e dati senza proprietà" a pagina 35 e "Relazioni con cardinalità singola che memorizzano la relazione nel child" a pagina 37.

**Attributi che rappresentano un child che memorizza la chiave del parent:** Per le operazioni Aggiorna sulla serie di oggetti business che memorizzano la relazione parent-child nel child, è presente un valore particolare per l'attributo che rappresenta il child: è possibile impostare KEEP\_RELATIONSHIP su true per impedire al connettore l'eliminazione dei dati child esistenti che non sono rappresentati nell'oggetto business di origine.

Ad esempio, si supponga che un contratto esistente sia associato ad un luogo esistente, come ad esempio New York. Si supponga, inoltre, che il connettore riceve una richiesta per aggiornare un oggetto business Contratto contenente un oggetto business child singolo che associa San Francisco come luogo. Se KEEP\_RELATIONSHIP è impostato su true per l'attributo che rappresenta i dati del luogo, il connettore aggiorna il contratto e ne aggiunge l'associazione con San Francisco senza eliminare l'associazione con New York.

Tuttavia, se KEEP\_RELATIONSHIP è impostato false, il connettore elimina tutti i dati child esistenti non contenuti nell'oggetto business di origine. In tale caso, il contratto viene associato solo con San Francisco.

Il formato di queste informazioni specifiche dell'applicazione è:

KEEP\_RELATIONSHIP=[true|false]

Il minuscolo/maiuscolo viene ignorato durante la verifica di queste informazioni specifiche dell'applicazione.

**Operazioni con i dati binari:** Se BYTEARRAY=true, il connettore legge e scrive i dati binari nel database. Poiché non è presente alcuna supporto per i dati binari nel sistema WebSphere Business Integration Server Express, tali dati binari vengono

convertiti in Stringa e inviati al broker di integrazione. Il formato di questa stringa è un numero esadecimale con 2 caratteri per byte. Ad esempio, se i dati binari nel database è di 3 byte con i valori (decimali) (1, 65, 255), la stringa sarà "0141ff".

## Informazioni specifiche dell'applicazione formato per i verbi

Il connettore utilizza le informazioni specifiche dell'applicazione per i verbi Retrieve e RetrieveByContent. Questo testo consente di specificare gli attributi da includere nella clause WHERE per un richiamo. È inoltre possibile specificare i valori dell'attributo e degli operatori.

Di seguito viene riportata la sintassi per le informazioni specifiche dell'applicazione per i verbi Retrieve e RetrieveByContent:

```
[condition_variable conditional_operator @ [...]:[..]attribute_name [, ...]]
```

dove:

<i>condition_variable</i>	Il nome della colonna di database.
<i>conditonal_operator</i>	L'operatore supportato dal database, ad esempio =, >, OR, AND e IN ( <i>value1</i> , <i>value2</i> ).
@	Una variabile che è sostituita con il valore richiamato da <code>getAttrValue(attribute_name)</code> . La sostituzione riguarda la posizione, ovvero il connettore sostituisce il primo @ con il valore della prima variabile <i>attribute_name</i> specificata dopo il delimitatore :.
..	L'attributo specificato nella variabile <i>attribute_name</i> appartiene all'oggetto business parent immediato, se questo valore manca, l'attributo appartiene all'oggetto business corrente.
<i>attribute_name</i>	Il nome dell'attributo il cui valore viene sostituito dal connettore con @.

Per comprendere la sintassi di questa proprietà, si supponga che un oggetto business Voce disponga di un attributo `item_id` il cui valore è XY45 e un attributo Colore il cui valore è ROSSO. Inoltre, si supponga di specificare la proprietà `AppSpecificInfo` del verbo Retrieve come:

```
Color='RED'
```

Il valore delle informazioni specifiche dell'applicazione consente la creazione da parte del connettore della seguente clause WHERE per un richiamo:

```
dove item_id=XY45 e Color = 'RED'
```

Per un esempio più complicato, si supponga che l'oggetto business Cliente dispone dell'attributo `customer_id` il cui valore è 1234 e un attributo `creation_date` il cui valore è 01/01/90. Inoltre, si supponga che questo parent dell'oggetto business dispone di un attributo `quantity` il cui valore è 20.

Inoltre, si supponga di specificare la proprietà `AppSpecificInfo` del verbo Retrieve come:

```
creation_date > @ OR quantity = @ AND customer_status IN ('GOLD', 'PLATINUM') : creation_date, ..quantity
```

Il valore delle informazioni specifiche dell'applicazione consente la creazione da parte del connettore della seguente clause WHERE per un richiamo:

```
dove customer_id=1234 e creation_date > '01/01/90'  
OR quantity = 20 AND customer_status IN ('GOLD', 'PLATINUM')
```

Il connettore ottiene il valore data ('01/01/90') dall'attributo `creation_date` nell'oggetto business corrente. Ottiene il valore quantity (20) dall'attributo quantity nell'oggetto business parent (come indicato da `..quantity` nelle informazioni specifiche dell'applicazione).

Dopo che il connettore ha analizzato le informazioni specifiche dell'applicazione per il verbo Richiama, aggiunge il testo alla clause WHERE dell'istruzione RETRIEVE che crea dalle chiavi principali o esterne dell'oggetto business. Il connettore aggiunge la guida AND alla clause WHERE. Il valore delle informazioni specifiche dell'applicazione deve essere costituito da una sintassi SQL valida. Nel caso di RetrieveByContent, le informazioni specifiche dell'applicazione vengono aggiunte alla clause WHERE dell'istruzione RETRIEVE create dagli attributi dell'oggetto business popolati con i relativi valori.

La clause WHERE può fare riferimento anche agli attributi segnalibro invece che agli attributi correnti nell'oggetto business parent. Questi segnalibri non dispongono di informazioni specifiche dell'applicazione. Un attributo può essere un segnalibro se soddisfa una delle seguenti condizioni per i relativi ASI:

1. Attributo semplice con ASI=null or ''
2. Attributo semplice con ASI=PH=TRUE

Ad esempio: un oggetto business Ordine contiene un oggetto business con la voce alla riga di cardinalità multipla ed è necessario il richiamo solo di voci della riga specifiche. Questo richiamo può essere gestito mediante un attributo del segnalibro nell'oggetto business Ordine. Questo segnalibro viene richiesto nell'oggetto parent. L'attributo del segnalibro può essere popolato al runtime dal broker di integrazione con un elenco delle voci della riga specifica, separate da una virgola (,).

Per questo esempio, aggiungere le seguenti informazioni alla clause WHERE per il verbo retrieve sull'oggetto business della voce della riga child:

```
line_item_id in(@,@, @):..placeholder1,..placeholder2,..placeholder3
```

Dove `line_item_id in` è l'ID nell'oggetto business child, `placeholder` è l'attributo nel parent. Se `placeholder` contiene i valori 12,13,14 la query seleziona quanto segue dalla clause WHERE:

```
line_item_id in(12,13,14)
```

Dove `SELECT:..FROM:..WHEREx in (1,2,3)` è una sintassi SQL del database.

Nell'espressione RetrieveByContent, se la lunghezza della WHERE clause è 0, il connettore utilizza le informazioni specifiche dell'applicazione nella WHERE dell'istruzione RETRIEVE. Con questa funzione, l'utente può inviare un oggetto business senza valori per gli attributi popolati e specificare le informazioni specifiche all'espressione per RetrieveByContent, quindi il connettore crea la WHERE clause in base a ciò che è stato specificato nelle informazioni specifiche dell'applicazione dell'espressione.

Per le espressioni Retrieve e RetrieveByContent, sono abilitati gli attributi ASI *variabile condizionale* e *operatore condizionale* per l'utilizzo con i linguaggi bidirezionali. Il formato per *condition variable* viene trasformato in base agli attributi impostati per la proprietà BiDi.Metadata. L'*Operatore condizionale* viene trasformato in base agli attributi impostati per la proprietà BiDi.Application. Per ulteriori informazioni su tali proprietà, fare riferimento a Appendice A, "Proprietà di configurazione standard per i connettori", a pagina 103



---

## Capitolo 5. Configurazione del connettore

In questo capitolo viene descritto come configurare WebSphere Business Integration Server Express Adapter per JDBC. Questo capitolo contiene le sezioni seguenti:

- “Abilitazione del supporto multi-driver”
- “Abilitazione della classe del gestore oggetti business personalizzati”
- “Impostazione delle proprietà di configurazione del connettore” a pagina 70
- “Utilizzo di Connector Configurator Express” a pagina 71

---

### Abilitazione del supporto multi-driver

E' possibile specificare il driver JDBC procedendo nel modo seguente:

1. Installare il driver sulla macchina.
2. Inserire tutte le librerie dinamiche richieste dal connettore al runtime nella directory connectors/JDBC nella directory del prodotto.
3. Modificare il file di avvio del connettore includendo tutti i percorsi di classe rilevanti (comprese le informazioni sulla licenza, se richieste) nella variabile JDBCDRIVERPATH.

Su Linux, il file di avvio è:

```
ProductDir/connectors/JDBC/start_JDBC.sh
```

Su Windows, il file di avvio è:

```
ProductDir\connectors\JDBC\start_JDBC.bat
```

Su i5/OS, il file di avvio è:

```
ProductDir/QIBM/UserData/WBIServer44/QWBIDFT/connectors/JDBC/sart_JDBC.sh
```

4. Specificare un valore per la JDBCDriverClass proprietà della configurazione.

**Nota:** Per tutte le funzioni supportate, il connettore può funzionare con qualunque driver che segue le specifiche JDBC 2.0 o versioni successive. Se il driver non supporta una funzione particolare il connettore non funziona correttamente. Ad esempio, se il driver non supporta tutte le chiamate di metodo usate da JDBC ODA, il log JDBC ODA indica il processo che il driver non supporta. In tal caso, occorre utilizzare un driver differente.

---

### Abilitazione della classe del gestore oggetti business personalizzati

Il connettore supporta la classe del gestore oggetti business personalizzati CustomBOH. Quindi, implementa l'interfaccia JDBCBOhandlerInterface. La sintassi dell'interfaccia è la seguente:

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

Quando si implementa il metodo doVerbForCustom, verificare che venga inoltrato ma che non restituisca le due eccezioni. Inoltre, impostare lo stato e il messaggio di ciascuna eccezione prima di inoltrarli.

- `VerbProcessingFailedException`— Restituita quando l'operazione specificata dall'espressione non riesce.
- `ConnectionFactoryException`— Restituita quando il connettore non può stabilire una connessione con l'applicazione.

Affinché il connettore supporti questo gestore dell'oggetto business:

- Specificare il nome della classe `CustomBOH` nelle informazioni specifiche dell'applicazione dell'espressione.

Il connettore ottiene il nome della classe del gestore dell'oggetto business personalizzato dalla classe specifica all'applicazione dell'espressione. Utilizzare la seguente sintassi:

```
CustomBOH=customBOhandlerClassName
```

Ad esempio, si supponga che le informazioni specifiche all'applicazione dell'espressione vengono indicate nel modo seguente:

```
CustomBOH=JDBCBOhandlerForOverrideSQL
```

In questo caso, `JDBCBOhandlerForOverrideSQL` è il nome della classe del gestore dell'oggetto business.

- Verificare che `CustomBOH` appartiene `com.crossworlds.connectors.JDBC`  
Se il connettore trova "`CustomBOH=`" nelle informazioni specifiche all'applicazione dell'espressione e trova a sua volta la classe nel pacchetto `com.crossworlds.connectors.JDBC`, esegue il gestore dell'oggetto business personalizzato. Se non trova `CustomBOH`, restituisce un errore nel cui messaggio viene specificato che non è stato possibile trovare la classe.

---

## Impostazione delle proprietà di configurazione del connettore

È necessario impostare le proprietà di configurazione standard e specifiche del connettore prima di poterle eseguire. Utilizzare lo strumento `Connector Configurator Express` per impostare le proprietà di configurazione del connettore. Accedere a questo strumento da `System Manager`.

Per ulteriori informazioni su `Connector Configurator Express`, consultare "Utilizzo di `Connector Configurator Express`" a pagina 71.

## Proprietà di configurazione standard

Le proprietà di configurazione standard forniscono informazioni utilizzate da tutti i connettori. Consultare Appendice A, "Proprietà di configurazione standard per i connettori", a pagina 103 per informazioni dettagliate su tali proprietà. Il connettore per JDBC supporta `InterChange Server Express`.

Inoltre, fare riferimento a Tabella 11 per le informazioni di configurazione specifiche di `WebSphere Business Integration Server Express Adapter per JDBC`. Le informazioni contenute in questa tabella integrano le informazioni contenute nell'appendice.

*Tabella 11. Informazioni sulle proprietà specifiche per questo connettore*

Proprietà	Note
<code>CharacterEncoding</code>	Questo connettore non utilizza la proprietà <code>CharacterEncoding</code>
<code>Locale</code>	Poiché tale connettore è stato internazionalizzato, è possibile modificare il valore della proprietà <code>Locale</code> .

Si noti che occorre fornire un valore per la proprietà della configurazione `ApplicationName` prima di eseguire il connettore.

## Proprietà di configurazione specifiche dei connettori

Le proprietà di configurazione specifiche dei connettori forniscono informazioni necessarie dal connettore al runtime. Le proprietà specifiche dei connettori inoltre forniscono un modo di modificare le informazioni statiche o logiche all'interno del connettore senza doverlo ricodificare o ricreare.

Consultare Appendice B, "Proprietà di configurazione specifiche dei connettori", a pagina 127 per informazioni dettagliate su tali proprietà.

---

## Utilizzo di Connector Configurator Express

In questa sezione viene descritto in che modo utilizzare Connector Express per impostare i valori delle proprietà di configurazione per l'adattatore.

Utilizzare Connector Configurator Express per:

- Creare una maschera di proprietà specifiche dei connettori per configurare il connettore.
- Creare un file di configurazione.
- Impostare le proprietà in un file di configurazione

**Nota:** In questo documento, le barre rovesciate (\) vengono utilizzare come convenzione per i percorsi di directory. Per le installazioni Linux e i5/OS, sostituire le barre (/) alle barre inverse e seguire le convenzioni per ciascun sistema operativo.

Gli argomenti in questa sezione sono:

- "Panoramica su Connector Configurator Express"
- "Avvio di Connector Configurator Express" a pagina 72
- "Esecuzione di Connector Configurator Express da System Manager" a pagina 73
- "Creazione di una maschera delle proprietà specifiche per un connettore" a pagina 73
- "Creazione di un nuovo file di configurazione" a pagina 76
- "Utilizzo di un file esistente" a pagina 77
- "Completare un file di configurazione" a pagina 78
- "Impostazione delle proprietà del file di configurazione" a pagina 78
- "Salvataggio del file di configurazione" a pagina 84
- "Modifica di un file di configurazione" a pagina 84
- "Completamento della configurazione" a pagina 84
- "Utilizzando Connector Configurator Express in un ambiente globalizzato." a pagina 85

---

## Panoramica su Connector Configurator Express

Connector Configurator Express consente di configurare il componente dei connettori dell'adattatore per utilizzarlo con InterChange Server Express.

Utilizzare Connector Configurator Express per:

- Creare una **maschera delle proprietà specifiche dei connettori** per configurare il connettore.
- Creare un **file di configurazione del connettore** ; occorre creare un file di configurazione per ciascun componente da installare.
- Impostare le proprietà in un file di configurazione  
Occorre modificare i valori predefiniti impostati per le proprietà nelle maschere dei connettori. Occorre inoltre designare le definizioni degli oggetti di business supportati e, con InterChange Server Express, le mappe da utilizzare con le collaborazioni così come specificare la messaggistica, la registrazione e la traccia ed i parametri del gestore dati, se necessario.

Il modo in cui viene eseguito Connector Configurator Express e il tipo di file di configurazione utilizzato, possono essere diversi in base al broker di integrazione eseguito.

Le proprietà della configurazione dei connettori comprendono le proprietà di configurazione standard (le proprietà possedute da tutti connettori) e le proprietà specifiche dei connettori (proprietà necessarie al connettore per una specifica applicazione o tecnologia).

Poiché le **proprietà standard** sono utilizzate da tutti i connettori, non occorre definire tali proprietà dal nulla, Connector Configurator Express le incorpora nel file di configurazione appena si crea il file. Tuttavia, occorre impostare il valore di ciascuna proprietà standard in Connector Configurator Express.

La gamma di proprietà standard potrebbe non essere la stessa per tutti i broker e tutte le configurazioni. Alcune proprietà sono disponibili solo se viene assegnato un valore specifico ad altre proprietà. La finestra delle proprietà standard in Connector Configurator Express mostrerà le proprietà disponibili per una particolare configurazione.

Per le **proprietà specifiche del connettore** , tuttavia, occorre prima definire le proprietà e impostarne i valori. Ciò è possibile creando una maschera per le proprietà scpecifiche per i connettori per un adattatore particolare. Potrebbe esistere già una maschera impostata nel sistema; in tal caso, utilizzare semplicemente questa. In caso contrario, eseguire i passi in “Creazione di una nuova maschera” a pagina 74 per impostarne una nuova.

**Nota:** Connector Configurator Express può essere eseguito solo in un ambiente Windows. Se si sta eseguendo il connettore in un ambiente Linux, u utilizzare Connector Configurator Express in Windows per modificare il file di configurazione e quindi copiarlo nell’ambiente e Linux.

---

## Avvio di Connector Configurator Express

È possibile avviare ed eseguire Connector Configurator Express in entrambe i due modi:

- Indipendentemente, in modo autonomo
- Da System Manager

## Esecuzione di Configurator Express in modalità autonoma

È possibile eseguire Connector Configurator Express indipendentemente e lavorare con i file di configurazione dei connettori, senza considerare il broker.

A tale scopo:

- Fare clic su **Start> Programmi**, quindi su **IBM WebSphere Business Integration Express> Toolset Express> Sviluppo> Connector Configurator Express**.
- Selezionare **File>Nuovo>Configurazione del connettore**.
- Quando si fa clic sul menu di riepilogo a discesa accanto a **System Connectivity Integration Broker**, selezionare **InterChange Server Express**.

È possibile effettuare la scelta di eseguire Connector Configurator Express indipendentemente per generare il file file e quindi collegare System Manager per salvarla in progetto di System Manager (consultare “Completare un file di configurazione” a pagina 78.)

---

## Esecuzione di Connector Configurator Express da System Manager

È possibile eseguire Connector Configurator Express da System Manager.

Per eseguire Connector Configurator Express, effettuare le seguenti operazioni:

1. Aprire System Manager.
2. Nella finestra System Manager, espandere l'icona delle **ICL (Integration Component Libraries)** ed evidenziare **Connettori**.
3. Dalla barra di menu di System Manager, fare clic su **Tools>Connector Configurator**. Viene visualizzata la finestra Connector Configurator Express contenente una casella di dialogo **Nuovo connettore**.
4. Quando si fa clic sul menu a discesa accanto a **System Connectivity Integration Broker**, selezionare **InterChange Server Express**.

Per modificare un file di configurazione esistente:

- Nella finestra System Manager, selezionare uno dei file di configurazione elencati nella cartella e fare clic con il pulsante destro su di esso. Viene aperto Connector Configurator e viene visualizzato il file di configurazione con il tipo broker di integrazione e il nome file all'inizio.
- Da Connector Configurator Express, selezionare **File>Apri**. Selezionare il nome del file di configurazione del connettore da un progetto oppure dalla directory in cui è memorizzato.
- Fare clic sulla scheda **Proprietà standard** per verificare quali proprietà sono comprese nel file di configurazione.

---

## Creazione di una maschera delle proprietà specifiche per un connettore

Per creare un file di configurazione per un connettore, occorre una maschera delle proprietà specifiche per un connettore così come le proprietà standard fornite dal sistema.

È possibile creare una maschera nuova per le proprietà specifiche per il connettore oppure utilizzare una definizione di un connettore esistente come schema.

- Per creare una nuova maschera, consultare “Creazione di una nuova maschera” a pagina 74.
- Per utilizzare un file esistente, modificare semplicemente una maschera esistente e salvarla con un nuovo nome. È possibile trovare maschere già esistenti nella directory `\ProductDir\bin\Data\App`.

## Creazione di una nuova maschera

In questa sezione viene descritto come creare le proprietà nella maschera, definire le caratteristiche e i valori generali per queste proprietà e specificare qualsiasi dipendenza tra le proprietà. Quindi salvare la maschera ed utilizzarla come base per la creazione di un nuovo file di configurazione del connettore.

Per creare una maschera in Connector Configurator Express:

1. Fare clic su **File>Nuovo>Maschera proprietà specifiche per connettore**.
2. Viene visualizzata la casella di dialogo **Maschera proprietà specifiche per connettore**.
  - Immettere un nome per una nuova maschera nel campo **Nome** al di sotto di **Immetti un nuovo nome maschera**. Questo nome verrà visualizzato quando si apre la finestra di dialogo per la creazione di un nuovo file di configurazione da una maschera.
  - Per visualizzare le definizioni delle proprietà specifiche per il connettore in qualsiasi maschera, selezionare questo nome della maschera nella finestra **Nome maschera**. Un elenco delle definizioni delle proprietà contenute in questa maschera viene visualizzato nella finestra **Anteprima schema**.
3. È possibile utilizzare una maschera esistente le cui definizioni delle proprietà sono simili a quelle richieste dal connettore come un punto di partenza per la maschera. Se non viene visualizzata nessuna maschera che visualizzi le proprietà specifiche per il connettore utilizzato, occorre crearne uno.
  - Se si sta pianificando di modificare una maschera esistente, selezionare il nome della maschera dall'elenco nella tabella **Nome maschera** al di sotto di **Seleziona la maschera esistente da modificare: Trova maschera**
  - Questa tabella visualizza i nomi di tutte le maschere attualmente disponibili. È inoltre possibile effettuare la ricerca di una maschera.

### Specifiche delle caratteristiche generali

Quando si da clic su **Avanti** per selezionare una maschera, viene visualizzata la finestra **Proprietà - Maschera proprietà specifiche connettore**. La finestra di dialogo contiene schede per le caratteristiche di tipo Generale delle proprietà definite e per le restrizioni di tipo Valore. La finestra Generale contiene i seguenti campi:

- **Generale:**
  - Tipo proprietà
  - Metodo aggiornato
  - Descrizione
- **Flag**
  - Flag standard
- **Flag personalizzato**
  - Flag

Dopo aver effettuato le selezioni per le caratteristiche generali della proprietà, fare clic sulla scheda **Valore**.

### Specifiche di valori

La scheda **Valore** consente di impostare la lunghezza massima, più valori massimi, un valore predefinito oppure una gamma di valori per la proprietà. Consente inoltre valori editabili. A tale scopo:

1. Fare clic sulla scheda **Valore**. Il pannello di visualizzazione per Valore sostituisce quello per Generale.
2. Selezionare il nome della proprietà nella finestra **Modifica proprietà**.

3. Nei campi per **Lunghezza massima** e **Valori multipli massimi**, specificare i valori desiderati.

Per creare un nuovo valore per le proprietà:

1. Selezionare la proprietà nell'elenco **Modifica proprietà** e fare clic con il tasto destro del mouse su questa voce.
2. Dalla finestra di dialogo, selezionare **Aggiungi**.
3. Immettere il nome del nuovo valore della proprietà e fare clic su OK. Viene visualizzato il valore nella finestra **Valore** posta sulla destra.

La finestra **Valore** visualizza una tabella con tre colonne:

La colonna **Valore** mostra il valore immesso nella casella di dialogo **Valore proprietà** e qualsiasi altro valore creato in precedenza.

La colonna **Valore predefinito** consente di designare qualsiasi valore come predefinito.

**Gamma valori** mostra la gamma immessa nella casella di dialogo **Valore proprietà**.

Una volta creato un valore e visualizzato nella griglia, può essere modificato dall'interno della finestra della tabella.

Per effettuare una modifica in un valore esistente contenuto nella tabella, selezionare una intera riga facendo clic sul numero della riga. Fare doppio clic nel campo **Valore** e fare clic su **Modifica valore**.

## Impostazione delle dipendenze

Una volta effettuate le modifiche nelle schede **Generale** e **Valore**, fare clic su **Avanti**. Viene visualizzata la casella di dialogo **Maschera proprietà specifiche per connettore**.

Una proprietà dipendente è una proprietà compresa nella maschera e utilizzata nel file di configurazione *solo se* il valore di un'altra proprietà soddisfa una specifica condizione. Ad esempio, `PollQuantity` viene visualizzato in una maschera solo se `JMS` è il meccanismo di trasporto e `DuplicateEventElimination` è impostato su `True`.

