

WebSphere Business Integration Server  
Express e Express Plus



# Guida per l'utente per l'adattatore JText

*Versione 44*



WebSphere Business Integration Server  
Express e Express Plus



# Guida per l'utente per l'adattatore JText

*Versione 44*

**Nota!**

Prima di utilizzare queste informazioni e il relativo prodotto, leggere attentamente le informazioni riportate in "Informazioni particolari" a pagina 117.

**22 Aprile 2005**

Questa edizione del documento si applica a WebSphere Business Integration Server Express Adapter for JText (5724-H03) versione 5.6.x, e a tutti i rilasci e le modifiche successivi se non altrimenti indicato nelle nuove edizioni.

Per inviare commenti sulla documentazione inviare un messaggio via email all'indirizzo [doc-comments@us.ibm.com](mailto:doc-comments@us.ibm.com). Siamo in attesa di ricevere i vostri commenti.

Tutti i commenti e i suggerimenti inviati potranno essere utilizzati liberamente dall'IBM e diventeranno esclusiva della stessa.

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

---

# Indice

<b>Informazioni</b> . . . . .	<b>v</b>
Contenuti di questo documento . . . . .	v
Cosa non è contenuto in questo documento . . . . .	v
A chi è rivolto questo manuale . . . . .	v
Documentazione correlata . . . . .	v
Convenzioni tipografiche . . . . .	vi
<b>Novità di questo rilascio</b> . . . . .	<b>vii</b>
Versione 5.6.x . . . . .	vii
<b>Capitolo 1. Panoramica dell'adattatore JText</b> . . . . .	<b>1</b>
Componenti adattatore . . . . .	1
Oggetti business utilizzati dal connettore JText . . . . .	3
Funzionamento del connettore . . . . .	6
Funzioni connettore . . . . .	13
Elaborazione dati dipendenti dalla locale . . . . .	14
<b>Capitolo 2. Installazione dell'adattatore JText</b> . . . . .	<b>15</b>
Panoramica delle attività di installazione . . . . .	15
Ambiente dell'adattatore . . . . .	15
Installazione dell'adattatore JText . . . . .	16
Verifica dell'installazione . . . . .	17
<b>Capitolo 3. Configurazione dell'adattatore JText</b> . . . . .	<b>19</b>
Panoramica di Connector Configurator Express . . . . .	19
Avvio Connector Configurator Express . . . . .	20
Esecuzione di Configurator da System Manager . . . . .	21
Creazione di una maschera delle proprietà specifica del connettore . . . . .	21
Creazione di un nuovo file di configurazione . . . . .	24
Utilizzo di un file esistente . . . . .	25
Completamento di un file di configurazione . . . . .	26
Impostazione delle proprietà del file di configurazione . . . . .	27
Salvataggio del file di configurazione . . . . .	37
Modifica file di configurazione . . . . .	37
Completamento della configurazione . . . . .	38
Utilizzo Connector Configurator Express in un ambiente globalizzato . . . . .	38
Avvio del connettore . . . . .	39
Arresto connettore . . . . .	40
Creazione più istanze del connettore . . . . .	41
Aggiunta oggetti aziendali supportati . . . . .	43
<b>Capitolo 4. Utilizzo dei metaoggetti del connettore JText</b> . . . . .	<b>45</b>
convenzioni di denominazione del metaoggetto JText . . . . .	45
Struttura del metaoggetto JText . . . . .	46
Attività di configurazione comune . . . . .	58
<b>Capitolo 5. Risoluzione di problemi del connettore JText</b> . . . . .	<b>81</b>
Registrazione del messaggio di errore . . . . .	81
Problemi con la denominazione del metaoggetto . . . . .	81
Problemi con il trigger degli eventi . . . . .	81
Gestione errore JText . . . . .	82
<b>Capitolo 6. Migrazione a o aggiornamento al connettore JText.</b> . . . . .	<b>87</b>
Aggiornamento scenari . . . . .	87

Aggiornamento alla versione 5.6.x dalla versione 5.3.x . . . . .	87
Motivi per l'aggiornamento alla versione 4.0.x dalla versione 3.2.0 . . . . .	88
Aggiornamento alla versione 4.0.x. . . . .	88
Motivi per l'aggiornamento dal connettore Text . . . . .	90
Aggiornamento al connettore JText . . . . .	90
<b>Appendice. Proprietà di configurazione standard per i connettori . . . . .</b>	<b>91</b>
Nuove proprietà. . . . .	91
Panoramica sulle proprietà del connettore standard . . . . .	91
Guida di riferimento rapido per le proprietà standard . . . . .	93
Proprietà standard . . . . .	98
<b>Indice analitico. . . . .</b>	<b>115</b>
<b>Informazioni particolari . . . . .</b>	<b>117</b>
Informazioni sull'interfaccia di programmazione. . . . .	118
Marchi e marchi di servizio. . . . .	119

---

## Informazioni

I prodotti <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 della guida 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.

Questo documento descrive l'installazione, la configurazione, lo sviluppo dell'oggetto business e la risoluzione di problemi per WebSphere Business Integration Server Express JText Adapter<sup>(TM)</sup>.

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 fanno riferimento a entrambi i prodotti.

---

## Contenuti di questo documento

Questo documento descrive l'installazione, la configurazione delle proprietà del connettore, lo sviluppo dell'oggetto business e la risoluzione di problemi per WebSphere Business Integration Server Express JText Adapter.

---

## Cosa non è contenuto in questo documento

Questo documento non descrive le metriche di distribuzione e gli argomenti di pianificazione della capacità come bilanciamento del carico del server, il numero dei thread di elaborazione dell'adattatore, le emissioni minime e massime e i limiti di tolleranza.

Tali argomenti sono univoci per ogni distribuzione cliente e deve essere misurato in o accanto all'ambiente esatto dove l'adattatore va distribuito. Contattare il rappresentante di servizi IBM per discutere la configurazione del sito di distribuzione e per dettagli sulla pianificazione e la valutazione di queste metriche, data la configurazione specifica.

---

## A chi è rivolto questo manuale

Questo documento è per i consulenti e i clienti WebSphere. E' necessario essere a conoscenza dei fondamentali del broker di integrazione, dello sviluppo dell'oggetto business e possibilmente dello sviluppo del gestore dati.

---

## Documentazione correlata

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

E' possibile scaricare, installare e visualizzare la documentazione di WebSphere Business Integration Server Express al seguente sito:  
<http://www.ibm.com/websphere/wbiserverexpress/infocenter>.

---

## Convenzioni tipografiche

Questo manuale utilizza le seguenti convenzioni:

---

carattere courier	Indica un valore letterale, come un nome comando, un nome file, le informazioni da immettere o quelle che il sistema stampa sullo schermo.
<b>grassetto</b>	Indica un termine nuovo la prima volta che viene visualizzato.
<i>maiuscolo, italic</i>	Indica un nome variabile o un riferimento incrociato.
<i>riquadro blu</i>	Uno schema blu, visibile solo quando si visualizza il manuale in linea, indica un collegamento ipertestuale a riferimento incrociato. Fare clic all'interno del riquadro per passare all'oggetto del riferimento.
{ }	Su una riga della sintassi, le parentesi graffe racchiudono una serie di opzioni di cui ne va selezionata 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.
< >	Nelle convenzioni di denominazione, le parentesi angolari racchiudono singoli elementi di un nome per distinguerli l'uno dall'altro, ad esempio <code>&lt;nome_server&gt;&lt;nome_connettore&gt;tmp.log</code> .
/, \	In questo documento, le barre retroverse (\) sono utilizzate come convenzioni per i percorsi delle directory per Windows. Linux e i5/OS utilizzano le barre (/) per i percorsi di directory. Tutti i nomi percorsi del prodotto di sistema di integrazione business WebSphere sono relativi alla directory dove è installato il prodotto sul sistema.
<code>%text%</code> e <code>\$text</code>	Un testo all'interno dei segni percentuale (%) indica il valore della variabile di sistema o della variabile utente <code>text</code> . L'annotazione equivalente in un ambiente Linux è <code>\$text</code> , indicando il valore della variabile di ambiente <code>text</code> Linux.
<i>ProductDir</i>	Rappresenta la directory in cui viene installato il prodotto. I valori predefiniti per ogni piattaforma sono i seguenti: <ul style="list-style-type: none"><li>• Windows: IBM\WebSphereServer</li><li>• i5/OS: /WQIBM/ProdData/WBIServer44/product</li><li>• Linux: /home/\${username}/IBM/WebSphereServer</li></ul>

---

---

## Novità di questo rilascio

---

### Versione 5.6.x

Questo rilascio del documento per la versione del connettore JText 5.6.x contiene le seguenti nuove funzioni:

- L'adattatore, in esecuzione su Windows, fornisce supporto allo script bidirezionale per i metadati JText
- Il supporto per il codice Cinese GB18030 è stato aggiunto per questo rilascio
- Il supporto è stato aggiunto per l'FTP sicuro
- Una nuova proprietà booleana specifica del connettore, "NoPoll", è stata introdotta per disattivare facoltativamente il poll. Il valore predefinito è false. Quando è impostato su true, l'adattatore elabora solo le richieste e non esegue il poll.

La versione 5.6.x dell'adattatore è supportata sulle seguenti piattaforme:

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

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



---

## Capitolo 1. Panoramica dell'adattatore JText

Questo capitolo descrive WebSphere Business Integration Server Express Adapter per JText. L'adattatore abilita un broker di integrazione per comunicare con un'applicazione scambiando il testo o i file binari. Questo connettore facilita l'integrazione dei dati con le applicazioni a cui manca l'API.

Gli adattatori consistono di due parti: il framework del connettore e il componente specifico dell'applicazione. Il framework del connettore, il cui codice è comune a tutti i connettori, agisce come un intermediario tra il broker di integrazione e il componente specifico dell'applicazione. Il componente specifico dell'applicazione contiene un codice su misura per l'applicazione particolare. Il framework del connettore fornisce i seguenti servizi tra il broker di integrazione e il componente specifico dell'applicazione:

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

Questo documento contiene informazioni relative al framework del connettore e al componente specifico dell'applicazione. Questo fa riferimento a questi componenti come adattatore.

Gli argomenti inclusi in questo capitolo sono:

- "Componenti adattatore"
- "Oggetti business utilizzati dal connettore JText" a pagina 3
- "Funzionamento del connettore" a pagina 6
- "Funzioni connettore" a pagina 13
- "Elaborazione dati dipendenti dalla locale" a pagina 14

Per ulteriori informazioni sul rapporto del broker di integrazione con il connettore, consultare *System Administration Guide*.

Utilizzare l'adattatore JText quando:

- Un'applicazione non ha le API C, C++, o Java standard tramite le quali un broker di integrazioni può comunicare.
- Non è semplice avere una tabella di eventi per un'applicazione costruita sul cliente.
- La Stringa o i file binari sono il metodo più appropriato per scambiare dati.

In questi casi, il metodo più semplice per integrare un'applicazione in un sistema più ampio potrebbe essere di scambiare stringhe o file binari tramite il connettore JText.

---

### Componenti adattatore

L'adattatore JText ha i seguenti componenti:

- "Componente specifico dell'applicazione" a pagina 2
- "Gestori dati" a pagina 2
- "Metaoggetti" a pagina 3

Figura 1 illustra l'architettura del connettore JText con WebSphere InterChange Server Express come broker di integrazione.

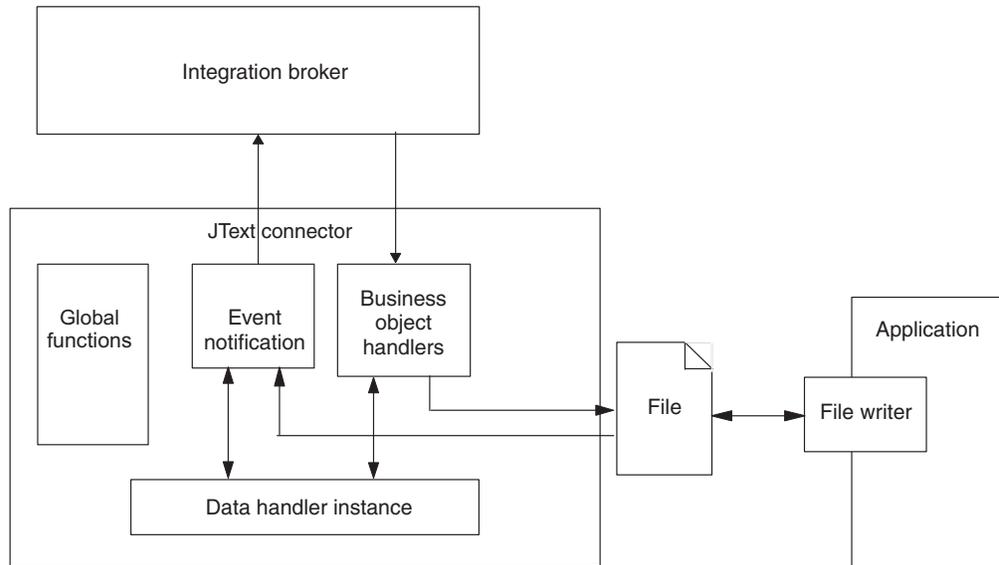


Figura 1. architettura adattatore JText

## Componente specifico dell'applicazione

Il componente specifico dell'applicazione dell'adattatore JText manipola i file e chiama un gestore dati specificato per convertire i dati tra oggetti business e stringhe o matrici di byte. Gestisce anche la comunicazione con il broker di integrazione.

## Gestori dati

L'obiettivo del connettore JText è di fornire la conversione tra il formato del file esistente e un oggetto business. Per fare ciò, utilizza il gestore dati specificato nella configurazione del metaoggetto del connettore.

Il gestore dati esegue la conversione senza interagire con il sistema di file in nessun modo, leggendo da o scrivendo nei file. Tutte le interazioni con il file di testo sono gestite da altri componenti del connettore.

Per gestire la conversione di dati, è possibile utilizzare i gestori di dati che WebSphere Business Integration Server Express Adapter Framework fornisce o i gestori di dati creati per gestire necessità specifiche di formattazione del testo. Il prodotto fornisce i seguenti gestori di dati:

- NameValue — Analizza dati di testo basati su campi denominati. In questo caso, il file di testo contiene campi che identificano il tipo di oggetto business (BusinessObject=B0name), il verbo (Verb=verbName), e il numero degli attributi (AttributeCount=numericValue).
- Delimitato — Utilizzato principalmente dove è più importante l'efficienza della lettura della macchina. Analizza dati di testo in base ad un delimitatore specifico che separa i campi singoli di dati dell'oggetto business.
- FixedWidth — Analizza dati di testo utilizzando i campi a lunghezza fissa. Le lunghezze del campo sono specificate dalla proprietà MaxLength di ogni attributo dell'oggetto business. Il valore di questa proprietà è memorizzato nella definizione dell'oggetto business.

Per ulteriori informazioni, consultare “Come funziona l’elaborazione del gestore dati” a pagina 12. Per ulteriori informazioni sui gestori dati forniti con il prodotto, consultare *Data Handler Guide*.

## Metaoggetti

Oltre alle proprietà di configurazione del connettore standard e specifiche dell’applicazione impostata in Connector Configurator Express, il connettore JText ha un insieme di proprietà di configurazione che consentono di configurare il connettore per elaborare diversi oggetti business in modo diverso. Vengono impostate queste proprietà utilizzando i metaoggetti JText. Un metaoggetto è un tipo speciale di oggetto business che contiene informazioni di configurazione.

Il connettore utilizza le informazioni del metaoggetto per determinare quali classi utilizzare per trasformare stringhe o matrici di byte che legge dai file negli oggetti business e per formattare le stringhe o le matrici di byte dagli oggetti business nei file. I metaoggetti JText specificano le directory, le estensioni di file, i nomi file, i delimitatori dell’oggetto business e i gestori dati da utilizzare durante l’evento e l’elaborazione della richiesta.

L’adattatore JText utilizza metaoggetti internamente. Non li invia tramite il broker di integrazione. Per ulteriori informazioni relative all’utilizzo dei metaoggetti per configurare il connettore, consultare Capitolo 4, “Utilizzo dei metaoggetti del connettore JText”, a pagina 45.

---

## Oggetti business utilizzati dal connettore JText

Gli oggetti business per il connettore JText devono fornire dati nel formato richiesto dal gestore dati specificato per la conversione. Tuttavia, il connettore JText potrebbe non richiedere un insieme di oggetti business progettati in modo specifico comparabili agli oggetti business specifici dell’applicazione per un connettore dell’applicazione.

Ad esempio, il gestore dati NameValue richiede che ogni parte di dati abbia una stringa che la identifica (come CustomerName=Kumar, Region=NE e Department=HR). Poiché ogni definizione di oggetto business generico contiene attributi i cui nomi identificano ogni parte di dati, il connettore JText può utilizzare oggetti business generici.

Tuttavia, poiché gli oggetti business generici rappresentano un superinsieme di informazioni richieste da una quantità di diverse applicazioni, ogni oggetto business generico di solito contiene molte più informazioni di quanto sia richiesto da una applicazione.

Quindi, per convertire i dati in dimensioni gestibili per ogni applicazione, è buona norma creare il proprio oggetto business per ogni tipo di dati da elaborare. Nell’oggetto business, fornire solo i dati richiesti dall’applicazione e le informazioni richieste dal gestore dati.

Ad esempio, per il gestore dati FixedWidth, è necessario verificare che ogni attributo dell’oggetto business ha un valore specificato per la proprietà di attributo MaxLength. Per WebSphere Business Integration Server Express Data Handler for XML, sono richieste altre informazioni specifiche. D’altro lato, per i gestori dati NameValue e Delimited, l’oggetto business non deve contenere le informazioni che non sono già contenute in un oggetto business generico. Consultare la *Data Handler Guide* per informazioni specifiche di ogni gestore dati.

Oltre a fornire dati, un'oggetto business può contenere informazioni che consentono al connettore di ottenere in modo dinamico il nome file evento dell'oggetto business o di restituire il nome file di output al broker di integrazione. Per configurare il connettore per questa elaborazione dinamica, le informazioni specifiche dell'applicazione al livello dell'oggetto business devono contenere la seguente coppia nome-valore:

- `cw_mo_JTextConfig = DynChildMOAttrName`

Se l'oggetto business contiene altre informazioni specifiche dell'applicazione utilizzata dal gestore dati, la coppia nome-valore deve apparire prima nell'oggetto business e deve essere separato dalle informazioni specifiche dell'applicazione da un punto e una virgola (;). Il connettore legge la coppia nome-valore fino al punto e virgola per determinare se utilizzare l'elaborazione dinamica, quindi passa le informazioni che appaiono dopo il punto e la virgola al gestore dati.

## Utilizzo di un metaoggetto secondario dinamico

Un metaoggetto secondario dinamico consente al nome file di essere scambiato con InterChange Server Express. Questa sezione descrive:

- "Perché utilizzare un metaoggetto secondario"
- "Utilizzo del metaoggetto secondario dinamico" a pagina 4
- "Attributi di un metaoggetto secondario dinamico" a pagina 5

### Perché utilizzare un metaoggetto secondario

Creare ed utilizzare un metaoggetto secondario dinamico per fare in modo che il connettore funzioni nel modo seguente:

#### Richieste chiamate servizio

- Generare in modo dinamico un nome file di output per ogni tipo di oggetto business (in base al valore inserito nell'attributo `OutFileName` secondario presso il broker di integrazione) o per ogni oggetto business individuale (se il broker di integrazione specifica la sequenza).

**Nota:** Il connettore utilizza l'attributo `FileWriteMode` secondario per determinare se scrivere o collegare al file specificato nell'attributo `OutFileName` secondario.

- Restituire il nome di ogni nome file di output generato dal connettore (se l'attributo `OutFileName` secondario non contiene un valore). In questo caso, il connettore funziona nel modo seguente:
  - deriva il nome dal nome dell'oggetto business principale
  - scrive l'oggetto in quel file
  - popola l'attributo del metaoggetto `OutFileName` con il nome derivato
  - passa il nome derivato al broker di integrazione, che ottiene il nome file di output creato dinamicamente senza averlo specificato

#### Elaborazione evento

Il connettore popola l'attributo `InFileName` secondario con il nome del file dal quale è stato letto l'oggetto business.

### Utilizzo del metaoggetto secondario dinamico

Per dare in modo che il connettore elabori il nome file in modo dinamico, è necessario:

1. Creare un metaoggetto dinamico con attributi specifici.

2. Nell'oggetto business dei dati, aggiungere un attributo che rappresenta il metaoggetto secondario dinamico.
3. Nell'oggetto business di dati, specificare quanto segue nelle informazioni specifiche dell'applicazione a livello dell'oggetto business:

`cw_mo_JTextConfig = DynChildMOAttrName`

dove *DynChildMOAttrName* è il nome dell'attributo nell'oggetto business di dati che rappresenta l'oggetto business secondario dinamico. Per un esempio, consultare Figura 2.

**Importante:** Il prefisso `cw_mo_` è richiesto quando si utilizza un gestore dati. Se il prefisso manca, il connettore scrive il metaoggetto secondario dinamico nel file di output specificato come se fosse un oggetto business di dati.

4. Nel metaoggetto secondario dinamico, specificare i valori per gli attributi nel metaoggetto secondario dinamico.

### Attributi di un metaoggetto secondario dinamico

Un metaoggetto secondario dinamico deve contenere i seguenti attributi:

- `FileWriteMode` — Un attributo di stringa il cui valore specifica se il connettore si collega o sovrascrive un file di output esistente. Il valore di questo attributo può essere "a" per il collegamento o "o" per la sovrascrittura. Il connettore esamina solo la prima lettera e non considera il valore.
- `InFileName` — Un attributo di stringa che è popolato con il nome file evento (file e percorso assoluto dal quale è ottenuto l'oggetto business).
- `OutFileName` — Un attributo di stringa il cui valore può contenere il nome file, il percorso assoluto e il nome file, o l'URL dell'FTP che il connettore utilizza quando scrive nel file di output.
  - Se questo attributo contiene solo il nome file, il connettore scrive il file specificato nella directory dalla quale è stato avviato.
  - Se questo attributo contiene solo il percorso assoluto e il nome file, il connettore scrive il file specificato nella directory specificata.
  - Se questo attributo contiene un URL FTP, il connettore ottiene l'id di accesso, la password, e i valori di porta dall'attributo `EventDir` del metaoggetto `JText` di livello superiore.
  - Se questo attributo contiene un URL FTP che include il login, password, ed i valori della porta, il connettore utilizza i valori specificati in questo attributo e sovrascrivere quelli specificati nell'attributo `EventDir` del metaoggetto `JText` del livello superiore.

Per ulteriori informazioni, consultare "Specifica di un file system FTP remoto" a pagina 68.

Figura 2 illustra un esempio dell'oggetto aziendale Customer che contiene un metaoggetto secondario dinamico.

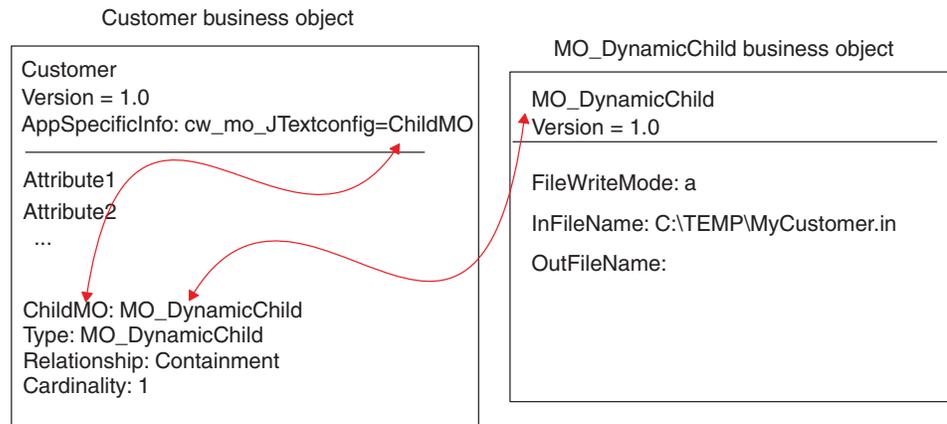


Figura 2. Esempio di un metaoggetto secondario dinamico

## Funzionamento del connettore

Il connettore JText comunica con l'applicazione tramite lo scambio del testo o dei file binari. Esegue le attività principali durante l'elaborazione degli oggetti business:

- Notifica eventi
- Elaborazione di una richiesta

Questa sezione descrive queste attività. Spiega inoltre come funziona l'elaborazione dei gestori dati e come il connettore JText elabora i verbi.

### Notifica evento

Il connettore JText gestisce gli eventi in modo diverso da altri connettori. Diversamente dai connettori che dipendono dalle applicazioni di terze parti, il connettore JText non ha una tabella degli eventi. Invece, tratta la directory di evento come tabella di evento.

Le seguenti operazioni si verificano quando il connettore JText gestisce gli eventi:

1. Il connettore esegue il poll per gli eventi controllando le directory specificate per i file con estensioni specificate. La presenza di un file con le estensioni specificate nella directory viene considerata l'equivalente di un evento. Il connettore legge i file di evento direttamente dalla directory di evento senza interpretazione. Questo utilizza un metodo di analisi per determinare quale sottosezione rappresenta ogni oggetto di business. Per ulteriori informazioni, consultare "Poll per gli oggetti business specifici" a pagina 65.
2. Il connettore crea un'istanza del gestore dati (in base ai valori specificati nel metaoggetto JText per l'oggetto business di dati).
3. Il connettore chiama `getBO()` o `getBOByteArray()` sull'istanza del gestore dati e invia la stringa o la matrice di byte che rappresenta l'oggetto business. Il connettore passa ogni elemento che rappresenta l'oggetto business al gestore di dati. Quando un file rappresenta più oggetti business, il connettore invia solo

- un elemento (cioé, una rappresentazione di stringa o di matrice di byte di un oggetto business singolo), non l'intero file.
4. Il gestore dati converte la stringa o la matrice di byte in un oggetto business e lo restituisce al connettore. Il gestore dati riporta anche errori e fornisce la traccia.
  5. Il gestore dati esegue l'elaborazione del verbo predefinito. La persona che sviluppa il gestore dati deve specificare la logica per l'impostazione dei verbi, poiché il connettore non fornisce questa logica. Per informazioni su come i gestori di dati di esempio impostano i verbi, consultare la *Data Handler Guide*.
  6. Se il gestore dati rileva un errore che impedisce la creazione di un oggetto business, il connettore archivia la stringa o la matrice di byte con l'estensione `.fail`. Se il gestore dati riesce, il connettore verifica la sottoscrizione nell'oggetto business.
    - Se il connettore non sottoscrive all'oggetto business, lo scrive in un file di archivio con l'estensione `.unsub`.
    - Se il connettore sottoscrive all'oggetto business, questo invia l'oggetto business al broker di integrazione.
  7. Se il connettore invia l'oggetto business con successo al broker di integrazione, esso archivia il file con l'estensione `.success` o `.partial`, a seconda se l'oggetto business nel file di evento è riuscito o meno nell'elaborazione. Se il connettore non riesce ad inviare l'oggetto business, esso archivia il file con l'estensione `.fail`.

Il connettore JText elabora i file di evento in ordine alfabetico in base ai nomi del file. In base alla sua configurazione, il connettore JText può prendere tutti i file nella directory di evento o prendere solo quelli con un'estensione specificata. Per ulteriori informazioni, consultare "Specificare più file di evento o più directory di evento" a pagina 65. La proprietà `SortFilesOnTimestamp` consente al connettore JText di elaborare i file di evento nell'ordine delle data/ora, dal più vecchio al più recente, indipendentemente dalla loro posizione. In altre parole, il connettore JText elabora i file posizionati in directory separate nell'ordine cronologico nell'ordine delle date/ore. Per ulteriori informazioni, consultare "SortFilesOnTimestamp" a pagina 32.

La proprietà `PollQuantity` specifica il numero massimo di oggetti business che il connettore può inviare al broker di integrazione in un lotto dato. Ad esempio, presupporre che il valore di `PollQuantity` viene impostato su 5 e che ci sono due file in una directory nella quale il connettore sta eseguendo il poll. Il primo file ha quattro oggetti business e il secondo ha 12 oggetti business. Sulla prima chiamata di poll, il connettore esegue le procedure di seguito riportate:

1. Invia tutti i quattro oggetti business per il primo file, archiviando ogni oggetto business a mano a mano che li elabora.
2. Invia il primo oggetto business dal secondo file.

Sulla seconda chiamata del poll, il connettore invia dal secondo al sesto oggetto business dal secondo file. Sulla terza chiamata del poll, il connettore invia dal settimo all'undicesimo oggetto business dal secondo file. Sul quarto poll, il connettore invia l'ultimo oggetto business. Il connettore archivia ogni oggetto business dopo averlo elaborato. Se gli oggetti business in un file non riesce l'elaborazione, il connettore archivia l'intero file con l'estensione `.orig`.

Per ulteriori informazioni, consultare:

- Per utilizzare la proprietà `PollQuantity` per ottimizzare la prestazione, consultare "Regolazione delle prestazioni del connettore JText" a pagina 78.

