

IBM® WebSphere® Commerce



Catalog Manager - Guida per l'utente

Versione 54

IBM® WebSphere® Commerce



Catalog Manager - Guida per l'utente

Versione 54

Nota

Prima di utilizzare queste informazioni e il prodotto supportato, consultare il paragrafo "Informazioni particolari" a pagina 127.

Prima edizione, prima revisione (Aprile 2002)

Questa edizione si applica al seguente prodotto:

- IBM WebSphere Commerce, Versione 5.4 (Numero programma 5724-A18)

e tutti i successivi rilasci e tutte le modifiche dei prodotti elencati, se non diversamente specificato nelle nuove edizioni. Accertarsi di utilizzare l'edizione corretta per il livello del prodotto.

Ordinare le pubblicazioni tramite il rappresentante IBM o presso la filiale IBM di zona. Le pubblicazioni non sono disponibili all'indirizzo riportato di seguito.

Come ultima pagina del manuale è stato predisposto un foglio riservato ai commenti del lettore. Se il foglio è stato rimosso, i commenti possono essere inviati al seguente indirizzo:

SELFIN S.p.A.
Translation Assurance
Via F. Giordani, 7
80122 Napoli
ITALY

Tutti i commenti e i suggerimenti inviati potranno essere utilizzati liberamente dall'IBM o dalla Selfin e diventeranno esclusiva delle stesse.

© Copyright International Business Machines Corporation 2002. Tutti i diritti riservati.

Indice

Prima di iniziare.	vii
Convenzioni utilizzate in questo manuale	vii
A chi si rivolge questo manuale	viii
Dove reperire informazioni	viii

Parte 1. Panoramica su Catalog Manager 1

Parte 2. Trasformazione, caricamento ed estrazione di dati 3

Capitolo 1. Introduzione	5
Programmi di utilità.	8
Strumenti di gestione	9

Capitolo 2. Trasformazione del testo . . . 11

Avvio dello strumento Text Transformation.	12
Utilizzo della Vista per la modifica schema testo	13
Creazione di un file schema	13
Apertura di un file schema	13
Salvataggio di un file schema	13
Modifica di un file schema	13
Aggiunta di un elemento	14
Rimozione di un elemento	14
Sostituzione di un elemento	14
Spostamento di un elemento una riga su	14
Spostamento di un elemento una riga giù	14
Aggiunta di un attributo	14
Rimozione di un attributo	14
Sostituzione di un attributo	14
Spostamento di un attributo una riga su.	14
Spostamento di un attributo una riga giù	14
Modifica della struttura di un file schema	15
Preparazione di un file schema per convertire i dati da un formato dati XML a un formato variabile delimitato da caratteri.	15
Utilizzo della Vista per la modifica comando trasformazione	16
Creazione di un comando	16
Rimozione di un comando	16
Modifica o sostituzione di un comando	16
Spostamento di un comando una riga su	16
Spostamento di un comando una riga giù	17
Eliminazione di un comando	17
Utilizzo della Vista processo trasformazione	17

Capitolo 3. Trasformazione di dati XML 19

Avvio di XSL Editor	20
Utilizzo delle aree di creazione delle regole di associazione	21
Creazione di un'area di creazione delle regole di associazione	21

Modifica di un'area di creazione delle regole di associazione	21
Eliminazione di un'area di creazione delle regole di associazione	21
Utilizzo di XSL Editor	22
Creazione di una regola di associazione	22
Associazione elemento-elemento	22
Associazione attributo-attributo.	22
Creazione di un'espressione di associazione personalizzata	22
Eliminazione di una regola di associazione	23
Elaborazione di una trasformazione XML	23
Personalizzazione della tabella delle regole di associazione	23
Visualizzazione di una regola XSL o di un'espressione valore completa.	24

Capitolo 4. Creazione di un DTD e di uno schema 25

Impostazione di DTD Generator	25
Creazione di un DTD	26
Creazione di uno schema e di un file XML dettagliato.	28

Capitolo 5. Risoluzione di identificativi 29

Impostazione di ID Resolver.	29
Impostazione delle modalità di gestione dei formati data/ora di ID Resolver	30
Impostazione delle modalità di gestione della memoria di ID Resolver	30
Impostazione delle modalità di gestione dei driver del database di ID Resolver.	31
Determinazione delle modalità di elaborazione dati	31
Selezione del metodo load	31
Selezione del metodo update	32
Selezione del metodo mixed.	32
Utilizzo delle tecniche di risoluzione di ID Resolver	33
Specificazione di un file properties con ID Resolver.	33
Utilizzo di un file properties per generare identificativi	34
Utilizzo di un file properties con chiavi composte	35
Utilizzo di un file properties con chiavi principali a cascata	36
Utilizzo della risoluzione alias interno	37
Esempio parziale di utilizzo della risoluzione ID alias interno	38
Utilizzo della risoluzione indice univoco.	39
Esempio parziale di risoluzione indice univoco.	39
Caricamento dei dati nella tabella MEMBER	41
Creazione di una relazione esterna utilizzando la tabella REFKEYS	42
Risoluzione dei problemi	43

Capitolo 6. Caricamento di dati 45

Impostazione del pacchetto Loader	45
Come ignorare gli elementi del file di immissione	46
Inserimento di un valore NULL in una colonna	46
Caricamento di dati relativi alla data e ai formati data/ora	46
Caricamento dei formati data/ora correnti	47
Esempio di caricamento dei formati data/ora correnti	49
Esempi di aggiunta di durate ai formati data/ora correnti	49
Gestione delle code eventi	50
Esecuzione su diversi sistemi operativi e software di database	51
Sostituzione di un componente	54
Utilizzo della sincronizzazione dello spazio di ricerca Product Advisor	55
Personalizzazione della sincronizzazione dello spazio di ricerca di Product Advisor	58
Determinazione della modalità di elaborazione dei dati utilizzando il pacchetto Loader	59
Selezione del metodo load	59
Selezione del metodo import	60
Selezione del metodo import SQL	60
Altre considerazioni	60
Caricamento di documenti di grandi dimensioni	61
Consigli per la risoluzione dei problemi	62

Capitolo 7. Estrazione di dati 65

Creazione di un filtro di estrazione	65
Impostazione di Extractor	68

Capitolo 8. Utilizzo del programma di registrazione del pacchetto Loader . . . 69

Configurazione del collegamento nell'ambiente per sistemi Windows NT, Windows 2000, AIX, Linux e Solaris	69
Esempio di impostazione della variabile classpath	69
Esempio di specificazione della proprietà di sistema com.ibm.wca.logging.configFile	69
Personalizzazione della registrazione per il pacchetto Loader	70
Handler	71
Filtri	72
Formati	73
Esempio: WCALoggerConfig.xml e WCALogger.dtd	73
WCALoggerConfig.xml	73
WCALogger.dtd	74

Capitolo 9. Utilizzo del programma di segnalazione degli errori del pacchetto Loader 77

Capitolo 10. Configurazione degli script e dei comandi del pacchetto Loader . . 79

Parte 3. Utilizzo di Web editor 81

Capitolo 11. Impostazione di Web editor 83

Configurazione di Web editor	84
Modifica del file webeditor.properties	84
Modifica dell'ubicazione dei file temporanei	85
Creazione di un file di descrizione moduli XML mediante DTD Generator.	86
Personalizzazione della descrizione dei moduli XML	92
Modifica dei nomi dei moduli	92
Modifica della descrizione di un campo	94
Aggiunta di un menu a discesa.	94
Aggiunta di un aiuto sui campi.	95
Personalizzazione dei risultati della ricerca e dell'elenco delle sessioni di lavoro.	96
Modifica del file weProcessList.	98
Modifica del file webeditor.xml	100

Capitolo 12. Gestione dei cataloghi 101

Aggiunta di un record a una tabella mediante Web editor	101
Modifica di un record in a una tabella mediante Web editor	102
Eliminazione di un record da una tabella mediante Web editor	104

Parte 4. Riferimento ai comandi 107

Capitolo 13. Comando DTD Generate 109

Comando DTD Generate per sistemi Windows, AIX, Linux, e Solaris	109
Comando DTD Generate per sistemi iSeries	110

Capitolo 14. Comando Extract 113

Comando Extract per sistemi Windows, AIX, Linux, e Solaris	113
Comando Extract per sistemi iSeries	114

Capitolo 15. Comando ID Resolve . . 115

Comando ID Resolve per sistemi Windows, AIX, Linux, e Solaris	115
Comando ID Resolve per sistemi iSeries	116

Capitolo 16. Comando Load 119

Comando Load per sistemi Windows, AIX, Linux, e Solaris	119
Comando Load per sistemi iSeries	121

Capitolo 17. Comando Text Transform 123

Comando Text Transform per sistemi Windows, AIX, Linux e Solaris	123
Comando Text Transform per sistemi iSeries	124

Capitolo 18. Comando XML Transform 125

Comando XML Transform per sistemi Windows, AIX, Linux, e Solaris	125
Comando XML Transform per sistemi iSeries	125

Informazioni particolari 127

Marchi e marchi di servizio. 129

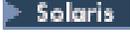
Prima di iniziare

Il manuale *IBM WebSphere Commerce Catalog Manager - Guida per l'utente* fornisce informazioni su WebSphere Commerce Catalog Manager. In particolare, contiene dettagli relativi ai seguenti argomenti:

- Trasformazione, caricamento ed estrazione dei dati mediante gli strumenti e i programmi di utilità di Catalog Manager
- Utilizzo di Catalog Manager Web editor per la gestione dei dati di catalogo
- Comandi di Catalog Manager

Convenzioni utilizzate in questo manuale

In questo manuale vengono utilizzate le seguenti convenzioni:

Grassetto	indica i comandi o i controlli della GUI (Graphical User Interface) come i nomi dei campi, i pulsanti o le scelte di menu.
Monospazio	indica gli esempi di testo da immettere esattamente come sono mostrati, i nomi dei file ed i percorsi delle directory.
<i>Corsivo</i>	utilizzato per evidenziare delle parti e per indicare le variabili cui l'utente attribuisce dei valori.
	indica informazioni specifiche di WebSphere Commerce per Windows®.
	indica informazioni specifiche di WebSphere Commerce per Windows NT®.
	indica informazioni specifiche di WebSphere Commerce per Windows 2000®.
	indica informazioni specifiche di WebSphere Commerce per AIX®.
	indica informazioni specifiche del software per l'ambiente operativo WebSphere Commerce per Solaris™.
	indica le informazioni specifiche di WebSphere Commerce per Linux.
	indica informazioni specifiche di WebSphere Commerce per IBM @server™ iSeries™ (in genere chiamato AS/400®).
	indica informazioni specifiche di DB2® Universal Database.
	indica informazioni specifiche del database Oracle®.
	indica le informazioni specifiche di WebSphere Commerce Business Edition.
	indica le informazioni specifiche di WebSphere Commerce Professional Edition.

A chi si rivolge questo manuale

Questo manuale si rivolge agli sviluppatori di siti, ai responsabili e agli addetti ai lavori che hanno la necessità di sapere come utilizzare WebSphere Commerce Catalog Manager e ad altri utenti aziendali che hanno la necessità di conoscerne le funzioni.

In particolare, si rivolge agli sviluppatori del negozio Websphere Commerce che hanno la necessità di sapere come utilizzare le varie funzioni di WebSphere Commerce Catalog Manager.

Gli sviluppatori del negozio WebSphere Commerce e i responsabili del sito che eseguono estensioni programmatiche devono inoltre avere familiarità con le seguenti aree:

- Tecnologia di database
- Architettura del componente Enterprise JavaBeans™
- Linguaggio HTML (Hypertext Markup Language)
- Java™
- Tecnologia JavaServer Pages™
- VisualAge® per Java™, Enterprise Edition, Versione 3.5 o successiva
- Linguaggio XML (Extensible Markup Language)

I responsabili di categoria e del prodotto devono inoltre leggere questo manuale per comunicare i requisiti di cataloghi, prodotti e articoli agli sviluppatori del negozio. I responsabili di categoria e del prodotto devono inoltre sapere come utilizzare Web editor per aggiornare il catalogo del negozio.

Dove reperire informazioni

Questa guida per l'utente è disponibile nel formato Adobe PDF (Portable Document Format) al sito Web di WebSphere Commerce. Verificare il sito Web di WebSphere Commerce Technical Library relativo alle versioni più recenti di questo documento e ad altre informazioni su WebSphere Commerce:

-  http://www.ibm.com/software/webservers/commerce/wc_be/lit-tech-general.html
-  http://www.ibm.com/software/webservers/commerce/wc_pe/lit-tech-general.html

Parte 1. Panoramica su Catalog Manager

WebSphere Commerce Catalog Manager fornisce un kit di strumenti generico contenente varie funzioni che possono essere unite insieme nella sequenza richiesta per risolvere problemi particolari relativi alla gestione dei cataloghi. Il prodotto è abbastanza flessibile da gestire le personalizzazioni effettuate allo schema di WebSphere Commerce.

Catalog Manager consente di unire le informazioni provenienti da fonti diverse in un sistema WebSphere Commerce consolidato e di creare una nuova associazione tra tutti i dati diversi in un formato standard di definizione del catalogo e del prodotto utilizzando file XML come strumenti standard per la gestione delle informazioni.

Catalog Manager consente di eseguire le attività di seguito riportate:

- Importazione di dati da fonti di immissioni diverse nel formato di file ASCII e XML in WebSphere Commerce
- Trasformazione di dati dal formato ASCII a XML e viceversa
- Creazione di una associazione di dati da un formato XML ad un altro
- Aggregazione di dati da più flussi di immissione in un unico database
- Creazione/modifica/eliminazione di dati mediante l'interfaccia di un browser Web

Catalog Manager comprende quanto segue:

- Pacchetto Loader di Catalog Manager

Questo pacchetto è composto principalmente da utilità di comandi per la preparazione e il caricamento dei dati nel database di WebSphere Commerce. E' possibile utilizzare il pacchetto Loader per caricare una grossa quantità di dati e per aggiornare i dati nel database di WebSphere Commerce.

Il pacchetto Loader consente di eseguire le attività di seguito riportate:

- Estrazione di dati da un database come un documento XML
- Conversione di dati XML in formati XML alternativi
- Conversione dei dati da un formato variabile delimitato da caratteri a un formato dati XML

- Strumenti di gestione di Catalog Manager

Catalog Manager include inoltre i due strumenti di seguito riportati con un'interfaccia utente per la gestione delle relative funzioni:

- Strumento Text Transformation
- XSL Editor

- Catalog Manager Web editor

Web editor consente di creare, eliminare e apportare modifiche ai dati di catalogo mediante un browser Web.

Parte 2. Trasformazione, caricamento ed estrazione di dati

Capitolo 1. Introduzione

Il pacchetto Loader di Catalog Manager è composto principalmente da programmi di utilità per la preparazione e il caricamento dei dati nel database di WebSphere Commerce. E' possibile utilizzare il pacchetto Loader per caricare una grossa quantità di dati e per aggiornare i dati nel database di WebSphere Commerce. Il pacchetto Loader consente di eseguire le attività di seguito riportate:

- Estrazione di dati da un database come un documento XML
- Conversione di dati XML in formati XML alternativi
- Conversione dei dati da un formato variabile delimitato da caratteri a un formato dati XML

Catalog Manager include inoltre i due strumenti di seguito riportati con un'interfaccia utente per la gestione delle relative funzioni:

- Strumento Text Transformation
- XSL Editor

Programmi di utilità

Il pacchetto Loader di WebSphere Commerce include programmi di utilità per la preparazione e il caricamento dei dati nel database di WebSphere Commerce. E' possibile utilizzare questi programmi di utilità per caricare quantità di dati di diverse dimensioni per aggiornare i dati nel database di WebSphere Commerce.

Il processo di caricamento è composto dai passi necessari per spostare i dati nel database di WebSphere Commerce:

1. Creazione di un DTD mediante DTD Generator
2. Risoluzione degli identificativi nei file di immissione utilizzando ID Resolver
3. Caricamento dei dati utilizzando il pacchetto Loader

Il pacchetto Loader include anche programmi di utilità per estrarre i dati da un database sotto forma di documento XML e per trasformare i dati XML in formati XML alternativi.

Il pacchetto Loader è composto dai seguenti programmi di utilità:

- **Text Transformer**

Text Transformer converte i dati da un formato variabile delimitato da caratteri a un formato dati XML.

Per ulteriori informazioni, fare riferimento al Capitolo 2, "Trasformazione del testo" a pagina 11.

- **XML Transformer**

XML Transformer modifica, aggrega e crea una associazione tra i dati di un documento XML in formati XML alternativi per l'utilizzo da parte di altri utenti o di altri sistemi.

Per ulteriori informazioni, fare riferimento al Capitolo 3, "Trasformazione di dati XML" a pagina 19.

- **DTD Generator**

Il DTD specifica gli elementi strutturali e le definizioni di markup che possono essere utilizzate nel documento dati XML. Ad esempio, un DTD può elencare gli elementi da utilizzare in un documento e specificare gli attributi che ciascun elemento può assumere.

DTD Generator è un programma di utilità che crea un DTD per il pacchetto Loader da utilizzare, basato sullo schema del database. Questo DTD descrive le tabelle e le colonne in cui il programma Loader importa i dati. DTD Generator può anche creare uno schema e un documento XML dettagliato che può essere utilizzato con Catalog Manager Web editor.

DTD Generator genera un DTD basato sul database di destinazione al quale si devono conformare i dati. Questo DTD verrà utilizzato durante il processo di caricamento. Il programma DTD Generator può essere eseguito una sola volta. Per ulteriori informazioni, fare riferimento al Capitolo 4, "Creazione di un DTD e di uno schema" a pagina 25.

- **ID Resolver**

ID Resolver è un programma di utilità che genera identificativi per gli elementi XML che li richiedono. Se il contenuto XML già fornisce identificativi, non è necessario eseguire ID Resolver.

ID Resolver aggiorna un insieme di elementi XML ed i relativi identificativi. Questo passo è necessario poiché i file XML del pacchetto Loader vengono associati direttamente allo schema del database di destinazione. Per lo stesso motivo, i file devono disporre di identificativi.

ID Resolver include un programma di segnalazione degli errori che crea un documento di eccezione in caso di errore.

Per ulteriori informazioni, fare riferimento al Capitolo 5, "Risoluzione di identificativi" a pagina 29.

- **Loader**

Il programma di utilità Loader utilizza XML validi e in formato corretto come input per caricare i dati nel database. Gli elementi del documento XML vengono associati ai nomi della tabella nel database e gli attributi dell'elemento vengono associati alle colonne. Il programma Loader è il metodo più comune di caricamento di dati nel sistema.

Loader consente gli aggiornamenti delle colonne di una tabella. Consente inoltre di eliminare i dati da un database.

Il seguente esempio mostra un elenco di dati XML validi e ben strutturati nel programma Loader:

```
<ADDRBOOK
  ADDRBOOK_ID="11801"
  MEMBER_ID="100"
  DISPLAYNAME="Friends"
  DESCRIPTION="All my friends"
  TYPE="P"
/>
```

Nell'esempio sopra riportato, ADDRBOOK è il nome della tabella e le colonne da aggiornare vengono indicate dagli attributi dell'elemento ADDRBOOK.

Il programma di utilità Loader include le seguenti funzioni:

- **Programma di segnalazione degli errori**

Il programma Loader include un programma di segnalazione degli errori che crea un documento di eccezione in caso di errore.

- **Sincronizzazione dello spazio di ricerca di Product Advisor**

Lo spazio di ricerca di Product Advisor viene creato estraendo informazioni dal catalogo WebSphere Commerce e rappresentando tali informazioni in un formato adatto alla ricerca. Gli spazi di ricerca ed il catalogo diventano desincronizzati quando il catalogo viene aggiornato. Per evitare un ritardo nella sincronizzazione degli spazi di ricerca con il catalogo, è possibile abilitare la funzione di sincronizzazione dello spazio di ricerca Product Advisor di Loader.

Se la sincronizzazione dello spazio di ricerca di Product Advisor viene abilitata, il programma Loader pianifica comandi dello spazio di ricerca appropriati per aggiungere le informazioni sul comando nella tabella del processo del programma di pianificazione SCHCONFIG e nella tabella dell'istanza del processo del programma di pianificazione SCHSTATUS, come indicato nella seguente tabella.

Tabella	Modo	Comando spazio di ricerca	Condizione
CATENTRY	Aggiorna	UpdateSearchSpaces	Aggiorna l'indicatore di pubblicazione
	Elimina	RemoveProductFromAllSearchSpaces	Elimina il prodotto
CATENTDESC	Aggiungi	UpdateSearchSpaces	Aggiunge la descrizione/lingua
	Aggiorna	UpdateSearchSpaces	Aggiorna l'indicatore di pubblicazione Aggiorna i dati di descrizione
	Elimina	UpdateSearchSpaces	Rimuove la descrizione/lingua
LISTPRICE	Aggiungi	UpdateSearchSpaces	Aggiunge il listino prezzi
	Aggiorna	UpdateSearchSpaces	Aggiorna il listino prezzi
	Elimina	UpdateSearchSpaces	Elimina il listino prezzi
ATTRVALUE	Aggiungi	UpdateSearchSpaces	Aggiunge il valore definito dall'utente
	Aggiorna	UpdateSearchSpaces	Aggiorna il valore definito dall'utente
	Elimina	UpdateSearchSpaces	Elimina il valore definito dall'utente
CATENTATTR	Aggiungi	UpdateSearchSpaces	Aggiunge l'attributo
	Aggiorna	UpdateSearchSpaces	Aggiorna l'attributo
	Elimina	UpdateSearchSpaces	Elimina l'attributo
CATGPENREL	Aggiungi	AddProductsToSearchSpace	Aggiunge il prodotto alla categoria
	Elimina	RemoveProductsFromSearchSpace	Rimuove il prodotto dalla categoria

Per ulteriori informazioni, fare riferimento al Capitolo 6, "Caricamento di dati" a pagina 45.

- **Extractor**

Il programma Extractor utilizza un'interrogazione rivolta a un database per estrarre sottoinsiemi selezionati di dati dal database in un documento XML. I dati da estrarre dal database vengono specificati utilizzando un documento XML di filtro per l'estrazione.

La funzione di Extractor è opposta a quella di Loader. Si utilizza Extractor per estrarre sottoinsiemi selezionati di dati dal database di WebSphere Commerce nel formato di file XML. E' possibile, ad esempio, estrarre dati su prodotti relativi ad una festività imminente, oppure è possibile estrarre informazioni da un database consolidato per riproporle ad altri sistemi.

Per ulteriori informazioni, fare riferimento al Capitolo 7, "Estrazione di dati" a pagina 65.

Il pacchetto Loader presenta anche la funzione di programma di registrazione. Ciascun programma di utilità del pacchetto Loader crea dei messaggi per indicare operazioni eseguite correttamente, operazioni non riuscite ed eventuali errori, oltre a fornire informazioni di traccia sul programma.

Strumenti di gestione

Catalog Manager include i due strumenti di seguito riportati per la gestione delle relative funzioni:

Strumento Text Transformation

Lo strumento Text Transformation consente al responsabile di preparare le informazioni necessarie per elaborare una conversione di dati da un formato variabile delimitato da caratteri a un formato dati XML.

XSL Editor

Lo strumento XML Transformer utilizza il linguaggio XSL (Extensible Stylesheet Language) per definire le regole per la trasformazione di un file XML in un altro file XML. La funzione di associazione di XSL Editor fornisce al responsabile un'interfaccia visiva con cui stabilire l'associazione tra un elemento presente in un DTD (Document Type Definition) di origine e un elemento presente in un DTD di destinazione.

Capitolo 2. Trasformazione del testo

Catalog Manager consente di convertire l'emissione di file ASCII di altri strumenti (ad esempio programmi per fogli di lavoro) in un formato dati XML che può essere inserito in un database di WebSphere Commerce.

Lo strumento Text Transformation consente di preparare le informazioni necessarie per elaborare una conversione di dati da un formato variabile delimitato da caratteri a un formato dati XML. Vengono fornite le seguenti visualizzazioni:

1. La Vista per la modifica schema testo consente di creare e modificare il file dello schema XML da utilizzare in un processo di conversione.
2. La Vista per la modifica comando trasformazione consente di creare e modificare i comandi effettivi utilizzati per eseguire il processo di trasformazione.
3. La Vista processo di trasformazione consente di avviare il processo di trasformazione.

Avvio dello strumento Text Transformation

Per avviare lo strumento Text Transformation, utilizzare lo script appropriato o il comando fornito nella directory di WebSphere Commerce:

-  `unità:\WebSphere\CommerceServer\bin\TextTrans.cmd`
-  `unità:\Program Files\WebSphere\CommerceServer\bin\TextTrans.cmd`
-  `/usr/WebSphere/CommerceServer/bin/texttrans.sh`
-   `/opt/WebSphere/CommerceServer/bin/texttrans.sh`

 In ambienti iSeries, i responsabili devono innanzitutto copiare i file necessari per l'esecuzione dello strumento Text Transformation sulle macchine Windows NT o Windows 2000. Di seguito è riportato un esempio della procedura da eseguire:

1. Creare una nuova directory sulla macchina Windows (ad esempio, `unità:\TextTrans`).
2. Creare le seguenti sottodirectory in `unità:\TextTrans`:
 - `\bin`
 - `\lib\loader`
 - `\wcsadmin`
3. Copiare i file dalla macchina iSeries in tali directory in base al grafico di seguito riportato:

Da		A
/QIBM/ProdData/WebCommerce		unità:\TextTrans
/bin		/bin
/TextTrans.cmd	----->	\TextTrans.cmd
/lib/loader		\lib\loader
/wcmxmlp.jar	----->	\wcmxmlp.jar
/wcmxslt.jar	----->	\wcmxslt.jar
/wcsadmin		\wcsadmin
/acsxml.ico	----->	\acsxml.ico
/swing.jar	----->	\swing.jar
/TextTransformerUI.zip	----->	\TextTransformerUI.zip
/TextTransform.cnt	----->	\TextTransform.cnt
/TextTransform.hlp	----->	\TextTransform.hlp
/TextTransform.reg	----->	\TextTransform.reg

4. Modificare il file `TextTrans.cmd` sulla macchina Windows inserendo il seguente testo:

```
set WCS_HOME=unità:\TextTrans
```

prima di:

```
set lib=%WCS_HOME%\lib\loader
```

5. Avviare lo strumento Text Transformation eseguendo `TextTrans.cmd` sulla macchina Windows.

Utilizzo della Vista per la modifica schema testo

Le procedure di seguito riportate si applicano alla creazione e alla modifica di un file di schema XML per la trasformazione dei dati da un formato variabile delimitato da caratteri a un formato XML.

Creazione di un file schema

Per creare un nuovo file schema, effettuare le seguenti operazioni:

1. Avviare lo strumento Text Transformation.
2. Selezionare **File > Nuovo**, o fare clic sull'icona Nuovo della barra degli strumenti.
3. Selezionare un percorso e immettere il nome del file di schema XML da creare.

Nota: Il nome file predefinito è "Default.xml."

4. Selezionare il **formato CSV**.

Nota: Non selezionare il formato **WebSphere Commerce Suite**. Questa opzione è utilizzata solo con le versioni precedenti di WebSphere Commerce.

5. Quindi, fare clic su **OK**.

E' ora possibile creare nuovi elementi e attributi seguendo i passi di seguito indicati.

Apertura di un file schema

Per aprire un file schema, attenersi alla seguente procedura:

1. Avviare lo strumento Text Transformation.
2. Selezionare **File > Apri**, o fare clic sull'icona Apri della barra degli strumenti.
3. Selezionare un file schema da aprire.
4. Quindi, fare clic su **OK**.

E' ora possibile modificare gli elementi e gli attributi seguendo i passi descritti di seguito.

Salvataggio di un file schema

Per salvare un file schema, attenersi alla seguente procedura:

1. Avviare lo strumento Text Transformation.
2. Per salvare le modifiche apportate allo schema, selezionare **File > Salva** oppure fare clic sull'icona Salva della barra degli strumenti.
3. Per salvare una copia dello schema con un nuovo nome, attenersi alla seguente procedura:
 - a. Selezionare **File > Salva con nome**.
 - b. Selezionare un percorso e immettere il nome del file XML da creare.
 - c. Quindi, fare clic su **OK**.

Modifica di un file schema

Per modificare un file schema, aprire il file schema come descritto sopra e seguire queste procedure.

Aggiunta di un elemento

Per aggiungere un elemento, seguire questa procedura:

1. Immettere il nome di un nuovo elemento nel campo Elenco elementi.
2. Fare clic sull'icona Aggiungi riga.

Rimozione di un elemento

Per rimuovere un elemento, seguire questa procedura:

1. Selezionare il nome di un elemento.
2. Fare clic sull'icona Rimuovi riga.

Sostituzione di un elemento

Per sostituire un elemento, seguire questa procedura:

1. Immettere il nome di un nuovo elemento nel campo Elenco elementi.
2. Selezionare il nome dell'elemento da sostituire.
3. Fare clic sull'icona Sostituisci riga.

Spostamento di un elemento una riga su

Per spostare un elemento una riga su, seguire questa procedura:

1. Selezionare il nome dell'elemento da spostare su.
2. Fare clic sull'icona per lo spostamento della riga su.

Spostamento di un elemento una riga giù

Per spostare un elemento una riga giù, seguire questa procedura:

1. Selezionare il nome dell'elemento da spostare giù.
2. Fare clic sull'icona per lo spostamento della riga giù.

Aggiunta di un attributo

Per aggiungere un attributo, seguire questa procedura:

1. Immettere il nome di un nuovo attributo nel campo Elenco attributi.
2. Fare clic sull'icona Aggiungi riga.

Rimozione di un attributo

Per rimuovere un attributo, effettuare le seguenti operazioni:

1. Selezionare il nome dell'attributo.
2. Fare clic sull'icona Rimuovi riga.

Sostituzione di un attributo

Per sostituire un attributo, effettuare le seguenti operazioni:

1. Immettere il nome di un nuovo attributo nel campo Elenco attributi.
2. Selezionare il nome dell'attributo da sostituire.
3. Fare clic sull'icona Sostituisci riga.