Per designare una proprietà come dipendente e per impostare la configurazione dalla quale essa dipende, effettuare le seguenti operazioni:

1. Nella finestra **Proprietà disponibili**, selezionare la proprietà che verrà resa dipendente.
2. Nel capo **Seleziona proprietà**, utilizzare il menu a discesa per selezionare la proprietà che manterrà il valore condizionale.
3. Nel campo **Operatore condizione**, selezionare uno dei seguenti valori:
  - == (uguale a)
  - != (non uguale a)
  - > (maggiore di)
  - < (minore di)
  - >= (maggiore di o uguale a)
  - <=(minore di o uguale a)
4. Nel campo **Valore condizionale**, immettere il valore richiesto affinché la proprietà dipendente venga inclusa nella maschera.

5. Con la proprietà dipendente evidenziata nel pannello **Proprietà disponibili**, fare clic su una freccia e spostarla sul pannello **Proprietà dipendenti**.
6. Fare clic su **Fine**. Connector Configurator Express memorizza le informazioni immesse come documento XML, sotto \data\app nella directory \bin dove è installato Connector Configurator Express.

---

## Creazione di un nuovo file di configurazione

Quando si crea un nuovo file di configurazione, occorre assegnargli un nome e selezionare un broker integrativo.

- Nella finestra Gestione sistema, fare clic con il tasto destro del mouse sulla cartella **Connettori**, quindi selezionare **Crea nuovo connettore**. Connector Configurator Express apre e visualizza la finestra di dialogo **Nuovo connettore**.
- In modalità autonoma: da Connector Configurator Express, selezionare **File >Nuovo >Configurazione connettore**. Nella finestra Nuovo connettore, immettere il nome del nuovo connettore

Inoltre è possibile selezionare un broker di integrazione. Il broker selezionato determina le proprietà che verranno visualizzate nella configurazione. Per selezionare un broker:

- Nel campo **Broker di integrazione**, selezionare InterChange Server Express.
- Scorrere i campi rimanenti nella finestra **Nuovo connettore** come descritto successivamente in questo capitolo.

## Creazione un file di configurazione da una maschera specifiche per un connettore.

Una volta creata la maschera specifica per un connettore, è possibile utilizzarlo per creare un file di configurazione.

1. Fare clic su **File>Nuovo>Configurazione del connettore**.
2. Viene visualizzata la casella di dialogo **Nuovo connettore** con i seguenti campi:

- **Nome**

Immettere il nome del connettore. I nomi sono sensibili al maiuscolo/minuscolo. Il nome immesso deve essere univoco e deve essere coerente con il nome del file per un connettore che è installato sul sistema.

**Importante:** Connector Configurator Express non effettua un controllo ortografico del nome immesso. Occorre assicurarsi che il nome sia corretto.

- **Connettività sistema**

Fare clic su InterChange Server Express.

- **Selezionare la maschera proprietà specifiche per connettore**

Immettere il nome della maschera designata per il connettore. Le maschere disponibili vengono mostrate nella finestra **Nome maschera**. Quando si seleziona un nome nella finestra Nome maschera, la finestra **Anteprima maschera proprietà** visualizza le proprietà specifiche per il connettore definite in questa maschera.

Selezionare la maschera che si desidera utilizzare e fare clic su **OK**.

3. Viene visualizzata una finestra di configurazione per il connettore che si sta configurando. La barra del titolo visualizza il broker di integrazione e il nome del connettore. È possibile compilare tutti i valori dei campi per scorrere ora le definizioni oppure salvare il file e completare i campi in seguito.



4. Per salvare il file, fare clic su **File>Salva>nel file** oppure **File>Salva>Nel progetto**. Per salvare un progetto, occorre che System Manager sia in esecuzione.

Se si salva come file, viene visualizzata la casella di dialogo **Salva connettore file**. Selezionare come tipo di file \*.cfg , verificare nel campo Nome file che il nome non sia specificato erroneamente e che abbia il carattere maiuscolo/minuscolo corretto, navigare fino alla navigare directory in cui si desidera posizionare il file e fare clic su **Salva**. La finestra dello stato nel pannello dei messaggi di Connector Configurator Express indica che il file di configurazione è stato creato con esito positivo.

**Importante:** Il percorso e il nome della directory stabiliti qui devono corrispondere con quelli del file di configurazione forniti nel file di avvio del connettore.

5. Per completare le definizioni del connettore, immettere i valori nei campi per ciascuna scheda della finestra Connector Configurator Express, come descritto successivamente in questo capitolo.

---

## Utilizzo di un file esistente

È possibile che vi sia un file esistente disponibile in uno o più dei seguenti formati:

- Un file di definizione connettore.  
Questo è un file di testo che elenca proprietà e valori predefiniti applicabili per un connettore specifico. Alcuni connettori comprendono questo genere di file in una directory \repository nel pacchetto di distribuzione (il file ha in genere l'estensione .txt; ad esempio, CN\_XML.txt per il connettore XML).
- Un file repository ICS.  
Le definizioni utilizzate in una implementazione precedente di InterChange Server Express del connettore potrebbero essere disponibili in un file repository utilizzati nella configurazione di tale connettore. Tale tipo di file in genere possiede un'estensione .in oppure out.
- Un file di configurazione precedente per il connettore.  
Tale tipo di file in genere possiede un'estensione\*.cfg

Sebbene tutte le origini di tali file potrebbero contenere la maggior parte o tutte le proprietà specifiche del connettore , il file di configurazione del connettore non sarà completo finché non è stato aperto il file e non sono state impostate le proprietà, come descritto successivamente in questo capitolo.

Per utilizzare un file esistente per configurare un connettore, occorre aprire il file in Connector Configurator Express, rivedere la configurazione e quindi salvare di nuovo il file.

Effettuare i seguenti passi per aprire un file \*.txt, \*.cfg o \*.in file da una directory:

1. In Connector Configurator Express, fare clic su **File>Apri>Da file**.
2. In casella di dialogo **Apri connettore di file**, selezionare uno dei seguenti tipi di file per visualizzare quelli disponibili:
  - Configurazione (\*.cfg)
  - Repository ICS (\*.in, \*.out)

Selezionare questa opzione se un file repository è stato utilizzato per configurare il connettore in un ambiente InterChange Server Express. È possibile che un file repository includa più definizioni di connettori, che verranno tutte visualizzate quando si apre il file.

- Tutti i file (\*.\*)

Selezionare questa opzione se un file \*.txt è stato fornito nel pacchetto dell'adattatore per il connettore oppure se un file di definizione è disponibile sotto un'altra estensione.

3. Nel pannello della directory, spostarsi fino al file di definizione del connettore appropriato e fare clic su **Apri**.

Effettuare i seguenti passi per aprire una configurazione di connettore da un progetto di System Manager:

1. Avviare System Manager. È possibile aprire una configurazione da System Manager oppure salvarla in esso solo se System Manager è stato avviato.
2. Avviare Connector Configurator Express.
3. Fare clic su **File>Apri>Da progetto**.

---

## Completare un file di configurazione

Quando si apre un file di configurazione o un connettore da un progetto, la finestra Connector Configurator Express visualizza il pannello di configurazione con gli attuali attributi e valori.

Il titolo del pannello di configurazione visualizza il broker di integrazione e il nome connettore come specificati nel file. Accertarsi di avere il broker corretto. In caso contrario, modificare il valore del broker prima di configurare il connettore. A tale scopo:

1. Sotto la scheda **Proprietà standard**, selezionare il campo del valore per la proprietà BrokerType. Nel menu a discesa selezionare il valore ICS.
2. La scheda Proprietà standard visualizzerà le proprietà associate al broker selezionato. È possibile salvare il file adesso oppure completare i restanti campi di configurazione, come descritto in "Specifica di definizioni di oggetti di business supportati" a pagina 81..
3. Una volta terminata la configurazione, fare clic su **File>Salva>nel progetto** oppure **File>Salva>nel file**.

Se si sta salvando un file, selezionare \*.cfg come estensione, selezionare la posizione corrente per il file e fare clic su **Salva**.

Se più configurazioni di connettori sono aperte, fare clic su **Salva tutto in file** per salvare tutte le configurazioni nel file oppure fare clic su **Salva tutto in progetto** per salvare tutte le configurazioni di connettori in un progetto di System Manager.

Prima di salvare il file, Connector Configurator Express verifica che i valori siano stati impostati per tutte le proprietà standard richieste. Se ad una proprietà standard richiesta manca un valore, Connector Configurator Express visualizza un messaggio di errore di mancata validazione. Occorre fornire un valore per la proprietà per salvare il file di configurazione.

---

## Impostazione delle proprietà del file di configurazione

Quando si crea e si assegna un nome ad un nuovo file di configurazione oppure quando si apre un file di configurazione di connettore esistente, Connector Configurator Express visualizza un pannello di configurazione con schede per le categorie di valori di configurazione richiesti.

Connector Configurator Express richiede valori per le proprietà di tali categorie per i connettori che eseguono tutti i broker:

- Proprietà standard
- Proprietà specifiche dei connettori
- Oggetti di business supportati
- Valori di traccia/Log del file
- Data Handler (applicabile per i connettori che utilizzano messaggistica JMS con consegna di eventi garantita)

**Nota:** Per connettori che utilizzano messaggistica JMS, è possibile che venga visualizzata una categoria aggiuntiva, per configurazioni di data handler che convertono dati in oggetti di business.

Per connettori che vengono eseguiti su **InterChange Server Express**, sono anche richiesti i valori per queste proprietà.

- Mappe associate
- Risorse
- Messaggistica (dove applicabile)

**Importante:** Connector Configurator Express accetta valori di proprietà in serie di caratteri in inglese o lingue diverse dall'inglese. Tuttavia, i nomi delle proprietà standard e specifiche dei connettori e i nomi di oggetti di business supportati, devono utilizzare solo serie di caratteri inglesi.

Le proprietà standard differiscono dalle proprietà specifiche dei connettori per i seguenti elementi:

- Le proprietà standard di un connettore vengono condivise dal componente specifico di un'applicazione di un connettore e dal componente del broker. Tutti i connettori hanno lo stesso insieme di proprietà standard. Tali proprietà vengono descritte nell'Appendice A per ciascuna guida per l'adattatore. È possibile modificare alcuni di questi valori ma non tutti.
- Le proprietà specifiche dell'applicazione si applicano solo al componente specifico di un'applicazione di un connettore, cioè, il componente che interfaccia direttamente con l'applicazione. Ciascun componente possiede le proprietà specifiche dell'applicazione che sono univoche per la relativa applicazione. Alcune di queste proprietà forniscono valori predefiniti e alcune non lo fanno; è possibile modificare alcuni valori predefiniti. I capitoli relativi all'installazione e alla configurazione di ciascuna guida descrivono le proprietà specifiche dell'applicazione e i valori raccomandati.

I campi per **Proprietà standard** e **Proprietà specifiche del connettore** sono codificate con un colore per indicare quali sono configurabili:

- Un campo con uno sfondo grigio indica che si tratta di una proprietà standard. È possibile modificare il valore ma non modificarne il nome o eliminare la proprietà.
- Un campo con uno sfondo bianco indica che si tratta di una proprietà specifica dell'applicazione. Tali proprietà variano a seconda delle necessità specifiche dell'applicazione o del connettore. È possibile modificare il valore ed eliminare tali proprietà.
- I campi del colore sono configurabili.
- Il campo **Metodo aggiornato** viene visualizzato per ciascuna proprietà. Con questo campo viene indicato se è necessario riavviare un componente o un agente per attivare i valori modificati. Non è possibile configurare tale impostazione.

## Impostazione delle proprietà standard del connettore

Per modificare il valore di una proprietà standard:

1. Fare clic sul campo di cui si desidera visualizzare il valore.
2. Immettere un valore oppure selezionarne uno dal menu a discesa, se viene visualizzato.
3. Una volta immessi i valori per le proprietà standard, è possibile effettuare una delle seguenti operazioni:
  - Per eliminare le modifiche, conservare i valori originali e uscire da Connector Configurator Express, fare clic su **File>Uscita** (oppure chiudere la finestra) e fare clic su **No** quando viene richiesto di salvare le modifiche.
  - Per immettere i valori per altre categorie in in Connector Configurator Express, selezionare la scheda per la categoria. I valori immessi per **Proprietà standard** (oppure per qualsiasi altra categoria) vengono conservati quando si passa alla categoria successiva. Quando si chiude la finestra, viene richiesto di salvare o eliminare i valori immessi in tutte le categorie nel loro insieme.
  - Per i valori revisionati, fare clic su **File>Uscita** (oppure chiudere la finestra) e fare clic su **Sì** quando richiesto di salvare le modifiche. In alternativa, fare clic su **Salva>in file** dal menu File o dalla barra degli strumenti.

## Impostazione delle proprietà di configurazione specifiche dell'applicazione

Per le proprietà di configurazione specifiche dell'applicazione, è possibile aggiungere o modificare i nomi delle proprietà, configurare i valori, eliminare una proprietà e crittografare una proprietà. La lunghezza della proprietà predefinita è di 255 caratteri.

1. Fare clic con il pulsante destro del mouse in parte superiore sinistra della griglia. Viene visualizzata una barra dei menu a discesa. Fare clic su **Aggiungi** per aggiungere una proprietà. Per aggiungere una proprietà secondaria, fare clic con il pulsante destro del mouse sul numero di righe principali e fare clic su **Aggiungi secondario**.
2. Immettere un valore per la proprietà o la proprietà secondaria.
3. Per crittografare una proprietà, selezionare la casella **Crittografa**.
4. Scegliere se salvare o eliminare le modifiche, come descritto per "Impostazione delle proprietà standard del connettore".

Il valore Metodo aggiornato visualizzato per ciascuna proprietà indica se è necessario riavviare un componente o un agente per attivare i valori modificati.

**Importante:** Modificando il nome di una proprietà di connettore di applicazioni specifiche potrebbe causare la non riuscita del connettore. Alcuni nomi di proprietà potrebbero essere necessari al connettore per connettersi ad un'applicazione o per essere eseguito correttamente.

### Crittografia delle proprietà del connettore

Le proprietà specifiche dell'applicazione possono essere crittografate selezionando la casella di controllo **Crittografa** nella finestra Proprietà specifiche del connettore. Per decrittografare un valore, fare clic per eliminare i valori dalla casella di controllo **Crittografa** ed immettere il valore corretto nella casella di dialogo **Verifica** e fare clic su **OK**. Se il valore immesso è corretto, il valore viene decrittato e viene visualizzato.

Nella guida per l'utente dell'adattatore per ciascun connettore è presente un elenco e la descrizione di ciascuna proprietà del relativo valore predefinito.

Se per una proprietà esistono più valori, verrà visualizzata la casella di controllo **Crittografa** per il primo valore della proprietà. Quando si seleziona **Crittografa**, tutti i valori verranno crittografati. Per decrittografare più valori di una proprietà, fare clic per fare clic per eliminare i valori dalla casella di controllo **Codifica** e quindi immettere un nuovo valore nella casella di controllo **Verifica**. Se il valore immesso corrisponde, tutti i valori multipli verranno decriptati.

### **Metodo aggiornato**

Fare riferimento alle descrizioni dei metodi di aggiornamento individuati nell'appendice *Proprietà di configurazione standard per i connettori*, sotto "Panoramica dei valori di proprietà di configurazione" a pagina 104.

## **Specifica di definizioni di oggetti di business supportati**

Utilizzare la scheda **Oggetti di business supportati** in Connector Configurator Express per specificare gli oggetti di business che il connettore utilizzare. Occorre specificare gli oggetti di business generici e quelli specifici delle applicazioni ed occorre specificare le associazioni per le mappe tra oggetti di business.

**Nota:** Per alcuni connettori è necessario che determinati oggetti di business siano specificati come supportati, per eseguire notifiche di eventi o configurazioni aggiuntive (utilizzando meta oggetti) con le relative applicazioni.

### **InterChange Server Express come broker di integrazione**

Per specificare che una definizione di oggetto di business sia supportata dal connettore oppure modificare le impostazioni di supporto per una definizione di oggetto di business esistente, fare clic sulla scheda **Oggetti di business supportati** ed utilizzare i seguenti campi.

**Nome oggetto di business:** Per designare che una definizione di un oggetto di business sia supportata dal connettore, con System Manager in esecuzione, effettuare le seguenti operazioni:

1. Fare clic in un campo vuoto nell'elenco **Nome oggetto di business**. Viene visualizzato un elenco a discesa contenente tutte le definizioni di oggetti di business che esistono nel progetto di System Manager.
2. Fare clic su un oggetto di business per aggiungerlo.
3. Impostare **Supporto agente** (descritto qui di seguito) per l'oggetto di business.
4. Nel menu File della finestra Connector Configurator Express, fare clic su **Salva in progetto**. La definizione del connettore modificata, compreso il supporto designato per la definizione dell'oggetto di business aggiunto, viene salvata in un progetto ICL (Integration Component Library) in System Manager.

Per eliminare un oggetto di business dall'elenco supportato:

1. Per selezionare un campo di oggetto di business, fare clic sul numero a sinistra dell'oggetto di business.
2. Dal menu **Modifica** la finestra Connector Configurator Express, fare clic su **Elimina riga**. L'oggetto di business viene eliminato dalla visualizzazione dell'elenco.
3. Dal menu **File**, fare clic su **Salva in progetto**.

Eliminando l'oggetto di business dall'elenco supportato viene modificata la definizione di connettore e rende l'oggetto di business eliminato non disponibile

all'utilizzo in questa implementazione del connettore. Il codice di connessione né viene influenzato né elimina la definizione dell'oggetto di business da System Manager.

**Supporto agente:** Se un oggetto di business ha un Supporto agente, il sistema cercherà utilizzare l'oggetto di business per la distribuzione di dati per un'applicazione tramite un agente connettore.

Tipicamente, vengono supportati oggetti di business per un connettore dall'agente del connettore, ma non gli oggetti di business generico.

Per indicare che l'oggetto di business viene supportato dall'agente del connettore, verificare la casella **Supporto agente**. La finestra Connector Configurator Express non convalida le selezioni Supporto agente.

**Livello di transazione massimo:** Il livello della transazione massima per un connettore è il livello di transazione più elevata che supporta connettori.

Per ulteriori connettori, Massimo sforzo è l'unica scelta possibile.

Occorre il server per le modifiche nel livello delle transazioni perché abbiano effetto.

## Mappe associate per InterChange Server Express

Ciascun connettore supporta un elenco di definizioni di oggetti di business e le relative mappe associate che sono cattive attualmente in InterChange Server Express. Questo elenco viene visualizzato quando si seleziona la scheda **Mappe associate**.

L'elenco degli oggetti di business contiene l'oggetto di business specifico dell'applicazione supportato dall'agente e l'oggetto generico corrispondente che il controllore invia alla collaborazione di sottoscrizione. L'associazione di una mappa determina quale mappa verrà utilizzata per trasformare l'oggetto di business specifico dell'applicazione per l'oggetto di business specifico per l'oggetto di business specifico dell'applicazione.

Se si stanno utilizzando mappe definite in modo univoco per oggetti di business di origine e destinazione specifici, le mappe verranno già associate ai relativi oggetti di business appropriati quando si apre la finestra e non occorrerà modificarli.

Se più di una mappa è disponibile per essere utilizzata da un oggetto di business supportato, occorrerà associare l'oggetto di business con la mappa da utilizzare.

La scheda **Mappe associate** visualizza i seguenti campi:

- **Nome oggetto di business**

Questi sono gli oggetti di business supportati da questo connettore, come designato nella scheda **Oggetti di business supportato**. Se si designano oggetti di business aggiuntivi nella scheda Oggetti di business supportati, ciò sia riflette in questo elenco dopo aver salvato le modifiche selezionando **Salva in progetto** dal menu **File** della finestra Connector Configurator Express.

- **Mappe associate**

La finestra visualizza tutte le mappe installate nel sistema da utilizzare con gli oggetti di business supportati del connettore. L'oggetto di business origine per ciascuna mappa viene visualizzato a sinistra del nome mappa, nella finestra **Nome oggetto di business**.

- **Esplicito**

In alcuni casi, occorre eseguire il bind in modo esplicito con una mappa associata.

Il bind esplicito è necessario solo quando esistono più mappe per un oggetto di business supportato particolare. Quando InterChange Server Express viene avviato, cerca di eseguire il bind automaticamente con una mappa per ciascun oggetto di business supportato per ciascun connettore. Se più mappe prendono come input lo stesso oggetto di business, il server cerca di individuare ed eseguire il bind con la mappa che è il soprainsieme delle altre.

Se non esiste una mappa che è il soprainsieme delle altre, il server non dovrà eseguire il bind con l'oggetto di business ad una singola mappa e occorrerà impostare il bind in modo esplicito.

Per eseguire il bind in modo esplicito, effettuare le seguenti operazioni:

1. Nella colonna **Esplicita**, selezionare la casella di controllo della mappa di cui si desidera eseguire il bind:
2. Selezionare la mappa che si desidera associare all'oggetto di business.
3. Nel menu **File** della finestra Connector Configurator Express, fare clic su **Salva in progetto**.
4. Distribuire il progetto di InterChange Server Express.
5. Riavviare il server per rendere definitive le modifiche.

## Impostazione dei valori dei file di traccia/log

Quando si apre un file di configurazione del connettore o un file di definizione del connettore, Connector Configurator Express utilizza i valori di log e di traccia di questo file come valori predefiniti. È possibile modificare tali valori in Connector Configurator Express.

Per modificare i valori di traccia e di log:

1. Fare clic sulla scheda **File di traccia/log**.
2. Per la traccia o il log, è possibile scegliere di scrivere messaggi in:
  - Console (STDOUT):  
Visualizza i messaggi di log o di traccia nel pannello STDOUT.

**Nota:** È possibile utilizzare l'opzione STDOUT dalla scheda **File di traccia/log** per connettori che sono in esecuzione su piattaforma Windows.

- File:  
I messaggi di log o di traccia vengono scritti nel file specificato. Per specificare il file, fare clic sul pulsante della directory (ellipsis), spostarsi fino alla posizione desiderata, fornire un nome file e fare clic su **Salva**. I messaggi di log e di traccia sono scritti nel file e la posizione specificata.

**Nota:** I file di log e di traccia sono semplici file di testo. È possibile utilizzare l'estensione file preferita quando se ne imposta il nome file. Per file di traccia tuttavia è consigliabile utilizzare l'estensione `.trace` piuttosto che `.trc`, per evitare confusione tra altri file che possono risiedere sul sistema. Per file di log, `.log` e `.txt` sono tipiche estensioni file.

## Gestori dati

La sezione relativa ai data handler è disponibile per la configurazione solo se si è designato un valore JMS for DeliveryTransport e un valore JMS per ContainerManagedEvents. Non tutti gli adattatori utilizzano data handler.

patrimonioconsultare le descrizioni in ContainerManagedEvents nell'Appendice A, Proprietà standard per valori da utilizzare per tali proprietà.

---

## Salvataggio del file di configurazione

Quando terminata la configurazione del connettore, salvare il file di configurazione del connettore. Connector Configurator Express salva il file nel modo broker selezionato durante la configurazione. The title bar of Connector Configurator Express visualizza sempre il modo broker.

Il file viene salvato come un documento XML. È possibile salvare il documento XML in tre modi:

- Da System Manager, come un file con un'estensione \*.con in una ICL (Integration Component Library) oppure
- In una directory specificata.
- In modo autonomo, come un file con un'estensione \*.cfg in una cartella della directory. Per impostazione predefinita, il file è salvato al percorso \ProductDir\bin\Data\App per tutte le piattaforme.

Per i dettagli sull'utilizzo dei progetti in Gestione sistema e per ulteriori informazioni sullo sviluppo, consultare il manuale *Guida all'installazione di WebSphere Business Integration Server Express per Windows, per Linux, o per i5/OS*.

---

## Modifica di un file di configurazione

È possibile modificare le impostazioni del broker di integrazione per un file di configurazione esistente. Ciò consente di utilizzare il file come maschera per la creazione di un nuovo file di configurazione, che può essere utilizzato con un broker differente.

**Nota:** Occorre modificare altre proprietà di configurazione così come la proprietà del modo broker se si passa ai broker di integrazione. <

Per modificare la selezione del broker all'interno di un file di configurazione esistente (facoltativo):

- Aprire il file di configurazione esistente in Connector Configurator Express .
- Selezionare la scheda **Proprietà standard**.
- Nel campo **BrokerType** della scheda Proprietà s standard, selezionare il valore appropriato per il broker.  
Quando si modifica il valore corrente, le schede disponibili e le selezioni del campo del pannello delle proprietà cambierà immediatamente, per mostrare solo quelle schede e campi che riguardano il nuovo broker selezionato.

---

## Completamento della configurazione

Una volta creato un file di configurazione un connettore e modificato, accertarsi che il connettore possa individuare il file di configurazione quando il connettore viene avviato.

A tale scopo, aprire il file di startup utilizzato per il connettore e verificare che la posizione e il nome file utilizzato per il file di configurazione del connettore corrisponda esattamente al nome fornito al file e la directory oppure il percorso dove viene posta.



