

IBM<sup>®</sup> WebSphere<sup>®</sup> Commerce



# Catalog Manager - Manual do Usuário

*Versão 54*



IBM<sup>®</sup> WebSphere<sup>®</sup> Commerce



# Catalog Manager - Manual do Usuário

*Versão 54*

**Nota**

Antes de utilizar estas informações e o produto suportado por elas, leia as informações em "Avisos" na página 125.

**Primeira Edição, Primeira Revisão (Abril de 2002)**

Esta edição se aplica ao seguinte produto:

- IBM WebSphere Commerce, Versão 5.4 (Programa 5724-A18)

e a todos os releases e modificações subseqüentes dos produtos listados acima, até que seja indicado de outra forma em novas edições. Certifique-se de que esteja utilizando a edição correta para o nível do produto.

Solicite publicações através de um representante IBM ou da filial IBM de sua localidade. As publicações não estão armazenadas no endereço fornecido abaixo.

A IBM agradece os seus comentários. Você pode enviar seus comentários pelos seguintes métodos:

1. Eletronicamente para o endereço de e-mail listado abaixo.

[torrcf@ca.ibm.com](mailto:torrcf@ca.ibm.com)

Inclua seu endereço de rede completo, se desejar uma resposta.

2. Por correio, para o seguinte endereço:

Centro Industrial IBM Brasil  
Centro de Traduções  
Caixa Postal 71  
CEP: 13001-970  
Campinas, SP - Brasil

Quando o Cliente envia seus comentários, concede direitos não-exclusivos à IBM para usá-los ou distribuí-los da maneira que achar conveniente, sem que isso implique em qualquer compromisso ou obrigação para com o Cliente.

© Copyright International Business Machines Corporation 2002. Todos os direitos reservados.

# Índice

<b>Antes de Começar</b> . . . . .	<b>v</b>
Convenções Utilizadas neste Manual . . . . .	v
Quem Deve Ler Este Manual . . . . .	vi
Onde Localizar mais Informações . . . . .	vi

## **Parte 1. Visão Geral do Catalog Manager** . . . . . **1**

## **Parte 2. Transformando, Carregando e Extraindo Dados** . . . . . **3**

<b>Capítulo 1. Introdução.</b> . . . . .	<b>5</b>
Utilitários . . . . .	8
Ferramentas Administrativas . . . . .	9

## **Capítulo 2. Transformando Texto** . . . . . **11**

Lançando a Ferramenta de Transformação de Texto	12
Utilizando a Exibição de Edição de Esquema de Texto . . . . .	13
Criando um Arquivo Esquema . . . . .	13
Abrindo um Arquivo Esquema . . . . .	13
Salvando um Arquivo Esquema . . . . .	13
Editando um Arquivo Esquema . . . . .	13
Incluindo um Elemento . . . . .	13
Removendo um Elemento . . . . .	14
Substituindo um Elemento . . . . .	14
Movendo um Elemento uma Linha Acima . . . . .	14
Movendo um Elemento uma Linha Abaixo . . . . .	14
Incluindo um Atributo . . . . .	14
Removendo um Atributo . . . . .	14
Substituindo um Atributo . . . . .	14
Movendo um Atributo uma Linha Acima . . . . .	14
Movendo um Atributo uma Linha Abaixo . . . . .	14
Alterando a Estrutura de um Arquivo Esquema	15
Preparando um Arquivo de Esquema para Transformar Dados de XML em um Formato Variável Delimitado por Caracteres . . . . .	15
Utilizando a Exibição de Edição de Comando de Transformação . . . . .	16
Criando um Comando . . . . .	16
Removendo um Comando . . . . .	16
Editando ou Substituindo um Comando . . . . .	16
Movendo um Comando uma Linha Acima . . . . .	16
Movendo um Comando uma Linha Abaixo . . . . .	17
Limpando um Comando . . . . .	17
Utilizando a Exibição de Processo de Transformação	17

## **Capítulo 3. Transformando Dados XML** **19**

Lançando o Editor XSL . . . . .	20
Trabalhando com Áreas de Construção da Regra de Mapeamento . . . . .	21
Criando uma Área de Construção da Regra de Mapeamento . . . . .	21

Modificando uma Área de Construção da Regra de Mapeamento . . . . .	21
Excluindo uma Área de Construção da Regra de Mapeamento . . . . .	21
Utilizando o Editor XSL . . . . .	22
Criando uma Regra de Mapeamento . . . . .	22
Mapeando Elemento a Elemento . . . . .	22
Mapeando Atributo a Atributo . . . . .	22
Criando uma Expressão de Mapeamento Personalizada. . . . .	22
Excluindo uma Regra de Mapeamento . . . . .	23
Processando uma Transformação de XML . . . . .	23
Personalizando a Tabela de Regras de Mapeamento . . . . .	23
Exibindo uma Regra XSL Completa/Expressão de Valor . . . . .	24

## **Capítulo 4. Gerando um DTD e um Esquema.** . . . . . **25**

Configurando o DTD Generator . . . . .	25
Gerando um DTD . . . . .	26
Gerando um Esquema e um Arquivo XML Detalhado . . . . .	28

## **Capítulo 5. Resolvendo Identificadores** **29**

Configurando o ID Resolver . . . . .	29
Definindo como o ID Resolver Trata Data e Hora	30
Definindo como o ID Resolver Trata Armazenamento. . . . .	30
Definindo como o ID Resolver Trata Drivers do Banco de Dados . . . . .	31
Determinando como Processar Dados. . . . .	31
Escolhendo o Método Load . . . . .	32
Escolhendo o Método Update . . . . .	32
Escolhendo o Método Mixed . . . . .	32
Utilizando Técnicas de Resolução de ID . . . . .	33
Especificando um Arquivo de Propriedades com o ID Resolver. . . . .	33
Utilizando um Arquivo de Propriedades para Gerar Identificadores . . . . .	35
Utilizando um Arquivo de Propriedades com Chaves Compostas . . . . .	36
Utilizando o Arquivo de Propriedades com Chaves Principais em Cascata . . . . .	37
Utilizando Resolução de Alias Interno . . . . .	38
Exemplo Parcial de Utilização da Resolução de ID de Alias Interno. . . . .	39
Utilizando Resolução de Índice Exclusivo . . . . .	40
Exemplo Parcial de Resolução de Índice Exclusivo . . . . .	40
Carregando Dados na Tabela MEMBER . . . . .	42
Criando um Relacionamento Externo Utilizando a Tabela REFKEYS. . . . .	43
Erros de Resolução de Problemas . . . . .	44

## Capítulo 6. Carregando Dados . . . . . 45

Configurando o Loader . . . . .	45
Ignorando Elementos no Arquivo de Entrada . . . . .	46
Inserindo NULL em uma Coluna . . . . .	46
Carregando Data e Hora e Dados de Data . . . . .	46
Carregando Data e Hora Atuais . . . . .	47
Exemplo de Carregamento de Data e Hora Atuais . . . . .	49
Exemplos de Inclusão de Durações em Data e Hora Atuais . . . . .	49
Gerenciando Filas de Eventos . . . . .	50
Executando com Software do Banco de Dados e Sistemas Operacionais Diferentes . . . . .	50
Substituindo um Componente . . . . .	53
Utilizando a Sincronização do Espaço de Procura do Consultor de Procura . . . . .	54
Personalizando Sincronização do Espaço de Procura do Consultor de Produto . . . . .	57
Determinando como Processar Dados ao Utilizar o Loader . . . . .	58
Escolhendo o Método Load . . . . .	58
Escolhendo o Método Importar . . . . .	59
Escolhendo o Método SQL Import . . . . .	59
Outras Considerações . . . . .	59
Carregando Documentos Grandes . . . . .	60
Dica de Resolução de Problemas . . . . .	62

## Capítulo 7. Extraindo Dados . . . . . 63

Criando um Filtro de Extração . . . . .	63
Configurando o Extractor . . . . .	66

## Capítulo 8. Utilizando o Logger do Loader Package . . . . . 67

Configurando o Log em seu Ambiente para Sistemas Windows NT, Windows 2000, AIX, Linux e Solaris . . . . .	67
Exemplo de Definição da Variável classpath . . . . .	67
Exemplo de Especificação da Propriedade do Sistema com.ibm.wca.logging.configFile . . . . .	67
Personalizando o log para o Loader package . . . . .	68
Rotinas de Tratamento . . . . .	69
Filtros . . . . .	70
Formatos . . . . .	70
Exemplo: WCALoggerConfig.xml e WCALogger.dtd . . . . .	71
WCALoggerConfig.xml . . . . .	71
WCALogger.dtd . . . . .	72

## Capítulo 9. Utilizando o Relator de Erros do Loader Package . . . . . 75

## Capítulo 10. Configurando Comandos e Scripts do Loader Package . . . . . 77

## Parte 3. Utilizando o Editor da Web 79

### Capítulo 11. Configurando o Editor da Web . . . . . 81

Configurando o Editor da Web . . . . .	82
--	----

Editando o Arquivo webeditor.properties . . . . .	82
Alterando a Localização de Arquivos Temporários . . . . .	83
Criando um Arquivo de Descrição de Formulário XML Utilizando o DTD Generator . . . . .	84
Personalizando a Descrição de Formulário XML . . . . .	90
Editando Nomes de Formulários . . . . .	90
Alterando a Descrição de um Campo . . . . .	91
Incluindo um Menu Drop-down . . . . .	91
Incluindo Ajuda para Campos . . . . .	93
Personalizando Resultados da Pesquisa e a Lista de Sessões de Trabalho . . . . .	93
Editando o Arquivo weProcessList . . . . .	95
Editando o Arquivo webeditor.xml . . . . .	97

### Capítulo 12. Trabalhando com Catálogos . . . . . 99

Incluindo um Registro em uma Tabela Utilizando o Editor da Web . . . . .	99
Modificando um Registro em uma Tabela Utilizando o Editor da Web . . . . .	100
Excluindo um Registro de uma Tabela Utilizando o Editor da Web . . . . .	102

## Parte 4. Referência de Comandos 105

### Capítulo 13. Comando DTD Generate 107

Comando DTD Generate para Sistemas Windows, AIX, Linux e Solaris . . . . .	107
Comando DTD Generate para Sistemas iSeries . . . . .	108

### Capítulo 14. Comando Extract . . . . . 111

Comando Extract para Sistemas Windows, AIX, Linux e Solaris . . . . .	111
Comando Extract para Sistemas iSeries . . . . .	112

### Capítulo 15. Comando ID Resolve . . . . . 113

Comando ID Resolve para Sistemas Windows, AIX, Linux e Solaris . . . . .	113
Comando ID Resolve para Sistemas iSeries . . . . .	114

### Capítulo 16. Comando Load . . . . . 117

Comando Load para Sistemas Windows, AIX, Linux e Solaris . . . . .	117
Comando Load para Sistemas iSeries . . . . .	119

### Capítulo 17. Comando Text Transform 121

Comando Text Transform para Sistemas Windows, AIX, Linux e Solaris . . . . .	121
Comando Text Transform para Sistemas iSeries . . . . .	122

### Capítulo 18. Comando XML Transform 123

Comando XML Transform para Sistemas Windows, AIX, Linux e Solaris . . . . .	123
Comando XML Transform para Sistemas iSeries . . . . .	123

### Avisos . . . . . 125

Marcas e Marcas de Serviço . . . . .	127
--------------------------------------	-----

---

## Antes de Começar

O *IBM WebSphere Commerce Catalog Manager - Manual do Usuário* fornece informações sobre o WebSphere Commerce Catalog Manager. Em particular, fornece detalhes sobre os seguintes tópicos:

- Transformando, carregando e extraindo dados utilizando ferramentas e utilitários do Catalog Manager
- Utilizando o Editor da Web do Catalog Manager para trabalhar com dados de catálogo
- Comandos do Catalog Manager

---

## Convenções Utilizadas neste Manual

Este manual utiliza as seguintes convenções:

<b>Negrito</b>	indica comandos ou controles da interface gráfica com o usuário (GUI) como nomes de campos, botões ou opções de menu.
Monoespaçado	indica exemplos de texto que você digita exatamente conforme mostrado e também nomes de arquivos e caminhos de diretórios.
<i>Itálico</i>	é utilizado para ênfase e para variáveis que você substitui por seus próprios valores.
	indica informações específicas do WebSphere Commerce para Windows®.
	indica informações específicas do WebSphere Commerce para Windows NT®.
	indica informações específicas do WebSphere Commerce para Windows 2000®.
	indica informações específicas do WebSphere Commerce para AIX®.
	indica informações específicas do software WebSphere Commerce para Solaris™ Operating Environment.
	indica informações específicas do WebSphere Commerce para Linux.
	indica informações específicas do WebSphere Commerce para IBM @server™ iSeries™ (anteriormente conhecido como AS/400®).
	indica informações específicas do DB2® Universal Database.
	indica informações específicas do banco de dados Oracle®.
	indica informações específicas do WebSphere Commerce Business Edition.
	indica informações específicas do WebSphere Commerce Professional Edition.