Spostamento di un attributo una riga su

Per spostare un attributo una riga su, seguire questa procedura:

1. Selezionare il nome dell'attributo da spostare su.
2. Fare clic sull'icona per lo spostamento della riga su.

Spostamento di un attributo una riga giù

Per spostare un attributo una riga giù, seguire questa procedura:

1. Selezionare il nome dell'attributo da spostare giù.
2. Fare clic sull'icona per lo spostamento della riga giù.

Modifica della struttura di un file schema

La vista Struttura file nel pannello inferiore della Vista per la modifica schema testo descrive il layout di un file variabile delimitato da caratteri. Nella struttura file prevista sono necessari i campi di seguito riportati:

Separatore Campo:

Specifica il delimitatore che separa i valori di attributi. Il delimitatore predefinito è una virgola (",").

Separatore Record:

Specifica il delimitatore che separa i record di dati. Il valore predefinito è "
" (equivalente ad un riferimento entità pari a \r\n.)

Delimitatore Stringa:

Specifica il delimitatore che indica i limiti iniziale e finale della stringa. Il valore predefinito è un singolo apice (" ' ").

Intestazione inclusa:

Un valore booleano, specificato come "true" se esiste una riga di intestazione nel file di dati del testo e "false" se non esiste alcuna riga di intestazione nel file di dati del testo. Se esiste una riga di intestazione, deve essere conforme alle regole XML per il nome di una tag, in quanto, in questo caso, l'intestazione viene utilizzata come nome tag XML. Il valore predefinito è "false."

Numero di righe intestazione:

Specifica il numero di righe di intestazione presenti nel file di dati del testo. Il valore predefinito è zero ("0").

Preparazione di un file schema per convertire i dati da un formato dati XML a un formato variabile delimitato da caratteri

Se si utilizza un file schema XML creato con la Vista per la modifica schema testo per convertire i dati da un formato XML a un formato variabile delimitato da caratteri, è necessario utilizzare un editor di testo per modificare la specificazione del tipo di dati nel file schema da "CSV Format" a "XML Format" prima di eseguire la conversione.

Utilizzo della Vista per la modifica comando trasformazione

Utilizzando la Vista per la modifica comando trasformazione, è possibile creare un nuovo file di comandi, aprire un file di comandi esistente oppure salvare le modifiche apportate a un file di comandi. Il file di comando predefinito è "Manifest.txt."

E' possibile creare un nuovo comando, rimuovere un comando, sostituire un comando con informazioni modificate oppure modificare l'ordine di un comando.

Nota: Il file di comando viene salvato automaticamente ogni volta che viene aggiornata la tabella dei comandi.

Creazione di un comando

Per creare un nuovo comando, effettuare le seguenti operazioni:

1. Specificare il file di origine, un file in formato separato da caratteri (con estensione .csv) o XML (con estensione .xml).
2. Specificare il file schema XML da utilizzare nella conversione.
3. Specificare il nome del file di emissione da creare o modificare durante il processo di conversione (cioè il file in cui vengono memorizzati i nuovi dati), un file in formato XML (con estensione .xml) o variabile separato da caratteri (con estensione .csv).
4. Specificare una modalità di comando.
Selezionare **Crea** se il file di emissione deve essere creato o **Aggiungi** se i dati di emissione devono essere aggiunti ad un file di dati esistente.
5. Fare clic sull'icona Aggiungi riga.

Rimozione di un comando

Per rimuovere un comando, effettuare le seguenti operazioni:

1. Selezionare un comando.
2. Fare clic sull'icona Rimuovi riga.

Modifica o sostituzione di un comando

Per modificare o sostituire un comando, effettuare le seguenti operazioni:

1. Fare clic sull'icona Modifica comando.
Nei campi di immissione appropriati vengono inseriti i dati della riga.
2. Modificare il testo nei campi di immissione appropriati.
3. Fare clic sull'icona Sostituisci riga per aggiornare la riga.

Spostamento di un comando una riga su

Per spostare un comando una riga su, seguire questa procedura:

1. Selezionare un comando.
2. Fare clic sull'icona per lo spostamento della riga su.

Nota: La sequenza del processo di trasformazione viene modificata.

Spostamento di un comando una riga giù

Per spostare un comando una riga giù, seguire questa procedura:

1. Selezionare un comando.
2. Fare clic sull'icona per lo spostamento della riga giù.

Nota: La sequenza del processo di trasformazione viene modificata.

Eliminazione di un comando

Per eliminare un comando, effettuare le seguenti operazioni:

1. Selezionare un comando.
2. Fare clic sull'icona per Cancella campi di immissione.

Il testo viene eliminato dai campi del file di origine, del file schema e del file di emissione.

Utilizzo della Vista processo trasformazione

Per avviare il processo di trasformazione del testo, eseguire le operazioni di seguito riportate:

1. Immettere o selezionare il nome del file parametri nel campo File.
2. Fare clic su **Processo**.

Nell'area di emissione al di sotto del pulsante Processo vengono visualizzate informazioni che indicano lo stato del processo di trasformazione. E' possibile salvare le informazioni di emissione facendo clic sul pulsante Salva al di sotto dell'area del testo oppure è possibile eliminare tutte le informazioni sullo stato facendo clic sul pulsante Elimina.

Capitolo 3. Trasformazione di dati XML

Il linguaggio XSL (Extensible Stylesheet Language) fornisce quanto di seguito riportato:

1. Linguaggio per la formattazione di documenti XML
2. Linguaggio che descrive le modalità per trasformare un file XML in un altro file regolarmente strutturato

La funzione di trasformazione del linguaggio XSL può essere utilizzata per trasformare un file XML in un altro file XML conforme ad uno schema XML o ad un DTD diverso.

Per trasformare un file XML in un formato diverso, è necessario specificare le regole per la trasformazione utilizzando un file di regole XSL di trasformazione.

Nel seguente esempio vengono trasformati i dati presenti nel file MemberSubsystemExtracted.xml che utilizza MemberSubsystem.xsl come file di regole XSL di trasformazione con il giapponese come lingua nazionale:



```
java com.ibm.wca.XMLTransformer.XMLTransformer -infile MemberSubsystemExtracted.xml  
-transform MemberSubsystem.xsl -outfile TransMbrStr.xml -param 'language="-10''
```

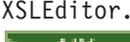
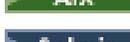


```
QWEECOMM/TRNWCSXML INFILE(MemberSubsystemExtracted.xml)  
TRANSFORM(MemberSubsystem.xsl) INSTROOT(/QIBM/UserData/WebCommerce/instances/my_inst)  
OUTFILE(TransMbrStr.xml) PARAM('language=-10')
```

Lo strumento XML Transformer utilizza il linguaggio XSL per definire le regole per la trasformazione di un file XML in un altro file XML. La funzione di associazione di XSL Editor fornisce un'interfaccia visiva con cui è possibile stabilire l'associazione tra un elemento presente in un DTD (Document Type Definition) di origine e un elemento presente in un DTD di destinazione. Dati due DTD, è possibile sviluppare regole XSL per stabilire il modo in cui un file XML conforme al primo DTD (origine) venga trasformato in un file conforme al secondo DTD (destinazione).

Avvio di XSL Editor

Per avviare XSL Editor, utilizzare lo script appropriato o il comando fornito nella directory di WebSphere Commerce:

-  `unità:\WebSphere\CommerceServer\bin\XSLEditor.cmd`
-  `unità:\Program Files\WebSphere\CommerceServer\bin\XSLEditor.cmd`
-  `/usr/WebSphere/CommerceServer/bin/xsleditor.sh`
-   `/opt/WebSphere/CommerceServer/bin/xsleditor.sh`

 In ambienti iSeries, i responsabili devono innanzitutto copiare i file necessari per l'esecuzione di XSL Editor sulle macchine Windows NT o Windows 2000. Di seguito è riportato un esempio della procedura da eseguire:

1. Creare una nuova directory sulla macchina Windows (ad esempio `unità:\XMLTrans`).
2. Creare le seguenti sottodirectory in `unità:\TextTrans`:
 - `\bin`
 - `\lib\loader`
 - `\wcsadmin`
3. Copiare i file dalla macchina iSeries in tali directory in base al grafico di seguito riportato:

Da		A
/QIBM/ProdData/WebCommerce		unità:\XMLTrans
/bin		/bin
/XSLEditor.cmd	----->	XSLEditor.cmd
/lib/loader		lib/loader
/wcmxmlp.jar	----->	wcmxmlp.jar
/wcmxslt.jar	----->	wcmxslt.jar
/wcsadmin		wcsadmin
/config.dtd	----->	acsxml.ico
/swing.jar	----->	swing.jar
/XML_transform.ico	----->	XML_transform.ico
/XMLTransformerUI.zip	----->	XMLTransformerUI.zip
/XMLTransformWP.xml	----->	XMLTransformWP.xml
/XMLTransformWP.dtd	----->	XMLTransformWP.dtd
/XMLRuleConfig.xml	----->	XMLRuleConfig.xml
/XMLTransform.cnt	----->	XMLTransform.cnt
/XMLTransform.hlp	----->	XMLTransform.hlp
/XMLTransform.reg	----->	XMLTransform.reg

4. Modificare il file `XSLEditor.cmd` sulla macchina Windows inserendo il seguente testo:

```
set WCS_HOME=unità:\XMLTrans
```

prima di:

```
set lib=%WCS_HOME%\lib\loader
```
5. Avviare lo strumento XML Transformation eseguendo il comando `XSLEditor.cmd` sulla macchina Windows.

Utilizzo delle aree di creazione delle regole di associazione

Quando si avvia XSL Editor, viene visualizzata la finestra Area di creazione regole di associazione. Utilizzare questa finestra per gestire le aree di creazione delle regole di associazione.

Creazione di un'area di creazione delle regole di associazione

Per creare un'area di creazione delle regole di associazione, procedere come segue:

1. Avviare XSL Editor.
2. Dal menu a discesa, selezionare **[Nuovo]**.
3. Nel campo Nome, immettere il nome della nuova area di creazione delle regole di associazione.
4. Nel campo Descrizione, immettere una breve descrizione della nuova area di creazione delle regole di associazione.
5. Nel campo Schema di origine, inserire il nome di un file esistente oppure spostarsi su un file esistente da utilizzare come schema di origine.
6. Nel campo Schema destinazione, inserire il nome di un file esistente oppure spostarsi su un file esistente da utilizzare come schema di destinazione.
7. Nel campo File di regole XSL, immettere il nome del nuovo file di regole da creare.
E' possibile specificare un percorso completo. Se non si specifica alcun percorso, il file viene creato nella directory attualmente in uso.
8. Fare clic su **Apri** per creare e aprire la nuova area di creazione delle regole di associazione.

Modifica di un'area di creazione delle regole di associazione

Per modificare un'area di creazione delle regole di associazione, procedere come segue:

1. Avviare XSL Editor.
2. Selezionare dal menu a discesa l'area di creazione delle regole di associazione da modificare.
3. Fare clic su **Apri** per aprire l'area di creazione delle regole di associazione.
4. Aggiornare i campi che si desidera modificare.
5. Fare clic su **Salva** per salvare le modifiche.

Eliminazione di un'area di creazione delle regole di associazione

Per eliminare un'area di creazione delle regole di associazione, procedere come segue:

1. Avviare XSL Editor.
2. Selezionare dal menu a discesa l'area di creazione delle regole di associazione da eliminare.
3. Fare clic sul pulsante Rimuovi per rimuovere la voce.

Nota: La rimozione di un'area di creazione delle regole di associazione non comporta l'eliminazione dei file fisici dal disco.

Utilizzo di XSL Editor

Quando si apre un'area di creazione delle regole di associazione mediante XSL Editor, tale area viene visualizzata nella finestra principale di XSL Editor.

Nella finestra principale di XSL Editor, il pannello di sinistra mostra una vista gerarchica del DTD di origine denominato "Schema origine." Il pannello di destra mostra una vista gerarchica del DTD di destinazione denominato "Schema destinazione."

Creazione di una regola di associazione

Associazione elemento-elemento

Selezionare e trascinare un elemento dalla gerarchia di origine e rilasciarlo su un elemento della gerarchia di destinazione. Una regola XSL viene generata e visualizzata tra le regole di associazione poste nella parte inferiore della finestra.

Di seguito viene riportato un esempio di una regola XSL generata:

```
<xsl:template match="merchant">
  <xsl:element name="MERCHANT">
    </xsl:element>
  </xsl:template>
```

Nota: Viene automaticamente generata qualsiasi relazione precedente necessaria ma non esistente.

Associazione attributo-attributo

Selezionare e trascinare un attributo dalla gerarchia di origine e rilasciarlo su un attributo della gerarchia di destinazione. Una regola XSL viene generata e visualizzata tra le regole di associazione poste nella parte inferiore della finestra.

Di seguito viene riportato un esempio di una regola XSL generata:

```
<xsl:attribute name="MEADDR1">
  <xsl:apply-templates select="@mecmail1"/>
</xsl:attribute>
```

Nota: Viene automaticamente generata qualsiasi relazione precedente necessaria ma non esistente.

Creazione di un'espressione di associazione personalizzata

Per creare un'espressione di associazione personalizzata, selezionare un elemento o attributo dalla gerarchia di destinazione. Quindi, fare clic con il pulsante destro del mouse e selezionare il menu Crea espressione personalizzata. Nella finestra Crea espressione personalizzata viene visualizzato un elenco di modelli e espressioni di regole disponibili in due menu a discesa. Completare l'espressione personalizzata attenendosi alla seguente procedura:

1. Selezionare un modello a cui aggiungere l'espressione personalizzata.
2. Selezionare un'espressione di regole da creare (ad esempio **Espressione costante**).
3. Inserire un valore nella colonna Valore per ciascun parametro presente nella tabella e premere **Invio** per confermare il valore.
4. Fare clic su **OK** per completare la creazione oppure fare clic su **Annulla** per annullare la creazione senza creare alcuna regola.

La regola XSL generata è basata sull'espressione personalizzata definita nel file di configurazione delle regole (XSLRuleConfig.xml). E' possibile modificare il file di configurazione delle regole e aggiungere nuove regole, se necessario. Per rendere disponibile una regola nell'elenco di espressioni delle regole, impostare l'attributo relativo alla visibilità per quella regola su "true."

Eliminazione di una regola di associazione

Per eliminare una regola di associazione, procedere come segue:

1. Selezionare una regola dalla tabella delle regole di associazione.
2. Fare clic con il pulsante destro del mouse e selezionare **Elimina**.

La regola e tutti i relativi discendenti vengono eliminati.

Nota: Le regole di associazione aggiornate e le regole XSL generate verranno rese permanenti automaticamente.

Elaborazione di una trasformazione XML

Per elaborare una trasformazione XML, effettuare le seguenti operazioni:

1. Selezionare **Strumenti > Trasforma** per visualizzare la finestra relativa al processo di trasformazione.
2. Completare i campi obbligatori:
 - a. Nel campo File XML di immissione, inserire o selezionare il percorso e il nome del file di dati XML di origine.
 - b. Nel campo File regole XSL, inserire o selezionare il percorso e il nome del file delle regole di associazione da utilizzare per la trasformazione.
Se è aperta un'area di creazione delle regole di associazione, in questo campo è presente il percorso del file delle regole di associazione attualmente aperto nell'area di creazione delle regole di associazione.
 - c. Nel campo File XML di emissione, inserire o selezionare il percorso e il nome del nuovo file di dati XML da creare durante il processo di trasformazione.
3. Fare clic su **Avvia** per avviare il processo di trasformazione XML oppure fare clic su **Chiudi** per chiudere la finestra senza elaborare alcuna trasformazione.

Personalizzazione della tabella delle regole di associazione

Per personalizzare la tabella delle regole di associazione, procedere come segue:

1. Per nascondere una colonna della tabella, fare clic con il pulsante destro del mouse su una cella della tabella e selezionare **Nascondi colonna**.
2. Per mostrare una colonna nascosta della tabella, attenersi alla seguente procedura:
 - a. Fare clic con il pulsante destro del mouse su una cella della tabella.
 - b. Selezionare **Mostra colonne** per visualizzare l'elenco delle colonne nascoste.
 - c. Selezionare la colonna dall'elenco.

Nota: Per selezionare più colonne, premere e tenere premuto il tasto **Maiusc**, quindi fare clic sui nomi delle colonne.

- d. Fare clic su **OK** per visualizzare le colonne selezionate o su **Annulla** per annullare l'operazione.
3. Per visualizzare tutte le colonne nascoste della tabella, fare clic con il pulsante destro del mouse su una cella della tabella e selezionare **Mostra tutte le colonne**.

Tutte le colonne vengono visualizzate nell'ordine predefinito.

Visualizzazione di una regola XSL o di un'espressione valore completa

Se dalle colonne Espressione valore o Regola XSL si fa clic su una cella, viene visualizzata una finestra con il contenuto completo della regola per la riga selezionata.

Capitolo 4. Creazione di un DTD e di uno schema

DTD Generator può creare un DTD ed uno schema da utilizzare con il pacchetto Loader. DTD Generator utilizza un file di immissione contenente i nomi della tabella del database e genera un DTD o un DTD e uno schema con un file XML dettagliato che descrive il database, in base al modo in cui si richiama il comando DTD Generate.

Impostazione di DTD Generator

Il DTD (Document Type Definition) del pacchetto Loader crea corrispondenze direttamente con lo schema del database di WebSphere Commerce. Ciascuna tabella è un elemento e ciascuna colonna è un attributo.

Esempio: Creazione di una corrispondenza tra un DTD del pacchetto Loader ed uno schema di database

Istruzioni DDL per la tabella CATENTRY	DTD
CREATE TABLE	<!ELEMENT CATENTRY EMPTY>
"CATENTRY" (<!ATTLIST CATENTRY
"CATENTRY_ID" BIGINT NOT NULL	CATENTRY_ID CDATA #REQUIRED
"MEMBER_ID" BIGINT NOT NULL	MEMBER_ID CDATA #REQUIRED
"CATENTTYPE_ID" CHAR(16) NOT NULL	CATENTTYPE_ID CDATA #REQUIRED
"MARKFORDELETE" INTEGER NOT NULL	MARKFORDELETE CDATA #REQUIRED
"PARTNUMBER" VARCHAR(64) NOT NULL	PARTNUMBER CDATA #REQUIRED
"MFPARTNUMBER" VARCHAR(64)	MFPARTNUMBER CDATA #IMPLIED
"MFNAME" VARCHAR(64)	MFNAME CDATA #IMPLIED
"URL" VARCHAR(254)	URL CDATA #IMPLIED
"FIELD1" INTEGER	FIELD1 CDATA #IMPLIED
"FIELD2" INTEGER	FIELD2 CDATA #IMPLIED
"FIELD3" DECIMAL(20,5)	FIELD3 CDATA #IMPLIED
"FIELD4" VARCHAR(254)	FIELD4 CDATA #IMPLIED
"FIELD5" VARCHAR(254)	FIELD5 CDATA #IMPLIED
"LASTUPDATE" TIMESTAMP	LASTUPDATE CDATA #IMPLIED
"OID" VARCHAR(64)	OID CDATA #IMPLIED
"ONSPECIAL" INTEGER	ONSPECIAL CDATA #IMPLIED
"ONAUCTION" INTEGER	ONAUCTION CDATA #IMPLIED
"BUYABLE" INTEGER	BUYABLE CDATA #IMPLIED
"BASEITEM_ID" INTEGER	BASEITEM_ID CDATA #IMPLIED
"CLASSIFGRP_ID" INTEGER	CLASSIFGRP_ID CDATA #IMPLIED
"ITEMSPC_ID" INTEGER	ITEMSPC_ID CDATA #IMPLIED
"STATE" INTEGER	STATE CDATA "1"
);	>

E' possibile impostare la modalità di funzionamento di DTD Generator effettuando le seguenti operazioni:

1. Creare un nuovo file property di personalizzazione di DTD Generator.



Il file DB2ConnectionCustomizer.properties si trova nell'archivio DTDGenerator.zip. Estrarre questo file, rinominarlo lasciando l'estensione .properties e posizionarlo in una directory nel classpath. **Importante:** Non rimuovere o modificare il file DB2ConnectionCustomizer.properties esistente.

• ▶ 400

ISeries_GENWCSDTD_Customizer.properties si trova nella directory /QIBM/ProdData/WebCommerce/properties. Copiare questo file nella directory /rootist/xml, rinominare il nuovo file mantenendo l'estensione .properties, quindi apportare tutte le modifiche necessarie al nuovo file. **Importante:** Non rimuovere o modificare il file ISeries_GENWCSDTD_Customizer.properties originale.

2. Modificare i valori del driver del database nel nuovo file. Ad esempio:

```
DBVendorName = DB2
DBDriverName = COM.ibm.db2.jdbc.app.DB2Driver
DBURL = jdbc:db2:
```

in cui:

- DBVendorName viene utilizzato per selezionare il tipo di database.

Le opzioni sono le seguenti:

- DB2 Universal Database per iSeries (DB2/iSeries)
- DB2 per altri sistemi operativi (DB2)
- Database Oracle (Oracle)

- DBDriverName viene utilizzato per selezionare il driver JDBC.

Le opzioni sono le seguenti:

- DB2 Universal Database per iSeries (com.ibm.db2.jdbc.app.DB2Driver)
- DB2 per altri sistemi operativi (COM.ibm.db2.jdbc.app.DB2Driver)
- Database Oracle (oracle.jdbc.driver.OracleDriver)

- DBURL viene utilizzato per specificare l'URL per accedere al database.

Le opzioni sono le seguenti:

- DB2 Universal Database per iSeries (jdbc:db2://)
- DB2 per altri sistemi operativi (jdbc:db2:)
- Database Oracle (jdbc:oracle:oci8:@)

3. Specificare il nuovo nome file come valore del parametro di personalizzazione del comando DTD Generate.

Creazione di un DTD

Il file di immissione TableNames.txt contiene i seguenti nomi di tabelle di database, uno su ogni riga:

```
MEMBER
ADDRBOOK
ADDRESS
```

Di seguito viene riportato un esempio di come sia possibile richiamare DTD Generator:

- ▶ NT ▶ 2000 ▶ AIX ▶ Solaris ▶ Linux

```
java com.ibm.wca.DTDGenerator.GenerateDTD -dbname MALL
-dbuser db2inst1
-dbpwd db2ibm -outfile wc.dtd -infile TableNames.txt
```

• ▶ 400

```

QWBECCOMM/GENWCSDTD DATABASE(DATABASE_NAME) SCHEMA(MALL)
INSTROOT(/QIBM/UserData/WebCommerce/instances/mser)
PASSWD(mypassword) OUTFILE(wc.dtd) INFILE(TableName.txt)

```

Il file di emissione wc.dtd contiene quanto segue:

```

<!ELEMENT MALL (( MEMBER | ADDRBOOK | ADDRESS)*)>
<!ELEMENT MEMBER EMPTY>
<!ATTLIST MEMBER
  MEMBER_ID CDATA #REQUIRED
  TYPE      CDATA #REQUIRED
  STATE     CDATA #IMPLIED
>
<!ELEMENT ADDRBOOK EMPTY>
<!ATTLIST ADDRBOOK
  ADDRBOOK_ID CDATA #REQUIRED
  MEMBER_ID   CDATA #REQUIRED
  TYPE        CDATA #IMPLIED
  DISPLAYNAME CDATA #REQUIRED
  DESCRIPTION  CDATA #IMPLIED
>
<!ELEMENT ADDRESS EMPTY>
<!ATTLIST ADDRESS
  ADDRESS_ID CDATA #REQUIRED
  ADDRESSTYPE CDATA #IMPLIED
  MEMBER_ID  CDATA #REQUIRED
  ADDRBOOK_ID CDATA #REQUIRED
  ORGUNITNAME CDATA #IMPLIED
  FIELD3     CDATA #IMPLIED
  BILLINGCODE CDATA #IMPLIED
  BILLINGCODETYPE CDATA #IMPLIED
  STATUS     CDATA #IMPLIED
  ORGNAME    CDATA #IMPLIED
  ISPRIMARY  CDATA #IMPLIED
  LASTNAME   CDATA #IMPLIED
  PERSONTITLE CDATA #IMPLIED
  FIRSTNAME  CDATA #IMPLIED
  MIDDLENAME CDATA #IMPLIED
  BUSINESSTITLE CDATA #IMPLIED
  PHONE1     CDATA #IMPLIED
  FAX1       CDATA #IMPLIED
  PHONE2     CDATA #IMPLIED
  ADDRESS1   CDATA #IMPLIED
  FAX2       CDATA #IMPLIED
  NICKNAME   CDATA #REQUIRED
  ADDRESS2   CDATA #IMPLIED
  ADDRESS3   CDATA #IMPLIED
  CITY       CDATA #IMPLIED
  STATE      CDATA #IMPLIED
  COUNTRY    CDATA #IMPLIED
  ZIPCODE    CDATA #IMPLIED
  EMAIL1     CDATA #IMPLIED
  EMAIL2     CDATA #IMPLIED
  PHONE1TYPE CDATA #IMPLIED
  PHONE2TYPE CDATA #IMPLIED
  PUBLISHPHONE1 CDATA #IMPLIED
  PUBLISHPHONE2 CDATA #IMPLIED
  BESTCALLINGTIME CDATA #IMPLIED
  PACKAGESUPPRESSION CDATA #IMPLIED
  LASTCREATE CDATA #IMPLIED
  OFFICEADDRESS CDATA #IMPLIED
  SELFADDRESS  CDATA "0"
  FIELD1      CDATA #IMPLIED

```

```

FIELD2          CDATA #IMPLIED
TAXGEOCODE CDATA #IMPLIED
SHIPPINGGEOCODE CDATA          #IMPLIED
>

```

Creazione di uno schema e di un file XML dettagliato

In questo esempio, DTD Generator viene richiamato come riportato di seguito:

- 

```

java com.ibm.wca.DTDGenerator.GenerateDTD -dbname SAMPLE
-dbuser johndoe -dbpwd password -xmlTableDesc c:\sample\sample.xml
-outfile tables.dtd -tablenames "employee,staff"

```
- 

```

java com.ibm.wca.DTDGenerator.GenerateDTD -dbname SAMPLE
-dbuser johndoe -dbpwd password -xmlTableDesc usr/sample/sample.xml
-outfile tables.dtd -tablenames "employee,staff"

```
- 

```

QWECOMM/GENWCSDTD DATABASE(MYDB) SCHEMA(SAMPLE)
INSTROOT(/QIBM/UserData/WebCommerce/instances/mser) PASSWD(mypassword)
OUTFILE(tables.dtd) TABNAMES('employee,staff') XMLTABDESC(/sample/sample.xml)

```

Il file schema viene creato nella directory `sample` ed è denominato `WCAWebForm.xsd`.

Il file di emissione `sample.xml` contiene quanto segue:

```

<?xml version="1.0" encoding="UTF-8"?>
<formList xmlns="WCAWebForm.xsd" dbname="SAMPLE" dtdname="tables.dtd">
  <form name="EMPLOYEE">
    <uniqueIndex name="U2" columns="FIRSTNME, LASTNAME"/>
    <uniqueIndex name="U3" columns="MIDINIT, LASTNAME"/>
    <field name="EMPNO" type="string" maxLength="6" minOccurs='1'
      uniqueKey="true" showColumnInList="true" />
    <field name="FIRSTNME" type="string" maxLength="32" minOccurs='1'
      showColumnInList="true" />
    <field name="MIDINIT" type="string" maxLength="1" minOccurs='1'
      showColumnInList="true" />
    <field name="LASTNAME" type="string" maxLength="15" minOccurs='1'
      showColumnInList="true" />
    <field name="WORKDEPT" type="string" maxLength="3" showColumnInList="true" />
    <field name="PHONENO" type="string" maxLength="4"/>
    <field name="HIREDATE" type="date" maxLength="10"/>
    <field name="JOB" type="string" maxLength="8"/>
    <field name="EDLEVEL" type="integer" maxLength="5" minOccurs='1' />
    <field name="SEX" type="string" maxLength="1"/>
    <field name="BIRTHDATE" type="date" maxLength="10"/>
    <field name="SALARY" type="decimal" maxLength="9"/>
    <field name="BONUS" type="decimal" maxLength="9"/>
    <field name="COMM" type="decimal" maxLength="9"/>
  </form>
  <form name="STAFF">
    <field name="ID" type="integer" maxLength="5" minOccurs='1'
      uniqueKey="true" showColumnInList="true" />
    <field name="NAME" type="string" maxLength="9" showColumnInList="true" />
    <field name="DEPT" type="integer" maxLength="5" showColumnInList="true" />
    <field name="JOB" type="string" maxLength="5" showColumnInList="true" />
    <field name="YEARS" type="integer" maxLength="5" showColumnInList="true" />
    <field name="SALARY" type="decimal" maxLength="7"/>
    <field name="COMM" type="decimal" maxLength="7"/>
  </form>
</formList>

```

Capitolo 5. Risoluzione di identificativi

I dati XML che devono essere caricati in un database di destinazione devono contenere un identificativo per gli elementi XML che lo richiedono. Per creare o inserire gli identificativi per le voci di catalogo nel documento XML, richiamare il comando ID Resolve.

ID Resolver risolve soltanto gli identificativi per una tabella principale. Una tabella principale è una tabella elencata nella tabella KEYS o SUBKEYS. Se è necessario risolvere gli identificativi per una tabella che non si trova in KEYS o SUBKEYS, aggiungere questa tabella alla tabella SUBKEYS prima di eseguire ID Resolver.