- Per specificare la directory di evento e l'estensione, consultare "Specifica directory evento ed estensioni" a pagina 58.
- Per specificare l'elaborazione di evento, consultare "Specifica notifica evento" a pagina 58.

Figura 3 mostra un'operazione di notifica di evento (i numeri nel grafico non sono relativi alle fasi precedentemente evidenziate).

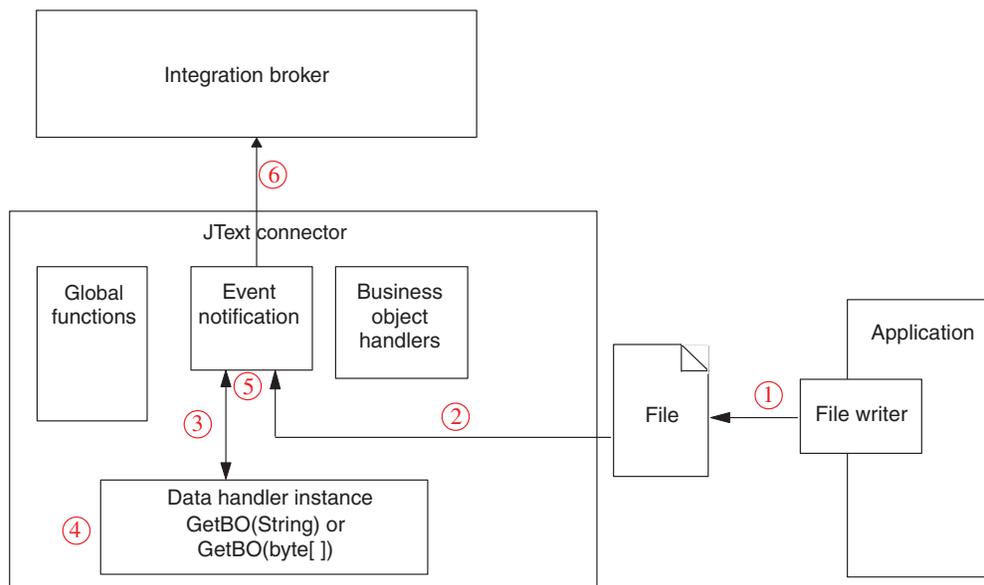


Figura 3. Operazione di notifica evento

## Archivio evento

Dopo aver elaborato un evento e se è configurato per abilitare l'archiviazione, il connettore JText scrive la stringa dell'oggetto business o la rappresentazione della matrice di dati di un oggetto business nel file nella directory di archivio locale. Denomina il file con un trattino basso (\_), una data/ora ed un'estensione file che corrisponde allo stato dell'evento. Le estensioni predefinite fornite sono success, partial, unsub, orig, e fail. Il trattino baso e la data/ora sono collegati al nome file e prima dell'estensione file.

La data/ora è un elenco separato da trattini bassi che contiene l'anno, il mese, il giorno, l'ora, il minuto, i secondi e i millisecondi dell'ora di sistema. Questo assicura che i nomi file archiviati sono univoci e che il connettore non sovrascriva un file esistente con lo stesso nome. Il formato del file archiviato è:

*BOName\_YYYY\_MM\_DD\_HH\_MM\_SS\_sss.[extension]*

Ad esempio, il connettore potrebbe ridenominare un file elaborato correttamente denominato Customer.in in Customer\_2003\_11\_15\_18\_24\_59\_999.success.

Il connettore JText archivia un oggetto business nel file .fail se si verifica un errore di formattazione, o se il connettore non riesce ad inviare l'oggetto business al broker di integrazione. Il connettore JText archivia un oggetto business ad un file con un'estensione .unsub se il connettore non lo sottoscrive. Dopo aver esaminato questi file di archivio e aver corretto gli errori di formattazione o aver avviato i processi che sottoscrivono agli oggetti business, inoltrare nuovamente gli oggetti business in questi file per l'elaborazione.

Per ulteriori informazioni sull'archiviazione, consultare "Specifiche dell'archivio di evento" a pagina 59.

## Le estensioni del file predefinito per i file di evento e di archivio

Poiché il connettore JText non utilizza tabelle di evento e di archivio, esso aggiorna lo stato di evento modificando le estensioni dei file. Tabella 1 mostra i valori dell'estensione file predefiniti che l'adattatore JText fornisce per i file di evento e archivio.

Tabella 1. Estensioni file predefinite

Tipo file	Stato/descrizione evento	Estensione file predefinita	Directory predefinita fornita
Evento	nuovo	in	Linux: /tmp/JTextConn/Default/Event Windows: C:\temp\JTextConn\Default\Event  i5/OS: Nessun predefinito fornito.
Archivio	riuscito (se tutti gli oggetti business nel file di evento sono elaborati correttamente, questo file contiene tutti gli oggetti business)	riuscito	Linux: /tmp/JTextConn/Default/Archive Windows: C:\temp\JTextConn\Default\Archive  i5/OS: Nessun predefinito fornito.
Archivio	riuscito (se alcuni oggetti business nel file di evento non riescono nell'elaborazione, questo file contiene solo quelli elaborati correttamente)	parziale	Linux: /tmp/JTextConn/Default/Archive Windows: C:\temp\JTextConn\Default\Archive  i5/OS: Nessun valore predefinito fornito.
Archivio	non sottoscritto	unsub	Linux: /tmp/JTextConn/Default/Archive Windows: C:\temp\JTextConn\Default\Archive  i5/OS: Nessun predefinito fornito.
Archivio	file di evento originale intero (creato solo se l'oggetto business non riesce nell'elaborazione o non è sottoscritto, anche se il file di evento contiene solo un oggetto business)	orig	Linux: /tmp/JTextConn/Default/Archive Windows: C:\temp\JTextConn\Default\Archive  i5/OS: Nessun predefinito fornito.
Archivio	non riuscito	non riuscito	Linux: /tmp/JTextConn/Default/Archive Windows: C:\temp\JTextConn\Default\Archive  i5/OS: Nessun predefinito fornito.

Tabella 1. Estensioni file predefinite (Continua)

Tipo file	Stato/descrizione evento	Estensione file predefinita	Directory predefinita fornita
Output	out	out	Linux: /tmp/JTextConn/Default/Out Windows: C:\temp\JTextConn\Default\Out  i5/OS: Nessun predefinito fornito.

**Importante:** La sequenza di accesso tra più applicazioni che accedono ed elaborano lo stesso file allo stesso tempo, è molto importante. Analizzare tutte le operazioni eseguite su un dato file per evitare problemi con il blocco del file e dati non completi.

**Nota:** Il connettore tratta ogni file nella directory di evento con estensione specificata come file di input. Verificare che l'estensione del file di input sia diversa dall'estensione del file di archivio, o che i file di input e di archivio sono memorizzati in directory diverse, per evitare che il connettore tratti un file archiviato come un evento.

Per informazioni sulla specifica delle estensioni file, della directory di evento e della directory di output, consultare Tabella 8 a pagina 48.

## Elaborazione richiesta

Quando si elabora una richiesta di chiamata di servizio, il connettore converte l'oggetto business in una stringa di output o in una matrice di byte, quindi lo scrive nel file.

Prima di convertire l'oggetto business, tuttavia, il connettore determina se l'oggetto business è stato configurato per la denominazione file dinamica; cioè, se l'oggetto business contiene un metaoggetto secondario dinamico. In questo caso, il connettore nomina dinamicamente il file di output o restituisce il nome del file di output che genera.

Questa sezione descrive l'elaborazione della richiesta di chiamata del servizio quando:

- "L'oggetto business di dati non specifica la denominazione file dinamica"
- "L'oggetto business di dati contiene un metaoggetto secondario dinamico" a pagina 11

### L'oggetto business di dati non specifica la denominazione file dinamica

Quando l'oggetto business di dati non specifica la denominazione file dinamica, il connettore esegue le operazioni di seguito riportate per gestire le richieste di chiamata di servizio:

1. Il connettore riceve una richiesta di oggetto business.
2. Il connettore determina che la proprietà `AppSpecificInfo` a livello oggetto business deve contenere quanto segue:

```
cw_mo_JTextConfig = DynChildMOAttrName
```

3. Il connettore controlla la configurazione del metaoggetto JText di livello superiore per determinare quale gestore dati chiamare. Per impostazione predefinita, questo metaoggetto specifica il metaoggetto del gestore dati `MO_DataHandler_DefaultNameValueConfig`, che rappresenta il gestore dati `NameValue`.
4. Il connettore crea un'istanza del gestore dati appropriato e gli invia l'oggetto `business`.
5. Il gestore dati converte l'oggetto `business` in una stringa o in una matrice di byte che restituisce alla configurazione. Il gestore dati riporta anche errori e fornisce la traccia.
6. Il connettore scrive la stringa o la matrice di byte in un file.

Per informazioni sulla configurazione del connettore per l'elaborazione delle richieste, consultare "Specifica l'elaborazione della richiesta" a pagina 61.

### **L'oggetto business di dati contiene un metaoggetto secondario dinamico**

Quando l'oggetto `business` di dati contiene un metaoggetto secondario dinamico, il connettore esegue le operazioni di seguito riportate per gestire le richieste di chiamata di servizio:

1. Il connettore riceve una richiesta di oggetto `business`.
2. Il connettore determina che la proprietà `AppSpecificInfo` a livello oggetto `business` contiene il seguente testo:

```
cw_mo_JTextConfig = DynChildMOAttrName
```

**Nota:** Se le informazioni specifiche dell'applicazione dell'oggetto `business` non specificano un metaoggetto secondario dinamico e non contengono questo elemento secondario, il connettore elabora l'oggetto `business` come descritto in "L'oggetto business di dati non specifica la denominazione file dinamica" a pagina 10.

3. Il connettore ottiene il nome del file di output dall'attributo `OutFileName` del metaoggetto secondario dinamico.
  - Se questo attributo contiene un valore, il connettore controlla se già esiste un file con quel nome. Se il file non esiste, il connettore crea un nuovo file di output, utilizzando il valore dell'attributo per denominare il file. Se il file già esiste, il connettore lo allega o lo sovrascrive al file esistente (in base al valore del `FileWriteMode` del metaoggetto secondario).

**Importante:** Se il valore dell'attributo `FileWriteMode` inizia con qualsiasi valore diverso da "o", il connettore viene impostato in modo predefinito sul modo collega.

- Se questo attributo non contiene un valore (cioè, `OutFileName=CxIgnore`), il connettore deriva il nome file dal nome dell'oggetto `business` principale che contiene il metaoggetto secondario e utilizza la configurazione del metaoggetto JText di livello superiore per determinare la posizione del file di output. Dopo aver scritto l'oggetto `business` nel file, il connettore restituisce il nome del file e il percorso in questo attributo.
4. Il connettore controlla la configurazione del metaoggetto JText di livello superiore per determinare quale gestore dati chiamare. Per impostazione predefinita, questo metaoggetto specifica il metaoggetto del gestore dati `MO_DataHandler_DefaultNameValueConfig`, che rappresenta il gestore dati `NameValue`.

5. Il connettore crea un'istanza del gestore dati appropriato e gli invia l'oggetto business.
6. Il gestore dati converte l'oggetto business in una stringa o in una matrice di byte che restituisce alla configurazione. Il gestore dati riporta anche errori e fornisce la traccia.
7. Il connettore scrive la stringa o la matrice di byte in un file. il cui nome lo deriva della fase 3 di seguito riportata.

Figura 4 illustra i componenti del connettore JText quando il connettore elabora le richieste dal broker di integrazione, InterChange Server Express, nell'applicazione di destinazione.

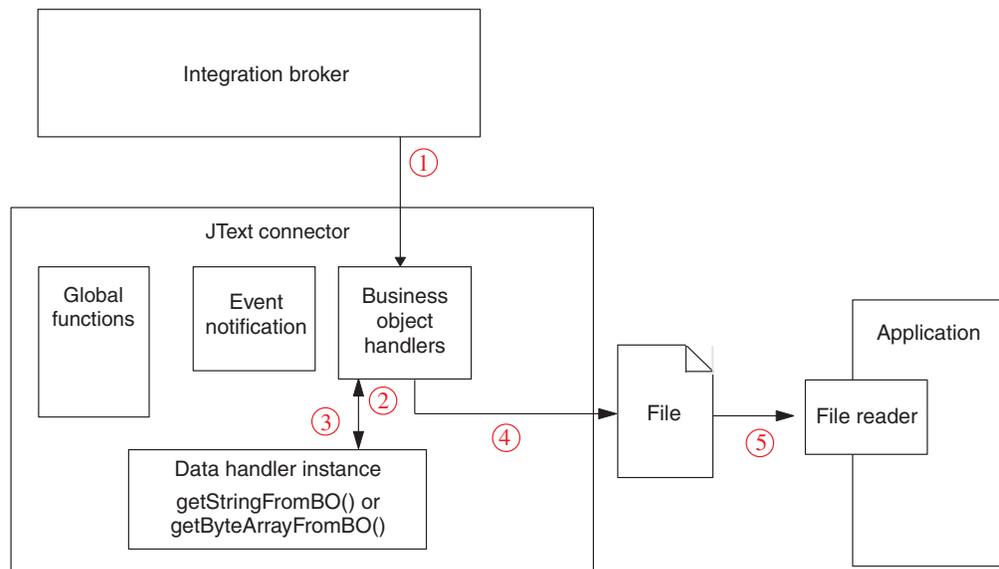


Figura 4. Operazione richiesta oggetto business

## Come funziona l'elaborazione del gestore dati

Il connettore utilizza un'istanza del gestore dati per convertire tra gli oggetti business e le stringhe o le matrici di byte che sono letti dai file di evento. L'istanza del gestore dati riporta anche gli errori e fornisce la traccia.

Il connettore crea un'istanza di un gestore dati basata sul valore degli attributi `EventDataHandler` e `OutputDataHandler` nel metaoggetto `JText` di livello superiore. Questi attributi identificano il metaoggetto del gestore che il connettore utilizza per creare l'istanza del gestore dati. Il metaoggetto del gestore dati può rappresentare un gestore dati fornito o personalizzato. Per ulteriori informazioni, consultare *Data Handler Guide*.

Il connettore determina quale interfaccia, stringa o matrice di byte, è basata sulle impostazioni del metaoggetto `DataProcessingMode`. Per ulteriori informazioni sul metaoggetto, consultare Tabella 8 a pagina 48

Dopo aver ricevuto le informazioni di configurazione, il connettore esegue le procedure di seguito riportate:

1. Istanza un gestore dati.

2. Chiama il metodo `setOption()` del gestore dati per impostare l'attributo `TracingSubSystem` del gestore dati sul nome del connettore. Il gestore dati utilizza questo valore per includere il nome del connettore nei messaggi di traccia che scrive.

Dopo aver creato e configurato il gestore dati, il connettore chiama i metodi appropriati nel gestore dati per eseguire la conversione di dati da o all'oggetto business.

- Per la notifica dell'evento, il connettore chiama il metodo `getBO(String)` o `getBO(byte[])` sul gestore dati. Il connettore passano al gestore dati la stringa da un file che è convertito in un oggetto business. Il gestore dati restituisce un oggetto business.
- Per l'elaborazione della richiesta, il connettore chiama il metodo `getStringFromBO()` o `getBytesFromBO()` sul gestore dati. Il connettore passa al gestore dati l'oggetto business da convertire in una stringa o in una matrice di byte. Il gestore dati restituisce una versione serializzata dell'oggetto business nella forma di una stringa o matrice byte.

I metodi `getBO(String)` o `getBO(byte[])` e `getStringFromBO()` o `getBytesFromBO()` inviano sempre o ricevono l'intera gerarchia dell'oggetto business di un elemento principale di livello superiore e tutti i suoi oggetti business, rispettivamente.

In entrambi i casi, il gestore dati è responsabile per il filtraggio dei dati del metaoggetto in modo che passi solo i dati specifici dell'oggetto business. I gestori dati forniti con il prodotto forniscono questa funzionalità. Se si utilizzano i gestori dati personalizzati, essi devono anche fornire questa funzionalità.

## Elaborazione verbo oggetto business per le richieste

Quando le richieste di gestione, il connettore JText non gestisce un verbo in modo diverso dall'altro. Scrive ai file senza eseguire operazioni di aggiornamento, di richiamo o di eliminazione, indipendentemente del verbo associato con l'oggetto business.

Quando si elaborano le richieste, il connettore JText imposta tutti gli attributi con un valore di `CxIgnore` sui valori predefiniti se le seguenti condizioni sono vere:

- Il verbo è `Create`.
- La proprietà `UseDefaults` del connettore è impostata su `true`.
- L'attributo è impostato su `Required`.
- I valori predefiniti sono stati impostati per gli attributi nella specifica dell'oggetto business.

---

## Funzioni connettore

Insieme alla notifica evento e alla elaborazione di richiesta dell'oggetto business, il connettore JText fornisce le seguenti caratteristiche:

- Diverse configurazioni per diversi oggetti business; ad esempio, è possibile configurare diversi oggetti business per utilizzare diverse directory ed estensioni file, o diversi formati di dati.
- Caratteristiche di configurazione per le estensioni file, l'ubicazione della directory per la memoria file di archivio, il tipo di formato e la sequenza di file.

- Caratteristiche di configurazione per la determinazione dinamica del nome file di output per ogni oggetto business, o per la restituzione del nome completo di un file di output generato.
- Errore recupero.
- Caratteristiche gestore dati personalizzato che significa che è possibile creare il gestore dati senza ricompilare il codice del connettore. E' necessario solo modificare le proprietà di configurazione per utilizzare la nuova classe creata.
- La capacità di scambiare dati con le ubicazioni FTP remote e le directory dei filesystem.

Per ulteriori informazioni, consultare Capitolo 2, "Installazione dell'adattatore JText", a pagina 15, Capitolo 4, "Utilizzo dei metaoggetti del connettore JText", a pagina 45, e *Data Handler Guide*.

## Differenze JText da altri adattatori

Mentre il connettore JText consente il trasferimento di dati da un'applicazione di origine ad un'applicazione di destinazione come altri adattatori, questo è univoco nei modi seguenti:

- Elabora tutti gli oggetti business allo stesso modo. In altre parole, poiché scrive sempre l'oggetto business in un file, esegue solo operazioni Create (indipendentemente dal verbo in entrata).
- Non interpreta i contenuti degli oggetti business che gestisce. In altre parole, legge ogni oggetto business come potenziale stringa o matrice di byte nel quale i valori di chiave non hanno più significato degli altri dati.
- Utilizza i valori dei metaoggetti per la maggior parte del suo contenuto. Per ulteriori informazioni, consultare Capitolo 4, "Utilizzo dei metaoggetti del connettore JText", a pagina 45.
- Non ha una tabella di evento. Invece, tratta la directory di evento configurata come una tabella di evento.

---

## Elaborazione dati dipendenti dalla locale

Il connettore è stato internazionalizzato in modo che possa supportare insiemi di dati double-byte, e consegnare testo di messaggio nella lingua specificata. Quando il connettore trasferisce i dati da una posizione che utilizza un insieme di codici di carattere in una posizione che utilizza un diverso insieme di codice, questo esegue una conversione di carattere per conservare il significato dei dati. L'ambiente Java runtime nella JVM (Java Virtual Machine) rappresenta i dati nell'insieme di codice di carattere Unicode. Unicode contiene le codifiche per i caratteri nella maggior parte degli insiemi di carattere conosciuti (single-byte e multibyte). La maggior parte dei componenti nel sistema WebSphere Business Integration Server Express sono scritti in Java. Quindi, quando i dati sono trasferiti tra più componenti di sistema WebSphere Business Integration Server Express, non è necessaria una conversione di carattere. Per registrare errori e messaggi di informazioni nella lingua giusta e per il paese o il territorio giusto, configurare la proprietà di configurazione standard della locale. Per ulteriori informazioni su queste proprietà, consultare "Proprietà di configurazione standard per i connettori", a pagina 91.

---

## Capitolo 2. Installazione dell'adattatore JText

Questo capitolo descrive come installare e configurare il connettore JText.

- "Panoramica delle attività di installazione"
- "Ambiente dell'adattatore"
- "Installazione dell'adattatore JText" a pagina 16
- "Verifica dell'installazione" a pagina 17

---

### Panoramica delle attività di installazione

Per installare l'adattatore per JText, è necessario eseguire le attività di seguito riportate:

- Installare il broker di integrazione. Questa attività, che include l'installazione del sistema di integrazione aziendale WebSphere e il broker di integrazione i5/OS, viene descritta nella documentazione di installazione per il broker ed il sistema operativo.
- Installare l'adattatore e i relativi file. Questa attività include l'installazione dei file per l'adattatore dal pacchetto del software sul sistema. Consultare "Installazione dell'adattatore JText" a pagina 16.

Prima di installare l'adattatore, è necessario conoscere l'ambiente dell'adattatore. Per ulteriori informazioni, consultare "Ambiente dell'adattatore".

### In questo capitolo

Le attività descritte in questo capitolo sono le seguenti:

*Tabella 2. Installazione dell'adattatore: attività roadmap*

Attività	Procedure associate (vedere...)	Per ulteriori informazioni (vedere...)
Installazione dell'adattatore	"Installazione dell'adattatore JText" a pagina 16.	<i>WebSphere Business Integration Server Express Installation Guide for Windows, for Linux, or for i5/OS</i>
Verifica dell'installazione	"Verifica dell'installazione" a pagina 17.	

---

### Ambiente dell'adattatore

Prima di installare, configurare e utilizzare l'adattatore, è necessario conoscere i requisiti ambientali:

- "Panoramica delle attività di installazione"
- "Piattaforme adattatore"
- "Prerequisiti adattatore" a pagina 16
- "Dati dipendenti dalla locale" a pagina 16

### Piattaforme adattatore

L'adattatore è supportato sulle seguenti piattaforme:

- Tutti gli ambienti del sistema operativo richiedono il programma di compilazione Java (IBM JDK 1.4.2 for Windows 2003) per la compilazione degli adattatori.
- IBM i5/OS V5R2 e OS/400 V5R3
- Windows 2003 (Standard Edition and Enterprise Edition)
- Windows XP con Service Pack 1A, per WebSphere Business Integration Adapter Framework (solo strumenti di gestione)
- 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)

## Prerequisiti adattatore

Prima di eseguire il connettore JText, creare autorizzazioni di lettura/scrittura sulle directory di evento, output, e archivio che contengono i file di testo da cui il connettore legge o su cui scrive. È necessario fare questo sui server locali e remoti.

## Dati dipendenti dalla locale

Il connettore è stato internazionalizzato in modo che possa supportare insiemi di dati double-byte, e consegnare testo di messaggio nella lingua specificata. Quando il connettore trasferisce i dati da una posizione che utilizza un codice di carattere in una posizione che utilizza un diverso insieme di dati, questo esegue una conversione di carattere per conservare il significato dei dati.

L'ambiente Java runtime nella JVM (Java Virtual Machine) rappresenta i dati nell'insieme di codice di carattere Unicode. Unicode contiene la codifica per i caratteri nella maggior parte degli insiemi di carattere conosciuti (single-byte e multibyte). La maggior parte dei componenti in nel sistema di integrazione aziendale WebSphere sono scritte in Java. Quindi, quando i dati vengono trasferiti tra più componenti di integrazione, non c'è bisogno di convertire i caratteri. Per registrare errori e messaggi di informazioni nella lingua giusta e per il paese o il territorio giusto, configurare la proprietà di configurazione standard della locale. Per ulteriori informazioni sulle proprietà di configurazione, consultare "Proprietà di configurazione standard per i connettori", a pagina 91.

Questo adattatore supporta l'elaborazione dei dati di script bidirezionali per le lingue come l'Arabo, l'Ebreo, l'Urdu, il Farsi e lo Yiddish. Per utilizzare la capacità bidirezionale, è necessario configurare le proprietà standard bidirezionali. Per ulteriori informazioni, fare riferimento alle proprietà di configurazione standard per i connettori in "Proprietà di configurazione standard per i connettori", a pagina 91.

---

## Installazione dell'adattatore JText

Per informazioni sull'installazione dei prodotti WebSphere Business Integration Server Express adapter, fare riferimento a WebSphere Business Integration Server Express - Guida all'installazione per Windows, per Linux o 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>.

---

## Verifica dell'installazione

Questa sezione di seguito riportata descrive i percorsi e i nomi file del prodotto dopo l'installazione e come verificare l'installazione dell'adattatore.

### Verifica installazione su un sistema Windows

Prima di iniziare, installare l'adattatore. Il programma di installazione copia i file standard associati con l'adattatore nel sistema. Il programma di utilità installa il connettore nella directory *ProductDir*\connectors\JText, e aggiunge un collegamento per il connettore al menu Start.

Per verificare l'installazione dell'adattatore su un sistema Windows, passare alla directory dove si è installato l'adattatore *ProductDir*\ e confrontare i contenuti a quelli elencati nella Tabella 3.

La Tabella 3 descrive la struttura del file Windows utilizzato dall'adattatore e mostra i file che sono automaticamente installati quando si sceglie di installare l'adattatore tramite il programma di installazione.

Tabella 3. Struttura del file Windows per il connettore

Sottodirectory di <i>ProductDir</i>	Descrizione
\connectors\JText\CWJText.jar	Contiene classi utilizzati solo dal connettore JText
\connectors\JText\start_JText.bat	Lo script di avvio per il connettore generico
\connectors\messages\JTextConnector.txt	File di messaggio per il connettore
\repository\JText\CN_JText.txt	Definizione repository per il connettore. Il nome predefinito è CN_JText.txt.
\connectors\JText\start_JText_service.bat	Lo script di avvio per il servizio del connettore
\connectors\JText\dependencies\commons-net-1.1.0.jar, dependencies\jakarta-oro-2.0.8.jar	Third party.jar impacchettato con il connettore

**Nota:** Tutti i nomi di percorso del prodotto sono relativi alla directory dove viene installato il prodotto sul sistema.

### Verifica dell'installazione su un sistema Linux

Prima di iniziare, installare l'adattatore. Il programma di installazione copia i file standard associati con l'adattatore nel sistema. Il programma di utilità installa l'agente del connettore nella directory *ProductDir*/connectors/JText.

Per verificare l'installazione dell'adattatore su un sistema Linux, passare alla directory dove si è installato l'adattatore *ProductDir*/ e confrontare i contenuti a quelli elencati nella Tabella 9.

La Tabella 4 descrive la struttura del file Linux utilizzata dall'adattatore, e mostra i file che vengono automaticamente installati quando si sceglie di installare l'adattatore tramite il Programma di installazione.

Tabella 4. Struttura del file di Linux per il connettore

Sottodirectory di <i>ProductDir</i>	Descrizione
/connectors/JText/CWJText.jar	Contiene classi utilizzati solo dal connettore JText
/connectors/JText/start_JText.sh	Lo script di avvio per il connettore generico
/connectors/messages/JTextConnector.txt	File di messaggio per il connettore
/repository/JText/CN_JText.txt	Definizione repository per il connettore. Il nome predefinito è CN_JText.txt.
/connectors/JText/dependencies/commons-net-1.1.0.jar, dependencies/jakarta-oro-2.0.8.jar	Third party.jar impacchettato con il connettore

**Nota:** Tutti i nomi di percorso del prodotto sono relativi alla directory dove viene installato il prodotto sul sistema.

## Verifica installazione sui sistemi i5/OS

Per impostazione predefinita, l'adattatore viene installato nella directory /QIBM/ProdData/WBIServer44/product (a cui si fa riferimento come a *ProductDir*). La Tabella 4 descrive la struttura del file i5/OS utilizzata dal connettore e mostra i file che sono automaticamente installati durante un'installazione tipica. Per verificare l'installazione dell'adattatore su un sistema i5/OS, modificare la directory dove si è installato l'adattatore e paragonare i contenuti a quelli elencati nella Tabella 5.

Tabella 5. Struttura file i5/OS per il connettore

Directory	Descrizione
ProductDir/connectors/JText	Contiene il file CWJText.jar del connettore e il file di avvio start_JText.sh.
ProductDir/repository/JText	Contiene la definizione repository BIA_CN_JText.txt.
ProductDir/connectors/JText/Dependencies	Contiene l'accordo di licenza, i file README, e i file commons-net-1.10.jar e jakarta-oro-2.0.0.jar di terza parte.
ProductDir/connectors/messages	Contiene il file JTextConnector.txt e i file JTextConnector_II_TT.Txt (file di messaggio specifico di una lingua (II) ed un paese o territorio (TT)).
ProductDir/lib	Contiene il file WBIA.jar.
ProductDir/bin	Contiene il file theCWConnEnv.sh.

**Nota:** È possibile avviare il connettore con la funzione Console. Per ulteriori informazioni, fare riferimento alla guida online fornita con la Console.

---

## Capitolo 3. Configurazione dell'adattatore JText

Questo capitolo descrive come installare e configurare il connettore JText.