---

## Quem Deve Ler Este Manual

Este manual deve ser lido por desenvolvedores de sites, administradores e colaboradores que precisam entender como utilizar o WebSphere Commerce Catalog Manager bem como por outros usuários de negócios que precisam entender suas funções.

Em particular, este manual deve ser lido por Desenvolvedores de Loja e por Administradores de Site do WebSphere Commerce que precisam entender como utilizar as diversas funções do WebSphere Commerce Catalog Manager.

Desenvolvedores de Loja e Administradores de Site do WebSphere Commerce que estão executando extensões programáticas também devem ter conhecimento nas seguintes áreas:



- Tecnologia de Banco de Dados
- Arquitetura do componente do Enterprise JavaBeans™
- Hypertext Markup Language (HTML)
- Java™
- Tecnologia JavaServer Pages™
- VisualAge® for Java™, Enterprise Edition, Versão 3.5 ou posterior
- Extensible Markup Language (XML)

Gerentes de Categorias e de Produtos devem ler este manual para entender como transmitir requisitos de catálogos, de produtos e de itens aos Desenvolvedores de Loja. Os Gerentes de Categorias e de Produtos também devem entender como o Editor da Web pode ser utilizado para atualizar o catálogo da loja.

---

## Onde Localizar mais Informações

Este manual do usuário está disponível em formato PDF (Portable Document Format) da Adobe no site do WebSphere Commerce na Web. Consulte o site na Web WebSphere Commerce Technical Library para obter as versões mais recentes deste documento e outras informações relacionadas ao WebSphere Commerce:

-  [http://www.ibm.com/software/webservers/commerce/wc\\_be/lit-tech-general.html](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](http://www.ibm.com/software/webservers/commerce/wc_pe/lit-tech-general.html)



---

## Parte 1. Visão Geral do Catalog Manager

O WebSphere Commerce Catalog Manager fornece um conjunto de ferramentas genérico que possui várias funções que podem ser agrupadas na seqüência necessária para resolver problemas específicos do gerenciamento de catálogos. É flexível o bastante para tratar de personalizações feitas no esquema do WebSphere Commerce.

O Catalog Manager fornece a capacidade de agregar informações de várias fontes em um sistema WebSphere Commerce consolidado e de remapear todos esses diferentes dados em um formato padrão de definição de catálogos e de produtos utilizando arquivos XML como o recurso padrão de gerenciar informações.

O Catalog Manager fornece os recursos para o seguinte:

- Importar dados de várias origens de entrada no formato de arquivos ASCII e XML para o WebSphere Commerce
- Transformar dados de formato ASCII em XML e vice-versa
- Remapear dados de um formato XML para outro
- Agregar dados de vários fluxos de entrada em um banco de dados agregado
- Criar/editar/excluir dados utilizando a interface de um navegador da Web

O Catalog Manager inclui:

- Loader package do Catalog Manager

Este pacote consiste principalmente em utilitários de comandos para preparar e carregar dados em um banco de dados do WebSphere Commerce. Você pode utilizar o Loader package para carregar grandes quantidades de dados e para atualizar dados no banco de dados do WebSphere Commerce.

O Loader package também permite:

- Extrair dados de um banco de dados como um documento XML
- Transformar dados XML em formatos XML alternativos
- Transformar dados entre um formato variável delimitado por caracteres e um formato de dados XML

- Ferramentas administrativas do Catalog Manager

O Catalog Manager também inclui as duas seguintes ferramentas com uma interface com o usuário para ajudar na administração de suas funções:

- Ferramenta de Transformação de Texto
- Editor XSL

- Editor da Web do Catalog Manager

O Editor da Web permite criar, excluir e fazer alterações nos dados do catálogo, por meio de um navegador da Web.



---

## Parte 2. Transformando, Carregando e Extraindo Dados



---

## Capítulo 1. Introdução

O Loader package do Catalog Manager consiste principalmente em utilitários para preparar e carregar dados para um banco de dados do WebSphere Commerce. Você pode utilizar o Loader package para carregar grandes quantidades de dados e para atualizar dados no banco de dados do WebSphere Commerce. O Loader package também permite:

- Extrair dados de um banco de dados como um documento XML
- Transformar dados XML em formatos XML alternativos
- Transformar dados entre um formato variável delimitado por caracteres e um formato de dados XML

O Catalog Manager também inclui as duas seguintes ferramentas com uma interface com o usuário para ajudar na administração de suas funções:

- Ferramenta de Transformação de Texto
- Editor XSL

---

### Utilitários

O Loader package do WebSphere Commerce inclui utilitários para preparar e carregar dados para um banco de dados do WebSphere Commerce. Você pode utilizar estes utilitários para carregar grandes e pequenas quantidades de dados, além de atualizar dados no banco de dados do WebSphere Commerce.

O processo de carregamento consiste nas etapas necessárias para mover dados para o banco de dados do WebSphere Commerce:

1. Gerando um DTD utilizando o DTD Generator
2. Resolvendo identificadores nos arquivos de entrada com o ID Resolver
3. Carregando dados utilizando o Loader

O Loader Package também inclui utilitários para extrair dados de um banco de dados como um documento XML e para transformar dados XML em formatos XML alternativos.

O Loader package consiste nos seguintes utilitários:

- **Text Transformer**

O Text Transformer transforma dados ente um formato variável delimitado por caracteres e um formato de dados XML.

Consulte Capítulo 2, “Transformando Texto” na página 11 para obter maiores informações.

- **XML Transformer**

O XML Transformer altera, agrega e remapeia os dados em um documento XML para alternar formatos de XML para utilização por outros usuários ou sistemas, conforme necessário.

Consulte Capítulo 3, “Transformando Dados XML” na página 19 para obter maiores informações.

- **DTD Generator**

O DTD especifica elementos estruturais e definições de marcação que podem ser utilizados em um documento de dados XML. Por exemplo, um DTD pode listar elementos a serem utilizados em um documento e especificar os atributos que cada elemento pode obter.

O DTD Generator é um utilitário que cria um DTD para o Loader utilizar, com base no esquema do banco de dados. Este DTD descreve as tabelas e colunas nas quais o Loader importa dados. O DTD Generator também pode criar um esquema e um documento XML detalhado que pode ser utilizado com o Editor da Web do Catalog Manager.

O DTD Generator gera um DTD com base no banco de dados de destino com o qual os dados devem estar de acordo. Este DTD será utilizado em todo o processo de carregamento. O DTD Generator precisa ser executado somente uma vez.

Consulte Capítulo 4, “Gerando um DTD e um Esquema” na página 25 para obter maiores informações.

- **ID Resolver**

O ID Resolver é um utilitário que gera identificadores para elementos XML que os requerem. Se seu conteúdo XML já fornece identificadores, você não precisa executar a Resolução de IDs.

O ID Resolver atualiza um conjunto de elementos XML com seus identificadores associados. Esta etapa é importante porque os arquivos XML do Loader package são mapeados diretamente para o esquema do banco de dados de destino. Dessa forma, eles devem ter identificadores.

O ID Resolver inclui um relator de erros que gera um documento de exceção se ocorrer um erro.

Consulte Capítulo 5, “Resolvendo Identificadores” na página 29 para obter maiores informações.

- **Loader**

O Loader utiliza um XML válido e bem formado como entrada para carregar dados para o banco de dados. Os elementos do documento XML são mapeados para nomes de tabelas no banco de dados, e os atributos de elementos são mapeados para colunas. O Loader é o meio mais comum de carregar dados para um sistema.

O Loader permite atualizações no nível da coluna para uma tabela. Ele também permite excluir dados de um banco de dados.

O exemplo a seguir mostra um segmento de entrada XML válida e bem formada para o Loader:

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

No exemplo acima, ADDRBOOK é o nome da tabela e as colunas a serem atualizadas são indicadas por atributos do elemento ADDRBOOK.

O utilitário Loader inclui os seguintes recursos:

- **Relator de erros**

O Loader inclui um relator de erros que gera um documento de exceção se ocorrer um erro.

- **Sincronização do espaço de procura do Consultor de Produto**

Os espaços de pesquisa do Consultor de Produto são criados pela extração de informações do catálogo do WebSphere Commerce e pela representação dessas informações em um formato adequado para pesquisa. Os espaços de pesquisa e o catálogo se tornam não sincronizados quando o catálogo é atualizado. Para evitar atraso na sincronização de espaços de pesquisa com o catálogo, você pode ativar o recurso de sincronização de espaço de pesquisa do Consultor de Produto do Loader.

Se a sincronização do espaço de pesquisa do Consultor de Produto for ativada, o Loader programará os comandos do espaço de pesquisa apropriados, incluindo as informações de comandos na tabela do programador de jobs SCHCONFIG e na tabela de instâncias do job do programador SCHSTATUS, conforme indicado na seguinte tabela.

Tabela	Modo	Comando de Espaço de Pesquisa	Condição
CATENTRY	Atualizar	UpdateSearchSpaces	Atualizar sinalizador de publicação
	Excluir	RemoveProductFromAllSearchSpaces	Excluir produto
CATENTDESC	Incluir	UpdateSearchSpaces	Incluir descrição/idioma
	Atualizar	UpdateSearchSpaces	Atualizar sinalizador de publicação Atualizar dados da descrição
	Excluir	UpdateSearchSpaces	Remover descrição/idioma
LISTPRICE	Incluir	UpdateSearchSpaces	Incluir preço da lista
	Atualizar	UpdateSearchSpaces	Atualizar preço da lista
	Excluir	UpdateSearchSpaces	Excluir preço da lista
ATTRVALUE	Incluir	UpdateSearchSpaces	Incluir valor definido pelo usuário
	Atualizar	UpdateSearchSpaces	Atualizar valor definido pelo usuário
	Excluir	UpdateSearchSpaces	Excluir valor definido pelo usuário
CATENTATTR	Incluir	UpdateSearchSpaces	Incluir atributo
	Atualizar	UpdateSearchSpaces	Atualizar atributo
	Excluir	UpdateSearchSpaces	Excluir atributo
CATGPENREL	Incluir	AddProductsToSearchSpace	Inclui produto na categoria
	Excluir	RemoveProductsFromSearchSpace	Remover produto da categoria

Consulte Capítulo 6, “Carregando Dados” na página 45 para obter maiores informações.