---

## Utilizzando Connector Configurator Express in un ambiente globalizzato.

Connector Configurator Express viene globalizzato e può gestire conversioni di caratteri tra il file di configurazione e il broker di integrazione. Connector Configurator Express utilizza la codifica originale. Quando scrive nel file di configurazione, utilizza la codifica UTF-8.

Connector Configurator Express supporta caratteri diversi dall'inglese in:

- Tutti i campi di valore:
- Il file di log e il percorso del file di traccia (specificato nella scheda **File di traccia/log**)

L'elenco a discesa per le proprietà di configurazione standard CharacterEncoding e Locale visualizza solo un sottoinsieme di valori supportati. Per aggiungere valori all'elenco a discesa, occorre modificare manualmente il file `\Data\Std\stdConnProps.xml` nella directory del prodotto.

Ad esempio, per aggiungere la locale `en_GB` all'elenco di valori per la proprietà `Locale`, aprire il file `stdConnProps.xml` e aggiungere la seguente riga in grassetto:

```
<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>
```



---

## Capitolo 6. Configurazione dell'applicazione

Questo capitolo descrive il modo in cui abilitare l'applicazione per eseguire il polling degli eventi. Contiene la sezione > "Abilitazione dell'applicazione per il connettore."

---

### Abilitazione dell'applicazione per il connettore

E' necessario configurare il meccanismo di notifica degli eventi nel database prima che il connettore elabori il recapito degli eventi. Per procedere in tal modo, è necessario completare le seguenti attività:

- Creare l'evento ed archiviare le tabelle nel database.
- Installare i trigger del database nelle tabelle dell'applicazione per supportare gli oggetti aziendali richiesti. Si suppone lo sviluppo dei propri trigger del database.
- Facoltativamente, installare una tabella di conteggio. Eseguire questo passo solo se si richiede che il connettore generi un ID univoco durante la creazione di un oggetto business. Per ulteriori informazioni sulla generazione di ID univoci, visualizzare il parametro `UID=CW.uidcolumnname[=UseIfMissing]`.

Le sezioni che seguono forniscono informazioni sulla creazione e configurazione delle tabelle di archivio e di eventi.

### Tabelle di archivio e di eventi

Il connettore utilizza la tabella di evento per mettere in coda gli eventi da raccogliere. Se si imposta la proprietà `ArchiveProcessed` su `true` o su nessun valore, il connettore utilizza la tabella di archivio per memorizzare gli eventi dopo aver aggiornato lo stato nella tabella degli eventi.

Per ciascun evento, il connettore ottiene il nome, l'espressione, e la chiave dell'oggetto business dalla tabella degli eventi. Il connettore utilizza queste informazioni per richiamare l'entità completa dall'applicazione. Se l'entità è stata modificata dopo la registrazione dell'evento, il connettore ottiene l'evento iniziale e tutte le successive modifiche. In altre parole, se un'entità viene creata e aggiornata prima di essere ottenuta dal connettore dalla tabella eventi, il connettore ottiene entrambe le modifiche dei dati in un unico richiamo.

Per ciascun evento elaborato da un connettore, sono possibili i tre risultati di seguito riportati:

- L'evento è stato elaborato correttamente
- L'evento non è stato elaborato correttamente
- L'evento non è stato inoltrato (per le informazioni di inoltro specifiche al broker di integrazione, fare riferimento al manuale *System Implementation Guide*).

Se gli eventi non vengono eliminati dalla tabella eventi una volta raccolti dal connettore, occupano uno spazio non necessario. Tuttavia, se vengono eliminati, tutti gli eventi non elaborati andranno perduti e non è possibile controllare l'elaborazione degli eventi. Pertanto, si consiglia di creare anche una tabella di archivio e impostare le proprietà `ArchiveProcessed` su `true`. Prima che un evento viene eliminato dalla tabella eventi, il connettore lo inserisce nella tabella dell'archivio.

**Nota:** Se i problemi di accesso al database dell'applicazione non consentono al connettore di eliminare un evento dalla tabella eventi o di inserire un evento nella tabella di archivio, il connettore restituisce APPRESPONSETIMEOUT.

## Configurazione di un evento ed elaborazione di un archivio

Per configurare l'elaborazione di un evento e di un archivio, è necessario utilizzare le proprietà di configurazione per specificare le seguenti informazioni:

- Il nome della tabella eventi ("EventTableName" a pagina 134). Non è necessario specificare un valore per questa proprietà se si utilizza il connettore solo per elaborare le richieste dell'oggetto business.
- L'intervallo di frequenza ("PollFrequency" a pagina 121).
- Il numero degli eventi per ciascun intervallo di polling ("PollQuantity" a pagina 135).
- Il nome della tabella di archivio ("ArchiveTableName" a pagina 129).
- Se il connettore archivia eventi non inoltrati e non elaborati ("ArchiveProcessed" a pagina 129). Per informazioni specifiche sul broker d'integrazione, consultare la sezione *Guida all'implementazione del sistema*.
- L'ID univoco del connettore, che è importante quando più connettori effettuano il polling della stessa tabella ("ConnectorID" a pagina 130).

Inoltre, è possibile specificare un valore per EventOrderBy property per indicare l'ordine degli eventi da elaborare. Per informazioni su questa ed altre proprietà di configurazione, consultare Appendice A, "Proprietà di configurazione standard per i connettori", a pagina 103 e Tabella 16 a pagina 127.

**Nota:** La creazione dell'evento e delle tabelle di archivio sono opzionali. Tuttavia, se si specifica un valore per EventTableName, ma non si utilizza il connettore per eseguire il polling degli eventi e non si crea una tabella eventi, il connettore va in timeout. Per impedire il timeout, lasciare il valore di EventTableName come null (come una stringa).

Per impostazione predefinita, il nome della tabella della coda eventi è `xworlds_events` e il nome della tabella della coda dell'archivio è `xworlds_archive_events`.

Per utilizzare il connettore solo per l'elaborazione delle richieste, utilizzare l'opzione `-fno` quando viene avviato e si imposta il valore di EventTableName su `null` (come una stringa).

Se il driver utilizzato non supporta la classe Java DatabaseMetaData e si desidera che il connettore ignori la verifica dell'esistenza delle tabelle di eventi e di archivi, disabilitare `CheckForEventTableOnInit` impostandone il valore su `false`. Come valore predefinito è `true` (vero). Si consiglia di non impostare il valore su `false`.

**Nota:** Se il sito non archivia gli eventi nella tabella archivi, impostare il valore di `ArchiveProcessed` su `false`.

## Script SQL per l'installazione di eventi e tabelle di archivio

Questi script vengono forniti solo come modelli per assistere durante la creazione delle tabelle richieste per il connettore. Per altri database oltre al server IBM DB2 e Microsoft SQL, creare gli script utilizzando questi come linee guida.

Gli script per installare tabelle eventi, archivio, ed identificatore unico per un database DB2 sono:

- event\_table\_db2.sql
- event\_package\_db2.sql
- archive\_table\_db2.sql
- uid\_table\_db2.sql

Gli script per installare tabelle eventi, archivio, ed identificatore unico per un database Microsoft SQL sono:

- event\_table\_mssqlserver.sql
- event\_package\_mssqlserver.sql
- archive\_table\_mssqlserver.sql
- uid\_table\_mssqlserver.sql

Un programma CL denominato QWBIJDBCSQ viene incluso nella libreria QWBISVR44 che può essere utilizzata per eseguire gli script SQL che creano tabelle di archivio e di eventi in i5/OS. I parametri sono raccolte e profili utente. Il profilo utente dovrebbe essere lo stesso che il connettore esegue deve avere le autorizzazioni alla raccolta (libreria). Esempio: CALL WBISVR44/QWBIJDBCSQ PARM (MYDB QWBISVR44). Gli script per installare le tabelle di eventi, di archivio e di identificativo univoci per un database i5/OS DB2 sono:

- EVENTTBL - script sql tabella eventi
- EVENTPKG - script sql pacchetto eventi
- ARCHVTBL - script sql tabella archivio
- UIDTBL - script sql tabella UID

Questi campi si trovano nelle seguenti directory:

**Linux:**

connectors/JDBC/dependencies/

**Windows:**

connectors\JDBC\dependencies\

**i5/OS:**

File QWBISRC nella libreria QWBISVR44

**Nota:** Questi script vengono forniti solo come modelli per assistere durante la creazione delle tabelle richieste per il connettore. Per gli altri database, creare gli script utilizzando questi come linee guida. L'ordine ed il tipo di dati nelle colonne della tabella è molto importante. Per visualizzare l'ordine e il tipo corretto, fare riferimento a "Schema di tabelle di archivio e eventi".

Si consiglia all'amministratore del database o all'utente che implementa il connettore di modificare questi script in modo che corrispondano ai requisiti di ottimizzazione e di query specifici. Ad esempio, questi script non creano indici sulle tabelle. E' responsabilità della persona che implementa il connettore per creare gli indici per ottimizzare le prestazioni con l'utilità di ottimizzazione della query.

## Schemi di tabelle di archivio e di eventi

Tabella 12 a pagina 90 descrive le colonne delle tabelle di eventi e di archivio.

Tabella 12. Schema di tabelle di archivio e di eventi

Nome	Descrizione	Tipo	Vincolo
event_id	Identificativo interno dell'evento	NUMERO	Chiave primaria
connector_id	ID univoco del connettore cui è destinato l'evento. Questo valore è importante quando più connettori eseguono il polling della stessa tabella.	VARCHAR	
object_key	Chiave primaria dell'oggetto business. La chiave può essere rappresentata come coppia name_value o come una serie di chiavi delimitata da due punti o un altro delimitatore configurabile (ad esempio, 1000065:10056:2333). Per ulteriori informazioni, consultare "EventKeyDel" a pagina 133.	VARCHAR	Non null
object_name	Nome dell'oggetto di business	VARCHAR	Non null
object_verb	Espressione associata all'evento	VARCHAR	Non null
event_priority	La priorità dell'evento (0 è la priorità più elevata e 1 è la priorità più bassa), che il connettore utilizza per ottenere gli eventi in base alla priorità. Il connettore non utilizza questo valore per diminuire o aumentare le priorità.	NUMERO	Non null
event_time	Data e ora in cui si è verificato l'evento	DATETIME	Data e ora corrente (per la tabella di archivio, ora evento corrente)
archive_time	Data e ora in cui l'evento è stato archiviato (si applica solo alla tabella di archivio)	DATETIME	Data/ora archivio
event_status	-2 (Errore durante l'invio dell'evento al broker di integrazione) -1 (Errore durante l'elaborazione dell'evento) 0 (Pronto per il polling) 1 (Inviato al broker di integrazione) 2 (Non è presente alcuna sottoscrizione per l'oggetto business) 3 (In corso). Questo stato è utilizzato solo nella tabella eventi e non nella tabella di archivio.	NUMERO	Non null
event_comment	Descrizione dell'evento o della stringa di errore.	VARCHAR	

---

## Capitolo 7. Esecuzione del connettore

Questo capitolo descrive il modo in cui avviare e arrestare il connettore, oltre al modo in cui creare più istanze del connettore. Questo capitolo contiene le sezioni seguenti:

- “Avvio del connettore”
- “Arresto del connettore” a pagina 92
- “Creazione di più istanze del connettore” a pagina 93

---

### Avvio del connettore

Un connettore deve essere avviato utilizzando lo **script di avvio del connettore**. Sui sistemi Windows lo script di avvio deve essere ubicato nella directory runtime del connettore: *dir\_prodotto\connectors\connName* dove *connName* identifica il connettore.

Sui sistemi Linux lo script di avvio deve essere ubicato nella directory *dir\_prodotto/bin*.

Sui sistemi i/OS lo script di avvio deve essere ubicato in */QIBM/UserData/WBIServer44/<instance>/connectors/<ConnInstance/* con cui il connettore viene eseguito.

Il nome dello script di avvio dipende dal sistema operativo come mostrato in Tabella 13.

Tabella 13. Script di avvio del connettore

Sistema operativo	Script di avvio
Sistemi basati su Linux	connector_manager
i5/OS	start_connName.sh
Windows	start_connName.bat

Durante l'esecuzione lo script di avvio cerca il file di configurazione nella *Productdir* (consultare i comandi che seguono). Questa è ubicazione del file di configurazione.

**Nota:** Si necessita di un file di configurazione locale se l'adattatore utilizza un trasporto JMS.

#### Avvio del connettore su un sistema Windows:

- Dal menu **Start**, selezionare **Programmi>IBM WebSphere Business Integration Server Express>Adattatori>Connettori**. Per impostazione predefinita, il nome del programma è “IBM WebSphere Business Integration Server Express”. Può però essere personalizzato. In alternativa è possibile creare un collegamento sul desktop per il proprio connettore.
- Dalla riga comandi di Windows: `start_connName connName brokerName {-configFile}`.
- Sui sistemi Windows, si può configurare il connettore in modo che parta come un servizio Windows. In questo caso, il connettore parte all'avvio del sistema Windows (servizio ad avvio automatico), o quando si decide di farlo partire tramite la finestra Servizi di Windows (servizio ad avvio manuale).

### Avvio del connettore su un sistema basato su Linux:

- Dalla riga comandi immettere:  
`connector_manager -start connName brokerName [-cconfigFile ]`

dove *connName* è il nome del connettore e *brokerName* identifica il proprio broker di integrazione.

- Per InterChange Server Express, specificare per *brokerName* il nome dell'istanza di InterChange Server Express.

### Avvio del connettore su un sistema i5/OS:

- Dal sistema Windows su cui è installata la console di WebSphere Business Integrations Server Express, selezionare **IBM WebSphere Business Integration Server Express>Toolset Express>Amministrazione>Console**. Quindi, specificare il nome del sistema OS/400 o i5/OS oppure l'indirizzo IP e un profilo utente e la password che abbia autorizzazioni speciali \*JOBCTL. Selezionare il connettore dall'elenco di connettori e fare clic su Avvia.
- Per avviare automaticamente l'adattatore utilizzando la Console, utilizzare lo script `submit_adapter.sh`. Questo è l'unico modo in cui l'adattatore verrà avviato utilizzando il sottosistema all'interno della voce autostart job per il server.
- In modalità batch, dalla riga comandi i5/OS, è necessario eseguire il comando `CL QSH` e dall'ambiente `QSHELL`. Eseguire `/QIBM/ProdData/WBIServer44/bin/submit_adapter.sh connName WebSphereICSName pathToConnNameStartScript jobDescriptionName`, dove *connName* è il nome del connettore, *WebSphereICSName* è il nome del server Interchange Server Express (il nome predefinito è `QWIBDFT44`), *pathToConnNameStartScript* è il percorso completo dello script di avvio del connettore e *jobDescriptionName* è il nome della descrizione del processo da utilizzare nella libreria `QWIBSVR44`.
- In modalità interattiva, è necessario eseguire il comando `CL QSH` e dall'ambiente `QSHELL`. Eseguire `/QIBM/UserData/WBIServer44/WebSphereICSName/connectors/connName/start_connName.sh connNameWebSphereICSName [-cConfigFile]` dove *connName* è il nome del connettore e *WebSphereICSName* è il nome dell'istanza InterChange Server Express.

*Per maggiori informazioni su come avviare un connettore, comprese le opzioni di avvio da linea comandi, consultare il manuale Guida all'ammistrazione del sistema..*

---

## Arresto del connettore

Il modo in cui si deve arrestare un connettore dipende da come è stato avviato.

- Windows:
  - È possibile richiamare lo script di avvio apre una finestra di "console" separata per il connettore. In questa finestra, digitare "q" e premere invio per arrestare il connettore.
  - È possibile configurare il connettore in modo che parta come un servizio Windows. In questo caso il connettore si arresta quando il sistema Windows viene spento.
- i5/OS:
  - Se si è avviato il connettore utilizzando la Console, o tramite lo script "submit\_adapter.sh" in `QSHELL`, è possibile utilizzare uno dei due metodi per arrestare il connettore:
  - Dal sistema Windows su cui è installata la console di WebSphere Business Integration Server Express, selezionare **IBM WebSphere Business Integration**



**Server Express>Toolset Express>Amministrazione>Console.** Quindi, specificare il nome del sistema OS/400 o i5/OS oppure l'indirizzo IP e un profilo utente e la password che abbia autorizzazioni speciali \*JOBCTL. Selezionare l'adattatore JDBC dall'elenco e fare clic sul pulsante Arresta. Utilizzare il comando WRKACTJOB SBS (QWIBSVR44) per visualizzare i processi su Server Express Product. Scorrere la lista fino a trovare il job con il nome corrispondente alla descrizione job del connettore. Ad esempio, per il connettore JDBC il nome job sarà QWBIJDBCC. Selezionare l'opzione 4 per questo job, e premere F4 per arrivare alla richiesta del comando ENDJOB. Specificare quindi \*IMMED per il parametro Option e premere invio.

**Nota:** Il connettore verrà terminato una volta arrestato il sistema secondario QWBISVR44.

- Se si è usato lo script start\_connName.sh script per avviare l'adattatore dalla QSHELL, premere F3 per arrestare il connettore. È inoltre possibile arrestare l'agente, utilizzando uno script denominato *stop\_adapter.sh* nella directory /QIBM/ProdData/WBIServer44/bin.

- **Linux:**

I connettori vengono eseguiti in background e quindi non hanno una finestra separata. Eseguire i seguenti comandi per arrestare il connettore:

```
connector_manager -stop connName
```

dove *connName* è il nome del connettore.

---

## Creazione di più istanze del connettore

La creazione di più istanze di un connettore è in molti casi lo stesso della creazione di un connettore personalizzato. È possibile impostare il sistema in modo da creare ed eseguire più istanze di un connettore seguendo le operazioni qui di seguito. Occorre:

- Creare una nuova directory per l'istanza del connettore
- Accertarsi di possedere le definizioni degli oggetti di business richiesti
- Creare un nuovo file di definizione del connettore
- Creare una nuova script di avvio

### Creazione di una nuova directory

- **Per piattaforme Windows:**

ProductDir\connectors\connectorInstance

Se il connettore dispone di metaoggetti specifici del connettore, occorre creare un metaoggetto per

L'istanza del connettore. Se si salva il metaoggetto come un file, creare la directory e salvare il file qui:

ProductDir\Repository\connectorInstance

dove *connectorInstance* identifica in modo univoco l'istanza del connettore.

È possibile specificare il nome del server InterChange Server Express come un parametro di startup.bat; un esempio è: start\_JDBC.bat connName serverName.

- **Per piattaforme i5/OS:**

/QIBM/UserData/WBIServer44/WebShareICSName/connectors/connectorInstance

dove connectorInstance identifica in modo univoco l'istanza del connettore e dove WebSphereICSName è il nome dell'istanza Interchange Server con cui viene eseguito il connettore.

Se il connettore dispone di metaoggetti specifici del connettore, occorre creare un metaoggetto per l'istanza del connettore. Se si salva il metaoggetto come file, creare questa directory e salvare il file qui:

```
/QIBM/UserData/WBIServer44/WebSphereICSName/repository  
/connectorInstance dove WebSphereICSName è il nome dell'istanza Interchange  
Server Express  
con cui viene eseguito il connettore.
```

- **Per piattaforme Linux:**

ProductDir/connectors/connectorInstance dove connectorInstance identifica in modo univoco l'istanza del connettore. Se il connettore dispone di metaoggetti specifici del connettore, occorre creare un metaoggetto per l'istanza del connettore. Se si salva il metaoggetto come file, creare questa directory e salvare il file qui:

```
ProductDir/repository/connectorInstance.
```

È possibile specificare il nome del server InterChange Server Express come un parametro di connector\_manager; un esempio è: start\_manager-start connName WebSphereICSName [-cConfigFile].

## Creazione di definizioni di oggetti di business

Se le definizioni degli oggetti di business per ciascuna istanza del connettore non esistono già all'interno del progetto, occorre crearle.

1. Se occorre modificare le definizioni degli oggetti di business che vengono associare al connettore iniziale, copiare i file appropriati ed utilizzare Business Object Designer Express per importarli. È possibile copiare qualunque file per il connettore iniziale. Rinominarli se si apportano delle modifiche a file.
2. I file per il connettore iniziale devono essere nella seguente directory:  
ProductDir\repository\initialConnectorInstance

Qualsiasi file aggiuntivo creato deve essere nell'appropriata subdirectory connectorInstance di ProductDir\repository.

## Creazione di una definizione del connettore

Si crea un file di configurazione (definizione connettore) per la istanza del connettore in Connector Configurator Express. A tale scopo:

1. Copiare il file di configurazione del connettore iniziale (definizione del connettore) e rinominarlo.
2. Accertarsi che l'istanza del connettore elenchi i relativi oggetti di business supportati (e tutti i metaoggetti associati).
3. Personalizzare le proprietà del connettore come appropriate.

## Creazione di una script di avvio

Per creare una script di avvio

1. Copiare la script di configurazione del connettore iniziale e assegnarle un nome per includere il nome della directory del connettore:  
dirname
2. Salvare lo script di avvio nella directory del connettore creato in “Creazione di definizioni di oggetti di business” a pagina 94.
3. Solo per Windows. Creare un collegamento alla script di avvio.
4. Solo per Windows. Copiare il testo del collegamento del connettore iniziale e modificare il nome del connettore iniziale (dalla riga comandi) affinché corrisponda alla nuova istanza del connettore.
5. Solo per i5/OS. Creare una descrizione del lavoro per il connettore utilizzando le informazioni qui di seguito:  
CRTDUPOBJ(QWBIJDBC) FROMLIB(QWBISVR44)OBJTYPE(\*JOB)  
TOLIB (QWBISVR44) NEWOBJ(newjdbcname) dove newjdbcname è un nome di 10 caratteri utilizzato per la descrizione del lavoro per il nuovo connettore.
6. Solo per i5/OS. Aggiungere il nuovo connettore a WebSphere Business Integration Server Express Console. Per informazioni sulla console di WebSphere Business Integration Server Express, fare riferimento all’aiuto in linea fornito con la console.



---

## Capitolo 8. Risoluzione dei problemi e gestione degli errori

Il capitolo descrive i problemi che possono verificarsi all'avvio o durante l'esecuzione del connettore per JDBC. Questo capitolo contiene le sezioni seguenti:

- "Avvio dei problemi"
- "Elaborazione degli eventi"
- "Mappatura di InterChange Server Express"
- "Gestione degli errori e registrazione" a pagina 98
- "Perdita della connessione all'applicazione" a pagina 100
- "Impossibilità di individuare le tabelle di archivio o di eventi quando viene utilizzato DB2" a pagina 100
- "JDBC ODA si comporta in modo non corretto a causa di un driver JDBC non supportato" a pagina 102

---

### Avvio dei problemi

Se si verificano difficoltà quando si tenta di avviare il connettore, verificare che il broker di integrazione sia attivo e in esecuzione.

---

### Elaborazione degli eventi

Se sono presenti eventi nella tabella eventi e non stanno per essere elaborati durante l'esecuzione del connettore, verificare che:

- Il processo aziendale rilevante sia in esecuzione.
- Il nome dell'oggetto business nella tabella eventi corrisponda al nome dell'oggetto business specificato per la porta del processo business.

---

### Mappatura di InterChange Server Express

Questa sezione discute dei seguenti argomenti:

- "Problemi relativi alla mappatura"
- "Conversione data"

#### Problemi relativi alla mappatura

Se gli oggetti business non sono mappati o la mappatura non è stata richiamata, verificare per assicurarsi che le mappe siano state installate nella directory corrente.

#### Conversione data

**Nota:** Questa procedura di conversione data si applica solo alle versioni del connettore precedenti alla versione 1.5.0.

Utilizzare le mappe per convertire i dati memorizzate nel formato Data nel database al formato Stringa utilizzato dall'oggetto business.

Ad esempio, si supponga di voler convertire la seguente data, memorizzata nel database:

```
Sun Jan 01 00:00:00 CEST 1999
```

Alla seguente stringa, elaborata dall'adattatore per l'oggetto business JDBC:  
Jan 01 1999 00:00:00

Per effettuare questa conversione, utilizzare le utilità di creazione `DtpDate()` e `DtpSplitString()` definite per la trasformazione dati nella mappatura. Per la sintassi e la descrizione di tali utilità di creazione e le classi degli oggetti creati, consultare il manuale *Map Development Guide*.

Per utilizzare una mappa per convertire il valore `Data` in `String`, procedere nel modo seguente:

1. Utilizzare `DtpSplitString()` con un delimitatore di spazi per separare la stringa in sei parti e riorganizzarla in un ordine che `DtpDate` può utilizzare. Per convertire la data di esempio, utilizzare:

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

Nell'istruzione precedente, `OurSplitString` è una variabile definita dall'utente del tipo `DtpSplitString` e viene specificato uno spazio come delimitatore.