Di seguito vengono riportati esempi di situazioni in cui è possibile utilizzare ID Resolver:

- Caricamento del nuovo contenuto nel modulo XML quando vengono richiesti gli identificativi per i dati
- Aggiornamento del contenuto quando già esistono gli identificativi per un oggetto nel database

ID Resolver può fornire identificativi reali o identificativi che possono essere trasformati utilizzando le seguenti tecniche:

- Risoluzione alias interno
- Specifica file properties
- Risoluzione indice univoco

Impostazione di ID Resolver

E' possibile impostare la modalità con cui ID Resolver gestisce i formati data/ora, la memorizzazione e i driver del database effettuando le seguenti operazioni:

1. Creare un nuovo file property di personalizzazione di ID Resolver.



Il file DB2ConnectionCustomizer.properties si trova nell'archivio IdResGen.zip. Estrarre questo file, rinominarlo lasciando l'estensione .properties e posizionarlo in una directory nel classpath. **Importante:** Non rimuovere il file DB2ConnectionCustomizer.properties esistente.



ISeries_RESWCSID_Customizer.properties si trova nella directory /QIBM/ProdData/WebCommerce/properties. Copiare questo file nella directory /rootist/xml, rinominare il nuovo file mantenendo l'estensione .properties, quindi apportare tutte le modifiche necessarie al nuovo file. **Importante:** Non rimuovere o modificare il file ISeries_RESWCSID_Customizer.properties originale.

2. Modificare i valori delle proprietà specificate nel nuovo file.
3. Specificare il nuovo nome file come valore del parametro di personalizzazione del comando ID Resolver.

Impostazione delle modalità di gestione dei formati data/ora di ID Resolver

Le seguenti maschere predefinite dei formati data/ora immissione vengono fornite nel file property di personalizzazione di ID Resolver:

```
InputTimeStampFormat.1 = yyyy-DD hh:mm:ss.SSSSSS
InputTimeStampFormat.2 = yyyy-MM-dd hh:mm:ss.SSSSSS
InputTimeStampFormat.3 = yyyy-DD-hh.mm.ss.SSSSSS
InputTimeStampFormat.4 = yyyy-MM-dd-HH.mm.ss.SSSSSS
InputTimeStampFormat.5 = yyyy-MM-dd-hh.mm.ss.SSSSSS
InputTimeStampFormat.6 = yyyy-MM-dd HH:mm:ss.SSSSSS
InputTimeStampFormat.7 = yyyy-DD HH:mm:ss.SSSSSS
```

E' possibile modificare le maschere dei formati data/ora o aggiungere tutte le maschere che si desidera al file property di personalizzazione di ID Resolver. Se si aggiunge un formato data/ora di immissione, è necessario utilizzare il numero successivo della sequenza corrente. Ad esempio, se viene fatta un'aggiunta all'elenco precedente, la maschera del formato data/ora di immissione successiva dovrebbe essere InputTimeStampFormat.8.

E' inoltre possibile personalizzare il formato data/ora di emissione, la maschera in microsecondi e il formato specifico del database modificando i valori delle seguenti proprietà nel file property di ID Resolver di personalizzazione:

```
TargetTimeStampFormat = yyyy-MM-dd HH:mm:ss.SSSSSS
MicroSecondMask = SSSSSS
DatabaseSpecificFormat = YYYY-MM-DD HH24:MI:SS
```

Impostazione delle modalità di gestione della memoria di ID Resolver

Di seguito viene riportata la sezione del file property di personalizzazione di ID Resolver che specifica i valori predefiniti per le proprietà rilevanti per gli hashmap permanenti:

```
////////////////////////////////////
/// 0 = Normal hashmap with no backend storage
/// 1 = JDBM
////////////////////////////////////

PersistentStorageType = 0

////////////////////////////////////
/// If PersistentStorageType != 0, set MemoryStorageSize to the maximum size
/// of the hashmap in memory data and after that the hashmap will stream
/// the data to a persistent storage as specified
/// If -1, then it uses the normal hashmap with no backend storage
////////////////////////////////////

MemoryStorageSize = 1
```

E' possibile specificare la modalità di gestione di ID Resolver della memoria permanente impostando un valore per PersistentStorageType nel file property di personalizzazione di ID Resolver.

- Se si imposta PersistentStorageType = 0, ID Resolver opera in modalità "normale" (in cui gli hashmap simbolo esistono in memoria).
- Se si imposta PersistentStorageType = 1, viene utilizzato JDBM per far persistere i simboli e le chiavi.

E' possibile specificare il numero di record memorizzati impostando un valore per `MemoryStorageSize` nel file property di personalizzazione di ID Resolver.

- Un valore pari a "1" per `MemoryStorageSize` indica che un solo record viene mantenuto in memoria.
- Un valore pari a "-1" per `MemoryStorageSize` ha un significato particolare, poiché indica che tutti i record vengono mantenuti in memoria.

In questo caso, ID Resolver ritorna al suo comportamento "normale".

Impostazione delle modalità di gestione dei driver del database di ID Resolver

Le seguenti righe del file property di personalizzazione di ID Resolver specificano i valori predefiniti per i driver del database:

```
DBVendorName = DB2
DBDriverName = COM.ibm.db2.jdbc.app.DB2Driver
DBURL = jdbc:db2:
```

in cui:

- `DBVendorName` viene utilizzato per selezionare il tipo di database.
Le opzioni sono le seguenti:
 - DB2 Universal Database per iSeries (DB2/iSeries)
 - DB2 per altri sistemi operativi (DB2)
 - Database Oracle (oracle)
- `DBDriverName` viene utilizzato per selezionare il driver JDBC.
Le opzioni sono le seguenti:
 - DB2 Universal Database per iSeries (com.ibm.db2.jdbc.app.DB2Driver)
 - DB2 per altri sistemi operativi (COM.ibm.db2.jdbc.app.DB2Driver)
 - Database Oracle (oracle.jdbc.driver.OracleDriver)
- `DBURL` viene utilizzato per specificare l'URL per accedere al database.
Le opzioni sono le seguenti:
 - DB2 Universal Database per iSeries (jdbc:db2://)
 - DB2 per altri sistemi operativi (jdbc:db2:)
 - Database Oracle (jdbc:oracle:oci8:@)

Determinazione delle modalità di elaborazione dati

Il comando ID Resolve consente di scegliere il metodo load, update o mixed per l'elaborazione del file di immissione.

- Utilizzare il metodo load per elaborare il file di immissione se **tutti** i record del file **non esistono** nel database.
- Utilizzare il metodo update per elaborare il file di immissione se **tutti** i record del file **esistono** nel database.
- Utilizzare il metodo mixed per elaborare il file di immissione se **solo alcuni** record del file **esistono** nel database.

Selezione del metodo load

Il metodo load per ID Resolver viene utilizzato per creare nuovi identificativi per i record caricati nel database. Con questo metodo, vengono creati nuovi identificativi per i record. Il seguente esempio viene utilizzato per creare identificativi per dati nuovi:

-  

```
idresgen -dbname db -dbuser user -dbpwd pwd -infile input.xml
-outfile output.xml -method load -customizer customizer -schemaname mall
```

Il metodo load è il metodo predefinito.

-   

```
./idresgen.sh -dbname db -dbuser user -dbpwd pwd -infile input.xml
-outfile output.xml -method load -customizer customizer -schemaname mall
```

Il metodo load è il metodo predefinito.

- 

```
QWEBCOMM/RESWCSID DATABASE(DATABASE_NAME) SCHEMA(MALL)
INSTROOT(/QIBM/UserData/WebCommerce/instances/mser)
PASSWD(mypassword) INFILE(input.xml) OUTFILE(output.xml)
METHOD(*LOAD)
```

Selezione del metodo update

Se viene specificato il metodo update per ID Resolver, i record nel file di immissione devono essere già presenti nel database. ID Resolver localizza gli identificativi nel database. Se un record non esiste nel database, ID Resolver non sarà in grado di risolvere l'identificativo per questo record e indicherà che si è verificato un errore. Il seguente esempio viene utilizzato per localizzare identificativi per i dati già presenti nel database:

-  

```
idresgen -dbname db -dbuser user -dbpwd pwd -infile input.xml
-outfile output.xml -method update -customizer customizer -schemaname mall
```

-   

```
./idresgen.sh -dbname db -dbuser user -dbpwd pwd -infile input.xml
-outfile output.xml -method update -customizer customizer -schemaname mall
```

- 

```
QWEBCOMM/RESWCSID DATABASE(DATABASE_NAME) SCHEMA(MALL)
INSTROOT(/QIBM/UserData/WebCommerce/instances/mser) PASSWD(mypassword)
INFILE(input.xml) OUTFILE(output.xml) METHOD(*UPD)
```

Selezione del metodo mixed

Se il file dei dati di immissione contiene record già presenti nel database e altri record completamente nuovi, ID Resolver deve essere eseguito utilizzando il metodo mixed. Con questo metodo, ID Resolver crea nuovi identificativi per i record soltanto se non esistono nel database. Altrimenti, nel database vengono rilevati gli identificativi già esistenti. Il seguente esempio viene utilizzato per creare identificativi per i dati nuovi e per localizzare gli identificativi per i dati già presenti nel database:

-  

```
idresgen -dbname db -dbuser user -dbpwd pwd -infile input.xml
-outfile output.xml -method mixed -customizer customizer -schemaname mall
```

-   

```
./idresgen.sh -dbname db -dbuser user -dbpwd pwd -infile input.xml
-outfile output.xml -method mixed -customizer customizer -schemaname mall
```

• **400**

```
QWECOMM/RESWCSID DATABASE(DATABASE_NAME) SCHEMA(MALL)
INSTROOT(/QIBM/UserData/WebCommerce/instances/mser) PASSWD(mypassword)
INFILE(input.xml) OUTFILE(output.xml) METHOD(*MIX)
```

Utilizzo delle tecniche di risoluzione di ID Resolver

ID Resolver può utilizzare un file properties Java per determinare le colonne di una tabella primaria da utilizzare come ricerca delle tabelle che richiedono l'identificativo di una tabella primaria. Una tabella viene definita primaria se è elencata nella tabella KEYS o nella tabella SUBKEYS.

Per utilizzare una risoluzione alias interno con ID Resolver, un alias viene inserito nell'attributo chiave principale (identificativo) del file XML. Quindi, l'alias può essere utilizzato nel file XML per indicare questo elemento. Grazie a questo processo non occorre utilizzare un programma per determinare gli indici univoci necessari per la creazione del file XML.

ID Resolver può inoltre analizzare lo schema del database per determinare se esiste un indice univoco che soddisfa i suoi requisiti. ID Resolver cerca un indice univoco soltanto quando non esistono voci nel file properties per la tabella da analizzare o quando non esiste il file properties. Se queste condizioni sono vere, viene eseguita una verifica dell'indice univoco. L'indice univoco viene considerato valido se esiste e non include la chiave primaria per la tabella.

Specifica di un file properties con ID Resolver

ID Resolver consente di utilizzare un file properties Java alternativo per descrivere le colonne di una voce primaria da utilizzare come ricerca delle tabelle che richiedono un identificativo della riga principale.

Il file properties predefinito è `IdResolveKeys.properties`, ma è possibile modificarlo o specificare il proprio file quando si richiama il comando ID Resolve.

- **NT** **2000** **AIX** **Solaris** **Linux**

`IdResolveKeys.properties` si trova nella seguente directory:

- **NT** *unità:* \WebSphere\CommerceServer\properties
- **2000** *unità:* \Program Files\WebSphere\CommerceServer\properties
- **AIX** /usr/WebSphere/CommerceServer/properties
- **Solaris** **Linux** /opt/WebSphere/CommerceServer/properties

Se quando si esegue ID Resolver, questo file non viene inserito nella directory corrente, è possibile inserirlo in una directory definita nella variabile di ambiente del classpath. E' anche possibile specificare il percorso completo di questo file.

• **400**

Per modificare il file `IdResolveKeys.properties`, copiarlo dalla directory `/QIBM/ProdData/WebCommerce/properties`, salvarlo nella directory `/rootist/xml`, quindi apportare tutte le modifiche necessarie al nuovo file.

Nota: La directory precedente si trova nel classpath utilizzato dal comando RESWCSID.

Utilizzo di un file properties per generare identificativi

Nel seguente esempio, è necessario risolvere gli identificativi ADDRBOOK_ID e ADDRESS_ID per i record ADDRBOOK e ADDRESS. Gli identificativi per i record MEMBER sono già conosciuti. Ogni record richiede un identificativo valido per il database di WebSphere Commerce. Inoltre, ADDRBOOK_ID nel record ADDRESS richiede l'identificativo della tabella principale per soddisfare il limite della relativa chiave esterna.

```
<MEMBER
  MEMBER_ID="100"
  TYPE="U"
  STATE="1"
/>
<MEMBER
  MEMBER_ID="101"
  TYPE="U"
  STATE="1"
/>
<ADDRBOOK
  MEMBER_ID="100"
  DISPLAYNAME="Friends"           Actual value of the DISPLAYNAME column
  DESCRIPTION="All my friends"
  TYPE="P"
/>
<ADDRESS
  ADDRBOOK_ID="@Friends"        Refers to the ADDRBOOK using the DISPLAYNAME
                                value as a lookup
  MEMBER_ID="101"
  NICKNAME="Bob"
  ADDRESS1="15 Brave Developers St."
  CITY="Toronto"
  ZIPCODE="A0A0A0"
  COUNTRY="Canada"
  STATUS="P"
/>
```

Un file properties è necessario per identificare le colonne nella riga primaria che verranno utilizzate dalle righe di relazione per risolvere il numero di riferimento per la colonna delle chiavi esterne. Di seguito vengono riportate le istruzioni necessarie per accertarsi che l'analisi del file precedente proceda in modo corretto.

In IdResolveKeys.properties specificare quanto segue:

```
NAMEDELIMITER=@
SELECTDELIMITER=:

ADDRBOOK=@DISPLAYNAME:DISPLAYNAME
ADDRESS=@NICKNAME:NICKNAME
```

NAMEDELIMITER e SELECTDELIMITER impostano i delimitatori utilizzati in tutto il file properties ed è necessario che vengano utilizzati costantemente.

ADDRBOOK=@DISPLAYNAME:DISPLAYNAME indica che quando viene ricevuto un record della rubrica, viene creato anche l'identificativo per la rubrica stessa. Il campo DISPLAYNAME viene estratto dal record di immissione e viene utilizzato per formare un'associazione al nuovo identificativo. La stringa DISPLAYNAME viene utilizzata per adattare la riga della rubrica DISPLAYNAME e per risolvere l'identificativo necessario per la chiave esterna.

Utilizzando l'esempio di immissione precedente, in cui DISPLAYNAME è Friends, supporre che l'identificativo creato per questo record sia 12951. DISPLAYNAME viene utilizzato come risoluzione chiave per 12951. L'elaborazione continua con il record successivo, ADDRESS, dove ADDRBOOK_ID ha il formato di @... che indica che ciò che segue il delimitatore deve essere utilizzato per effettuare la ricerca dell'identificativo della rubrica. La stringa corrisponde a DISPLAYNAME e 12951 viene restituito e posizionato nell'attributo ADDRBOOK_ID.

```

<MEMBER
  MEMBER_ID="100"
  TYPE="U"
  STATE="1"
/>
<MEMBER
  MEMBER_ID="101"
  TYPE="U"
  STATE="1"
/>
<ADDRBOOK
  ADDRBOOK_ID="12951"
  MEMBER_ID="100"
  DISPLAYNAME="Friends"
  DESCRIPTION="All my friends"
  TYPE="P"
/>
<ADDRESS
  ADDRESS_ID="13051"
  ADDRBOOK_ID="12951"
  MEMBER_ID="101"
  NICKNAME="Bob"
  ADDRESS1="15 Brave Developers St."
  CITY="Toronto"
  ZIPCODE="A0A0A0"
  COUNTRY="Canada"
  STATUS="P"
/>

```

Generated primary key
Value of ADDRBOOK DISPLAYNAME unchanged

Generated primary key
ADDRESS refers to correct ADDRBOOK

Utilizzo di un file properties con chiavi composte

Una chiave costituita da più di due colonne è una chiave composta. E' possibile definire una chiave composta effettuando la ricerca nel file properties specificando NAMEDELIMITER e SELECTDELIMITER seguiti dai nomi del campo. Per impostare i criteri di ricerca per i record ADDRBOOK come composti dall'ID membro e dal nome visualizzato, specificare quanto segue nel file properties:

```
ADDRBOOK=@DISPLAYNAME@MEMBER_ID:DISPLAYNAME MEMBER_ID
```

quindi il frammento del file di immissione XML di seguito riportato:

```

<ADDRBOOK
  MEMBER_ID="100"
  DISPLAYNAME="Friends"
  DESCRIPTION="All my friends"
  TYPE="P"
/>
<ADDRBOOK
  MEMBER_ID="101"
  DISPLAYNAME="Friends"
  DESCRIPTION="All my friends"
  TYPE="P"
/>

```

ADDRBOOK "Friends" of MEMBER 100

ADDRBOOK "Friends" of MEMBER 101

```

<ADDRESS
  ADDRBOOK_ID="@Friends@100"
  MEMBER_ID="101"
  NICKNAME="Bob"
  ADDRESS1="15 Brave Developers St."
  CITY="Toronto"
  ZIPCODE="A0A0A0"
  COUNTRY="Canada"
  STATUS="P"
/>

```

**Lookup the primary key for ADDRBOOK
"Friends" of MEMBER 100**

dopo la risoluzione, il file verrà visualizzato nel modo di seguito riportato:

```

<MEMBER
  MEMBER_ID="100"
  TYPE="U"
  STATE="1"
/>
<MEMBER
  MEMBER_ID="101"
  TYPE="U"
  STATE="1"
/>
<ADDRBOOK
  ADDRBOOK_ID="12951"
  MEMBER_ID="100"
  DISPLAYNAME="Friends"
  DESCRIPTION="All my friends"
  TYPE="P"
/>
<ADDRBOOK
  ADDRBOOK_ID="12952"
  MEMBER_ID="101"
  DISPLAYNAME="Friends"
  DESCRIPTION="All my friends"
  TYPE="P"
/>
<ADDRESS
  ADDRESS_ID="13051"
  ADDRBOOK_ID="12951"
  MEMBER_ID="101"
  NICKNAME="Bob"
  ADDRESS1="15 Brave Developers St."
  CITY="Toronto"
  ZIPCODE="A0A0A0"
  COUNTRY="Canada"
  STATUS="P"
/>

```

ADDRBOOK of interest

ADDRESS refers to correct ADDRBOOK

Utilizzo di un file properties con chiavi principali a cascata

La tabella principale STOREENT definisce una chiave principale STOREENT_ID. La tabella STORE, una tabella esterna che fa riferimento a STOREENT, definisce una chiave principale STORE_ID che rappresenta una chiave principale per la tabella principale STOREENT. Ciò significa che il valore STORE_ID deve essere uno dei valori di STOREENT_ID. STORE_ID, la chiave principale della tabella esterna STORE, ha un duplice ruolo, principale ed esterna.

Si supponga che un'altra tabella, CONTRACT, sia la tabella esterna per STORE e che la chiave esterna per CONTRACT, STORE_ID, faccia riferimento alla chiave principale STORE_ID di STORE. La tabella STORE è pertanto una tabella principale per la tabella CONTRACT.

Poiché il valore STORE_ID della tabella STORE invece di essere creato costituisce un riferimento di STOREENT_ID, ID Resolver non crea un'associazione alias interno e valore ID per la tabella STORE. Quando la tabella CONTRACT prova a risolvere il valore STORE_ID della tabella STORE, viene rilevato un valore vuoto.

A causa di queste particolari condizioni, è necessario specificare esplicitamente la creazione di un alias interno creando una voce all'interno del file properties. In IdResolveKeys.properties specificare quanto segue:

```
"STORE=@STORE_ID:STORE_ID"
```

Ciò implica che ID Resolver effettui quanto riportato di seguito:

- Creazione dell'alias interno e dell'associazione ID-valore durante la risoluzione di STORE_ID come riferimento esterno
- Utilizzo dell'associazione durante la risoluzione di STORE_ID per la tabella CONTRACT

Utilizzo della voce STORE=@STORE_ID:STORE_ID nel file properties e nel seguente frammento del file di immissione XML:

```
<STOREENT
  IDENTIFIER="Out Fashions"
  MEMBER_ID="-2000"
  STOREENT_ID="@storeent_id_1"
  TYPE="G"
/>
<STORE
  STORE_ID="@storeent_id_1"
  STOREGRP_ID="1"
  STORELEVEL="store_level"
/>
<CONTRACT
  CONTRACT_ID="@contract_id_1"
  STATE="0"
  STORE_ID="@storeent_id_1"
/>
```

dopo la risoluzione, il file verrà visualizzato nel modo di seguito riportato:

```
<STOREENT
  IDENTIFIER="Out Fashions"
  MEMBER_ID="-2000"
  STOREENT_ID="10501"
  TYPE="G"
/>
<STORE
  STORE_ID="10501"
  STOREGRP_ID="1"
  STORELEVEL="store_level"
/>
<CONTRACT
  CONTRACT_ID="@contract_id_1"
  STATE="0"
  STORE_ID="10501"
/>
```

Utilizzo della risoluzione alias interno

Per utilizzare una risoluzione alias interno con ID Resolver, un alias viene inserito nell'attributo chiave principale (identificativo) del file XML. Quindi, l'alias può

essere utilizzato nel file XML per indicare questo elemento. Grazie a questo processo non occorre utilizzare un programma per determinare gli indici univoci necessari per la creazione del file XML.

Gli alias interni devono essere utilizzati costantemente nel file. Se per un ID di rubrica ADDRBOOK_ID viene creato un alias @addrbook_1, tutti i riferimenti della chiave esterna a tale ID devono utilizzare @addrbook_1. Gli alias sono transitori. Gli alias non vengono salvati e non è possibile utilizzarli in un altro file XML senza introdurli di nuovo.

Esempio parziale di utilizzo della risoluzione ID alias interno

Prima della risoluzione:

```
<MEMBER
  MEMBER_ID="100"
  TYPE="U"
  STATE="1"
/>
<ADDRBOOK
  ADDRBOOK_ID="@addrbook_1"           Alias for ADDRBOOK
  MEMBER_ID="100"
  DISPLAYNAME="Friends"
  DESCRIPTION="All my friends"
  TYPE="P"
/>
<ADDRESS
  ADDRESS_ID="@address_1"             Alias for ADDRESS
  ADDRBOOK_ID="@addrbook_1"         Refers to the alias for ADDRBOOK
  MEMBER_ID="101"
  NICKNAME="Bob"
  ADDRESS1="1 Brave Developer St."
  CITY="Toronto"
  ZIPCODE="A3B0F4"
  COUNTRY="Canada"
  STATUS="P"
/>
```

Dopo la risoluzione:

```
<MEMBER
  MEMBER_ID="100"
  TYPE="U"
  STATE="1"
/>
<ADDRBOOK
  ADDRBOOK_ID="11801"                 Generated primary key
  MEMBER_ID="100"
  DISPLAYNAME="Friends"
  DESCRIPTION="All my friends"
  TYPE="P"
/>
<ADDRESS
  ADDRESS_ID="11901"                 Generated primary key
  ADDRBOOK_ID="11801"               Refers to ADDRBOOK entry
  MEMBER_ID="100"
  NICKNAME="Bob"
  ADDRESS1="1 Brave Developer St."
  CITY="Toronto"
  ZIPCODE="A3B0F4"
  COUNTRY="Canada"
  STATUS="P"
/>
```

Utilizzo della risoluzione indice univoco

La risoluzione indice univoco, la modalità di comportamento predefinita di ID Resolver, viene utilizzata se non esiste alcuna voce nel file properties per la tabella che viene analizzata oppure se non esiste alcun file properties. Questo tipo di risoluzione utilizza un qualsiasi indice univoco specificato su una tabella come mezzo per la localizzazione dell'identificativo. Ad esempio, MEMBER_ID, insieme a IDENTIFIER, è un indice univoco della tabella CATALOG e può essere utilizzato come punto di risoluzione nella chiave primaria CATALOG_ID della tabella CATALOGDSC.

Per aggiornare il contenuto del database, è necessario conoscere la chiave univoca dalla tabella principale del database. E' possibile effettuare un'interrogazione nel database per rilevare tale chiave. Ad esempio, un comando DB2 per richiamare una chiave univoca potrebbe avere il seguente aspetto:

```
db2 describe indexes for table schema.tablename show detail
```

Esempio parziale di risoluzione indice univoco

Prima della risoluzione:

```
<MEMBER
  MEMBER_ID="100"
  TYPE="0"
  STATE="1"
/>

<CATALOG
  DESCRIPTION="Winter Catalog"
  IDENTIFIER="WC2001"
  MEMBER_ID="100"
  TPCLEVEL="2"
/>

<CATALOGDSC
  CATALOG_ID="@WC2001@100"
  FULLIMAGE="c:\store\img\wc.gif"
  LANGUAGE_ID="-1"
  LONGDESCRIPTION="2001 Winter Catalog"
  SHORTDESCRIPTION="2001 Winter Catalog"
  NAME="InFashion 2001 Winter Catalog"
  THUMBNAIL="c:\store\img\wc_th.gif"
/>
```

Refers back to catalog "WC2001" of member "100" (Note: The order is important.)

Dopo la risoluzione:

```
<MEMBER
  MEMBER_ID="100"
  TYPE="0"
  STATE="1"
/>

<CATALOG
  CATALOG_ID="10351"
  DESCRIPTION="Winter Catalog"
  IDENTIFIER="WC2001"
  MEMBER_ID="100"
  TPCLEVEL="2"
/>

<CATALOGDSC
  CATALOG_ID="10351"
  FULLIMAGE="c:\store\img\wc.gif"
/>
```

Automatically generated primary key

Refers to the correct catalog

```
LANGUAGE_ID="-1"  
LONGDESCRIPTION="2001 Winter Catalog"  
SHORTDESCRIPTION="2001 Winter Catalog"  
NAME="InFashion 2001 Winter Catalog"  
THUMBNAIL="c:\store\img\wc_th.gif"  
/>
```

Caricamento dei dati nella tabella MEMBER

ID Resolver gestisce risoluzioni per le tabelle che dispongono di identificativi generati dal sistema. Ciò include ogni tabella e colonna registrata nella tabella KEYS o SUBKEYS. Questa risoluzione ha due componenti:

1. Determina se una tabella principale (ovvero, una tabella elencata in KEYS o SUBKEYS) è presente nel database.

Tale risoluzione si basa sui contenuti dei dati XML per quell'elemento utilizzando risoluzioni indice univoco o specifiche di file properties.

2. Determina se è presente una chiave esterna in una tabella principale. Ciò viene fatto con una specifica della risoluzione nell'attributo della chiave esterna della relativa tabella.

La tabella MEMBER viene utilizzata come "classe superiore" delle tabelle ORGENTITY, MBRGRP, e USER. Ciò crea un modello "is-a" utile per gestire l'integrità referenziale quando le tabelle sono vincolate dalla chiave esterna ai sottotipi della tabella MEMBER. Poiché tutti i sottotipi MEMBER condividono un tipo di database comune, l'identificativo deve essere unico per tutti i sottotipi. Ciò significa che ORGENTITY_ID deve essere unico nell'impostazione MBRGRP_ID e USER_ID. Per realizzare ciò, la tabella KEYS si riferisce soltanto alle tabelle ORGENTITY, MBRGRP, e USER e specifica intervalli limitati per gli identificativi. Ogni sottotipo dispone di una chiave principale; ogni chiave principale rappresenta anche una chiave esterna per la chiave principale della tabella MEMBER.

Il vincolo tra MEMBER ed i relativi sottotipi crea una situazione in cui un MEMBER ed un sottotipo non possono avere un ID sincronizzato. Per caricare le tabelle ORGENTITY, MBRGRP e USER nel sistema, ID Resolver riconosce il modello "is-a" gestendolo in maniera appropriata. La seguente sintassi XML per ID Resolver:

```
<ORGENTITY
  ORGENTITY_ID="@orgAlias"
  ORGENTITYNAME="Test Org"
  ORGENTITYTYPE="0">
  <ISA>
    <MEMBER
      TYPE="0"
      STATE="1"
    />
  </ISA>
</ORGENTITY>
```

genera quanto segue:

```
<MEMBER
  MEMBER_ID="12345"
  TYPE="0"
  STATE="1"
/>
<ORGENTITY
  ORGENTITY_ID="12345"           Synchronized with member element
  ORGENTITYNAME="Test Org"
  ORGENTITYTYPE="0"
/>
```

In questo modo, ID Resolver gestisce il sottoelemento <isa> e crea un identificativo sincronizzato.