- "Panoramica di Connector Configurator Express"
- "Avvio Connector Configurator Express" a pagina 20
- "Esecuzione di Configurator da System Manager" a pagina 21
- "Creazione di una maschera delle proprietà specifica del connettore" a pagina 21
- "Creazione di un nuovo file di configurazione" a pagina 24
- "Utilizzo di un file esistente" a pagina 25
- "Completamento di un file di configurazione" a pagina 26
- "Impostazione delle proprietà del file di configurazione" a pagina 27
- "Salvataggio del file di configurazione" a pagina 37
- "Modifica file di configurazione" a pagina 37
- "Completamento della configurazione" a pagina 38
- "Utilizzo Connector Configurator Express in un ambiente globalizzato" a pagina 38
- "Avvio del connettore" a pagina 39
- "Avvio del connettore" a pagina 39
- "Arresto connettore" a pagina 40
- "Creazione più istanze del connettore" a pagina 41
- "Aggiunta oggetti aziendali supportati" a pagina 43

---

### Panoramica di 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.

La modalità nella quale si esegue Connector Configurator Express, ed il tipo di file di configurazione utilizzato potrebbero essere diversi in base al broker di integrazione che si sta eseguendo.

Le proprietà della configurazione dei connettori comprendono le proprietà di configurazione standard (le proprietà possedute da tutti i 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 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 specifiche 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 21 per impostarne una nuova.

## Esecuzione dei connettori su Linux

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

Alcune proprietà in Connector Configurator Express utilizzano percorsi di directory, che per impostazione predefinita sono sui percorsi di directory di Windows. Se si utilizza il file di configurazione in un ambiente Linux, rivedere i percorsi di directory affinché questi corrispondano alle convenzioni Linux. Selezionare il sistema operativo di destinazione nell’elenco a discesa della barra degli strumenti in modo da utilizzare le giuste regole di sistema operativo per la convalida estesa.

---

## Avvio 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 in modalità silente

È possibile eseguire Connector Configurator Express senza eseguire System Manager e lavorare con i file di configurazione dei connettori.

A tale scopo:

- **Start>Tutti i programmi**, fare clic su **IBM WebSphere Business Integration Express>Toolset Express>Development>Connector Configurator Express**.
- Selezionare **File>Nuovo>Configurazione del connettore**.
- Quando si fa clic sul menu a discesa accanto a **System Connectivity Integration Broker**, è possibile 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 “Completamento di un file di configurazione” a pagina 26.)

---

## Esecuzione di Configurator 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**, è possibile 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 specifica del 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 21.
- 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.

### Specifica 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
  - Sottotipo prototipo
  - Metodo aggiornato
  - Descrizione
- **Flag**
  - Flag standard
- **Flag personalizzato**
  - Flag

Il **Sottotipo proprietà** può essere selezionato quando il **Tipo di proprietà** è una Stringa. È un valore facoltativo che fornisce il controllo della sintassi quando si salva il file di configurazione. Il valore predefinito è uno spazio vuoto, e significa che la proprietà non è un sottotipo.

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

### Specifica 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. Fare clic con il pulsante destro del mouse sul quadrato a sinistra dell'intestazione della colonna **Valore**.
2. Dal menu a comparsa, selezionare **Aggiungi** per visualizzare la casella di dialogo **Valore proprietà**. In base al tipo di proprietà, la casella di dialogo consente di immettere un valore, o un valore e un intervallo.
3. Immettere il valore della nuova 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.

### Nomi percorso impostazione

Alcune regole generali per i nomi percorso delle impostazioni sono:

- La lunghezza massima di un nome file in Windows, Linux, e i5/OS è di 255 caratteri.
- In Windows, il nome percorso assoluto deve seguire il formato [Drive:][Directory]\filename: ad esempio, C:\WebSphereAdapters\bin\Data\Std\StdConnProps.xml  
In Linux il primo carattere dovrebbe essere /.
- I nomi coda potrebbero non avere spazi di intestazione o integrati.

---

## Creazione di un nuovo file di configurazione

Quando si crea un nuovo file di configurazione, è necessario denominarlo e selezionare InterChange Server Express come broker di integrazione.

Selezionare anche un sistema operativo per una convalida estesa sul file. La barra degli strumenti ha un elenco a discesa chiamato **sistema di destinazione** che consente di selezionare il sistema operativo di destinazione per una convalida estesa delle proprietà. Le opzioni disponibili sono: Windows, Linux, i5/OS, o Altre (se non Windows o Linux o i5/OS); e convalida estesa diversa da None (disattiva convalida estesa). Il valore predefinito all'avvio è Windows.

Per avviare Connector Configurator Express:

- Nella finestra di System Manager, selezionare **Connector Configurator Express** dal menu **Strumenti**. Connector Configurator Express si apre.
- In modalità silente, avviare Connector Configurator Express.

Per impostare il sistema operativo per una convalida estesa del file di configurazione:

- Aprire l'elenco a discesa **Sistema di destinazione**: dalla barra dei menu.
- Selezionare il sistema operativo su cui si è in esecuzione.

Quindi selezionare **File>Nuovo>Configurazione del 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. Completare i campi rimanenti nella finestra **Nuovo connettore**, come descritto successivamente in questo capitolo.

## Creazione di un file di configurazione da una maschera specifica di un connettore

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

1. Impostare il sistema operativo per la convalida estesa del file di configurazione utilizzando l'elenco a discesa **Sistema di destinazione**: sulla barra di menu (vedere "Creazione di un file di configurazione" di sopra riportato).
2. Fare clic su **File>Nuovo>Configurazione del connettore**.

3. 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**.
4. 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 completare la definizione ora, o è possibile salvare il file e completare i campi successivamente.
5. 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 navigate 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.
6. 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 di 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**.

---

## Completamento di 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 visualizza le proprietà del connettore associate al broker selezionato. La tabella mostra **Nome proprietà**, **Valore**, **Tipo**, **Sottotipo** (se il Tipo è una stringa), **Descrizione**, e **Aggiorna metodo**.

3. È possibile salvare il file adesso oppure completare i restanti campi di configurazione, come descritto in "Specifiche di definizioni di oggetti di business supportati" a pagina 33..

4. 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 sono aperte più configurazioni connettore, 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 connettore nel progetto del System Manager.

Prima di creare un file di configurazione, è stato utilizzato un elenco a discesa **sistema di destinazione** che consente di selezionare il sistema operativo di destinazione per una convalida estesa delle proprietà.

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.

Se si è scelto di utilizzare la funzione di convalida estesa selezionando un valore di Windows, Linux, i5/OS, o Altro dall'elenco a discesa **sistema di destinazione**, il sistema convalida il sottotipo della proprietà e il tipo, e visualizza un messaggio di avvertenza se la convalida non riesce.

---

## 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 i valori per le proprietà in queste categorie per i connettori:

- Proprietà standard
- Proprietà specifiche dei connettori
- Oggetti di business supportati
- Valori di traccia/Log del file
- Gestore dati (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 gestore dati 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)
- Sicurezza

**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. Queste proprietà sono descritte nell'appendice Proprietà standard. 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 proprietà connettore standard

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.

**Nota:** Se la proprietà ha un Tipo di stringa, potrebbe avere un valore di sottotipo nella colonna Sottotipo. Questo sottotipo è utilizzato per una convalida estesa della proprietà.

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>Esci** (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.

Per ottenere ulteriori informazioni su una proprietà standard particolare, fare clic con il pulsante sinistro del mouse sulla voce nella colonna Descrizione per quella proprietà nel foglio a tabella Proprietà standard. Se si è installato Extended Help, appare un pulsante a freccia sulla destra. Quando si fa clic sul pulsante, si apre una finestra Guida e vengono visualizzati dettagli della proprietà standard.

**Nota:** Se il pulsante non appare, non è stata trovata nessuna Guida estesa per quella proprietà.

Se installati, i file Guida estesa si trovano nella directory  
<ProductDir>\bin\Data\Std\Help\<RegionalSetting>\.

## Impostazione delle proprietà di configurazione specifiche del connettore

Per le proprietà di configurazione specifiche del connettore, è possibile aggiungere o modificare i nomi delle proprietà, i valori di configurazione, eliminare una proprietà o 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.

**Nota:** Se la proprietà ha un Tipo di stringa, è possibile selezionare un sottotipo dall'elenco a discesa Sottotipo. Questo sottotipo è utilizzato per una convalida estesa della proprietà.

3. Per crittografare una proprietà, selezionare la casella **Crittografa**.
4. Per ottenere altre informazioni su una proprietà particolare, fare clic con il pulsante sinistro sulla voce nella colonna Descrizione per quella proprietà. Se si ha installata la guida estesa, appare un pulsante. Quando si fa clic sul pulsante, si apre una finestra Guida e vengono visualizzati dettagli della proprietà standard.

**Nota:** Se il pulsante non appare, non è stata trovata nessuna Guida estesa per quella proprietà.

5. Scegliere se salvare o eliminare le modifiche, come descritto per "Impostazione proprietà connettore standard" a pagina 28.

Se i file Guida estesa sono installati e la proprietà AdapterHelpName è vuota, Connector Configurator Express punta ai file Guida estesa specifici dell'adattatore in <ProductDir>\bin\Data\App\Help\<RegionalSetting>. Altrimenti, Connector Configurator Express punta i file specifici dell'adattatore Guida estesa posizionati in <ProductDir>\bin\Data\App\Help\<AdapterHelpName>\<RegionalSetting>. Consultare la proprietà AdapterHelpName descritta nell'appendice Proprietà standard.

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.

### Proprietà specifiche del connettore

Le proprietà di configurazione specifiche dei connettori forniscono informazioni necessarie dal connettore al runtime. Questi forniscono inoltre un modo di modificare le informazioni statiche o logiche nel connettore senza dover ricodificare e ricreare il connettore.

La tabella 6 elenca le proprietà di configurazione specifiche del connettore per il connettore. Vedere le sezioni che seguono per spiegazioni delle proprietà.

Tabella 6. Proprietà di configurazione specifiche del connettore

Nome	Valori possibili	Valore predefinito	Obbligatorio?
ArchivingEnabled	true oppure false	Vero	Sì
EventLog	Nome e posizione del file	event.log	No
EventRecovery	arresta o riprova	riprova	Sì
FTPPollFrequency	numero di cicli di poll		No
GenerateTemplate	BOName		No
OutputLog	File che registra il numero di sequenza successiva per ogni oggetto business in entrata durante l'elaborazione della richiesta	Output.Log	No
PollQuantity	Numero di eventi elaborati a ogni poll	25	No
SortFilesOnTimestamp	true oppure false	Falso	No
NoPoll	true oppure false	Falso	No

**ArchivingEnabled:** Attiva l'archiviazione. Se questa proprietà è impostato su true, il file di evento viene archiviato nella directory di archivio con l'estensione specificata. Se la proprietà viene impostata su false, il file di evento non è archiviato. In questo caso, il connettore elimina il file dopo aver inviato tutti gli eventi al broker di integrazione. Per ulteriori informazioni, consultare "Specifica dell'archivio di evento" a pagina 59.

Il valore predefinito è true.

**EventLog:** Fornisce ubicazione di memorizzazione file per gli eventi generati dal connettore. Questo file si trova nella sottodirectory JText nella directory connectors dove è installato il prodotto.

Il valore predefinito è event.log.

**EventRecovery:** Specifica il funzionamento di recupero. Se questa proprietà è impostata su riprova, il connettore utilizza il file event.log per recuperare gli eventi con errori. Se questa proprietà è impostata su arresta, il connettore termina quando si rileva un evento non riuscito. Per ulteriori informazioni, consultare "File log di evento" a pagina 82.

Il valore predefinito è riprova.

**FTPPollFrequency:** Determina quanto frequentemente il connettore esegue il poll di un server FTP misurato nel numero di cicli di poll standard. Ad esempio, se la proprietà di configurazione standard `PollFrequency` è impostata su 10000, e `FTPPollFrequency` è impostato su 6, il connettore esegue il poll della directory eventi locale ogni 10 secondi e della directory remota ogni 60 secondi. Il connettore esegue il poll del FTP solo se si specifica un valore per questa proprietà. Se `FTPPollFrequency` valuta 0 o vuoto, il connettore non esegue il poll di FTP. Per impostazione predefinita non lo fa.

Non c'è un valore predefinito per questa proprietà.

**GenerateTemplate:** Abilita il connettore a generare una maschera per ogni oggetto business supportato dopo l'avvio del connettore. La sintassi per questa proprietà è `BOName;BOName` dove il nome di un oggetto business specifico viene sostituito per `BOName`. Ad esempio, per generare due maschere, una per un oggetto business Customer e uno per un oggetto business Item, specificare `Customer;Item`. Per ulteriori informazioni, consultare "Generazione degli oggetti business di esempio per la verifica" a pagina 79.

Non c'è un valore predefinito per questa proprietà.

**OutputLog:** Specifica il nome del file che memorizza il numero della sequenza che il connettore utilizza per creare file di output singoli per ogni tipo di oggetto business durante l'elaborazione della richiesta. Il formato del file è:

*BusinessObjectName = NextSequenceNumber*

dove *BusinessObjectName* è il nome dell'oggetto business di richiesta e *NextSequenceNumber* rappresenta il numero di sequenza dell'oggetto business ricevuto più di recente, incrementato di uno. Ad esempio, se il connettore sta elaborando oggetti business Customer e Item, il file di log di output potrebbe contenere quanto segue:

```
Customer = 12  
Item = 2
```

Questo file indica che il connettore ha già elaborato 11 Customer e 1 Item. Il prossimo oggetto business Customer e Item sarà scritto nei file `Customer_12.out` e `Item_2.out`, rispettivamente. Quando riceve un oggetto business Order di richiesta, il connettore aggiunge una nuova riga al file di log di emissione e scrive l'oggetto business al file `Order_1.out`.

Se il metaoggetto `FileSeqEnabled` è impostato su `true`, il connettore utilizza questo numero di sequenza per nominare unicamente i file di emissione che crea per ogni oggetto business. Il connettore denomina ogni file di emissione facendo seguire un trattino basso (`_`) ed il numero di sequenza al nome dell'oggetto business o ad un file il cui nome è specificato nell'attributo del metaoggetto `OutputFileName`. Poiché il log di emissione viene archiviato nel formato lettura utente, è possibile utilizzare un editor di testo standard per leggere il file o per impostare il suo valore.

Per ulteriori informazioni sull'attributo `OutputFileName`, consultare "Specifica del nome del file di output" a pagina 56. Per ulteriori informazioni sul log di emissione, consultare "Specifica l'elaborazione della richiesta" a pagina 61. Per ulteriori informazioni sulla restituzione del nome del file generato, consultare "Restituzione del nome file" a pagina 62.

Il valore predefinito è `Output.Log`.

**PollQuantity:** Specifica il numero di eventi da elaborare per ogni poll. Il metodo del poll del connettore richiama il numero specificato di record di eventi e li elabora in un singolo poll. L'elaborazione di più eventi per poll può migliorare la prestazione quando l'applicazione genera grandi numeri di eventi. Tuttavia, poiché le richieste del broker di integrazione sono bloccate mentre il metodo del poll sta elaborando gli eventi, non impostare gli eventi su un numero troppo alto. Se ogni chiamata di poll impiega molto tempo, questa ritarda le operazioni di richiesta del broker di integrazione. Per ulteriori informazioni, consultare "Regolazione delle prestazioni del connettore JText" a pagina 78.

Il valore predefinito è 25.

**SortFilesOnTimestamp:** Consente all'adattatore di prendere i file di evento in base alla data/ora. Questa proprietà è idealmente impostata su true quando ci sono solo pochi grossi file di evento presi dall'adattatore durante il poll. Quando ci sono molti piccoli file di evento, questo valore dovrebbe essere impostato su false per evitare il tempo eccessivo impiegato per ordinare i file ad ogni poll.

Il valore predefinito è False.

**Nota:** L'ordinamento in base alla data/ora è supportato solo sulle piattaforme Windows, i5/OS, e Linux. L'ordinamento in base alla data/ora non è supportato sulle piattaforme MVS. Inoltre, l'adattatore JText può eseguire il poll dei file remoti sulla data/ora dal sito FTP solo se il server FTP supporta il comando `mdtm` (modifica data e ora).

**NoPoll:** Se questa proprietà è impostata su true, allora l'adattatore non esegue l'elaborazione dell'evento e l'adattatore può essere solo utilizzato per l'elaborazione della richiesta.

Il valore predefinito è False.

### **Crittografia delle proprietà del connettore**

Le proprietà specifiche dell'applicazione possono essere crittografate selezionando la casella di controllo **Crittografia** nella finestra Proprietà specifiche del connettore. Per decrittografare un valore, fare clic per eliminare i valori dalla casella di controllo **Crittografia** 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 **Crittografia** per il primo valore della proprietà. Quando si seleziona **Crittografia**, 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 decrittati.

### **Metodo aggiornato**

Fare riferimento alle descrizioni dei metodi di aggiornamento individuati nell'appendice Proprietà standard.

## 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

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 business e le relative mappe associate che sono al momento attive 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**.

- **Binding 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 su InterChange Server Express.
5. Riavviare il server per rendere definitive le modifiche.

## Risorse per InterChange Server Express

La scheda **Risorse** consente di impostare un valore che determina se e in che misura l'agente del connettore gestirà più processi contemporaneamente, utilizzando il parallelismo di agenti del connettore.

Non tutti connettori supportano questa funzione. Se si sta eseguendo un agente connettore che è stato creato in Java per essere multithreading, si viene informati di non utilizzare questa funzione, poiché di solito è più efficiente utilizzare più thread che processi multipli.

## Messaggistica per InterChange Server Express

La scheda **Messaggistica** consente di configurare le proprietà di messaggistica. Le proprietà di messaggistica sono disponibili solo se JMS è stato impostato come valore della proprietà standard `DeliveryTransport` e InterChange Server Express come tipo broker. Tali proprietà influenzano su come il connettore utilizzerà le code.

### Convalida delle code di messaggistica

Prima di poter convalidare una coda di messaggistica, è necessario:

- Assicurarsi che sia installato WebSphere MQ Series.
- Creare una coda di messaggistica con il canale e la porta su una macchina host.
- Impostare una connessione sulla macchina host.

Per convalidare la coda, utilizzare il pulsante **Convalida** a destra dei campi tipo di messaggistica e Nome host sulla scheda Messaggistica.

## Sicurezza per InterChange Server Express

È possibile utilizzare la scheda **Sicurezza** in Connector Configurator Express per ottenere diversi livelli di protezione per un messaggio. È possibile utilizzare questa funzione solo quando la proprietà `DeliveryTransport` è impostata su JMS.

Per impostazione predefinita, la Riservatezza viene attivata. Selezionare la casella **Riservatezza** per abilitarla.

Il **Nome percorso assoluto del sistema di destinazione keystore** è:

- Per Windows:  
`<ProductDir>\connectors\security\<connectorname>.jks`
- Per Linux:  
`/ProductDir/connectors/security/<connectorname>.jks`
- Per i5/OS:  
`/ProductDir/Connectors/security/<connectoryname>.jks`

Questo percorso e questo file dovrebbero essere sul sistema dove si pensa di avviare il connettore, cioè, il sistema di destinazione.

È possibile utilizzare il pulsante Sfoglia alla destra solo se il sistema di destinazione è quello al momento in esecuzione. Diventa grigio a meno che **Riservatezza** non viene abilitata e il **Sistema di destinazione** nella barra del menu viene impostato su Windows.

Il **Livello di protezione del messaggio** potrebbe essere impostato nel modo seguente per le tre categorie di messaggi (Tutti i messaggi, Tutti i messaggi amministrativi e Tutti i messaggi di oggetto business):

- "" è il valore predefinito; utilizzato quando non è stato impostato nessun livello di riservatezza per una categoria di messaggio.
- nessuno  
Non lo stesso di quello predefinito: utilizzarlo per impostare deliberatamente un livello di riservatezza su none per una categoria di messaggio.
- integrità
- riservatezza
- integrity\_plus\_privacy

La funzione **Manutenzione chiave** consente di generare, importare ed esportare le chiavi pubbliche per il server e l'adattatore.

- Quando si seleziona **Genera chiavi**, la casella di dialogo Genera chiavi appare con i valori predefiniti per il keytool che genera la chiave.
- Il valore keystore per impostazione predefinita viene impostata sul valore immesso in **Nome percorso assoluto del sistema di destinazione keystore** sulla scheda Sicurezza.
- Quando si seleziona OK, le voci vengono convalidate, il certificato di chiave viene generato e l'output viene inviato alla finestra di log del Connector Configurator Express.

Prima di poter importare un certificato nel keystore dell'adattatore, è necessario esportarlo dal keystore del server. quando si seleziona **Esporta chiave pubblica adattatore**, appare la casella di dialogo Esporta chiave pubblica adattatore.

- Il certificato di esporta si imposta in modo predefinito sullo stesso valore del keystore, fatta eccezione per il fatto che l'estensione del file è <filename>.cer.

Quando si seleziona **Importa chiave pubblica del server**, appare la casella di dialogo Importa chiave pubblica del server.

- Il certificato di importazione viene impostato in modo predefinito su <ProductDir>\bin\ics.cer (se il file esiste sul sistema).
- L'importazione dell'associazione di certificato dovrebbe essere il nome server. Se un server viene registrato, è possibile selezionarlo dall'elenco a discesa.

La funzione **Controllo accesso adattatore** viene abilitata solo quando il valore di DeliveryTransport è IDL. Per impostazione predefinita, l'adattatore accede con l'identità ospite. Se la casella **Utilizza identità ospite** non viene selezionata, i campi **Identità adattatore** e **Password adattatore** sono abilitati.

## 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 registrazione e di traccia di quel 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 gestori dati è disponibile per la configurazione solo se si è designato un valore JMS per DeliveryTransport e un valore JMS per ContainerManagedEvents. Non tutti gli adattatori utilizzano i gestori dati.

Consultare 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 selezionata durante la configurazione. La barra del titolo di 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 viene salvato in `\WebSphereAdapters\bin\Data\App`.
- È inoltre possibile salvarlo in un progetto WebSphere Application Server se si è già impostato uno.

Per i dettagli relativi all'utilizzo dei progetti System Manager, e per ulteriori informazioni sulla distribuzione, consultare la *Guida all'implementazione di sistema*.

---

## Modifica 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.

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 della finestra 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 avvio 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.

---

## Utilizzo 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 non Inglesi 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>
```