2. Utilizzare il metodo `nextElement()` della classe `DtpSplitString` per il ciclo infinito mediante la variabile `OurSplitString` appena creata, posizionando ciascuno dei sei elementi della variabile in una serie i cui elementi sono di tipo `String`. Il seguente esempio specifica `OurStringPieces` come serie di output:

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

Questo loop produce i seguenti elementi della serie:

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

3. Concatenare le parti della stringa necessaria per l'input `DtpDate`. La conversione di esempio utilizza "M G A h:m:s" come formato di input per `DtpDate`, che richiede la stringa convertita in questo modo "Jan 01 1999 00:00:00". Questo esempio `String` utilizza gli elementi 1, 2, 5 e 3 della serie `OurStringPieces`:

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

4. Utilizzare la nuova stringa concatenata come input in `DtpDate`:

```
DtpDate OurDtpDate = new DtpDate(OurConcatenatedString, "M D Y h:m:s");
```

Una volta modificato il valore `Data` in formato `DtpDate`, è possibile effettuare operazioni con la data nella mappa.

---

## Gestione degli errori e registrazione

Il connettore registra un messaggio di errore ogni volta in cui trova una condizione in cui non riesce l'elaborazione di un oggetto business e di un'espressione. Quando si verifica tale errore, il connettore stampa anche una rappresentazione di testo dell'oggetto business non riuscito come è stato ricevuto. Il testo viene scritto nel file di log del connettore o nel flusso di output standard, in base alla configurazione. E' possibile utilizzare il testo come aiuto per la determinazione dell'origine dell'errore.

## Tipi di errori

Tabella 14 descrive i tipi di messaggi di traccia che il connettore restituisce a ciascuna livello di traccia. Questi messaggi sono aggiunti all'output dei messaggi di traccia dall'architettura di WebSphere Business Integration Server Express Adapter, come ad esempio il wrapper di esecuzione del connettore Java.

Tabella 14. Messaggi di traccia del connettore

Livello di traccia	Messaggi di traccia
Livello 0	Messaggio che identifica la versione del connettore. A questo livello non viene effettuata alcuna altra traccia. Questo è il valore predefinito.
Livello 1	<ul style="list-style-type: none"><li>• Messaggi di stato</li><li>• Messaggi che forniscono le informazioni di identificazione (chiavi) per ciascun oggetto business elaborato</li><li>• Messaggi che vengono recapitati ogni volta che viene eseguito il metodo <code>pollForEvents</code></li></ul>
Livello 2	<ul style="list-style-type: none"><li>• I messaggi del gestore dell'oggetto business che contengono le informazioni quali gli oggetti business child e di serie che il connettore incontra o richiama durante l'elaborazione di un oggetto business.</li><li>• I messaggi registrati ogni volta che un oggetto business viene inviato al broker di integrazione, da <code>gotAppEvent()</code> o da <code>executeCollaboration()</code></li><li>• I messaggi che indicano che un oggetto business è stato ricevuto come richiesta del broker di integrazione</li></ul>
Livello 3	<ul style="list-style-type: none"><li>• La chiave esterna che elabora i messaggi contenenti tali informazioni, come quando il connettore ha trovato o impostato una chiave esterna in un oggetto business</li><li>• I messaggi che forniscono le informazioni sull'elaborazione dell'oggetto business. Ad esempio, questi messaggi vengono recapitati quando il connettore trova una corrispondenza tra gli oggetti business o trova un oggetto business in una serie di oggetti business child</li></ul>
Livello 4	<ul style="list-style-type: none"><li>• I messaggi informativi specifici dell'applicazione, ad esempio quelli che visualizzano i valori restituiti dalle funzioni che analizzano i campi relativi alle informazioni specifiche dell'applicazione degli oggetti business</li><li>• I messaggi che identificano quando il connettore entra o esce da una funzione e che consentono di eseguire la traccia del flusso del processo del connettore</li><li>• Tutti i messaggi specifici del thread. Se il connettore inoltra più thread, viene visualizzato un messaggio per la creazione di ciascun nuovo thread</li></ul>

Tabella 14. Messaggi di traccia del connettore (Continua)

Livello di traccia	Messaggi di traccia
Livello 5	<ul style="list-style-type: none"><li>• I messaggi che indicano l'inizializzazione del connettore, ad esempio quelli che visualizzano il valore di ciascuna proprietà di configurazione richiamata dal broker di integrazione</li><li>• I messaggi che comprendono le istruzioni eseguite nell'applicazione. A questo livello di traccia, il file di log del connettore contiene tutte le istruzioni eseguite nell'applicazione di destinazione e il valore di qualunque variabile sostituita.</li><li>• I messaggi che comprendono una rappresentazione di un oggetto business prima che il connettore inizi ad elaborarlo (visualizzandone lo stato al momento in cui il connettore lo riceve) e dopo che il connettore ne ha completato l'elaborazione (visualizzandone lo stato al momento in cui il connettore lo restituisce)</li><li>• I messaggi che comprendono un dump di un oggetto business</li><li>• I messaggi che indicano lo stato di ciascun thread che il connettore inoltra durante l'esecuzione</li></ul>

## Messaggi di errore

### File di messaggi del connettore

Tutti i messaggi di errore generati dal connettore vengono memorizzati in un file di messaggi denominato `JDBCCConnector.txt` o `JDBCCConnector_II_TT.txt` (dove *II* specifica una lingua e *TT* specifica un paese o una regione). Ciascun errore dispone di un numero di errore seguito dal messaggio di errore. Ad esempio:

20017

La versione dell'infrastruttura del connettore non corrisponde.

20018

La connessione da {1} all'applicazione si è interrotta! Immettere 'q' per arrestare il connettore, quindi riavviarlo dopo aver risolto il problema.

20019

Error: `ev_id` is NULL in `pollForEvent()`.

---

## Perdita della connessione all'applicazione

Se il connettore non riesce a stabilire una connessione, invia FAIL al broker di integrazione e termina.

Quando `AutoCommit` viene impostato su `false` e `PingQuery` non riesce, il connettore tenterà di creare una nuova connessione al database. Se è possibile creare una nuova connessione al database, l'elaborazione contiene, altrimenti il connettore restituisce un `APPRESPONSETIMEOUT` e verrà terminato.

---

## Impossibilità di individuare le tabelle di archivio o di eventi quando viene utilizzato DB2

Durante l'avvio, il connettore tenta di localizzare le tabelle di archivio e di eventi all'interno del database specificato dalla proprietà di configurazione `SchemaName`. Se si utilizza DB2 come database, talvolta il connettore non riesce a trovare le tabelle di archivio e di eventi e restituisce il seguente errore:

La tabella eventi/archivio `table_name` non esiste nel database.



Per evitare questo problema, specificare sempre i nomi dello schema DB2 in lettere maiuscole (ad esempio, SUSER) nella proprietà di configurazione SchemaName del connettore

---

## Abilitazione del connettore per effettuare operazioni con un database DB2

Prima di utilizzare il connettore con un database DB2, è necessario procedere nel modo seguente (per Linux, passare alla serie successiva di istruzioni):

**Nota:** Per i5/OS, non è richiesta alcuna azione per abilitare il connettore per effettuare operazioni con un database DB2.

1. Copiare il file denominato db2java.zip dall'host DB2 nella directory *ProductDir*\lib sulla macchina in cui è in esecuzione il connettore.
2. Copiare il file denominato db2jdbc.dll dall'host DB2 alla directory *ProductDir*\bin sulla macchina in cui sta per essere eseguito il connettore.
3. In base al sistema operativo, modificare quanto segue nel file YP del connettore (start\_JDBC.sh o start\_JDBC.bat):

**Windows:** impostare JDBC\_DRIVER\_PATH=*ProductDir*\lib\db2java.zip

4. Sulla macchina host DB2, avviare il processo DB2/bin/db2jstrt. Verificare di specificare il numero di porta utilizzata (ad esempio, DB2/bin/db2jstrt 50000).
5. Impostare il valore della proprietà JDBC\_DRIVER\_CLASS del connettore su COM.ibm.db2.jdbc.net.DB2Driver (o COM.ibm.db2.jdbc.app.DB2Driver se il database DB2 si trova sulla stessa macchina in cui sta per essere eseguito il connettore).
6. Impostare il valore della proprietà DATABASE\_URL del connettore su jdbc:db2://*MachineName*:*PortNumber*/*DBname* (or jdbc:db2:*DBname* se il database DB2 si trova sulla stessa macchina in cui il connettore sta per essere eseguito).

Per Linux prima di utilizzare il connettore con un database DB2, è necessario procedere nel modo seguente:

1. Copiare il file denominato db2java.zip dalla directory DB2, ad esempio /opt/IBM/db2/v8.1/java nella directory *\$ProductDir*/lib sulla macchina in cui sta per essere eseguito il connettore.
2. Copiare il file denominato libdb2dbc.so dalla directory DB2, ad esempio /opt/IBM/db2/v8.1/lib nella directory *\$ProductDir*/bin sulla macchina in cui sta per essere eseguito il connettore.
3. Modificare quanto segue nel file di avvio del connettore start\_JDBC.sh: JDBC\_DRIVER\_PATH="*\$(CROSSWORLDS)*/lib/db2java.zip".
4. Creare un catalogo sulla macchina in cui il connettore sta per essere eseguito per il database DB2 cui accederà il connettore, anche se si trova nella macchina host locale. Si supponga che il database sia denominato SMBDB e che il nome host della macchina in cui si trova sia denominato remhost. Sulla macchina locale, immettere i seguenti comandi nella riga comandi DB2:  
db2=> CATALOG TCPIP MODE remnode REMOTE remhost SERVER 50000  
Questo comando crea un nodo TCP/IP sulla macchina locale denominata remnode per l'host remhost e remhost può essere il nome host della macchina locale se il database è locale.  
db2=> CATALOG DATABASE SMBDB AS SMBDBALI AT MODE remnode  
Questo comando crea un alias denominato SMBDBALI per il database SMBDB,

che si trova sul nodo remnode. Impostare il valore della proprietà DatabaseURL del connettore su jdbc:db2:SMDBDALI. Impostare il valore della proprietà JDBCClass del connettore su COM.ibm.db2.jdbc.app.DB2Driver.

**Nota:** E' importante eseguire il passo 4 quando il database DB2 si trova sulla piattaforma Linux. Se si accede direttamente al database invece di accedervi mediante i relativi alias, potrebbe verificarsi un errore a causa di un problema sconosciuto con il driver DB2 CLI sulla piattaforma Linux. Questo passo deve essere eseguito affinché JDBCODA acceda al database DB2 sulla piattaforma Linux.

Se si specifica il valore della proprietà JDBCClass del connettore COM.ibm.db2.jdbc.net.DB@Driver, procedere nel modo seguente:

1. Copiare i file denominati db2java.zip e db2jcc.jar dalla directory DB2, ad esempio la directory /opt/IBM/db2/v8.1/java alla directory \$Product/Dir/lib sulla macchina in cui sta per essere eseguito il connettore.
2. Copiare il file denominato libdb2jdbc.so dalla directory DB2, ad esempio, /opt/IBM/db2/v8.1/lib, alla directory \$ProductDir/bin sul computer in cui sta per essere eseguito il connettore.
3. Modificare quanto segue nel file di avvio del connettore start\_JDBC.sh:  
JBCDRIVERPATH=  
"\${CROSSWORLDS}/lib/db2java.zip:\${CROSSWORLDS}/lib/db2jcc.jar".
4. Impostare il valore della proprietà DatabaseURL del connettore su jdbc:db2://MachineName:PortNumber/DBname.
5. Sulla macchina host DB2, avviare il processo /opt/IBM/db2/v8.1/bin/db2jstrt. Assicurarsi di specificare il numero di porta che si utilizza.

---

## JDBC ODA si comporta in modo non corretto a causa di un driver JDBC non supportato

Se il driver JDBC non supporta una funzione di JDBC, ODA, l'utilità object discovery agent non funziona correttamente. Ad esempio, se il driver non supporta tutte le chiamate di metodo usate da JDBC ODA, il log JDBC ODA indica il processo non riuscito. Di seguito viene riportato un esempio del log:

```
[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 [ProductName][ODBC ProductName Driver]Optional feature not
implemented]
```

In tal caso, è necessario utilizzare un driver JDBC diverso.

---

## Appendice A. Proprietà di configurazione standard per i connettori

In questa appendice vengono descritti le proprietà standard di configurazione per il componente del connettore degli adattatori di WebSphere Business Integration Server Express. Le informazioni riguardano InterChange Server Express.

Se l'adattatore supporta DB2 Information Integrator, utilizzare le opzioni WMQI e le proprietà standard DB2 II (consultare la colonna Note in Tabella 15 a pagina 105.)

Per informazioni sulle proprietà specifiche di questo connettore, consultare la relativa sezione in questa guida.

---

### Nuove proprietà

Queste proprietà standard sono state aggiunte in questo rilascio:

- AdapterHelpName
- BiDi.Application
- BiDi.Broker
- BiDi.Metadata
- BiDi.Transformation
- ControllerEventSequencing
- jms.ListenerConcurrency
- jms.TransportOptimized
- ResultsSetEnabled
- ResultsSetSize
- TivoliTransactionMonitorPerformance

---

### Panoramica sulle proprietà del connettore standard

I connettori possiedono due tipi di proprietà di configurazione:

- Le proprietà di configurazione standard, utilizzate dalla struttura
- Proprietà delle applicazione oppure specifiche del connettore, di configurazione, utilizzate dall'agente

Tali proprietà determinano la struttura dell'adattatore e il comportamento di esecuzione dell'adattatore.

In questa sezione viene descritto come avviare Connector Configurator Express e descrive le caratteristiche comuni a tutte le proprietà. Per informazioni sulle proprietà di configurazione specifiche per un connettore, consultare la guida per l'utente relativa all'adattatore.

### Avvio di Connector Configurator Express

Le proprietà del connettore vengono configurate da Connector Configurator Express, che si accede da System Manager. Per ulteriori informazioni utilizzando Connector Configurator Express, fare riferimento alle sezioni su Connector Configurator Express in questa guida.

Connector Configurator Express e System Manager vengono eseguiti solo su sistema Windows. Se si sta eseguendo il connettore su un sistema Linux, occorre avere una macchina Windows con questi tool installati.

Per impostare le proprietà per un connettore che viene eseguito su Linux, occorre avviare System Manager sulla macchina Windows, collegarsi al broker di integrazione di Linux e visualizzare Connector Configurator Express per il connettore.

## Panoramica dei valori di proprietà di configurazione

Per il connettore utilizzare il seguente ordine per determinare un valore della proprietà:

1. Predefinito
2. Repository per il broker di integrazione di InterChange Server Express.
3. File di configurazione locale
4. Riga comandi

La lunghezza predefinita di un campo proprietà è di 255 caratteri. Non esiste limite di lunghezza per un tipo di proprietà STRING. La lunghezza di un tipo INTEGER viene determinata dal server su cui è in esecuzione l'adattatore.

Un connettore ottiene i valori di configurazione all'avvio. Se si modifica il valore di una o più proprietà del connettore durante una sessione runtime, il metodo di aggiornamento della proprietà determina come avrà effetto tale modifica.

Per le caratteristiche aggiornate di una proprietà, cioè, come e quando diventa effettiva una modifica alle proprietà del connettore, dipende dalla natura della proprietà.

Esistono quattro metodi per le proprietà del connettore standard:

- **Dinamica**  
Il nuovo valore diventa effettivo immediatamente dopo che la modifica viene salvata in System Manager. Tuttavia, se il connettore è in modo autonomo (indipendentemente da System Manager).
- **L'agente viene riavviato (solo InterChange Server Express)**  
Il nuovo valore ha effetto solo dopo l'arresto e il riavvio dell'agente di connessione.
- **Riavvio del componente**  
Il nuovo valore diventa effettivo solo dopo che il connettore viene arrestato e riavviato in System Manager. Non occorre arrestare e riavviare l'agente o il processo del server.
- **Riavvio del sistema**  
Il nuovo valore diventa effettivo solo dopo l'arresto e il riavvio dell'agente del connettore e del server.

Per determinare come viene aggiornata una proprietà specifica, fare riferimento alla colonna **Metodo aggiornato** nella finestra Connector Configurator Express oppure consultare la colonna di Metodo aggiornato in Tabella 15 a pagina 105.

Vi sono tre posizioni in cui può essere localizzata una proprietà standard. Alcune proprietà possono essere localizzate in più di una posizione.

- **ReposController**  
La proprietà viene localizzata nel controller del connettore ed ha effetto solo in tale posizione. Se si modifica il valore da parte dell'agente, ciò non ha effetto sul controller.
- **ReposAgent**  
La proprietà viene localizzata nell'agente ed ha effetto solo in tale posizione. Una configurazione locale può sostituire tale valore, a seconda della proprietà.
- **LocalConfig**  
La proprietà viene localizzata nel file di configurazione per il connettore e può agire solo attraverso il file di configurazione. Il controller controller non può modificare il valore della proprietà e non è al corrente delle modifiche effettuate del file di configurazione fino a che il sistema non è ridistribuito per aggiornare il controller in modo esplicito.

## Guida di riferimento rapido per le proprietà standard

Tabella 15 fornisce una guida di riferimento rapido per le proprietà di configurazione del connettore standard. Non per tutti i connettori sono necessarie tutte queste proprietà e le impostazioni delle proprietà potrebbero essere differenti.

Consultare la tabella nella la sezione qui di seguito per una descrizione di ciascuna proprietà.

**Nota:** Nella colonna Note in Tabella 15, la frase "RepositoryDirectory è impostato su <REMOTE>" indica che il broker è InterChange Server Express.

Tabella 15. Riepilogo delle proprietà di configurazione standard

Nome proprietà	Valori possibili	Valore predefinito	Metodo aggiornato	Note
AdapterHelpName	Una delle subdirectory in <ProductDir>\bin\Data\App\Help che possiede una directory per <Impostazioni internazionali> valida	Il nome della maschera, se <Componente> restart	Componente restart	Impostazioni internazionali supportate. Sono comprese chs_chn, cht_twn, deu_deu, esn_esp, fra_fra, ita_ita, jpn_jpn, kor_kor, ptb_bra, e enu_usa (valore predefinito).
AdminInQueue	Nome di coda JMS valido	<CONNECTORNAME>/ADMININQUEUE	Componente restart	Questa proprietà è valida solo quando il valore di DeliveryTransport è JMS
AdminOutQueue	Nome di coda JMS valido	<CONNECTORNAME>/ADMINOUTQUEUE	Componente restart	Questa proprietà è valida solo quando il valore di DeliveryTransport è JMS
AgentConnections	Da 1 a 4	1	Componente restart	Questa proprietà è valida solo quando il valore di DeliveryTransport è IDL, il valore di Repository Directory è impostato a <REMOTE> e il valore di BrokerType è ICS.
AgentTraceLevel	Da 0 a 5	0	Dinamica  Per ICS; otherwise Componente restart	

Tabella 15. Riepilogo delle proprietà di configurazione standard (Continua)

Nome proprietà	Valori possibili	Valore predefinito	Metodo aggiornato	Note
ApplicationName	Nome dell'applicazione	Il valore specificato per il file di configurazione	Componente restart	Applicazione del connettore
BiDi.Application	Qualsiasi combinazione valida di: 1a lettera I,V 2a lettera L,R 3a lettera Y, N 4a lettera S, N 5a lettera H, C, N	Il valore specificato nel file di configurazione	Componente restart	Questa proprietà è valida solo se il valore di BiDi.Transformation è true
BiDi.Broker	Qualsiasi combinazione valida di: 1a lettera I,V 2a lettera L,R 3a lettera Y, N 4a lettera S, N 5a lettera H, C, N	Il valore specificato nel file di configurazione	Componente restart	Questa proprietà è valida solo se il valore di BiDi.Transformation è true. Se il valore di BrokerType è ICS, la proprietà è di sola lettura.
BiDi.Metadata	Qualsiasi combinazione valida di: 1a lettera I,V 2a lettera L,R 3a lettera Y, N 4a lettera S, N 5a lettera H, C, N	Il valore specificato nel file di configurazione	Componente restart	Questa proprietà è valida solo se il valore di BiDi.Transformation è true.
BiDi.Transformation	true oppure false	false	Componente restart	Questa proprietà è valida solo se il valore di BrokerType è non WAS
BrokerType	ICS	ICS	Componente restart	
CharacterEncoding	Qualsiasi codice supportato. L'elenco visualizza questo insieme: ascii7, ascii8, SJIS, Cp949, GBK, Big5, Cp297, Cp273, Cp280, Cp284, Cp037, Cp437	ascii7	Componente restart	Questa proprietà è valida solo per i connettori C++.
CommonEventInfrastructure	true oppure false	false	Componente restart	
CommonEventInfrastructure	Una stringa URL, ad esempio corbaloc:iiop: host:2809.	Nessun valore predefinito.	Componente restart	Questa proprietà è valida solo se il valore di CommonEvent Infrastructure è true.
ConcurrentEventTriggeredFlows	Da 1 a 32,767	1	Componente restart	Questa proprietà è valida solo se il valore di RepositoryDirectory è impostato su <REMOTE> e il valore di BrokerType è ICS.
ContainerManagedEvents	Vuoto o JMS	Vuoto	Componente restart	Questa proprietà è valida solo quando il valore di Delivery Transport è JMS.
ControllerEventSequencing	true oppure false	true	Dinamica	Questa proprietà è valida solo se il valore di Repository Directory è impostato su <REMOTE> e il valore di BrokerType è ICS.

Tabella 15. Riepilogo delle proprietà di configurazione standard (Continua)

Nome proprietà	Valori possibili	Valore predefinito	Metodo aggiornato	Note
ControllerStoreAndForwardMode	true oppure false	true	Dinamica	Questa proprietà è valida solo se il valore di Repository Directory è impostato su <REMOTE> e il valore di BrokerType è ICS.
ControllerTraceLevel	Da 0 a 5	0	Dinamica	Questa proprietà è valida solo se il valore di RepositoryDirectory è impostato su <REMOTE> e il valore di BrokerType è ICS.
DeliveryQueue	Qualsiasi nome di coda JMS	<CONNECTORNAME>/DELIVERYQUEUE	Componente restart	Questa proprietà è valida solo quando il valore di Delivery Transport è JMS.
DeliveryTransport	IDL o JMS	IDL quando il valore di RepositoryDirectory è <REMOTE>, altrimenti JMS	Componente restart	Se il valore di RepositoryDirectory non è <REMOTE>, l'unico valore valido per questa proprietà è JMS.
DuplicateEventElimination	true oppure false	false	Componente restart	Questa proprietà è valida solo se il valore di DeliveryTransport è JMS.
EnableOidForFlowMonitoring	true oppure false	false	Componente restart	Questa proprietà è valida solo se il valore di BrokerType è ICS.
FaultQueue	Qualsiasi nome di coda valido	<CONNECTORNAME>/FAULTQUEUE	Componente restart	Questa proprietà è valida solo se il valore di DeliveryTransport è JMS.
jms.FactoryClassName	CxCommon.Messaging.jms.IBMQueueConnectionFactory, CxCommon.Messaging.jms.SonicMQFactory, oppure qualsiasi nome di classe Java	CxCommon.Messaging.Jms.IBMQueueConnectionFactory	Componente restart	Questa proprietà è valida solo se il valore di DeliveryTransport è JMS.
jms.ListenerConcurrency	Da 1 a 32767	1	Componente restart	Questa proprietà è valida solo se il valore di jms.TransportOptimized è true.
jms.MessageBrokerName	Se il valore di jms.FactoryClassName è IBM, utilizzare crossworlds.queue.manager.	crossworlds.queue.manager	Componente restart	Questa proprietà è valida solo se il valore di DeliveryTransport è JMS.
jms.NumConcurrentRequests	Intero positivo	10	Componente restart	Questa proprietà è valida solo se il valore di DeliveryTransport è JMS.
jms.Password	Qualsiasi password valida		Componente restart	Questa proprietà è valida solo se il valore di DeliveryTransport è JMS.
jms.TransportOptimized	true oppure false	false	Componente restart	Questa proprietà è valida solo se il valore di DeliveryTransport è JMS e il valore di BrokerType è ICS.