- **Extractor**

O Extractor utiliza uma consulta em um banco de dados para extrair subconjuntos de dados selecionados do banco de dados em um documento XML. Os dados a serem extraídos do banco de dados são especificados utilizando um documento documento XML de filtro de extração.

A função do Extractor é oposta à do Loader. Utilize o Extractor para extrair subconjuntos de dados seletivos do banco de dados do WebSphere Commerce no formato de arquivos XML. Você pode extrair dados sobre produtos relacionados em um feriado próximo, por exemplo, pode extrair informações de um banco de dados consolidado para utilização com outros sistemas.

Consulte Capítulo 7, “Extraindo Dados” na página 63 para obter maiores informações.

O Loader package também tem um recurso logger. Cada utilitário no Loader package cria mensagens para indicar êxito, falha e erros, bem como para fornecer informações de rastreamento do programa.



---

## Ferramentas Administrativas

O Catalog Manager inclui as seguintes ferramentas para ajudar na administração de suas funções:

### **Ferramenta de Transformação de Texto**

A ferramenta de Transformação de Texto ajuda o administrador a preparar as informações necessárias para processar uma transformação de dados entre um formato variável delimitado por caracteres e um formato de dados XML.

### **Editor XSL**

O XML Transformer utiliza XSL (Extensible Stylesheet Language) para definir as regras para transformar um arquivo XML em outro arquivo XML. A função de mapeamento no Editor XSL fornece ao administrador uma interface visual com a qual é estabelecida a associação entre um elemento em uma definição de tipo de documento (DTD) de origem em um elemento em uma DTD de destino.



---

## Capítulo 2. Transformando Texto






O Catalog Manager fornece a capacidade de converter saída de arquivo ASCII de outras ferramentas (como programas de planilha) em um formato de dados XML que pode ser inserido no banco de dados do WebSphere Commerce.


A ferramenta de Transformação de Texto prepara as informações necessárias para processar uma transformação de dados entre um formato variável delimitado por caracteres e um formato de dados XML. São fornecidas as seguintes exibições:

1. A Exibição de Edição de Esquema de Texto permite a criação do arquivo do esquema XML a ser utilizado em uma transformação.
2. A Exibição de Edição de Comando de Transformação permite criar e modificar os comandos reais utilizados para executar o processo de transformação.
3. A Exibição do Processo de Comando de Transformação permite lançar o processo de transformação.

## Lançando a Ferramenta de Transformação de Texto

Para lançar a ferramenta de Transformação de Texto, utilize o script ou comando apropriado fornecido no diretório do WebSphere Commerce:

-  *unidade:*\WebSphere\CommerceServer\bin\TextTrans.cmd
-  *unidade:*\Arquivos de Programas\WebSphere\CommerceServer\bin\TextTrans.cmd
-  /usr/WebSphere/CommerceServer/bin/texttrans.sh
-   /opt/WebSphere/CommerceServer/bin/texttrans.sh

 Em ambientes do iSeries, os administradores devem primeiro copiar os arquivos necessários para executar a ferramenta de Transformação de Texto para as máquinas do Windows NT ou Windows 2000. A seguir, um exemplo de como fazer isso:

1. Crie um novo diretório na máquina do Windows NT (*unidade:*\TextTrans por exemplo).
2. Crie os seguintes novos subdiretórios em *unidade:*\TextTrans:
  - \bin
  - \lib\loader
  - \wcsadmin
3. Copie os arquivos da máquina do iSeries para estes diretórios, de acordo com o seguinte quadro:

De		Para
/QIBM/ProdData/WebCommerce		<i>unidade:</i> \TextTrans
/bin		\bin
/TextTrans.cmd	----->	\TextTrans.cmd
/lib/loader		\lib\loader
/wcmxmp.jar	----->	\wcmxmp.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. Modifique o arquivo TextTrans.cmd na máquina do Windows, inserindo o seguinte texto:  

```
set WCS_HOME=unidade:\TextTrans
```

  
antes do seguinte:  

```
set lib=%WCS_HOME%\lib\loader
```
5. Inicie a ferramenta de Transformação de Texto executando TextTrans.cmd na máquina do Windows.

---

## Utilizando a Exibição de Edição de Esquema de Texto

Os procedimentos a seguir estão relacionados à criação e modificação de um arquivo do esquema XML para transformar dados entre um formato variável delimitado por caracteres e um formato XML.

### Criando um Arquivo Esquema

Para criar um novo arquivo esquema, faça o seguinte:

1. Lance a ferramenta de Transformação de Texto.
2. Selecione **Arquivo > Novo** ou clique no ícone Novo na barra de ferramentas.
3. Selecione um caminho e digite o nome do arquivo do esquema XML a ser criado.

**Nota:** O nome do arquivo padrão é "Default.xml."

4. Selecione **Formato CSV**.

**Nota:** Não selecione **Formato do WebSphere Commerce Suite**. Esta opção é utilizada somente com versões mais anteriores do WebSphere Commerce.

5. Clique em **OK** quando terminar.

Agora você pode criar novos elementos e atributos seguindo as etapas indicadas abaixo.

### Abrindo um Arquivo Esquema

Para abrir um arquivo esquema, faça o seguinte:

1. Lance a ferramenta de Transformação de Texto.
2. Selecione **Arquivo > Abrir** ou clique no ícone Abrir na barra de ferramentas.
3. Selecione um arquivo esquema a ser aberto.
4. Clique em **OK** quando terminar.

Agora você pode modificar os elementos e atributos seguindo as etapas descritas abaixo.

### Salvando um Arquivo Esquema

Para salvar um arquivo esquema, faça o seguinte:

1. Lance a ferramenta de Transformação de Texto.
2. Para salvar as alterações feitas no esquema, selecione **Arquivo > Salvar** ou clique no ícone Salvar na barra de ferramentas.
3. Para salvar uma cópia do esquema com um novo nome, faça o seguinte:
  - a. Selecione **Arquivo > Salvar Como**.
  - b. Selecione um caminho e digite o nome do arquivo XML a ser criado.
  - c. Clique em **OK** quando terminar.

### Editando um Arquivo Esquema

Para editar um arquivo esquema, abra-o conforme descrito acima e siga esses procedimentos.

#### Incluindo um Elemento

Para incluir um elemento, faça o seguinte:

1. Digite um novo nome de elemento no campo Lista de Elementos.

2. Clique no ícone Incluir linha.

### **Removendo um Elemento**

Para remover um elemento, faça o seguinte:

1. Selecione um nome de elemento.
2. Clique no ícone Remover linha.

### **Substituindo um Elemento**

Para substituir um elemento, faça o seguinte:

1. Digite um novo nome de elemento no campo Lista de Elementos.
2. Selecione o nome do elemento a ser substituído.
3. Clique no ícone Substituir linha.

### **Movendo um Elemento uma Linha Acima**

Para mover um elemento uma linha acima, faça o seguinte:

1. Selecione o nome do elemento a ser movido para cima.
2. Clique no ícone Mover linha acima.

### **Movendo um Elemento uma Linha Abaixo**

Para mover um elemento uma linha abaixo, faça o seguinte:

1. Selecione o nome do elemento a ser movido para baixo.
2. Clique no ícone Mover linha abaixo.

### **Incluindo um Atributo**

Para incluir um atributo, faça o seguinte:

1. Digite um novo nome de atributo no campo Lista de Atributos.
2. Clique no ícone Incluir linha.

### **Removendo um Atributo**

Para remover um atributo, faça o seguinte:

1. Selecione um nome de atributo.
2. Clique no ícone Remover linha.

### **Substituindo um Atributo**

Para substituir um atributo, faça o seguinte:

1. Digite um novo nome de atributo no campo Lista de Atributos.
2. Selecione o nome do atributo a ser substituído.
3. Clique no ícone Substituir linha.

### **Movendo um Atributo uma Linha Acima**

Para mover um atributo uma linha acima, faça o seguinte:

1. Selecione o nome do atributo a ser movido para cima.
2. Clique no ícone Mover linha acima.

### **Movendo um Atributo uma Linha Abaixo**

Para mover um atributo uma linha abaixo, faça o seguinte:

1. Selecione o nome do atributo a ser movido para baixo.
2. Clique no ícone Mover linha abaixo.

## Alterando a Estrutura de um Arquivo Esquema

A exibição Estrutura de Arquivo no painel inferior da Exibição de Edição do Esquema de Texto descreve o layout de um arquivo variável delimitado por caracteres. Os campos a seguir são obrigatórios na estrutura de arquivo esperada:

### Separador de Campo:

Especifica o delimitador que separa valores de atributos. O padrão é uma vírgula (",").

### Separador de Registro:

Especifica o delimitador que separa registros de dados. O padrão é "&#10;&#13;" (É equivalente a uma referência de entidade a `\r\n`).

### Delimitador de Cadeia:

Especifica o delimitador que indica os limites de início e fim de cadeia. O padrão é uma aspa simples (" ' ").

### Cabeçalho Incluído:

Um valor Boolean, especificado como "true" se houver uma linha de cabeçalho no arquivo de dados de texto e "false" se não houver uma linha de cabeçalho no arquivo de dados de texto. Se houver uma linha de cabeçalho, ela deve estar de acordo com as regras de XML para um nome de tag porque o cabeçalho será utilizado como o nome da tag XML neste caso. O padrão é "false."

### Número de linhas de cabeçalho:

Especifica o número de linhas do cabeçalho existentes no arquivo de dados de texto. O padrão é zero ("0").

## Preparando um Arquivo do Esquema para Transformar Dados de XML em um Formato Variável Delimitado por Caracteres

Se você estiver utilizando um arquivo esquema XML que foi criado utilizando a Exibição da Edição do Esquema de Texto para transformar dados do formato XML em um formato variável delimitado por caracteres, utilize um editor de texto para alterar a especificação de tipo de dados no arquivo esquema do "Formato CSV" para o "Formato XML" antes de processar a transformação.

---

## Utilizando a Exibição de Edição de Comando de Transformação

Utilizando a Exibição de Edição do Comando de Transformação, você pode criar um novo arquivo de comando, abrir um arquivo de comando já existente ou salvar alterações em um arquivo de comando. O nome do arquivo de comando padrão é "Manifest.txt".

Você pode criar um novo comando, remover ou substituir um comando com informações editadas ou alterar a ordem de um comando.

**Nota:** O arquivo de comando é salvo automaticamente sempre que a tabela de comandos é atualizada.

### Criando um Comando

Para criar um novo comando, faça o seguinte:

1. Especifique o arquivo fonte, um arquivo de formato variável delimitado por caracteres (com extensão .csv) ou um arquivo de formato XML (com extensão .xml).
2. Especifique o arquivo esquema XML a ser utilizado na transformação.
3. Especifique o nome do arquivo de saída a ser criado ou modificado durante o processo de transformação (isto é, onde os novos dados serão armazenados), no formato XML (com extensão .xml) ou no formato variável delimitado por caracteres (com extensão .csv).
4. Especifique um modo de comando.  
Selecione **Criar** se o arquivo de saída for criado ou **Anexar** se os dados de saída forem anexados a um arquivo de dados já existente.
5. Clique no ícone Incluir linha.