Creazione di una relazione esterna utilizzando la tabella REFKEYS

La tabella REFKEYS viene creata per rappresentare una relazione esterna tra le tabelle che non esistono ancora nel database. Generalmente, lo schema del database descrive le relazioni esterne creando una dichiarazione di chiave esterna che collega una colonna di una tabella ad un'altra tabella. Se nello schema del database non è definita la relazione esterna e gli identificativi vengono risolti come una chiave esterna, allora procedere come segue:

1. Creare una tabella REFKEYS come illustrato nel seguente DDL di esempio:

```
CREATE TABLE "REFKEYS" (  
    "FKTABLE_NAME" CHAR(18) NOT NULL ,  
    "FKCOLUMN_NAME" CHAR(18) NOT NULL ,  
    "TABLENAME" CHAR(18) NOT NULL  
);
```

in cui:

FKTABLE_NAME è il nome della tabella esterna (o "child")

FKCOLUMN_NAME è il nome della colonna esterna

TABLENAME è il nome della tabella principale (o "parent")

2. Creare una voce nella tabella REFKEYS che descrive le relazioni esterne richieste.

Risoluzione dei problemi

Se durante la risoluzione degli identificativi si verificano errori, fare riferimento alla seguente tabella:

Errore	Metodo utilizzato	Possibile causa	Possibile soluzione	
Chiave primaria non risolta	Tutti	ID Resolver non risolve la chiave primaria (identificativo) in una tabella non specificata nella tabella KEYS o SUBKEYS.	<p>Aggiungere il nome della tabella per la quale occorre risolvere la chiave primaria alla tabella SUBKEYS prima di eseguire ID Resolver.</p> <p>Inoltre, assicurarsi che la relazione della chiave esterna sia presente nello schema di database.</p>	
	Aggiorna	La chiave primaria viene risolta eseguendo una query nel database. La query del database viene generata utilizzando le informazioni sull'entrata del file properties o l'indice univoco per la specifica tabella. L'entrata del file properties ha la priorità.	<p>Assicurarsi che le informazioni sull'indice univoco presenti nel file di immissione siano corrette.</p> <p>E' anche possibile creare o modificare l'entrata appropriata nel file properties per generare la query di database corretta per risolvere la chiave primaria.</p>	
Chiave esterna non risolta	Tutti	ID Resolver non risolve la chiave esterna in una tabella in cui i riferimenti esterni puntano a tabelle non specificate nella tabella KEYS o SUBKEYS.	<p>Aggiungere il nome della tabella che fa riferimento alla tabella SUBKEYS prima di eseguire ID Resolver.</p> <p>Inoltre, assicurarsi che la relazione della chiave esterna sia presente nello schema di database.</p>	
		Una chiave esterna viene risolta utilizzando l'alias interno o eseguendo una query nel database. La query del database viene eseguita solo se l'alias interno non riesce a risolvere la chiave.	L'alias interno viene generato utilizzando la chiave primaria e la voce del file di proprietà.	Assicurarsi che i valori di alias interno utilizzati come origine e destinazione del file di immissione siano corretti.
			La query del database viene generata utilizzando le informazioni sull'entrata del file properties o l'indice univoco per la specifica tabella. L'entrata del file properties ha la priorità.	<p>Assicurarsi che le informazioni sull'indice univoco presenti nel file di immissione siano corrette.</p> <p>E' anche possibile creare o modificare l'entrata appropriata nel file properties per generare la query di database corretta per risolvere la chiave primaria.</p>

Capitolo 6. Caricamento di dati

Prima di poter caricare i dati, è necessario svolgere le seguenti operazioni:

1. Generare un DTD ed uno schema per l'utilizzo con il pacchetto Loader (i primi dati sull'orario da caricare)

Nota: Se si stanno caricando dei dati per un archivio negozio e se è stato creato il file XML utilizzando i DTD forniti con l'archivio negozio, questo passaggio non è necessario.

2. Risolvere gli identificativi (se necessario)

I dati devono avere un modulo XML con il DTD associato. Per caricare i dati, richiamare il comando Load.

Impostazione del pacchetto Loader

Con il pacchetto Loader è possibile impostare il funzionamento effettuando le seguenti operazioni:

- Esecuzione della procedura per ignorare gli elementi del file di immissione
- Inserimento di NULL in una colonna
- Caricamento di dati relativi alla data e ai formati data/ora
- Caricamento dei formati data/ora correnti
- Gestione delle code eventi
- Esecuzione su diversi sistemi operativi e software di database
- Sostituzione di un componente
- Utilizzo della sincronizzazione dello spazio di ricerca di Product Advisor

E' possibile personalizzare le funzioni del pacchetto Loader mediante le seguenti operazioni:

1. Creare un nuovo file property di personalizzazione di Loader.



MassLoadCustomizer.properties si trova nell'archivio MassLoader.zip. Estrarre questo file, rinominarlo lasciando l'estensione .properties e posizionarlo in una directory nel classpath. **Importante:** Non rimuovere o modificare il file MassLoadCustomizer.properties esistente.



ISeries_LODWCSDTA_Customizer.properties si trova nella directory /QIBM/ProdData/WebCommerce/properties. Copiare questo file nella directory /rootist/xml, rinominare il nuovo file mantenendo l'estensione .properties, quindi apportare tutte le modifiche necessarie al nuovo file. **Importante:** Non rimuovere o modificare il file ISeries_LODWCSDTA_Customizer.properties originale.

2. Modificare i valori delle proprietà specificate nel nuovo file property di personalizzazione del pacchetto Loader.
3. Specificare il nuovo nome file come valore del parametro di personalizzazione del comando Load.

Come ignorare gli elementi del file di immissione

Se il file di immissione contiene elementi che non sono associati al database di destinazione, è possibile impostare il pacchetto Loader in modo che tali elementi vengano ignorati nel file property di personalizzazione del pacchetto Loader. Utilizzare IgnoreElements per specificare gli elementi da ignorare e separare tali elementi con un punto e virgola (;). Per ignorare gli elementi import, literals e ProductRepository, ad esempio, specificare quanto segue nel file property di personalizzazione del pacchetto Loader:

```
IgnoreElements = import;literals;ProductRepository
```

Inserimento di un valore NULL in una colonna

E' possibile abilitare il pacchetto Loader per l'inserimento di un valore NULL in una colonna impostando la proprietà EnableNULLCheck su "true" nel file property di personalizzazione del pacchetto Loader. Ad esempio:

```
EnableNULLCheck = true
```

Per motivi di prestazioni, questa funzione è disabilitata per impostazione predefinita.

Utilizzare la proprietà NULLStringLiteral per determinare la rappresentazione della stringa di un valore null all'interno dei dati. Per impostare il pacchetto Loader in modo che la stringa "-" venga utilizzata per rappresentare un valore null, ad esempio, specificare la proprietà e il valore di seguito riportati nel file property di personalizzazione del pacchetto Loader:

```
NULLStringLiteral = -
```

Per impostazione predefinita, il valore di questa proprietà è "NULL" (senza virgolette).

Caricamento di dati relativi alla data e ai formati data/ora

Il pacchetto Loader può caricare i dati all'interno di colonne con i formati data/ora e le date. Il formato dei dati per data/ora e data nel documento vengono determinati dai modelli personalizzabili. L'utente può modificare un modello esistente oppure può aggiungere altri modelli all'elenco dei modelli esistenti.

I dati per un formato data/ora o per una data vengono controllati con i modelli disponibili. Il primo modello che corrisponde ai dati viene utilizzato per convertire i dati nel formato data/ora di destinazione prima che avvenga il caricamento nel database.

Sono disponibili due modelli personalizzabili di formato data/ora di emissione, TimeStampFormat.JDBC e TimeStampFormat.Load.

1. TimeStampFormat.JDBC viene utilizzato quando il pacchetto Loader utilizza connessioni JDBC per eseguire un'operazione.
SQL importa ed elimina i metodi del pacchetto Loader utilizzando le connessioni JDBC per l'aggiornamento del database.
2. TimeStampFormat.Load viene utilizzato quando il pacchetto Loader utilizza i programmi di utilità nativi.
I metodi import e load del pacchetto Loader utilizzano i programmi di utilità nativi.

E' possibile personalizzare i formati data/ora modificando o aggiungendo maschere nel file property di personalizzazione del pacchetto Loader.

Vengono fornite le seguenti maschere per i formati data/ora di immissione:

```
InputTimeStampFormat.1 = yyyy-DD hh:mm:ss.SSSSSS
InputTimeStampFormat.2 = yyyy-MM-dd hh:mm:ss.SSSSSS
InputTimeStampFormat.3 = yyyy-DD-hh.mm.ss.SSSSSS
InputTimeStampFormat.4 = yyyy-MM-dd-HH.mm.ss.SSSSSS
InputTimeStampFormat.5 = yyyy-MM-dd-hh.mm.ss.SSSSSS
InputTimeStampFormat.6 = yyyy-MM-dd HH:mm:ss.SSSSSS
InputTimeStampFormat.7 = yyyy-DD HH:mm:ss.SSSSSS
```

I modelli predefiniti per i formati della data di immissione sono i seguenti:

```
InputDateFormat.1 = MM-dd-yyyy
InputDateFormat.2 = yyyy-dd-MM
InputDateFormat.3 = yyyy-MM-dd
InputDateFormat.4 = MM/dd/yyyy
InputDateFormat.5 = yyyy/dd/MM
InputDateFormat.6 = yyyy-DD
```

E' possibile modificare queste maschere relative ai formati data/ora e data oppure aggiungere il numero di maschere desiderate. Specificare queste maschere nel file property di personalizzazione del pacchetto Loader nella sequenza numerica secondo la quale si desidera che vengano confrontate con il formato data/ora di immissione. Se si aggiunge un formato data/ora di immissione, è necessario utilizzare il numero successivo della sequenza corrente. Ad esempio, se viene fatta un'aggiunta all'elenco precedente, la maschera del formato data/ora di immissione successiva dovrebbe essere InputTimeStampFormat.8.

I modelli per la formattazione dei dati di immissione per l'emissione del formato data/ora e data sono i seguenti:

```
TimeStampFormat.JDBC = aaaa-MM-gg hh:mm:ss.SSSSSS
TimeStampFormat.Load = yyyy-MM-dd-hh.mm.ss.SSSSSS

DateFormat.JDBC = yyyy-MM-dd
DateFormat.Load = yyyy-MM-dd
```

In genere, i formati data e data/ora *non* sono personalizzati.

Caricamento dei formati data/ora correnti

Con il pacchetto Loader è possibile inserire all'interno delle colonne i valori con un tipo di dati relativi al formato data/ora in base alla lettura dell'orario giornaliero. Ad esempio, STARTDATE e ENDDATE per un'offerta in WebSphere Commerce possono avere valori basati sull'orario in cui l'offerta è stata inserita nella tabella. Per supportare questa funzione, il pacchetto Loader utilizza la tabella MLTIME per mantenere le istanze dei formati data/ora. Lo schema di questa tabella è il seguente:

```
table MLTIME
(
  INSTANCEID BIGINT not null,
  MLTIMESTAMP TIMESTAMP
)
```

E' possibile personalizzare il nome della tabella e le relative colonne modificando le proprietà di seguito riportate nel file property di personalizzazione del pacchetto Loader:

```
TimestampTableName = MLTIME
TimestampIdColumn = INSTANCEID
TimestampValueColumn = MLTIMESTAMP
```

I dati di immissione per la specifica dei valori del formato data/ora corrente si basano sui modelli della stringa del formato data/ora. Le seguenti maschere vengono utilizzate per la specifica della durata dei formati data/ora:

```
%D per i giorni
%M per i mesi
%Y per gli anni
%H per le ore
%m per i minuti
%s per i secondi
```

E' possibile personalizzare i formati data/ora correnti modificando o aggiungendo maschere nel file property di personalizzazione del pacchetto Loader. Vengono fornite le seguenti maschere di immissione:

```
InputCurrentTimestampFormat.1 = CURRENT TIMESTAMP
InputCurrentTimestampFormat.2 = CURRENT TIMESTAMP %D DAYS
InputCurrentTimestampFormat.3 = CURRENT TIMESTAMP %D DAYS %M MONTHS
InputCurrentTimestampFormat.4 = CURRENT TIMESTAMP %D DAYS %M MONTHS %Y YEARS
InputCurrentTimestampFormat.5 = CURRENT TIMESTAMP %Y YEARS %M MONTHS %D DAYS
InputCurrentTimestampFormat.6 = SYSDATE
InputCurrentTimestampFormat.7 = ADDDAYS(SYSDATE,%D)
InputCurrentTimestampFormat.8 = ADDDAYS(ADDMONTHS(SYSDATE,%M),%D)
InputCurrentTimestampFormat.9 = ADDDAYS(ADDMONTHS(ADDYEARS(SYSDATE,%Y),%M),%D)
```

I dati di immissione per il formato data/ora corrente corrispondono ai modelli specificati. Se i dati corrispondono ad un modello di immissione specificato, tale modello viene utilizzato per analizzare i dati di immissione ed il pacchetto Loader converte i dati nel formato di output appropriato inserendoli nel database. E' possibile aggiungere nuovi modelli all'elenco precedente se i numeri degli script secondari sono ordinati in maniera sequenziale.

Sono disponibili due formati di emissione di destinazione per la specifica dei formati data/ora correnti:

1. CurrentTimestampFormat.Load viene utilizzato quando il pacchetto Loader funziona in modalità load o import.
2. CurrentTimestampFormat.JDBC viene utilizzato quando il pacchetto Loader utilizza JDBC per inserire, aggiornare o eliminare i valori nel database.

I modelli di destinazione predefiniti nel pacchetto Loader sono i seguenti:

```
CurrentTimestampFormat.Load = CURRENT TIMESTAMP %Y YEARS %M MONTHS %D DAYS
                              %h HOURS %m MINUTES %s SECONDS
CurrentTimestampFormat.JDBC = CURRENT TIMESTAMP %Y YEARS %M MONTHS %D DAYS
                              %h HOURS %m MINUTES %s SECONDS
```

E' possibile personalizzare le proprietà per tali modelli all'interno del file property di personalizzazione del pacchetto Loader. Quando si personalizzano le proprietà CurrentTimestampFormat.Load e CurrentTimestampFormat.JDBC, è necessario verificare che la sintassi delle istruzioni che ne risultano siano valide per il sistema di gestione del database utilizzato.

La proprietà `CurrentTimestampLiteral` viene utilizzata dal pacchetto Loader per determinare se il valore della colonna del formato data/ora è nel formato data/ora corrente, evitando in questo modo costose elaborazioni per determinare se il valore non è una rappresentazione della stringa del formato data/ora.

`CurrentTimestampLiteral = CURRENT TIMESTAMP`



Il valore predefinito per questa proprietà per DB2 è `CURRENT TIMESTAMP`.



Il valore predefinito per il database Oracle è `SYSDATE`.

Esempio di caricamento dei formati data/ora correnti

Il pacchetto Loader fornisce le informazioni di seguito riportate per aggiornare l'offerta con `OFFER_ID` pari a 10123. La data di inizio ha un valore "`CURRENT TIMESTAMP`", mentre la data di fine ha un valore "`CURRENT TIMESTAMP + 14 DAYS`".

```
<OFFER
  OFFER_ID="10123"
  STARTDATE="CURRENT TIMESTAMP">
  ENDDATE="CURRENT TIMESTAMP + 14 DAYS"
/>
```

Il pacchetto Loader riconosce che le colonne `STARTDATE` e `ENDDATE` hanno il formato data/ora presente nel database. In base alle proprietà `CurrentTimeStampLiteral`, i valori vengono riportati in modo tale che vengano specificati nel formato data/ora corrente. Il valore per `STARTDATE` corrisponde al modello `InputCurrentTimeStampFormat.1` e viene convertito nel modello specificato dalla proprietà `CurrentTimeStampFormat.JDBC`. Il valore per `ENDDATE` corrisponde al formato della proprietà `InputCurrentTimeStampFormat.2` e viene convertito nel modello specificato dalla proprietà `CurrentTimeStampFormat.JDBC`.

Esempi di aggiunta di durate ai formati data/ora correnti

Il pacchetto Loader consente di aggiungere una durata ai formati data/ora correnti, ad esempio, nel caso in cui si desideri caricare un'offerta senza immettere una data specifica. Affinché ciò sia possibile, è necessario creare una data di fine che abbia una durata dopo la data di inizio. Il seguente esempio funziona perfettamente con DB2:

```
<Offer
  Startdate="Current Timestamp"
  Enddate="Current Timestamp +14 Days +4 Months +1 Year +0 Hours
  +0 Minutes +0 Seconds"
/>
```

Per gestire le durate dei formati data/ora correnti in una modalità indipendente dalla piattaforma, è necessario personalizzare i formati data/ora correnti modificando le maschere all'interno del file property di personalizzazione del pacchetto Loader. Di seguito viene riportato un esempio di specifiche della proprietà del formato data/ora corrente personalizzato:

`CurrentTimestampLiteral=Current Timestamp`

```
InputCurrentTimeStampFormat.0=Current Timestamp
InputCurrentTimeStampFormat.1=Current Timestamp %D Days
InputCurrentTimeStampFormat.2=Current Timestamp %M Months
InputCurrentTimeStampFormat.3=Current Timestamp %Y Years
```

```

InputCurrentTimestampFormat.4=Current Timestamp %D Days %M Months
InputCurrentTimestampFormat.5=Current Timestamp %D Days %M Months %Y Years
InputCurrentTimestampFormat.5=Current Timestamp %H Hours %m Minutes %s Seconds

```

```

CurrentTimestampFormat.JDBC=Current Timestamp %D Days %M Months %Y Years
%H Hours %m Minutes %s Seconds

```

Utilizzando l'esempio di offerta e le specifiche della proprietà, la data di fine per l'offerta corrisponderà al modello InputCurrentTimestampFormat.5. Ciò fornisce le seguenti informazioni sull'offerta utilizzando CurrentTimestampFormat.JDBC.

```

<Offer
  Startdate="Current Timestamp"
  Enddate="Current Timestamp +14 Days +4 Months +1 Year +0 Hours +0 Minutes +0 Seconds"
/>

```

L'esempio precedente illustra come il pacchetto Loader possa immettere più formati data/ora correnti e trasformarli nel formato di emissione desiderato. Il seguente esempio illustra come gestire formati indipendenti dalla piattaforma e come creare un'associazione con i formati di emissione specifici per una data piattaforma.

```

<Offer
  Startdate="Now"
  Enddate="Now +14D +4M +1Y"
/>

```

```

CurrentTimestampLiteral=Now

```

```

InputCurrentTimestampFormat.0=Now
InputCurrentTimestampFormat.1=Now %DD
InputCurrentTimestampFormat.2=Now %MM
InputCurrentTimestampFormat.3=Now %YY
InputCurrentTimestampFormat.4=Now %DD %MM
InputCurrentTimestampFormat.5=Now %DD %MM %YY
InputCurrentTimestampFormat.5=Sysdate %HH %mm %ss

```

```

CurrentTimestampFormat.JDBC=AddYears (AddMonths (AddDays (AddHours (AddMinutes (AddSeconds
(Sysdate,%s),%m),%H),%D),%M),%Y)

```

Nota: L'istruzione precedente è semplicemente un esempio. Viene utilizzata soltanto per illustrare la funzione di personalizzazione per il sistema di gestione di un ipotetico database. Non è valida per DB2 o per un database Oracle.

Utilizzando l'esempio di offerta e le specifiche della proprietà, la data di fine per l'offerta corrisponderà al modello InputCurrentTimestampFormat.5. Ciò fornisce le seguenti informazioni sull'offerta utilizzando CurrentTimestampFormat.JDBC.

```

<Offer
  Startdate="Current Timestamp"
  Enddate="AddYears (AddMonths (AddDays (AddMinutes (AddSeconds (Sysdate,0),0),0),14),4),1)"
/>

```

Gestione delle code eventi

E' possibile personalizzare le code eventi del pacchetto Loader modificando le impostazioni nel file property di personalizzazione del pacchetto Loader. Ad esempio:

```

QueueLowCount = 35
QueueHighCount = 90

```

L'origine degli eventi che riempiono la coda viene bloccata quando il numero di elementi della coda raggiunge il limite superiore, evitando così che altri elementi vengano messi in coda. La coda comincia ad accettare nuovamente gli eventi quando il numero di elementi presenti nella coda eccede il limite inferiore.

Esecuzione su diversi sistemi operativi e software di database

E' possibile personalizzare il pacchetto Loader affinché venga eseguito con diversi sistemi operativi e software di database modificando i parametri relativi agli elementi di seguito riportati nel file property di personalizzazione del pacchetto Loader per specificare i diversi sistemi operativi e software di database:

- Comando database-connection
- Comando del database load-table
- Comando database-import
- Comando system che richiama il caricamento

Per personalizzare uno di questi elementi, rimuovere i caratteri a doppia barra relativi ai commenti (//) che precedono il comando nel file property di personalizzazione del pacchetto Loader e modificare i valori predefiniti.

Comando database-connection: E' possibile modificare i parametri del comando database-connection se si desidera cambiare le impostazioni predefinite (si suppone che si stia utilizzando DB2).

```
DBConnectCommand = connect to {0} user {1} using {2};
```

in cui:

0 = nome database
1 = utente database
2 = password utente

Comando load-table del database: E' possibile modificare i parametri del comando load-table del database se si desidera cambiare le impostazioni predefinite.

```
DBLoadTableCommand = load from {0} of del modified by coldel{1}  
chardel{2} insert into {3} ({4});
```

in cui:

0 = nome file
1 = delimitatore colonna
2 = delimitatore carattere
3 = nome tabella
4 = nomi colonna, separati da virgole (,)

Comando database-import: E' possibile modificare i parametri del comando database-import se si desidera cambiare le impostazioni predefinite.

```
DBImportCommand = import from {0} of del modified by coldel{1} chardel{2}  
insert_update in {3} ({4});
```

in cui:

0 = nome file
1 = delimitatore colonna

- 2 = delimitatore carattere
- 3 = nome tabella
- 4 = nomi colonna, separati da virgole (,)

Comando di sistema che richiama il caricamento: E' possibile modificare i parametri del comando di sistema che richiama il caricamento se si desidera cambiare i valori predefiniti. Questo comando esegue gli script di importazione e caricamento nativi generati dal pacchetto Loader.

- 

```
DBLoadCommand = db2c1pex DB2 -z {0} -astvf {1}
```

in cui:

0 = nome file di registrazione

1 = nome file comando

Per DB2 in esecuzione su AIX, ad esempio, il valore della proprietà DBLoadCommand è il seguente:

```
db2 -tvf {1} -z {0}
```



```
DBLoadCommand = sqlldr log={0} control={1} USERID={2}
```

in cui:

0 = nome file di registrazione

1 = nome file comando

2 = nome utente database

Utilizzare le seguenti impostazioni per le combinazioni dei diversi database e sistemi operativi:



Impostare la variabile d'ambiente del sistema classpath per includere db2/dbconnect.zip per DB2 in esecuzione su Windows NT o Windows 2000 con i metodi sqlimport, load, import o delete.



Per DB2 in esecuzione in ambienti AIX, Solaris, o Linux, effettuare le seguenti operazioni:

- Con i metodi sqlimport, load, import o delete, impostare la variabile d'ambiente del sistema classpath per includere db2/dbconnect.zip.
- Con i metodi load o import, modificare le proprietà di seguito riportate nel file property di personalizzazione del pacchetto Loader.

```
/**
 * Connection command. (Default is for DB2)
 * parameter 0 = dbName
 * parameter 1 = dbUser
 * parameter 2 = userPasswd
 */
```

```
DBConnectCommand = connect to {0} user {1} using {2};
```

```
/**
 * Load Data into Table command. (Default is for DB2)
 * parameter 0 = filename
 * parameter 1 = column delimiter
 * parameter 2 = character delimiter
 * parameter 3 = name of the table
 * parameter 4 = name of the columns, separated by comma(,)s
 */
```

```
DBLoadTableCommand = load from {0} of del modified by coldel{1}
insert into {3} ({4});
```

```
/**
 * Insert Data into Table command. (Default is for DB2)
```

```

* parameter 0 = filename
* parameter 1 = column delimiter
* parameter 2 = character delimiter
* parameter 3 = name of the table
* parameter 4 = name of the columns, separated by comma(,)s
*/

DBUpdateTableCommand = import from {0} of del modified by coldel{1}
insert_update in {3} ({4});

/**
 * System command to invoke load (Default is for DB2)
 * parameter 0 = logFileName
 * parameter 1 = commandFileName
 */

DBLoadCommand = db2 -z {0} -tf {1}

```

- 
 Modificare le proprietà di seguito riportate nel file property di personalizzazione del pacchetto Loader per DB2 in esecuzione su iSeries con i metodi sqlimport, load, import o delete:

```

/**
 * The connect string.
 */

ConnectionStringID = jdbc:db2://

/**
 * The JDBC driver information.
 */

JDBCDriverName = com.ibm.db2.jdbc.app.DB2Driver
DbVendorName=DB2/iSeries

/**
 * Custom writer for load/import methods.
 */

WriterName=com.ibm.wca.MassLoader.Writer.ISeriesWriter

```

- 






Impostare la variabile d'ambiente del sistema classpath in modo da includere oracle/dbconnect.zip per il database Oracle in esecuzione in ambiente Windows NT, Windows 2000, AIX, Solaris o Linux con il metodo sqlimport, load, import o delete.

Sostituzione di un componente

E' possibile sostituire un componente del pacchetto Loader assegnando agli elementi di seguito riportati del file property di personalizzazione del pacchetto Loader i valori delle classi che si desidera sostituire per le implementazioni predefinite:

ParserName

Nome del parser da utilizzare

ValidatorName

Nome del validator da utilizzare

FormatterName

Nome del formatter da utilizzare

JDBCFormatterName

Nome del formatter quando viene utilizzato il metodo di importazione SQL

WriterName

Nome del writer da utilizzare

JDBCWriterName

Nome del writer quando viene utilizzato il metodo di importazione SQL

Per sostituire il valore writer predefinito del pacchetto Loader (DefaultWriter) con il valore com.abc.writer.SpecialWriter, ad esempio, specificare quanto segue nel file property di personalizzazione del pacchetto Loader:

```
WriterName = com.abc.writer.SpecialWriter
```

Il pacchetto Loader utilizzerà quindi il valore "com.abc.writer.SpecialWriter" per eseguire le funzioni di scrittura.

Utilizzo della sincronizzazione dello spazio di ricerca Product Advisor

Per utilizzare la sincronizzazione dello spazio di ricerca di Product Advisor, attenersi alla seguente procedura.