Tabella 15. Riepilogo delle proprietà di configurazione standard (Continua)

Nome proprietà	Valori possibili	Valore predefinito	Metodo aggiornato	Note
jms.UserName	Qualsiasi nome valido		Componente restart	Questa proprietà è valida solo se il valore di Delivery Transport è JMS.
JvmMaxHeapSize	Dimensione heap in megabyte	128 M	Componente restart	Questa proprietà è valida solo se il valore di Repository Directory è impostato su <REMOTE> e il valore di BrokerType è ICS.
JvmMaxNativeStackSize	Dimensione dello stack in kilobyte	128k	Componente restart	Questa proprietà è valida solo se il valore di Repository Directory è impostato su <REMOTE> e il valore di BrokerType è ICS.
JvmMinHeapSize	Dimensione heap in megabyte	1 MB	Componente restart	Questa proprietà è valida solo se il valore di Repository Directory è impostato su <REMOTE> e il valore di BrokerType è ICS.
ListenerConcurrency	Da 1 a 100	1	Componente restart	Questa proprietà è valida solo se il valore di DeliveryTransport è MQ.
Locale	Questo è un sottoinsieme di quelli supportate: en_US, ja_JP, ko_KR, zh_CN, zh_TW, fr_FR, de_DE, it_IT, es_ES, pt_BR		Componente restart	
LogAtInterchangeEnd	true oppure false	false	Componente restart	Questa proprietà è valida solo se il valore di Repository Directory è impostato su <REMOTE> e il valore di BrokerType è ICS.
MaxEventCapacity	Da 1 a 2147483647	2147483647	Dinamica	Questa proprietà è valida solo se il valore di Repository Directory è impostato su <REMOTE> e il valore di BrokerType è ICS.
MessageFileName	Nome di file valido	InterchangeSystem.txt	Componente restart	
MonitorQueue	Qualsiasi nome di coda valido	<CONNECTORNAME> /MONITORQUEUE	Componente restart	Questa proprietà è valida solo se il valore di DuplicateEventElimination è true e ContainerManagedEvents non dispone di alcuna valore.
OADAutoRestartAgent	true oppure false	false	Dinamica	Questa proprietà è valida solo se il valore di Repository Directory è impostato su <REMOTE> e il valore di BrokerType è ICS.



Tabella 15. Riepilogo delle proprietà di configurazione standard (Continua)

Nome proprietà	Valori possibili	Valore predefinito	Metodo aggiornato	Note
OADMaxNumRetry	Un numero intero positivo	1000	Dinamica	Questa proprietà è valida solo se il valore di Repository Directory è impostato su <REMOTE> e il valore di BrokerType è ICS.
OADRetryTimeInterval	Un numero intero positivo in minuti	10	Dinamica	Questa proprietà è valida solo se il valore di Repository Directory è impostato su <REMOTE> e il valore di BrokerType è ICS.
PollEndTime	HH = da 0 a 23 MM = da 0 a 59	HH:MM	Componente restart	
PollFrequency	Un numero intero positivo (in millisecondi)	10000	Dinamica se il broker è ICS; otherwise Componente restart	
PollQuantity	Da 1 a 500	1	Riavvio dell'agente	Questa proprietà è valida solo se il valore di ContainerManagedEvents è JMS.
PollStartTime	HH = da 0 a 23 MM = da 0 a 59	HH:MM	Componente restart	
RepositoryDirectory	<REMOTE> se il broker è ICS; altrimenti qualsiasi directory locale valida.	Per ICS, il valore è impostato su <REMOTE>.	Riavvio <Remote>	
RequestQueue	Nome di coda JMS valido	<CONNECTORNAME>/REQUESTQUEUE	Componente restart	Questa proprietà è valida solo se il valore di DeliveryTransport è JMS
ResponseQueue	Nome di coda JMS valido	<CONNECTORNAME>/RESPONSEQUEUE	Componente restart	Questa proprietà è valida solo se il valore di DeliveryTransport è JMS.
RestartRetryCount	Da 0 a 99	3	Dinamica se ICS; otherwise Componente restart	
RestartRetryInterval	Un valore in minuti da 1 a 2147483647	1	Dinamica se ICS; otherwise Componente restart	
ResultSetEnabled	true oppure false	false	Componente restart	Utilizzato solo da connettori che supportano DB2II.
ResultSetSize	Intero positivo	0 (significa che la dimensione della serie è illimitata)	Componente restart	Utilizzato solo da connettori che supportano DB2II.  Questa proprietà è valida solo se il valore di ResultSetEnabled è true.

Tabella 15. Riepilogo delle proprietà di configurazione standard (Continua)

Nome proprietà	Valori possibili	Valore predefinito	Metodo aggiornato	Note
RHF2MessageDomain	mrm or xml	mrm	Componente restart	Questa proprietà è valida solo se il valore di DeliveryTransport è JMS e il valore di WireFormat è CwXML.
SourceQueue	Qualsiasi nome di coda MQ	<CONNECTORNAME>/SOURCEQUEUE	Riavvio dell'agente	Questa proprietà è valida solo se il valore di ContainerManagedEvents è JMS.
SynchronousRequest Coda	Qualsiasi nome di coda valido	<CONNECTORNAME>/SYNCHRONOUSREQUEST QUEUE	Componente restart	Questa proprietà è valida solo se il valore di DeliveryTransport è JMS.
SynchronousRequest Timeout	0 su qualsiasi numero (millisecondi)	0	Componente restart	Questa proprietà è valida solo se il valore di DeliveryTransport è JMS.
SynchronousResponse Coda	Qualsiasi nome di coda valido	<CONNECTORNAME>/SYNCHRONOUSRESPONSE QUEUE	Componente restart	Questa proprietà è valida solo se il valore di DeliveryTransport è JMS.
TivoliMonitorTransaction Performance	true oppure false	false	Componente restart	
WireFormat	CwXML oppure CwB0	CwXML	Riavvio dell'agente	Il valore di questa proprietà deve essere CwXML se il valore di RepositoryDirectory non è impostato su <REMOTE>. Il valore deve essere CwB0 se il valore di RepositoryDirectory è impostato su <REMOTE>.
WsifSynchronousRequestTimeout	0 per qualsiasi numero (millisecondi)	0	Componente restart	Questa proprietà è valida solo se il valore di BrokerType è ICS.
XMLNameSpaceFormat	short oppure long	short	Riavvio dell'agente	Questa proprietà non è valida se il valore di BrokerType è ICS.

## Proprietà standard

In questa sezione vengono descritte le proprietà di configurazione del connettore standard.

### AdapterHelpName

La proprietà AdapterHelpName è il nome di una directory in cui si trovano i file della guida estesa specifici del connettore. La directory deve essere posizionata in <ProductDir>\bin\Data\App\Help e deve contenere almeno una directory della lingua enu\_usa. È possibile che contenga altre directory a seconda della locale.

Il valore predefinito è il nome della maschera, se è valido, oppure è vuoto.

### AdminInQueue

La proprietà AdminInQueue specifica la coda utilizzata dal broker di integrazione per inviare messaggi amministrativi al connettore.

Il valore predefinito è <CONNECTORNAME>/ADMININQUEUE

## AdminOutQueue

La proprietà AdminOutQueue specifica la coda utilizzata dal connettore per inviare messaggi amministrativi al broker di integrazione.

Il valore predefinito è <CONNECTORNAME>/ADMINOUTQUEUE

## AgentConnections

La proprietà AgentConnections controlla il numero di connessioni ORB (Object Request Broker) aperte quando ORB viene inizializzato.

Il valore predefinito di questa proprietà è 1.

## AgentTraceLevel

La proprietà AgentTraceLevel imposta il livello dei messaggi di traccia per il componente specifico delle applicazioni. Il connettore fornisce tutti i messaggi di traccia applicabili al livello di traccia impostato e più basso.

Il valore predefinito è 0.

## ApplicationName

La proprietà ApplicationName identifica unicamente il nome dell'applicazione del connettore. Questo nome viene utilizzato dall'amministratore di sistema per verificare l'ambiente di integrazione. Questa proprietà deve avere un valore prima di poter eseguire il connettore.

Il valore predefinito è il nome del connettore.

## BiDi.Application

La proprietà BiDi.Application specifica il formato bidirezionale per dati provenienti da un'applicazione esterna nell'adattatore sotto forma di qualsiasi oggetto di business supportato dall'adattatore. La proprietà definisce gli attributi bidirezionali dei dati delle applicazioni. Tali attributi sono:

- Tipo di testo: implicito o visuale (I oppure V)
- Direzione del testo: da sinistra a destra oppure da destra a sinistra (L oppure R)
- Simbolo tipografico: attivo oppure disattivo (Y oppure N)
- Simbolo (Arabo): attivo oppure disattivo (S oppure N)
- Simbolo numerico (arabo): Hindi, contestuale o nominale (H, C oppure N)

Questa proprietà è valida solo se il valore della proprietà BiDi.Transformation è impostato su true.

Il valore predefinito è ILYNN (implicito, da sinistra a destra, attivo, disattivo, nominale).

## BiDi.Broker

La proprietà BiDi.Broker specifica il formato bidirezionale per i dati inviati dall'adattatore al broker di integrazione sotto forma di qualsiasi oggetto di business supportato. Esso definisce gli attributi bidirezionali dei dati, che sono così come elencati in BiDi.Application sopra citato.

Questa proprietà è valida solo se il valore della proprietà `BiDi.Transformation` è impostato su `true`. Se la proprietà `BrokerType` è `ICS`, il valore della proprietà è di sola lettura.

Il valore predefinito è `ILYNN` (implicito, da sinistra a destra, attivo, disattivo, nominale).

## BiDi.Metadata

La proprietà `BiDi.Metadata` definisce il formato bidirezionale oppure gli attributi per i metadati, che vengono utilizzati dal connettore per stabilire e mantenere un collegamento all'applicazione esterna. Le impostazioni dell'attributo sono specifiche per ciascun adattatore utilizzando le capacità bidirezionali. Se l'adattatore supporta l'elaborazione bidirezionale, fare riferimento alla sezione relativa alla proprietà specifiche dell'adattatore per ulteriori informazioni.

Questa proprietà è valida solo se il valore della proprietà `BiDi.Transformation` è impostato su `true`.

Il valore predefinito è `ILYNN` (implicito, da sinistra a destra, attivo, disattivo, nominale).

## BiDi.Transformation

La proprietà `BiDi.Transformation` definisce se il sistema effettua una trasformazione bidirezionale al runtime.

Se il valore della proprietà è impostato su `true`, le proprietà `BiDi.Application`, `BiDi.Broker` e `BiDi.Metadata` sono disponibili. Se il valore della proprietà è impostato su `false`, le proprietà vengono nascoste.

Il valore predefinito è `false`.

## BrokerType

La proprietà `BrokerType` identifica il broker di integrazione che si sta utilizzando. Il valore è `ICS`, oppure `.`

## CharacterEncoding

La proprietà `CharacterEncoding` specifica il codice dei caratteri utilizzato per effettuare la mappatura da un carattere (quale una lettera dell'alfabeto, una rappresentazione numerica o un segno di interpunzione) ad un valore numerico.

**Nota:** Connettori basati su Java non utilizzano questa proprietà. I connettori C++ utilizzano un valore `ascii7` per questa proprietà.

Per impostazione predefinita, viene visualizzato solo un sottoinsieme di codifiche di caratteri supportati. Per aggiungere altri valori supportati all'elenco, occorre modificare manualmente il file `\Data\Std\stdConnProps.xml` nella directory del prodotto (`<ProductDir>`). Per ulteriori informazioni, consultare l'appendice di `Connector Configurator Express` in questa guida.

## ConcurrentEventTriggeredFlows

La proprietà `ConcurrentEventTriggeredFlows` determina quanti oggetti di business possono essere elaborati contemporaneamente dal connettore per la consegna degli eventi. Il valore di questo attributo viene impostato sul numero di oggetti di

business di cui viene eseguita la mappatura e che vengono consegnati contemporaneamente. Ad esempio, se si imposta il valore di questa proprietà su 5, vengono eseguiti contemporaneamente cinque oggetti di business.

Impostando tale proprietà sul un valore maggiore di 1 consente ad un connettore per un'applicazione origine di eseguire contemporaneamente la mappatura a molteplici oggetti di business degli eventi e di fornirli a più istanze di collaborazione simultaneamente. Ciò rende più veloce la consegna degli oggetti di business del broker di integrazione particolarmente se gli oggetti di business utilizzano le mappe complete. Aumentando la velocità di arrivo gli oggetti di business per collaborazione possono migliorare le prestazioni generali nel sistema.

Per implementare l'elaborazione contemporanea per un intero flusso (da un'applicazione di origine da una destinazione), devono essere configurate le seguenti proprietà:

- La collaborazione deve essere configurata per utilizzare più thread impostando la relativa proprietà Massimo numero di eventi simultanei ad un valore elemento a sufficiente per utilizzare più thread.
- Il componente specifico dell'applicazione di destinazione deve essere configurato per elaborare le richieste simultaneamente. Ciò significa che deve essere a thread multipli oppure deve essere in grado di utilizzare un parallelismo di agenti di connettori e deve essere configurato per più processi. La proprietà di configurazione Parallel Process Degree deve essere impostata su un valore maggiore di 1.

La proprietà ConcurrentEventTriggeredFlows non influisce sul polling del connettore, che è a singolo thread e deve essere eseguito in modo seriale.

Tale proprietà è valida solo se il valore della proprietà RepositoryDirectory è impostata su <REMOTE>.

Il valore predefinito è 1.

## ContainerManagedEvents

La proprietà ContainerManagedEvents consente ad un connettore abilitato JMS, con una memoria eventi JMS, di fornire la consegna di eventi garantiti, in cui un evento viene eliminato dalla coda origine e posizionato in una coda di destinazione come una transazione JMS.

Quando tale proprietà è impostata su JMS, le seguenti proprietà devono essere anche impostate per abilitare la consegna garantita degli eventi.

- PollQuantity = Da 1 a 500
- SourceQueue = /SOURCEQUEUE

Occorre anche configurare un data handler con le proprietà MimeType e DHClass (data handler class). È inoltre possibile aggiungere DataHandlerConfigMOName (il nome del meta oggetto, che è facoltativo). Per impostare tali valori, utilizzare la scheda **Data Handler** in Connector Configurator Express.

Sebbene tali proprietà siano specifiche dell'adattatore, qui sono elencati alcuni valori di esempio:

- MimeType = text/xml
- DHClass = com.crossworlds.DataHandlers.text.xml
- DataHandlerConfigMOName = MO\_DataHandler\_Default

I campi per questi valori nella scheda **Data Handler** vengono visualizzati solo se è stata impostata la proprietà `ContainerManagedEvents` sul valore `JMS`.

**Nota:** Quando `ContainerManagedEvents` è impostato su `JMS`, il connettore non richiama il metodo `pollForEvents()`, disabilitando perciò questa funzionalità del metodo.

La proprietà `ContainerManagedEvents` è valida solo se il valore della proprietà `DeliveryTransport` è impostato su `JMS`.

Non esiste alcun valore predefinito.

## ControllerEventSequencing

La proprietà `ControllerEventSequencing` abilita la sequenza di eventi nel controller del connettore.

Questa proprietà è valida solo se il valore della proprietà `RepositoryDirectory` è impostato su `<REMOTE>` (`BrokerType` è `ICS`).

Il valore predefinito è `true`.

## ControllerStoreAndForwardMode

La proprietà `ControllerStoreAndForwardMode` imposta il comportamento del controller del connettore dopo aver rilevato che il componente specifico dell'applicazione di destinazione non è disponibile.

Se viene impostata questa proprietà `true` e il componente specifico dell'applicazione di destinazione non è disponibile quando un evento raggiunge `ISC` (InterChange Server Express), il controller del connettore blocca la richiesta al componente specifico dell'applicazione. Quando il componente specifico dell'applicazione diventa operativo, il controller gli inoltra la richiesta.

Tuttavia, se il componente specifico dell'applicazione di destinazione diventa non disponibile dopo che il controller del connettore gli inoltra una richiesta di chiamata di servizio, il controller del connettore non riesce a portare a termine la richiesta.

Se la proprietà è impostata su `false`, il controller del connettore inizia a non portare a termine tutte le richieste delle chiamate di servizio così come rileva che il componente specifico dell'applicazione di destinazione non è disponibile.

Questa proprietà è valida solo se il valore di `RepositoryDirectory` è impostato su `<REMOTE>` (il valore della proprietà `BrokerType` è `ICS`),

Il valore predefinito è `true`.

## ControllerTraceLevel

La proprietà `ControllerTraceLevel` imposta il livello dei messaggi di traccia per il controller del connettore.

Tale proprietà è valida solo se il valore della proprietà `RepositoryDirectory` è impostata su `<REMOTE>`.

Il valore predefinito è `0`.

## DeliveryQueue

La proprietà DeliveryQueue definisce la coda utilizzata dal connettore per inviare messaggi di business al broker di integrazione.

Questa proprietà è valida solo se il valore della proprietà DeliveryTransport è impostato su JMS.

Il valore predefinito è <CONNECTORNAME>/DELIVERYQUEUE

## DeliveryTransport

La proprietà DeliveryTransport specifica il meccanismo di trasporto per la consegna degli eventi. Per Java Messaging Service , il valore è JMS.

- Se il valore della proprietà RepositoryDirectory è impostato su <REMOTE>, il valore della proprietà DeliveryTransport può essere IDL, oppure JMS, e il valore predefinito è IDL.
- Se il valore della proprietà RepositoryDirectory è una directory locale, il valore può essere solo JMS.

Il connettore invia le richieste di chiamata servizio e i messaggi amministrativi su CORBA IIOP se il valore della proprietà RepositoryDirectory è IDL. Il connettore invia le richieste di chiamata servizio e i messaggi di gestione su CORBA IIOP, se il valore della proprietà RepositoryDirectory è IDL.

## JMS

Il meccanismo di trasporto JMS consente la comunicazione tra il connettore e la struttura del connettore del client utilizzando JMS (Java Messaging Service)

Se si seleziona JMS come trasporto della consegna, le proprietà JMS aggiuntive, quali `jms.MessageBrokerName`, `jms.FactoryClassName`, `jms.Password` e `jms.UserName` vengono elencate in Connector Configurator Express. Le proprietà `jms.MessageBrokerName` e `jms.FactoryClassName` vengono richieste per questo trasporto.

È possibile che esista una limitazione di memoria, se si utilizza il meccanismo di trasporto JMS per un connettore nel seguente ambiente, quando ICS (InterChange Server Express) è nel broker di integrazione.

In tale ambiente, è possibile che si abbiano delle difficoltà nell'avviare il controller del connettore (da parte del server) e il connettore (da parte del client) a causa dell'utilizzo della memoria all'interno del client WebSphere MQ. Se l'installazione utilizza meno di 768 MB di dimensione di memoria del processo, impostare la seguente proprietà e variabile:

- Impostare la variabile di ambiente LDR\_CNTRL nella script CWSharedEnv.sh.

Questa script si trova nella directory `\bin` al di sotto della directory del prodotto (<ProductDir>). Utilizzando un editor di testo, aggiungere le seguenti righe come prima riga nella script CWSharedEnv.sh:

```
export LDR_CNTRL=MAXDATA=0x30000000
```

Questa riga limita l'utilizzo della memoria heap ad un massimo di 768 MB (3 segmenti da \* 256 MB). Se la memoria del processo aumenta oltre il limite, è possibile che si verifichi lo swap della pagina, che può influire negativamente sulle prestazioni del sistema.

- Impostare il valore della proprietà `IPCCBaseAddress` su 11 oppure 12. Per ulteriori informazioni su questa proprietà fare riferimento a *WebSphere Business Integration Server Express - Guida all'installazione per Windows, per Linux o per OS/400*.

## DuplicateEventElimination

Quando un valore di questa proprietà è `true`, un connettore abilitato JMS può assicurare che gli eventi duplicati non vengano distribuiti alla coda di consegna. Per utilizzare questa funzione, durante lo sviluppo del connettore, il connettore deve avere un identificativo di evento univoco impostato come l'attributo `ObjectEventId` dell'oggetto di business nel codice specifico dell'applicazione.

**Nota:** Quando un valore di questa proprietà è `true`, la proprietà `MonitorQueue` deve essere abilitata a fornire una consegna garantita degli eventi.

Il valore predefinito è `false`.

## EnableOidForFlowMonitoring

Quando un valore di questa proprietà è `true`, il runtime dell'adattatore contrassegnerà `ObjectEventID` in arrivo come una foreign key per il controllo del flusso.

Questa proprietà è valida solo se la proprietà `BrokerType` è impostata su `ICS`.

Il valore predefinito è `false`.

## FaultQueue

Se il connettore individua un errore durante l'elaborazione di un messaggio, sposta il messaggio (e l'indicatore di stato e la descrizione del problema) alla coda specificata nella proprietà `FaultQueue`.

Il valore predefinito è `<CONNECTORNAME>/FAULTQUEUE`

## jms.FactoryClassName

La proprietà `jms.FactoryClassName` specifica il nome della classe per creare l'istanza per un provider JMS. Questa proprietà deve essere impostata se il valore della proprietà `DeliveryTransport` è impostato su `JMS`.

Il valore predefinito è `CxCommon.Messaging.jms.IBMMQSeriesFactory`.

## jms.ListenerConcurrency

La proprietà `jms.ListenerConcurrency` specifica il numero di listener simultanei per il controller JMS. Specifica il numero di thread che esegue una lettura sequenziale ed elabora messaggi simultaneamente ad un controller.

Questa proprietà è valida solo se il valore della proprietà `jms.OptimizedTransport` è `true`.

Il valore predefinito è `1`.



## jms.MessageBrokerName

`javax.jms.MessageBrokerName` specifica il nome del broker da utilizzare per il provider JMS. Occorre impostare questa proprietà del connettore se si specifica JMS come meccanismo di trasporto di consegna (nella proprietà `DeliveryTransport`).

Quando ci si connette ad un broker di messaggi remoto, questa proprietà richiede i seguenti valori:

`QueueMgrName:Channel:HostName:PortNumber`

dove:

`QueueMgrName` è il nome del manager della coda.

`Channel` è il canale utilizzato dal client.

`HostName` è il nome della macchina dove il manager della coda deve risiedere.

`PortNumber` è il numero della porta utilizzato dal manager della coda per l'ascolto.

Ad esempio:

```
javax.jms.MessageBrokerName = WBIMB.Queue.Manager:CHANNEL1:RemoteMachine:1456
```

Il valore predefinito è `crossworlds.queue.manager`. Utilizzare il valore predefinito quando ci si collega ad un broker di messaggi locale.

## jms.NumConcurrentRequests

La proprietà `javax.jms.NumConcurrentRequests` specifica il numero massimo di richieste di chiamate di servizi simultanei che possono essere inviati ad un connettore contemporaneamente. Una volta raggiunto il massimo numero, le nuove chiamate del servizio vengono bloccate e occorre attendere un'altra richiesta per il completamento prima di proseguire.

Il valore predefinito è 10.

## jms.Password

La proprietà `javax.jms.Password` specifica la password per il provider JMS. Un valore per questa proprietà è opzionale.

Non esiste alcun valore predefinito.

## jms.TransportOptimized

La proprietà `javax.jms.TransportOptimized` determina se il WIP (work in progress) è ottimizzato. Occorre avere un provider WebSphere MQ provider per ottimizzare il WIP. Affinché WIP ottimizzati siano operativi, il provider dei messaggi deve essere in grado di:

1. Leggere un messaggio senza eliminarlo dalla coda
2. Eliminare un messaggio con un ID specifico senza trasferire l'intero messaggio allo spazio di memoria del ricevitore.
3. Leggere un messaggio utilizzando un ID specifico (necessario per scopi di recupero)
4. Tenere traccia del punto in cui appaiono gli eventi che non sono stati letti.

Le API JMS non possono essere utilizzate per WIP ottimizzato poiché non soddisfano le condizioni 2 e 4 su specificate, mentre le API Java MQ soddisfano tutte e quattro le condizioni e pertanto sono richieste per il WIP ottimizzato.

Questa proprietà è valida solo se il valore di `DeliveryTransport` è JMS e il valore di `BrokerType` è ICS.

Il valore predefinito è false.

## **jms.UserName**

La proprietà `jms.UserName` specifica il nome utente per il provider JMS. Un valore per questa proprietà è opzionale.

Non esiste alcun valore predefinito.

## **JvmMaxHeapSize**

La proprietà `JvmMaxHeapSize` specifica la massima dimensione di memoria heap per l'agente (in megabyte).

Questa proprietà è valida solo se il valore per la proprietà `RepositoryDirectory` è impostato su `<REMOTE>`.

Il valore predefinito è 128 m.

## **JvmMaxNativeStackSize**

La proprietà `JvmMaxNativeStackSize` specifica la massima dimensione di memoria originaria per l'agente (in kilobyte).

Questa proprietà è valida solo se il valore per la proprietà `RepositoryDirectory` è impostato su `<REMOTE>`.

Il valore predefinito è 128 k.

## **JvmMinHeapSize**

La proprietà `JvmMinHeapSize` specifica la dimensione di memoria heap minima per l'agente (in megabyte).

Questa proprietà è valida solo se il valore per la proprietà `RepositoryDirectory` è impostato su `<REMOTE>`.

Il valore predefinito è 1 m.

## **ListenerConcurrency**

La proprietà `ListenerConcurrency` supporta multithreading in WebSphere MQ Listener quando ICS è il broker di integrazione. Consente la scrittura in batch di più eventi sul database, migliorando le prestazioni del sistema.

Questa proprietà è valida solo con connettori che utilizzano MQ. Il valore della proprietà `DeliveryTransport` deve essere MQ.

Il valore predefinito è 1.