### Removendo um Comando

Para remover um comando, faça o seguinte:

1. Selecione um comando.
2. Clique no ícone Remover linha.

### Editando ou Substituindo um Comando

Para editar ou substituir um comando, faça o seguinte:

1. Clique no ícone Editar comando.  
Os dados da linha são preenchidos para os campos de entrada apropriados.
2. Altere o texto nos campos de entrada apropriados.
3. Clique no ícone Substituir linha para atualizar a linha.

### Movendo um Comando uma Linha Acima

Para mover um comando uma linha acima, faça o seguinte:

1. Selecione um comando.
2. Clique no ícone Mover linha acima.

**Nota:** Isto altera a seqüência do processo de transformação.



## Movendo um Comando uma Linha Abaixo

Para mover um comando uma linha abaixo, faça o seguinte:

1. Selecione um comando.
2. Clique no ícone Mover linha abaixo.

**Nota:** Isto altera a seqüência do processo de transformação.

## Limpendo um Comando

Para limpar um comando, faça o seguinte:

1. Selecione um comando.
2. Clique no ícone Limpar campos de entrada.

Isto limpa o texto nos campos arquivo fonte, arquivo esquema e arquivo de saída.

---

## Utilizando a Exibição de Processo de Transformação

Para lançar o processo de transformação de texto, faça o seguinte:

1. Digite ou procure o nome do arquivo de parâmetros no campo Arquivo.
2. Clique em **Processar**.

A área de saída abaixo do botão Processar mostra informações que indicam o status do processo de transformação. Você pode salvar as informações de saída clicando no botão Salvar na parte inferior da área de texto ou pode limpar todas as informações de status clicando no botão Limpar.



---

## Capítulo 3. Transformando Dados XML

O XSL (Extensible Stylesheet Language) fornece:

1. Linguagem para especificar a formatação de documentos XML
2. Linguagem que descreve como transformar um arquivo XML em outro arquivo regularmente estruturado

A capacidade de transformação de XSL pode ser utilizada para transformar um arquivo XML em outro arquivo XML que esteja de acordo com um esquema ou DTD XML.

Para transformar um arquivo XML em um formato XML substituto, é necessário especificar as regras para a transformação utilizando um arquivo de transformação de regras XSL.

O exemplo a seguir transforma os dados em MemberSubsystemExtracted.xml utilizando MemberSubsystem.xsl como o arquivo de transformação de regras XSL com japonês como o idioma nacional:



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









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

O XML Transformer utiliza XSL para definir as regras para transformação de um arquivo XML em outro arquivo XML. A função de mapeamento no Editor XSL oferece uma interface visual com a qual você pode estabelecer a associação entre um elemento em um DTD de origem e um elemento em um DTD de destino. Com dois DTDs, você pode desenvolver regras de XSL que determinam como um arquivo XML que está de acordo com o primeiro DTD (origem) é transformado em um arquivo que está de acordo com o segundo DTD (destino).

## Lançando o Editor XSL

Para lançar o Editor XSL, utilize o script ou o comando apropriados, fornecidos no diretório do WebSphere Commerce:

-  `unidade:\WebSphere\CommerceServer\bin\XSLEditor.cmd`
-  `unidade:\Arquivos de Programas\WebSphere\CommerceServer\bin\XSLEditor.cmd`
-  `/usr/WebSphere/CommerceServer/bin/xsleditor.sh`
-   `/opt/WebSphere/CommerceServer/bin/xsleditor.sh`

 Em ambientes do iSeries, os administradores devem primeiro copiar os arquivos necessários para executar o Editor de XSL para as máquinas do Windows NT ou Windows 2000. A seguir, um exemplo de como fazer isso:

1. Crie um novo diretório na máquina do Windows (*unidade:\XMLTrans* por exemplo).
2. Crie os seguintes novos subdiretórios em *unidade:\TextTrans*:
  - \bin
  - \lib\loader
  - \wcsadmin
3. Copie os arquivos da máquina do iSeries para estes diretórios, de acordo com o seguinte quadro:

De		Para
/QIBM/ProdData/WebCommerce		unidade:\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. Modifique o arquivo XSLEditor.cmd na máquina do Windows inserindo o seguinte texto:

```
set WCS_HOME=unidade:\XMLTrans
```

antes do seguinte:

```
set lib=%WCS_HOME%\lib\loader
```
5. Lance a ferramenta XMLTransformation, executando XSLEditor.cmd na máquina do Windows.

---

## Trabalhando com Áreas de Construção da Regra de Mapeamento

Quando o Editor XSL é lançado, ele exibe a janela Área de Construção da Regra de Mapeamento. Utilize esta janela para gerenciar suas áreas de construção da regra de mapeamento.

### Criando uma Área de Construção da Regra de Mapeamento

Para criar uma área de construção da regra de mapeamento, faça o seguinte:

1. Lance o Editor XSL.
2. No menu drop-down, selecione **[Novo]**.
3. No campo Nome, digite um nome para a nova área de construção da regra de mapeamento.
4. No campo Descrição, digite uma descrição resumida da nova área de construção da regra de mapeamento.
5. No campo Esquema de Origem, digite o nome de um arquivo existente ou navegue para um arquivo existente a ser utilizado como o esquema de origem.
6. No campo Esquema de Destino, digite o nome de um arquivo existente ou navegue para um arquivo existente a ser utilizado como o esquema de destino.
7. No campo Arquivo de Regras XSL, digite um nome para o novo arquivo de regras a ser criado.  
Você pode especificar um caminho completo aqui. Se não for fornecido um caminho, o arquivo será criado no diretório de trabalho atual.
8. Clique em **Abrir** para criar e abrir a nova área de construção da regra de mapeamento.

### Modificando uma Área de Construção da Regra de Mapeamento

Para modificar uma área de construção da regra de mapeamento, faça o seguinte:

1. Lance o Editor XSL.
2. Selecione a área de construção da regra de mapeamento que deseja modificar no menu drop-down.
3. Clique em **Abrir** para abrir a área de construção da regra de mapeamento.
4. Atualize os campos que deseja alterar.
5. Clique em **Salvar** para salvar suas alterações.

### Excluindo uma Área de Construção da Regra de Mapeamento

Para excluir uma área de construção da regra de mapeamento, faça o seguinte:

1. Lance o Editor XSL.
2. Selecione a área de construção da regra de mapeamento que deseja excluir no menu drop-down.
3. Clique no botão Remover para remover a entrada.

**Nota:** A remoção de uma área de construção da regra de mapeamento não exclui os arquivos físicos do disco.

---

## Utilizando o Editor XSL

Quando você abre uma área de construção da regra de mapeamento utilizando o Editor XSL, essa área é exibida na janela principal do Editor XSL.

Na janela principal do Editor XSL, o painel esquerdo mostra uma exibição hierárquica do DTD de origem rotulada de "Esquema de Origem." O painel direito mostra uma exibição hierárquica do DTD de destino rotulada de "Esquema de Destino."

### Criando uma Regra de Mapeamento

#### Mapeando Elemento a Elemento

Selecione e arraste um elemento da hierarquia de origem e solte-o em um elemento na hierarquia de destino. Uma regra XSL é gerada e mostrada na exibição Regra de Mapeamento localizada na parte inferior da janela.

A seguir está um exemplo de uma regra XSL gerada:

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

**Nota:** Quaisquer relacionamentos de ancestrais obrigatórios mas não existentes são gerados automaticamente.

#### Mapeando Atributo a Atributo

Selecione e arraste um atributo da hierarquia de origem e solte-o em um atributo na hierarquia de destino. Uma regra XSL é gerada e mostrada na exibição Regra de Mapeamento localizada na parte inferior da janela.

A seguir está um exemplo de uma regra XSL gerada:

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

**Nota:** Quaisquer relacionamentos de ancestrais obrigatórios mas não existentes são gerados automaticamente.

#### Criando uma Expressão de Mapeamento Personalizada

Para criar uma expressão de mapeamento personalizada, primeiro selecione um elemento ou atributo na hierarquia de destino. Em seguida, clique com o botão direito do mouse e selecione o menu Criar Expressão Personalizada. A janela Criar Expressão Personalizada é exibida com uma lista de Modelos e Expressões de Regras disponíveis em dois menus drop-down. Conclua a expressão personalizada fazendo o seguinte:

1. Selecione um modelo no qual a expressão personalizada deve ser incluída.
2. Selecione uma expressão de regra a ser criada (**Expressão Constante** por exemplo).
3. Digite um valor na coluna Valor para cada parâmetro listado na tabela e pressione **Enter** para consolidar o valor.
4. Clique em **OK** para concluir a etapa de criação ou clique em **Cancelar** para cancelar sem criar uma regra.

A regra XSL gerada é baseada na expressão personalizada definida no arquivo de configuração de regras (XSLRuleConfig.xml). Você pode modificar o arquivo de configuração de regras e incluir novas regras, se necessário. Para tornar uma regra disponível para utilização na lista de Expressões de Regras, defina o atributo Visibility para essa regra como "true."

## Excluindo uma Regra de Mapeamento

Para excluir uma regra de mapeamento, faça o seguinte:

1. Selecione uma regra da tabela de Regras de Mapeamento.
2. Clique com o botão direito do mouse e selecione **Excluir**.

A regra e todos os seus descendentes são excluídos.

**Nota:** As regras de mapeamento atualizadas e as regras XSL geradas são persistidas automaticamente.

## Processando uma Transformação de XML

Para processar uma transformação de XML, faça o seguinte:

1. Selecione **Ferramentas > Transformar** para exibir a janela Processar Transformação.
2. Preencha os campos obrigatórios:
  - a. No campo Arquivo XML de Entrada, digite ou procure o caminho e o nome do arquivo de dados XML de origem.
  - b. No campo Arquivo de Regras XSL, digite ou procure o caminho e o nome do arquivo de regras de mapeamento a ser utilizado para a transformação. Se uma área de construção da regra de mapeamento estiver aberta, este campo já estará preenchido com o caminho do arquivo de regras de mapeamento atualmente aberto na área de construção da regra de mapeamento.
  - c. No campo Arquivo XML de Saída, digite ou procure um caminho e nome para o novo arquivo de dados XML a ser criado durante o processo de transformação.
3. Clique em **Iniciar** para iniciar o processo de transformação de XML; ou clique em **Fechar** para sair da janela sem processar uma transformação.

## Personalizando a Tabela de Regras de Mapeamento

Para personalizar a tabela de Regras de Mapeamento, faça o seguinte:

1. Para ocultar uma coluna na tabela, clique com o botão direito do mouse em uma célula na tabela e selecione **Ocultar coluna**.
2. Para exibir uma coluna na tabela, faça o seguinte:
  - a. Clique com o botão direito do mouse na tabela.
  - b. Selecione **Exibir colunas** para tornar visível a lista de colunas ocultas.
  - c. Selecione a coluna da lista.

**Nota:** Para selecionar várias colunas, mantenha pressionada a tecla **Shift** e, em seguida, clique nos nomes de colunas.

- d. Clique em **OK** para exibir as colunas selecionadas ou em **Cancelar** para cancelar a operação.
3. Para exibir todas as colunas ocultas na tabela, clique com o botão direito do mouse em uma célula na tabela e selecione **Exibir todas as colunas**.

Todas as colunas são exibidas na ordem padrão.

## **Exibindo uma Regra XSL Completa/Expressão de Valor**

Nas colunas Expressão de Valor ou Regra XSL, clicar em uma célula torna visível uma janela com o conteúdo completo da regra para a linha selecionada.