1. Creare un file di informazioni di configurazione XML per la sincronizzazione denominato "PASyncInfo.xml".
2. In PASyncInfo.xml, specificare PASync.xsd come schema XML da utilizzare. Ad esempio:

```
<PASync
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation='PASync.xsd'
```

Viene fornito il file PASync.xsd. Il seguente testo mostra il contenuto di PASync.xsd:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd='http://www.w3.org/2001/XMLSchema'>

  <xsd:element name="PASync">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="SearchScheme" />
        <xsd:element ref="Command" minOccurs="1" maxOccurs="unbounded" />
      </xsd:sequence>
      <xsd:attribute name = "member" type="xsd:string" use="required" />
      <xsd:attribute name = "store" type="xsd:string" use="required" />
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="SearchScheme">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="RelatedTable" />
        <xsd:element ref="Search" minOccurs="1" maxOccurs="unbounded" />
      </xsd:sequence>
      <xsd:attribute name = "tableName" type="xsd:string" use="required" />
      <xsd:attribute name = "primary" type="xsd:string" use="required" />
      <xsd:attribute name = "colName" type="xsd:string" use="required" />
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="RelatedTable">
```

```

<xsd:complexType>
  <xsd:attribute name = "tableName" type="xsd:string" use="required" />
  <xsd:attribute name = "from" type="xsd:string" use="required" />
  <xsd:attribute name = "to" type="xsd:string" use="required" />
</xsd:complexType>
</xsd:element>

<xsd:element name="Search">
  <xsd:complexType>
    <xsd:attribute name = "value" type="xsd:string" use="required" />
  </xsd:complexType>
</xsd:element>

<xsd:element name="Command">
  <xsd:complexType>
    <xsd:attribute name = "tableName" type="xsd:string" use="required" />
    <xsd:attribute name = "idColumnName" type="xsd:string" use="required" />
    <xsd:attribute name = "addCommand" type="xsd:string" />
    <xsd:attribute name = "updateCommand" type="xsd:string" />
    <xsd:attribute name = "deleteCommand" type="xsd:string" />
  </xsd:complexType>
</xsd:element>

</xsd:schema>

```

3. In `PASyncInfo.xml`, specificare l'ID membro e l'ID entità negozio per i quali è necessaria la sincronizzazione. Ad esempio:

```

member = "-2000"
store = "10351"

```

4. Nell'elemento schema di ricerca in `PASyncInfo.xml`, specificare gli identificativi CATGROUP che costituiscono lo spazio di ricerca. Ad esempio:

```

<SearchScheme
  tableName = "catgroup"
  primary = "CATGROUP_ID"
  colName = "identifier" >

  <RelatedTable
    tableName = "catgpenrel"
    from = "CATGROUP_ID"
    to = "CATENTRY_ID" />

  <Search value="Pants" />
  <Search value="Shirts" />

</SearchScheme>

```

Nell'esempio vengono specificati "Pants" e "Shirts". E' possibile specificare tutti gli identificativi CATGROUP che si desidera.

5. In `PASyncInfo.xml`, specificare gli attributi per determinare i comandi da programmare. Ad esempio:

```

<Command tableName = "CATENTRY" idColumnName = "CATENTRY_ID"
  updateCommand = "UpdateSearchSpaces"
  deleteCommand = "RemoveProductsFromAllSearchSpaces"
/>

<Command tableName = "CATENTDESC" idColumnName = "CATENTRY_ID"
  addCommand = "UpdateSearchSpaces"
  updateCommand = "UpdateSearchSpaces"
  deleteCommand = "UpdateSearchSpaces"
/>

<Command tableName = "LISTPRICE" idColumnName = "CATENTRY_ID"
  addCommand = "UpdateSearchSpaces"
  updateCommand = "UpdateSearchSpaces"
  deleteCommand = "UpdateSearchSpaces"

```

```

/>

<Command tableName = "ATTRVALUE" idColumnName = "CATENTRY_ID"
  addCommand = "UpdateSearchSpaces"
  updateCommand = "UpdateSearchSpaces"
  deleteCommand = "UpdateSearchSpaces"
/>

<Command tableName = "CATENTATTR" idColumnName = "CATENTRY_ID"
  addCommand = "UpdateSearchSpaces"
  updateCommand = "UpdateSearchSpaces"
  deleteCommand = "UpdateSearchSpaces"
/>

<Command tableName = "CATGPENREL" idColumnName = "CATENTRY_ID"
  addCommand = "AddProductsToSearchSpace"
  deleteCommand = "RemoveProductsFromSearchSpace"
/>

</PASync>

```

6. Nel nuovo file property di personalizzazione del pacchetto Loader, specificare il file delle informazioni di configurazione XML. Ad esempio:

```
PASyncDocumentURL = PASyncInfo.xml
```

7. Nel nuovo file property di personalizzazione del pacchetto Loader, abilitare la sincronizzazione. Ad esempio:

```
PASyncEnabled = true
```

8. Utilizzare il metodo import SQL o il metodo delete con il comando Load. Di seguito viene riportato un esempio di immissione XML appropriato nel pacchetto Loader:

```

<store-asset>

  <catentry
    CATENTRY_ID="10351"
    MEMBER_ID="-2000"
    PARTNUMBER="000051"
    CATENTTYPE_ID="ProductBean"
    MFPARTNUMBER="m000051"
    MARKFORDELETE="0"
    BUYABLE="1"
  />

  <catentry
    CATENTRY_ID="10352"
    MEMBER_ID="-2000"
    PARTNUMBER="000052"
    CATENTTYPE_ID="ProductBean"
    MFPARTNUMBER="m000052"
    MARKFORDELETE="0"
    BUYABLE="1"
  />

  <catentry
    CATENTRY_ID="10353"
    MEMBER_ID="-2000"
    PARTNUMBER="000053"
    CATENTTYPE_ID="ProductBean"
    MFPARTNUMBER="m000053"
    MARKFORDELETE="0"
    BUYABLE="1"
  />

  <catentry
    CATENTRY_ID="10358"
    MEMBER_ID="-2000"

```

```

PARTNUMBER="000058"
CATENTTYPE_ID="ProductBean"
MFPARTNUMBER="m000058"
MARKFORDELETE="0"
BUYABLE="1"
/>

<catentry
  CATENTRY_ID="10365"
  MEMBER_ID="-2000"
  PARTNUMBER="000065"
  CATENTTYPE_ID="ProductBean"
  MFPARTNUMBER="m000065"
  MARKFORDELETE="0"
  BUYABLE="1"
/>

<catentry
  CATENTRY_ID="10372"
  MEMBER_ID="-2000"
  PARTNUMBER="000072"
  CATENTTYPE_ID="ProductBean"
  MFPARTNUMBER="m000072"
  MARKFORDELETE="0"
  BUYABLE="1"
/>

<catgpenrel
  CATGROUP_ID="10354"
  CATENTRY_ID="10372"
  CATALOG_ID="10351"
  SEQUENCE="3"
/>

<catgpenrel
  CATGROUP_ID="10354"
  CATENTRY_ID="10365"
  CATALOG_ID="10351"
  SEQUENCE="4"
/>

<catgpenrel
  CATGROUP_ID="10354"
  CATENTRY_ID="10358"
  CATALOG_ID="10351"
  SEQUENCE="5"
/>

<catgpenrel
  CATGROUP_ID="10355"
  CATENTRY_ID="10372"
  CATALOG_ID="10351"
  SEQUENCE="3"
/>

</store-asset>

```

Nota: La disabilitazione della sincronizzazione dello spazio di ricerca di Product Advisor fornisce migliori prestazioni del pacchetto Loader; quindi, utilizzare tale funzione solo quando è necessario.

Personalizzazione della sincronizzazione dello spazio di ricerca di Product Advisor

Il pacchetto Loader consente di personalizzare la sincronizzazione dello spazio di ricerca di Product Advisor modificando il file property di personalizzazione del pacchetto per effettuare quanto segue:

- **Abilitare o disabilitare la sincronizzazione**

E' possibile abilitare o disabilitare la sincronizzazione specificando true o false come valore per la seguente proprietà nel file property di personalizzazione del pacchetto Loader:

```
PASyncEnabled = true
```

- **Specificare il file delle informazioni di configurazione per la sincronizzazione**

E' possibile specificare quale file delle informazioni di configurazione XML utilizzare impostando il valore per la seguente proprietà nel file property di personalizzazione del pacchetto Loader:

```
PASyncDocumentURL = PASyncInfo.xml
```

- **Specificare la lunghezza dell'interrogazione di pianificazione**

E' possibile specificare la lunghezza dell'interrogazione di pianificazione impostando il valore per la seguente proprietà nel file property di personalizzazione del pacchetto Loader:

```
PAScheduleQueryLength = 30
```

Il valore per questa proprietà deve essere compreso nell'intervallo da 20 a 900.

- **Specificare l'ora di avvio pianificata**

E' possibile definire l'ora di avvio pianificata specificando una data/ora assoluta, una data/ora corrente o una data/ora corrente con durata come valore per la proprietà PAScheduledStartTime nel file property di personalizzazione del pacchetto Loader.

Nota: Il formato data/ora deve essere appropriato per il database.

Di seguito viene riportato un esempio per DB2 che eseguirà il processo pianificato 5 minuti dopo il caricamento:

```
PAScheduledStartTime = CURRENT TIMESTAMP + 5 MINUTES
```



Di seguito viene riportato un esempio per il database Oracle che eseguirà il processo immediatamente:

```
PAScheduledStartTime = SYSDATE
```

Determinazione della modalità di elaborazione dei dati utilizzando il pacchetto Loader

Il pacchetto Loader offre le seguenti opzioni per l'elaborazione dei dati mediante l'utilizzo del comando Load:

- Caricamento
- Importazione
- Utilizzo della funzione di importazione SQL

Prima di caricare i dati, è necessario stabilire quale metodo di elaborazione è in grado di produrre i migliori risultati.

Selezione del metodo load

Considerare questo metodo nelle seguenti situazioni:

- Se si è certi che i dati siano puliti e se il database non contiene dati
- Se si è certi che i dati siano puliti e se il database non contiene i dati che devono essere caricati

- Se si è certi che i dati siano puliti, se le tabelle di destinazione non contengono chiavi primarie e se il database non contiene i dati che devono essere caricati
- Se il tempo di caricamento è l'obiettivo primario
- Se il database è un database DB2 locale

400

Con il metodo load, i dati vengono caricati nel database. Se i dati già esistono, il comando non viene eseguito a causa di un errore nella chiave duplicata e pertanto viene visualizzato un messaggio di errore di duplicazione.

Selezione del metodo import

NT 2000 AIX Solaris Linux

Con il metodo import per DB2, i dati vengono caricati nel database. Se i dati sono già presenti, questi non verranno eliminati, ma verranno aggiornati con i nuovi valori. Prendere in considerazione questo metodo nelle seguenti situazioni:

- Se il sistema di gestione del database è DB2
- Se non si è certi che i dati siano puliti
- Se è necessario aggiornare serie estese di dati omogenei a livello di colonna
- Se il tempo di caricamento non è l'obiettivo primario
- Se la tabella in cui vengono importati i dati dispone di una chiave principale

400

Anche con il metodo import i dati vengono caricati nel database. Se i dati sono già presenti, questi non verranno eliminati, ma verranno aggiornati con i nuovi valori. Prendere in considerazione questo metodo nelle seguenti situazioni:

- Se non si è certi che i dati siano puliti
- Se i dati sono già presenti nel database
- Se il tempo di caricamento non è l'obiettivo primario
- Se la tabella in cui vengono importati i dati dispone di una chiave principale

Selezione del metodo import SQL

Con il metodo import SQL, le istruzioni JDBC o SQL vengono utilizzate per aggiornare o inserire i dati nel database. I dati vengono inseriti se non sono già presenti e i dati esistenti vengono aggiornati. Prendere in considerazione questo metodo nelle seguenti situazioni:

- Se si stanno aggiornando dati esistenti e si richiedono aggiornamenti al livello di colonna

Con questo metodo ci sono prospettive di errore sulle violazioni del limite e sugli errori tipo-dati migliori.

- Se si è certi che i dati non siano puliti
- Se l'integrità dei dati è l'obiettivo primario
- Se il database non è locale
- Se si sta utilizzando la sincronizzazione dello spazio di ricerca Product Advisor

Altre considerazioni

- **Restrizioni sull'utilizzo del metodo load**

Con il metodo load non è possibile inserire o aggiornare dati nei campi di dati in bit.

DB2 Con il metodo load, vengono inseriti nel database soltanto i record nuovi; i record esistenti non vengono aggiornati.

DB2 E' possibile utilizzare il metodo load solo per i database DB2, locali e non remoti.

- **Restrizioni sull'utilizzo del metodo import**

Il sistema di gestione del database deve essere DB2 per poter utilizzare il metodo import.

Con il metodo import non è possibile inserire o aggiornare dati nei campi di dati in bit.

Con il metodo import il Loader inserisce o aggiorna soltanto le tabelle che dispongono di una chiave principale definita; il metodo import non consente di inserire o aggiornare dati in tabelle che non dispongono di una chiave principale. Se soltanto il record di immissione dispone di valori primari per le colonne, il record viene rifiutato.

- **Confronto dei metodi import SQL e load**

Il metodo import SQL verifica la consistenza dei dati, compresi i riferimenti esterni e consente di aggiornare i dati esistenti. Il metodo load non lo consente.

- **Confronto tra i metodi import ed import SQL**

I metodi import ed import SQL eseguono funzioni analoghe. Il metodo import è in genere più veloce, ma richiede spazio sul disco per i file temporanei.

Il metodo import, al contrario del metodo import SQL, può soltanto inserire o aggiornare le tabelle che dispongono di una chiave principale definita.

- **Confronto tra i metodi basati sul prodotto del database utilizzato**

I metodi import e load utilizzano programmi di utilità nativi ottimizzati per DB2, mentre il metodo import SQL utilizza le chiamate JDBC (che sono generiche per molti prodotti del database).

- **Ulteriori considerazioni**

Il metodo delete viene utilizzato per eliminare i dati presenti nel documento XML di immissione dal database. Tale elemento deve contenere i valori per la chiave principale e l'indice univoco per la tabella. Se i dati da eliminare hanno una dipendenza di dati in altre tabelle con "cascade o delete" abilitati, vengono eliminati anche i dati dipendenti.

Se si sta utilizzando la sincronizzazione dello spazio di ricerca di Product Advisor, è necessario utilizzare il metodo import SQL per il caricamento dei dati.

Caricamento di documenti di grandi dimensioni

Quando si utilizzano i programmi di utilità del pacchetto Loader per caricare documenti di grandi dimensioni in un database, considerare le seguenti opzioni:

- **Dimensioni della memoria riservata JVM (Java Virtual Machine)**

Per impostazione predefinita, la quantità massima di memoria assegnata a JVM è 64 MB. Se non viene aumentato tale valore, JVM potrebbe non avere sufficiente memoria a disposizione durante il processo di caricamento. La quantità massima di memoria assegnata a Java può essere modificata utilizzando l'opzione JVM -mx nel comando Java.

- **Registrazione traccia**

Il programma di registrazione della traccia può esaurire la memoria assegnata a JVM durante il caricamento di un documento XML di grandi dimensioni. Le informazioni della traccia vengono utilizzate principalmente per la correzione degli errori di un processo di esecuzione se questo non riesce correttamente. Se

la traccia del processo di caricamento non è necessaria, è opportuno disattivarla. Esiste una prestazione significativa che si ottiene quando la traccia viene disattivata. La traccia viene disattivata modificando il documento XML di configurazione della registrazione.

Il file di configurazione della registrazione predefinito è WCALoggerConfig.xml. Per disattivare la registrazione di traccia, modificare la configurazione del programma di registrazione della traccia per il pacchetto Loader da:

```
<logger type="trace">
  <handler type="file">
    <filePath>MassLoadTrace.log</filePath>
    <filter type="Any">
      <messageType name="PUBLIC" />
    </filter>
  </handler>
</logger>
```

in:

```
<logger type="trace">
  <handler type="file">
    <filePath>MassLoadTrace.log</filePath>
    <filter type="Any">
      </filter>
  </handler>
</logger>
```

Per ulteriori informazioni sulla modifica del file WCALoggerConfig.xml, fare riferimento alla sezione "Personalizzazione della registrazione per il pacchetto Loader" a pagina 70.

- **Conteggio del commit**

Il conteggio del commit predefinito per il pacchetto Loader quando è operativo in modalità di importazione SQL è 1. Pertanto, per impostazione predefinita, le transazioni vengono eseguite per ogni aggiornamento o inserimento nel database. Per migliorare le prestazioni del pacchetto Loader per i documenti di grandi dimensioni, è necessario aumentare il conteggio del commit. Si consiglia un valore pari a 100; tuttavia è possibile aumentarlo a seconda della quantità di memoria fisica sul server, della dimensione del file di registrazione della transazione DBMS e così via.

E' possibile modificare il conteggio del commit per il pacchetto Loader utilizzando l'opzione *conteggio -commitcount* per il comando Load (dove *conteggio* è il numero delle istruzioni eseguite prima della transazione).

Consigli per la risoluzione dei problemi

Se l'avanzamento è particolarmente lento durante il caricamento dei dati, potrebbe essere presente un handler di file configurato non correttamente nel programma di registrazione del pacchetto Loader. Ciò potrebbe derivare da una delle seguenti situazioni:

- L'utente che ha richiamato il pacchetto Loader non dispone dell'autorizzazione per la scrittura nella directory o per aggiornare il file specificato nel documento relativo alla configurazione della registrazione.
- La directory di ubicazione del file specificata nel documento relativo alla configurazione della registrazione non esiste.
- L'unità specificata come ubicazione del file nel documento di configurazione della registrazione non dispone di sufficiente spazio.

Una volta risolti questi problemi, potrebbe essere necessario cambiare l'ubicazione del file specificata modificando il documento relativo alla configurazione della registrazione (per impostazione predefinita `WCALoggerConfig.xml`). Per ulteriori informazioni sugli handler di file e sul file `WCALoggerConfig.xml`, fare riferimento alla sezione "Personalizzazione della registrazione per il pacchetto Loader" a pagina 70.

Capitolo 7. Estrazione di dati

Per estrarre dei dati da un database mediante Extractor, è necessario specificare i dati che si desidera estrarre dal database utilizzando un file filtro di estrazione. Il filtro di estrazione che si utilizza dipende dal tipo di dati che si desidera estrarre.

Nel seguente esempio vengono estratti i dati del sottosistema membro da un database utilizzando MemberSubsystemFilter.xml come filtro di estrazione.

- ▶ NT ▶ 2000 ▶ AIX ▶ Solaris ▶ Linux
- java com.ibm.wca.MassExtract.Extract -filter MemberSubsystemFilter.xml
-outfile MemberSubsystemExtracted.xml -dbname mall -dbuser myname
-dbpwd mypassword -customizer MemberSubsystemCustomizer
- ▶ 400
- QWEBCOMM/EXTWCSDTA FILTER(MemberSubsystemFilter.xml)
OUTFILE(MemberSubsystemExtracted.xml) DATABASE(database_name)
SCHEMA(mall) INSTRROOT(/QIBM/UserData/WebCommerce/instances/mser)
PASSWD(mypassword)

Creazione di un filtro di estrazione

Il seguente esempio di filtro di estrazione consente di estrarre informazioni su prodotti e categorie dalle tabelle CATGROUP, CATGRPDESC, CATGRPREL, CATENTRY, CATENTSHIP, OFFER, CATENTREL, CATGPENREL, CATENTDESC e ATTRVALUE:

```
<sqlx>

<!-- ***** -->
<!-- extract Category information -->
<!-- ***** -->

<functionDef id="Category" description="Extract Categories" schemaentity="catgroup">
  <paramDef name=":lastRecord" type="string" value="10301" description="Last record
  before loading new data" />
  <body>
    select * from catgroup where catgroup_id > :lastRecord
  </body>
</functionDef>

<execute id="Category" description="Extract Categories" schemaentity="catgroup">
  <param name=":lastRecord" type="string" value="10300" description="Last record
  before loading new data" />
</execute>

<functionDef id="Category Description" description="Extract Category Descriptions
for a Locale" schemaentity="catgrpdesc">
  <paramDef name=":lastRecord" type="string" value="10300" description="Last record
  before loading new data" />
  <body>
    select * from catgrpdesc where catgroup_id > :lastRecord
  </body>
</functionDef>

<execute id="Category Description" description="Extract Category Descriptions
for a Locale" schemaentity="catgrpdesc">
  <param name=":lastRecord" type="string" value="10300" description="Last record
  before loading new data" />
</execute>

<functionDef id="Category Relationship" description="Extract Category-Relations
```

```

for a Locale" schemaentity="catgrprel">
  <paramDef name=":lastRecord" type="string" value="10300" description="Last record
  before loading new data" />
  <body>
    select * from catgrprel where catgroup_id_child > :lastRecord
  </body>
</functionDef>

<execute id="Category Relationship" description="Extract Category-Relations for
a Locale" schemaentity="catgrprel">
  <param name=":lastRecord" type="string" value="10300" description="Last record
  before loading new data" />
</execute>

<!-- ***** -->
<!-- extract Product information -->
<!-- ***** -->

<functionDef id="Product" description="Extract Product" schemaentity="catentry">
  <paramDef name=":lastrecord" type="string" value="10300" description="Last record
  before loading new data" />
  <body>
    select * from catentry where catentry_id > :lastrecord
  </body>
</functionDef>

<execute id="Product" description="Extract Product" schemaentity="catentry">
  <param name=":lastrecord" type="string" value="10300" description="Last record
  before loading new data" />
</execute>

<functionDef id="Product Relationship" description="Extract Product Ship
information" schemaentity="catentrel">
  <paramDef name=":lastrecord" type="string" value="10300" description="Last record
  before loading new data" />
  <body>
    select * from catentrel where catentry_id_child > :lastrecord
  </body>
</functionDef>

<execute id="Product Relationship" description="Extract Product Ship information"
schemaentity="catentrel">
  <param name=":lastrecord" type="string" value="10300" description="Last record
  before loading new data" />
</execute>

<functionDef id="Product Description" description="Extract Product Description"
schemaentity="catentdesc">
  <paramDef name=":lastrecord" type="string" value="10300" description="Last record
  before loading new data" />
  <body>
    select * from catentdesc where catentry_id > :lastrecord
  </body>
</functionDef>

<execute id="Product Description" description="Extract Product Description"
schemaentity="catentdesc">
  <param name=":lastrecord" type="string" value="10300" description="Last record
  before loading new data" />
</execute>

<functionDef id="Product Ship" description="Extract Product Ship information"
schemaentity="catentship">
  <paramDef name=":lastrecord" type="string" value="10300" description="Last record
  before loading new data" />
  <body>
    select * from catentship where catentry_id > :lastrecord
  </body>
</functionDef>

<execute id="Product Ship" description="Extract Product Ship information"
schemaentity="catentship">
  <param name=":lastrecord" type="string" value="10300" description="Last record

```

```

    before loading new data" />
</execute>

<functionDef id="Category Product Relationship" description="Extract Category
Product Relations" schemaentity="catgpenrel">
  <paramDef name=":lastrecord" type="string" value="10300" description="Last record
before loading new data" />
  <body>
    select * from catgpenrel where catgroup_id > :lastrecord
  </body>
</functionDef>

<execute id="Category Product Relationship" description="Extract Category Product
Relations" schemaentity="catgpenrel">
  <param name=":lastrecord" type="string" value="10300" description="Last record
before loading new data" />
</execute>

<!-- ***** -->
<!-- Extract Product Attribute Information -->
<!-- ***** -->

<functionDef id="Product Attribute Values" description="Extract Product Attribute
values for a Locale" schemaentity="attrvalue">
  <paramDef name=":lastrecord" type="string" value="10300" description="Last record
before loading new data" />
  <body>
    select * from attrvalue where catentry_id > :lastrecord
  </body>
</functionDef>

<execute id="Product Attribute Values" description="Extract Product Attribute values
for a Locale" schemaentity="attrvalue">
  <param name=":lastrecord" type="string" value="10300" description="Last record
before loading new data" />
</execute>

<!-- ***** -->
<!-- Extract Product Price Information -->
<!-- ***** -->

<functionDef id="Offer" description="Extract Offer" schemaentity="offer">
  <paramDef name=":lastrecord" type="string" value="10300" description="Last record
before loading new data" />
  <body>
    select * from offer where catentry_id > :lastrecord
  </body>
</functionDef>

<execute id="Offer" description="Extract Offer" schemaentity="offer">
  <param name=":lastrecord" type="string" value="10300" description="Last record
before loading new data" />
</execute>

</sqlx>

```

Impostazione di Extractor

E' possibile modificare i driver del database utilizzati da Extractor effettuando le seguenti operazioni:

1. Creare un nuovo file property di personalizzazione di Extractor.



Il file `DB2ConnectionCustomizer.properties` si trova nell'archivio `MassExtract.zip`. Estrarre questo file, rinominarlo lasciando l'estensione `.properties` e posizionarlo in una directory nel classpath. **Importante:** Non rimuovere o modificare il file `DB2ConnectionCustomizer.properties` esistente.



`ISeries_EXTWCSDTA_Customizer.properties` risiede nella directory `/QIBM/ProdData/WebCommerce/properties`. Copiare questo file nella directory `/rootist/xml`, rinominare il nuovo file mantenendo l'estensione `.properties`, quindi apportare tutte le modifiche necessarie al nuovo file. **Importante:** Non rimuovere o modificare il file `ISeries_EXTWCSDTA_Customizer.properties` originale.

2. Modificare i valori del driver del database nel nuovo file.
3. Specificare il nuovo nome file come valore del parametro di personalizzazione del comando `Extract`.

Di seguito viene riportato un esempio del file property di personalizzazione di Extractor:

```
DBVendorName = DB2
DBDriverName = COM.ibm.db2.jdbc.app.DB2Driver
DBURL = jdbc:db2:
```

in cui:

- `DBVendorName` viene utilizzato per selezionare il tipo di database.
Le opzioni sono le seguenti:
 - DB2 Universal Database per iSeries (DB2/iSeries)
 - DB2 per altri sistemi operativi (DB2)
 - Database Oracle (Oracle)
- `DBDriverName` viene utilizzato per selezionare il driver JDBC.
Le opzioni sono le seguenti:
 - DB2 Universal Database per iSeries (`com.ibm.db2.jdbc.app.DB2Driver`)
 - DB2 per altri sistemi operativi (`COM.ibm.db2.jdbc.app.DB2Driver`)
 - Database Oracle (`oracle.jdbc.driver.OracleDriver`)
- `DBURL` viene utilizzato per specificare l'URL per accedere al database.
Le opzioni sono le seguenti:
 - DB2 Universal Database per iSeries (`jdbc:db2://`)
 - DB2 per altri sistemi operativi (`jdbc:db2:`)
 - Database Oracle (`jdbc:oracle:oci8:@`)

Capitolo 8. Utilizzo del programma di registrazione del pacchetto Loader

Ciascun programma di utilità del pacchetto Loader crea dei messaggi per indicare operazioni eseguite correttamente, operazioni non riuscite ed eventuali errori, oltre a fornire informazioni di traccia sul programma.

I programmi di utilità del pacchetto Loader fanno riferimento al file `WCALoggerConfig.xml`.



Questo file si trova in una directory specificata nella variabile di ambiente del sistema classpath. Può anche essere specificata dalla proprietà dei sistemi Java `com.ibm.wca.logging.configFile`.



Questo file risiede nella directory `/rootist/xml`.

`WCALoggerConfig.xml` determina quali informazioni di registrazione ciascun programma di utilità fornisce e dove vengono visualizzate o memorizzate le informazioni. E' possibile personalizzare questo file e specificare quali tipi di file di registrazione sono stati creati oltre a quali tipi di messaggi sono stati registrati.

Configurazione del collegamento nell'ambiente per sistemi Windows NT, Windows 2000, AIX, Linux e Solaris

Per impostare la registrazione nell'ambiente utilizzato, è necessario impostare la variabile classpath sistema-ambiente per includere il file `WCALoggerConfig.xml` o specificare la proprietà di sistema `com.ibm.wca.logging.configFile`.

Esempio di impostazione della variabile classpath

Se il file `WCALoggerConfig.xml` si trova nella directory `d:\WebSphere\CommerceServer\xml\loader` su una macchina Windows NT, ad esempio, è possibile utilizzare l'istruzione di seguito riportata per impostare la variabile classpath:

```
SET CLASSPATH=%CLASSPATH%;D:\WebSphere\CommerceServer\xml\loader
```

Esempio di specificazione della proprietà di sistema `com.ibm.wca.logging.configFile`

Per specificare la proprietà di sistema `com.ibm.wca.logging.configFile`, utilizzare l'opzione `-D` quando si richiama l'interprete Java. Di seguito è riportato un esempio:

```
java -Dcom.ibm.wca.logging.configFile=D:\ice_tea\src\classlib\logger\xml\WC.xml  
com.ibm.wca.DTDGenerator.GeneratedDTD
```

Personalizzazione della registrazione per il pacchetto Loader

Per personalizzare la registrazione per il pacchetto Loader, utilizzare il file WCALoggerConfig.xml.

-     

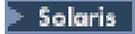
Questo file si trova in una directory specificata nella variabile di ambiente del sistema classpath. Può anche essere specificata dalla proprietà dei sistemi Java `com.ibm.wca.logging.configFile`.

- 

Questo file risiede nella directory `/rootist/xml`.

WCALoggerConfig.xml contiene una o più tag del componente, ad esempio `<component name="DTDGenerator">`. All'interno di tali tag, è possibile aggiungere programmi di registrazione e handler. *Non* è possibile modificare le tag del programma di utilità e di registrazione fornite con il sistema, ma è possibile aggiungere le tag dell'handler al programma di registrazione. Per informazioni relative a cosa è possibile includere in questo file, fare riferimento al file WCALogger.dtd.

I file di registrazione del pacchetto Loader si trovano nel file `messages.txt` nelle seguenti directory:

-  `unità:\WebSphere\CommerceServer\instances\nome_istanza\logs`
-  `unità:\Program Files\WebSphere\CommerceServer\instances\nome_istanza\logs`
-  `/usr/WebSphere/CommerceServer/instances/nome_istanza/logs`
-   `/opt/WebSphere/CommerceServer/instances/nome_istanza/logs`
-  `/QIBM/UserData/WebCommerce/instances/nome_istanza/logs`

Handler

Per aggiungere un handler ad un programma di registrazione, specificare il tipo di handler nel file WCALoggerConfig.xml. E' possibile aggiungere più di un handler a un unico programma di registrazione. Ciascun handler dispone di attributi propri e di tag subordinate che non sono necessariamente valide per altri handler. Tra i tipi di handler vi sono:

Tipo di handler	Descrizione e attributi
console	Invia messaggi all'emissione standard, in genere la riga comandi
file	Memorizza i messaggi in un file di testo E' necessario aggiungere "<filePath>log path</filePath>" a questo handler come tag subordinata.
multifile	Crea una registrazione circolare di file E' necessario specificare "<filePath>log path</filePath>". Vengono creati i file di registrazione da 1 a <i>n</i> . E' possibile aggiungere i seguenti attributi: MaxFiles Un numero intero che indica quanti file utilizzare prima di cancellare il primo file di registrazione MaxKBFileSize Un numero intero che indica il numero massimo di kilobyte da archiviare in ciascun file di registrazione
 NT  2000  AIX  Solaris  Linux database	Archivia i messaggi in una tabella DB2 in una registrazione circolare. E' possibile aggiungere i seguenti attributi: brand Nome del database. DB2 è il solo database supportato al momento. maxRows Il numero massimo di record da archiviare in una tabella prima di cancellare le voci più vecchie E' possibile includere "<jdbc/>" come tag subordinata e includere i seguenti attributi: url L'URL utilizzato in JDBC per accedere a un database (ad esempio,"Jdbc:db2:" <i>wcm</i> """, dove " <i>wcm</i> " è il nome del database). Il database deve essere presente prima di eseguire il programma di utilità. table Il nome della tabella del database in cui verranno registrati i messaggi. E' necessario che venga creato con la seguente istruzione DB2: "CREATE TABLE" <i>nometabella</i> (KEY char(13) FOR BIT DATA NOT NULL, COMPONENTNAME VARCHAR(30), ENTRY VARCHAR(2000), PRIMARY KEY(key))" userid Il nome utente del database. All'utente devono essere assegnate le autorizzazioni ad aggiornare la tabella. L'istruzione DB2 di seguito riportata esegue questa azione: "GRANT SELECT, UPDATE, INSERT, DELETE ON TABLE" <i>nometabella</i> TO USER <i>idutente</i> "" password La password del database per il nome utente specificato

Nel seguente esempio viene aggiunto un handler di tipo "database" a un programma di registrazione:

```

<handler type="database" brand="DB2" maxRows="50">
  <jdbc url="jdbc:db2:wcm"
    table="wcm.log"
    userid="wasuser"
    password="123456"/>
  <filter type="Any">
    <messageType name="FATAL"/>
    <messageType name="ERROR"/>
    <messageType name="WARNING"/>
  </filter>
</handler>

```

Filtri

I filtri possono essere aggiunti o rimossi dagli handler per includere ed escludere tipi di messaggi. Se un programma di registrazione non ha filtri, nessun messaggio verrà registrato. Ogni tag dei filtri contiene una tag subordinata messageType in cui viene elencato il tipo di messaggi, di solito uno dei seguenti:

- INFO
- ERROR
- FATAL
- WARNING

Altri tipi di messaggi vengono elencati nel file WCALogger.dtd, ma la maggior parte non vengono utilizzati con questo pacchetto.

I tipi di filtro comprendono:

Tipo di filtro	Descrizione e attributi
Any	Include nel file di registrazione tutti i messaggi indicati come uno dei tipi specificati. Ad esempio, se l'elenco di messageType include ERROR e l'applicazione genera un messaggio di tipo ERROR, il messaggio viene registrato.
Tutti	Richiede che un messaggio contenga tutti gli attributi del tipo messageType specificato da inserire in una registrazione
Exclude	Registra tutti i messaggi non specificati nell'elenco delle tag messageType.

Il seguente esempio di aggiunta di un filtro a un handler illustra come i tipi di messaggio FATAL e ERROR vengono registrati, mentre tutti gli altri messaggi vengono ignorati:

```

<handler type="file">
  <filter type="Any">
    <messageType name="FATAL"/>
    <messageType name="ERROR"/>
  </filter>
</handler>

```

Formati

E' possibile specificare uno tra i due tipi di programmi per la formattazione dei messaggi:

Tipo di programma di formattazione	Descrizione e attributi
safe (default)	Evita che venga impostata una eccezione se un messaggio non viene trovato in un file properties Questo programma di formattazione crea un messaggio che indica che la risorsa è mancante.
xml	Formatta il messaggio in formato XML Se non è possibile trovare un messaggio, il programma di formattazione scrive un messaggio anziché impostare un'eccezione.

Esempio: WCALoggerConfig.xml e WCALogger.dtd

WCALoggerConfig.xml

```

<?xml version="1.0" standalone="no"?>
<!DOCTYPE WCALoggerConfig SYSTEM "WCALogger.dtd">
<WCALoggerConfig>
<component name="MassLoader">
  <logger type="message">
    <handler type="file">
      <filePath>c:\temp\out.txt</filePath>
      <filter type="Any">
        <messageType name="FATAL"/>
        <messageType name="ERROR"/>
        <messageType name="WARNING"/>
        <messageType name="INFO"/>
      </filter>
    </handler>
  </logger>
  <logger type="trace">
    <handler type="file">
      <filePath>out2.txt</filePath>
      <filter type="Any">
        <messageType name="PUBLIC"/>
      </filter>
    </handler>
  </logger>
  <logger type="typedMessage">
    <handler type="file">
      <filePath>tout.txt</filePath>
      <filter type="Any">
        <messageType name="FATAL"/>
        <messageType name="ERROR"/>
        <messageType name="WARNING"/>
        <messageType name="INFO"/>
      </filter>
    </handler>
  </logger>
  <logger type="progress">
    <handler type="console" format="safe">
      <filter type="Any">
        <messageType name="FATAL"/>
        <messageType name="ERROR"/>
        <messageType name="WARNING"/>
        <messageType name="INFO"/>
      </filter>
    </handler>
  </logger>
</component>
<component name="DTDGenerator">

```

```

<logger type="message">
  <handler type="console">
    <filter type="Any">
      <messageType name="FATAL"/>
      <messageType name="ERROR"/>
      <messageType name="WARNING"/>
      <messageType name="INFO"/>
    </filter>
  </handler>
</logger>
<logger type="trace">
  <handler type="console">
    <filter type="Any">
      <messageType name="FATAL"/>
      <messageType name="ERROR"/>
      <messageType name="WARNING"/>
      <messageType name="INFO"/>
    </filter>
  </handler>
</logger>
</component>
</WCALoggerConfig>

```

WCALogger.dtd

```

<!-- This DTD describes how a WCALoggerConfig XML can be structured.
A WCALoggerConfig XML document is the input configuration file for
the WCALoggerFactory class.
-->

<!ELEMENT WCALoggerConfig (component)+>

<!ELEMENT component (logger)+>
<!ATTLIST component name CDATA #REQUIRED>
<!ELEMENT logger (handler+,messageFile?)>
<!ATTLIST logger type (message | trace | typedMessage | progress) "typedMessage">

<!-- messageFile is an optional default properties files that can be used to
make messages locale specific
-->
<!ELEMENT messageFile (#PCDATA)>
<!ELEMENT handler (filePath?, filter, jdbc?)>
<!ATTLIST handler
type ( file|multiFile|console|error|textArea|database|ejbQueue|queue ) "console">

<!-- maxFiles & maxKBFileSize only applies to the multiFile type of handler
-->
<!-- filePath & encoding applies only when the handler is of type file or
multiFile
-->
<!ATTLIST handler maxFiles CDATA #IMPLIED>
<!ATTLIST handler maxKBFileSize CDATA #IMPLIED>
<!ATTLIST handler encoding CDATA #IMPLIED>
<!ATTLIST handler format (safe | xml) "safe">
<!-- maxRecords & brand are only applicable to database handler type
-->
<!ATTLIST handler maxRecords CDATA #IMPLIED>
<!ATTLIST handler brand (DB2) #IMPLIED>
<!-- the jdbc tag must be present within a database handler type tag
-->
<!ELEMENT jdbc EMPTY>
<!ATTLIST jdbc url CDATA #IMPLIED>
<!ATTLIST jdbc table CDATA #IMPLIED>
<!ATTLIST jdbc userid CDATA #IMPLIED>
<!ATTLIST jdbc password CDATA #IMPLIED>

<!ELEMENT filter (messageType)+>

```

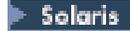
```
<!ATTLIST filter type (Any | All | Exclude ) "Any">

<!-- the messageType attribute name is one of these JLog IRecordType
JLog IRecordType
-->
<!ELEMENT messageType EMPTY>
<!ATTLIST messageType name ( NONE | ALL | INFO |
INFORMATION | WARN | WARNING | ERR | ERROR |
FATAL | DEFAULT_MESSAGE | API | CALLBACK |
ENTRY_EXIT | ENTRY | EXIT | ERROR_EXC |
MISC_DATA | OBJ_CREATE | OBJ_DELETE |
PRIVATE | PUBLIC | STATIC | SVC | PERF |
LEVEL1 | LEVEL2 | LEVEL3 ) "ALL">
<!ELEMENT filePath (#PCDATA)>
```

Capitolo 9. Utilizzo del programma di segnalazione degli errori del pacchetto Loader

Loader e ID Resolver includono un programma di segnalazione degli errori che genera un documento di eccezione se si verifica un errore.

Per impostazione predefinita, il documento di eccezione viene scritto nella seguente directory:

-      la directory in cui risiede il documento di immissione
-  /rootist/logs

Per specificare la directory in cui è scritto il documento di eccezione, utilizzare la proprietà Java `com.ibm.wcm.ErrorReporterDir`. Viene di seguito riportato un esempio di inizio di Loader in ambiente Windows NT:

```
java -Dcom.ibm.wcm.ErrorReporterDir=d:\massloaderrors  
com.ibm.wca.MassLoader.MassLoad -dbname . . .
```

Nota: L'utente deve avere autorizzazione alla scrittura per la directory specificata.

Di seguito viene riportato un DTD di esempio (`store-all-error.dtd`) per il programma di segnalazione degli errori:

```
<!ENTITY % TABLE "calrule | catentry">  
<!ELEMENT store-asset (error, (%TABLE;)*)>  
<!ELEMENT message (#PCDATA) >  
<!ELEMENT error ( message ) >  
<!ATTLIST error  
  locus          CDATA    #REQUIRED  
  id             CDATA    #REQUIRED  
>  
<!ELEMENT calrule (error)>  
<!ATTLIST calrule  
  identifier      CDATA    #REQUIRED  
  calrule_id     CDATA    #REQUIRED  
  calcode_id     CDATA    #REQUIRED  
  startdate      CDATA    #IMPLIED  
  taxcgry_id     CDATA    #IMPLIED  
  enddate        CDATA    #IMPLIED  
  sequence       CDATA    #REQUIRED  
  combination    CDATA    #REQUIRED  
  calmethod_id   CDATA    #REQUIRED  
  calmethod_id_qfy CDATA    #REQUIRED  
  flags          CDATA    #REQUIRED  
  field1         CDATA    #IMPLIED  
  field2         CDATA    #IMPLIED  
>  
<!ELEMENT catentry (error)>  
<!ATTLIST catentry  
  catentry_id    CDATA    #REQUIRED  
  member_id     CDATA    #REQUIRED  
  catenttype_id  CDATA    #REQUIRED  
  partnumber     CDATA    #IMPLIED  
  mfpnumber     CDATA    #IMPLIED  
  mfname        CDATA    #IMPLIED  
  markfordelete CDATA    #REQUIRED  
  url           CDATA    #IMPLIED
```

```

field1          CDATA    #IMPLIED
field2          CDATA    #IMPLIED
lastupdate     CDATA    #IMPLIED
field3          CDATA    #IMPLIED
onspecial      CDATA    #IMPLIED
onauction      CDATA    #IMPLIED
field4          CDATA    #IMPLIED
field5          CDATA    #IMPLIED
buyable        CDATA    #IMPLIED
>

```

Di seguito viene riportato un documento di segnalazione errori del Loader:

```

<?xml version="1.0"?>
<!DOCTYPE store-asset SYSTEM "store-all-error.dtd">
<store-asset>
  <error
    locus="Parser"
    id="SAXParseFatalError" >
    <message>
      Error The string "--" is not permitted within comments. : 155 : 18
    </message>
  </error>
  <calrule
    calcode_id="30"
    enddate="2100-01 10:20:30.000000"
    calmethod_id="-47"
    identifier="7"
    taxcgry_id="9"
    calmethod_id_qfy="-46"
    startdate="1900-01-01-00.00.00.000000"
    flags="1"
    combination="2"
    calrule_id="44"
    sequence="9.0E+1">
    <error
      locus="Writer"
      id="SQLException" >
      <message>
        A SQL Exception was received [IBM][CLI Driver][DB2/NT] SQL0530N
        The insert or update value of the FOREIGN KEY
        "JANTONY.CALRULE.F_CALRULE4" is not equal to any value of the
        parent key of the parent table. SQLSTATE=23503
      </message>
    </error>
  </calrule>
  <catentry
    catentry_id="10118"
    member_id="-2001"
    partnumber="1254"
    mfpartment="sku-163"
    mfname="InFashion"
    markfordelete="0"
    buyable="1"
    field1="abc" >
    <error
      locus="Formatter"
      id="FormattingError" >
      <message>
        Error when formatting value for CATENTRY.FIELD1 : abc with error
        [class java.lang.NumberFormatException(abc)].
      </message>
    </error>
  </catentry>
</store-asset>

```

Capitolo 10. Configurazione degli script e dei comandi del pacchetto Loader

Per avviare il pacchetto Loader ed eseguire i relativi comandi, utilizzare gli script o i comandi forniti nella directory di WebSphere Commerce:

-  *unità*:\WebSphere\CommerceServer\bin
-  *unità*:\Program Files\WebSphere\CommerceServer\bin
-  /usr/WebSphere/CommerceServer/bin
-   /opt/WebSphere/CommerceServer/bin
-  QWEBCOMM native library

Gli script e i comandi sono riportati di seguito:



GENWCSDTD

Comando DTD Generate

RESWCSID

Comando ID Resolve

EXTWCSDTA

Comando Extract

LODWCSDTA

Comando Load

TRNWCSTXT

Comando Text Transform

TRNWCXML

Comando XML Transform



dtdgen.cmd

Comando DTD Generate

idresgen.cmd

Comando ID Resolve

massextract.cmd

Comando Extract

massload.cmd

Comando Load

txttransform.cmd

Comando Text Transform

xmltransform.cmd

Comando XML Transform



- dtdgen.sh**
Script shell DTD Generate
- idresgen.sh**
Script shell ID Resolve
- massextract.sh**
Script shell Extract
- massload.sh**
Script shell Load
- txttransform.sh**
Script shell Text Transform
- xmltransform.sh**
Script shell XML Transform

Parte 3. Utilizzo di Web editor

Questa sezione descrive le modalità di gestione e utilizzo di Catalog Manager Web editor.

Web editor consente di creare, eliminare e apportare modifiche ai dati di catalogo mediante un browser Web. I formati dei dati per la visualizzazione e l'aggiornamento delle informazioni sono fondamentali in Web editor. Nei casi più semplici, i formati corrispondono alle tabelle del database di WebSphere Commerce. Il responsabile può scegliere di utilizzare i formati predefiniti o di personalizzare i formati disponibili.

Nota: Web editor utilizza Internet Explorer 5 e versioni successive.

Capitolo 11. Impostazione di Web editor

Il responsabile può configurare Web editor in modo da fornire dati a WebSphere Commerce, incluse estensioni e personalizzazioni per il relativo schema. Web editor non è stato progettato come un insieme di formati Web specifici di una particolare istanza di WebSphere Commerce. E' stata invece progettato per essere flessibile e personalizzabile in modo da supportare le singole necessità ed i singoli ruoli delle diverse organizzazioni.

Una visualizzazione di database è un'interrogazione memorizzata in una o più tabelle del database. Durante l'installazione di Catalog Manager, un file di esempio per la creazione di una vista logica di un prodotto WebSphere Commerce viene inserito in ciascuna sottodirectory specifica del sistema (db2, oracle e os400) della seguente directory:

-  *unità:*\WebSphere\CommerceServer\schema
-  *unità:*\Program Files\WebSphere\CommerceServer\schema
-  /usr/WebSphere/CommerceServer/schema
-   /opt/WebSphere/CommerceServer/schema
-  /QIBM/ProdData/WebCommerce/schema

Questo file di esempio, *wcs.view.sql*, contiene una vista di prodotto creata per combinare le informazioni sui prodotti ricavate da più tabelle. Il file contiene la definizione dei dati SQL della visualizzazione del prodotto. I responsabili possono studiare questo file per pianificare lo sviluppo delle visualizzazioni di un database.

Il file XML predefinito di descrizione dei formati (*forms51_be.xml*) contiene i formati per semplificare l'aggiunta, la modifica, e l'eliminazione dei dati in un database di WebSphere Commerce. Una copia di questo file si trova nella seguente directory:

-  *unità:*\WebSphere\CommerceServer\xml\wcwebeditor\xml
-  *unità:*\Program Files\WebSphere\CommerceServer\xml\wcwebeditor\xml
-  /usr/WebSphere/CommerceServer/xml/wcwebeditor/xml
-   /opt/WebSphere/CommerceServer/xml/wcwebeditor/xml
-  /rootist/xml/wcwebeditor/xml

Questo file di configurazione può essere utilizzato senza alcuna variazione oppure può essere modificato e migliorato dal responsabile dell'impostazione di Web editor. Per personalizzare il file XML di descrizione dei formati, fare riferimento alle istruzioni presenti nella parti successive di questa sezione.

Nota: DTD Generator può creare automaticamente i formati utilizzati da Web editor.

Configurazione di Web editor

Questa sezione fornisce informazioni sulla configurazione di Web editor. Nonostante il processo di installazione gestisca le fasi iniziali della configurazione, un responsabile può utilizzare queste informazioni per eseguire attività quali riconfigurare Web editor per utilizzare un altro database.

Modifica del file `webeditor.properties`

Web editor presenta alcuni parametri di applicazione impostati nel file `webeditor.properties`. Tale file si trova nella seguente directory:

- ▶ **NT** `unità:\WebSphere\AppServer\installedApps\WC_Enterprise_App_demo.ear\wcwebeditor.war\WEB-INF\classes\webeditor.properties`
- ▶ **2000** `unità:\Program Files\WebSphere\AppServer\installedApps\WC_Enterprise_App_demo.ear\wcwebeditor.war\WEB-INF\classes\webeditor.properties`
- ▶ **AIX** `/usr/WebSphere/AppServer/installedApps/WC_Enterprise_App_demo.ear/wcwebeditor.war/WEB-INF/classes/webeditor.properties`
- ▶ **Solaris** ▶ **Linux** `/opt/WebSphere/AppServer/installedApps/WC_Enterprise_App_demo.ear/wcwebeditor.war/WEB-INF/classes/webeditor.properties`
- ▶ **400** `/QIBM/UserData/WEBASADV4/nome_istanza_was/installedApps/WC_Enterprise_App_nome_istanza_wcs.ear/wcwebeditor.war/WEN-INF/classes`

Modificando il file `webeditor.properties`, un responsabile può eseguire attività quali la modifica del file utilizzato per descrivere i formati visualizzati in Web editor. Di seguito viene riportato un esempio del contenuto di un file `webeditor.properties`:

```
# Properties file for WebEditor

(The following specifies where the customized process list and the Catalog Manager
utility configuration envelopes are located.)
# URI Location of Process and WCM Subsystem configuration envelopes
ProcessConfigFile=file:///D:/WebSphere/CommerceServer/xml/wcwebeditor/xml/weProcessList.xml

(The following specifies where the forms are defined.)
# Location of Forms file
FormsURL=file:///D:/WebSphere/CommerceServer/xml/wcwebeditor/xml/forms51_be.xml

(The following setting specifies where the XML-to-HTML style sheet is located)
# Location of XML to HTML StyleSheet
StyleSheetURI=file:///D:/WebSphere/CommerceServer/xml/wcwebeditor/xsl/webeditor.xsl

(The following setting specifies the location for temporary files)
# Location of Temporary Directory
#temp.dir=

# WebSphere datasource. This is used to build form drop-down lists and field
# default values. However, when publishing and searching, the Web Editor utilizes
# the WCM subsystems ID Resolver, Mass Loader, and Mass Extractor. Database access
# for these WCM subsystems must be configured separately.
# Name of WAS database source
dbsource=jdbc/WebSphere Commerce DB2 DataSource demo

# Name of WAS database. If specified, this takes preference over the database name
# value in the forms XML file. This value will be utilized as a parameter when
# invoking the WCM subsystems such as the Mass Loader.
```

```
dbname=all
```

(The following setting specifies the character set to use. By default, it is set to the single-byte character set. For other countries, choose from the values that are commented out.)

```
# Encoding Character Set
Encoding=ISO-8859-1
```

```
# CN
#Encoding=gb2312
# TW
#Encoding=Big5
# KR
#Encoding=EUC-KR
# JP
#Encoding=Shift_JIS
```

(The following setting is used to allow images to be previewed. The hostname of the server plus any leading directory information before the image information that is stored in WebSphere Commerce should be specified here. The WebSphere Commerce catalog information is appended to this value to construct an image URL.)

```
# Specifies the base href location for images
# This should be set so that this info plus the info stored in WCS
# (e.g. /image/char.gif) combines to create a URL to an image
```

```
#imageRootURL=http://%HOSTDOMAIN
imageRootURL=http://localhost/webeditor
```

(The following setting is used to set a date format for the application. If these properties are not available (i.e., they are commented out), then the Java-locale specific format will be used. Below is a reference table for setting these values.)

```
# Use these properties to specify a date format if the Java locale-specific
# format is not desired
```

```
#dateFormat=yyyy-MM-dd
#dateTimeFormat=yyyy-MM-dd HH:mm:ss
```

# Symbol	Meaning	Presentation	Example
# -----	-----	-----	-----
# G	era designator	(Text)	AD
# y	year	(Number)	1996
# M	month in year	(Text & Number)	July & 07
# d	day in month	(Number)	10
# h	hour in am/pm	(1~12) (Number)	12
# H	hour in day	(0~23) (Number)	0
# m	minute in hour	(Number)	30
# s	second in minute	(Number)	55
# S	millisecond	(Number)	978
# E	day in week	(Text)	Tuesday
# D	day in year	(Number)	189
# F	day of week in month	(Number)	2 (2nd Wed in July)
# w	week in year	(Number)	27
# W	week in month	(Number)	2
# a	am/pm marker	(Text)	PM
# k	hour in day	(1~24) (Number)	24
# K	hour in am/pm	(0~11) (Number)	0
# z	time zone	(Text)	Pacific Standard Time
# '	escape for text	(Delimiter)	'
# ''	single quote	(Literal)	'

Modifica dell'ubicazione dei file temporanei

L'ubicazione dei file temporanei può essere modificata rimuovendo il segno di commento e aggiungendo un valore alla proprietà temp.dir nel file webeditor.properties.

In alternativa, per stabilire l'ubicazione in cui vengono creati i file temporanei viene utilizzata la proprietà Java java.io.tmpdir.

Creazione di un file di descrizione moduli XML mediante DTD Generator

Il file `forms51_be.xml` è un esempio di file di descrizione moduli XML. Tale file fornisce un insieme di moduli utilizzati da Web editor. Una copia di questo file si trova nella seguente directory:

-  `unità:\WebSphere\CommerceServer\xml\wcwebeditor\xml`
-  `unità:\Program Files\WebSphere\CommerceServer\xml\wcwebeditor\xml`
-  `/usr/WebSphere/CommerceServer/xml/wcwebeditor/xml`
-   `/opt/WebSphere/CommerceServer/xml/wcwebeditor/xml`
-  `/rootist/xml/wcwebeditor/xml`

I passi di seguito riportati descrivono il modo in cui il responsabile di sistema può utilizzare DTD Generator per aggiungere nuovi moduli XML.

Nota: Prima di eseguire la procedura descritta di seguito, rinominare il file `forms51_be.xml` esistente nella directory di Web editor. In alternativa, seguendo la procedura descritta di seguito, creare un file di emissione con un nuovo nome e riconfigurare Web editor per utilizzare il file creato. Per le istruzioni, fare riferimento alla sezione precedente.

Per creare moduli XML, eseguire il comando DTD Generate.

I passi riportati di seguito descrivono come creare moduli XML:

1. Creare un file temporaneo denominato "tables.txt" contenente i nomi delle tabelle che si desidera utilizzare nei moduli.

Inserire ciascun nome di tabella su un'unica riga, come nel seguente esempio:

```
catentry
catentdesc
catentship
inventory
```

2. Salvare il file `tables.txt` nella directory in cui si trova il comando DTD Generate. (Per l'ubicazione di questo comando, consultare il Capitolo 10, "Configurazione degli script e dei comandi del pacchetto Loader" a pagina 79.)
3. Al prompt di comandi del sistema operativo, passare alla directory in cui si trova il comando DTD Generate.

4. Eseguire il comando DTD, immettendo quanto segue:

▶ NT ▶ 2000

```
dtdgen -infile tables.txt -outfile tables51.dtd
-dbname nomedb -dbuser idutente -dbpwd password
-xmlTableDesc tableFORMS.xml -schemaname schema -propfile nomefile
```

▶ AIX ▶ Solaris ▶ Linux

```
./dtdgen.sh -infile tables.txt -outfile tables51.dtd
-dbname nomedb -dbuser idutente -dbpwd password
-xmlTableDesc tableFORMS.xml -schemaname schema -propfile nomefile
```

▶ 400

```
QWEECOMM/GENWCSDTD DATABASE(database) SCHEMA(schema)
INSTROOT(rootist) PASSWD(password) OUTFILE(tables51.dtd)
INFILE(tables.txt) XMLTABDESC(tableFORMS.xml)
```

Il passaggio alla descrizione della tabella (-xmlTableDesc o XMLTABDESC) comporta la creazione da parte di DTD Generator di una descrizione delle tabelle in un nuovo formato, oltre a un DTD.

▶ NT ▶ 2000 ▶ AIX ▶ Solaris ▶ Linux

- L'opzione -propFile specifica il nome del file properties esterno in cui è possibile memorizzare le informazioni sulla descrizione dei campi, il testo di aiuto e i valori predefiniti.
5. Riconfigurare Web editor per utilizzare i file creati come descritto nella sezione precedente.
 6. Riavviare Web editor nella Console di gestione di WebSphere Advanced. Per eseguire questa operazione, completare i seguenti passi:
 - a. Espandere **WebSphere Administrative Domain**.
 - b. Espandere **Enterprise Applications**.
 - c. Fare clic con il pulsante destro del mouse su **WebSphere Commerce Enterprise Application - demo** e selezionare **Arresta**.
 - d. Attendere che il messaggio indichi che l'applicazione è terminata.
 - e. Fare clic con il pulsante destro del mouse su **WebSphere Commerce Enterprise Application - demo** e selezionare **Avvia**.
 - f. Attendere che il messaggio indichi che l'applicazione enterprise è stata avviata.
 7. Per visualizzare il nuovo modulo in un browser Web, aprire il seguente URL:

```
https://nome_host:8000/wcm/webeditor
```

dove *nome_host* è il nome host HTTP completo di WebSphere Application Server.

Web editor viene visualizzato nella finestra del browser con un elenco di tutti i nomi di tabelle.

Personalizzazione della descrizione dei moduli XML

Questa sezione descrive le modalità con cui il responsabile può migliorare i moduli visualizzati da Web editor.

La descrizione dei moduli XML può essere personalizzata e migliorata impostando attributi e valori nel file di descrizione dei moduli XML o in un file properties diverso.

Nota: Il nome del file properties deve essere specificato come valore dell'attributo resourcePackage nella tag formList. Se il nome del file viene visualizzato in una sottodirectory di una directory del classpath, deve utilizzare la specifica package(dot).

Nella tabella di seguito riportata vengono elencati gli attributi dei campi relativi ai moduli che possono essere modificati da un responsabile.

Attributo campo	Descrizione
Currency	I valori vengono visualizzati con un separatore numerico specifico della locale (ad esempio una virgola come separatore delle migliaia per gli Stati Uniti)
DbColumn	Utilizzato per associare il nome del campo alla chiave del file properties Se vengono utilizzati i file properties specifici della locale, il valore di questa voce deve corrispondere al valore immesso nel file properties. DTD aggiunge uno schema a questa voce.
DefaultValue	Specifica un valore visualizzato nel formato dati quando un utente completa un nuovo modulo. Questo attributo può essere impostato da DTD Generator sul valore predefinito del database. Può essere una stringa statica, ma può anche includere un'interrogazione SQL scalare con un tabella di un'unica riga. Per richiamare i dati esterni, può essere utilizzata nell'interrogazione una funzione definita dall'utente. Ad esempio: DefaultValue="SELECT CURRENT TIMETAMP FROM EXEC" dove EXEC è una funzione definita e inserita nel seguente modo: CREATE TABLE EXEC (A CHAR(1)); INSERT INTO A VALUES('A');
dynamicSqlSelectionList	Un menu a discesa viene ricreato per ciascun modulo

Attributo campo	Descrizione
FieldDescription	<p>Fornisce una descrizione visualizzata accanto al campo di immissione del modulo</p> <p>Quando si crea questo attributo, DTD Generator utilizza i commenti nella colonna, se esistenti. Se non esistono commenti nella colonna, il valore predefinito è il nome della colonna.</p> <p>Questo attributo può essere impostato nel file property specifico della locale o in un file di descrizione del modulo XML. Se nel file properties è specificato un valore, questo valore ha la precedenza.</p>

Attributo campo	Descrizione
FieldHelp	<p>Fornisce una breve descrizione di aiuto relativa al campo, visualizzata in basso nella barra dei messaggi del browser quando il campo è attivo nel modulo.</p> <p>Per impostazione predefinita, contiene un semplice messaggio per inserire i dati relativi ad un determinato campo insieme al tipo di colonna.</p> <p>Questo attributo può essere impostato nel file property specifico della locale o in un file di descrizione del modulo XML. Se nel file properties è specificato un valore, questo valore ha la precedenza.</p>
formatNumber	<p>Utilizzato per indicare che Web editor non elabora un numero in nessun modo</p> <p>Impostare l'attributo formatNumber su "false" per considerare il valore immesso come una stringa, eccetto durante le interrogazioni di estrazione (dove il valore non viene inserito tra virgolette come nel caso di una stringa). Il valore predefinito di questo attributo è "false."</p>
Hidden	<p>Indica che il valore non viene visualizzato nel modulo ma è ancora disponibile come un campo HTML nascosto</p>
HideOnCreate	<p>Indica che il campo è disponibile quando viene composto un nuovo modulo</p> <p>Simile a showInCreateMode="false"; ma aggiunge il nome del campo come carattere nascosto.</p>
Maxlength	<p>Specifica la lunghezza della colonna del database</p> <p>Viene utilizzato per garantire che l'utente non immetta un valore di una lunghezza maggiore di quella memorizzata nel database.</p>
MinOccurs	<p>Indica se il campo è obbligatorio</p> <p>Un valore pari a "1" indica che il campo è obbligatorio, un valore pari a "0" indica che il campo è facoltativo.</p>
Name	<p>Specifica il nome della colonna del database.</p>
readOnly readOnlyForCreate readOnlyForEdit	<p>Controlla quando un campo è disponibile per la modifica e quando è disponibile per la lettura da parte dell'utente</p> <p>readOnly="true" indica che un campo è sempre nella modalità solo lettura. readOnlyForCreate="true" indica che un campo è nella modalità solo lettura quando viene composto un nuovo modulo. readOnlyForEdit="true" indica che un campo è nella modalità solo lettura quando un modulo esistente viene modificato.</p>

Attributo campo	Descrizione
ShowColumnInList	<p data-bbox="862 237 1438 323">Specifica che il campo è rappresentato da una delle colonne che costituiscono la visualizzazione di dati a più record quando impostate su "true"</p> <p data-bbox="862 348 1360 405">DTD Generator imposta su "true" le prime sei colonne.</p>

Attributo campo	Descrizione
showInCreateMode	<p>Utilizzato per nascondere un campo del modulo creato</p> <p>Impostare l'attributo showInCreateMode su "false" per nascondere il campo nel modulo creato visualizzato. Il valore predefinito di questo attributo è "false."</p>
ShowInSearchMode	<p>Utilizzato per nascondere alcune colonne della pagina dei criteri di ricerca</p> <p>Se questo attributo è impostato su "false," il campo non viene visualizzato nel modulo di ricerca.</p>
SqlSelectionList	<p>Crea un menu a discesa da cui un utente può eseguire una selezione durante la composizione di un nuovo modulo</p> <p>L'interrogazione deve restituire un insieme di risultati a una o due colonne. La prima colonna è rappresentata dall'elenco di etichette da cui l'utente può eseguire la selezione; la seconda colonna è rappresentata dall'elenco dei valori effettivi da memorizzare nella tabella del database. Se viene utilizzata una sola colonna, il valore visualizzato viene memorizzato nella tabella. Questa funzione è molto utile per rafforzare le relazioni delle chiavi esterne.</p>
Type	<p>Indica il tipo di colonna del database e i controlli di convalida da eseguire sui dati</p> <p>Ad esempio, se il tipo è "numero intero," l'applicazione garantisce che solo un numero intero valido venga immesso in questo campo.</p>
UniqueKey	<p>Indica che il campo è rappresentato da una chiave principale per una tabella</p> <p>Se questo attributo è impostato su "true," l'applicazione rafforza un vincolo univoco sui dati immessi in questa colonna. Questa verifica viene eseguita solo sui record caricati nell'applicazione, se non si estende ai record del database.</p>
ValidateInput	<p>Utilizzato per disattivare la verifica della convalida</p> <p>Questo attributo consente all'utente di inserire il testo in un campo numerico che verrà gestito da ID Resolver o XML Transformer.</p>

Modifica dei nomi dei moduli

La tag del modulo nella descrizione del modulo XML presenta gli attributi name e displayName.

- L'attributo name deve essere impostato sul nome delle tabella del database o della visualizzazione rappresentata.
- L'attributo displayName fornisce un nome semplice da utilizzare nell'applicazione Web editor. In alternativa, un attributo displayName relativo a

un modulo può essere impostato nel file properties specificato nell'attributo resourcePackage della tagformList. L'attributo resourcePackage viene utilizzato per il supporto di più lingue.

Nota: Se nel file properties è presente una voce, questa voce ha la precedenza sulla voce immessa nel file di descrizione del modulo XML.

Modifica della descrizione di un campo

Di seguito viene riportato un esempio di descrizione di un campo in un file di descrizione del modulo XML:

```
<form name = "CATALOG.CATENTRY"  
.  
.  
<field name="MEMBER_ID"  
  showInCreateMode="false"  
  fieldDescription="MEMBER_ID"  
  type="integer"  
  maxLength="19"  
  defaultValue=""  
.  
.  
<field name="CATENTTYPE_ID"  
  fieldDescription="CATENTTYPE_ID"  
  type="string"  
  maxLength="16"  
  defaultValue=""  
.  
.
```

E' possibile modificare il file come segue:

```
<form name = "CATALOG.CATENTRY"  
  displayName="Product"  
.  
.  
<field name="MEMBER_ID"  
  showInCreateMode="false"  
  fieldDescription="Member Identifier"  
  type="integer"  
  maxLength="19"  
  defaultValue=""  
.  
.  
<field name="CATENTTYPE_ID"  
  fieldDescription="Product Type"  
  type="string"  
  maxLength="16"  
  defaultValue=""  
.  
.
```

Aggiunta di un menu a discesa

E' possibile aggiungere al modulo di Web Editor un elenco di selezione sotto forma di un menu a discesa per semplificare il completamento del modulo e per limitare la selezione dell'utente a un insieme di scelte valido.

Per aggiornare gli elenchi di selezione, il responsabile può riavviare Web editor. Esiste anche un attributo di descrizione del modulo, `dynamicSqlSelectionList`, che il responsabile può impostare su "true" per fare in modo che il menu a discesa venga caricato ogni volta.

L'interrogazione per un elenco di selezione può restituire un insieme di risultati a una o due colonne. Se il risultato restituisce due colonne, la seconda colonna contiene i valori effettivi memorizzati e la prima colonna contiene le etichette dell'utente.

Un elenco di selezione viene creato inserendo un'interrogazione SQL nell'attributo sqlSelectionList della tag del campo o creando un'enumerazione nel file di descrizione del modulo XML. Entrambi i metodi sono visualizzati nell'esempio di seguito riportato:

```
<field name="MEMBER_ID"
  showInCreateMode="false"
  fieldDescription="Member Identifier"
  .
  .
  readOnly="false"
  sqlSelectionList="select orgentityname,orgentity.orgentity_id
  from member,orgentity where member.type='0'
  and member.member_id=orgentity.orgentity_id"
  .
  .
  fieldHelp=""
  .
  .
  </field>
  <field name="MARKFORDELETE"
    fieldDescription="Mark for Delete"
    type="NMTOKEN"
    sqlSelectionList=""
  >
  <datatype source="integer">
    <enumeration label="No" value="0"/>
    <enumeration label="Yes" value="1"/>
  </datatype>
  </field>
>
```

Nelle tabelle del database di WebSphere Commerce esistono campi che utilizzano 1 e 0 per true e false. Per creare campi che siano più comprensibili per l'utente, possono essere stabilite delle enumerazioni. Catalog Manager fornisce uno script SQL per creare una tabella di enumerazione denominata "NUMDESC." Questo script è denominato "createEnum.sql." Un responsabile può modificare l'attributo sqlSelectionList per creare un elenco di selezione mediante la tabella ENUMDESC come indicato nell'esempio di seguito riportato:

```
fieldDescription="On Special"
.
.
readOnly="false"
sqlSelectionList="select description,value from
enumdesc where columnname='ALL' and type='YESNO'"
fieldHelp=""
.
.
```

Aggiunta di un aiuto sui campi

L'aiuto sui campi viene visualizzato nella parte inferiore della finestra del browser Web per il campo attualmente attivo. Per impostare questo valore, può essere utilizzato l'attributo fieldHelp della tag del campo presente nel file di descrizione del modulo XML oppure la chiave fieldHelp del file properties.

Ad esempio, il file properties può contenere le specifiche di aiuto sui campi di seguito riportate:

```
CATEGORY.MARKFORDELETE.defaultValue=No
CATEGORY.MARKFORDELETE.fieldDescription=Delete Entry
```

Se non è specificato alcun valore, viene creato un messaggio predefinito "Immettere un valore per *nome_campo* qui *tipo_campo*".

Nota: Se esiste un file properties contenente informazioni sul modulo, tale file ha la precedenza sulle voci presenti nel file di descrizione del modulo XML.

Personalizzazione dei risultati della ricerca e dell'elenco delle sessioni di lavoro

E' possibile personalizzare l'insieme di colonne visualizzato nella pagina dei risultati della ricerca. Esiste un attributo `showColumnInList` della tag del campo presente nel file di descrizione del modulo XML. Per inserire un campo tra i risultati della ricerca, impostare questo attributo su "true"; in caso contrario, questo attributo non verrà visualizzato come una delle colonne nella visualizzazione dei risultati della ricerca. Il campo è presente quando il modulo viene visualizzato.

Modifica del file `weProcessList`

Il file `weProcessList` consente al responsabile di personalizzare i programmi di utilità di Catalog Manager eseguiti quando viene elaborata una sessione di lavoro di Web editor.

-       Il file `weProcessList.xml` ubicato nella seguente directory:
 -  `unità:\WebSphere\CommerceServer\xml\wcwebeditor\xml`
 -  `unità:\Program Files\WebSphere\CommerceServer\xml\wcwebeditor\xml`
 -  `/usr/WebSphere/CommerceServer/xml/wcwebeditor/xml`
 -   `/opt/WebSphere/CommerceServer/xml/wcwebeditor/xml`
-       Il file `weProcessListOracle.xml` file ubicato nella seguente directory:
 -  `unità:\WebSphere\CommerceServer\xml\wcwebeditor\xml`
 -  `unità:\Program Files\WebSphere\CommerceServer\xml\wcwebeditor\xml`
 -  `/usr/WebSphere/CommerceServer/xml/wcwebeditor/xml`
 -   `/opt/WebSphere/CommerceServer/xml/wcwebeditor/xml`
-  Il file `weProcessListAS400.xml` risiede nella seguente directory:
 - `/rootist/xml/wcwebeditor/xml`

Questo file contiene i modelli di buste per i vari programmi di utilità. Può inoltre contenere riferimenti ad applicazioni personalizzate che il responsabile desidera eseguire.

In questo file è possibile utilizzare un insieme di variabili di sistema. La variabile di sistema `%-dbname%`, ad esempio, fa in modo che il nome del database venga inserito nella busta creata richiamando un programma di utilità, ad esempio Loader. Il file di descrizione del modulo XML contiene riferimenti a questi processi che indicano quali processi richiamare per l'aggiunta, la modifica o l'eliminazione.

Di seguito viene riportato l'esempio di un file weProcessList.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<processSet>
  <!-- Do not change name of extract -->
  <process name="extract"
    subsystem="com.ibm.wca.MassExtract.extract.ExtractSubSystem">
    <envelope-input xmlns='saf_params.xsd'>
      <param name="-dbname" type="scalar" value="%-dbname%"/>
      <param name="-dbuser" type="scalar" value="%-dbuser%"/>
      <param name="-dbpwd" type="scalar" value="%-dbpwd%"/>
      <param name="-outfile" type="file" reside="local" value="%tempFilePath%"/>
      <param name="-filter" type="file" reside="local" value="%tempFileURI%"/>
    </envelope-input>
  </process>
  <process name="transformer"
    subsystem="com.ibm.wca.XMLTransformer.XMLTransformerSubSystem">
    <envelope-input xmlns='saf_params.xsd'>
      <param name="-infile" type="file" reside="local" value="%tempFileURI%"/>
      <param name="-transform" type="file" reside="local"
        value="%webEditorDir%/xsl/ViewsToWCS51.XSL"/>
      <param name="-outfile" type="file" reside="local" value="%tempFilePath1%"/>
      <param name="-param" value="root=%-dbname%"/>
      <param name="-param" value="dtdname=%-dtdname%"/>
    </envelope-input>
  </process>
  <process name="transformerForDelete"
    subsystem="com.ibm.wca.XMLTransformer.XMLTransformerSubSystem">
    <envelope-input xmlns='saf_params.xsd'>
      <param name="-infile" type="file" reside="local" value="%tempFileURI%"/>
      <param name="-transform" type="file" reside="local"
        value="%webEditorDir%/xsl/ViewsToWCS51.XSL"/>
      <param name="-outfile" type="file" reside="local" value="%tempFilePath1%"/>
      <param name="-param" value="root=%-dbname%"/>
      <param name="-param" value="dtdname=%-dtdname%"/>
      <param name="-param" value="forDelete=true"/>
    </envelope-input>
  </process>
  <process name="resolver"
    subsystem="com.ibm.wca.IdResGen.IdResGenSubSystem">
    <envelope-input xmlns='saf_params.xsd'>
      <param name="-dbname" type="scalar" value="%-dbname%"/>
      <param name="-dbuser" type="scalar" value="%-dbuser%"/>
      <param name="-dbpwd" type="scalar" value="%-dbpwd%"/>
      <param name="-infile" type="file" reside="local"
        value="%previousOutFileAsURI%"/>
      <param name="-outfile" type="file" reside="local"
        value="%tempFilePath2%"/>
      <param name="-propfile" type="file" reside="local"
        value="propertyFiles.IdKeys"/>
      <param name="-method" type="scalar" value="mixed"/>
    </envelope-input>
  </process>
  <!-- Resolver as first process -->
  <process name="resolverFirstProcess"
    subsystem="com.ibm.wca.IdResGen.IdResGenSubSystem">
    <envelope-input xmlns='saf_params.xsd'>
      <param name="-dbname" type="scalar" value="%-dbname%"/>
      <param name="-dbuser" type="scalar" value="%-dbuser%"/>
      <param name="-dbpwd" type="scalar" value="%-dbpwd%"/>
      <param name="-infile" type="file" reside="local" value="%tempFileURI%"/>
      <param name="-outfile" type="file" reside="local" value="%tempFilePath2%"/>
      <param name="-propfile" type="file" reside="local" value="propertyFiles.IdKeys"/>
      <param name="-method" type="scalar" value="mixed"/>
    </envelope-input>
  </process>
  <process name="loader"
    subsystem="com.ibm.wca.MassLoader.MassLoadSubSystem">
    <envelope-input xmlns='saf_params.xsd'>
      <param name="-dbname" type="scalar" value="%-dbname%"/>
      <param name="-dbuser" type="scalar" value="%-dbuser%"/>
      <param name="-dbpwd" type="scalar" value="%-dbpwd%"/>
      <param name="-commitcount" type="scalar" value="1000"/>
      <param name="-infile" type="file" reside="local" value="%previousOutFileAsURI%"/>
      <param name="-method" type="scalar" value="sqlimport"/>
    </envelope-input>
  </process>
</processSet>
```

```

        <param name="-nopriamry" type="scalar" value="insert"/>
    </envelope-input>
</process>
<process name="loaderFirstProcess"
  subsystem="com.ibm.wca.MassLoader.MassLoadSubSystem">
  <envelope-input xmlns='saf_params.xsd'>
    <param name="-dbname" type="scalar" value="%-dbname%"/>
    <param name="-dbuser" type="scalar" value="%-dbuser%"/>
    <param name="-dbpwd" type="scalar" value="%-dbpwd%"/>
    <param name="-commitcount" type="scalar" value="1000"/>
    <param name="-infile" type="file" reside="local" value="%tempFileURI%"/>
    <param name="-method" type="scalar" value="sqlimport"/>
    <param name="-nopriamry" type="scalar" value="insert"/>
  </envelope-input>
</process>
<process name="loaderForDelete"
  subsystem="com.ibm.wca.MassLoader.MassLoadSubSystem">
  <envelope-input xmlns='saf_params.xsd'>
    <param name="-dbname" type="scalar" value="%-dbname%"/>
    <param name="-dbuser" type="scalar" value="%-dbuser%"/>
    <param name="-dbpwd" type="scalar" value="%-dbpwd%"/>
    <param name="-commitcount" type="scalar" value="1000"/>
    <param name="-infile" type="file" reside="local" value="%previousOutFileAsURI%"/>
    <param name="-delete" type="scalar" value=""/>
  </envelope-input>
</process>
<process name="loaderForDeleteFirstProcess"
  subsystem="com.ibm.wca.MassLoader.MassLoadSubSystem">
  <envelope-input xmlns='saf_params.xsd'>
    <param name="-dbname" type="scalar" value="%-dbname%"/>
    <param name="-dbuser" type="scalar" value="%-dbuser%"/>
    <param name="-dbpwd" type="scalar" value="%-dbpwd%"/>
    <param name="-commitcount" type="scalar" value="1000"/>
    <param name="-infile" type="file" reside="local" value="%tempFileURI%"/>
    <param name="-delete" type="scalar" value=""/>
  </envelope-input>
</process>
<process name="saveToFile"
  cmd="cmd.exe /c c:\temp\theBatchFile.bat"
  args="-infile %tempFilePath% -dbname %-dbname%"
/>
</processSet>

```

Nota: I riferimenti ai file sono sensibili al maiuscolo/minuscolo.

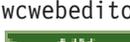
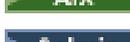
La tabella di seguito riportata contiene un elenco di variabili di sostituzione valide accettate dall'applicazione.

Variabile di sostituzione %	Restituzione
%-dbname%	Il nome del database corrente
%-dbuser%	Il nome utente del database
%-dtdname%	L'ubicazione URI del DTD per i file XML
%-dbpwd%	La password del nome utente del database
%tempFilePath%	Il percorso completo al file temporaneo
%tempFilePath1%	Si tratta di nomi di file temporanei univoci. Tali nomi possono essere inseriti nella sintassi del modello di una definizione di busta o della riga comandi. Ad esempio, se %tempFilePath% è inserito nell'attributo del valore del parametro -infile di un modello di busta, Web editor scrive i dati della sessione di lavoro nell'ubicazione del file temporaneo.
%tempFilePath2%	

On SubSystem	URI temporaneo
%tempFileURI%	Gli URI temporanei sono gli URI dello stesso file rappresentato da %tempFilePath%...%tempFilePath2%. Non si tratta di un insieme supplementare di file ma di un modo per richiamare lo stesso file temporaneo restituito con una sintassi diversa.
%tempFileURI1%	
%tempFileURI2%	
%previousOutFileAsURI%	Fornisce una rappresentazione URI del parametro -outfile delle attività precedenti
%webEditorDir%	L'ubicazione di installazione di Web editor

Modifica del file `weditor.xml`

Web editor utilizza il file `weditor.xml` come foglio di stile XSL predefinito. Tale file si trova nella seguente directory:

-  `unità:\WebSphere\CommerceServer\xml\weditor.xml`
-  `unità:\Program Files\WebSphere\CommerceServer\xml\weditor.xml`
-  `/usr/WebSphere/CommerceServer/xml/weditor.xml`
-   `/opt/WebSphere/CommerceServer/xml/weditor.xml`
-  `/rootist/xml/weditor.xml`

Modificando il file `weditor.xml`, un responsabile può modificare il formato di emissione di Web editor. Di seguito viene riportato un esempio del contenuto di un file `weditor.xml`:

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
<xsl:output method="html"/>
  <!-- Largest single line entry field value for fields larger than this a
  TEXTAREA is created -->
  <xsl:variable name="maxEntryFieldSize" select="80"/>
  <xsl:template match="/">
    <xsl:apply-templates/>
  </xsl:template>
  <!-- Read Only field -->
  <xsl:template match="readOnly" name="readOnly">
    <xsl:element name="input">
      <xsl:attribute name="name"><xsl:value-of select="@name"/></xsl:attribute>
      <xsl:attribute name="type">hidden</xsl:attribute>
      <xsl:attribute name="value"><xsl:value-of select="@defaultValue"/></xsl:attribute>
      <!--
      <xsl:attribute name="onFocus">this.blur()</xsl:attribute>
      <xsl:attribute name="style">border-style:groove</xsl:attribute>
      -->
    </xsl:element>
    <table border="1" cellpadding="0" cellspacing="0" width="155" bgcolor="#C0C0C0">
      <tr>
        <td>
          <xsl:value-of select="@defaultValue"/>
        </td>
      </tr>
    </table>
  </xsl:template>
</xsl:stylesheet>
```

Capitolo 12. Gestione dei cataloghi

Le procedure di seguito elencate descrivono come gestire i cataloghi utilizzando Web editor per aggiungere, modificare ed eliminare i record presenti nelle tabelle di database contenenti dati di catalogo.

Aggiunta di un record a una tabella mediante Web editor

Per aggiungere un record a una tabella mediante Web editor, attenersi alla seguente procedura:

1. Da un browser Web, aprire il seguente URL:

`https://nome_host:8000/wcm/webeditor`

dove *nome_host* è il nome host HTTP completo di WebSphere Application Server.

Viene visualizzata la finestra di collegamento al database Web editor.

2. Immettere la password e il nome utente del database, quindi fare clic su **Collegamento**.

Web editor viene visualizzato nella finestra del browser con un elenco di nomi di tabelle nella barra dei menu di sinistra.

3. Fare clic sul collegamento ipertestuale appropriato nel sottomenu Aggiungi della barra dei menu di sinistra.

Viene visualizzato il modulo appropriato.

4. Inserire nel modulo tutti i dati necessari.
5. Fare clic su **Vai alla sessione di lavoro**.

Vengono visualizzati i risultati della sessione di lavoro relativi al modulo.

Questi risultati contengono tutte le aggiunte, le modifiche e le eliminazioni effettuate ma non cancellate o elaborate durante questa sessione di lavoro di Web editor.

6. Se si desidera rimuovere le modifiche ai record dalla sessione di lavoro, selezionare la casella corrispondente a ciascuna modifica che si desidera rimuovere e fare clic su **Elimina selezionati**.
7. Fare clic su **Elabora sessione di lavoro** per inoltrare le modifiche selezionate al database.

In una pagina di status viene visualizzato un messaggio che indica che il processo è riuscito.

8. Spostarsi sul sito Web, fare clic sui collegamenti ipertestuali appropriati e verificare che le modifiche siano state eseguite.

Modifica di un record in a una tabella mediante Web editor

Per modificare un record di una tabella mediante Web editor, attenersi alla seguente procedura:

1. Da un browser Web, aprire il seguente URL:

`https://nome_host:8000/wcm/webeditor`

dove *nome_host* è il nome host HTTP completo di WebSphere Application Server.

Viene visualizzata la finestra di collegamento al database Web editor.

2. Immettere la password e il nome utente del database, quindi fare clic su **Collegamento**.

Web editor viene visualizzato nella finestra del browser con un elenco di nomi di tabelle nella barra dei menu di sinistra.

3. Fare clic sul collegamento ipertestuale appropriato nel sottomenu Ricerca della barra dei menu di sinistra.

Viene visualizzata la pagina di ricerca appropriata.

4. Specificare i criteri di ricerca attenendosi alla seguente procedura:
 - a. Selezionare la casella di spunta accanto a ciascun attributo che si desidera specificare nella ricerca.
 - b. Utilizzare il menu a discesa appropriato per ciascun attributo selezionato per scegliere la logica che si desidera utilizzare nella ricerca.
 - c. Nel campo accanto a ciascun attributo selezionato, inserire o selezionare il valore che si desidera utilizzare nella ricerca.

5. Fare clic su **Trova**.

I criteri di ricerca vengono inoltrati a Web editor.

Viene visualizzata una pagina di status con il numero di record corrispondente ai criteri di ricerca.

6. Effettuare una delle seguenti operazioni:

- Fare clic su **Caricare dati** per visualizzare l'elenco dei record trovati. Web editor visualizza l'elenco dei record richiamati con l'interrogazione. Andare al passo 6.
- Fare clic su **Nuova ricerca** per ritornare alla pagina di ricerca. Ritornare al passo 3.

7. Selezionare il record da modificare.

Viene visualizzato il modulo appropriato.

8. Scorrere il record fino al campo che si desidera modificare e cambiarne il contenuto.

9. Fare clic su **Vai alla sessione di lavoro**.

Vengono visualizzati i risultati della sessione di lavoro relativi al modulo. Questi risultati contengono tutte le aggiunte, le modifiche e le eliminazioni effettuate ma non cancellate o elaborate durante questa sessione di lavoro di Web editor.

10. Se si desidera rimuovere le modifiche ai record dalla sessione di lavoro, selezionare la casella corrispondente a ciascuna modifica che si desidera rimuovere e fare clic su **Elimina selezionati**.
11. Fare clic su **Elabora sessione di lavoro** per inoltrare le modifiche selezionate al database.

In una pagina di status viene visualizzato un messaggio che indica che il processo è riuscito.

12. Spostarsi sul sito Web, fare clic sui collegamenti ipertestuali appropriati e verificare che le modifiche siano state eseguite.

Eliminazione di un record da una tabella mediante Web editor

Per eliminare un record da una tabella mediante Web editor, attenersi alla seguente procedura:

1. Da un browser Web, aprire il seguente URL:

`https://nome_host:8000/wcm/webeditor`

dove *nome_host* è il nome host HTTP completo di WebSphere Application Server.

Viene visualizzata la finestra di collegamento al database Web editor.

2. Immettere la password e il nome utente del database, quindi fare clic su **Collegamento**.

Web editor viene visualizzato nella finestra del browser con un elenco di nomi di tabelle nella barra dei menu di sinistra.

3. Fare clic sul collegamento ipertestuale appropriato nel sottomenu Ricerca della barra dei menu di sinistra.

Viene visualizzata la pagina di ricerca appropriata.

4. Specificare i criteri di ricerca attenendosi alla seguente procedura:
 - a. Selezionare la casella di spunta accanto a ciascun attributo che si desidera specificare nella ricerca.
 - b. Utilizzare il menu a discesa appropriato per ciascun attributo selezionato per scegliere la logica che si desidera utilizzare nella ricerca.
 - c. Nel campo accanto a ciascun attributo selezionato, inserire o selezionare il valore che si desidera utilizzare nella ricerca.

5. Fare clic su **Trova**.

I criteri di ricerca vengono inoltrati a Web editor.

Viene visualizzata una pagina di status con il numero di record corrispondente ai criteri di ricerca.

6. Effettuare una delle seguenti operazioni:

- Fare clic su **Caricare dati** per visualizzare l'elenco dei record trovati. Web editor visualizza l'elenco dei record richiamati con l'interrogazione. Andare al passo 6.
- Fare clic su **Nuova ricerca** per ritornare alla pagina di ricerca. Ritornare al passo 3.

7. Selezionare la casella di spunta accanto a ciascun record che si desidera eliminare.

8. Fare clic su **Sposta nell'elenco cancella**.

9. Fare clic sul collegamento ipertestuale appropriato nel sottomenu Sessione di lavoro della barra dei menu di sinistra.

Vengono visualizzati i risultati della sessione di lavoro relativi al modulo. Questi risultati contengono tutte le aggiunte, le modifiche e le eliminazioni effettuate ma non cancellate o elaborate durante questa sessione di lavoro di Web editor.

10. Se si desidera rimuovere le modifiche ai record dalla sessione di lavoro, selezionare la casella corrispondente a ciascuna modifica che si desidera rimuovere e fare clic su **Elimina selezionati**.
11. Fare clic su **Elabora sessione di lavoro** per inoltrare le modifiche al database.

In una pagina di status viene visualizzato un messaggio che indica che il processo è riuscito.

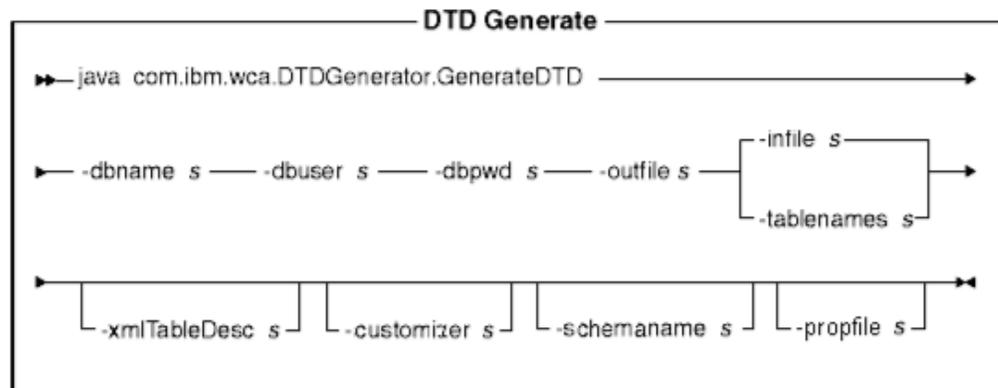
12. Spostarsi sul sito Web, fare clic sui collegamenti ipertestuali appropriati e verificare che le modifiche siano state eseguite.

Parte 4. Riferimento ai comandi

Capitolo 13. Comando DTD Generate

Questo comando crea i file DTD e i file schema da utilizzare con il pacchetto Loader.

Comando DTD Generate per sistemi Windows, AIX, Linux, e Solaris



Note:

1. Il diagramma su riportato rappresenta un riferimento per i parametri di comando. Il file o lo script di comandi fornito per questo comando e presente in Capitolo 10, "Configurazione degli script e dei comandi del pacchetto Loader" a pagina 79 agisce come wrapper per l'effettivo comando Java e accetta gli stessi parametri; per questo motivo, si consiglia di utilizzare il file o lo script di comandi piuttosto che richiamare direttamente il comando Java.
2. I nomi file specificati come parametri per questo comando possono essere preceduti da percorsi relativi o assoluti.

Valori dei parametri:

-dbname

Il nome del database di destinazione

-dbuser

Il nome dell'utente che si collega al database

-dbpwd

La password dell'utente che si collega al database

-outfile

Il nome del file DTD di emissione

-infile Il nome di un file di immissione contenente un nome della tabella di database in ogni riga

-tablenames

I nomi delle tabelle, separati da una virgola

-xmlTableDesc

Il percorso del file dello schema da creare

-customizer

Il nome del file properties di personalizzazione da utilizzare.

DB2ConnectionCustomizer.properties è il file predefinito. Il file property di personalizzazione può essere specificato come illustrato nel seguente esempio:

```
-customizer d:\wc\prop\dttdgen.properties
```

Se questo file si trova in una directory specificata nella variabile di ambiente del sistema classpath, lo stesso file può essere specificato come illustrato nel seguente esempio:

```
-customizer dttdgen
```

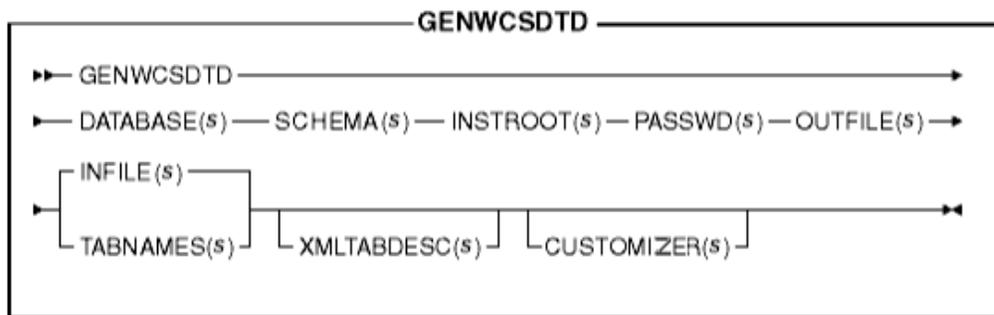
-schemaname

Il nome dello schema del database di destinazione

-propFile

Un file contenente proprietà, ad esempio un file di proprietà esterne in cui il testo di aiuto, i valori predefiniti e le informazioni di descrizione dei campi possono essere memorizzati per una descrizione del modulo di Web editor

Comando DTD Generate per sistemi iSeries



Nota: I nomi file specificati come parametri per questo comando possono essere preceduti da percorsi relativi o assoluti.

Valori dei parametri:

DATABASE

Il nome del database di destinazione, come visualizzato nella directory del database relazionale

SCHEMA

Il nome dello schema del database di destinazione, uguale al nome dell'istanza

INSTROOT

Il nome completo del percorso principale dell'istanza di WebSphere Commerce, ad esempio

/QIBM/UserData/WebCommerce/instances/nome_istanza

PASSWD

La password per l'istanza di WebSphere Commerce

OUTFILE

Il nome del file DTD di emissione

INFILE

Il nome di un file di immissione contenente un nome della tabella di database in ogni riga

TABNAMES

I nomi delle tabelle, separati da una virgola

XMLTABDESC

Il percorso del file dello schema da creare. Questo parametro è facoltativo.

CUSTOMIZER

Il nome del file properties di personalizzazione da utilizzare. Il file predefinito è `ISeries_GENWCSDTD_Customizer.properties`. Il file property di personalizzazione può essere specificato come illustrato nel seguente esempio:

```
CUSTOMIZER(/wc/prop/dtdgen.properties)
```

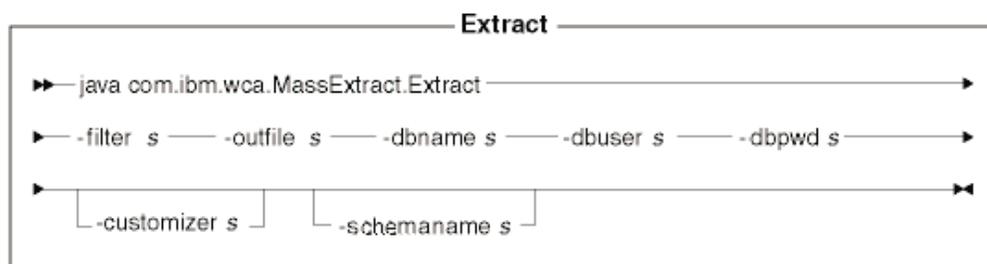
Se questo file si trova in una directory specificata nella variabile di ambiente del sistema `classpath`, lo stesso file può essere specificato come illustrato nel seguente esempio:

```
CUSTOMIZER(dtdgen)
```

Capitolo 14. Comando Extract

Questo comando consente di estrarre da un database un sottoinsieme di dati selezionato in formato file XML.

Comando Extract per sistemi Windows, AIX, Linux, e Solaris



Note:

1. Il diagramma su riportato rappresenta un riferimento per i parametri di comando. Il file o lo script di comandi fornito per questo comando e presente in Capitolo 10, "Configurazione degli script e dei comandi del pacchetto Loader" a pagina 79 agisce come wrapper per l'effettivo comando Java e accetta gli stessi parametri; per questo motivo, si consiglia di utilizzare il file o lo script di comandi piuttosto che richiamare direttamente il comando Java.
2. I nomi file specificati come parametri per questo comando possono essere preceduti da percorsi relativi o assoluti.

Valori dei parametri:

-filter Il nome del file del filtro di estrazione

-outfile

Il nome del file XML di emissione in cui vengono memorizzati i dati estratti

-dbname

Il nome del database dal quale vengono estratti i dati

-dbuser

Il nome utente del database dal quale vengono estratti i dati

-dbpwd

La password associata al nome utente del database dal quale vengono estratti i dati

-customizer

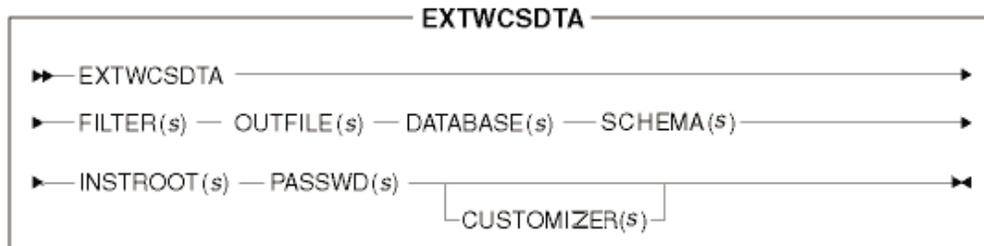
Il nome del file properties di personalizzazione da utilizzare. `DB2ConnectionCustomizer.properties` è il file predefinito. Il file property di personalizzazione può essere specificato come illustrato nel seguente esempio:

```
-customizer d:\wc\prop\extract.properties
```

Se questo file si trova in una directory specificata nella variabile di ambiente del sistema `classpath`, lo stesso file può essere specificato come illustrato nel seguente esempio:

-customizer extract
-schemaname
Il nome dello schema del database di destinazione

Comando Extract per sistemi iSeries



Nota: I nomi file specificati come parametri per questo comando possono essere preceduti da percorsi relativi o assoluti.

Valori dei parametri:

FILTER

Il nome del file del filtro di estrazione

OUTFILE

Il nome del file XML di emissione in cui vengono memorizzati i dati estratti

DATABASE

Il nome del database dal quale vengono estratti i dati come visualizzato nella directory del database relazionale

SCHEMA

Il nome dello schema del database dal quale vengono estratti i dati; questo nome è uguale al nome istanza

INSTROOT

Il nome completo del percorso principale dell'istanza di WebSphere Commerce, ad esempio
`/QIBM/UserData/WebCommerce/instances/nome_istanza`

PASSWD

La password per l'istanza di WebSphere Commerce

CUSTOMIZER

Il nome del file properties di personalizzazione da utilizzare. Il file predefinito è `ISeries_EXTWCSDTA_Customizer.properties`. Il file property di personalizzazione può essere specificato come illustrato nel seguente esempio:

```
CUSTOMIZER(/wc/prop/extract.properties)
```

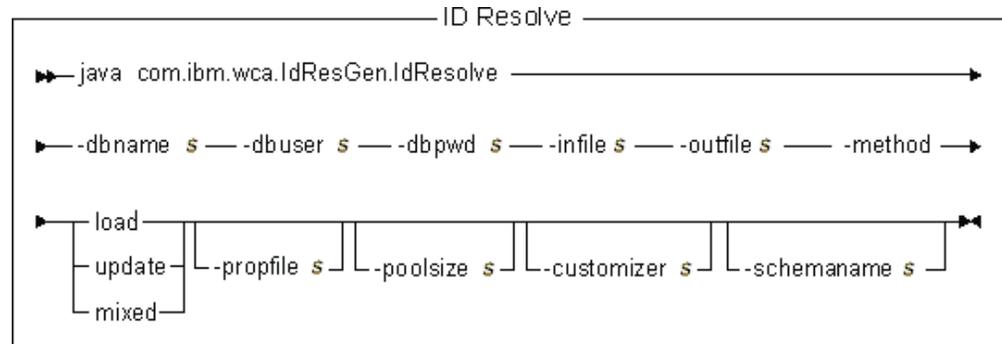
Se questo file si trova in una directory specificata nella variabile di ambiente del sistema `classpath`, lo stesso file può essere specificato come illustrato nel seguente esempio:

```
CUSTOMIZER(extract)
```

Capitolo 15. Comando ID Resolve

Questo comando genera gli identificativi per gli elementi XML che richiedono il caricamento in un database.

Comando ID Resolve per sistemi Windows, AIX, Linux, e Solaris



Note:

1. Il diagramma su riportato rappresenta un riferimento per i parametri di comando. Il file o lo script di comandi fornito per questo comando e presente in Capitolo 10, "Configurazione degli script e dei comandi del pacchetto Loader" a pagina 79 agisce come wrapper per l'effettivo comando Java e accetta gli stessi parametri; per questo motivo, si consiglia di utilizzare il file o lo script di comandi piuttosto che richiamare direttamente il comando Java.
2. I nomi file specificati come parametri per questo comando possono essere preceduti da percorsi relativi o assoluti.

Valori dei parametri:

-dbname

Il nome del database di destinazione

-dbuser

Il nome dell'utente che si collega al database

-dbpwd

La password dell'utente che si collega al database

-infile Il nome del documento XML di immissione che contiene descrizioni dei record della tabella

-outfile

Il nome del file XML di emissione da produrre; tale file deve essere utilizzato come immissione per il pacchetto Loader

-method

Il metodo da utilizzare nell'elaborazione del file di immissione

- Utilizzare il metodo `load` per elaborare il file di immissione se **tutti** i record del file **non esistono** nel database.
- Utilizzare il metodo `update` per elaborare il file di immissione se **tutti** i record del file **esistono** nel database.

Valori dei parametri:

DATABASE

Il nome del database di destinazione, come visualizzato nella directory del database relazionale

SCHEMA

Il nome dello schema del database di destinazione, uguale al nome dell'istanza

INSTROOT

Il nome completo del percorso principale dell'istanza di WebSphere Commerce, ad esempio
`/QIBM/UserData/WebCommerce/instances/nome_istanza`

PASSWD

La password per l'istanza di WebSphere Commerce

INFILE

Il nome del documento XML di immissione che contiene descrizioni dei record della tabella

OUTFILE

Il nome del file XML di emissione da produrre; tale file deve essere utilizzato come immissione per il pacchetto Loader

METHOD

Il metodo da utilizzare nell'elaborazione del file di immissione

- Utilizzare il metodo load (*LOAD) per elaborare il file di immissione se *tutti* i record del file *non esistono* nel database.
- Utilizzare il metodo update (*UPD) per elaborare il file di immissione se *tutti* i record del file *esistono* nel database.
- Utilizzare il metodo mixed (*MIX) per elaborare il file di immissione se *solo alcuni* record del file *esistono* nel database.

PROFILE

Il file di testo contenente le proprietà Java sotto forma di coppie nome=valore. Questo file viene utilizzato per definire i nomi delle colonne look-aside per la ricerca dell'identificativo della chiave esterna e per la selezione del predicato delle interrogazioni della tabelle principali (come CATEGORY e PRODUCT). E' possibile omettere le voci in questo file per le tabelle che abbiano un indice univoco predefinito che non include l'identificativo. Il file predefinito è `IdResolveKeys.properties`. Sebbene questo file abbia una estensione `.properties`, non utilizzare l'estensione quando si specifica il valore.

POOLSIZE

Il numero di identificativi da riservare. Il numero predefinito è 50.

CUSTOMIZER

Il nome del file properties di personalizzazione da utilizzare. Il file predefinito è `ISeries_RESWCSID_Customizer.properties`. Il file property di personalizzazione può essere specificato come illustrato nei seguenti esempi:

```
CUSTOMIZER(/wc/prop/idres.properties)
CUSTOMIZER(/wc/prop/idres)
```

Se questo file esiste nella directory corrente, lo stesso file può essere specificato come illustrato nel seguente esempio:

```
CUSTOMIZER(idres.properties)
```

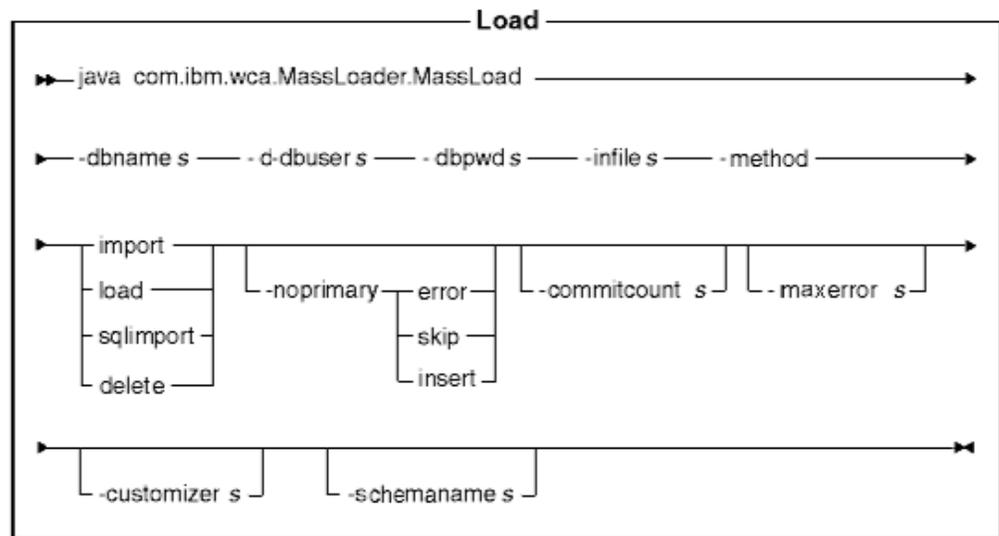
Se questo file si trova in una directory specificata nella variabile di ambiente del sistema classpath, lo stesso file può essere specificato come illustrato nel seguente esempio:

```
CUSTOMIZER(idres)
```

Capitolo 16. Comando Load

Questo comando carica un file XML di immissione in un database di destinazione.

Comando Load per sistemi Windows, AIX, Linux, e Solaris



Note:

1. Il diagramma su riportato rappresenta un riferimento per i parametri di comando. Il file o lo script di comandi fornito per questo comando e presente in Capitolo 10, "Configurazione degli script e dei comandi del pacchetto Loader" a pagina 79 agisce come wrapper per l'effettivo comando Java e accetta gli stessi parametri; per questo motivo, si consiglia di utilizzare il file o lo script di comandi piuttosto che richiamare direttamente il comando Java.
2. I nomi file specificati come parametri per questo comando possono essere preceduti da percorsi relativi o assoluti.

Valori dei parametri:

-dbname

Il nome del database di destinazione

-dbuser

Il nome dell'utente che si collega al database

-dbpwd

La password dell'utente che si collega al database

-infile Il nome del file XML di immissione

-method

La modalità di operazione per il pacchetto Loader da utilizzare durante l'inserimento dei dati nel database

- Il metodo `load` utilizza la funzione di caricamento nativo dal fornitore del database. E' possibile utilizzare il metodo `load` per i database Oracle locale e remoto, ma il metodo `load` può essere utilizzato solo per i database DB2 locali.

- Utilizzare il metodo import per caricare i dati nei database DB2 remoti. Il metodo import utilizza l'opzione import o update se disponibili dal fornitore del database. Se viene specificato questo metodo per un database in cui l'opzione import o update non è disponibile, ad esempio Oracle, le istruzioni SQL che utilizzano JDBC consentono di aggiornare il database.
- Il metodo SQL import (sqlimport) può essere utilizzato sia con i database locali che con quelli remoti.
- Il metodo delete elimina i dati dal database.

Se si utilizza la sincronizzazione dello spazio di ricerca di Product Advisor, è necessario utilizzare il metodo sqlimport o il metodo delete.

-noprimary

L'operazione che il pacchetto Loader deve eseguire nel caso in cui la chiave primaria risulti mancante per un record in un file di immissione. L'opzione error indica che la mancanza della chiave principale deve essere riportata come un errore e quindi l'operazione deve terminare. L'opzione skip salta i record nel file di immissione che non dispongono di una chiave primaria. L'opzione insert tenta di elaborare (inserire o eliminare) i dati. L'opzione predefinita è error.

-commitcount

Il numero dei record elaborati prima che si verifichi il commit del database durante l'utilizzo della modalità di aggiornamento SQL dell'operazione. Il valore predefinito è 1.

-maxerror

Il numero di errori dopo il quale il pacchetto Loader termina la modalità di aggiornamento SQL

-customizer

Il nome del file properties di personalizzazione da utilizzare. MassLoadCustomizer.properties è il file predefinito. Il file property di personalizzazione può essere specificato come illustrato nel seguente esempio:

```
-customizer d:\wc\prop\ml.properties
```

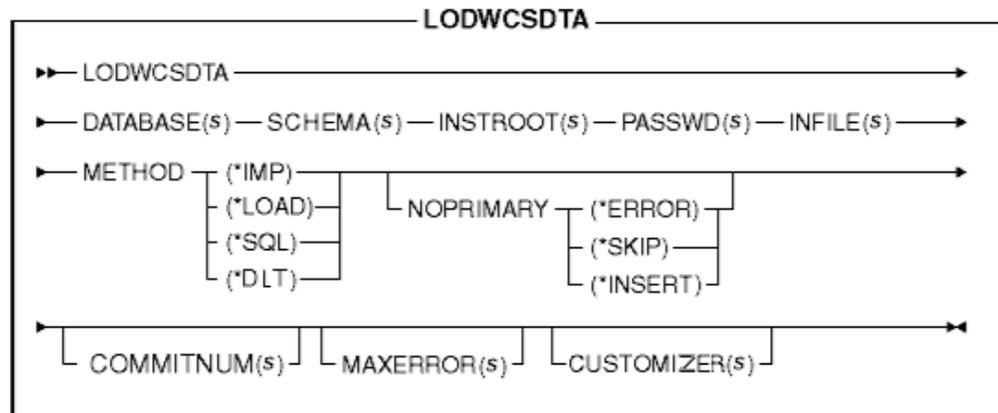
Se questo file si trova in una directory specificata nella variabile di ambiente del sistema classpath, lo stesso file può essere specificato come illustrato nel seguente esempio:

```
-customizer ml
```

-schemaname

Il nome dello schema del database di destinazione

Comando Load per sistemi iSeries



Nota: I nomi file specificati come parametri per questo comando possono essere preceduti da percorsi relativi o assoluti.

Valori dei parametri:

DATABASE

Il nome del database di destinazione come visualizzato nella directory del database relazionale

SCHEMA

Il nome dello schema del database di destinazione, uguale al nome dell'istanza

INSTROOT

Il nome completo del percorso principale dell'istanza di WebSphere Commerce, ad esempio
/QIBM/UserData/WebCommerce/instances/nome_istanza

PASSWD

La password per l'istanza di WebSphere Commerce

INFILE

Il nome del file XML di immissione

METHOD

La modalità di operazione per il pacchetto Loader da utilizzare durante l'inserimento dei dati nel database

- Il metodo load (*LOAD) utilizza la funzione di caricamento nativo dal fornitore del database. E' possibile utilizzare il metodo load (*LOAD) per i database Oracle locale e remoto, ma il metodo load (*LOAD) può essere utilizzato solo per i database DB2 locali.
- Utilizzare l'opzione import (*IMP) per caricare i dati nei database DB2 remoti. Il metodo import utilizza l'opzione import o update se disponibili dal fornitore del database. Se l'opzione import o update non è disponibile, le istruzioni SQL che utilizzano JDBC consentono di aggiornare il database.
- Il metodo SQL import (*SQL) può essere utilizzato sia con i database locali che con quelli remoti.
- Il metodo delete (*DLT) elimina i dati dal database.

NOPRIMARY

L'operazione che il pacchetto Loader deve eseguire nel caso in cui la chiave primaria risulti mancante per un record in un file di immissione. L'opzione error (*ERROR) indica la mancanza della chiave principale come un errore e quindi l'operazione deve terminare. L'opzione skip (*SKIP) salta i record nel file di immissione che non dispongono di una chiave primaria. L'opzione insert (*INSERT) tenta di elaborare (inserire o eliminare) i dati. L'opzione predefinita è error.

COMMITNUM

Il numero dei record elaborati prima che si verifichi il commit del database durante l'utilizzo della modalità di aggiornamento SQL dell'operazione. Il valore predefinito è 1.

MAXERROR

Il numero di errori dopo il quale il pacchetto Loader termina la modalità di aggiornamento SQL

CUSTOMIZER

Il nome del file properties di personalizzazione da utilizzare. Il file predefinito è ISeries_LODWCSDTA_Customizer.properties. Il file property di personalizzazione può essere specificato come illustrato nel seguente esempio:

```
CUSTOMIZER(/wc/prop/ml.properties)
```

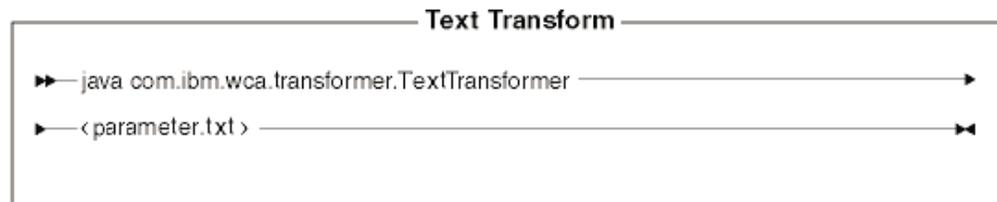
Se questo file si trova in una directory specificata nella variabile di ambiente del sistema classpath, lo stesso file può essere specificato come illustrato nel seguente esempio:

```
CUSTOMIZER(ml)
```

Capitolo 17. Comando Text Transform

Questo comando converte i dati da un formato variabile delimitato da caratteri in un formato XML.

Comando Text Transform per sistemi Windows, AIX, Linux e Solaris



Nota: Il diagramma su riportato rappresenta un riferimento per i parametri di comando. Il file o lo script di comandi fornito per questo comando e presente in Capitolo 10, "Configurazione degli script e dei comandi del pacchetto Loader" a pagina 79 agisce come wrapper per l'effettivo comando Java e accetta gli stessi parametri; per questo motivo, si consiglia di utilizzare il file o lo script di comandi piuttosto che richiamare direttamente il comando Java.

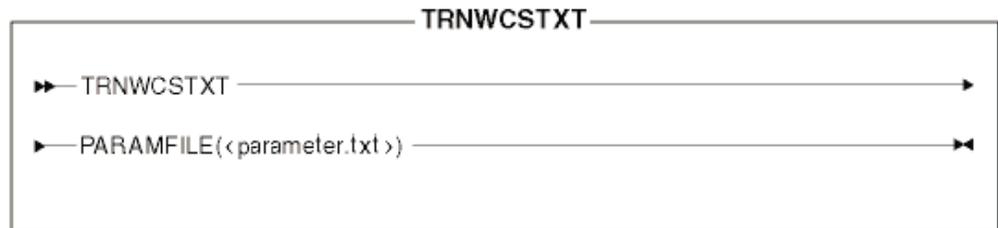
Valori dei parametri

I valori di seguito riportati vengono specificati e separati da virgole in un file di parametri (*parametro.txt*):

- input file
Il nome del file da convertire
- schema file
Il nome del file di schema XML da utilizzare nella conversione
- output file
Il nome del file di emissione nel quale verranno memorizzati i dati convertiti
- transformation method
Il metodo da utilizzare nell'aggiunta dei dati al file di emissione. Specificare **Crea** se deve essere creato un nuovo file oppure specificare **Aggiungi** se i dati di emissione devono essere aggiunti ad un file di dati esistente.

Nota: Questo file viene indicato anche come file "manifest" o "command".

Comando Text Transform per sistemi iSeries



Valori dei parametri

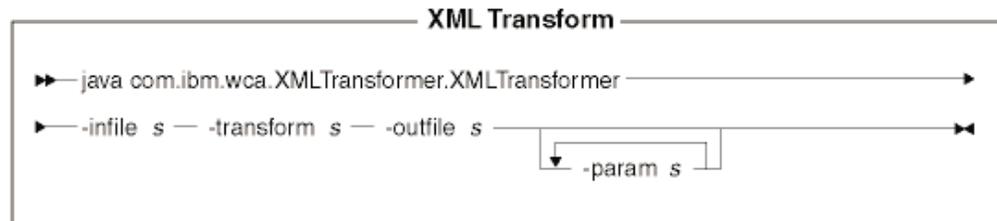
I valori di seguito riportati vengono specificati e separati da virgole in un file di parametri (*parametro.txt*):

- input file
Il nome del file da convertire
- schema file
Il nome del file di schema XML da utilizzare nella conversione
- output file
Il nome del file di emissione nel quale verranno memorizzati i dati convertiti
- transformation method
Il metodo da utilizzare nell'aggiunta dei dati al file di emissione. Specificare **Crea** se deve essere creato un nuovo file oppure specificare **Aggiungi** se i dati di emissione devono essere aggiunti ad un file di dati esistente.

Capitolo 18. Comando XML Transform

Questo comando converte un file XML in un formato diverso.

Comando XML Transform per sistemi Windows, AIX, Linux, e Solaris



Note:

1. Il diagramma su riportato rappresenta un riferimento per i parametri di comando. Il file o lo script di comandi fornito per questo comando e presente in Capitolo 10, "Configurazione degli script e dei comandi del pacchetto Loader" a pagina 79 agisce come wrapper per l'effettivo comando Java e accetta gli stessi parametri; per questo motivo, si consiglia di utilizzare il file o lo script di comandi piuttosto che richiamare direttamente il comando Java.
2. I nomi file specificati come parametri per questo comando possono essere preceduti da percorsi relativi o assoluti.

Valori dei parametri:

-infile Il nome del file da convertire

-transform

Il nome del file della regola XSL di conversione

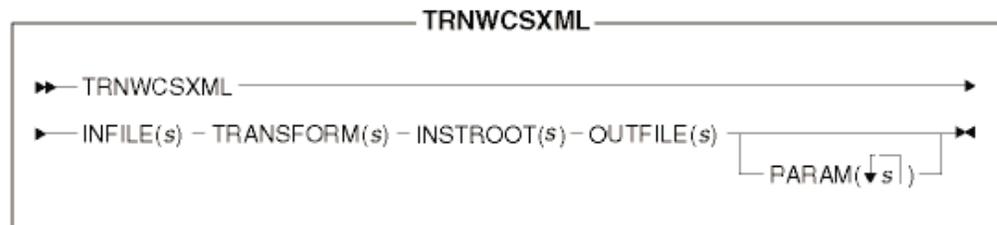
-outfile

Il nome del file XML di emissione nel quale verranno memorizzati i dati convertiti

-param

Il parametro da passare al file della regola XSL. *Questo parametro è facoltativo.* Questo parametro può essere specificato più volte per passare più coppie "nome=valore".

Comando XML Transform per sistemi iSeries



Nota: I nomi file specificati come parametri per questo comando possono essere preceduti da percorsi relativi o assoluti.

Valori dei parametri:

INFILE

Il nome del file da convertire

TRANSFORM

Il nome del file della regola XSL di conversione

INSTROOT

Il nome completo del percorso principale dell'istanza di WebSphere Commerce, ad esempio
`/QIBM/UserData/WebCommerce/instances/nome_istanza`

OUTFILE

Il nome del file XML di emissione nel quale verranno memorizzati i dati convertiti

PARAM

Il parametro da passare al file della regola XSL. *Questo parametro è facoltativo.* La stringa può contenere più valori per passare più coppie "nome=valore".

Informazioni particolari

Queste informazioni sono state sviluppate per i prodotti e i servizi offerti negli Stati Uniti. E' possibile che negli altri paesi l'IBM non offra i prodotti, i servizi o le funzioni illustrati in questo documento. Consultare il rappresentante IBM locale per informazioni sui prodotti e sui servizi disponibili nel proprio paese. Qualsiasi riferimento a prodotti, programmi o servizi IBM non implica che l'IBM intenda renderli disponibili in tutti i paesi in cui opera. In sostituzione a quelli forniti dall'IBM, possono essere usati prodotti, programmi o servizi funzionalmente equivalenti che non comportino la violazione dei diritti di proprietà intellettuale o di altri diritti dell'IBM. E' 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 nella presente pubblicazione. La fornitura di questa pubblicazione non implica la concessione di alcuna licenza su di essi. Per eventuali domande sulle licenze, scrivere al seguente indirizzo:

Director of Commercial Relations
IBM Europe
Schoenaicher Str. 220
D-7030 Boeblingen
Deutschland

Per richieste di informazioni relative alla licenza riguardanti informazioni double-byte (DBCS), fare richiesta scrivendo 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 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. 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 dalla Selfin e diventeranno esclusiva delle stesse.

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 Canada Ltd.
Office of the Lab Director
8200 Warden Avenue, Markham, Ontario L6G 1C7
Canada

Queste informazioni possono essere rese disponibili secondo condizioni contrattuali appropriate, compreso, in alcuni casi, il pagamento di un addebito.

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 relativi alle prestazioni contenuti in questa pubblicazione sono stati determinati in un ambiente controllato. Pertanto, i risultati ottenuti in ambienti operativi diversi possono variare in modo considerevole. Alcune misure 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 misure potrebbero essere state ricavate mediante estrapolazione. I risultati possono quindi variare. Gli utenti di questa pubblicazione devono verificare che i dati siano applicabili al loro specifico ambiente.

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.

Tutte le dichiarazioni riguardanti la futura direzione o le intenzioni della IBM sono soggette a sostituzione o al ritiro senza preavviso, e rappresentano unicamente scopi e obiettivi della IBM stessa.

Tutti i prezzi IBM riportati sono prezzi al dettaglio suggeriti dalla IBM stessa, sono correnti e soggetti a cambiamenti senza preavviso. I prezzi al fornitore possono variare.

Queste informazioni hanno solo scopo di pianificazione. Queste informazioni possono essere soggette a variazioni prima che i prodotti descritti siano disponibili.

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 nel manuale sono fittizi e ogni riferimento a nomi ed indirizzi reali è puramente casuale.

LICENZA SOGGETTA ALLE LEGGI SUL DIRITTO D'AUTORE:

Queste informazioni contengono esempi di programmi applicativi in lingua originale, che illustrano le tecniche di programmazione su diverse piattaforme operative. Potete copiare, modificare e distribuire questi 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 la affidabilità, la praticità o il funzionamento di questi programmi. Potete copiare, modificare e distribuire questi 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) IBM.

Ogni copia o parte di tali programmi di esempio deve includere un'informazione sul copyright come questa di seguito indicata:

©Copyright International Business Machines Corporation 2001.
Parti di questo programma derivano da programmi di esempio della IBM Corp.
©Copyright IBM Corp. 2000, 2001. Tutti i diritti riservati.

Se questa pubblicazione viene visualizzata in formato elettronico, è possibile che le fotografie e le illustrazioni a colori non vengano visualizzate.

Marchi e marchi di servizio

I seguenti termini sono marchi dell'IBM Corporation:

AIX
DB2
DB2 Universal Database
IBM
iSeries
OS/400
WebSphere

Microsoft, Windows, Windows NT e Windows 2000 sono marchi registrati della Microsoft Corporation negli Stati Uniti, e/o in altri paesi.

Oracle è un marchio registrato della Oracle Corporation negli Stati Uniti e/o in altri paesi.

Solaris, Java e tutti i marchi e i logo basati su Java sono marchi o marchi registrati della Sun Microsystems, Inc. negli Stati Uniti e/o in altri paesi.

Altri nomi di prodotti, società e servizi potrebbero essere marchi o marchi di servizi di altre società.

IBM