## **Locale**

La proprietà `Locale` specifica il codice della lingua, la nazione o il territorio e, facoltativamente, l'insieme del codice di caratteri associato. Il valore di questa proprietà determina le convenzioni culturali così come la trasformazione e l'ordinamento dei dati, i formati della data e dell'ora e i simboli utilizzati nelle specifiche monetarie.

Un nome locale dispone del seguente formato:

*ll\_TT.codeset*

dove:

*ll* è un codice della lingua a due caratteri (in lettere minuscole).

*TT* è una nazione a due caratteri o un codice territoriale (lettere maiuscole).

*codeset* è il nome dell'insieme di codici di caratteri associato (potrebbe essere facoltativo).

Per impostazione predefinita, viene elencato solo un sottoinsieme di locale supportate. Per aggiungere altri valori supportati all'elenco, modificare il file `\Data\Std\stdConnProps.xml` file nella directory `<ProductDir>\bin`. Per ulteriori informazioni, fare riferimento all'appendice di Connector Configurator Express in questa guida.

Se il connettore non è stato internazionalizzato, il solo valore valido per questa proprietà è `en_US`. Per determinare se un connettore specifico è stato globalizzato, fare riferimento alla guida per l'utente per questo adattatore.

Il valore predefinito è `en_US`.

## LogAtInterchangeEnd

La proprietà `LogAtInterchangeEnd` specifica se registrare gli errori nella destinazione del log del broker di integrazione.

La registrazione della destinazione del log inoltre attiva la notifica delle email, che genera messaggi di email per il destinatario specificato come valore di `MESSAGE_RECIPIENT` nel file `InterchangeSystem.cfg` quando si verificano errori o errori irreversibili. Ad esempio, quando il connettore perde la connessione con l'applicazione, se il valore di `LogAtInterChangeEnd` è `true`, viene inviato un messaggio di email al destinatario del messaggio specificato.

Questa proprietà è valida solo se il valore di `RespositoryDirectory` è impostato su `<REMOTE>` (il valore della proprietà `BrokerType` è `ICS`).

Il valore predefinito è `false`.

## MaxEventCapacity

La proprietà `MaxEventCapacity` specifica il numero massimo di eventi nel buffer del controller. Questa proprietà viene utilizzata dalla funzione di controllo di flusso.

Questa proprietà è valida solo se il valore di `RespositoryDirectory` è impostato su `<REMOTE>` (il valore della proprietà `BrokerType` è `ICS`).

Il valore può essere un numero intero positivo tra 1 e 2147483647.

Il valore predefinito è 2147483647.

## MessageFileName

La proprietà `MessageFileName` specifica il nome del file di messaggi del connettore. La posizione standard per il file di messaggi è `\connectors\messages` nella directory del prodotto. Specifica il nome del file di messaggi in un percorso assoluto se il file di messaggi non è localizzato nella posizione standard.

Se il file di messaggio del connettore non esiste, il connettore utilizza `InterchangeSystem.txt` come file di messaggio. Tale file si trova nella directory del prodotto.

**Nota:** Per determinare se un connettore ha il proprio file di messaggi, consultare la guida per l'utente dell'adattatore.

Il valore predefinito è `InterchangeSystem.txt`.

## MonitorQueue

La proprietà `MonitorQueue` specifica la coda logica utilizzata dal connettore per controllare eventi duplicati.

È valida solo se il valore della proprietà `DeliveryTransport` è `JMS` e il valore di `DuplicateEventElimination` è `true`.

Il valore predefinito è `<CONNECTORNAME>/MONITORQUEUE`

## OADAutoRestartAgent

La proprietà `OADAutoRestartAgent` specifica se il connettore utilizza la funzione di riavvio automatico o remoto. Questa funzione utilizza WebSphere OAD (Object Activation Daemon) con trigger MQ per riavviare il connettore dopo una chiusura anormale -triggered oppure per avviare un connettore remoto da System Monitor.

Questa proprietà deve essere impostata su `true` per abilitare la funzione di riavvio automatico e remoto. Per informazioni su come configurare la funzione OAD con trigger MQ di WebSphere consultare *WebSphere Business Integration Server Express - Guida all'installazione per Windows, per Linux oppure per OS/400*.

Questa proprietà è valida solo se il valore di `RespositoryDirectory` è impostato su `<REMOTE>` (il valore della proprietà `BrokerType` è `ICS`).

Il valore predefinito è `false`.

## OADMaxNumRetry

La proprietà `OADMaxNumRetry` specifica il numero massimo di volte in cui il OAD (Object Activation Daemon) con trigger MQ di WebSphere automaticamente cerchi di riavviare il connettore dopo una chiusura anormale. La proprietà `OADAutoRestartAgent` deve essere impostata su `true` perché abbia effetto.

Questa proprietà è valida solo se il valore di `RespositoryDirectory` è impostato su `<REMOTE>` (il valore della proprietà `BrokerType` è `ICS`).

Il valore predefinito è `1000`.

## OADRetryTimeInterval

La proprietà `OADRetryTimeInterval` specifica il numero di minuti nell'intervallo di tentativi per OAD con trigger MQ di WebSphere. Se l'agente del connettore non si riavvia all'interno dell'intervallo di tentativi, il controller del connettore chiede a OAD di riavviare di nuovo l'agente del connettore. L'OAD ripete questo processo di tentativi tante volte quante specificate dalla proprietà `OADMaxNumRetry`. La proprietà `OADAutoRestartAgent` deve essere impostata su `true` perché abbia effetto.

Questa proprietà è valida solo se il valore di RespositoryDirectory è impostato su <REMOTE> (il valore della proprietà BrokerType è ICS).

Il valore predefinito è 10.

## PollEndTime

La proprietà PollEndTime specifica quando arrestare il polling della coda di eventi. Il formato è HH:MM , dove HH è un'ora da 0 a 23 e MM da 0 a 59 minuti.

Occorre fornire un valore valido per questa proprietà. Il valore predefinito è HH:MM senza un valore e deve essere modificato.

Se il runtime dell'adattatore rileva che:

- PollStartTime è impostato e PollEndTime non lo è, oppure
- PollEndTime è impostato e PollStartTime non lo è

effettuerà il poll utilizzando il valore configurato per la proprietà PollFrequency.

## PollFrequency

La proprietà PollFrequency specifica la quantità di tempo (in millisecondi) tra la fine dell'azione di polling e l'avvio di quella successiva. Non si tratta di un intervallo tra azioni di polling; al contrario la logica è la seguente.

- Poll per ottenere il numero di oggetti specificati dal valore della proprietà PollQuantity.
- Elabora questi oggetti. Per alcuni connettori, è possibile che sia eseguito su thread separati, che eseguono in modo asincrono fino alla successiva azione di polling.
- Il ritardo dell'intervallo specificato dalla proprietà PollFrequency.
- Ripetere tutte le operazioni

I seguenti valori sono validi per questa proprietà:

- Il numero di millisecondi tra le azioni di polling (un numero intero positivo).
- La parola no, che impedisce al connettore di eseguire il poll. Immettere la parola in lettere minuscole.
- La parola key, impedisce al connettore di eseguire il poll solo quando si digita la lettera p nella finestra Riga comandi del connettore. Immettere la parola in lettere minuscole.

Il valore predefinito è 10000.

**Importante:** Alcuni connettori possiedono delle limitazioni all'utilizzo di questa proprietà. Quando esistono, tali limitazioni vengono documentate nel capitolo sull'installazione e configurazione dell'adattatore.

## PollQuantity

La proprietà PollQuantity designa il numero di voci provenienti dall'applicazione per le quali il connettore esegue il poll. Se l'adattatore ha una proprietà specifica del connettore per l'impostazione della quantità di poll, l'impostazione del valore nella proprietà specifica del connettore sostituisce il valore della proprietà standard.

Questa proprietà è valida solo se il valore della proprietà DeliveryTransport è JMS e la proprietà ContainerManagedEvents ha un valore.

Anche un messaggio di email viene considerato un evento. Le azioni del connettore sono le seguenti quando viene effettuato il poll per una email.

- Quando viene eseguito il poll una volta, il connettore rileva il corpo del messaggio e lo legge come un allegato. Poiché non è stato specificato alcun data handler per questo tipo mine, ignorerà il messaggio.
- Il connettore elabora il primo allegato BO. Data handler è disponibile per questo tipo MIME, pertanto invia l'oggetto di business al Visual Test Connector.
- Quando viene eseguito il poll per una seconda volta, il connettore elabora il secondo allegato BO. Data handler è disponibile per questo tipo MIME, pertanto invia l'oggetto di business al Visual Test Connector.
- Una volta accettato, il terzo allegato BO deve essere trasmesso.

## PollStartTime

La proprietà PollStartTime specifica quando arrestare il polling della coda di eventi. Il formato è *HH:MM*, dove *HH* è un'ora da 0 a 23 e *MM* da 0 a 59 minuti.

Occorre fornire un valore valido per questa proprietà. Il valore predefinito è *HH:MM* senza un valore e deve essere modificato.

Se il runtime dell'adattatore rileva che:

- PollStartTime è impostato e PollEndTime non lo è, oppure
- PollEndTime è impostato e PollStartTime non lo è

effettuerà il poll utilizzando il valore configurato per la proprietà PollFrequency.

## RepositoryDirectory

La proprietà RepositoryDirectory è l'ubicazione del repository dal quale il connettore legge i documenti dello schema XML che memorizzano i metadati per le definizioni degli oggetti di business.

Se il broker di integrazione è ICS, questo valore deve essere impostato su `<REMOTE>` poiché il connettore ottiene queste informazioni dal repository di InterChange Server Express.

Quando il broker di integrazione è un broker di messaggi WebSphere oppure WAS, questo valore viene impostato su `<ProductDir>\repository` per impostazione predefinita. Tuttavia, è possibile che venga impostato su un altro nome di directory valida.

## RequestQueue

La proprietà RequestQueue specifica la coda utilizzata dal broker di integrazione per inviare messaggi di business al connettore.

Questa proprietà è valida solo se il valore della proprietà DeliveryTransport è JMS

Il valore predefinito è `<CONNECTORNAME>/REQUESTQUEUE`

## ResponseQueue

La proprietà ResponseQueue specifica la coda di risposta JMS, che fornisce un messaggio di risposta dalla struttura del connettore al broker di integrazione.

Quando il broker di integrazione è ICS (InterChange Server Express), il server invia la richiesta ed attende un messaggio di risposta nella coda di risposta JMS.

Questa proprietà è valida solo se il valore della proprietà DeliveryTransport è JMS

Il valore predefinito è <CONNECTORNAME>/RESPONSEQUEUE

## RestartRetryCount

La proprietà RestartRetryCount specifica il numero di volta che il connettore cerca di riavviarsi. Quando questa proprietà viene utilizzata per un connettore connesso in parallelo, specifica il numero di volte che il componente specifico dell'applicazione del connettore principale cerca di riavviare il componente specifico dell'applicazione del connettore del client.

Il valore predefinito è 3.

## RestartRetryInterval

La proprietà RestartRetryInterval specifica l'intervallo in minuti in cui il connettore cerca di riavviarsi. Quando questa proprietà viene utilizzata per un connettore connesso in parallelo, specifica l'intervallo in cui il componente specifico dell'applicazione del connettore principale cerca di riavviare il componente specifico dell'applicazione del connettore del client.

I valori possibili per la proprietà sono compresi tra 1 e 2147483647.

Il valore predefinito è 1.

## ResultsSetEnabled

La proprietà ResultsSetEnabled abilita o disabilita il supporto dell'insieme di risultati quando Information Integrator è attivo. Questa proprietà può essere utilizzata solo se l'adattatore supporta DB2 Information Integrator.

Il valore predefinito è false.

## ResultsSetSize

La proprietà ResultsSetSize definisce il numero massimo di oggetti di business che possono essere restituiti a Information Integrator. Questa proprietà può essere utilizzata solo se l'adattatore supporta DB2 Information Integrator.

Questa proprietà è valida solo se il valore della proprietà ResultsSetEnabled è true

Il valore predefinito è 0. Ciò significa che la dimensione dell'insieme di risultati è illimitato.

## RHF2MessageDomain

La proprietà RHF2MessageDomain consente di configurare il valore del nome del dominio del campo nell'intestazione JMS. Quando i dati vengono inviati al broker di messaggi WebSphere su trasporto JMS, la struttura dell'adattatore scrive le informazioni dell'intestazione JMS con un nome di dominio e un valore fisso pari a mrm. Un nome dominio configurabile consente di tenere traccia di come il broker di messaggi WebSphere elabora i dati dei messaggi.

Questa è una intestazione di esempio:

```
<mcd><Msd>mrm</Msd><Set>3</Set><Type>  
Retek_POPhyDesc</Type><Fmt>CwXML</Fmt></mcd>
```

Questa proprietà non è valida se il valore di `BrokerType` è `ICS`. Inoltre, è valida solo se il valore della proprietà `DeliveryTransport` è `JMS` e il valore di `WireFormat` è `CwXML`.

I valori possibili sono `mrm` e `xml`. Il valore predefinito è `mrm`.

## SourceQueue

La proprietà `SourceQueue` designa una coda origine `JMS` per la struttura del connettore a supporto della consegna garantita degli eventi per connettori abilitati `JMS` che utilizzano la memoria eventi `JMS`. Per ulteriori informazioni, consultare "ContainerManagedEvents" a pagina 113.

Questa proprietà è valida solo se il valore di `DeliveryTransport` è `JMS` e il valore di `ContainerManagedEvents` è specificato.

Il valore predefinito è `<CONNECTORNAME>/SOURCEQUEUE`

## SynchronousRequestQueue

La proprietà `SynchronousRequestQueue` fornisce messaggi di richiesta che necessitano di una risposta sincrona dalla struttura del connettore al broker. Questa coda è necessaria solo se il connettore utilizza esecuzioni sincrone. Con l'esecuzione sincrona, la struttura del connettore invia i messaggi alla coda di richieste sincrone ed attende le risposte da un broker della coda di risposte sincrone. Il messaggio di risposta inviato al connettore possiede un'ID di correlazione che corrisponde all'ID del messaggio originale.

Questa proprietà è valida solo se il valore di `DeliveryTransport` è `JMS`

Il valore predefinito è `<CONNECTORNAME>/SYNCHRONOUSREQUESTQUEUE`

## SynchronousRequestTimeout

La proprietà `SynchronousRequestTimeout` specifica il tempo in millisecondi che il connettore attende per una risposta da una richiesta sincrona. Se non viene ricevuta una risposta all'interno del tempo specificato, il connettore si sposta il messaggio di richiesta sincrona originale (e messaggio di errore) alla coda di errore.

Questa proprietà è valida solo se il valore di `DeliveryTransport` è `JMS`

Il valore predefinito è `0`.

## SynchronousResponseQueue

La proprietà `SynchronousResponseQueue` fornisce messaggi in risposta a richieste sincrone dal broker alla struttura del connettore. Questa coda è necessaria solo se il connettore utilizza esecuzioni sincrone.

Questa proprietà è valida solo se il valore di `DeliveryTransport` è `JMS`

Il valore predefinito è `<CONNECTORNAME>/SYNCHRONOUSRESPONSEQUEUE`

## TivoliMonitorTransactionPerformance

La proprietà `TivoliMonitorTransactionPerformance` specifica se `IBM Tivoli Monitoring for Transaction Performance (ITMTP)` viene richiamato al momento del runtime.



Il valore predefinito è false.

## **WireFormat**

La proprietà WireFormat specifica il formato del messaggi sul trasporto:

- Se il valore della proprietà RepositoryDirectory è una directory locale, il valore è CwXML.
- Se il valore della proprietà RepositoryDirectory è una directory locale, il valore è CwB0.



## Appendice B. Proprietà di configurazione specifiche dei connettori

Proprietà di configurazione specifiche del connettore fornisce le informazioni necessarie per il connettore al runtime. Le proprietà specifiche dei connettori inoltre forniscono un modo di modificare le informazioni statiche o logiche all'interno del connettore senza doverlo ricodificare o ricreare.

Tabella 16 elenca le proprietà di configurazione specifiche del connettore. Fare riferimento alle sezioni seguenti per ulteriori chiarimenti sulle proprietà.

Tabella 16. Proprietà di configurazione specifiche del connettore

Nome	Valori possibili	Valore predefinito	Richiesto
ApplicationPassword	<i>Password per l'account utente del connettore</i>		Yes*
ApplicationUserName	<i>Nome dell'account utente del connettore</i>		Yes*
ArchiveProcessed	true oppure false	Vero	No
ArchiveTableName	<i>Nome della tabella di coda dell'archivio</i>	xworlds_archive_events	Yes se l'archivio elaborato è true
AutoCommit	true oppure false	Falso	No
CheckforEventTableInInit	true oppure false	Vero	No
ChildUpdatePhyDelete	true oppure false	Falso	No
CloseDBConnection	true oppure false	Falso	No
ConnectorID	<i>ID univoci per il connettore</i>	null	No
DatabaseURL	<i>Nome del server di database</i>		Si
DateFormat	<i>Una stringa di esempio dell'ora</i>	MM/gg/aaaa HH:mm:ss	No
DisableLocaleFormatting	true o false	false	No
DisableNumberFormtGrouping	true o false	false	No
DriverConnectionProperties	<i>Proprietà di connessione del driver JDBC aggiuntive</i>		No
DriverSupportForLong	true oppure false	Vero	No
EventKeyDel	<i>Carattere/i delimitatori per la colonna della chiave dell'oggetto per la tabella eventi</i>	punto e virgola (;)	No
EventOrderBy	nessuno, <i>ColumnName</i> , <i>ColumnName</i> , ...]		No
EventQueryType	Fisso o Dinamico	Fisso	No
EventTableName	<i>Nome della tabella di coda degli eventi</i>	xworlds_events	Si, se è richiesto il polling null (come stringa) se il polling non è richiesto
JDBCClass	nome classe del driver		Si
MaximumDatabaseConnections	<i>Numero di connessioni al database simultanee</i>	5	Si
PingQuery	SELEZIONARE 1 DA <nome tabella>		No
PollQuantity	I valori sono compresi tra 1 e 500	1	No

Tabella 16. Proprietà di configurazione specifiche del connettore (Continua)

Nome	Valori possibili	Valore predefinito	Richiesto
PreparedStatementCacheSize	Numero massimo di istruzioni preparate per la memorizzazione nella cache per ciascuna connessione nel pool di connessione JDBC	40	No
PreserveUIDSeq	true oppure false	Vero	No
QueryTimeOut	Valore intero in secondi		No
RDBMS.initsession	L'istruzione SQL che inizializza ogni sessione di database		No
RDBMSVendor	DB2		Sì
ReplaceAllStr	true oppure false	Falso	No
ReplaceStrList	Una serie composta di singoli caratteri, un delimitatore di caratteri e una stringa di sostituzione del carattere. Inoltre, più serie con una delimitatore di terminazione.	Q,DSQ Nota: nello strumento di configurazione del connettore, questi caratteri rappresentano un apice seguito da una virgola, seguito da due singoli apici.	No
RetryCountAndInterval	Conteggio, intervallo in secondi	3,20	No
ReturnDummyBOForSP	true oppure false	Falso	No
SchemaName	Schema in cui si trovano gli eventi		No
SelectiveCommitForPoll	true oppure false	Falso	No
SPBeforePollCall	None della procedura memorizzata da eseguire per ciascun richiamo di polling		No
StrDelimiter	Il delimitatori di carattere e terminazione utilizzati nella proprietà ReplaceStrList.	,:	No
TimingStats	0, 1, 2	0	No
UniqueIDTableName	Nome della tabella utilizzata per gli ID di generazione	xworlds_uid	No
UseDefaults	true oppure false	Falso	Sì
UseDefaultsForCreatingChildBOs	true oppure false	Falso	No
UseDefaultsWhenPolling	true oppure false	Vero	No

\*ApplicationPassword e ApplicationUserNames non sono richiesti se si utilizza l'autenticazione autorizzata. Per i5/OS, quando si accede ai database i5/OS, il nome utente e la password possono essere impostati su vuoto. Quindi, i5/OS utilizza il profilo utente eseguito da Adapter Job per accedere all'applicazione. Il nome utente predefinito può essere modificato cambiando l'oggetto della descrizione del lavoro QWBIJDBCC nella libreria QWBISVR44.

## Dettagli proprietà

Di seguito vengono descritte le proprietà di configurazione specifiche del connettore.

### ApplicationPassword

Password per l'accout utente del connettore.

Non esiste alcun valore predefinito.

## ApplicationUserName

Nome dell'account utente del connettore.

Non esiste alcun valore predefinito.

## ArchiveProcessed

Specifica se il connettore archivia gli eventi per cui non sono presenti sottoscrizioni correnti.

Impostare questa proprietà su `true` affinché gli eventi siano inseriti nella tabella di archivio prima di essere eliminati dalla tabella eventi.

Impostare questa proprietà su `false` affinché il connettore non esegua l'elaborazione dell'archivio. In questo caso, non verifica il valore della proprietà `ArchiveTableName`. Se `ArchiveProcessed` viene impostato su `false`, il connettore si comporta nel modo seguente:

- Se l'evento non viene elaborato correttamente, il connettore lo elimina dalla tabella eventi e non lo archivia.
- Se il connettore non sottoscrive l'oggetto business dell'evento, il connettore lascia l'evento nella tabella e modifica lo stato degli eventi su `Non sottoscritto`. Per informazioni specifiche sul broker d'integrazione, consultare la sezione *Guida all'implementazione del sistema*.
- Se l'oggetto business incontra un problema durante l'elaborazione, il connettore lascia l'evento nella tabella eventi con lo stato eventi di `Errore`.

Se questa proprietà viene impostata su `false` e la quantità di polling è bassa, sembra che il connettore esegua il polling della tabella eventi, ma sta semplicemente raccogliendo ripetutamente gli stessi eventi.

Se questa proprietà non dispone di alcun valore, il connettore presume che il valore sia `true`. Se anche la proprietà `ArchiveTableName` non dispone di un valore, il connettore presume che il nome dell'archivio sia `xworlds_archive_events`.

Il valore predefinito è `true`.

## ArchiveTableName

Nome della tabella di coda dell'archivio.

Se la proprietà `ArchiveProcessed` viene impostata su `false`, non è necessario impostare un valore per questa proprietà.

Questa proprietà è abilitata per utilizzare le lingue bidirezionali.

Il nome predefinito è `xworlds_archive_events`.

## AutoCommit

Questa proprietà rende configurabile l'impostazione `AutoCommit`. Quando impostata su `true`, viene automaticamente eseguito il commit di tutte le transazioni. Alcuni database (come Sybase) richiedono `AutoCommit` per essere impostati su `true`. Se si impostato su `false`, le procedure memorizzate su Sybase non riescono.

Se la connessione al database viene perduta, il connettore tenta di creare una nuova connessione per riavviare l'elaborazione completa finché AutoCommit è impostato su false. Se la nuova connessione non è valida o se AutoCommit è impostato su true, il connettore restituisce APPRESPONSETIMEOUT, che termina il connettore.

Il valore predefinito è false.

## CheckforEventTableInInit

Impostando la proprietà del connettore su false si impedisce che quest'ultimo verifichi l'esistenza delle tabelle di archivio e di eventi durante l'inizializzazione del connettore. Si consiglia di impostare sempre su true se il driver JDBC che si utilizza non supporta la classe JDBC DatabaseMetaData.

Quando la proprietà è impostata su false, sebbene il connettore non verifichi l'esistenza di EventTable e ArchiveTable, le tabelle di archivio e di eventi dovrebbero esistere sempre, poiché il connettore le utilizza durante il processo di inizializzazione. Per impedire al connettore di utilizzare le tabelle di archivio e di eventi durante l'inizializzazione, impostare la proprietà EventTableName su null.

Il valore predefinito è true.

## ChildUpdatePhyDelete

Durante un'operazione di aggiornamento, specifica il modo in cui il connettore gestisce i dati rappresentati da un oggetto business child che manca dall'oggetto business in entrata ma esiste nel database.

Impostare questa proprietà su true affinché il connettore elimini fisicamente la registrazione dati dal database.

Impostare questa proprietà su false affinché il connettore elimini logicamente il record dati dal database impostando la colonna di stato al valore appropriato. Le informazioni specifiche dell'applicazione ottengono il nome della colonna di stato ed il relativo valore dal parametro StatusColumnValue (SCN) specificato nelle informazioni specifiche dell'applicazione del a livello dell'oggetto business. Per ulteriori informazioni, consultare "Informazioni specifiche dell'applicazione a livello di oggetto business" a pagina 58.

Il valore predefinito è false.

## CloseDBConnection

Questa proprietà rende configurabile la chiusura della connessione al database. Quando impostata su true, per ciascuna richiesta di chiamata di servizio e chiamata di polling, la connessione al database viene chiusa. Impostando questa proprietà su true rende le prestazioni non stabili.

Il valore predefinito è false.

## ConnectorID

Un ID univoco per il connettore. Questo ID è utile per richiamare gli eventi di una determinata istanza del connettore.