## Capítulo 4. Gerando um DTD e um Esquema

O DTD Generator pode criar um DTD e um esquema para utilização com o Loader package. O DTD Generator utiliza um arquivo de entrada contendo nomes de tabelas do banco de dados e gera um DTD ou um DTD e um esquema com um arquivo XML detalhado que descreve o banco de dados, dependendo de como você chama o comando DTD Generate.

### Configurando o DTD Generator

O DTD do Loader é mapeado diretamente para o esquema do banco de dados do WebSphere Commerce. Cada tabela é um elemento e cada coluna um atributo.

**Exemplo: Mapeando um DTD do Loader para um esquema do banco de dados**

Instruções DDL para a tabela 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"
);	>

Você pode definir a forma de funcionamento do DTD Generator fazendo o seguinte:

1. Crie um novo arquivo de propriedades do personalizador do DTD Generator.



DB2ConnectionCustomizer.properties está localizado no archive DTDGenerator.zip. Extraia esse arquivo, renomeie-o mas mantenha a extensão .properties e coloque-o em um diretório que esteja no classpath.

**Importante:** Não remova nem modifique o arquivo DB2ConnectionCustomizer.properties existente.

• **400**

ISeries\_GENWCSDTD\_Customizer.properties está localizado no diretório /QIBM/ProdData/WebCommerce/properties. Copie este arquivo para o diretório /instroot/xml, renomeie o novo arquivo, mas mantenha a extensão .properties; em seguida, faça as alterações necessárias no novo arquivo.

**Importante:** Não remova ou modifique o arquivo ISeries\_GENWCSDTD\_Customizer.properties original.

2. Modifique os valores do driver do banco de dados no novo arquivo. Por exemplo:

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

em que:

- DBVendorName é utilizado para selecionar o tipo de banco de dados.

As opções são as seguintes:

- DB2 Universal Database para iSeries (DB2/iSeries)
- DB2 para outros sistemas operacionais (DB2)
- Banco de dados Oracle (Oracle)

- DBDriverName é utilizado para selecionar o driver JDBC.

As opções são as seguintes:

- DB2 Universal Database para iSeries (com.ibm.db2.jdbc.app.DB2Driver )
- DB2 para outros sistemas operacionais (COM.ibm.db2.jdbc.app.DB2Driver)
- Banco de dados Oracle (oracle.jdbc.driver.OracleDriver)

- DBURL é utilizado para especificar a URL para acessar o banco de dados.

As opções são as seguintes:

- DB2 Universal Database para iSeries (jdbc:db2://)
- DB2 para outros sistemas operacionais (jdbc:db2:)
- Banco de dados Oracle (jdbc:oracle:oci8:@)

3. Especifique o novo nome do arquivo como o valor do parâmetro customizer do comando DTD Generate.

---

## Gerando um DTD

O arquivo de entrada TableNames.txt contém os seguintes nomes de tabelas de bancos de dados, um em cada linha:

```
MEMBER
ADDRBOOK
ADDRESS
```

A seguir está um exemplo de como o DTD Generator pode ser chamado:

- 

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

- **400**

```
QWEBCOMM/GENWCSDTD DATABASE(DATABASE_NAME) SCHEMA(MALL)
INSTROOT(/QIBM/UserData/WebCommerce/instances/mser)
PASSWD(my password) OUTFILE(wc.dtd) INFILE(TableNames.txt)
```

O arquivo de saída wc.dtd contém o seguinte:

```

<!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
>

```

## Gerando um Esquema e um Arquivo XML Detalhado

Neste exemplo, o DTD Generator é chamado da seguinte forma:

• 

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

• 

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

• 

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

O arquivo esquema é criado no diretório `sample` e tem o nome de arquivo `WCAWebForm.xsd`. O arquivo de saída `sample.xml` contém o seguinte:

```
<?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>
```

---

## Capítulo 5. Resolvendo Identificadores

Dados XML a serem carregados em um banco de dados de destino precisam conter identificadores para elementos XML que os requerem. Para gerar ou localizar identificadores para entidades do catálogo no documento XML, chame o comando ID Resolve.

O ID Resolver somente resolve identificadores para uma tabela primária. Uma tabela principal é a que está listada na tabela KEYS ou SUBKEYS. Se for necessário resolver identificadores para uma tabela que não está em KEYS ou SUBKEYS, inclua a tabela na tabela SUBKEYS antes de executar o ID Resolver.

Estes são exemplos de situações em que você pode utilizar o ID Resolver:

- Ao carregar novo conteúdo em formato XML quando são necessários identificadores para os dados
- Ao atualizar o conteúdo quando já existem identificadores para um objeto no banco de dados

O ID Resolver pode fornecer identificadores reais ou os identificadores podem ser resolvidos utilizando as seguintes técnicas:

- Resolução de alias interno
- Especificação de arquivo de propriedades
- Resolução de índice exclusivo

---

### Configurando o ID Resolver

Você pode definir como o ID Resolver trata data e hora, armazenamento e drivers do banco de dados fazendo o seguinte:

1. Crie um novo arquivo de propriedades do personalizador de ID Resolver.



• DB2ConnectionCustomizer.properties está localizado no archive IdResGen.zip. Extraia esse arquivo, renomeie-o mas mantenha a extensão .properties e coloque-o em um diretório que esteja no classpath.

**Importante:** Não remova e nem modifique o arquivo DB2ConnectionCustomizer.properties já existente.



• ISeries\_RESWCSID\_Customizer.properties está localizado no diretório /QIBM/ProdData/WebCommerce/properties. Copie este arquivo para o diretório /instroot/xml, renomeie o novo arquivo, mas mantenha a extensão .properties; em seguida, faça as alterações necessárias no novo arquivo.

**Importante:** Não remova ou modifique o arquivo ISeries\_RESWCSID\_Customizer.properties original.

2. Modifique os valores das propriedades especificadas no novo arquivo.
3. Especifique o novo nome do arquivo como o valor do parâmetro customizer do comando ID Resolve.

## Definindo como o ID Resolver Trata Data e Hora

As máscaras de data e hora de entrada padrão a seguir são fornecidas no arquivo de propriedades do personalizador do ID Resolver:

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

Você pode modificar estas máscaras de data e hora ou incluir quantas máscaras desejar no arquivo de propriedades do personalizador do ID Resolver. Se você incluir uma data e hora de entrada, será necessário utilizar o número seguinte na seqüência atual. (Por exemplo, a máscara de data e hora seguinte seria `InputTimeStampFormat.8` se você estivesse incluindo na lista acima).

Você também pode personalizar o formato de data e hora de saída, a máscara de microssegundos e o formato específico do banco de dados modificando os valores das seguintes propriedades no arquivo de propriedades do personalizador do ID Resolver:

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

## Definindo como o ID Resolver Trata Armazenamento

Esta é a seção no arquivo de propriedades do personalizador do ID Resolver que especifica os valores padrão para as propriedades relacionadas aos hashmaps persistentes:

```
////////////////////////////////////
/// 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
```

Você pode especificar como o ID Resolver trata armazenamento persistente definindo um valor para `PersistentStorageType` no arquivo de propriedades do personalizador de ID Resolver.

- Se você definir `PersistentStorageType = 0`, o ID Resolver funcionará no modo "normal" (onde os hashmaps de símbolo existem na memória).
- Se você definir `PersistentStorageType = 1`, JDBM será utilizado para persistir os símbolos e chaves.

Você pode especificar o número de registros armazenados na memória definindo um valor para `MemoryStorageSize` no arquivo de propriedades do personalizador de ID Resolver.

- Um valor de "1" para `MemoryStorageSize` indica que somente um registro é guardado na memória.
- Um valor de "-1" para `MemoryStorageSize` tem um significado especial, indicando que todos os registros são guardados na memória.

Neste caso, o ID Resolver é revertido para seu comportamento "normal".

## Definindo como o ID Resolver Trata Drivers do Banco de Dados

As linhas a seguir no arquivo de propriedade do personalizador de ID Resolver especificam os valores padrão para os drivers do banco de dados:

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

em que:

- `DBVendorName` é utilizado para selecionar o tipo de banco de dados. As opções são as seguintes:
  - DB2 Universal Database para iSeries (DB2/iSeries)
  - DB2 para outros sistemas operacionais (DB2)
  - Banco de dados Oracle (oracle)
- `DBDriverName` é utilizado para selecionar o driver JDBC. As opções são as seguintes:
  - DB2 Universal Database para iSeries (com.ibm.db2.jdbc.app.DB2Driver )
  - DB2 para outros sistemas operacionais (COM.ibm.db2.jdbc.app.DB2Driver)
  - Banco de dados Oracle (oracle.jdbc.driver.OracleDriver)
- `DBURL` é utilizado para especificar a URL para acessar o banco de dados. As opções são as seguintes:
  - DB2 Universal Database para iSeries (jdbc:db2://)
  - DB2 para outros sistemas operacionais (jdbc:db2:)
  - Banco de dados Oracle (jdbc:oracle:oci8:@)

---



## Determinando como Processar Dados

O comando ID Resolve permite escolher o método load, update ou mixed para processar o arquivo de entrada.

- Utilize o método load para processar o arquivo de entrada se *todos* os registros no arquivo *não existirem* no banco de dados.
- Utilize o método update para processar o arquivo de entrada se *todos* os registros no arquivo *existirem* no banco de dados.
- Utilize o método mixed para processar o arquivo de entrada se *apenas alguns* registros no arquivo *existirem* no banco de dados.

## Escolhendo o Método Load

O método load para o ID Resolver é utilizado para gerar novos identificadores para os registros que são carregados no banco de dados. Com este método, novos identificadores são criados para os registros. O exemplo a seguir é utilizado para gerar identificadores para novos dados:

-    


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

O método load é o padrão.

-     

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



O método load é o padrão.

-   

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

## Escolhendo o Método Update


Se você especificar o método update para o ID Resolver, os registros no arquivo de entrada já devem existir no banco de dados. O ID Resolver localiza os identificadores no banco de dados. Se não existir um registro no banco de dados, o ID Resolver não poderá resolver o identificador desse registro e ele indicará que um erro ocorreu. O exemplo a seguir é utilizado para localizar identificadores que já existem no banco de dados:

-    

```
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/RESWC SID DATABASE(DATABASE_NAME) SCHEMA(MALL)  
INSTROOT(/QIBM/UserData/WebCommerce/instances/mser) PASSWD(my password)  
INFILE(input.xml) OUTFILE(output.xml) METHOD(*UPD)
```

## Escolhendo o Método Mixed

Se o arquivo de dados de entrada contiver registros já existentes no banco de dados e também alguns registros novos, o ID Resolver deverá ser executado utilizando o método mixed. Com este método, o ID Resolver cria novos identificadores para registros somente se os registros não existirem no banco de dados. Caso contrário, o identificador existente é obtido do banco de dados. O exemplo a seguir é utilizado para gerar identificadores para novos dados e para localizar identificadores para dados que já existem no banco de dados:

-  



```
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
```

• 

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

## Utilizando Técnicas de Resolução de ID

O ID Resolver pode utilizar um arquivo de propriedades Java para determinar quais colunas de uma tabela principal devem ser utilizadas como procuras para tabelas que requerem o identificador de uma tabela principal. Uma tabela é principal, se estiver listada na tabela KEYS ou SUBKEYS.