```

        <DefaultValue>en_US</DefaultValue>
    </ValidValues>
</Property>

```

---

## 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: *ProductDir\connectors\connNamedove connName* identifica il connettore.

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

Sui sistemi i5/OS lo script di avvio dovrebbe risiedere nella directory */QIBM/UserData/WBIServer44/<instance>/connectors/<ConnInstance/* con la quale viene eseguito il connettore.

Il nome dello script di avvio dipende dal sistema operativo come mostrato in Tabella 7.

Tabella 7. 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 **Avvio**, 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 [-configFile ]`

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 dove è installato WebSphere Business Integrations Server Express Console, selezionare **IBM WebSphere Business Integration Server Express>Toolset Express>Administrative>Console**. Quindi specificare il nome del sistema OS/400 o i5/OS o l'indirizzo IP ed un profilo utente e una password che ha autorizzazione \*JOBCTL speciale. Selezionare il connettore dall'elenco di connettori, quindi fare clic su Avvio
- Per avviare automaticamente l'adattatore utilizzando la Console, utilizzare lo script `submit_adapter.sh`. Questo è l'unico modo per cui l'adattatore si avvia utilizzando il sottosistema nella voce del lavoro di avvio automatico 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 valore predefinito `QWIBDFT44`), `pathToConnNameStartScript` è il percorso completo allo script di avvio del connettore, `jobDescriptionName` è il nome della descrizione del lavoro 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 ulteriori informazioni su come avviare un connettore, incluso le opzioni di avvio della riga comandi, fare riferimento a `.System Administration Guide`.*

---

## Arresto connettore

Il modo in cui si deve arrestare il connettore dipende dal modo in cui è stato avviato il connettore.

- Windows:
  - È possibile richiamare lo script di avvio che crea una finestra "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`, allora è possibile utilizzare uno dei due metodi per arrestare il connettore.
  - Dal sistema Windows dove è installato WebSphere Business Integration Server Express Console, selezionare **IBM WebSphere Business Integration Express>Toolset Express>Console>di gestione**. Quindi specificare il nome del sistema OS/400 o i5/OS o l'indirizzo IP ed un profilo utente e una password che ha autorizzazione \*JOBCTL speciale. Selezionare l'adattatore `JText` dall'elenco e selezionare il pulsante Arresta. Utilizzare il comando `CL Command WRKACTJOB SBS (QWIBSVR44)` per visualizzare i lavori del 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 `JText` il nome

lavoro sarà QWBIJTEXTC. 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 termina quando il sottosistema QWBISVR44 è terminato.

- Se si è usato lo script `start_connName.sh` script per avviare l'adattatore dalla QSHELL, premere F3 per arrestare il connettore. È possibile arrestare l'agente, utilizzando uno script denominato `stop_adapter.sh` posizionato 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 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_JText.bat connName serverName`.

- **Per le piattaforme i5/OS:**

`/QIBM/UserData/WBIServer44/WebSphereICSName/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 qui il file:

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 sottodirectory connectorInstance di ProductDir\repository.

## Creazione di una definizione del connettore

Occorre creare un file di configurazione (definizione del connettore) per l'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 la script di avvio nella directory del connettore creata in "Creazione di definizioni di oggetti di business".
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(QWBIIJTEXT) FROMLIB(QWBISVR44)OBJTYPE(\*JOB)  
TOLIB (QWBISVR44) NEWOBJ(newjtextname) dove newjtextname è un nome a 10 caratteri utilizzato per la descrizione del lavoro per il nuovo connettore.
6. (Solo per i5/OS.) Aggiungere il nuovo connettore alla console di WebSphere Business Integration Server Express. Per informazioni sulla console di WebSphere Business Integration Server Express, fare riferimento alla guida in linea fornita con la console.

---

## Aggiunta oggetti aziendali supportati

Per impostazione predefinita, il connettore JText supporta i metaoggetti `MO_JTextConnector_Default` e `MO_DataHandler_Default`. Per configurare completamente il connettore, utilizzare Connector Configurator Express per aggiungere altri oggetti aziendali all'elenco di oggetti aziendali supportati. In base a come si utilizza il connettore, potrebbe essere necessario aggiungere tutti o molti dei seguenti oggetti aziendali:

- Il metaoggetto per il gestore dati (specificato negli attributi `EventDataHandler` e `OutputDataHandler` del metaoggetto `MO_JTextConnector_Default`). Per impostazione predefinita, questi attributi specificano il metaoggetto del gestore dati `MO_DataHandler_DefaultNameValueConfig` che rappresenta il gestore dati `NameValue`. Per ulteriori informazioni, consultare "Specificare un gestore dati" a pagina 76.
- `MO_JTextConnector_BObjectName` — se si creano i metaoggetti per oggetti business specifici. Per ulteriori informazioni, consultare "Creazione di un metaoggetto JText per un oggetto business specifico" a pagina 76.
- Gli oggetti business che devono essere letti da o scritti in un file. Per ulteriori informazioni, consultare "Oggetti business utilizzati dal connettore JText" a pagina 3.



---

## Capitolo 4. Utilizzo dei metaoggetti del connettore JText

I **metaoggetto** sono oggetti business di WebSphere Business Integration Adapters che contengono informazioni di configurazione utilizzate da un connettore o da un gestore dati. Il connettore JText richiede che ognuno degli oggetti aziendali supportati abbia un metaoggetto JText associato per quel tipo di oggetto business. Questo metaoggetto di livello superiore contiene almeno un metaoggetto secondario.

- Il connettore utilizza il metaoggetto JText di livello superiore per ottenere le informazioni di configurazione come quale gestore dati utilizzare per la conversione dei dati, i percorsi dell'evento di oggetto aziendale, l'archivio e le directory di output, le estensioni file per l'evento, l'archivio, i file di output, le informazioni richieste se il connettore sta elaborando file su un sistema FTP e se il connettore genera identificativi file univoci per i file di output.
- Il connettore utilizza un metaoggetto secondario per specificare i valori di configurazione che il gestore dati deve utilizzare quando converte i dati tra un oggetto business e una stringa o una matrice di byte. Per impostazione predefinita, il metaoggetto di livello superiore specifica il gestore dati NameValue per convertire i dati.

Per fornire diverse informazioni di configurazione per ogni oggetto business che il connettore supporta, è possibile creare un metaoggetto JText di livello superiore per ognuno. Poiché ogni metaoggetto di livello superiore specifica il proprio metaoggetto di gestore dati, il connettore può elaborare ogni tipo di oggetto business in un formato diverso. Il metaoggetto del gestore dati elimina il bisogno di modificare una definizione di oggetto business o di modificare il connettore stesso quando si introducono nuovi formati di dati o si eseguono modifiche ai formati esistenti.

I metaoggetti sono caricati nella memoria all'avvio, rendendo le informazioni di configurazione disponibili al connettore. Notare che i metaoggetti vengono inviati al broker di integrazione per l'elaborazione. Essi influenzano il funzionamento solo del connettore.

Questo capitolo descrive come configurare il connettore JText utilizzando i metaoggetti JText. Per informazioni sull'utilizzo dei metaoggetti del gestore dati, consultare *Data Handler Guide*. Gli argomenti inclusi in questo capitolo includono:

- "convenzioni di denominazione del metaoggetto JText"
- "Struttura del metaoggetto JText" a pagina 46
- "Attività di configurazione comune" a pagina 58

---

### convenzioni di denominazione del metaoggetto JText

Il nome del metaoggetto JText di livello superiore ha tre componenti, come illustrato dal nome del metaoggetto di livello superiore predefinito, `MO_JTextConnector_Default`. I componenti di un nome metaoggetti JText di livello superiore sono i seguenti:

- `MO_` è un prefisso che indica un metaoggetto.
- `ConnectorInstanceName_` specifica il nome dell'istanza del connettore, come JText. Questo nome è configurabile per supportare l'utilizzo di più istanze di

connettore. Ad esempio, un connettore denominato `JText2` potrebbe avere un metaoggetto denominato `MO_JText2Connector_Default`.

- `Default` specifica il nome dell'oggetto business associato. Per creare un metaoggetto per un oggetto business specifico, modificare la stringa `Default` nel nome dell'oggetto business, come in `MO_JTextConnector_Customer` per un oggetto business denominato *Customer*. È possibile includere altri componenti e trattini bassi nel nome del metaoggetto. Il connettore utilizza i metaoggetti predefiniti se non esistono metaoggetti specifici dell'oggetto aziendale.

Per informazioni sulla creazione di metaoggetti per un oggetto business specifico, consultare "Creazione di un metaoggetto `JText` per un oggetto business specifico" a pagina 76.

---

## Struttura del metaoggetto `JText`

Un metaoggetto `JText` ha una struttura gerarchica. Il metaoggetto di livello superiore predefinito è denominato `MO_JTextConnector_Default`. Due attributi del metaoggetto di livello superiore, `EventDataHandler` e `OutputDataHandler`, rappresentano i metaoggetti secondari che forniscono informazioni di configurazione per il gestore dati che il connettore utilizza. Il connettore utilizza il gestore dati per convertire i dati tra gli oggetti aziendale e le stringhe o le matrici di byte.

Per impostazione predefinita, questi attributi specificano lo stesso metaoggetti del gestore dati (`MO_DataHandler_DefaultNameValueConfig`). Questo metaoggetto del gestore dati chiama il gestore dati `NameValue` per convertire effettivamente i dati. In altre parole, la configurazione predefinita fornita specifica che la conversione evento e file di output utilizza lo stesso gestore dati. Per informazioni sull'istanziamento di un gestore dati, consultare *Data Handler Guide*.

**Nota:** Poiché l'utilizzo del programma di formattazione è stato rifiutato in favore dell'utilizzo del gestore dati, gli attributi `EventFormat` e `OutputFormat` che precedentemente rappresentavano un programma di formattazione sono stati rimossi dal metaoggetto `MO_JTextConnector_Default`. Per utilizzare un programma di formattazione, è necessario:

- aggiungere gli attributi `EventFormat` e `OutputFormat` al metaoggetto di livello superiore
- specificare l'oggetto business appropriato come il Tipo di questi attributi
- modificare il Tipo degli attributi `EventDataHandler` e `OutputDataHandler` su `Stringa`

Per informazioni sull'utilizzo di un programma di formattazione, consultare la documentazione per il rilascio 3.0.0 o 2.3.0 del connettore `JText`.

Figura 5 mostra la struttura gerarchica per i metaoggetti `JText` predefiniti e ogni nome e tipo di attributo.

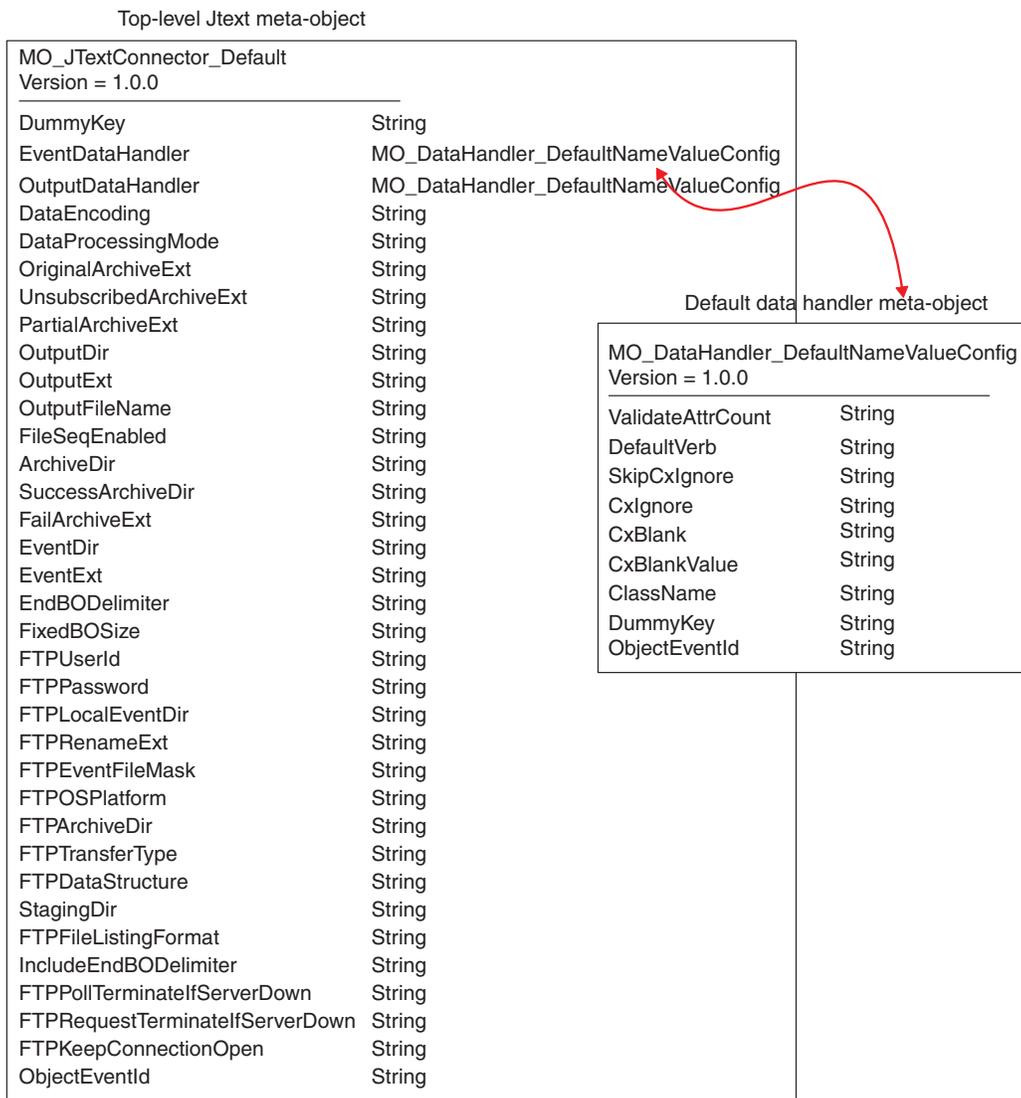


Figura 5. Struttura gerarchica del metaoggetto JText

## Creazione metaoggetti personalizzati

MO\_JTextConnector\_Default, come il metaoggetti JText di livello superiore, contiene informazioni di configurazione e metaoggetti secondari per il connettore. È possibile creare metaoggetti di livello superiore separati per ogni tipo di oggetto business che il connettore gestisce. Questi metaoggetti personalizzati possono contenere gli stessi o diversi metaoggetti secondari per configurare il tipo di gestore di dati. Ad esempio, per configurare l'elaborazione in modo diverso per gli oggetti business Customer e Item, creare i metaoggetti MO\_JTextConnector\_Customer e MO\_JTextConnector\_Item e progettare questi metaoggetti di livello superiore per contenere diversi metaoggetti del gestore dati.

All'inizializzazione, il connettore richiama un elenco di metaoggetti supportati e oggetti business dal broker di integrazione. Dai nomi di questi oggetti, il connettore determina quali oggetti business hanno i propri metaoggetti di livello

superiore associati. Al runtime, il connettore corrisponde al nome di un oggetto business di richiesta con uno dei metaoggetti supportati per posizionare le informazioni di configurazione appropriate.

Ad esempio, presupporre che il connettore supporta i seguenti metaoggetti:

- `MO_JTextConnector_Default`
- `MO_JTextConnector_Customer`
- `MO_JTextConnector_Item`

e i seguenti oggetti business:

- `Customer`
- `Item`
- `Order`

Quando il broker di integrazione invia un oggetto business `Customer` di richiesta, il connettore utilizza le informazioni di configurazione specificate nel metaoggetto `MO_JTextConnector_Customer`. quando il broker di integrazione invia un oggetto business `Order` di richiesta, il connettore utilizza le informazioni di configurazione specificate nel metaoggetto `MO_JTextConnector_Default`.

## Attributi `MO_JTextConnector_Default`

Questa sezione descrive gli attributi nei metaoggetti `MO_JTextConnector_Default`.

**Nota:** Tutti i valori in una proprietà `DefaultValue` di attributi sono sensibili al maiuscolo/minuscolo. Le informazioni di directory devono specificare il percorso assoluto di una directory.

Tabella 8 e le sezioni seguenti descrivono la funzionalità di ogni attributo nel metaoggetto `MO_JTextConnector_Default`. Tra le altre informazioni, questa tabella include il valore fornito per ogni proprietà `DefaultValue` di ogni attributo semplice. È possibile sostituire il valore fornito con il prodotto con il proprio valore.

Tabella 8. Attributi nella definizione del metaoggetto `MO_JTextConnector_Default`

Nome attributo	Descrizione
<code>ArchiveDir</code>	Specifica il percorso assoluto della directory <code>Archive</code> . La directory deve già esistere. I valori predefiniti forniti sono: Linux: <code>/tmp/JTextConn/Default/archive</code>  Windows: <code>C:\temp\JTextConn\Default\Archive</code> i5/OS: <code>/QIBM/UserData/WBIServer44/&lt;instance&gt;/connectors/JText/archive</code> BiDi supportato per Windows. Tutte le proprietà abilitate BiDi verranno trasformate dal formato Windows BiDi nel formato BiDi.Metadata, che è una proprietà standard del connettore JText.
<code>DataEncoding</code>	<code>DataEncoding</code> è la codifica da utilizzare per scrivere e leggere le stringhe di oggetti business. Se questa proprietà non viene specificata nel metaoggetto statico, il connettore prova a leggere o scrivere la stringa dell'oggetto aziendale senza utilizzando la codifica specifica. È possibile specificare la codifica supportata di Java impostato per questo attributo.

Tabella 8. Attributi nella definizione del metaoggetto *MO\_JTextConnector\_Default* (Continua)

Nome attributo	Descrizione
DataProcessingMode	Questo attributo fornisce nuova flessibilità per la lettura e la scrittura dei file binari. Quando viene impostato su Binaria, questa proprietà MO abilita JText a leggere e scrivere i file binari dal sistema di file mentre richiama l'interfaccia del gestore dati appropriata per le trasformazioni di BO nella matrice di byte e viceversa. L'impostazione tradizionale per questo è Text. Nel modo Text, viene utilizzata l'interfaccia del gestore dati di BO a String e viceversa. Quando la proprietà non è impostata viene predefinita su Text. Il modo Binary dovrebbe essere solo utilizzato con un gestore dati che appropriatamente implementa i metodi <code>getBO(byte[])</code> e <code>getByteArrayFromBO()</code> .
DummyKey	Questo attributo esiste per soddisfare il requisito che un attributo in ogni definizione oggetto business ha la proprietà Key abilitata.
EndBODelimiter	Specifica un delimitatore che separa gli oggetti business nel file di input. Per ulteriori informazioni sull'attributo <code>EndBODelimiter</code> , consultare "Poll per gli oggetti business specifici" a pagina 65. Se non si fornisce un valore predefinito durante la configurazione e <code>DataProcessingMode</code> è impostato su Text, la proprietà viene impostata in modo predefinito sul seguente valore: <code>&lt;EndBO:BOName&gt;</code> . Quando <code>DataProcessingMode</code> è binario, la proprietà è impostata in modo predefinito sul seguente valore: <code>FF01</code> . <b>Nota:</b> Poiché il gestore dati del <code>NameValue</code> viene impostato come gestore dati predefinito, il valore <code>EndBODelimiter</code> viene impostato su <code>&lt;EndBO:BOName&gt;</code> . Se si desidera utilizzare un altro gestore dati, come il gestore dati <code>Delimitato</code> , sarà necessario specificare il valore corrispondente. Per il gestore dati <code>Delimitato</code> , la stringa <code>EOL</code> è un valore <code>EndBODelimiter</code> valido.
EndOfFileDelimiter	Quando <code>DataProcessingMode</code> è binario e <code>FTPDataStructure</code> è <code>Record</code> , vengono utilizzati <code>EndBODelimiter</code> e <code>EndOfFileDelimiter</code> . Questa proprietà viene impostata sul byte esadecimale che è utilizzata per la fine del marker del file nel file di record. Se non è impostata, viene utilizzato il valore predefinito <code>FF02</code> .
EventDataHandler	Rappresenta un metaoggetto secondario i cui attributi forniscono valori di configurazione per il gestore dati da utilizzare per l'elaborazione eventi (stringa oggetto business convertito all'oggetto business). Il valore predefinito fornito è <code>MO_DataHandler_DefaultNameValueConfig</code>

Tabella 8. Attributi nella definizione del metaoggetto *MO\_JTextConnector\_Default* (Continua)

Nome attributo	Descrizione
EventDir	<p>Specifica il percorso assoluto della directory Event. La directory deve già esistere. Se si creano metaoggetti separati per diversi oggetti business diversi e si specifica lo stesso percorso EventDir per entrambi, è necessario specificare valori univoci per l'attributo EventExt in ogni metaoggetto. Per ulteriori informazioni, consultare "Specifica directory evento ed estensioni" a pagina 58. Per configurare il connettore affinché utilizzi un sistema di file FTP remoto per l'elaborazione eventi, specificare l'URL FTP in questo attributo. Facoltativamente, è possibile utilizzare questo attributo per specificare le seguenti ulteriori informazioni nell'URL:</p> <ul style="list-style-type: none"> <li>• L'id e la password di un utente con i privilegi per connettersi al server FTP ed eseguire le operazioni FTP; se non specificato in EventDir, è necessario specificarlo in FTPUserId eFTPPassword.</li> <li>• La porta FTP; se non specificata in EventDir, il connettore utilizza la porta FTP predefinita.</li> <li>• la directory evento remota; se non specificata in EventDir, il connettore esegue il poll dei file di evento dalla directory nella quale è stabilita la connessione al server FTP.</li> </ul> <p>La sintassi per specificare le informazioni FTP nell'attributo EventDir è:  <code>ftp://[UserId:password@]FTPserver[:port][RemoteEventDirectory]</code></p> <p>Per ulteriori informazioni, consultare "Elaborazione eventi remota" a pagina 68. Per specificare le informazioni di file locale nell'attributo EventDir, utilizzare il percorso completo del file. In alternativa, è possibile utilizzare un FILE URL, che utilizza il seguente formato:  <code>[file://]FullPathname</code></p> <p>I valori predefiniti forniti sono:  Linux: <code>/tmp/JTextConn/Default/event</code>  Windows:<code>C:\temp\JTextConn\Default\Event</code>  i5/OS:  <code>/QIBM/UserData/WBIServer44/&lt;instance&gt;/connectors/JText/event</code>  BiDi supportato per Windows. Tutte le proprietà abilitate BiDi verranno trasformate dal formato Windows BiDi nel formato BiDi.Metadata, che è una proprietà standard del connettore e-Mail.</p> <p><b>Nota:</b> Una nuova proprietà booleana specifica del connettore, "NoPoll", è stata introdotta per disattivare facoltativamente il poll. Il valore predefinito è false. Quando è impostato su true, l'adattatore elabora solo le richieste e non esegue il poll.</p>
EventExt	<p>Specifica l'estensione del file utilizzato per la notifica dell'evento. Se non è specificato alcun valore, il connettore JText esegue poll per i file con nessuna estensione file. Per ulteriori informazioni, consultare "Specificare più file di evento o più directory di evento" a pagina 65.</p> <p><b>Nota:</b> L'utilizzo di un asterisco (*) per questo attributo per specificare che il poll del connettore per tutti i file in una directory di evento singola indipendentemente dalla estensione, non è più supportato. Il valore predefinito fornito è in.</p>
FailArchiveExt	<p>Specifica l'estensione del file utilizzata per archiviare gli oggetti business che non sono stati elaborati correttamente. Per ulteriori informazioni, consultare "Specifica dell'archivio di evento" a pagina 59. Il valore predefinito fornito è fail.</p>
FileSeqEnabled	<p>Specifica la sequenza nome file, che emette ogni oggetto business in un file separato. Il nome del file include un numero di sequenza univoco. Per ulteriori informazioni, consultare "Specifica l'elaborazione della richiesta" a pagina 61. Il valore predefinito fornito è true.</p>

Tabella 8. Attributi nella definizione del metaoggetto *MO\_JTextConnector\_Default* (Continua)

Nome attributo	Descrizione
FixedBOSize	Quando è presente con un valore valido, questa proprietà del metaoggetto sovrascriverla proprietà <code>EndBODElimiter</code> e fornisce agli utenti un'alternativa al delimitatore tradizionale basta sull'analisi BO.
FTPArchiveDir	<p>Specifica il percorso relativo della directory di archivio sul server FTP. La directory deve già esistere. Ci sono diverse opzioni per utilizzare questo attributo in modo da specificare l'archivio:</p> <ul style="list-style-type: none"> <li>• Specificare un valore per questo attributo, ma nessun valore per l'attributo <code>FTPRenameExt</code> in modo che il connettore colleghi una data/ora al nome file dell'evento e lo sposti alla directory di archivio del server FTP in questo attributo.</li> <li>• Specificare un valore per questo attributo e l'attributo <code>FTPRenameExt</code> in modo che il connettore ridenomini il nome file dell'evento elaborato con una data/ora ed un valore specificato in <code>FTPRenameExt</code> e lo sposti nella directory di archivio del server FTP specificata in questo attributo.</li> <li>• Non specificare alcun valore per questo attributo o per l'attributo <code>FTPRenameExt</code> in modo che il connettore elimini il file di evento elaborato senza archivarlo.</li> <li>• Non specificare alcun valore per questo attributo, ma specificare un valore per l'attributo <code>FTPRenameExt</code> in modo che il connettore ridenomini il file di evento elaborato, aggiungendo una data/ora ed il valore specificato in <code>FTPRenameExt</code>, e spostandolo nella directory specificata nell'attributo <code>EventExt</code>.</li> <li>• Specificare / (barra) per questo attributo ma nessun valore per l'attributo <code>FTPRenameExt</code> in modo che il connettore sposti il file di evento elaborato nella directory root sul server FTP.</li> <li>• Specificare / (barra) per questo attributo ed un valore per l'attributo <code>FTPRenameExt</code> in modo che il connettore ridenomini il nome del file di evento elaborato con l'estensione specificato in <code>FTPRenameExt</code>, e lo sposti nella directory root sul server FTP.</li> </ul> <p>Per ulteriori informazioni, consultare "Specifica dell'archivio di evento" a pagina 59. Non c'è nessun valore fornito per questo attributo.</p> <p>BiDi supportato per Windows. Tutte le proprietà abilitate BiDi verranno trasformate dal formato Windows BiDi nel formato BiDi.Metadata, che è una proprietà standard del connettore e-Mail.</p>
FTPDataStructure	Questo attributo è del tipo Stringa. L'utente può specificare la struttura dei dati FTP (File o Record) per ottenere o inserire i file da o al sito remoto. Se non viene specificato niente, Jtext utilizza 'File' come valore predefinito

Tabella 8. Attributi nella definizione del metaoggetto *MO\_JTextConnector\_Default* (Continua)

Nome attributo	Descrizione
FTPEventFileMask	<p>Utilizza caratteri jolly integrati per specificare la maschera o il prefisso o i file FTP remoti per l'elaborazione evento. Specificare un valore per questo attributo solo per identificare la maschera del file su una struttura principale che non aderisce agli standard di denominazione che si applicano ai sistemi Windows, Linux, o i5/OS. Utilizzando i caratteri jolly nel nome file è possibile specificare più files per l'elaborazione evento. Ad esempio, è possibile utilizzare il seguente formato per specificare più file di evento: ACT.Z1UC.INPT*Per ulteriori informazioni, consultare "Identificazione file su una struttura principale: Configurazione facoltativa" a pagina 71. Non esiste alcun valore predefinito fornito.</p> <p>Se si esegue il poll, è necessario fornire una maschera molto specifica. Ad esempio, se si desidera eseguire il poll di tutti i file di evento seguenti: USER.JTEXT.TEST001.EVENT, USER.JTEXT.TEST002.EVENT, USER.JTEXT.TEST003.EVENT, USER.JTEXT.TEST004.EVENT, e FTPArchiveDir è impostato come / o lasciato vuoto, e FTPRenameExt è impostato come RENAME, quindi questi file vengono archiviati come il seguente: USER.JTEXT.TEST001.RENAME, USER.JTEXT.TEST002.RENAME, USER.JTEXT.TEST003.RENAME. Quindi, se si fornisce un FTPEventFileMask=USR.JTEXT.TEST*.*, tutti gli eventi verranno presi al primo poll. Tutti i file archiviati verranno presi al poll successivo, poiché sono anche conformi alla stessa maschera del file. Per evitare ciò, è necessario fornire una maschera molto specifica. Ad esempio, come USR.JTEXT.TEST*.EVENT, in modo che USR.JTEXT.TEST*.RENAME non sia preso durante il polling.</p> <p><b>Nota:</b> Non specificare una maschera che è applicabile per i file di evento e archiviati.</p> <p>BiDi supportato per Windows. Tutte le proprietà abilitate BiDi verranno trasformate dal formato Windows BiDi nel formato BiDi.Metadata, che è una proprietà standard del connettore e-Mail.</p>
FTPFileListingFormat	<p>Specifica il formato nel quale il connettore JText prevede che le informazioni del file appaiano durante la lettura nei file. Questo abilita il connettore a leggere nei file in diverse locali dove potrebbero essere memorizzate le informazioni di data e ora in ordini diversi nelle informazioni del formato di file. Per configurare il connettore per utilizzare il formato per la locale, specificare una serie di caratteri delimitati da un punto e una virgola che rappresentano l'ordine nel quale si verificano gli attributi del file; di seguito viene riportato un elenco che associa i caratteri possibili con gli attributi del file che rappresentano.</p> <p>P Permission L Links U User G Group S Size D Date M Month T Time N Name</p> <p>Un valore adatto per questo attributo, quindi, potrebbe essere P:L:U:G:S:D:M:T:N.</p>
FTPGetQuantity	<p>Determina il numero di file richiamati dall'URL FTP remoto con ogni poll remoto.</p>

Tabella 8. Attributi nella definizione del metaoggetto *MO\_JTextConnector\_Default* (Continua)

Nome attributo	Descrizione
FTPKeepConnectionOpen	Impostare la proprietà Valore predefinito di questo attributo sul valore true per fare in modo che il connettore JText conservi la connessione con un sito FTP. Se questo attributo è impostato sul valore true allora il connettore chiude solo la connessione quando il connettore viene arrestato o se il server FTP chiude la connessione stessa (a causa del timeout configurato, ad esempio). Il connettore verifica che la connessione è ancora attiva ogni volta che esegue un'operazione remota per gestire la situazione quando il server FTP potrebbe aver chiuso la connessione a causa del timeout. Se la connessione è stato chiuso allora il connettore lo ristabilisce. Impostare la proprietà Valore predefinito di questo attributo sul valore false per fare in modo che il connettore JText apra una connessione con il server FTP ogni volta che esegue un'operazione e chiude quando è terminata la connessione. La configurazione del connettore per mantenere la connessione attiva può migliorare la prestazione del connettore quando si esegue l'elaborazione della richiesta sul sito FTP.
FTPLocalEventDir	Specifica la directory del sistema locale nella quale il connettore scarica i file di evento dal sito FTP. È necessario specificare un valore per questo attributo per abilitare il connettore ad elaborare gli eventi utilizzando FTP. Per ulteriori informazioni, consultare "Specifiche della directory locale" a pagina 69. Non esiste alcun valore predefinito fornito. BiDi supportato per Windows. Tutte le proprietà abilitate BiDi verranno trasformate dal formato Windows BiDi nel formato BiDi.Metadata, che è una proprietà standard del connettore e-Mail.
FTPOSPPlatform	Utilizzare questo attributo solo se si configura il connettore per utilizzare un sistema di file FTP remoto quando il server FTP remoto è una piattaforma MVS. In questo caso, specificare il valore di questo attributo come MVS. Il maiuscolo/minuscolo non è significativo. Per ulteriori informazioni, consultare "Specifiche di un file system FTP remoto" a pagina 68. Non esiste alcun valore predefinito fornito.
FTPPassword	Specifica la password dell'utente che ha autorizzazioni per connettersi al server FTP ed eseguire le operazioni FTP. Non è necessario specificare un valore per questo attributo se la password è inclusa nell'URL specificato nell'attributo EventDir o OutputDir. Per ulteriori informazioni, consultare "Specifiche dell'URL FTP ed informazioni di login" a pagina 68. Non c'è nessun valore fornito per questo attributo.
FTPPollTerminateIfServerDown	Specifica il funzionamento del connettore quando è configurato per eseguire il poll del sito FTP per gli eventi ed il sito FTP non è disponibile. Se la proprietà Valore predefinito dell'attributo FTPPollTerminateIfServerDown è impostato sul valore true ed il sito FTP non è disponibile quando il connettore tenta di eseguire il poll di una chiamata, allora il connettore termina. Se la proprietà Valore predefinito dell'attributo FTPPollTerminateIfServerDown è impostato sul valore false ed il sito FTP non è disponibile quando il connettore tenta di eseguire il poll di una chiamata, allora il connettore non termina. Non esiste alcun valore predefinito fornito.
FTPRenameExt	Specifica l'estensione file o il suffisso che il connettore utilizza per ridenominare il file FTP remoto dopo che il connettore ne ha eseguito il poll. La ridenominazione del file evita che il connettore esegua il poll dello stesso file nel ciclo di poll successivo. In alternativa, è possibile configurare il connettore per ridenominare il file di evento elaborato e spostarlo in una directory di archivio. Per ulteriori informazioni, consultare l'attributo FailArchiveExt. Per ulteriori informazioni, consultare "Identificazione file su una struttura principale: Configurazione facoltativa" a pagina 71. Non esiste alcun valore predefinito fornito.

Tabella 8. Attributi nella definizione del metaoggetto *MO\_JTextConnector\_Default* (Continua)

Nome attributo	Descrizione
FTPRequestTerminateIfServerDown	Specifica il funzionamento del connettore quando è configurato per eseguire l'elaborazione della richiesta e comunicare con un sito FTP, ed il sito FTP non è disponibile. Se la proprietà Valore predefinito dell'attributo FTPRequestTerminateIfServerDown è impostato sul valore true ed il sito FTP non è disponibile quando il connettore tenta di eseguire l'elaborazione della richiesta, quindi il connettore termina. Se la proprietà Valore predefinito dell'attributo FTPRequestTerminateIfServerDown viene impostato sul valore false ed il sito FTP non è disponibile quando il connettore tenta di eseguire l'elaborazione della richiesta, allora il connettore non termina.
FTPTransferType	Non esiste alcun valore predefinito fornito . Questa proprietà del metaoggetto JText viene utilizzata durante l'elaborazione dell'evento e della richiesta. I valori possibili per questa proprietà sono Binary e ASCII. La proprietà detta il tipo di trasferimento che JText utilizza quando si posizionano in modo remoto o si richiamano i file da un server FTP. Quando la proprietà non esiste, il funzionamento dell'adattatore viene impostato in modo predefinito su Binary.
FTPUserId	Specifica il nome dell'utente che ha privilegi per connettersi al server FTP ed eseguire le operazioni FTP. Non è necessario specificare un valore per questo attributo se l' id utente è incluso nell'URL specificato nell'attributo EventDir o OutputDir. Il connettore ignora questo attributo se non trova un URL di FTP nell'attributo EventDir (durante l'elaborazione dell'evento) o l'attributo OutputDir (durante l'elaborazione della richiesta). Per ulteriori informazioni, consultare "Specifiche dell'URL FTP ed informazioni di login" a pagina 68. Non c'è nessun valore fornito per questo attributo.
IncludeEndBODelimiter	Specifica se il valore specificato per l'attributo del metaoggetto EndBODelimiter è incluso o meno nella stringa scritta in un file dal connettore JText. Se la proprietà Valore predefinito di questo attributo è impostata su true allora il connettore include il valore specificato per l'attributo EndBODelimiter quando scrive i file. Se la proprietà Valore predefinito di questo attributo è impostata su false allora il connettore non include il valore specificato nell'attributo EndBODelimiter quando scrive i file.
LargeObject	Un flag utilizzato per attivare le funzioni di ottimizzazione degli oggetti ampi dell'adattatore JText (quando è impostato su true). L'impostazione di questo flag su true risulta nella seguente modifiche nel funzionamento del programma di archiviazione: <ol style="list-style-type: none"> <li>1. Se un file di evento ha più oggetti business, l'archiviazione verrà eseguita solo dopo che tutti gli oggetti business per quel file di evento vengono elaborati.</li> <li>2. Il file originale non verrà archiviato nel caso in cui l'elaborazione degli oggetti business in quel file di evento risulti in un errore o in uno stato non sottoscritto.</li> </ol> <p>Verrà inoltre creato un file di log in più internamente per mantenere traccia dello stato di archiviazione.</p>
MVSSiteCommand	Utilizzato per emettere i comandi del sito FTP MVS. Il comando del sito dovrebbe essere specificato senza le parole chiave SITE o QUOTE. Un esempio del valore del comando del sito è: LRECL=<value> BLKSIZE=<value>, dove <value> rappresenta gli argomenti del comando del sito passati.
ObjectEventID	Placeholder non utilizzato dal connettore in un metaoggetto ma richiesto dal broker di integrazione. Questo attributo deve essere l'ultimo attributo nel metaoggetto. Non esiste alcun valore predefinito fornito.

Tabella 8. Attributi nella definizione del metaoggetto *MO\_JTextConnector\_Default* (Continua)

Nome attributo	Descrizione
OriginalArchiveExt	Specifica l'estensione del file utilizzata per archiviare il file di evento originale, che conserva l'intero file di evento per riferimento in caso gli oggetti business sbagliano l'elaborazione o non sono sottoscritti. Per ulteriori informazioni, consultare "Specifica dell'archivio di evento" a pagina 59. Il valore predefinito fornito è orig.
OutputDataHandler	Rappresenta un metaoggetto secondario i cui attributi forniscono valori di configurazione per il gestore dati da utilizzare per le richieste di chiamata del servizio (oggetto business convertito alla stringa dell'oggetto business). Il valore predefinito fornito è MO_DataHandler_DefaultNameValueConfig
OutputDir	<p>Specifica il percorso assoluto della directory Output. La directory deve già esistere. Per configurare il connettore affinché utilizzi un sistema di file FTP remoto per l'elaborazione richiesta, specificare l'URL FTP in questo attributo. Facoltativamente, è possibile utilizzare questo attributo per specificare le seguenti ulteriori informazioni nell'URL:</p> <ul style="list-style-type: none"> <li>• L'id e la password di un utente con i privilegi per connettersi al server FTP ed eseguire le operazioni FTP; se non specificato in EventDir, è necessario specificarlo in FTPUserId eFTPPassword.</li> <li>• La porta FTP; se non specificata in OutputDir, il connettore utilizza la porta FTP predefinita.</li> <li>• la directory di output remota; se non specificata OutputDir, il connettore carica i file di richiesta nella directory di connessione predefinita (la directory sul server FTP sulla quale viene stabilita la connessione).</li> </ul> <p>La sintassi per specificare le informazioni FTP nell'attributo è OutputDir:ftp://[UserId:password@]FTPserver[:port] Per ulteriori informazioni, consultare "Elaborazione richiesta remota" a pagina 72. Per specificare le informazioni di file locale nell'attributo OutputDir, utilizzare il percorso completo del file. In alternativa, è possibile utilizzare un FILE URL, che utilizza il seguente formato:</p> <p>[file://]FullPathname</p> <p>I valori predefiniti forniti sono:</p> <p>Linux: /tmp/JTextConn/Default/out</p> <p>Windows:c:\temp\JTextConn\Default\Out</p> <p>i5/OS: /QIBM/UserData/WBIServer44/&lt;instance&gt;/connectors/JText/out</p>
OutputExt	<p>Specifica l'estensione del file utilizzato per l'elaborazione dell'evento. Il valore predefinito è out.</p> <p><b>Nota:</b> Se OutputFileName non contiene nessuna estensione, ma l'attributo OutputExt contiene un'estensione, il file di output viene generato con il nome e l'estensione del file. Se non contiene un'estensione, il file di output viene generato senza uno.</p>
OutputFileName	<p>Specifica il nome e il percorso del file di output nel quale il connettore scrive l'oggetto business in entrata durante l'elaborazione della richiesta. Se l'attributo OutputDir contiene una directory di output valida, il file di output viene generato nella directory specificata. Per ulteriori informazioni, consultare "Specifica del nome del file di output" a pagina 56.</p> <p><b>Nota:</b> Se gli attributi OutputFileName eOutputExt non contengono un'estensione, il file di output viene generato senza un'estensione. Il valore predefinito fornito è Native.</p> <p>BiDi supportato. Se il formato BiDi utilizzato nelle piattaforme di destinazione è diverso dal formato Windows 2003, i valori per questi attributi di metadati verranno trasformati.</p>

Tabella 8. Attributi nella definizione del metaoggetto *MO\_JTextConnector\_Default* (Continua)

Nome attributo	Descrizione
PartialArchiveExt	Specifica l'estensione del file utilizzata per archiviare gli oggetti business elaborati con successo (quando il file di evento contiene più oggetti business, non tutti elaborati con successo). Per ulteriori informazioni, consultare "Specifica dell'archivio di evento" a pagina 59. Il valore predefinito fornito è <code>partial</code> .
StagingDir	<p>Specifica una directory nella quale il connettore dovrebbe scrivere i file prima di spostarli nella directory specificata dall'attributo <code>OutputDir</code>. Questo è progettato per gestire gli ambienti dove altri processi software potrebbero star elaborando e manipolando la directory nella quale il connettore JText emette i file (come un processo FTP che rileva i file creati dal connettore e li sposta in un'altra posizione). In situazioni come queste, c'è un rischio che il processo esterno sposti il file prima di essere stato scritto completamente. E' possibile specificare una directory di blocco nell'attributo <code>StagingDir</code>, in questo modo il connettore scrive il file completamente nella directory di blocco e quindi lo sposta nella directory di emissione quando è terminata, eliminando il rischio che il processo esterno prenda un file incompleto.</p> <p>Si consiglia la directory di blocco e di emissione sia sullo stesso file system o unità per adattare diversi approcci dei sistemi operativi alle operazioni di spostamento del file. Notare che <code>StagingDir</code> può essere una directory remota.</p> <p>Non esiste alcun valore predefinito fornito.</p> <p>BiDi supportato per Windows. Tutte le proprietà abilitate BiDi verranno trasformate dal formato Windows BiDi nel formato BiDi.Metadata, che è una proprietà standard del connettore e-Mail.</p>
SuccessArchiveExt	Specifica l'estensione del file utilizzata per archiviare gli oggetti business che non sono stati elaborati correttamente. Per ulteriori informazioni, consultare "Specifica dell'archivio di evento" a pagina 59. Il valore predefinito fornito è <code>success</code> .
UnsubscribedArchiveExt	Specifica l'estensione del file utilizzata per archiviare gli oggetti business non sottoscritti. Per ulteriori informazioni, consultare "Specifica dell'archivio di evento" a pagina 59. Il valore predefinito fornito è <code>unsub</code> .

**Nota:** Gli attributi `FTPTransferType`, `FTPDataStructure`, `DataProcessingMode`, `EndOfFileDelimiter`, e `FixedBOSize` non sono parte del metaoggetto JText come fornito. Per utilizzare questi attributi, è necessario aggiungerli esplicitamente al metaoggetto ed è necessario impostare i valori predefiniti.

### Specifica del nome del file di output

Ci sono tre modi per specificare il nome del file di output:

- Utilizzare l'attributo `OutputFileName`  
Utilizzare questo attributo quando si desidera che il connettore scriva ogni oggetto business dello stesso tipo in file separati con numeri di sequenza unici, o colleghi oggetti business in un file singolo con un nome specifico.
- Utilizzare un metaoggetto secondario  
Utilizzare un metaoggetto secondario dinamico quando si desidera generare dinamicamente un nome file di output per ogni tipo di oggetto business o restituire il nome del file di output generato dal connettore. Consultare "Utilizzo di un metaoggetto secondario dinamico" a pagina 4 per i dettagli.

Ci sono diversi modi di utilizzare l'attributo `OutputFileName` per specificare il nome del file di output:

- Se `OutputFileName` è impostato sulla stringa `Native` e l'attributo `FileSeqEnabled` viene impostato su `true`, il connettore invia la stringa dell'oggetto business ad un file univoco il cui nome è derivato dal nome dell'oggetto business in entrata, la cui estensione è derivata dall'attributo `OutputExt` e il cui percorso è derivata dall'attributo `OutputDir`. In questo caso, il funzionamento predefinito del connettore è di scrivere ogni oggetto business dello stesso tipo per separare i file con i numeri di sequenza unica. Per fare in modo che il connettore sovrascriva il file di output ogni volta che riceve gli oggetti business dello stesso tipo, impostare l'attributo `FileSeqEnabled` su `false`.
- Se `OutputFileName` viene impostato su una stringa diversa da `Native` e l'attributo `FileSeqEnabled` viene impostato su `true`, il connettore gestisce il valore del file di output in uno dei modi seguenti:
  - Se `OutputFileName` contiene un percorso assoluto (incluso il nome file e l'estensione del file di output, ad esempio, `OutputFileName= C:\temp\Out\test.out`), il connettore utilizza solo questo attributo per generare il file di output. In questo caso, il funzionamento predefinito del connettore è di scrivere ogni oggetto business dello stesso tipo per separare i file con il nome specificato e i numeri di sequenza unica.
  - Se `OutputFileName` contiene il percorso completo ed il nome file, ma nessuna estensione dell'attributo `OutputExt` contiene un valore, (ad esempio `OutputFileName= C:\temp\Out\test` e `OutputExt=out`), il connettore utilizza il valore di entrambi gli attributi per generare il file di output. In questo caso, il connettore genera un file denominato `C:\temp\Out\test_1.out`.
  - Se `OutputFileName` contiene il percorso completo ed il nome file, ma nessuna estensione, e l'attributo `OutputExt` non contiene un valore, il connettore genera il file di output senza nessuna estensione. In questo caso, il connettore genera un file denominato `C:\temp\Out\test_1`.
  - Se `OutputFileName` contiene solo il nome file e non l'estensione del percorso, e l'attributo `OutputDir` contiene un valore, il connettore genera il file di output nella directory specificata da `OutputDir`. Se `OutputExt` contiene un valore, il connettore utilizza anche quel valore. In caso contrario, esso crea il nome del file senza estensione.

**Nota:** Se il connettore sta elaborando più di un tipo di oggetto business e `OutputFileName` è impostato su una stringa diversa da `Native`, ogni oggetto business deve avere il suo metaoggetto di livello superiore che specifica un nome di file di output univoco. Ad esempio, il metaoggetto utilizzato dall'oggetto business `Custom` potrebbe essere `MO_JTextConnector_Customer`, e il metaoggetto utilizzato da `Item` potrebbe essere `MO_JTextConnector_Item`. Impostare il valore dell'attributo `OutputFileName` in ogni metaoggetto su un valore univoco.

- Per fare in modo che il connettore colleghi più oggetti business ad un file con il nome specificato, specificare un valore per `OutputFileName` ed impostare l'attributo `FileSeqEnabled` su `false`.
- Per fare in modo che il connettore sovrascriva il file di output ogni volta che riceve gli oggetti business dello stesso tipo, utilizzare il metaoggetto dinamico secondario. Specificare il percorso assoluto ed il nome file nell'attributo `InFileName` ed impostare l'attributo `FileWriteMode` su `"o"`. Per ulteriori informazioni sull'utilizzo del metaoggetto secondario dinamico, consultare "Utilizzo di un metaoggetto secondario dinamico" a pagina 4.

`Native` è una parola riservata.

Per ulteriori informazioni, consultare “Specifica l’elaborazione della richiesta” a pagina 61.

---

## Attività di configurazione comune

Questa sezione descrive le attività di configurazione comune.

- “Specifica notifica evento” a pagina 58
- “Specifica dell’archivio di evento” a pagina 59
- “Specifica l’elaborazione della richiesta” a pagina 61
- “Specificare più file di evento o più directory di evento” a pagina 65
- “Poll per gli oggetti business specifici” a pagina 65
- “Specifica di un file system FTP remoto” a pagina 68
- “Configurazione FTP sicuro” a pagina 75
- “Specificare un gestore dati” a pagina 76
- “Creazione di un metaoggetto JText per un oggetto business specifico” a pagina 76
- “Lettura di più oggetti business di tipi diversi dallo stesso file” a pagina 77
- “Specifica dei valori per gli attributi ObjectEventID” a pagina 77
- “Impostazione di una seconda istanza di un connettore JText” a pagina 77
- “Regolazione delle prestazioni del connettore JText” a pagina 78
- “Generazione file di esempio per la verifica” a pagina 79
- “Generazione degli oggetti business di esempio per la verifica” a pagina 79

## Specifica notifica evento

Questa sezione descrive quanto segue:

- “Specifica directory evento ed estensioni”
- “Configurazione funzionamento polling” a pagina 59

### Specifica directory evento ed estensioni

Se si invia più di un tipo di oggetto business al connettore per l’elaborazione, ed ogni tipo di oggetto business ha il proprio metaoggetto di livello superiore, la combinazione di valori specificati per gli attributi EventDir e EventExt deve essere unica per ogni coppia directory/estensione per ogni oggetto business.

In altre parole, se si specifica la stessa directory di evento per due tipi di oggetto business, è necessario specificare diverse estensioni di evento per questi oggetti business. Se si specifica la stessa estensione per due tipi di di oggetto business, è necessario specificare diverse directory di evento per questi oggetti business.

Ad esempio, presupporre di aver creato i metaoggettiM0\_JTextConnector\_Customer e M0\_JTextConnector\_Item per fornire valori di configurazione per gli oggetti business Customer e Item, rispettivamente. Se si istruisce il connettore ad ubicare i file di input per entrambi gli oggetti business nella stessa directory (specificando lo stesso percorso nell’attributo EventDir), è necessario identificare unicamente i file di input specificando valori diversi per l’attributo EventExt.

Quindi, se l’attributo EventDir valuta C:\temp\event per Customers e Items, il valore degli attributi EventExt per questi due oggetti business devono essere diversi (come in per i file di input Customer e inp per Items).

**Nota:** Una nuova proprietà booleana specifica del connettore, "NoPoll", è stata introdotta per disattivare facoltativamente il poll. Il valore predefinito è false. Quando è impostato su true, l'adattatore elabora solo le richieste e non esegue il poll.

## Configurazione funzionamento polling

Per configurare questa funzione di polling, procedere nel modo seguente:

1. Configurare i seguenti attributi del metaoggetto `MO_JTextConnector_Default`:
  - `EventDir`—Specificare il percorso assoluto di una directory esistente i cui file attivano la notifica dell'evento.
  - `EventExt`—Il connettore cerca i file con l'estensione predefinita fornita di in. Se si utilizza questo attributo per specificare una diversa estensione, il connettore cerca l'estensione specificata. Se si lascia questo attributo vuoto, il connettore esegue il poll per i file con nessuna estensione.
  - `EventDataHandler`—Specifica il gestore dati da utilizzare per la conversione dei dati durante la notifica dell'evento.
2. Utilizzare Connector Configurator Express per configurare le seguenti proprietà del connettore:
  - `PollFrequency`—Specificare la frequenza di intervallo.
  - `PollQuantity`—Specificare il numero di eventi per ogni intervallo poll.
  - `PollEndTime`—Specificare l'ora per completare il poll degli eventi.
  - `PollStartTime`—Specificare l'ora per iniziare il poll degli eventi.
3. Stabilire le autorizzazioni di lettura sulla directory dell'evento.

## Specifica dell'archivio di evento

In base a se tutti o alcuni oggetti business nel file di evento vengono elaborati correttamente, il connettore JText utilizza diverse estensioni quando crea il file di archivio per oggetti business elaborati correttamente. Il connettore inoltre scrive gli oggetti business la cui elaborazione non riesce e quelli di cui è annullata la sottoscrizione in file di archivio denominati in modo diverso.

Questa sezione descrive quanto segue:

- "nomi file archivio locale"
- "Configurazione archivio locale" a pagina 60

### nomi file archivio locale

Se si conserva i valori predefiniti forniti per gli attributi dell'estensione di archivio, il connettore crea i file di archivio denominati come di seguito riportato:

- Il file di evento ha un oggetto business singolo

Dopo che il connettore JText elabora un file di evento che contiene un oggetto business singolo, esso crea uno dei file seguenti nella directory di archivio:

- `filename_timestamp.success`, per archiviare un oggetto business elaborato correttamente
- `filename_timestamp.fail`, per archiviare un oggetto business che non è stata elaborata correttamente
- `filename_timestamp.unsub`, per archiviare un oggetto business al quale non si sottoscrive

Se l'oggetto business non riesce nell'elaborazione o non è sottoscritto, il connettore crea anche il file `filename_timestamp.orig`, che conserva il file di evento come il connettore lo ha ricevuto originariamente.

- Il file di evento ha più oggetti business e tutti sono elaborati correttamente  
Dopo che il connettore JText ha elaborato correttamente un file di evento con più oggetti business, esso crea *filename\_timestamp.success* nella directory di archivio.
- Il file di evento ha più oggetti business e alcuni non sono sottoscritti o non riescono nell'elaborazione  
Dopo che il connettore JText elabora un file di evento che contiene più oggetti business, esso crea tutti i seguenti file nella directory di archivio:
  - *filename\_timestamp.partial*, per archiviare tutti gli oggetti business la cui elaborazione è riuscita correttamente
  - *filename\_timestamp.fail*, per archiviare tutti gli oggetti business la cui elaborazione non è riuscita correttamente
  - *filename\_timestamp.unsub*, per archiviare tutti gli oggetti business a cui il connettore non si sottoscrive
  - *filename\_timestamp.orig*, per conservare il file di evento come il connettore che lo ha originariamente ricevuto

Ad esempio, presupporre che il file *LegacyApp.in* contenga quattro oggetti business:

- *Contract*, che è correttamente elaborato
- *Customer*, che non riesce nella formattazione
- *Order*, che è correttamente elaborato
- *Item*, al quale il connettore non si sottoscrive

In tal caso, il connettore crea i seguenti file nella directory di archivio:

- *LegacyApp\_timestamp.partial*, che contiene *Contract* e *Order*
- *LegacyApp\_timestamp.fail*, che contiene *Customer*
- *LegacyApp\_timestamp.unsub*, che contiene *Item*
- *LegacyApp\_timestamp.orig*, che contiene *Contract*, *Customer*, *Order*, e *Item*

## Configurazione archivio locale

Per configurare il connettore per l'archivio, seguire queste procedure:

1. Configurare i seguenti attributi del metaoggetto *MO\_JTextConnector\_Default*:
  - *ArchiveDir*—Specifica il percorso assoluto di una directory server FTP o locale esistente nel quale il connettore deve posizionare gli eventi (con estensioni file che indicano lo status di elaborazione) dopo che sono stati elaborati.
  - *SuccessArchiveExt*—Specifica l'estensione per il file che contiene gli oggetti business elaborati correttamente (quando tutti gli oggetti business sono elaborati correttamente).
  - *PartialArchiveExt*—Specifica l'estensione per il file che contiene tutti gli oggetti business elaborati correttamente (quando alcuni degli oggetti business nel file di evento non sono elaborati in modo corretto).
  - *UnsubscribedArchiveExt*—Specifica l'estensione per il file che contiene gli oggetti business ai quali il connettore non si sottoscrive.
  - *OriginalArchiveExt*—Specifica l'estensione per il file che conserva tutti gli oggetti business contenuti nel file di evento.
  - *FailArchiveExt*—Specifica l'estensione per il file che contiene gli oggetti business che non sono riusciti nell'elaborazione.
2. Utilizzare *Connector Configurator Express* per configurare le proprietà del connettore *ArchivingEnabled*.
3. Stabilisce le autorizzazioni di scrittura sulla directory di archivio.

## Specifica l'elaborazione della richiesta

E' possibile fare in modo che il connettore JText scriva gli oggetti business nei file i cui nomi sono specificati in modo dinamico (in ogni istanza dell'oggetto business) o in modo statico (tramite i metaoggetti). E' inoltre possibile fare in modo che il connettore restituisca ogni nome file che genera in modo statico; questa funzione è utile per ottenere i nomi file generati con un numero di sequenza unico. Questa sezione contiene le seguenti sottosezioni:

- "Denominazione file dinamica" a pagina 61
- "Denominazione file statica" a pagina 61
- "Restituzione del nome file" a pagina 62
- "Differenze tra l'elaborazione locale e remota" a pagina 63
- "Configurazione del file di output" a pagina 63

### Denominazione file dinamica

Per fare in modo che il connettore generi in modo dinamico un nome file di output per ogni tipo di oggetto business, creare un metaoggetto secondario dinamico.

Utilizzare il metaoggetto:

- per specificare il nome del file di output o per ricevere il nome del nome file generato
- per specificare se allegare o sovrascrivere il file di output

**Importante:** Oltre a creare un metaoggetto secondario dinamico per abilitare il connettore a generare o restituire il nome file, se si sta utilizzando InterChange Server Express come broker di integrazione, è inoltre necessario modificare le associazioni o la logica di collaborazione per inserire nell'attributo `InFileName` del metaoggetto secondario dinamico un percorso ed un nome file per ogni oggetto business e, se richiesto, numeri di sequenza univoci.

Per ulteriori informazioni, consultare "Utilizzo di un metaoggetto secondario dinamico" a pagina 4.

Per informazioni su come il connettore elabora il metaoggetto, consultare "Elaborazione richiesta" a pagina 10.

Per informazioni su come configurare il connettore per utilizzare un nome file di output generato in modo dinamico, consultare "Configurazione del file di output" a pagina 63.

### Denominazione file statica

Quando si utilizzano metaoggetti per specificare il nome dei file di output, è necessario riavviare il connettore affinché le modifiche diventino effettive. E' possibile specificare se il connettore collega tutti gli oggetti business di un dato tipo ad un file singolo o crea un file separato per ogni oggetto business.

Quando utilizza la configurazione predefinita fornita, il connettore crea un file di output per ogni oggetto business che elabora. Esso denomina il file di output per l'oggetto business in entrata e aggiunge un numero di sequenza per rendere il nome unico; esso gli da l'estensione di `.out`. Ad esempio, se riceve gli oggetti business `Customer` e `Item`, il connettore scrive i loro dati nel file di output `Customer_1.out` e `Item_1.out`. Per informazioni su come ottenere i nomi dei file di output generati, consultare "Restituzione del nome file" a pagina 62. Per informazioni sul file che archivia i numeri di sequenza, consultare "OutputLog" a pagina 31.

Per utilizzare il metaoggetto per configurare il nome dei file di output, procedere nel modo di seguito riportato:

1. Configurare i seguenti attributi del metaoggetto `MO_JTextConnector_Default`:
  - `OutputDir`—Specifica il percorso assoluto di una directory esistente nella quale il connettore scrive i file quando elabora le richieste. Per ulteriori informazioni, consultare “Configurazione del file di output” a pagina 63.
  - `OutputExt`—Utilizza questo attributo per specificare la propria estensione se si desidera modificare la configurazione predefinita fornita, che fa in modo che il connettore crei i file con estensione `out`.
  - `FileSeqEnabled`—mantenere impostato su `true` per fare in modo che il connettore emetta un oggetto business per file, ognuno con un numero di sequenza univoco. Impostare su `false` per fare in modo che il connettore per emettere tutti gli oggetti business di un tipo dato in un file singolo, sovrascrivendo o allegando. Per informazioni sulla configurazione del funzionamento di sovrascrittura o collegamento, consultare Tabella 9 a pagina 63.
  - `OutputFileName`—Per fare in modo che il connettore colleghi gli oggetti business in un singolo file di output, piuttosto che sovrascrivere i dati nel file o generare file univoci per ogni oggetto business, specificare il percorso completo e il nome file del file di output.  
Per fare in modo che il connettore sovrascrive il file di output ogni volta che riceve lo stesso tipo di oggetto business, non specificare un valore per `OutputFileName`.  
In ognuno di questi casi, impostare `FileSeqEnabled` su `false`.  
Per informazioni sulla configurazione del funzionamento di sovrascrittura o collegamento, consultare Tabella 9 a pagina 63.
2. Stabilire autorizzazioni di scrittura sulla directory di output.

**Nota:** E' necessario creare metaoggetti per oggetti business specifici se il connettore deve utilizzare formati di dati diversi o convenzioni di denominazione file per diversi oggetti business.

### Restituzione del nome file

Per fare in modo che il connettore restituisca i nomi dei file che genera, procedere nel modo seguente:

- Utilizzare i metaoggetti per specificare il percorso ed i nomi file e per fare in modo che il connettore generi un numero di sequenza unico per ogni file di output. Per ulteriori informazioni, consultare “Denominazione file statica” a pagina 61.
- Utilizzare un metaoggetto secondario dinamico per fare in modo che il connettore restituisca il nome di ogni file che genera. Seguire questa procedura in “Utilizzo di un metaoggetto secondario dinamico” a pagina 4, ma non specificare un valore per il suo attributo `InFileName`. Quando il connettore riceve un oggetto business il cui metaoggetto secondario dinamico specifica `OutFileName=CxIgnore`, esso crea un nome file basato sulla configurazione del metaoggetto di livello superiore e restituisce il percorso completo ed il nome file come un valore nell'attributo `InFileName`.

**Nota:** Il connettore popola l'attributo `InFileName` solo con un percorso locale, anche quando elabora i file su un server FTP.

**Importante:** Oltre a creare un metaoggetto secondario dinamico per abilitare il connettore a generare o restituire il nome file, se si sta utilizzando

InterChange Server Express come broker di integrazione, è inoltre necessario modificare le associazioni o la logica di collaborazione per inserire nell'attributo InFileName del metaoggetto secondario dinamico un percorso ed un nome file per ogni oggetto business e, se richiesto, numeri di sequenza univoci.

## Differenze tra l'elaborazione locale e remota

Il connettore elabora i file in modo remoto nello stesso modo in cui li elabora localmente. Ci sono, tuttavia, poche differenze:

- Quando si elaborano gli eventi e si generano nomi file in modo dinamico, il connettore popola l'attributo InFileName del metaoggetto secondario dinamico con un nome percorso locale e non con un percorso remoto.
- Quando si elaborano le richieste, se il connettore non è configurato per la denominazione file dinamica e FileSeqEnabled valuta false ed il file di output già esiste:
  - Se elaborato in modo locale, il connettore sovrascrive il file esistente.
  - Se elaborato in modo remoto, il connettore invia un'eccezione.
- Oltre alla configurazione degli attributi di estensione archivio standard per l'elaborazione evento locale, quando si utilizza il connettore per elaborare i file in modo remoto su un server FTP, è inoltre possibile configurare gli attributi FTPArchiveDir e FTPRenameExt. Questi attributi consentono di abilitare alla ridenominazione e allo spostamento del file archiviato indipendentemente dalla riuscita dell'elaborazione.

Per ulteriori informazioni consultare "Specifica dell'archivio remoto" a pagina 69

## Configurazione del file di output

Tabella 9 illustra le possibili opzioni di configurazione per il file di output:

Tabella 9. Specifica dei file di output

condizione output desiderata	Attributi/proprietà che richiedono configurazione	Valore attributo/proprietà
Ogni oggetto business di un dato tipo è collegato ad un file il cui percorso assoluto ed il nome file viene derivato al runtime da un attributo nell'oggetto business.	Utilizzare un metaoggetto secondario AppSpecificInfo (a livello oggetto business) Per un metaoggetto secondario dinamico: OutFileName FileWriteMode	cw_mo_JTextConfig = DynChildMOName nome percorso e nome file specificato dall'utente a o append
Ogni oggetto business di un dato tipo sovrascrive il file di output il cui percorso assoluto ed il nome file viene derivato al runtime da un attributo nell'oggetto business.	Utilizzare un metaoggetto secondario AppSpecificInfo (a livello oggetto business) Per un metaoggetto secondario dinamico: OutFileName FileWriteMode	cw_mo_JTextConfig = DynChildMOName nome percorso e nome file specificato dall'utente o o overwrite
Ogni oggetto business di un dato tipo è scritto nel suo file univoco il cui nome è derivato dal nome dell'oggetto business ed un numero di sequenza univoco generato.	OutputDir  FileSeqEnabled OutputFileName	nome percorso specificato dall'utente  Vero Native

Tabella 9. Specifica dei file di output (Continua)

condizione output desiderata	Attributi/proprietà che richiedono configurazione	Valore attributo/proprietà
Il connettore restituisce il nome di ogni file che genera. Ogni oggetto business di un dato tipo è scritto nel suo file univoco il cui nome è derivato dal nome dell'oggetto business ed un numero di sequenza univoco generato.	Utilizzare un metaoggetto secondario dinamico	
	AppSpecificInfo (a livello oggetto business)	cw_mo_JTextConfig = DynChildMOName
	InFileName (nel metaoggetto secondario dinamico)	CxIgnore
	FileWriteMode (nel metaoggetto secondario dinamico)	N/A
	Utilizzare configurazione del metaoggetto: MO_JTextConnector_businessobjectname:	
Tutti gli oggetti business di un tipo dato sono collegati ad un file singolo il cui nome è specificato dall'utente.	OutputDir	nome percorso specificato dall'utente
	FileSeqEnabled	Vero
	OutputFileName	Native
	FileSeqEnabled	Falso
Ogni oggetto business di un tipo dati è scritto in un file univoco il cui nome è specificato dall'utente più un numero di sequenza univoca.	OutputFileName	nome percorso e nome file specificato dall'utente
	FileSeqEnabled	Vero
Se il connettore sta elaborando più di un tipo di oggetto business e OutputFileName è impostato su una stringa diversa da Native, ogni oggetto business deve avere il suo metaoggetto di livello superiore. Per ulteriori informazioni, consultare "Specifiche del nome del file di output" a pagina 56.	OutputFileName	nome percorso e nome file specificato dall'utente
	OutputDir	nome percorso specificato dall'utente
Ogni oggetto business di un dato tipo sovrascrive il file di output, il cui nome è derivato dal nome dell'oggetto business.	FileSeqEnabled	Falso
	OutputFileName	Native
	Utilizzare un metaoggetto secondario dinamico	
Il connettore restituisce il nome di ogni file che genera. Ogni oggetto business di un tipo dati è scritto in un file univoco il cui nome è specificato dall'utente più un numero di sequenza univoca.	AppSpecificInfo (a livello oggetto business)	cw_mo_JTextConfig = DynChildMOName
	InFileName (nel metaoggetto secondario dinamico)	CxIgnore
	FileWriteMode (nel metaoggetto secondario dinamico)	N/A
	Utilizzare la configurazione del metaoggetto: MO_JTextConnector_BOName.	
	FileSeqEnabled	Vero
	OutputFileName	nome percorso e nome file specificato dall'utente

## Specificare più file di evento o più directory di evento

È possibile configurare il connettore in modo da prendere solo file con un'estensione specificata. È inoltre possibile configurare il connettore in modo da prendere i file da più directory.

**Importante:** L'utilizzo di un asterisco (\*) per l'attributo `EventExt` per specificare che il poll del connettore per tutti i file in una directory di evento singola indipendentemente dalla estensione, non è più supportato.

Per specificare una directory di evento separata per ogni tipo di oggetto business, eseguire la procedura di seguito riportata:

1. Creare un metaoggetto separato per ogni oggetto business supportato; ad esempio, creare `MO_JTextConnector_Customer` e `MO_JTextConnector_Item`. Per ulteriori informazioni, consultare "Creazione di un metaoggetto JText per un oggetto business specifico" a pagina 76.
2. Specificare la directory appropriata in ogni attributo `EventDir` del metaoggetto.

**Nota:** Il connettore JText elabora i file di evento nell'ordine della data/ora, dal più vecchio al più recente, indipendentemente dalla posizione. In altre parole, il connettore JText elabora i file posizionati in directory separate nell'ordine cronologico nell'ordine delle date/ore.

## Poll per gli oggetti business specifici

La configurazione del connettore JText è diversa a seconda se tutti i file di evento si trovano in una directory singola, hanno tutti la stessa estensione, contengono un oggetto business singolo o più oggetti business, contengono oggetti business di uno o più tipi e rappresentano ogni oggetto business su una o più righe.

Questa sezione spiega quanto segue:

- "Utilizzo del metodo di analisi `EndBODElimiter`"
  - "Utilizzo dei caratteri non stampabili per `EndBODElimiter`" a pagina 66
- "Utilizzo del metodo di analisi `FixedBOSize`" a pagina 68

### Utilizzo del metodo di analisi `EndBODElimiter`

Se non viene specificato alcun valore per l'attributo `metaoggettoEndBODElimiter`, il connettore:

- prevede che il file di evento delimiti le stringhe dell'oggetto business con `<EndBO:BOName>`
- specifica `<EndBO:BOName>` come delimitatore quando scrive le stringhe di oggetti business nei file di output.

Se un file di evento contiene solo un oggetto business, è possibile specificare EOF (end of file) per questo attributo.

Se si imposta il valore dell'attributo `EndBODElimiter` su una stringa non vuota, si presuppone che questa dia il delimitatore dell'oggetto business per tutti i file. Se il valore non è impostato o è annullato, il connettore presuppone che il delimitatore sia `<EndBO:BOName>`.

**Importante:** Se `DataProcessingMode` è impostato su `binary` e se non c'è nessun valore specificato per `EndBODElimiter`, JText imposta il `EndBODElimiter` predefinito su `FF01` (2 byte) e `EndOfFileDelimiter` su `FF02` (2 byte).

Tabella 10 illustra le opzioni del delimitatore.

Tabella 10. Utilizzo dell' attributo EndBODelimiter

Condizioni	Delimitatore	Note
Il file contiene una o più stringhe dell'oggetto business con uno o più tipi di oggetto business o File contiene più stringhe di oggetti business dello stesso tipo dell'oggetto business; ogni stringa è in esecuzione su diverse righe.	<EndB0:BOName>o EOL o specificato dall'utente valore	<ul style="list-style-type: none"> <li>• Specificare tanti EOL separati da un punto e virgola quante sono le nuove righe tra le stringhe di oggetti business.</li> <li>• Specificare un delimitatore personalizzato insieme agli EOL. Un delimitatore personalizzato deve essere il primo elemento quando viene utilizzato con EOL. Il seguente esempio è valido: customEndB0;EOL;EOL. Il seguente esempio <b>non</b> è valido: EOL;customEndB0;EOL.</li> </ul>
Ogni file contiene solo una stringa di oggetto business	EOL Fo valore specificato dall'utente	<ul style="list-style-type: none"> <li>• Specificare tanti EOL separati da un punto e virgola quante sono le nuove righe tra le stringhe di oggetti business.</li> <li>• Specificare un delimitatore specificato dall'utente insieme agli EOL e EOF se richiesto dalle stringhe di input. Un delimitatore personalizzato deve sempre essere il primo elemento se utilizzato con EOL. Il seguente esempio è valido: customEndB0;EOL;EOL. Il seguente esempio <b>non</b> è valido: EOL;customEndB0;EOL.</li> </ul>
Il file contiene più stringhe di un oggetto business, una per riga	EOL	
Il File contiene più stringhe di oggetti business dello stesso tipo dell'oggetto business; ogni stringa è in esecuzione su diverse righe senza separazione tra le stringhe degli oggetti business.	Nessuno	<p>Può utilizzare il metaoggetto predefinito fornito o un metaoggetto personalizzato</p> <p><b>Nota:</b> Questa opzione è disponibile solo durante le richieste di chiamata al servizio e non per la notifica degli eventi. Non utilizzare il delimitatore insieme ad altri.</p>

**Nota:** Se il file di origine contiene righe vuote, il connettore le ignora.

**Utilizzo dei caratteri non stampabili per EndBODelimiter:** Per eseguire il poll dei file in più directory, è necessario creare un metaoggetto per ogni oggetto business supportato. Il valore specificato per ogni attributo EndBODelimiter del metaoggetto dipende dal fatto che il file di origine contiene o meno un oggetto business singolo o più oggetti business.

- File che contengono un oggetto business singolo

E' possibile specificare EOF come EndBODelimiter se l'intero file di dati contiene solo una stringa dell'oggetto business.

- File che contengono più oggetti business

Se il file di input contiene più oggetti business che hanno solo una nuova riga come delimitatore dell'oggetto business, specificare l'EOL di stringa nell'attributo

EndBODelimiter. In questo caso, il file di origine contiene stringhe che rappresentano più oggetti business dello stesso tipo.

**Importante:** Per eseguire il poll da un file che contiene più tipi di oggetti business, è necessario utilizzare il metaoggetto MO\_JTextConnector\_Default e verificare che gli attributi EventExt e EventDir sono puntati correttamente alla directory dove si trova questo file di evento. Per eseguire il poll dei tipi di oggetto business che sono rappresentati in file di evento separati o i cui file di evento si trovano in diverse directory, è necessario creare un metaoggetto di livello superiore separato. Utilizzare gli attributi EventExt e EventDir per puntare alla directory appropriata.

Per utilizzare un gestore dati personalizzato quando si esegue il poll dei file che contengono più oggetti business di tipi diversi, consultare “Lettura di più oggetti business di tipi diversi dallo stesso file” a pagina 77.

Se si utilizza il formato nome/valore, non è possibile specificare il delimitatore dell’oggetto business EOL se il file di evento separa i dati dell’oggetto business su più righe. Per ulteriori informazioni, consultare *Data Handler Guide*.

I seguenti esempi mostrano il delimitatore da utilizzare per diversi formati di file di evento:

- Il file contiene quattro stringhe di oggetti business ed utilizza il carattere non stampabile EOL come fine del delimitatore dell’oggetto business:  
Sample\_BO~Create~1~TableGenKey5~strange~TextConnector\_924055528\_0  
Sample\_BO~Create~2~TableGenKey5~strange~TextConnector\_924055528\_0  
Sample\_BO~Create~3~TableGenKey5~strange~TextConnector\_924055528\_0  
Sample\_BO~Create~4~TableGenKey5~strange~TextConnector\_924055528\_0
- Il file contiene quattro stringhe di oggetti business ed utilizza il valore specificato dall’utente ed il carattere non stampabile EOL come fine del delimitatore dell’oggetto business, cioè CustomEndBO;EOL:  
Sample\_BO~Create~1~TableGenKey5~strange~TextConnector\_924055528\_0CustomEndBO  
Sample\_BO~Create~2~TableGenKey5~strange~TextConnector\_924055528\_0CustomEndBO  
Sample\_BO~Create~3~TableGenKey5~strange~TextConnector\_924055528\_0CustomEndBO  
Sample\_BO~Create~4~TableGenKey5~strange~TextConnector\_924055528\_0CustomEndBO
- Il file che contiene quattro stringhe di oggetti business ed utilizza il carattere non stampabile EOL;EOL come fine del delimitatore dell’oggetto business:  
Sample\_BO~Create~1~TableGenKey5~strange~TextConnector\_924055528\_0  
  
Sample\_BO~Create~2~TableGenKey5~strange~TextConnector\_924055528\_0  
  
Sample\_BO~Create~3~TableGenKey5~strange~TextConnector\_924055528\_0  
  
Sample\_BO~Create~4~TableGenKey5~strange~TextConnector\_924055528\_0
- Il file che contiene quattro stringhe di oggetti business ed utilizza None come fine del delimitatore dell’oggetto business:  
Sample\_BO~Create~1~TableGenKey5~strange~TextConnector\_924055528\_0Sample\_BO  
~Create~2~TableGenKey5~strange~TextConnector\_924055528\_0Sample\_BO~Create~3  
~TableGenKey5~strange~TextConnector\_924055528\_0Sample\_BO~Create~4  
~TableGenKey5~strange~TextConnector\_924055528\_0

**Nota:** Il connettore è sensibile al maiuscolo/minuscolo nella stringa specificata, fatta eccezione per i delimitatori EOL e EOF.

Per ulteriori informazioni sulla creazione dei metaoggetti, consultare “Creazione di un metaoggetto JText per un oggetto business specifico” a pagina 76.

## Utilizzo del metodo di analisi FixedBOSize

Questa proprietà del metaoggetto è valida solo nelle seguenti istanze:

1. Quando si esegue l'elaborazione dell'evento.
2. Quando DataProcessingMode è impostato su Binary.

Quando è presente con un valore valido, questa proprietà del metaoggetto sovrascrive la proprietà EndBODElimiter e fornisce agli utenti un'alternativa al delimitatore tradizionale basta sull'analisi dell'oggetto business. Questa proprietà consente al connettore di correlare un numero fisso di byte con un singolo oggetto business. Ad esempio, se un file era composto di 300 byte e la proprietà FixedBOSize era impostata su 100, JText Adapter converte questi tre pacchetti di lunghezza a 100 byte tramite un gestore dati abilitato binario e li invia a InterChange Server Express.

Se FixedBOSize e EndBODElimiter hanno un valore impostato, allora Jtext utilizza FixedBOSize per l'analisi del file ed ignora EndBODElimiter.

## Specifica di un file system FTP remoto

Questa sezione descrive come configurare l'adattatore JText per utilizzare un file system FTP remoto per l'elaborazione di eventi e richieste.

**Importante:** Per consentire al connettore di abilitare un file system FTP remoto, è necessario specificare un URL FTP nell'attributo EventDir (per l'elaborazione degli eventi) o l'attributo OutputDir (per l'elaborazione delle richieste). E' inoltre necessario risolvere tutti i problemi del firewall prima di utilizzare il connettore per eseguire le operazioni FTP.

Questa sezione descrive quanto segue:

- "Elaborazione eventi remota"
- "Elaborazione richiesta remota" a pagina 72
- "Note sulla configurazione del connettore per il trasferimento FTP" a pagina 74

### Elaborazione eventi remota

Per configurare il connettore affinché utilizzi il file system FTP per l'elaborazione degli eventi, è necessario specificare un URL FTP, le informazioni di login FTP, una directory locale nella quale il connettore scarica i file di evento dalla directory remota, archivia le informazioni e le informazioni relative a come funziona il connettore quando il server FTP non è disponibile. Questa sezione descrive tutte queste configurazioni e le altre configurazioni facoltative.

- "Specifica dell'URL FTP ed informazioni di login"
- "Specifica della directory locale" a pagina 69
- "Specifica dell'archivio remoto" a pagina 69
- "Specifica del poll remoto" a pagina 70
- "Il connettore elabora gli eventi da un sito remoto" a pagina 71
- "Identificazione file su una struttura principale: Configurazione facoltativa" a pagina 71
- "Riepilogo delle operazioni di configurazione per l'elaborazione dell'evento" a pagina 72

**Specifica dell'URL FTP ed informazioni di login:** Il connettore esegue il poll degli eventi dalla directory specificata nell'attributo del metaoggetto EventDir. Per configurare il connettore affinché utilizzi un file system FTP remoto per

l'elaborazione degli eventi, specificare l'URL dell'FTP come valore per questo attributo. L'URL FTP deve essere conforme agli standard IETF.

Oltre alla specifica del server FTP nell'URL, è possibile specificare facoltativamente le seguenti informazioni nell'attributo del metaoggetto EventDir:

- Denominare un utente con privilegi da connettere al server FTP ed eseguire le operazioni FTP—Se non si specifica il nome utente in EventDir, specificarlo nell'attributo del metaoggettoFTPUserId.
- La password di un utente con privilegi da connettere al server FTP ed eseguire le operazioni FTP—Se non si specifica la password in EventDir, specificarla nell'attributo del metaoggetto FTPPassword.
- Numero porta—Se la porta non è specificata in EventDir, il connettore utilizza la porta predefinita.
- Directory evento remota—Se non si specifica la directory di evento remota in EventDir, il connettore esegue il poll dei file di evento dalla directory nella quale viene stabilita la connessione al server FTP.

**Importante:** E' possibile specificare i valori FTP in un metaoggetto di livello superiore statico o in un metaoggetto secondario dinamico. Se il nome utente e la password non sono specificate nell'attributo del metaoggetto, il connettore si arresta quando si prova a connetterlo al server FTP. Per ulteriori informazioni, consultare "Utilizzo di un metaoggetto secondario dinamico" a pagina 4.

Gli esempi di seguito riportati illustrano tre formati diversi per i valori dell'attributo EventDir:

**URL solo con valori richiesti:**

`ftp://ftp.companyA.com`

**URL con valori facoltativi di nome utente e numero porta:**

`ftp://companyA:admin@ftp.companyA.com:1433`

**URL con valori predefiniti di nome utente, numero porta e directory evento remota:**

`ftp://companyA:admin@ftp.companyA.com:1433/temp/JTextConn/Default/Event`

**URL per l'impostazione FTP relativa a Linux / MVS**

`ftp://ftpuser:ftppwd@ftpsrvr.in.ibm.com:21/home/ftpuser/JText/event`

**Specifica della directory locale:** Oltre a specificare l'URL del FTP e le informazioni relative al login, è necessario specificare la posizione della directory locale nella quale il connettore scarica i file di evento dalla directory remota. Per specificare la directory locale, utilizzare l'attributo del metaoggetto FTPLocalEventDir.

**Importante:** Se il connettore trova un URL FTP valido in EventDir, ma non trova l'attributo del metaoggetto FTPLocalEventDir o trova un valore non valido o vuoto per questo attributo, il connettore non si avvia. Il connettore non valuta l'attributo FTPLocalEventDir quando è stato configurato per essere eseguito in modo locale.

**Specifica dell'archivio remoto:** Ci sono diverse opzioni per specificare come il connettore gestisce l'archivio remoto. Per specificare una directory di archivio

remota, utilizzare l'attributo del metaoggetto FTPArchiveDir. Questo attributo specifica il percorso relativo della directory di archivio sul server FTP. La directory deve già esistere. Ci sono diverse opzioni per utilizzare questo attributo in modo da specificare l'archivio:

- Specificare un valore per l'attributo FTPArchiveDir ma nessun valore per l'attributo FTPRenameExt fa in modo che il connettore colleghi una data/ora al nome file dell'evento e lo sposti alla directory di archivio del server FTP specificata nell'attributo FTPArchiveDir.
- Specificare un valore per l'attributo FTPArchiveDir e l'attributo FTPRenameExt in modo che il connettore ridenomini il nome file dell'evento elaborato, aggiunga una data/ora e ignori FTPRenameExt e quindi lo sposti nella directory di archivio del server FTP specificata nell'attributo FTPArchiveDir.
- Non specificare alcun valore per gli attributi FTPArchiveDir o FTPRenameExt in modo che il connettore elimini il file di evento elaborato senza archivarlo.
- Non specificare alcun attributo per l'attributo FTPArchiveDir ma specificare un valore per l'attributo FTPRenameExt in modo che il connettore ridenomini il nome file dell'evento elaborato con il valore specificato in FTPRenameExt, e lo sposti nella directory specificata nell'attributo EventDir.

**Denominazione file con data/ora per i server FTP remoti:** Il supporto per i file system host (MVS) utilizzando Sequential datasets sono stati modificati fornendo data/ora per evitare di duplicare i nomi file. MVS non supporta caratteri speciali, come "\_", in un nome dataset o recordset. Sulle piattaforme Windows, Linux, o i5/OS, si utilizza una data/ora nel nome file originale durante l'archivio del file. Questo evita nomi file duplicati nella cartella di archivio, evitando così la sovrascrittura di un file esistente.

Si utilizza il seguente formato per i sistemi MVS per superare questa limitazione:

File di evento: Test.in

File archiviato: Test.TSyyyyMM.TSDDHHMM.TSSsSss

Dove:        yyyy -- anno  
             MM -- mese  
             DD -- data  
             HH -- ora  
             MM -- minuti  
             Ss -- secondi  
             Sss -- millisecondi

Sulle piattaforme MVS il separatore del dataset o di recordset è "." (punto) e numero massimo di "'" (punti) consentiti in un dataset o recordset è 6 (sei). Il nome del dataset o recordset non deve superare 8 caratteri per "." (punto) e il numero totale di caratteri non deve superare i 44 caratteri. Viene di seguito riportato un esempio di un nome file in questo formato:

FTPRenameExt -- ARCHIVE

File archiviato -- (SAMPLE).ARCHIVE.TS200304.TS290535.TS42234

**Nota:** I membri di PDS non possono essere ridenominato con data/ora durante l'archiviazione. Tuttavia, un meccanismo alternativo viene fornito per l'archivio di PDS. Ogni membro di un PDS che non riesce nel FTPEventFileMask viene archiviato nel PDS principale, con nome file specificato come FTPRenameExt. Il file di archivio verrà riscritto ogni volta con l'ultimo file elaborato.

**Specifica del poll remoto:** E' possibile utilizzare la proprietà di configurazione "FTPPollFrequency" a pagina 31 per impostare quanto frequentemente il connettore esegue il poll di un server FTP misurato nel numero di cicli poll

standard. Questa impostazione è utile se il connettore sta ancora leggendo i file della directory di evento locale quando avvia il ciclo successivo di polling.

Ad esempio, se "PollFrequency" a pagina 109 è impostato su 10000, e FTSPollFrequency è impostato su 6, il connettore esegue il poll della directory di evento locale ogni 10 secondi ed esegue il poll della directory remota ogni 60 secondi. Il connettore esegue il poll del FTP solo se si specifica un valore per questa proprietà. Se FTSPollFrequency valuta 0 o vuoto, il connettore non esegue il poll di FTP.

Per ulteriori informazioni, consultare "Regolazione delle prestazioni del connettore JText" a pagina 78.

**Il connettore elabora gli eventi da un sito remoto:** Quando si esegue il poll degli eventi da un sito remoto, il connettore esegue le fasi seguenti:

1. Ottenere il nome del server, il numero di porta, il nome utente, la password, e la directory evento remota dagli attributi del metaoggetto o valori predefiniti.
2. Stabilisce una connessione ad un sito FTP remoto per ottenere file di evento dalla directory di evento remota.
3. Scarica i file di evento dalla directory remota alla directory locale specificata nell'attributo del metaoggetto FTPLocalEventDir.

**Nota:** È necessario che questo attributo abbia un valore affinché abiliti il connettore ad elaborare gli eventi utilizzando FTP.

4. Esegue il poll della directory locale.

Figura 6 illustra l'elaborazione degli eventi locale e remota.

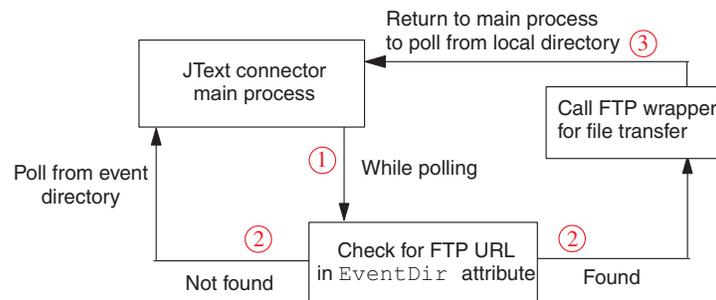


Figura 6. Operazioni di notifica evento locale e remoto

**Identificazione file su una struttura principale: Configurazione facoltativa:**

Utilizzare l'attributo FTPEventFileMask per identificare le estensioni file su una struttura principale che non aderisce agli stessi standard di denominazione che si applicano ai sistemi Windows, Linux, o i5/OS. Se non viene fornito alcun valore per questo attributo, il connettore utilizza il valore specificato nell'attributo EventExt.

Quando si specifica un valore per FTPEventFileMask, è necessario includere caratteri jolly. Il seguente esempio illustra diversi formati possibile per questo attributo:

```

ACT.Z1UC.*
ACT.*.INPT
*.Z1UC.INPT

```

Se il connettore trova più di un file nel sito remoto che rispetta i criteri specificati per `FTPEventFileMask`, questo procede nel modo seguente:

1. Scarica tutti i file di evento specificati nella directory specificata nell'attributo `FTPLocalEventDir`.
2. Ridenomina l'estensione dei file remoti con il valore specificato nell'attributo del metaoggetto in `FTPRenameExt`. La ridenominazione dei file evita che il connettore esegua il poll dello stesso file nel ciclo di poll successivo.
3. Si disconnette dal server FTP
4. Elabora i file in modo locale nella directory specificata nell'attributo del metaoggetto `FTPEventFileMask`.

**Riepilogo delle operazioni di configurazione per l'elaborazione dell'evento:** Per configurare il connettore affinché utilizzi un file system FTP remoto per l'elaborazione evento, specificare i seguenti valori di configurazione:

- Specificare l'URL FTP nell'attributo del metaoggetto `EventDir`. Alternativamente, specificare il nome e la password di un utente con privilegi per connettersi al server FTP ed eseguire le operazioni FTP.
- Se non si specifica il nome di accesso e la password nell'attributo del metaoggetto `EventDir`, farlo negli attributi del metaoggetto `FTPUserId` e `FTPPassword`.
- Se non si specifica la porta nell'attributo del metaoggetto `EventDir`, il connettore utilizza la porta FTP predefinita.
- Utilizzare l'attributo del metaoggetto `FTPLocalEventDir` per specificare la directory del sistema locale nella quale il connettore scarica i file di evento dal sito FTP.
- Su una struttura principale che non aderisce agli stessi standard di denominazione che si applicano ai sistemi Windows, Linux, o i5/OS, utilizzare l'attributo del metaoggetto `FTPEventFileMask` per identificare i file di cui eseguire il poll.
- Per configurare il connettore da gestire con il server FTP MVS quando il sistema remoto è MVS, specificare `MVS` nell'attributo `FTPSPFormat`.

### **Elaborazione richiesta remota**

Per configurare il connettore affinché utilizzi il file system FTP per l'elaborazione delle richieste, è necessario specificare un URL FTP, le informazioni di login FTP, una directory remota nella quale il connettore carica i file di richiesta dalla directory locale. Questa sezione descrive tutte queste configurazioni e le altre configurazioni facoltative.

- "Specificare l'URL FTP e le informazioni di accesso"
- "Come il connettore elabora le richieste di chiamata del servizio in un sito remoto" a pagina 73
- "Riepilogo delle operazioni di configurazione per l'elaborazione della richiesta" a pagina 74

**Specificare l'URL FTP e le informazioni di accesso:** Il connettore carica i file di richiesta della chiamata di servizio nella directory specificata nell'attributo del metaoggetto `OutputDir`. Per configurare il connettore affinché utilizzi un file system FTP remoto per l'elaborazione delle richieste, specificare l'URL dell'FTP come valore per questo attributo. L'URL FTP deve essere conforme agli standard IETF.

Oltre all'URL del FTP, è possibile specificare facoltativamente le seguenti informazioni nell'attributo del metaoggetto `OutputDir`:

- Denominare un utente con privilegi da connettere al server FTP ed eseguire le operazioni FTP—Se non si specifica il nome utente in `OutputDir`, specificarlo nell'attributo del metaoggetto `FTPUserId`.
- La password di un utente con privilegi da connettere al server FTP ed eseguire le operazioni FTP—Se non si specifica la password in `OutputDir`, specificarla nell'attributo del metaoggetto `FTPPassword`.
- Numero porta—Se la porta non è specificata in `EventDir`, il connettore utilizza la porta predefinita.
- Directory di output remota—Se non si specifica la directory di output remota in `OutputDir`, il connettore carica i file di richiesta nella directory di connessione predefinita (la directory sul server FTP nella quale è stabilita la connessione).

**Importante:** E' possibile specificare i valori FTP in un metaoggetto di livello superiore statico o in un metaoggetto secondario dinamico. Se il nome utente e la password non sono specificate nell'attributo del metaoggetto, il connettore si arresta con un'eccezione. Per ulteriori informazioni, consultare "Utilizzo di un metaoggetto secondario dinamico" a pagina 4.

Gli esempi illustrano tre formati diversi per i valori dell'attributo `OutputDir`:

**URL solo con valori richiesti:**

`ftp://ftp.companyA.com`

**URL con valori facoltativi di nome utente e numero porta:**

`ftp://companyA:admin@ftp.companyA.com:1433`

**URL con valori predefiniti di nome utente, numero porta e directory di output remota:**

`ftp://companyA:admin@ftp.companyA.com:1433/temp/JTextConn/Default/Out`

**Come il connettore elabora le richieste di chiamata del servizio in un sito**

**remoto:** Quando il connettore viene configurato per l'elaborazione FTP e riceve una richiesta di chiamata di servizio, questo elabora le seguenti fasi:

1. Ottenere il nome del server, il numero di porta, il nome utente, e la password, e gli attributi del metaoggetto o valori predefiniti.
2. Stabilisce una connessione ad un sito FTP remoto per posizionare file di richiesta di servizio dalla directory di evento remota.
3. Carica i file di richiesta dalla directory locale a quella remota.
4. Si disconnette dal server remoto.

Figura 7 illustra l'elaborazione della richiesta locale e remota.

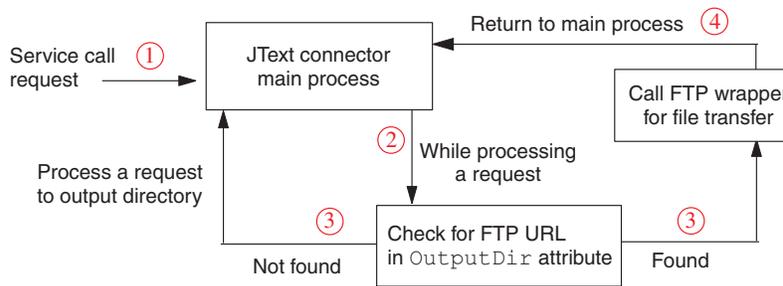


Figura 7. Operazioni di richiesta locale e remota

### Riepilogo delle operazioni di configurazione per l'elaborazione della richiesta:

Per configurare il connettore affinché utilizzi un file system FTP remoto per l'elaborazione della richiesta, specificare i seguenti valori di configurazione:

- Specificare l'URL FTP nell'attributo del metaoggetto `OutputDir`. Alternativamente, specificare il nome e la password di un utente con privilegi per connettersi al server FTP ed eseguire le operazioni FTP.
- Se non si specifica il nome di accesso e la password nell'attributo del metaoggetto `OutputDir`, farlo negli attributi del metaoggetto `FTPUserId` e `FTPPassword`.
- Se non si specifica la porta nell'attributo del metaoggetto `OutputDir`, il connettore utilizza la porta predefinita.
- Per configurare il connettore da gestire con il server FTP MVS quando il sistema remoto è MVS, specificare MVS nell'attributo `FTPOSP1at form`.

### Note sulla configurazione del connettore per il trasferimento FTP

La seguente funzione si applica al trasferimento di dati FTP:

- Il connettore utilizza il modo Binario di trasferimento dati quando si eseguono operazioni FTP.
- Il connettore non utilizza il trasferimento di dati FTP se il valore dell'attributo del metaoggetto `EventDir` o `OutputDir` non inizia con `ftp://`.
- Durante l'elaborazione dell'evento, se l'oggetto business dell'evento contiene un metaoggetto secondario dinamico con un attributo `InFileName`, il connettore popola questo attributo con il percorso completo del file specificato in `FTPLocalEventDir`, ma non il percorso sul sistema remoto.
- I valori immessi negli attributi del metaoggetto `EventExt` e `FTPRenameExt` non possono essere gli stessi; se fossero gli stessi, il connettore prende continuamente i file che aveva preso precedentemente.
- Il connettore non supporta le dimensioni del file che non sono supportate da un FTP.
- E' necessario considerare la sensibilità al maiuscolo/minuscolo per i nomi dei file, le estensioni e altri componenti in accordo con la piattaforma del sito FTP.
- Il trasferimento di file da un sito FTP remoto potrebbe influenzare la prestazione del connettore.
- Quando i dati vengono scambiati a o da un sito FTP remoto, viene presentata una possibilità che i dati possano essere corrotti o persi a causa di una perdita di connessione di rete o problemi simili.
- Il broker di integrazione non conserva nessun tipo di cache di connessione o pool. Le connessioni sono aperte o chiuse per ogni ciclo di polling ed elaborazione di richiesta. La latenza di rete e altre configurazioni al di fuori del controllo del connettore ne possono influenzare le prestazioni.

- Il valore specificato per l'attributo del metaoggetto `FTPLocalEventDir` non può essere specificato come valore dell'attributo del metaoggetto `EventDir` di qualsiasi metaoggetto che non specifica i valori FTP. Questa limitazione evita che il connettore utilizzi i valori specificato in tipi diversi di oggetti business nella stessa directory che richiede tipi totalmente diversi di elaborazione.
- Se la directory di evento remota o di output specificata alla fine dell'URL FTP non esiste, il connettore termina quando interagisce con il sito FTP. Non si arresta all'avvio del connettore.
- Quando si configura il connettore per l'elaborazione dei file su un server FTP, è necessario configurare il server FTP per utilizzare le impostazioni Linux richieste da Apache Commons Net API.

## Configurazione FTP sicuro

L'adattatore JText utilizza il protocollo SSL (Secure Socket Layer) per stabilire un canale di comunicazione sicuro tra l'adattatore e il server FTP. Per questa funzione, il server FTP deve essere un server FTP sicuro che supporta questo protocollo. L'adattatore JText fa uso del pacchetto IBM JSSE che è l'implementazione del protocollo SSL. L'adattatore funziona in modo FTP passivo e SSL implicito. Questa sezione descrive le configurazioni aggiuntive richieste per utilizzare la funzione FTP sicura dell'adattatore JText.

Per configurare un FTP sicuro:

1. Installare e configurare un server FTP sicuro che supporta il protocollo SSL. L'adattatore JText utilizza un protocollo SSL per trasferire i dati tra l'applicazione client ed il server FTP. Quindi, è necessario installare un server FTP sicuro che supporta il protocollo SSL e configurato in modo appropriato per la comunicazione SSL. Il server dovrebbe avere una chiave privata ed un certificato.
2. Se applicabile, configurare le impostazioni Firewall. L'adattatore JText utilizza un modo FTP passivo di trasferimento di dati con il server FTP sicuro. Quindi, se c'è un firewall tra il client ed il server, potrebbe essere necessario configurare le impostazioni del firewall abilitare questa modalità.
3. Impostare l'archivio di sicurezza del client. Durante la comunicazione SSL, il server invia i suoi certificati al client per la verifica. Il client verifica il certificato per accertarsi che sia in comunicazione con il server previsto. Per abilitare questo processo di verifica, il certificato del server dovrebbe essere presente nell'archivio sicuro del client. Il certificato del server può essere importato nell'archivio sicuro del client utilizzando il programma di utilità `keytool`, ad esempio:

```
keytool -import -v -alias serverCert -file server.cert
-keystore clientTrustStore
```

dove `server.cert` è il certificato del server e `clientTrustStore` è l'archivio sicuro del client.

4. Impostare la proprietà di sistema `trustStore` nello script di avvio dell'adattatore. Lo script di avvio dell'adattatore dovrebbe includere la seguente proprietà di sistema,
 

```
-Djavax.net.ssl.trustStore=C:\MyKeyStore\clientTrustStore
```

dove `clientTrustStore` è l'archivio sicuro del client come specificato nel Passo 3.

5. Verificare che si stanno utilizzando gli `ftp` nell'URL FTP. Per comunicazioni sicure, l'URL del FTP dovrebbe avere gli `ftp` come il protocollo, ad esempio:
 

```
ftps://host:port/ftpdire
```

## Specificare un gestore dati

Per specificare un gestore dati che JText Connector deve utilizzare, eseguire le operazioni di seguito riportate:

1. Determinare il formato utilizzato dall'applicazione con la quale il connettore JText comunica. Notare che solo una classe gestore dati può essere registrata per ogni tipo di formato dato.
2. Configurare i seguenti oggetti secondati del metaoggetto JText di livello superiore:
  - `EventDataHandler`—Per specificare il metaoggetto del gestore dati da utilizzare per l'elaborazione dell'evento (stringa di oggetto business o matrice dati per la conversione oggetto business).
  - `OutputDataHandler`—Per specificare il metaoggetto del gestore dati da utilizzare per l'elaborazione della richiesta (conversione da oggetto business a stringa dell'oggetto business o matrice di byte).

### Modifica del gestore dati specificato

Per modificare il gestore dati dalla predefinita fornita (in una diversa fornita a un gestore dati personalizzato) procedere nel modo seguente:

- Verificare che il connettore supporti l'oggetto business specificato come valore predefinito negli attributi `EventDataHandler` e `OutputDataHandler`.
- Verificare che la classe del file jar che contiene il gestore dati sia inclusa nel percorso di classe quando il connettore viene avviato. Se si utilizza il gestore dati fornito, o si aggiunge un gestore dati personalizzato, al file `CustDataHandler.jar` (come consigliato in *Data Handler Guide*), il file viene incluso nello script di avvio fornito (`start_JText.bat` o `connector_manager_JText.sh`).
- Verificare di aver specificato un valore `EndBODEl imi ter` appropriato per il gestore dati che si sta utilizzando.

Per informazioni sulla creazione di un gestore dati, consultare *Data Handler Guide*.

## Creazione di un metaoggetto JText per un oggetto business specifico

Quando si crea un metaoggetto JText per un oggetto business specifico, ridenominare il metaoggetto per identificare l'oggetto business particolare. Ad esempio, per creare i metaoggetti per gli oggetti business `Customer` e `Item`, è necessario ridenominare i metaoggetti `MO_JTextConnector_Customer` e `MO_JTextConnector_Item`.

**Suggerimenti:** Utilizzare i metaoggetti predefiniti quando tutti gli oggetti business da scrivere nei file hanno la stessa configurazione. In altre parole, tutti i file di testo risiedono nella stessa directory di evento e sono scritte nella stessa directory di output, utilizzano lo stesso gestore dati e hanno la stessa estensione file (o vanno messe nello stesso file). Creare i propri metaoggetti se il connettore deve utilizzare elaborazioni diverse per diversi oggetti business su richiesta, o se le istruzioni di elaborazione specifiche sono richieste per il polling. Se si creano metaoggetti separati per oggetti business specifici, il connettore utilizza i metaoggetti per le richieste del broker di integrazione e le operazione di fornitura della sottoscrizione.

Tutti gli oggetti business per i quali non c'è bisogno di creare un metaoggetto sono configurati dai valori nel metaoggetto `MO_JTextConnector_Default` predefinito. Per la definizione dell'oggetto business per questo metaoggetto predefinito, vedere la directory `\repository\JText`.

## Letture di più oggetti business di tipi diversi dallo stesso file

Se un file di testo contiene più oggetti business di tipi diversi, è necessario utilizzare il metaoggetto `MO_JTextConnector_Default` e assicurare che gli attributi `EventExt` e `EventDir` puntano correttamente alla directory dove si trova questo file di esempio. Ogni oggetto business nel file deve essere separato dallo stesso delimitatore.

I gestori dati forniti possono determinare il nome di ogni oggetto business dalla stringa di immissione. In altre parole, quando si utilizza il metaoggetto `JText` di livello superiore e i gestori dati forniti, non è necessario utilizzare il delimitatore `<EndBO:BOName>` per identificare ogni tipo di oggetto business in un file che contiene più tipi.

Se si sviluppa un gestore dati personalizzato per convertire le stringhe di oggetto business a oggetti business, verificare che questo possa interpretare il tipo di oggetto business dalla stringa di input.

## Specifiche dei valori per gli attributi `ObjectEventID`

Non è necessario aggiungere gli attributi `ObjectEventId` alle stringhe di oggetto business. Per gli oggetti business di notifica degli eventi, il framework del connettore popola questi attributi dell'oggetto business se gli ID non sono popolati dal connettore.

Per gli oggetti business di richiesta della chiamata del servizio, gli attributi `ObjectEventId` sono ignorati o inclusi nella stringa scritta in un file. Se gli attributi `ObjectEventId` sono inclusi nel file di output dipende dal gestore dati utilizzato.

## Impostazione di una seconda istanza di un connettore `JText`

Per impostare una seconda istanza del connettore `JText`, seguire questa procedura:

1. Eseguire una copia della directory del connettore `JText` e della directory di `repository` e ridenominarle. Ad esempio, presuppone di denominare la definizione del connettore `JText2`. Dopo aver creato la seconda directory, la struttura della directory somiglia a quanto segue:  

```
\connectors\JText  
\connectors\JText2  
\repository\JText  
\repository\JText2
```
2. Copiare tutti i metaoggetti per il connettore `JText` (dovrebbero essercene almeno due) e modificare il nome degli oggetti business. Ad esempio, per il connettore `JText2`, modificare i nomi da `MO_JText2ConnectorBOName` in `MO_JText2ConnectorBOName`.

Ci sono due modi per poter copiare i metaoggetti:

- Creare un file di testo che contiene il metaoggetto `MO_JText2Connector_BOName` e quello secondario. Utilizzare una ricerca dell'editor di testo e sostituire l'opzione per sostituire `MO_JTextConnector_` con `MO_JText2Connector_BOName`.
- Utilizzare `Business Object Designer Express` per copiare i metaoggetti uno alla volta.

- Importante:** Prima di poter manipolare una definizione dell'oggetto business in Business Object Designer Express, è necessario copiare il testo dalla parte superiore del file `\repository\ReposVersion.txt` alla parte superiore di ogni file di definizione.
3. In Connector Configurator Express, copiare la definizione del connettore e ridenominarlo su `JText2Connector`. Modificare i metaoggetti supportati e gli oggetti business.
  4. Copiare i nuovi file di definizione nel repository. Per utilizzare Business Object Designer Express per copiare le definizioni di oggetto business nel repository, selezionare il sottomenu Salva sul server dal menu File. Alternativamente, su InterChange Server Express, eseguire le fasi successive per copiare le definizioni dell'oggetto business nel repository dal sistema operativo:
    - a. Copiare il testo dalla parte superiore del file `\repository\ReposVersion.txt` alla parte superiore di ogni file di definizione.
    - b. Utilizzare il seguente comando `repos_copy` da copiare nei nuovi metaoggetti e oggetti business:

```
repos_copy -sServerName -iFileName
```
  5. Aggiornare il programma di utilità di amministrazione del broker di integrazione per verificare i nuovi oggetti business.
  6. Per Linux, eseguire una copia dello script del gestore connettore esistente per il connettore `JText` e modificare i parametri per fare riferimento a `JText2`. Per Windows, eseguire una copia del collegamento esistente per il connettore `JText` e modificare i parametri per fare riferimento a `JText2`, e modificarlo per puntare alla directory `JText2` piuttosto che alla directory `JText`.
  7. Per i sistemi i5/OS, aggiungere la nuova istanza del connettore alla Console.
  8. Riavviare il broker di integrazione.
  9. In Linux, eseguire lo script del gestore del connettore. In Windows, fare clic sul collegamento. In i5/OS, utilizzare la Console per avviare il connettore.

## Regolazione delle prestazioni del connettore JText

Per regolare la prestazione del polling del connettore `JText`, impostare le proprietà di configurazione del seguente connettore come di seguito descritto.

- `PollQuantity` – Questa proprietà imposta il numero massimo di oggetti business che il connettore può fornire al broker di integrazione in una chiamata singola per eseguire il poll degli eventi. Se si imposta `PollQuantity` su un valore alto, il connettore prova ad inoltrare più oggetti business in un poll. Questo può migliorare le prestazioni ad aiutare a ripulire le code interne e l'utilizzo di memoria.

Abilitazione del connettore per inviare grosse quantità di oggetti business al broker di integrazione, tuttavia, può influenzare altri componenti di integrazione business. Ad esempio, se il sistema di coda del messaggio è stato impostato con i valori predefiniti, la coda si può riempire velocemente se il connettore `JText` invia molti oggetti business ampi attraverso il sistema. Quindi, quando si regolano le prestazioni, ricordarsi che c'è un'impostazione di prestazione ottimale per `PollQuantity`.

- `PollFrequency` – Questa proprietà di configurazione del connettore specifica la quantità di tempo tra le azioni di polling. Impostazione di questa proprietà su un tempo più lungo rallenta il connettore durante l'elaborazione dell'evento. L'impostazione su un tempo più breve assicura che gli eventi vengono impostati, convertiti su oggetti business e forniti velocemente.

In altre parole, il connettore prende nuovi file durante ogni chiamata del poll. Se il connettore non esegue il poll spesso, ci vuole più tempo per fornire i file che si

affollano nella directory di evento. Se il connettore esegue il poll frequentemente, questo prende i file più spesso e li consegna più frequentemente.

Più frequentemente il connettore esegue il poll degli eventi, tuttavia, e minore è il tempo che ha a disposizione per le richieste di elaborazione. Se si utilizza il connettore principalmente per l'elaborazione della richiesta, impostare `PollFrequency` su un valore più basso se si utilizza il connettore principalmente per l'elaborazione dell'evento.

Così come con la proprietà di configurazione `PollQuantity` precedentemente discussa, l'impostazione di `PollFrequency` su un valore estremo, come un tempo lungo o breve, può influenzare la prestazione di altri componenti di integrazione business.

- `FTPPollFrequency` – Questa proprietà di configurazione connettore specifica quanto frequentemente il connettore esegue il poll di un server FTP misurato nel numero di cicli di poll standard. Ad esempio, se `PollFrequency` è impostato su 10000, e `FTPPollFrequency` è impostato su 6, il connettore esegue il poll della directory di evento locale ogni 10 secondi ed esegue il poll della directory remota ogni 60 secondi. Il connettore esegue il poll del FTP solo se si specifica un valore per questa proprietà. Se `FTPPollFrequency` valuta 0 o vuoto, il connettore non esegue il poll di FTP.

In sintesi, il miglior approccio per migliorare le prestazioni durante il polling è di impostare `PollQuantity`, `PollFrequency`, e `FTPPollFrequency` in modo che siano complementari l'uno per l'altra.

## Generazione file di esempio per la verifica

E' possibile che si desideri generare un file che somigli a quello di input che il connettore `JText` aspetta. Questo file può supportare nell'impostazione dei formati di output nell'applicazione di origine. E' inoltre possibile utilizzare un file di esempio per la verifica.

Su `InterChange Server Express`, il modo più facile di generare un file simile a quello di input è il seguente:

1. Creare una collaborazione pass-through che prende l'oggetto business come input e lo invia alla destinazione che va scritta al di fuori del file.
2. Associare la porta di origine al connettore che supporta quell'oggetto business e può essere emulato da `Test Connector`.
3. Associare la porta di destinazione al connettore `JText`.
4. Immettere valori di esempio per l'oggetto business nel `Test Connector`, ed inviare l'oggetto business al connettore `JText`. Il connettore `JText` scrive i valori nel file di output nel formato configurato.

Questo processo consente di vedere più oggetti business scritti in un file singolo che è possibile utilizzare come immissione durante la verifica.

## Generazione degli oggetti business di esempio per la verifica

E' possibile che si desideri generare gli oggetti business che somigliano a quelli che ci si aspetta dal connettore `JText`. E' possibile popolare gli oggetti business con i valori da utilizzare durante la verifica.

Per fare in modo che il connettore per generare automaticamente la maschera dell'oggetto business, utilizzare la proprietà di configurazione `GenerateTemplate`. E' possibile generare una definizione per l'oggetto business che il connettore supporta.

Il connettore utilizza il valore della proprietà `GenerateTemplate` per creare un'istanza di un oggetto business serializzato quando il connettore si avvia. Un oggetto business **serializzato** è la rappresentazione della stringa dell'oggetto business che il gestore dati crea. Utilizzare `Connector Configurator Express` per specificare i nomi degli oggetti business per questa proprietà.

La sintassi per questa proprietà è `BOName;BOName`, dove il nome di un oggetto business specifico viene sostituito per `BOName`. Il maiuscolo/minuscolo è significativo. Per specificare più di un oggetto business, separare i nomi con un punto e virgola, come in `Customer;Item`. La punteggiatura finale non è richiesta. Le maschere per questi oggetti business vengono creati la prossima volta che si avvia il connettore.

Le maschere generate contengono i valori predefiniti forniti che sono impostati per gli attributi degli oggetti business nella definizione dell'oggetto business. Se non ci sono valori predefiniti forniti per un attributo, viene ignorato (utilizzando `CxIgnore`) o lasciato vuoto (utilizzando `CxBlank`). Un oggetto business secondario viene creato per ogni oggetto business secondario con cardinalità singola e due istanze identiche di un oggetto business secondario vengono creati per gli oggetti business a più cardinalità.

Per iniziare a generare maschere per un oggetto business specificato, avviare il connettore. Il connettore scrive la maschera nello stesso file come il file di output. Se non si desidera utilizzare questa funzione, lasciare la proprietà `GenerateTemplate` vuota.

---

## Capitolo 5. Risoluzione di problemi del connettore JText

Questo capitolo include le seguenti informazioni per aiutare a diagnosticare i problemi con il connettore JText.

- “Registrazione del messaggio di errore”
- “Problemi con la denominazione del metaoggetto”
- “Problemi con il trigger degli eventi”
- “Gestione errore JText” a pagina 82
- “File log di evento” a pagina 82
- “Recupero errori” a pagina 83
- “Recupero dagli errori del delimitatore oggetto business” a pagina 85
- “Recupero dagli errori di sottoscrizione” a pagina 85
- “Recupero dagli errori di formattazione” a pagina 85
- “Recupero dall’invio di errori” a pagina 86
- “Gestori dati e oggetti business supportati” a pagina 86

---

### Registrazione del messaggio di errore

I messaggi di errore sono registrati nel file di log del connettore standard, STDOUT, o nel file specificato dalla proprietà `LogFileName` del connettore standard.

Gli errori sono anche registrati nel file di log degli eventi. Per ulteriori informazioni sul file di log degli eventi, consultare “File log di evento” a pagina 82.

---

### Problemi con la denominazione del metaoggetto

Durante l’avvio del connettore, il seguente messaggio di errore significa che il nome del metaoggetto non corrisponde al nome istanza del connettore.

Sottoscrizione errata: JText\_Customer non ha supporto di M0:  
è stata annullata la sottoscrizione di questo B0."

Se il nome del metaoggetto non corrisponde al nome dell’istanza del connettore, il metaoggetto non riconosce l’oggetto business supportato dal connettore. Per evitare ciò, denominare il metaoggetto affinché corrisponda all’istanza del connettore. Ad esempio, un metaoggetto denominato `M0_JTextConnector2_Default` riconosce gli oggetti business supportati dal connettore JText2.

---

### Problemi con il trigger degli eventi

Il connettore ignora i file di evento con i seguenti problemi di delimitatore:

- L’attributo `EndB0Delimiter` nel metaoggetto di livello superiore è impostato su un valore valido, come il segno più (+) o il simbolo (|), ma il file di evento non contiene il delimitatore specificato alla fine di ogni oggetto business.
- Il connettore viene configurato per cercare il delimitatore dell’oggetto business `EndB0:B0Name`, ma il file di evento non contiene questo delimitatore. Il connettore registra un messaggio di avviso che stabilisce:

Impossibile creare Workunits dal file *filename*. Controllare `EndB0Delimiter` nel file.

In entrambi i casi precedentemente riportati, il file resta nella directory degli eventi senza alcuna modifica.

Il connettore conserva anche il file nella directory evento senza modifiche quando si verificano gli errori di unità mentre si sta accedendo, aprendo o chiudendo un file. Ad esempio, se il sistema prova ad accedere ad un file quando la memoria non è sufficiente, il connettore ignora il file.

---

## Gestione errore JText

Per il connettore JText, possono verificarsi i seguenti tipi di errore:

*Tabella 11. Tipi di errore JText*

Tipo di errore	Descrizione
Errori delimitatore oggetto business	Gli errori del delimitatore dell'oggetto business si verificano quando l'attributo <code>EndBODElimiter</code> nel metaoggetto di livello superiore è impostato su un valore valido ed il file di evento contiene il delimitatore specificato alla fine di ogni oggetto business, ma i dati stessi utilizzano il valore di delimitatore nel testo. Quando il connettore rileva il valore del delimitatore nel testo, invia una stringa dell'oggetto business parziale al programma di formattazione, che genera un errore nell'elaborazione. In questo caso, il connettore scrive l'evento al file <code>filename_timestamp.fail</code> che contiene record per tutti gli oggetti business che hanno rilevato errori del delimitatore.
Errori di sottoscrizione	Possono verificarsi se il connettore può trovare il delimitatore dell'oggetto business e richiamare il nome dell'oggetto business, ma l'oggetto business non è sottoscritto. In questo caso, un evento viene inviato al file <code>filename_timestamp.unsub</code> che contiene record per tutti gli oggetti business non sottoscritti.
Errori di formattazione	Si possono verificare se il connettore trova il delimitatore con un nome dell'oggetto business che non corrisponde al nome oggetto business di immissione o il formato nel file dell'oggetto business non corrisponde al formato del metaoggetto. Un evento viene inviato al file <code>filename_timestamp.fail</code> che contiene record per tutti gli oggetti business con errori di formattazione.
Invio errori	Può verificarsi se il connettore prova ad inviare un oggetto business quando il broker di integrazione non funziona. Se l'operazione di Invio non riesce, viene inviato un evento al file <code>filename_timestamp.fail</code> che contiene record per tutti gli oggetti business che non sono stati inviati correttamente.

## File log di evento

Le connettore registra le informazioni sugli oggetti business elaborati in modo corretto nel file `event.log`. Se il connettore si arresta prima di elaborare tutti gli oggetti business in un file di evento, questo utilizza il file di log durante il recupero per assicurare che invii ogni oggetto business solo una volta al broker di integrazione.

Il formato del file di log è:

`EventFileName::1,2,n`

dove `EventFileName` è il nome del file di evento corrente e ogni numero rappresenta il numero di sequenza di un oggetto business elaborato in modo corretto nel file.

Ad esempio, presupporre che il connettore ha elaborato correttamente tre dei primi quattro oggetti business nel file Customer.in e che il secondo oggetto business non è riuscito nell'elaborazione. Presupporre inoltre che il connettore non ha ancora finito l'elaborazione di Customer.in. In questo caso, il file event.log potrebbe essere come il seguente su Linux:

```
$ProductDir/JText/Event/Customer.in:: 1,3,4
```

Windows:

```
C:\JText\Event\Customer.in:: 1,3,4
```

i5/OS:

```
/QIBM/UserData/WebBIICS/JText/Event/Customer.in:: 1,3,4
```

Se il connettore ha generato un errore prima di elaborare l'intero file Customer.in, all'avvio il connettore utilizza le informazioni nel file di log per ripristinare l'elaborazione del file di evento al momento in cui ha arrestato l'elaborazione. Il connettore legge il log per ottenere il nome del file di evento da recuperare e l'ultimo numero della sequenza dell'oggetto business. Allora il connettore inizia ad inviare al broker di integrazione tutti gli oggetti business nel file di evento il cui numero di sequenza è maggiore dell'ultimo numero del file di log. Ad esempio, dato il file riportato, il connettore inizia l'elaborazione del quinto oggetto business nel file Customer.in.

Il connettore conserva i contenuti del file di log nella memoria per potenziare la prestazione. Accede al file sul disco solo per aggiornarlo con una nuova voce. Il connettore legge il file di log solo al momento del recupero.

Per informazioni su come il connettore utilizza il file event.log nel processo di recupero, consultare "Recupero errori".

## Recupero errori

**Nota:** Le seguenti fasi di recupero non si applicano se si verifica l'errore del disco o se il disco è pieno.

Per recuperare dagli errori durante la notifica dell'evento, il connettore procede nel modo seguente:

1. Il connettore elabora le stringhe di oggetto business dal file di evento. Quando elabora con successo una voce, il connettore registra la voce nel file event.log. Esso lo scrive in un file in una directory di archivio (specificato nell'attributo del metaoggetto ArchiveDir).
  - Se nessun degli oggetti business nel file di evento non è riuscito nell'elaborazione, il connettore archivia quelli elaborati correttamente nel file di archivio con l'estensione specificata in un file di archivio con l'estensione specificata nell'attributo SuccessArchiveExt.
  - Se gli oggetti business nel file di evento non è riuscito nell'elaborazione, il connettore archivia quelli elaborati correttamente nel file di archivio con l'estensione specificata in un file di archivio con l'estensione specificata nell'attributo PartialArchiveExt.
  - Dopo aver scritto gli oggetti business nel file specificato nell'attributo SuccessArchiveExt, se l'oggetto business non riesce nell'elaborazione, il connettore cambia l'estensione di questo file in uno specificato in PartialArchiveExt.

I valori predefiniti forniti per queste estensioni sono .success e .partial.

2. Se si verifica un errore, il connettore procede nel modo seguente:
  - Errori di sottoscrizione— il connettore crea il file di archivio nella directory di archivio con l'estensione specificata nell'attributo del metaoggetto `UnsubscribedArchiveExt`. Il valore predefinito fornito per questa estensione è `.unsub`.
  - Errori di formattazione o di invio—il connettore crea il file di archivio nella directory di archivio con l'estensione specificata nell'attributo del metaoggetto `FailArchiveExt`. Il valore predefinito fornito per questa estensione è `.fail`.
  - Errori di delimitatore oggetto business—il connettore crea il file di archivio nella directory di archivio con l'estensione specificata nell'attributo `FailArchiveExt`. Esegue la copia di riserva del file di evento spostandolo nella directory di archivio e cambiando l'estensione in quella specificata in `OriginalArchiveExt`.

Il connettore non registra gli oggetti business con errori su `event.log`.

3. Dopo che il connettore elabora tutti gli oggetti business in un file di evento, annulla il file `event.log` ed inizia a scrivere le voci dal file di evento successivo.
4. Se il connettore si arresta prima di elaborare gli oggetti business in un file di evento, questo utilizza le informazioni in `event.log` per determinare dove iniziare l'elaborazione durante il processo di recupero. Quando esegue la copia di riserva, il connettore controlla se ci sono voci nel file di log.

- Se non ci sono voci, il connettore invia tutti gli oggetti business nel file di evento al broker di integrazione.
- Se ci sono voci, il connettore utilizza queste informazioni per ripristinare un file di evento nel punto in cui è stata arrestata l'elaborazione. Il connettore legge il log per ottenere il nome del file di evento da recuperare e l'ultimo numero della sequenza dell'oggetto business. Allora il connettore invia al broker di integrazione tutti gli oggetti business nel file di evento il cui numero di sequenza è maggiore dell'ultimo numero del file di log. Ad esempio, se il file di evento contiene 15 oggetti business e l'ultimo numero di sequenza nel file di log è 8, il connettore invia gli ultimi sette oggetti business al broker di integrazione.

L'utilizzo del file di log evita che il connettore invii lo stesso evento più volte al broker di integrazione. Il connettore conserva il file di log nella memoria per potenziare la prestazione. Il connettore accede al file sul disco solo per aggiornarlo con una nuova voce e legge il file di log solo al momento del recupero.

Se si imposta la proprietà di configurazione "EventRecovery" a pagina 30 su riprova, il connettore all'avvio recupera automaticamente gli eventi rimanenti da un file elaborato precedentemente. Tuttavia, se si imposta la proprietà su arresta, il connettore termina durante l'avvio se ci sono eventi da recuperare.

5. Per recuperare dagli errori che si sono verificati durante il processo di notifica degli eventi, è necessario riavviare il connettore. Prima di procedere in questo modo, tuttavia, procedere nel modo seguente:
  - Esaminare i file che il connettore ha creato per gli oggetti business non riusciti e non sottoscritti. Eseguire le correzioni appropriate così che le stringhe dell'oggetto business possano essere elaborate con successi quando si avvia il connettore.
  - Copiare i file appropriati dalla directory di archivio alla directory di evento e modificare tutte le estensioni `.fail` o `.unsub` nell'estensione specificata nell'attributo `EventExt` (per impostazione predefinita, `.in`). Per facilitare la conservazione dei record, ridenominare questi file in modo significativo. Ad esempio, ridenominare `Customer.unsub` in `Customer_unsub_resubmit.in`.

- Potrebbe essere necessario eseguire altre fasi per il recupero, in base al tipo di errore verificatosi.

Le seguenti indicazioni possono aiutare a determinare quali fasi di recupero eseguire, in base al tipo di errore che si è verificato.

### **Recupero dagli errori del delimitatore oggetto business**

Il connettore scrive l'oggetto business in una directory del file di archivio, dandogli l'estensione specificata nell'attributo del metaoggetto `FailArchiveExt`. Per gestire il recupero da questo errore, procedere nel modo seguente:

1. Verificare che il file di evento contiene il delimitatore dell'oggetto business, che il delimitatore sia corretto e che non contiene il valore del delimitatore nei dati stessi come testo. Se l'utilizzo del delimitatore non è corretto, correggerlo.
2. Rivedere il file di log del connettore (specificato nell'attributo di configurazione `LogFile`) per determinare altre ragioni per le quali il processo ha generato un errore.
3. Copiare il file dalla directory di archivio nella directory di evento e modificare l'estensione `.fail` nell'estensione specificata nell'attributo `EventExt` (per impostazione predefinita, `.in`). Per facilitare la conservazione dei record, ridenominare il file in modo significativo. Ad esempio, ridenominare `Customer.fail` in `Customer_delimiter_error.in`.

### **Recupero dagli errori di sottoscrizione**

Il connettore scrive l'oggetto business in un file posizionato nella directory di archivio, dandogli l'estensione specificata nell'attributo del metaoggetto `UnsubscribedArchiveExt`. Per gestire il recupero da questo errore, procedere nel modo seguente:

1. Aprire il file archiviato, trovare la stringa dell'oggetto business e verificare che il nome dell'oggetto business e il verbo siano sottoscritti. Eseguire le correzioni appropriate, se necessario.
2. Verificare che il broker di integrazione sia in esecuzione.
3. Copiare il file dalla directory di archivio nella directory di evento e modificare l'estensione `.unsub` nell'estensione specificata nell'attributo `EventExt` (per impostazione predefinita, `.in`). Per facilitare la conservazione dei record, ridenominare il file in modo significativo. Ad esempio, ridenominare `Customer.unsub` in `Customer_unsub_resubmit.in`.

### **Recupero dagli errori di formattazione**

Il connettore scrive l'oggetto business in un file posizionato nella directory di archivio, dandogli l'estensione specificata nell'attributo del metaoggetto `FailArchiveExt`. Per gestire il recupero da questo errore, procedere nel modo seguente:

1. Aprire il file archiviato e verificare che:
  - Il formato della stringa oggetto business corrisponda al formato previsto nel metaoggetto. Se c'è una non corrispondenza, modificare il tipo di formato nel metaoggetto o nella stringa dell'oggetto business.
  - La sintassi di formattazione della stringa dell'oggetto business è corretta. Se non è corretta, correggerla.
2. Copiare il file dalla directory di archivio nella directory di evento e modificare l'estensione `.fail` nell'estensione specificata nell'attributo `EventExt` (per impostazione predefinita, `.in`). Per facilitare la conservazione dei record, ridenominare il file in modo significativo. Ad esempio, ridenominare `Customer.fail` in `Customer_fail_formatting.in`.

## Recupero dall'invio di errori

Il connettore scrive l'oggetto business in un file posizionato nella directory di archivio, dandogli l'estensione specificata nell'attributo del metaoggetto `FailArchiveExt`. Per gestire il recupero da questo errore, procedere nel modo seguente:

1. Verificare che tutti i componenti del sistema di integrazione business siano in esecuzione.
2. Copiare il file dalla directory di archivio nella directory di evento e modificare l'estensione `.fail` nell'estensione specificata nell'attributo `EventExt` (per impostazione predefinita, `.in`). Per facilitare la conservazione dei record, ridenominare il file in modo significativo. Ad esempio, ridenominare `Customer.fail` su `Customer_fail_sending.in`.
3. Riavviare il connettore.

## Gestori dati e oggetti business supportati

Se il connettore restituisce un errore che stabilisce che il gestore dati non è stato configurato, verificare che il metaoggetto per il gestore dati sia incluso nell'elenco di oggetti business supportati. L'errore più comune restituito dal connettore stabilisce che `BOPrefix` non è impostato.

L'elenco di oggetti business supportati per `DHFormatter` dovrebbe includere quanto segue:

- `MO_JTextConnector_Default`
- `MO_JTextConnector_BusObjName` (i metaoggetti creati per gli oggetti business specifici)
- Gli oggetti business che devono essere letti da o scritti in un file.
- Il metaoggetto per il gestore dati (specificato nell'attributo `DataHandlerConfigMO` del metaoggetto `MO_JTextConnector_Default`).

---

## Capitolo 6. Migrazione a o aggiornamento al connettore JText

Questo capitolo descrive come aggiornare alla versione 5.6.x del connettore JText dalla versione 5.3.0 e alla versione 4.0.x del connettore JText dalla versione 3.2.0. Descrive inoltre come aggiornare al connettore JText dal connettore Testo.

**Nota:** Le versioni del connettore JText 4.4.x e 4.3.x non contengono nessuna modifica di configurazione specifica. Contengono solo una configurazione facoltativa, nessuna delle quali è necessaria fino a che non si desidera trarre vantaggio dalle nuove opzioni. Consultare la sezione “Novità di questo rilascio” a pagina vii per dettagli.

Gli argomenti inclusi in questo capitolo includono:

- “Aggiornamento scenari”
- “Aggiornamento alla versione 5.6.x dalla versione 5.3.x”
- “Motivi per l’aggiornamento alla versione 4.0.x dalla versione 3.2.0” a pagina 88
- “Aggiornamento alla versione 4.0.x” a pagina 88
- “Motivi per l’aggiornamento dal connettore Text” a pagina 90
- “Aggiornamento al connettore JText” a pagina 90

---

### Aggiornamento scenari

Se si sta aggiornando al rilascio 4.1.x del connettore JText dal rilascio 4.0.x, seguire le istruzioni in “Aggiornamento scenari” a pagina 87.

Se si sta aggiornando al rilascio 4.1.x del connettore JText dal rilascio 4.0.x, seguire le istruzioni in “Aggiornamento alla versione 4.0.x” a pagina 88 e “Aggiornamento scenari” a pagina 87.

Se si sta aggiornando al connettore JText dal connettore Text, seguire le istruzioni in “Aggiornamento al connettore JText” a pagina 90.

---

### Aggiornamento alla versione 5.6.x dalla versione 5.3.x

Questa sezione spiega:

- “Modifiche metaoggetto” a pagina 87
- “Modifiche architettura” a pagina 88

#### Modifiche metaoggetto

Il metaoggetto `MO_JTextConnector_Default` contiene sei nuovi attributi (`FTPTransferType`, `FixedBOSize`, `DataProcessingMode`, `FTPDataStructure`, `MVSSiteCommand`, e `LargeObject`) che hanno potenziato la funzionalità di JText.

È necessario aggiungere questi attributi al metaoggetto di livello superiore JText, e configurare i valori per questi. Utilizzare Business Object Designer Express per aggiungere i nuovi attributi, fornire i valori predefiniti per questi e salvare le modifiche alla definizione.

Aggiungere sei nuovi attributi ad ognuno dei metaoggetti che si sono personalizzati per `MO_JTextConnector_Default`. Ad esempio, se si è creato il proprio

metaoggetto per gli oggetti business Customer e Item, aggiungere i nuovi attributi a questi metaoggetti, fornire i propri valori per questi e salvare le modifiche al repository.

Per ulteriori informazioni, consultare Tabella 8 a pagina 48.

## Modifiche architettura

Questa sezione spiega:

- “Modifiche proprietà di configurazione”

### Modifiche proprietà di configurazione

Per il rilascio 5.6.0, sono state aggiunte due nuove proprietà di configurazione specifiche del connettore:

- **SortFilesOnTimestamp** - È stata aggiunta una nuova proprietà specifica del connettore all'adattatore e consente a questo di prendere file di evento casati sulla data/ora della modifica.
- **NoPoll** - È stata introdotta una nuova proprietà booleana specifica del connettore, per disattivare facoltativamente il polling. Il valore predefinito è false. Quando è impostato su true, l'adattatore elabora solo le richieste e non esegue il poll.

---

## Motivi per l'aggiornamento alla versione 4.0.x dalla versione 3.2.0

Con la versione 4.0.x del connettore JText, la struttura dei metaoggetti richiesti per configurare il connettore è stata nettamente semplificata, semplificando, in questo modo il processo di configurazione.

Se versioni precedenti del connettore utilizzando una struttura del metaoggetto che ha incluso tre livelli della gerarchia e almeno dieci diversi metaoggetti, la versione 4.0.x utilizza solo due metaoggetti e solo due livelli di gerarchia. Questa nuova versione cambia il modo in cui si configura il connettore ma non ne cambia la funzionalità.

---

## Aggiornamento alla versione 4.0.x

Poiché i nuovi metaoggetti utilizzano gli stessi dati di configurazione come versioni precedenti, l'aggiornamento non richiede la modifica dei valori di configurazione. Tuttavia, poiché i nuovi metaoggetti archiviano i dati in attributi denominati in modo diverso in molti meno metaoggetti, l'aggiornamento non richiede le seguenti operazioni:

- Creare nuovi metaoggetti.
- Sostituire il valore di ogni proprietà DefaultValue dell'attributo in ogni nuovo metaoggetto con i valori predefiniti personalizzati nei metaoggetti esistenti.
- Rimuovere tutti i metaoggetti obsoleti dal repository.

IBM WBIS Support fornisce un programma di utilità che automatizza le operazioni precedenti. Per eseguire queste operazioni in modo manuale, eseguire le procedure di seguito riportate:

1. Eseguire una copia di backup del repository utilizzando il programma di utilità repos\_copy. Ad esempio, il seguente comando esegue la copia di backup di tutti i contenuti del repository di Server1 nel file di output, InterChangeRepository.out:  

```
repos_copy -oInterChangeRepository.out -sServer1 -pmypassword
```

2. Per ogni metaoggetti di livello superiore esistente, creare un nuovo metaoggetto con gli stessi attributi del metaoggetto di livello superiore fornito, `MO_JTextConnector_Default`. Ad esempio, se si è creato il proprio metaoggetto per l'oggetto business Customer denominato in base alla convenzione di denominazione precedente (`MO_JTextConnector_Customer`), creare un nuovo metaoggetto per Customer che utilizza la nuove convenzione di denominazione (`MO_JTextConnector_Customer`).
3. Impostare i valori predefiniti dei nuovi metaoggetti in base ai valori nei metaoggetti originali. Vedere Tabella 12 per la corrispondenza tra gli attributi nei metaoggetti originali ed i nuovi.
4. Utilizzare System Manager per eliminare l'insieme originale delle definizioni di metaoggetto dal repository, conservando solo quelli appena creati e `MO_JTextConnector_Default_DHFormatter`.

Tabella 12 illustra la corrispondenza tra gli attributi nuovi e originali, incluso i nomi dei metaoggetti originali. Se più metaoggetti contengono nuovi attributi originali, il metaoggetto `MO_JTextConnector_Default` contiene tutti i nuovi attributi.

Tabella 12. Corrispondenza dei metaoggetti originali e degli attributi con più attributi

Nome metaoggetto originale	Nome attributo originale	Nuovo nome attributo
<code>MO_JTextConnector_BOName_Connector</code>	DummyKey	N/A
<code>MO_JTextConnector_BOName_ArchiveDir</code>	Directory	ArchiveDir
<code>MO_JTextConnector_BOName_ArchiveFileExt</code>	Success	SuccessArchiveExt PartialArchiveExt
	Fail	FailArchiveExt UnsubscribedArchiveExt
<code>MO_JTextConnector_BOName_EventDir</code>	Directory	EventDir
	FileExt	EventExt
<code>MO_JTextConnector_BOName_OutputDir</code>	Directory	OutputDir
	FileExt	OutputExt
	FileSequencingEnabled	FileSeqEnabled
<code>MO_JTextConnector_BOName_FormatType</code>		N/A
<code>MO_JTextConnector_BOName_ServicePolicy</code>	OutputFileName	OutputFileName
	EndBODelimiter	EndBODelimiter
<code>MO_JTextConnector_BOName_FormatService</code>	EventService	EventFormat
	OutputService	OutputFormat

Tabella 12 non include le seguenti informazioni

- Attributi corrispondenti al metaoggetto `MO_JTextConnector_BOName_FormatType`  
Versioni precedenti del connettore richiedono di elencare formati possibili nel metaoggetto `MO_JTextConnector_BOName_FormatType` prima di configurare i formati da utilizzare per i file di evento e di output. Nella nuova struttura nel metaoggetto, è necessario configurare solo i formati da utilizzare per i file di evento e di output. Questa modifica è indicata in Tabella 12 dall'assenza degli attributi di corrispondenza per il metaoggetto `MO_JTextConnector_BOName_FormatType`.
- I metaoggetti per i programmi di formattazione singoli  
Il metaoggetto di livello superiore ha due attributi che contengono un metaoggetto del programma di formattazione. I metaoggetti del programma di formattazione hanno gli stessi attributi della struttura originale e sono utilizzati allo stesso modo. Poiché sono stati scartati tre dei quattro programmi di formattazione, il metaoggetto importante del programma di formattazione è `MO_JTextConnector_BOName_DHFormatter`.

---

## Motivi per l'aggiornamento dal connettore Text

Se il sito al momento utilizza il connettore Text per la comunicazione tra un'applicazione ed un broker di integrazione, considerare la possibilità di aggiornare il connettore JText per i seguenti motivi:

- Prestazione. Il connettore Text elabora solo un file alla volta che può ostacolare la prestazione durante l'elaborazione dei file ampi o un gran numero di file.
- Disponibilità formato. Il connettore Text gestisce pochi tipi di formato.
- Facilità di modifica. Il connettore Text è più difficile da modificare.

Al contrario, il connettore JText può essere configurato a:

- Elaborare più file alla volta.
- Ricercare più posizioni per specifici oggetti business, quindi per aumentare le prestazioni.
- Aggiustare un intervallo più ampio di tipi di formato.

---

## Aggiornamento al connettore JText

Per aggiornare al connettore JText dal connettore Text:

1. Dal CD del prodotto, copiare la directory JText nella directory %CROSSWORLD%connectors.
2. Aprire una finestra di Prompt dei comandi e utilizzare repos\_copy per aggiungere i seguenti due file al repository: CN\_JText.txt e MO\_JTextConnector\_Default.txt.
3. Verificare che sono state create le directory specificate per i seguenti attributi dei metaoggetti: ArchiveDir, EventDir, e OutputDir. Se queste directory sono state create, crearle.
4. Configurare gli attributi del metaoggetto.
5. Sottoscrivere negli oggetti business desiderati.

---

## Appendice. 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 13 a pagina 93.)

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 13 a pagina 93.

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 13 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 13, la frase "RepositoryDirectory è impostato su <REMOTE>" indica che il broker è InterChange Server Express.

Tabella 13. 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> oppure riavviare	Componente riavviare	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 riavviare	Questa proprietà è valida solo se il valore DeliveryTransport è JMS
AdminOutQueue	Nome di coda JMS valido	<CONNECTORNAME>/ADMINOUTQUEUE	Componente riavviare	Questa proprietà è valida solo se il valore DeliveryTransport è JMS
AgentConnections	Da 1 a 4	1	Componente riavviare	Questa proprietà è valida solo quando il valore di DeliveryTransport è MQ oppure 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; altrimenti Componente riavviare	

Tabella 13. 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 componente	Componente riavviare	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	La stringa "true" (questo valore) bidirezionale	Componente riavviare	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	La stringa "true" (questo valore) bidirezionale	Componente riavviare	Questa proprietà è valida solo se il valore di BiDi.Transformation è true. Se il valore di BrokerType è ICS, la proprietà è in 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	La stringa "true" (questo valore) bidirezionale	Componente riavviare	Questa proprietà è valida solo se il valore di BiDi.Transformation è true.
BiDi.Transformation	true oppure false	false	Componente riavviare	Questa proprietà è valida solo se il valore di BrokerType è WAS
BrokerType	ICS	ICS	Componente riavviare	
CharacterEncoding	Qualsiasi codice supportato. L'elenco visualizza questo sottoinsieme: ascii7, ascii8, SJIS, Cp949, GBK, Big5, Cp297, Cp273, Cp280, Cp284, Cp037, Cp437	ascii7	Componente riavviare	Questa proprietà è valida solo per connettori C++.
CommonEventInfrastructure	true oppure false	false	Componente riavviare	
CommonEventInfrastructure	Una stringa URL, ad esempio corbaloc:iiop: host:2809.	Nessun valore predefinito.	Componente riavviare	Questa proprietà è valida solo se il valore di CommonEvent.Transformation è true.
ConcurrentEventTriggeredFlows	Da 1 a 32,767	1	Componente riavviare	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 riavviare	Questa proprietà è valida solo se il valore di Delivery.Transformation 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 13. 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 riavviare	Questa proprietà è valida solo se il valore di Delivery.Transformation è JMS.
DeliveryTransport	IDL o JMS	IDL quando il valore di RepositoryDirectory è <REMOTE>, altrimenti JMS	Componente riavviare	Se il valore di RepositoryDirectory non è <REMOTE>, l'unico valore valido per questa proprietà è JMS.
DuplicateEventElimination	true oppure false	false	Componente riavviare	Questa proprietà è valida solo se il valore di DeliveryTransport è JMS.
EnableOidForFlowMonitoring	true oppure false	false	Componente riavviare	Questa proprietà è valida solo se il valore di BrokerType è ICS.
FaultQueue	Qualsiasi nome di coda valido	<CONNECTORNAME>/FAULTQUEUE	Componente riavviare	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.IBMQueueConnectionFactory	Componente riavviare	Questa proprietà è valida solo se il valore di DeliveryTransport è JMS.
jms.ListenerConcurrency	Da 1 a 32767	1	Componente riavviare	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 riavviare	Questa proprietà è valida solo se il valore di DeliveryTransport è JMS .
jms.NumConcurrentRequests	Intero positivo	10	Componente riavviare	Questa proprietà è valida solo se il valore di DeliveryTransport è JMS .
jms.Password	Qualsiasi password valida		Componente riavviare	Questa proprietà è valida solo se il valore di DeliveryTransport è JMS .
jms.TransportOptimized	true oppure false	false	Componente riavviare	Questa proprietà è valida solo se il valore DeliveryTransport è JMS e il valore di BrokerType è ICS.

Tabella 13. Riepilogo delle proprietà di configurazione standard (Continua)

Nome proprietà	Valori possibili	Valore predefinito	Metodo aggiornato	Note
jms.UserName	Qualsiasi nome valido		Componente riavviare	Questa proprietà è valida solo se il valore di Delivery.Transport è JMS.
JvmMaxHeapSize	Dimensione heap in megabyte	128 M	Componente riavviare	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 riavviare	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 riavviare	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 riavviare	Questa proprietà è valida solo se il valore di Delivery.Transport è MQ.
Locale	Questo è un sottoinsieme di quelle supportate: en_US, ja_JP, ko_KR, zh_CN, zh_TW, fr_FR, de_DE, it_IT, es_ES, pt_BR		Componente riavviare	
LogAtInterchangeEnd	true oppure false	false	Componente riavviare	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 riavviare	
MonitorQueue	Qualsiasi nome di coda valido	<CONNECTORNAME> /MONITORQUEUE	Componente riavviare	Questa proprietà è valida solo se il valore di è true e
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
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

Tabella 13. Riepilogo delle proprietà di configurazione standard (Continua)

Nome proprietà	Valori possibili	Valore predefinito	Metodo aggiornato	Note
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 riavviare	
PollFrequency	Un numero intero positivo (in millisecondi)	10000	Dinamica se il broker è ICS; altrimenti Componente riavviare	
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 riavviare	
RepositoryDirectory	<REMOTE> se il broker è ICS; altrimenti qualsiasi	Per ICS, il valore è impostato su <REMOTE>. Per directory locale valida.	Riavvio <REMOTE>	
RequestQueue	Nome di coda JMS valido	<CONNECTORNAME>/REQUESTQUEUE	Componente riavviare	Questa proprietà è valida solo se il valore di DeliveryTransport è JMS
ResponseQueue	Nome di coda JMS valido	<CONNECTORNAME>/RESPONSEQUEUE	Componente riavviare	Questa proprietà è valida solo se il valore di DeliveryTransport è JMS.
RestartRetryCount	Da 0 a 99	3	Dinamica se ICS; altrimenti Componente riavviare	
RestartRetryInterval	Un valore in minuti da 1 a 2147483647	1	Dinamica se ICS; altrimenti Componente riavviare	
ResultSetEnabled	true oppure false	false	Componente riavviare	Utilizzato solo da connettori che supportano DB2II.
ResultSetSize	Intero positivo	0 (significa che la dimensione è illimitata)	Componente riavviare	Utilizzato solo da connettori che supportano DB2II.  Questa proprietà è valida solo se il valore di ResultSetEnabled è true.
RHF2MessageDomain	mrm or xml	mrm	Componente riavviare	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.

Tabella 13. Riepilogo delle proprietà di configurazione standard (Continua)

Nome proprietà	Valori possibili	Valore predefinito	Metodo aggiornato	Note
SynchronousRequest Queue	Qualsiasi nome di coda valido	<CONNECTORNAME>/SYNCHRONOUSREQUEST QUEUE	Componente riavviare	Questa proprietà è valida solo se il valore di DeliveryTransport è JMS.
SynchronousRequestTimeout	0 a qualsiasi numero (millisecondi)	0	Componente riavviare	Questa proprietà è valida solo se il valore di DeliveryTransport è JMS.
SynchronousResponse Queue	Qualsiasi nome di coda valido	<CONNECTORNAME>/SYNCHRONOUSRESPONSE QUEUE	Componente riavviare	Questa proprietà è valida solo se il valore di DeliveryTransport è JMS.
TivoliMonitorTransaction Performance	true oppure false	false	Componente riavviare	
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 a qualsiasi numero (millisecondi)	0	Componente riavviare	Questa proprietà è valida solo se il valore di BrokerType è ICS.
XMLNamespaceFormat	short oppure long	short	Riavvio dell'agente	Questa proprietà è valida solo 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 applicazione. 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 sopracitato.

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.

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 gestore dati 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 **Gestore dati** 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 **Gestore dati** 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 Se 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 valore predefinito è JMS.

### 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 cosa 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

`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:

```
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 `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 `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 `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 messaggio 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 gestore dati per questo tipo mine, ignorerà il messaggio.
- Il connettore elabora il primo allegato BO. Il gestore dati 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. Il gestore dati 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.

## ResultSetEnabled

La proprietà ResultSetEnabled 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.

## ResultSetSize

La proprietà ResultSetSize 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 ResultSetEnabled 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 101.

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 dei 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.



---

## Indice analitico

### A

Adattatore JText  
aggiornamento 87  
aggiornamento procedure 90  
ambiente 15  
architettura 2  
arresto 40  
avvio 39  
componenti 1  
configurazione 19  
differenze 14  
funzioni 13  
gestione errore 82  
impostazione della seconda istanza 77  
installazione 15, 16  
migrazione 87  
panoramica 1  
piattaforme 15  
prerequisiti 16  
regolazione delle prestazioni 78  
Risoluzione dei problemi 81  
tipi di errore 82  
verifica dell'installazione 17  
aggiornamento 87  
5.3.0 a 5.6.0 87  
modifiche architettura 88  
scenari 87  
ambiente dell'adattatore 15  
architettura 2  
archivio evento 8  
specifica 59  
archivio locale  
configurazione 60  
nomi file 59  
archivio remoto  
specifica 69  
Attività di configurazione 58  
Attributi ObjectEventID  
specifica valori 77  
attributo EndBODelimiter  
utilizzo 66

### C

componenti adattatore 1  
Connector Configurator  
ambiente globalizzato 38  
avvio 20  
modalità silente 20  
panoramica 19  
System Manager, in esecuzione da 21  
Unix 20

### D

dati dipendenti dalla locale 16  
elaborazione 14  
denominazione file dinamica 61  
denominazione file statica 61

directory locale  
specifica 69

### E

elaborazione eventi  
operazioni di configurazione 72  
elaborazione eventi remota 68  
elaborazione richiesta 10  
operazioni di configurazione 74  
specifica 61  
elaborazione richiesta remota 72  
EndBODelimiter  
utilizzo dei caratteri non stampabili 66  
errori di formattazione  
recupero 85  
errori di sottoscrizione  
recupero 85

### F

file di configurazione  
completamento 26  
creazione dalla maschera 24  
creazione nuovo 24  
impostazione proprietà 27  
modifica 37  
salvataggio 37  
file di esempio  
generazione 79  
file di evento multipli  
specifica 65  
File di output  
configurazione 63  
specifica nome 56  
file log di evento 82  
file system FTP remoto  
specifica 68  
FTP sicuro  
configurazione 75

### G

gestori dati 2  
Delimitato 2  
elaborazione 12  
FixedWidth 2  
modifica 76  
NameValue 2  
oggetti business supportate 86  
specifica 76

### I

installazione 15, 16  
panoramica attività 15  
Struttura dei file di Windows 17, 18  
verifica 17

installazione (*Continua*)  
verifica su Linux 17, 18  
verifica su OS/400 e i5/OS 18  
verifica su Windows 17

invio errori  
recupero 86  
istanze del connettore  
creazione più 41

### M

maschera proprietà specifica di un connettore  
creazione 21  
Messaggi di errore  
registrazione 81  
metaoggetti 3  
aggiorna modifiche 87  
attributi secondari dinamici 5  
convenzioni di denominazione 45  
creazione di oggetti business specifici 76  
creazione personalizzati 47  
definiti 45  
problemi di denominazione 81  
struttura 46  
struttura gerarchica 47  
utilizzo 45  
utilizzo elementi secondari dinamici 4  
Metodo di analisi EndBODelimiter  
utilizzo 65  
Metodo di analisi FixedBOSize  
utilizzo 68  
migrazione 87  
MO\_JTextConnector\_Default  
attributi 48

### N

notifica evento 6  
specifica 58  
nuova maschera  
creazione 21

### O

Oggetti business 3  
aggiunta supportato 43  
elaborazione verbi di richiesta 13  
recupero errore delimitatore 85  
oggetti business di esempio  
generazione 79

### P

piattaforme supportate 15  
piattaforme supportate dell'adattatore 15

- più directory di evento
  - specifica 65
- poll remoto
  - specifica 70
- polling
  - funzionamento di configurazione 59
  - oggetti business specifici 65
- prerequisiti 16
- prestazione
  - regolazione 78
- Proprietà del connettore 30
  - ArchivingEnabled 30
  - EventLog 30
  - EventRecovery 30
  - FTPPollFrequency 30, 31
  - GenerateTemplate 30, 31
  - NoPoll 32
  - OutputLog 30, 31
  - PollQuantity 30, 32
  - SortFilesOnTimestamp 30, 32
- proprietà del connettore standard
  - impostazione 28
- proprietà di configurazione specifiche del connettore
  - impostazione 29
- proprietà di configurazione standard 91

## R

- recupero errori 83
- Risoluzione dei problemi 81

## S

- sito remoto
  - elaborazione eventi 71

## T

- tipi di errore 82
- trasferimento FTP
  - note di configurazione 74
- trigger dell'evento
  - problemi 81

## U

- URL FTP
  - specifica 68

---

## 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 Europe  
Schoenaicher Str. 220  
D - 7030 Boeblingen  
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 questa pubblicazione viene visualizzata in formato elettronico, 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 di scrivere il software applicativo che ottiene i servizi degli strumenti di questo programma. Tuttavia, queste informazioni possono contenere anche le informazioni sull'ottimizzazione, modifica e diagnosi. 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 altri paesi: 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 e il logo Windows sono marchi di Microsoft Corporation negli Stati Uniti e/o in altri paesi. MMX, Pentium e ProShare sono marchi di Intel Corporation negli Stati Uniti e/o in altri paesi. Java e tutti i marchi basati su Java sono marchi di Sun Microsystems, Inc. negli Stati Uniti e/o in altri paesi. Linux è un marchio di Linus Torvalds negli Stati Uniti e/o in altri paesi. Altri nomi di società, di servizi e prodotti possono essere marchi di altre società.



WebSphere Business Integration Server Express, Versione 4.4, e WebSphere Business Integration Server Express Plus, Versione 4.4







Stampato in Italia