Il valore predefinito è null.

## DatabaseURL

Il nome del server di database cui il connettore dovrebbe connettersi.

Se si utilizza il driver WebSphere Business Integration Server Express system branded SQLServer, l'URL consigliato è:

```
jdbc:ibm-crossworlds:sqlserver://MachineName:PortNumber;DatabaseName=DBname
```

### Importante

Se AutoCommit è impostato su false, è necessario impostare un parametro aggiuntivo SelectMethod: jdbc:ibm-crossworlds:sqlserver://MachineName:PortNumber;DatabaseName=DBname;SelectMethod=cursor

Per impostazione predefinita, SelectMethod è impostato su direct. Per ulteriori informazioni, consultare "AutoCommit" a pagina 129.

E' necessario fornire un valore a tale proprietà per l'esecuzione corretta del connettore.

Questa proprietà potrebbe essere abilitata per l'utilizzo con lingue bidirezionali, in base al database RDBMS utilizzato. Se il database supporta lo script bidirezionale, l'adattatore per JDBC trasforma solo il componente del nome del database di questa proprietà.

## DateFormat

Specifica il formato data che il connettore prevede di ricevere una restituzione. Questa proprietà supporta qualunque formato che si basa sulla sintassi indicata in Tabella 17 a pagina 132.

Tabella 17 a pagina 132 definisce la sintassi DateFormat utilizzando una stringa di modello di tempo. In questo modello, tutte le lettere ASCII sono riservate come lettere di modello.

Tabella 17. Sintassi DateFormat

Simbolo	Significato	Presentazione	Esempio
G	era designator	(Testo)	AD
a	anno	(Numero)	1996
M	mese dell'anno	(Testo e numero)	7 luglio
g	giorno del mese	(Numero)	10
h	ore am/pm(1-12)	(Numero)	12
H	ora del giorno (0-23)	(Numero)	0
m	minuti	(Numero)	30
s	secondi	(Numero)	55
S	millisecondi	(Numero)	978
E	giorno della settimana	(Testo)	Martedì
G	giorno dell'anno	(Numero)	189
F	giorno della settimana nel mese	(Numero)	2 (2° mercoledì di luglio)
s	settimana dell'anno	(Numero)	27
W	settimana del mese	(Numero)	2
a	indicatore am/pm	(Testo)	PM
k	ore del giorno (1-24)	(Numero)	24
K	ora in am/pm(0-11)	(Numero)	0
z	fuso orario	(Testo)	Pacifico Standard Ora
'	esci dal testo	(Delimitatore)	
''	apici	(Letterale)	'

Tabella 18. Esempi utilizzando le impostazioni internazionali per gli Stati Uniti

Modello formato	Risultato
"aaaa.MM.gg G 'a' hh:mm:ss z"	1996.07.10 AD a 15:08:56 PDT
"EEE, MMM g, 'yy"	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
"yyyy.MMMM.dd GGG hh:mm aaa"	1996.July.10 AD 12:08 PM

## DisableLocaleFormatting

Impostando questo valore su true in combinazione con l'impostazione DisableNumberFormatGrouping su true, l'adattatore gestisce correttamente i numeri di sequenza superiori a 999. Quando il valore viene impostato su false, l'adattatore può posizionare le virgole nei numeri richiamati dal database e può verificarsi un errore NumberFormatException.

Il valore predefinito è false.

## DisableNumberFormatGrouping

Quando impostato su true, questo valore sovrascrive tutte le limitazioni di formattazione. Per evitare le eccezioni relative al formato del numero quando viene utilizzata la sequenza del tipo CrossWorlds UID, impostare questo valore su true.

Il valore predefinito è false.



## DriverConnectionProperties

Oltre al nome utente e alla password, un driver JDBC potrebbe richiedere ulteriori proprietà o informazioni. La proprietà del connettore `DriverConnectionProperties` ottiene ulteriori proprietà richieste da un driver JDBC, come ad esempio le coppie nome-valore. Le proprietà dovrebbero essere specificate come segue:

```
property1=value1[:property2=value2...]
```

Le proprietà devono essere date come coppie di valore, separate da punti e virgola. La proprietà è separata dal valore da un segno di uguale (senza altri spazi).

Ad esempio, si supponga che il driver JDBC necessiti delle informazioni sulla licenza e del numero di porta. Il nome della proprietà previsto per le informazioni sulla licenza è `MyLicense` e il valore è `ab23jk5`. Il nome della proprietà prevista per il numero di porta è `PortNumber` e il valore è `1200`. `DriverConnectionProperties` dovrebbe essere impostato al valore `MyLicense=ab23jk5;PortNumber=1200`.

## DriverSupportForLong

Specifica il modo in cui i parametri interi vengono gestiti da `PreparedStatement`. Quando questa proprietà è `true`, viene utilizzato `setLong` per impostare il parametro intero. Quando è `false`, viene utilizzato `setInt` per impostare il parametro.

Il valore predefinito è `true`.

## EventKeyDel

Specifica il delimitatore quando la colonna `object_key` della tabella eventi contiene più valori dell'attributo.

Esistono due modi per richiamare l'oggetto business che è stato creato, aggiornato o eliminato durante il trigger dell'applicazione.

- Il primo è di popolare la colonna `object_key` con i valori degli attributi che sono chiavi in un oggetto business. Impostare la proprietà di configurazione `EventKeyDel` su un carattere unico che non è parte del campo della chiave. Ad **esempio**, se il delimitatore è specificato come `“;”`, `object_key` sarà: `xxx;123`
- Il secondo modo consiste nel popolare la colonna `object_key` con i valori per qualunque attributo in un oggetto business. Questi valori dovrebbero essere rappresentati come coppia `name_value`. Il primo delimitatore sarà per `name_value` e il secondo è per le chiavi. Ad esempio, se il delimitatore è specificato come `“=”`, `object_key` sarà: `CustomerName=xxx;CustomerId=123;`

Se il delimitatore è specificato come `“=”`, `object_key` sarà:  
`CustomerName=xxx:CustomerId=123:`

**Nota:** L'ordine in cui i valori chiave sono definiti dovrebbe seguire lo stesso ordine come attributi chiave nell'oggetto business.

**Importante:** Se si utilizza i dati dell'attributo `Data`, non utilizzare il delimitatore con il simbolo dei due punti (`:`), poiché potrebbe essere comprese nei dati dell'attributo.

Il valore predefinito è il punto e virgola (`;`), che si basa sulle chiavi, non sulle coppie `name_value`.

## EventOrderBy

Specifica se disattivare l'ordine degli eventi o specifica un ordine di elaborazione eventi diverso da quello predefinito.

Per impostazione predefinita, a ciascun polling, il connettore ottiene solo il numero degli eventi specificato nella relativa `PollQuantity` proprietà e ordina l'elaborazione degli eventi con i valori nelle colonne `event_time` e `event_priority` della tabella `Eventi`.

Affinché il connettore non ordini gli eventi, impostare il valore di questa proprietà su `none`.

Affinché il connettore ordini per colonne diverse nella tabella eventi, specificare i nomi di tali colonne. Separare i nomi delle colonne con una virgola (,). Se si specifica un valore per questa proprietà, il comportamento predefinito viene sovrascritto.

Questa proprietà è abilitata per utilizzare le lingue bidirezionali.

Per questa proprietà non è presente alcun valore predefinito.

## EventQueryType

La proprietà `EventQueryType` viene utilizzata per indicare se il connettore deve generare in modo dinamico una query per richiamare gli eventi dalla tabella eventi o utilizzare la relativa query integrata. Per le query dinamicamente generate, il connettore mappa la relativa struttura eventi alle colonne nella tabella eventi. L'ordine dei dati nelle colonne della tabella è molto importante. Fare riferimento a "Schemi di tabelle di archivio e di eventi" a pagina 89 per visualizzare l'ordine corretto.

Se il valore in `EventQueryType` è `Fixed` (come una stringa), la query predefinita. Se il valore è impostato su `Dynamic` (come una stringa), una nuova query viene creata ottenendo i nomi della colonna dalla tabella specificata nella proprietà "`EventTableName`".

I nomi della colonna nella tabella eventi possono essere modificati, ma l'ordine e il tipo di dati delle colonne devono rimanere uguali a quelli specificati nella sezione della creazione della tabella eventi. "`EventOrderBy`" verrà aggiunto al valore predefinito o alla query generata in modo dinamico.

Se la proprietà `EventQueryType` non viene aggiunta o non contiene alcun valore, il valore predefinito viene impostato su `Fixed`.

Il valore predefinito è `Fixed` (come una stringa).

## EventTableName

Nome della tabella della coda eventi utilizzato dal meccanismo di polling del connettore.

Impostare su `null` (come una stringa) quando il polling viene disattivato per il connettore. In questo modo, viene impedita la convalida dell'esistenza delle tabelle di eventi e di archivio.

Per una tabella eventi definita dall'utente, verificare che event\_id sia mappato su uno dei seguenti tipi JDBC: INTEGER, BIGINT, NUMERIC, VARCHAR.

Questa proprietà è abilitata per utilizzare le lingue bidirezionali.

Il nome predefinito è `xworlds_events`.

## JDBCDriverClass

Specifica il nome della classe di un driver. Per utilizzare un driver JDBC, specificare il nome della classe del driver in questa proprietà di configurazione.

Per ulteriori informazioni consultare "Abilitazione del supporto multi-driver" a pagina 69 e "UseDefaultsForCreatingChildBOs" a pagina 140.

Non viene fornito alcun valore predefinito.

## MaximumDatabaseConnections

Specifica il numero massimo consentito di connessioni del database simultanee. Al runtime, il numero di connessioni del database aperto è la somma di questo valore più 1.

Se la proprietà "PreserveUIDSeq" a pagina 136 è impostata su false, al runtime, il numero di connessioni del database aperto è la somma di questo valore più 2.

Il valore predefinito è 5.

## PingQuery

Specifica l'istruzione SQL o la procedura memorizzata che il connettore esegue per verificare la connettività del database.

Di seguito viene riportato un esempio dell'istruzione SQL utilizzata come query di ping:

```
SELEZIONARE 1 DA <nome tabella>
```

Di seguito viene riportato un esempio del richiamo della procedura memorizzata (sampleSP) utilizzata come query di ping con un database DB2:

```
call sampleSP( )
```

I richiami della procedura memorizzata non possono disporre di parametri di output. Se un parametro di input viene richiesto dal database, il valore di input deve essere specificato come parte della query di ping. Ad esempio:

```
Call checkproc(2)
```

Non esiste alcun valore predefinito. Per ulteriori informazioni consultare "Connessioni al database perdute" a pagina 6 e "Perdita della connessione all'applicazione" a pagina 100.

## PollQuantity

Numero di righe nella tabella di database che il connettore richiama l'intervallo di polling. I valori consentiti sono compresi tra 1 a 500.

Il valore predefinito è 1.

## PreparedStatementCacheSize

Il numero massimo di istruzioni preparate che verranno memorizzate nella cache per ciascuna connessione nel pool di connessione JDBC, se la memorizzazione nella cache è abilitata.

Il valore predefinito è 40.

## PreserveUIDSeq

Specifica se la sequenza dell'ID univoco in entrata viene conservata nella tabella dell'identificativo univoco.

Se impostato su `true`, non viene eseguito il commit dell'ID univoco fino a quando l'oggetto business non viene elaborato correttamente nell'applicazione di destinazione. Tutti gli altri processi che tentano di accedere alla tabella dell'identificativo univoco devono attendere fino a quando non viene eseguito il commit della transazione.

Se impostato su `false`, viene eseguito il commit dell'ID univoco quando l'oggetto business lo richiede. L'elaborazione dell'oggetto business e dell'ID univoco dispone del proprio blocco di transazione (interno al connettore). Ciò è possibile solo se la transazione relativa alla tabella dell'identificativo univoco dispone della propria connessione.

**Nota:** Se la proprietà non viene aggiunta alla configurazione del connettore, il comportamento predefinito è uguale se la proprietà è stata aggiunta e impostata su `true`. Inoltre, se "AutoCommit" a pagina 129 viene impostato su `true`, il connettore esegue lo stesso comportamento come se `PreserveUIDSeq` è impostato su `false`.

Se la proprietà "PreserveUIDSeq" viene impostata su `false`, al runtime il numero delle connessioni di database aperte è la somma di questo valore più 2.

Il valore predefinito è `true`.

## QueryTimeOut

Il valore di questa proprietà è un numero intero in secondi che imposta `QueryTimeOut` per tutti i Richiami sul numero di secondi specificato. Se il valore non viene specificato, significa che non è impostato alcun timeout per la query. Se la query non richiede più del numero di secondi specificato, il database produce un'eccezione SQL che è catturata. Il messaggio associato viene registrato nel file di log.

Non viene fornito alcun valore predefinito.

## RDBMS.initsession

Istruzione SQL che inizializza ogni sessione con il database. Il connettore richiede una query che viene eseguita all'avvio. Non dovrebbe essere restituito un valore per questa query. Il nome della proprietà è richiesto, ma un valore non lo è.

Non esiste alcun valore predefinito.

## RDBMSVendor

Specifica l'RDBMS che il connettore utilizza per l'elaborazione speciale. Impostare il valore di questa proprietà su DB2 o MSSQLServer se si utilizza un database IBM DB2 o Microsoft SQL Server.

Affinché il connettore sia eseguito correttamente, è richiesto un valore.

Non viene fornito alcun valore predefinito.

## ReplaceAllStr

Specifica se il connettore sostituisce tutte le istanze di ciascun carattere identificato nella proprietà ReplaceStrList con la stringa di sostituzione specificata nella proprietà. Il connettore valuta ReplaceAllStr solo se il parametro ESC=[true|false] di ciascuna proprietà AppSpecificInfo dell'attributo non contiene un valore. In altre parole, se è stato specificato il parametro ESC, il relativo valore ha la precedenza sul valore impostato per la proprietà ReplaceAllStr. Affinché il connettore utilizzi il valore ReplaceAllStr, verificare che il parametro ESC non è stato specificato.

Il valore predefinito di ReplaceAllStr è false.

**Nota:** Il parametro ESC e le proprietà ReplaceAllStr e ReplaceStrList forniscono supporto per le funzioni del carattere per l'uscita dal database (ad esempio, per uscire dagli apici singoli).

## ReplaceStrList

Specifica una o più serie di sostituzioni, ciascuna composta da un singolo carattere da sostituire, un delimitatore di carattere e una stringa di sostituzione. Il connettore esegue questa sostituzione su un valore dell'attributo solo se il valore è stato specificato per il parametro ESC=[true|false] della proprietà AppSpecificInfo dell'attributo o per la proprietà ReplaceAllStr del connettore.

**Nota:** Il parametro ESC e le proprietà ReplaceAllStr e ReplaceStrList forniscono supporto per le funzioni del carattere per l'uscita dal database (ad esempio, per uscire dagli apici singoli).

La sintassi per questo attributo è:

```
single_char1,substitution_str1[:single_char2,substitution_str2[:...]]
```

dove:

*single\_char* Un carattere da sostituire.

*substitution\_str* La stringa di sostituzione che il connettore utilizza per sostituire il carattere.

,

Il delimitatore del carattere, che separa il carattere da sostituire dalla stringa che lo sostituisce. Per impostazione predefinita, il delimitatore di carattere è una virgola (.). E' possibile configurare questo delimitatore impostando il primo nella proprietà StrDelimiter.

:

Il delimitatore di terminazione, che separa le serie di sostituzione (ciascuna delle quali è composta dal carattere da sostituire, un delimitatore di carattere e la stringa di sostituzione). Per impostazione

predefinita, il delimitatore di terminazione è costituito da due punti (:). E' possibile configurare questo delimitatore impostando il secondo nella proprietà `StrDelimiter`.

Ad esempio, si supponga che si desidera sostituire un solo segno di percentuale (%) con due segni di percentuale (%%) e una barra (^) con una barra inversa e una barra (\^). Per impostazione predefinita `StrDelimiter` specifica una virgola (,) come delimitatore di carattere e i due punti (:) come delimitatore di terminazione. Se si conservano i delimitatori predefiniti, utilizzare la seguente stringa come valore di `ReplaceStrList`:

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

**Nota:** Una limitazione dello strumento di configurazione del connettore impedisce l'immissione di apici singoli. Pertanto, è necessario rappresentare con un carattere di citazione singolo Q e due singoli apici con i caratteri DSQ. Nell'esempio precedente, se si desidera sostituire un apice singolo (') con due singoli apici (' '), utilizzare la seguente notazione: `Q,DSQ:%,%:^\,^`

Questa proprietà è abilitata per utilizzare le lingue bidirezionali.

## RetryCountAndInterval

Specifica il numero di tentativi e l'intervallo in secondi che il connettore deve utilizzare quando non è possibile bloccare i dati durante l'esecuzione di un'operazione di aggiornamento.

Prima di eseguire un aggiornamento, il connettore blocca le righe relative all'aggiornamento e tenta di richiamare i dati correnti. Se il connettore non può bloccare le righe, tenta di nuovo di ottenere il blocco per il conteggio e l'intervallo specificato in questa proprietà di configurazione. Il connettore va in timeout se il blocco non viene ottenuto all'interno dei valori specificati qui.

Specificare il valore nel formato: conteggio, intervallo in secondi. Ad esempio, un valore `3,20` specifica tre tentativi con un intervallo di 20 secondi.

Il valore predefinito è `3,20`.

## ReturnDummyBOForSP

Questa proprietà viene utilizzata per restituire i parametri di output anche quando la serie di risultati è vuota.

Nel caso di `RetrieveSP`, viene restituita una serie di risultati. Se la serie di risultati è vuota, non vengono creati oggetti business e i parametri di output restituiti dal richiamo della procedura non possono essere richiamati. Tuttavia, se `ReturnDummyBOForSP` è `true`, verrà restituito un oggetto business fittizio con i valori dai parametri di output e input/output popolati negli attributi corrispondenti.

Il valore predefinito è `false`.

## SchemaName

Questa proprietà limita la ricerca per le tabelle di archivio ed eventi all'interno di un determinato schema. Se questa proprietà non viene aggiunta o se è lasciata

vuota, il connettore ricerca tutti gli schemi cui l'utente può accedere. Questo SchemaName è anche utilizzato durante la creazione delle query per accedere alle tabelle di archivio e di eventi.

Per il server Microsoft SQL o DB2, un nome dello schema può fare riferimento ad un nome di appartenenza del database. Tuttavia, per informazioni specifiche, fare riferimento alla documentazione del rispettivo driver JDBC.

**Nota:** I nomi dello schema DB2 sono sensibili al maiuscolo/minuscolo. E' necessario specificare il nome dello schema in lettere maiuscole.

Questa proprietà è abilitata per utilizzare le lingue bidirezionali.

Non viene fornito alcun valore predefinito.

## SelectiveCommitForPoll

Specifica quando si verificano i commit al database. Se true, si verifica solo un commit, una volta elaborato un evento. Se false, si verifica il comportamento standard, ovvero i commit al database avvengono a ciascuno stadio di elaborazione dell'evento.

Il valore predefinito è false.

## SPBeforePollCall

Questa proprietà denomina la procedura memorizzata eseguita per ciascun richiamo di polling. Se la proprietà SPBeforePollCall ha un valore (il nome della procedura memorizzata), all'avvio di ciascun richiamo di polling, il connettore richiama la procedura memorizzata, inoltrando i valori delle proprietà del connettore ConnectorID e PollQuantity. La procedura aggiorna il numero di righe di PollQuantity, l'impostazione della colonna ID connettore su ConnectorID dove stato=0 e ID l'connettore è null. Ciò consente il bilanciamento del carico nel connettore.

**Nota:** Nel caso in cui un richiamo di polling non riesce prematuramente (il database non è attivo o la connessione è stata perduta) l'ID del connettore resta impostato. Ciò potrebbe non consentire la registrazione durante il polling. Pertanto, si consiglia di reimpostare periodicamente l'ID del connettore su null, per tutti i record nella tabella eventi con uno stato di 0.

Questa proprietà è abilitata per utilizzare le lingue bidirezionali.

## StrDelimiter

Specifica i delimitatori di terminazione e di carattere per l'utilizzo nella proprietà ReplaceStrList.

- Il delimitatore del carattere separa il carattere da sostituire dalla stringa che lo sostituisce. Il delimitatore del carattere occupa la prima posizione (a sinistra) dei valori della proprietà e dei valori predefiniti con una virgola (,).
- Il delimitatore di terminazione separa le serie di sostituzione (ciascuna delle quali è composta dal carattere da sostituire, un delimitatore di carattere e la stringa di sostituzione). Il delimitatore di terminazione occupa la seconda posizione (destra) dei valori della proprietà e i valori predefiniti con i due punti (:).

E' possibile specificare il valore per uno o entrambi i delimitatori. In questo caso, non specificare uno spazio o altri caratteri tra i due.

Il valore predefinito è una virgola seguita immediatamente da due punti (, :)

## TimingStats

Consente di periodizzare ciascuna operazione di verbi del connettore per ricercare i problemi. Le impostazioni disponibili sono:

- 0 Nessuna statistica relativa al tempo
- 1 Tempo visualizzato all'inizio e alla fine dell'operazione del verbo per un intero oggetto business gerarchico completo.
- 2 Tempo visualizzato all'inizio e alla fine dell'operazione di ciascun verbo per ciascun singolo oggetto business in un oggetto business gerarchico.

I messaggi relativi al tempo sono messaggi di log piuttosto che messaggi di traccia. Possono essere attivati o disattivati, indipendentemente dai livelli di traccia.

Il valore predefinito è 0.

## UniqueIDTableName

Specifica la tabella contenente i valori più aggiornati utilizzati per la generazione di un ID univoco. Per impostazione predefinita, la tabella dispone di un (id) di colonna. E' possibile personalizzare la tabella per aggiungere una colonna per ciascun attributo che richiede la generazione di un UID.

Questa proprietà è abilitata per utilizzare le lingue bidirezionali.

Il valore predefinito è `xworlds_uid`.

## UseDefaults

Se `UseDefaults` è impostato su `true` o non è impostato, il connettore verifica se viene fornito un valore valido o se un valore predefinito per ciascun attributo dell'oggetto business richiesto. Se viene fornito un valore, l'operazione Crea viene completata correttamente, altrimenti non riesce.

Se `UseDefaults` è impostato su `false`, il connettore verifica solo se viene fornito un valore valido per ciascun attributo dell'oggetto business richiesto, l'operazione Crea non riesce se non viene fornito un valore valido.

Il valore predefinito è `false`.

## UseDefaultsForCreatingChildBOs

Se `UseDefaultsForCreatingChildBOs` è impostato su `true` o non è impostato, il connettore verifica se viene fornito un valore valido o se un valore predefinito per ciascun attributo dell'oggetto business richiesto. Se viene fornito un valore, l'operazione Crea viene completata correttamente, altrimenti non riesce.

Se `UseDefaultsForCreatingChildBOs` è impostato su `false`, il connettore verifica solo se viene fornito un valore valido per ciascun attributo dell'oggetto business richiesto, l'operazione Crea non riesce se non viene fornito un valore valido.

Il valore predefinito è `false`.



## UseDefaultsWhenPolling

Se UseDefaultsWhenPolling è impostato su true, i valori predefiniti verranno impostati nell'oggetto business prima che quest'ultimo sia richiamato dal database e inviato al server.

Se UseDefaultsWhenPolling non è impostato su false, i valori predefiniti verranno impostati nell'oggetto business prima che quest'ultimo sia richiamato dal database e inviato al server.

Il valore predefinito è true.

**Nota:** Questa proprietà di configurazione specifica del connettore sostituisce UseDefaultsForRetrieve.



---

## Appendice C. Esempi di oggetti business

Questa appendice illustra dettagliatamente gli oggetti business di esempio che sono inclusi con il connettore per JDBC. Il connettore JDBC include i seguenti esempi di oggetto business.

- AfterUpdateSPSampleBO.txt
- BeforeCreateSPSampleBO.txt
- BOwithDifferentParameterOrder.txt
- BOwithIOandOPParams.txt
- BOwithFewerSPParamsthanBOAttribs.txt
- CreateSPUpdateSPSampleBO.txt

---

### AfterUpdateSPSampleBO.txt

**Nome attributo:** AfterUpdateSP

Contiene l'oggetto business `Jdbctest_Customer`. Le informazioni specifiche dell'applicazione di questo attributo contengono le seguenti informazioni: `SPN=UpdateAllColumns;IP=fid:CustomerName:CustomerNumber:CustomerDesc`. `UpdateAllColumns` è il nome della procedura memorizzata che utilizza tutti i quattro attributi dell'oggetto business (`fid`, `CustomerName`, `CustomerNumber` e `CustomerDesc`) come parametri di input. Questa procedura memorizzata viene eseguita una volta completata l'operazione `Aggiorna`.