Para utilizar a resolução de alias interno com o ID Resolver, um alias é colocado no atributo de chave principal (identificador) no arquivo XML. O alias pode ser utilizado em todo o arquivo XML para referir-se a esse elemento. Este processo elimina a necessidade de um programa para determinar os índices exclusivos necessários para criar o arquivo XML.

O ID Resolver também pode analisar o esquema do banco de dados para determinar se existe um índice exclusivo que preencha seus requisitos. Esse utilitário procura um índice exclusivo somente quando não existe uma entrada no arquivo de propriedades para a tabela que está sendo analisada ou quando não existe um arquivo de propriedades. Se essas condições são verdadeiras, uma verificação de índice exclusivo é realizada. O índice exclusivo é considerado válido se ele existir e não incluir a chave principal para a tabela.






## Especificando um Arquivo de Propriedades com o ID Resolver

O ID Resolver permite utilizar um arquivo de propriedades Java alternativo para descrever quais colunas de uma entrada principal devem ser utilizadas como procuras para tabelas que requerem o identificador de uma linha principal.

O arquivo de propriedades padrão é `IdResolveKeys.properties`, mas é possível modificá-lo ou especificar seu próprio arquivo ao chamar o comando ID Resolve, se desejar.

•     

`IdResolveKeys.properties` está localizado no seguinte diretório:

-  *unidade:* \WebSphere\CommerceServer\properties
-  *unidade:* \Arquivos de Programas\WebSphere\CommerceServer\properties
-  /usr/WebSphere/CommerceServer/properties
-   /opt/WebSphere/CommerceServer/properties

Se você não colocar este arquivo no diretório atual ao executar o ID Resolver, poderá colocá-lo em um diretório definido na variável de ambiente `classpath`. Você também poderá especificar o caminho completo para esse arquivo.

- ▶ 400

Para alterar `IdResolveKeys.properties`, copie-o do diretório `/QIBM/ProdData/WebCommerce/properties`, salve-o no diretório `/instroot/xml`, em seguida, faça as alterações necessárias no novo arquivo.

**Nota:** O diretório acima é o classpath utilizado pelo comando `RESWCSID`.

## Utilizando um Arquivo de Propriedades para Gerar Identificadores

No exemplo a seguir, é necessário resolver os identificadores ADDRBOOK\_ID e ADDRESS\_ID para os registros ADDRBOOK e ADDRESS, respectivamente. Os identificadores para os registros MEMBER já são conhecidos. Cada registro requer um identificador válido para o banco de dados do WebSphere Commerce. Além disso, ADDRBOOK\_ID no registro ADDRESS requer que o identificador da tabela principal atenda sua restrição de chave externa.

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

Valor real da coluna DISPLAYNAME

Refere-se a ADDRBOOK utilizando DISPLAYNAME como uma exibição

Pode ser necessário um arquivo de propriedades para identificar quais colunas na linha principal serão utilizadas pelas linhas de relacionamento para resolver o identificador para a coluna de chave externa. Será utilizado o seguinte procedimento para assegurar que a análise do arquivo acima seja feita corretamente.

Em `IdResolveKeys.properties`, especifique o seguinte:

```
NAMEDELIMITER=@
SELECTDELIMITER=:

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

NAMEDELIMITER e SELECTDELIMITER definem os delimitadores utilizados em todo o arquivo de propriedades e eles devem ser utilizados consistentemente.

ADDRBOOK=@DISPLAYNAME:DISPLAYNAME afirma que quando um registro do catálogo de endereços for recebido, o identificador da linha do catálogo de endereços será criado. O campo DISPLAYNAME é extraído do registro de entrada e utilizado para formar uma associação com o novo identificador. A cadeia DISPLAYNAME é utilizada para corresponder a linha do catálogo de endereços DISPLAYNAME e resolver o identificador requerido pela chave externa.

Utilizando o exemplo de entrada anterior, no qual DISPLAYNAME é Friends, assumo que o identificador criado para este registro é 12951. O DISPLAYNAME é utilizado como procura de chave para 12951. O processamento continua com o próximo registro, ADDRESS, em que ADDRBOOK\_ID tem o formato de "@..." (que indica que o que segue o delimitador deve ser utilizado para procurar o identificador do catálogo de endereços). A cadeia corresponde DISPLAYNAME, e 12951 é retornado e colocado no atributo 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"
/>

```

**Chave principal gerada**

**Valor de ADDRBOOK DISPLAYNAME não alterado**

**Chave principal gerada**  
**ADDRESS refere-se ao ADDRBOOK correto**

### Utilizando um Arquivo de Propriedades com Chaves Compostas

Uma chave formada por mais de duas colunas é uma chave composta. É possível definir a procura de chave composta no arquivo de propriedades especificando NAMEDELIMITER e SELECTDELIMITER seguidos pelos nomes de campos. Para que os critérios de procura dos registros ADDRBOOK sejam o composto do nome de exibição e do ID do membro, por exemplo, especificando o seguinte no arquivo de propriedades:

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

em seguida, o seguinte fragmento de arquivo de entrada XML:

```

<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" de MEMBER 100**

**ADDRBOOK "Friends" de 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"
/>

```

**Procurar a chave principal para ADDRBOOK "Friends" de MEMBER 100**

resultaria no seguinte após a resolução:

```

<MEMBER
  MEMBER_ID="100"
  TYPE="U"
  STATE="1"
/>
<MEMBER
  MEMBER_ID="101"
  TYPE="U"
  STATE="1"
/>
<ADDRBOOK
  ADDRBOOK_ID="12951" ADDRBOOK de interesse
  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"
/>

```

**ADDRESS refere-se ao ADDRBOOK correto**

## Utilizando o Arquivo de Propriedades com Chaves Principais em Cascata

A tabela principal STOREENT define uma chave principal STOREENT\_ID. STORE, uma tabela externa que se refere a STOREENT, define uma chave principal STORE\_ID que é uma chave externa para a tabela principal STOREENT. Isto significa que o valor de STORE\_ID deve ser um dos valores de STOREENT\_ID. STORE\_ID, a chave principal da tabela externa STORE, tem portanto dupla função—principal e externa.

Vamos supor que outra tabela, CONTRACT, seja uma tabela externa em STORE e que a chave externa de CONTRACT, STORE\_ID, faça referência à chave principal STORE\_ID em STORE. A tabela STORE é, portanto, uma tabela principal para a tabela CONTRACT.

Como STORE\_ID da tabela STORE é referido em STOREENT\_ID em vez de criado, o ID Resolver não cria uma associação entre alias interno e valor de ID para a tabela STORE. Quando a tabela CONTRACT tenta resolver o STORE\_ID a partir da tabela STORE, obtém um valor vazio.

Devido a essa condição especial, é necessário especificar explicitamente a criação do alias interno, criando uma entrada no arquivo de propriedades. Em `IdResolveKeys.properties`, especifique o seguinte:

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

Isso força o ID Resolver a fazer o seguinte:

- Crie uma associação de alias interno e valor de ID enquanto resolve o STORE\_ID como um referência externa
- Utilize a associação enquanto resolve STORE\_ID para a tabela CONTRACT

Utilizando a entrada `STORE=@STORE_ID:STORE_ID` no arquivo de propriedades e o seguinte fragmento de arquivo de entrada 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"
/>
```

resultaria no seguinte após a resolução:

```
<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"
/>
```

## Utilizando Resolução de Alias Interno

Para utilizar a resolução de alias interno com o ID Resolver, um alias é colocado no atributo de chave principal (identificador) no arquivo XML. O alias pode ser

utilizado em todo o arquivo XML para referir-se a esse elemento. Este processo elimina a necessidade de um programa para determinar os índices exclusivos necessários para criar o arquivo XML.

Os aliases internos devem ser utilizados de forma consistente em todo o arquivo. Se um ID de catálogo de endereços ID ADDRBOOK\_ID for colocado como o alias @addrbook\_1, todas as referências de chaves externas para o ID no arquivo precisam utilizar @addrbook\_1. Observe que os aliases são temporários. Eles não são salvos e não podem ser utilizados em um arquivo XML separado sem uma nova introdução dos aliases.

## Exemplo Parcial de Utilização da Resolução de ID de Alias Interno

### Resolução Anterior:

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

Alias para ADDRESS  
Refere-se ao alias para ADDRBOOK

### Resolução Posterior:

```
<MEMBER
  MEMBER_ID="100"
  TYPE="U"
  STATE="1"
/>
<ADDRBOOK
  ADDRBOOK_ID="11801"
  MEMBER_ID="100"
  DISPLAYNAME="Friends"
  DESCRIPTION="All my friends"
  TYPE="P"
/>
<ADDRESS
  ADDRESS_ID="11901"
  ADDRBOOK_ID="11801"
  MEMBER_ID="100"
  NICKNAME="Bob"
  ADDRESS1="1 Brave Developer St."
  CITY="Toronto"
```

Chave principal gerada

Chave principal gerada  
Refere-se à entrada ADDRBOOK

```
    ZIPCODE="A3B0F4"
    COUNTRY="Canada"
    STATUS="P"
  />
```

## Utilizando Resolução de Índice Exclusivo

Resolução de índice exclusivo, o comportamento padrão do ID Resolver, é utilizado quando não há entrada no arquivo de propriedades para a tabela que está sendo analisada ou quando não há arquivo de propriedades. A resolução de índice exclusivo utiliza qualquer um dos índices exclusivos em uma tabela como meio de localizar o identificador. Por exemplo, MEMBER\_ID mais IDENTIFIER é um índice exclusivo na tabela CATALOG e, portanto, pode ser utilizado como um ponto de resolução para a chave principal CATALOG\_ID da tabela CATALOGDSC.

Para atualizar o conteúdo do banco de dados, é necessário saber a chave exclusiva da tabela principal no banco de dados. Você pode consultar seu banco de dados para descobrir isso. Por exemplo, um comando do DB2 para recuperar uma chave exclusiva pode parecer o seguinte:

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

### Exemplo Parcial de Resolução de Índice Exclusivo

#### Resolução Anterior:

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

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

<CATALOGDSC
  CATALOG_ID="@WC2001@100" Refere-se novamente ao catálogo "WC2001" do membro
  "100" (Nota: A ordem é importante).
  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"
/>
```

#### Resolução Posterior:

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

<CATALOG
  CATALOG_ID="10351" Chave principal gerada automaticamente
  DESCRIPTION="Winter Catalog"
  IDENTIFIER="WC2001"
  MEMBER_ID="100"
  TPCLEVEL="2"
```



```
/>  
<CATALOGDSC  
  CATALOG_ID="10351"           Refere-se ao catálogo correto  
  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"  
>
```

---

## Carregando Dados na Tabela MEMBER

O ID Resolver trata a resolução para tabelas que têm identificadores gerados para elas pelo sistema. Isto inclui qualquer tabela e coluna registradas na tabela KEYS ou SUBKEYS. Essa resolução tem dois componentes:

1. Determinar se uma tabela principal (ou seja, uma tabela listada em KEYS ou SUBKEYS) existe no banco de dados.

Essa resolução é baseada no conteúdo dos dados XML desse elemento utilizando resolução de índice exclusivo ou especificação de arquivo de propriedades.

2. Determinar se há uma chave externa para uma tabela principal.

Isso é feito com uma especificação de resolução no atributo de chave externa da tabela relacionada.

A tabela MEMBER é utilizada como "superclasse" para as tabelas ORGENTITY, MBRGRP e USER. Isto cria um padrão "is-a" que é útil para manter a integridade referencial quando as tabelas têm limitações de chave externa para os subtipos da tabela MEMBER. Porém, como todos os subtipos de MEMBER compartilham um tipo base comum, o identificador deve ser exclusivo entre os subtipos. Isso significa que um ORGENTITY\_ID deve ser único no conjunto MBRGRP\_ID e USER\_ID. Para realizar isso, a tabela KEYS refere-se somente às tabelas ORGENTITY, MBRGRP e USER e especifica intervalos mutuamente exclusivos para seus identificadores. Cada um dos subtipos tem uma chave principal; cada uma dessas chaves principais também é uma chave externa para a chave principal da tabela MEMBER.

As restrições entre MEMBER e seus subtipos cria uma situação em que um MEMBER e subtipo não podem ter um ID sincronizado. Para carregar as tabelas ORGENTITY, MBRGRP e USER no sistema, o ID Resolver reconhece o padrão "is-a" e trata-o adequadamente. Essa é a sintaxe XML para o ID Resolver:

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

gera o seguinte:

```
<MEMBER
  MEMBER_ID="12345"
  TYPE="0"
  STATE="1"
/>
<ORGENTITY
  ORGENTITY_ID="12345"           Sincronizado com o elemento de membro
  ORGENTITYNAME="Test Org"
  ORGENTITYTYPE="0"
/>
```

Dessa forma, o ID Resolver trata o sub-elemento <isa> e cria um identificador sincronizado.

---

## Criando um Relacionamento Externo Utilizando a Tabela REFKEYS

A tabela REFKEYS é criada para representar um relacionamento externo entre tabelas que ainda não existem no banco de dados. Geralmente o esquema do banco de dados descreve o relacionamento externo criando uma declaração de chave externa que liga uma coluna de uma tabela a outra tabela. Se o esquema do banco de dados não tem um relacionamento definido e os identificadores precisam ser resolvidos como chave externa, então faça o seguinte:

1. Crie uma tabela REFKEYS, conforme mostrado no seguinte DDL de exemplo:

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

em que:

FKTABLE\_NAME é o nome da tabela externa (ou "filho")

FKCOLUMN\_NAME é o nome da coluna externa

TABLENAME é o nome da tabela principal (ou "pai")

2. Crie uma entrada na tabela REFKEYS que descreva o relacionamento externo necessário.

## Erros de Resolução de Problemas

Se ocorrerem erros ao resolver identificadores, consulte a seguinte tabela:

Erro	Método Utilizado	Causa Possível	Solução Possível	
Chave principal não resolvida	All	O ID Resolver <b>não</b> resolve a chave principal (identificador) em uma tabela que <b>não</b> esteja especificada na tabela KEYS ou SUBKEYS.	Inclua o nome da tabela para a qual a chave principal deve ser resolvida na tabela SUBKEYS antes de executar o ID Resolver.  Além disso, certifique-se de que o relacionamento de chave externa exista no esquema de banco de dados.	
	Atualizar	A chave principal é resolvida consultando-se o banco de dados. A consulta do banco de dados é gerada utilizando-se as informações de entrada do arquivo de propriedades ou o índice exclusivo da tabela fornecida. A entrada do arquivo de propriedades tem prioridade.	Certifique-se de que as informações do índice exclusivo no arquivo de entrada estejam corretas.  Você também pode criar ou modificar a entrada apropriada no arquivo de propriedades para gerar a consulta de banco de dados apropriada para resolver a chave principal.	
Chave externa não resolvida	All	O ID Resolver <b>não</b> resolve a chave externa em uma tabela em que as referências externas são para tabelas que <b>não</b> estejam especificadas na tabela KEYS ou SUBKEYS.	Inclua o nome da tabela referida na tabela SUBKEYS antes de executar o ID Resolver.  Além disso, certifique-se de que o relacionamento de chave externa exista no esquema de banco de dados.	
		Uma chave externa é resolvida utilizando-se o alias interno ou consultando-se o banco de dados. A consulta ao banco de dados será feita apenas se o alias interno falhar ao resolver a chave externa.	O alias interno é gerado utilizando-se a chave principal e a entrada do arquivo de propriedades.	Certifique-se de que os valores de alias interno utilizados como origem e destino no arquivo de entrada estejam corretos.
			A consulta do banco de dados é gerada utilizando-se as informações de entrada do arquivo de propriedades ou o índice exclusivo da tabela fornecida. A entrada do arquivo de propriedades tem prioridade.	Certifique-se de que as informações do índice exclusivo no arquivo de entrada estejam corretas.  Você também pode criar ou modificar a entrada apropriada no arquivo de propriedades para gerar a consulta de banco de dados apropriada para resolver a chave principal.

---

## Capítulo 6. Carregando Dados

Antes de carregar dados, você deve fazer o seguinte:

1. Gere um DTD e um esquema para utilização com o Loader (os primeiros dados de hora estão sendo carregados)

**Nota:** Se estiver carregando dados para um archive de loja e se criou o arquivo XML utilizando os DTDs fornecidos com o archive de loja, esta etapa não é necessária.

2. Resolve identificadores (se necessário)

Os dados precisam estar no formato XML com uma DTD associada. Para carregar dados, chame o comando Load.

---

### Configurando o Loader

O Loader package permite configurar como o Loader funciona, fazendo o seguinte:

- Ignorando elementos no arquivo de entrada
- Inserindo NULL em uma coluna
- Carregando dados de data e hora e data
- Carregando data e horas atuais
- Gerenciando filas de eventos
- Executando com software de banco de dados e sistemas operacionais diferentes
- Substituindo um componente
- Utilizando a sincronização de espaço de pesquisa do Consultor de Produto

Você pode personalizar estes recursos do Loader fazendo o seguinte:

1. Crie um novo arquivo de propriedades do personalizador do Loader.



MassLoadCustomizer.properties está localizado no archive MassLoader.zip. Extraia esse arquivo, renomeie-o mas mantenha a extensão .properties e coloque-o em um diretório que esteja no classpath. **Importante:** Não remova ou modifique o arquivo MassLoadCustomizer.properties existente.



ISeries\_LODWCSDTA\_Customizer.properties está localizado no diretório /QIBM/ProdData/WebCommerce/properties. Copie este arquivo para o diretório /instroot/xml, renomeie o novo arquivo, mas mantenha a extensão .properties; em seguida, faça as alterações necessárias no novo arquivo.

**Importante:** Não remova ou modifique o arquivo ISeries\_LODWCSDTA\_Customizer.properties original.

2. Modifique os valores das propriedades especificadas no novo arquivo de propriedades do personalizador do Loader.
3. Especifique o novo nome do arquivo como o valor do parâmetro customizer do comando Load.

## Ignorando Elementos no Arquivo de Entrada

Se seu arquivo de entrada contiver elementos que não são mapeados para o banco de dados de destino, você poderá definir o Loader para ignorar estes elementos no arquivo de propriedades do personalizador do Loader. Utilize IgnoreElements para especificar os elementos a serem ignorados e separe esses elementos por ponto e vírgula (;). Para ignorar os elementos import, literals e ProductRepository, por exemplo, especifique o seguinte no arquivo de propriedades do personalizador do Loader:

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

## Inserindo NULL em uma Coluna

Você pode ativar o Loader para inserir NULL em uma coluna, definindo a propriedade EnableNULLCheck como "true" no arquivo de propriedades do personalizador do Loader. Por exemplo:

```
EnableNULLCheck = true
```

Por motivos de desempenho, este recurso é desativado por padrão.

Utilize a propriedade NULLStringLiteral para determinar a representação de cadeia de um valor nulo em seus dados. Para definir o Loader para que a cadeia "-" seja utilizada para representar um valor nulo, por exemplo, especifique a propriedade e valor a seguir no arquivo de propriedades do personalizador do Loader:

```
NULLStringLiteral = -
```

Por padrão, o valor desta propriedade é "NULL" (sem aspas).

## Carregando Data e Hora e Dados de Data

O Loader pode carregar dados nas colunas com tipos de dados de data e hora e de data. Os formatos de dados para dados de data e hora no documento são determinados por padrões que podem ser personalizados. O usuário pode editar um padrão existente ou incluir mais padrões na lista de padrões existentes.

Os dados da data e hora ou data são verificados contra os padrões disponíveis (máscaras). O primeiro padrão que corresponder aos dados será utilizado para converter os dados no formato de data e hora de destino antes de carregá-los no banco de dados.

Há dois padrões de data e hora de saída personalizáveis, TimestampFormat.JDBC e TimestampFormat.Load.

1. TimestampFormat.JDBC é utilizado quando o Loader utiliza conexões JDBC para executar uma operação.  
Os métodos importar e excluir de SQL do Loader utilizam conexões JDBC para atualizar o banco de dados.
2. TimestampFormat.Load é utilizado quando o Loader usa utilitários nativos.  
Os métodos importar e carregar do Loader empregam utilitários nativos.

Você pode personalizar os formatos de data e hora modificando ou incluindo máscaras no arquivo de propriedades do personalizador do Loader.

São fornecidas as seguintes máscaras de data e hora de entrada:

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

Os padrões para os formatos de data de entrada são os seguintes:

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

Você pode modificar essas máscaras de data e hora e de data ou incluir quantas máscaras quiser. Especifique estas máscaras no arquivo de propriedades do personalizador do Loader na seqüência numérica na qual deseja que elas sejam comparadas com a data e hora de entrada. Se você incluir uma data e hora de entrada, será necessário utilizar o número seguinte na seqüência atual. (Por exemplo, a máscara de data e hora seguinte seria `InputTimeStampFormat.8` se você estiver incluindo na lista acima).

Os padrões para formatação de dados de entrada para a saída de data e hora e data são os seguintes:

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

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

Em geral, os formatos de data e de data e hora de saída *não* são personalizados.

## Carregando Data e Hora Atuais

O Loader pode inserir valores nas colunas com um tipo de dados de data e hora baseado em uma leitura do relógio da hora do dia. Por exemplo, `STARTDATE` e `ENDDATE` de uma oferta no WebSphere Commerce podem ter valores baseados na hora na qual a oferta foi inserida na tabela. Para suportar essa funcionalidade, o Loader package utiliza a tabela `MLTIME` para manter as instâncias de data e hora. O esquema para esta tabela é o seguinte:

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

Você pode personalizar o nome da tabela e suas colunas alterando as seguintes propriedades no arquivo de propriedades do personalizador do Loader:

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

Os dados de entrada para especificar valores de data e hora atuais são baseados nos padrões de cadeia de data e hora. As máscaras a seguir são utilizadas para especificar as durações da data e hora:

%D para dias  
%M para meses  
%Y para anos  
%H para horas  
%m para minutos  
%s para segundos

Você pode personalizar os formatos de data e hora atuais modificando ou incluindo máscaras no arquivo de propriedades do personalizador do Loader. As seguintes máscaras de entrada são fornecidas:

```
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 = ADDDDAYS(SYSDATE,%D)
InputCurrentTimestampFormat.8 = ADDDDAYS(ADDMONTHS(SYSDATE,%M),%D)
InputCurrentTimestampFormat.9 = ADDDDAYS(ADDMONTHS(ADDEYEARS(SYSDATE,%Y),%M),%D)
```

Os dados de entrada para a data e hora atuais correspondem aos padrões especificados. Se os dados corresponderem a um padrão de entrada especificado, esse padrão será utilizado para analisar os dados de entrada e o Loader converterá os dados para o formato de saída apropriado antes de inseri-los no banco de dados. Novos padrões podem ser incluídos na lista acima, desde que os números dos subscripts estejam ordenados seqüencialmente.

Há dois formatos de saída de destino para especificar data e hora atuais:

1. CurrentTimestampFormat.Load é utilizado quando o Loader está operando em modo carregar ou importar.
2. CurrentTimestampFormat.JDBC é utilizado quando o Loader utiliza JDBC para inserir, atualizar ou excluir valores no banco de dados.

Os padrões de destino padrão do Loader são os seguintes:

```
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
```

As propriedades para estes padrões também podem ser personalizadas no arquivo de propriedades do personalizador do Loader. Quando você personalizar as propriedades CurrentTimestampFormat.Load e CurrentTimestampFormat.JDBC, deverá assegurar que a sintaxe da instrução resultante seja válida para o sistema de gerenciamento de banco de dados fornecido.

A propriedade CurrentTimestampLiteral é utilizada pelo Loader para fazer uma determinação prévia de se o valor da coluna de data e hora está no formato de data e hora atual, evitando, assim, vários cálculos para determinar que o valor não é uma representação de cadeia de data e hora.