---

### BeforeCreateSPSampleBO.txt

**Nome attributo:** BeforeCreateSP

Contiene l'oggetto business `Jdbctest_Customer`. Le informazioni specifiche dell'applicazione di questo attributo contengono le seguenti informazioni: `SPN=GetCustomerID;OP=fid GetCustomerID` è il nome della procedura memorizzata che utilizza l'attributo dell'oggetto business `fid` come parametro di output (preferibilmente per ottenere un valore di ID da una tabella `MasterID`). Questa procedura memorizzata viene eseguita prima dell'operazione `Crea`.

---

### BOwithDifferentParameterOrder.txt

**Nome attributo:** AfterRetrieveSP

Contiene l'oggetto business `Jdbctest_Address`. Le informazioni specifiche dell'applicazione di questo attributo contengono le seguenti informazioni: `SPN=UpdateAddress;IP=addressid;IP=zipcode:city:street`. `UpdateAddress` è il nome della procedura memorizzata che utilizza tutti gli attributi dell'oggetto business come parametri di input (preferibilmente per aggiornare l'indirizzo in una tabella diversa da `Jdbctest_Address`). Si noti che l'ordine del parametro è diverso da quello dell'attributo dell'oggetto business e che contiene più di un nome: la coppia di valori per i parametri di input. Questa procedura memorizzata viene eseguita una volta completata l'operazione `Recupera`.

---

## BOwithIOandOPParams.txt

**Nome attributo:** RetrieveSP

Contiene l'oggetto business Jdbctest\_Address. Le informazioni specifiche dell'applicazione di questo attributo contengono le seguenti informazioni: SPN=RetrieveAddress;IO=addressid;OP=street:city:zipcode. RetrieveAddress è il nome della procedura memorizzata che utilizza l'attributo dell'oggetto business addressid come parametro di input/output. Inoltre, utilizza i restanti attributi dell'oggetto business come parametri di output: zipcode, city, street. La procedura memorizzata viene eseguita invece dell'operazione Recupera.

---

## BOwithFewerSPPParamsthanBOAttribs.txt

**Nome attributo:** AfterUpdateSP

Contiene l'oggetto business Jdbctest\_Address. Le informazioni specifiche dell'applicazione di questo attributo contengono le seguenti informazioni: SPN=UpdateZipOnly;IP=addressid:zipcode. UpdateZipOnly è il nome della procedura memorizzata che utilizza gli attributi dell'oggetto business addressid e zipcode come parametri di input. Si noti che il numero totale dei parametri della procedura memorizzata è inferiore al numero totale degli attributi degli oggetti business.

---

## CreateSPUpdateSPSampleBO.txt

**Nome attributo:** CreateSP

Contiene l'oggetto business Jdbctest\_Address. Le informazioni specifiche dell'applicazione di questo attributo contengono le seguenti informazioni: SPN=CreateAddress;IP=addressid;IP=street:city:zipcode. CreateAddress è il nome della procedura memorizzata che utilizza tutti i quattro attributi dell'oggetto business come parametri di input. Si noti che contiene più di un nome: la coppia di valori per i parametri di input. La procedura memorizzata viene eseguita invece dell'operazione Crea oggetto. Inoltre, questo oggetto business contiene l'attributo UpdateSP. Contiene il seguente testo: SPN=UpdateCity;IP=addressid:city. UpdateCity è il nome della procedura memorizzata che utilizza addressid e city come parametri di input. Si noti che il numero totale dei parametri della procedura memorizzata è inferiore al numero totale di attributi aziendali. La procedura memorizzata viene eseguita invece dell'operazione Aggiorna.

---

## Appendice D. Supporto per valori vuoti o nulli

In questa appendice vengono fornite informazioni dettagliate su differenti scenari riusciti e non riusciti. Inoltre questa appendice contiene le modifiche funzionali richieste per valori di oggetti business vuoti o nulli.

---

### Scenari riusciti o non riusciti

Se un valore chiave in un oggetto business è vuoto o possiede un valore nullo nel database, creare la clausola where con il tipo "is null" invece del tipo operatore "=".

Accertarsi che gli oggetti business abbiano almeno un attributo chiave che non abbia un valore vuoto.

Il seguente scenario è un oggetto principale con una chiave che ha un valore nullo. Questo scenario non ha esito positivo in queste circostanze.

*Tabella 19. Cliente*

Attributo	Tipo
cid	Intero (chiave)
nome	Stringa
commenti	Stringa

Il seguente scenario è un oggetto principale con due chiavi e una chiave ha un valore nullo. Questo scenario ha esito positivo in queste tre circostanze.

*Tabella 20. Cliente*

Attributo	Tipo
cid	Intero (chiave)
nome	Stringa
commenti	Stringa

Nello scenario due, creare la query di recupero selezionando l'id, il nome e i commenti del cliente, dove cid=1000 e il nome è impostato su nullo.

Il seguente scenario è un oggetto principale con un oggetto secondario in un oggetto contenitore con un riferimento chiave estraneo. Questo scenario non ha esito positivo in queste circostanze.

*Tabella 21. Cliente*

Attributo	Tipo
cid	Intero (chiave)
nome	Stringa (chiave)
commenti	Stringa
Indirizzo	Indirizzo
Aiuto	Intero (chiave) ASI:FK=cid
Acity	Stringa

Tabella 21. Cliente (Continua)

Attributo	Tipo
Azip	Stringa

Se cid contiene un valore nullo, creare la query di recupero selezionando Aid, Acity e Azip dall'indirizzo. Impostare Aid su un valore nullo.

Il seguente scenario è un oggetto principale con un oggetto secondario in un oggetto contenitore con due riferimenti chiave. Questo scenario ha esito positivo in queste tre circostanze.

Tabella 22. Cliente

Attributo	Tipo
cid	Intero (chiave)
nome	Stringa
commenti	Stringa
Indirizzo	Indirizzo
Aiuto	Intero (chiave) ASI:FK=cid
Acity	Stringa /chiave) ASI:FK=name
Azip	Stringa

Se il nome contiene un valore nullo, creare la query di recupero selezionando Aid, Acity e Azip dall'indirizzo, dove Aid=Cid e Acity ha un valore nullo.

---

## Funzionalità

Se il connettore incontra un valore vuoto in una chiave, lo confronta con il valore UseNull nell'attributo. Se il valore è true, il connettore aggiunge un valore nullo alla query. Ciò ha influenza sulle seguenti operazioni di istruzione:

- Retrieve
- Recupera da contenuto
- Update
- Delete

---

## Appendice E. Mainframe Data Access

L'adattatore per JDBC supporta mainframe data access utilizzo di IBM DB2 Information Integrator Classic Federation per z/OS, V 8.1. Questo prodotto DB2 fornisce applicazioni web e distribuite con connettività in lettura e scrittura ai database mainframe. Fornisce la federazione delle origini dati mainframe e l'accesso basato su SQL di elevate prestazioni. E' possibile utilizzare il desktop con strumenti e applicazioni Internet a scelta per accedere alle informazioni critiche di mainframe. L'adattatore JDBC è conforme alle specifiche JDBC 2.1.

---

### Supporto del connettore JDBC per mainframe data access

Il connettore JDBC effettua le richieste di accesso dati SQL a un DB2 II Classic Federation Server. Il connettore JDBC utilizza il driver JDBC, denominato DB2 II Classic Federation JDBC Connector.

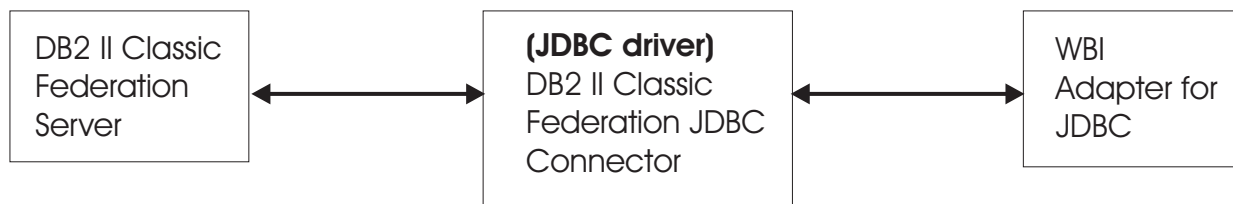


Figura 14. Supporto mainframe data access

L'adattatore per JDBC utilizza DB2 II Classic Federation Server per fornire le funzioni di seguito indicate.

- **Inoltro sottoscrizioni:** il connettore JDBC non può essere utilizzato per l'inoltro delle sottoscrizioni, poiché DB2 II Classic Federation Server non dispone del supporto di inoltro eventi.
- **Elaborazione richieste:** L'adattatore per JDBC supporta queste origini dati mainframe:
  - Software AG ADABAS
  - VSAM
  - CA IDMS
  - IBM IMS™ Transaction Manager and Database
  - IBM DB2 Databases

---

### Utilizzo di mainframe data access

Per prepararsi all'utilizzo di mainframe data access:

1. Installare DB2 II Classic Federation per z/OS, V 8.1 client software.
2. Includere DB2 JDBC Driver (cacjdbc21.jar) nello script di avvio del connettore JDBC JDBC\_DRIVER\_PATH (consultare la sezione "Avvio del connettore" a pagina 91).

**Nota:** E' necessario utilizzare il driver JDBC fornito con il software del client.

Per collegarsi a DB2 II Classic Federation Server:

1. Collegarsi al seguente URL per effettuare il login a VLAN:  
jdbc:cac:DATASOURCENAME:tcp/nome host o indirizzo IP/numero di porta  
Ad esempio:  
jdbc:cac:CACWBI:tcp/9.30.136.99/8094
2. Quando richiesto, immettere ID utente e password per VLAN. Una volta completato il login, viene visualizzata una casella a comparsa indicante che l'autenticazione è stata effettuata correttamente. Chiudere questa finestra ma non fare clic su alcun pulsante. La connessione va in timeout dopo un determinato periodo di tempo. Se si riceve un messaggio di errore, è necessario effettuare di nuovo il login a VLAN.

---

## Componenti di mainframe data access

Questi componenti interagiscono per abilitare l'accesso dei dati SQL ai dati mainframe:

- **DB2 II Classic Federation ODBC e Connettore JDBC:** elabora le chiamate delle funzioni, inoltra le richieste SQL a un'origine dati specificata e restituisce risultati all'adattatore per JDBC.
- **Definizione origine dati:** è costituita dal nome e dalla posizione dei dati cui si desidera accedere. Le definizioni dell'origine dati richiesta contengono un nome dell'origine dati e i parametri di comunicazione per TCP/IP o WebSphere MQ. Il nome dell'origine dati identifica un server dell'applicazione specifico o un server enterprise utilizzato per eseguire le richieste di accesso dati.
- **Interfaccia a livello di richiamo** utilizzata per collegare DB2 II Classic Federation ODBC o il connettore JDBC all'attività Query Processor in esecuzione su un server.
- **DB2 II Classic Federation Connection Handler:** utilizzato per comunicare con un server dell'applicazione o un server enterprise. DB2 II Classic Federation fornisce un Gestore connessioni che supporta le implementazioni TCP/IP che utilizzano un'interfaccia Winsock e WebSphere MQ.

L'adattatore per JDBC effettuare le richieste di accesso dati SQL a DB2 II Classic Federation Server e supporta l'accesso ai metadati mediante i richiami dei metadati JDBC standard (utilizzati da JDBC ODA). L'adattatore utilizza DB2 II Classic Federation ODBC e il connettore JDBC.



---

## Indice analitico

### A

Accesso alle informazioni di mainframe 147  
Adattatore  
  database supportati 9  
  dipendenze 9  
  requisiti dell'ambiente 9  
  sistemi operativi supportati 9  
  software non IBM supportato 9  
  struttura dei file installati 10  
Aggiornamento di procedure memorizzate 52  
ALL SCHEMAS 21  
Architettura dei connettori 2  
attributo semplice 33

### B

Business Object Designer 17  
  Finestra Proprietà BO 23  
  selezione degli oggetti di database 21

### C

Classe degli oggetti business personalizzati 69  
Commit e rollback delle transazioni 55  
Componente specifico dell'applicazione 1  
Comportamento di archiviazione 5  
Configurazione di un evento ed elaborazione di un archivio 88  
Connector Configurator 70  
Connessioni al database perdute 6  
Connettore  
  distinto 1  
  Proprietà di configurazione 70  
Convalida 55  
Conversione data 97  
Creazione delle operazioni 40  
Creazione di procedure memorizzate 51

### D

Database supportati 9  
Dati binari 66  
DB2 II Classic Federation for z/OS 147  
DB2 II Classic Federation Server  
  collegamento a 147  
  nessun inoltra degli eventi 147  
Definizioni di oggetto business generazione con JDBC ODA 13

### E

Elaborazione dei verbi 38, 40  
Elaborazione di archivi ed eventi 88  
Eliminazione delle procedure memorizzate 52

### F

File di definizione degli oggetti business di esempio 30  
File di errori e messaggi di tracce 16  
File di messaggi, errori e tracce 16

### G

Gestione degli errori e registrazione 98

### I

Informazioni specifiche sull'applicazione  
  a livello di attributo 60  
  parametri nome-valore 61  
  a livello di oggetto business 58  
  formato per i verbi 67  
  per gli attributi che rappresentano child 66  
Inoltra sottoscrizione non disponibile 147  
installazione  
  adattatori  
  informazioni su 10  
Istruzioni SQL  
  elaborazione 51  
  panoramica 2  
  utilizzo 46

### J

Java Virtual Machine (JVM) 10  
JDBC  
  driver 9  
  specifica 1  
  supporto multi-driver 69

### L

Livelli di traccia 17

### M

Mainframe data access  
  elaborazione richieste 147  
  preparazione all'utilizzo 147  
  richieste accesso dati SQL 147  
  utilizzo di DB2 II Classic Federation per z/OS 147  
meccanismo SmartFiltering 5  
Messaggi di errore 100  
Messaggi di traccia  
  tipi 99  
Metadati 2

### N

Nomi degli attributi 33  
Notifica eventi 5

### O

Object discovery agent (ODA) 13  
ODA JDBC 13  
  avvio 15  
  definizione dell'oggetto business 27  
  file di esempio 30  
  esecuzione di più istanze 16  
  installazione 13  
  proprietà di inizializzazione 19  
Oggetti business di esempio 143  
Oggetto business  
  confronto 40  
  convenzioni di denominazione 33  
  creazione delle operazioni 3, 40  
  delta 38  
  di esempio 143  
  dopo immagine 38  
  elaborazione 3  
  Elaborazione verbi 38, 40  
  gerarchico 3, 34  
  ID eventi 5  
  livello superiore 3, 34  
  metadati 2  
  nomi degli attributi 33  
  Operazione RetrievalByContent 3  
  Operazioni DeltaUpdate 45  
  operazioni di aggiornamento 4, 39, 43  
  operazioni di eliminazione 4, 46  
  operazioni di richiamo 3, 42  
  operazioni di richiamo per aggiornamento 54  
  Operazioni RetrieveByContent 43  
  proprietà dell'attributo 55  
  relazione con cardinalità multipla 34  
  relazione con cardinalità singola 34  
  senza nessun effetto 34  
  singolo 3, 34  
  struttura 33  
  valore chiave vuoto o nullo 145  
Oggetto wrapper 34, 38, 46  
Operazioni DeltaUpdate 45  
Operazioni di aggiornamento 39, 43  
Operazioni di eliminazione 46  
  fisica 46  
  logica 46  
Operazioni di richiamo 42  
Operazioni RetrieveByContent 43

### P

Polling degli eventi 87  
Problemi  
  di avvio 97  
  mappatura degli oggetti business 97

- Procedure memorizzate
  - creazione delle operazioni 51
  - definito 47
  - elaborazione 51
  - esempi 49
  - operazioni di aggiornamento 52
  - operazioni di eliminazione 52
  - Operazioni RetrieveByContent 54
  - panoramica 2
  - selezione degli attributi 25
- Procedure memorizzate di richiamo per aggiornamento 54
- procedure memorizzate
  - RetrieveByContent 54
- Proprietà configurazione
  - specifiche dei connettori 70, 127
  - dettagli proprietà 128
  - standard 70
- Proprietà dell'attributo 55

## R

- Recapito eventi 87
- Relazione con cardinalità multipla 34, 36
- Relazioni con cardinalità singola
  - che memorizzano la relazione nel
    - child 37
  - definito 34
  - senza proprietà 35
- Rollback 55

## S

- Scenari riusciti/non riusciti
  - valore chiave vuoto o nullo 145
- Script
  - gestione connettore 10
- Script di gestione connettore 10
- Serie del codice del carattere Unicode 10
- Serie del codice di caratteri Unicode 7
- Serie di caratteri a due byte 6
- Sistemi operativi supportati 9
- Software non IBM supportato 9
- Struttura dei file installati 10
- Struttura del connettore 1
- supporto lingua bidirezionale 7

## T

- Tabella di archivio 87
- Tabella di archivio e di eventi
  - installazione
    - script SQL 88
    - schema 89
- Tabella eventi 87
- Tipi di errori 99
- Trigger 5
- Trigger del database 5

## V

- valore chiave vuoto o nullo 145
- Verifica dell'installazione 10

---

## Informazioni particolari

Queste informazioni sono state sviluppate per prodotti e servizi offerti negli Stati Uniti. È possibile che negli altri paesi l'IBM non offra i prodotti, le funzioni o i servizi illustrati in questo documento. Consultare il rappresentante locale IBM per informazioni sui prodotti e sui servizi disponibili attualmente nel proprio paese. Qualunque riferimento relativo a prodotti, programmi o servizi IBM non implica che solo quei prodotti, programmi o servizi IBM possano essere utilizzati. In sostituzione a quelli forniti dall'IBM possono essere utilizzati prodotti, programmi o servizi funzionalmente equivalenti che non comportino violazione dei diritti di proprietà intellettuale e di altri diritti dell'IBM. È comunque responsabilità dell'utente valutare e verificare la possibilità di utilizzare altri programmi e/o prodotti, fatta eccezione per quelli espressamente indicati dall'IBM. L'IBM può avere brevetti o domande di brevetto in corso relativi a quanto trattato nel presente documento. La fornitura di questa pubblicazione non implica la concessione di alcuna licenza su di essi. E' possibile inviare domande di licenza, per iscritto a:

*IBM Director of Commercial Relations  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
Deutschland*

Per domande di autorizzazioni relative a informazioni DBCS, contattare IBM Intellectual Property Department nel proprio paese oppure inviare le domande a:

*IBM World Trade Asia Corporation Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106-0032, Japan*

Il seguente paragrafo non è valido per il Regno Unito o per tutti i paesi le cui leggi nazionali siano in contrasto con le disposizioni in esso contenute:

L'INTERNATIONAL BUSINESS MACHINES CORPORATION FORNISCE QUESTA PUBBLICAZIONE "NELLO STATO IN CUI SI TROVA", SENZA ALCUNA GARANZIA, ESPLICITA O IMPLICITA, IVI INCLUSE EVENTUALI GARANZIE DI COMMERCIALIZZABILITÀ ED IDONEITÀ AD UNO SCOPO PARTICOLARE. Alcuni stati non consentono la rinuncia a garanzie esplicite o implicite in determinate transazioni; quindi la presente dichiarazione potrebbe essere non essere a voi applicabile. Questa pubblicazione potrebbe contenere imprecisioni tecniche o errori tipografici. Le informazioni incluse in questo documento vengono modificate su base periodica; tali modifiche verranno incorporate nelle nuove edizioni della pubblicazione. L'IBM si riserva il diritto di apportare miglioramenti e/o modifiche al prodotto o al programma descritto nel manuale in qualsiasi momento e senza preavviso. Tutti i riferimenti a siti Web non dell'IBM contenuti in questo documento sono forniti solo per consultazione e non rappresentano in alcun modo un'approvazione di tali siti Web. I materiali disponibili presso i siti Web non fanno parte di questo prodotto e l'utilizzo di questi è a discrezione dell'utente. Tutti i commenti e i suggerimenti inviati potranno essere utilizzati liberamente dall'IBM e diventeranno esclusiva della stessa. Coloro che detengono la licenza su questo programma e desiderano avere informazioni su di esso allo scopo di consentire: (i) uno scambio di informazioni tra programmi indipendenti ed altri (compreso questo) e (ii) l'uso reciproco di tali informazioni, dovrebbero rivolgersi a:

*IBM Corporation  
577 Airport Blvd., Suite 800  
Burlingame, CA 94010  
U.S.A*

Tali informazioni possono essere disponibili, in base ad appropriate clausole e condizioni, includendo in alcuni casi, il pagamento di una tassa. Il programma su licenza descritto in questo manuale e tutto il materiale su licenza ad esso relativo sono forniti dall'IBM nel rispetto delle condizioni previste dalla licenza d'uso. Tutti i dati contenuti in questa pubblicazione sono stati determinati in ambiente controllato. Pertanto, i risultati ottenuti in ambienti operativi diversi possono variare in modo considerevole. Alcune misurazioni potrebbero essere state fatte su sistemi di livello di sviluppo per cui non si garantisce che queste saranno uguali su tutti i sistemi disponibili. Inoltre, alcune misurazioni possono essere state stimate tramite estrapolazione. I risultati attuali possono quindi variare. Gli utenti del presente documento dovranno verificare i dati applicabili per i propri ambienti specifici. Le informazioni relative a prodotti non-IBM sono state ottenute dai fornitori di tali prodotti. L'IBM non ha verificato tali prodotti e, pertanto, non può garantirne l'accuratezza delle prestazioni. Eventuali commenti relativi alle prestazioni dei prodotti non IBM devono essere indirizzati ai fornitori di tali prodotti. Tutti le dichiarazioni riguardanti la direzione o le decisioni future di IBM sono soggette a variazione o ritiro senza preavviso e costituiscono solo degli obiettivi. Questa pubblicazione contiene esempi di dati e prospetti utilizzati quotidianamente nelle operazioni aziendali. Pertanto, può contenere nomi di persone, società, marchi e prodotti. Tutti i nomi contenuti nella pubblicazione sono fittizi e ogni riferimento a nomi e indirizzi reali è puramente casuale. LICENZA DI COPYRIGHT: Queste informazioni contengono programmi applicativi di esempio in lingua originale, che illustrano le tecniche di programmazione su diverse piattaforme operative. E' possibile copiare, modificare e distribuire queste esempi di programmi sotto qualsiasi forma senza alcun pagamento alla IBM, allo scopo di sviluppare, utilizzare, commercializzare o distribuire i programmi applicativi in modo conforme alle API (Application Programming Interface) a seconda della piattaforma operativa per cui gli esempi dei programmi sono stati scritti. Questi esempi non sono stati testati approfonditamente tenendo conto di tutte le condizioni possibili. La IBM, quindi, non può garantire o assicurare l'affidabilità, la praticità o il funzionamento di questi programmi. Se si stanno visualizzando queste informazioni come softcopy, è possibile che le fotografie e le illustrazioni a colori non vengano visualizzate.

---

## Informazioni sull'interfaccia di programmazione

Le informazioni sull'interfaccia di programmazione, se fornite, sono designate per creare il software applicativo utilizzando questo programma. Le interfacce di programmazione di utilizzo generale consentono all'utente di scrivere il software di applicazione che ottiene i servizi di questi strumenti del programma'. Tuttavia, queste informazioni possono contenere anche diagnosi, modifiche ed informazioni sull'ottimizzazione. Le informazioni sulle diagnosi, le modifiche e sull'ottimizzazione vengono fornite come supporto per l'esecuzione del debug del software applicativo.

**Avvertenza:** Non utilizzare queste informazioni sulle diagnosi, le modifiche e sull'ottimizzazione come un'interfaccia di programmazione poiché sono soggette a cambiamenti.

---

## Marchi e marchi di servizio

I seguenti termini sono marchi della International Business Machines Corporation negli Stati Uniti e/o in altre nazioni: i5/OS

IBM

il logo IBM

AIX

CICS

CrossWorlds

DB2

DB2 Universal Database

Domino

IMS

Informix

iSeries

Lotus

Lotus Notes

MQIntegrator

MQSeries

MVS

OS/400

Passport Advantage

SupportPac

WebSphere

z/OS

Microsoft, Windows, Windows NT ed il logo Windows sono marchi della Microsoft Corporation negli Stati Uniti e/o in altri paesi. MMX, Pentium e ProShare sono marchi o marchi registrati della Intel Corporation negli Stati Uniti e/o negli altri paesi. Java e tutti i marchi basati su Java sono marchi della Sun Microsystems, Inc. negli Stati Uniti e/o negli altri paesi. Linux è un marchio della Linus Torvalds negli Stati Uniti e/o negli altri paesi. Altri nomi di società, di servizi e prodotti possono essere marchi o marchi di servizi di altre società.

WebSphere Business Integration Server Express e Express Plus comprendono un software sviluppato da Eclipse Project (<http://eclipse.org/>).



WebSphere Business Integration Server Express, Versione 4.4, e WebSphere Business Integration Server Express Plus, Versione 4.4







Printed in Denmark by IBM Danmark A/S