```
CurrentTimestampLiteral = CURRENT TIMESTAMP
```





O valor padrão desta propriedade para o DB2 é CURRENT\_TIMESTAMP.



O valor padrão para o banco de dados Oracle é SYSDATE.

### Exemplo de Carregamento de Data e Hora Atuais

O Loader recebe as informações abaixo para atualizar a oferta com um OFFER\_ID de 10123. A data de início tem o valor de "CURRENT\_TIMESTAMP" e a data de encerramento tem um valor de "CURRENT\_TIMESTAMP + 14 DAYS."

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

O Loader reconhece que as colunas STARTDATE e ENDDATE são de um tipo de dados de data e hora no banco de dados. Com base na propriedade CurrentTimeStampLiteral, os valores são determinados a ter valores especificados no formato de data e hora atual. O valor para STARTDATE corresponde ao padrão InputCurrentTimeStampFormat.1 e é convertido para o padrão especificado pela propriedade CurrentTimeStampFormat.JDBC. O valor de ENDDATE corresponde ao formato da propriedade InputCurrentTimeStampFormat.2 e também é convertido para o padrão especificado pela propriedade CurrentTimeStampFormat.JDBC.

### Exemplos de Inclusão de Durações em Data e Hora Atuais

O Loader fornece a capacidade de incluir durações na data e hora atuais. Por exemplo, você pode querer carregar uma oferta sem inserir uma data específica. Para fazer isso, é necessário criar uma data de encerramento que tenha uma certa duração após a data de início. O exemplo a seguir funciona bem com o DB2:

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

No entanto, para tratar as durações de data e hora atuais de forma independente da plataforma, você deve personalizar os formatos de data e hora atuais modificando as máscaras no arquivo de propriedades do personalizador do Loader. Aqui está um exemplo de especificações da propriedade de data e hora atual personalizadas:

```
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
```

Utilizando o exemplo da oferta e essas especificações da propriedade, a data de encerramento da oferta corresponde ao padrão InputCurrentTimeStampFormat.5. Isso libera as seguintes informações da oferta, utilizando CurrentTimeStampFormat.JDBC.

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

O exemplo acima mostra como o Loader pode inserir vários formatos de data de hora atuais e formatá-los corretamente para o formato de saída desejado. O exemplo a seguir mostra como você pode lidar com formatos independentes de plataforma e mapeá-los para formatos de saída específicos de plataforma.

```
<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:** A instrução acima é apenas um exemplo. É utilizada simplesmente para ilustrar o recurso de personalização para um sistema de gerenciamento de banco de dados hipotético. Não é válida para um banco de dados DB2 ou Oracle.

Utilizando o exemplo da oferta e essas especificações da propriedade, a data de encerramento da oferta corresponderá ao padrão `InputCurrentTimestampFormat.5`. Isso libera as seguintes informações da oferta, utilizando `CurrentTimestampFormat.JDBC`.

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

## Gerenciando Filas de Eventos

Você pode personalizar as filas de eventos do Loader modificando as definições no arquivo de propriedades do personalizador do Loader. Por exemplo:

```
QueueLowCount = 35
QueueHighCount = 90
```

A origem dos eventos que preenchem a fila é bloqueada quando o número de elementos na fila atinge o limite mais alto, impedindo que mais eventos sejam colocados na fila. A fila começa a aceitar eventos novamente quando o número de elementos nela cai abaixo do limite inferior.

## Executando com Software do Banco de Dados e Sistemas Operacionais Diferentes

Você pode personalizar o Loader para execução com software do banco de dados e sistemas operacionais diferentes modificando os parâmetros para os seguintes

elementos no arquivo de propriedades do personalizador do Loader para especificar software do banco de dados e sistemas operacionais diferentes:

- Comando de conexão de bancos de dados
- Comando de tabela de carregamento de banco de dados
- Comando de importação de bancos de dados
- Comando do sistema que chama o carregamento

Para personalizar um destes itens, remova os caracteres de comentários de barra dupla (//) que precedem o comando no arquivo de propriedades do personalizador do Loader e modifique os padrões.

**Comando de conexão de banco de dados:** Você pode alterar os parâmetros do comando de conexão de banco de dados se desejar modificar os padrões (que assumem que você esteja utilizando o DB2).

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

em que:

- 0 = nome do banco de dados
- 1 = usuário do banco de dados
- 2 = senha do usuário

**Comando de tabela de carregamento de banco de dados:** Você pode alterar os parâmetros do comando de tabela de carregamento de banco de dados se desejar modificar os padrões.

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

em que:

- 0 = nome do arquivo
- 1 = delimitador de coluna
- 2 = delimitador de caractere
- 3 = nome da tabela
- 4 = nomes de colunas, separados por vírgulas (,)

**Comando de importação de banco de dados:** Você pode alterar os parâmetros do comando de importação de banco de dados se desejar modificar os padrões.

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

em que:

- 0 = nome do arquivo
- 1 = delimitador de coluna
- 2 = delimitador de caractere
- 3 = nome da tabela
- 4 = nomes de colunas, separados por vírgulas (,)

**Comando do sistema que chama o carregamento:** Você pode alterar os parâmetros do comando do sistema se desejar modificar os padrões. Esse comando executa os scripts load e import nativos gerados pelo Loader.

- ▶ **DB2**

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

em que:

0 = nome do arquivo de log  
1 = nome do arquivo de comandos

Para DB2 em execução no AIX, por exemplo, o valor para a propriedade DBLoadCommand é o seguinte:

db2 -tvf {1} -z {0}

- ▶ **AIX** ▶ **Solaris** ▶ **Linux** ▶ **Oracle**

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

em que:

0 = nome do arquivo de log  
1 = nome do arquivo de comandos  
2 = nome do usuário do banco de dados

Utilize as seguintes definições para as diversas combinações de bancos de dados e sistemas operacionais.

- ▶ **NT** ▶ **2000** ▶ **DB2**

Defina a variável de ambiente classpath do sistema para incluir db2/dbconnect.zip para DB2 em execução no Windows NT ou Windows 2000 com o método sqlimport, load, import ou delete.

- ▶ **AIX** ▶ **Solaris** ▶ **Linux** ▶ **DB2**

Para DB2 em execução em ambientes AIX, Solaris ou Linux, faça o seguinte:

- Com o método sqlimport, load, import ou delete, defina a variável de ambiente classpath do sistema para incluir db2/dbconnect.zip.
- Com o método load ou import, modifique as seguintes propriedades no arquivo de propriedades do personalizador do Loader.

```
/**
 * Comando de conexão. (0 padrão é para DB2)
 * parâmetro 0 = dbName
 * parâmetro 1 = dbUser
 * parâmetro 2 = userPasswd
 */
DBConnectCommand = connect to {0} user {1} using {2};

/**
 * Dados de Carga para comando de Tabela. (0 padrão é para DB2)
 * parâmetro 0 = nome do arquivo
 * parâmetro 1 = delimitador de coluna
 * parâmetro 2 = delimitador de caractere
 * parâmetro 3 = nome da tabela
 * parâmetro 4 = nome das colunas, separado por vírgulas (,)
 */
DBLoadTableCommand = load from {0} of del modified by coldel{1}
```

```



insert into {3} ({4});

/**
 * Dados de Inserção para comando de Tabela. (0 padrão é para DB2)
 * parâmetro 0 = nome do arquivo
 * parâmetro 1 = delimitador de coluna
 * parâmetro 2 = delimitador de caractere
 * parâmetro 3 = nome da tabela
 * parâmetro 4 = nome das colunas, separado por vírgulas (,)
 */

DBUpdateTableCommand = import from {0} of del modified by coldel{1}
insert_update into {3} ({4});
/**
 * Comando do sistema para chamar a carga. (0 padrão é para DB2)
 * parâmetro 0 = logFileName
 * parâmetro 1 = commandFileName
 */

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

```

- 

 Modifique as seguintes propriedades no arquivo de propriedades do personalizador do Loader para o DB2 em execução no iSeries com o método sqlimport, load, import ou delete:

```

/**
 * A cadeia de conexão.
 */

ConnectionStringID = jdbc:db2://

/**
 * As informações do driver JDBC.
 */

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

/**
 * Gravador personalizado para métodos load/import.
 */

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

```

- 






Defina a variável de ambiente classpath do sistema para incluir oracle/dbconnect.zip no banco de dados Oracle em execução em ambientes Windows NT, Windows 2000, AIX, Solaris ou Linux, com o método sqlimport, load, import ou delete.

## Substituindo um Componente

Você pode substituir um componente do Loader fornecendo aos seguintes elementos no arquivo de propriedades do personalizador do Loader os valores de classes que deseja substituir pelas implementações padrão:

### ParserName

Nome do analisador a ser utilizado

### ValidatorName

Nome do validador a ser utilizado

### FormatterName

Nome do formatador a ser utilizado

**JDBCFormatterName**

Nome do formatador quando o método de importação SQL é utilizado

**WriterName**

Nome do escritor a ser utilizado

**JDBCWriterName**

Nome do gravador quando o método de importação SQL é utilizado

Para substituir o escritor padrão do Loader (DefaultWriter) pelo escritor com.abc.writer.SpecialWriter, por exemplo, especifique o seguinte no arquivo de propriedades do personalizador do Loader:

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

O Loader utilizará "com.abc.writer.SpecialWriter" para executar a função de escritor.

## Utilizando a Sincronização do Espaço de Procura do Consultor de Procura

Para utilizar a sincronização de espaço de pesquisa do Consultor de Produto, faça o seguinte.

1. Crie um arquivo de informações de configuração XML para a sincronização chamada "PASyncInfo.xml."
2. Em PASyncInfo.xml, especifique PASync.xsd como o esquema XML a ser utilizado. Por exemplo:

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

O arquivo PASync.xsd é fornecido. O texto a seguir mostra o conteúdo de 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. Em PASyncInfo.xml, especifique o ID do membro e o ID de entidade de loja para os quais a sincronização precisa ser feita. Por exemplo:

```

member = "-2000"
store = "10351"

```

4. Sob o elemento esquema de procura em PASyncInfo.xml, especifique os identificadores do CATGROUP que formam o espaço de procura. Por exemplo:

```

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

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

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

</SearchScheme>

```

"Pants" e "Shirts" são especificados no exemplo. Você pode especificar quantos identificadores de CATGROUP quiser.

5. Em PASyncInfo.xml, especifique atributos para determinar quais comandos programar. Por exemplo:

```

<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>

```

- No novo arquivo de propriedades do personalizador do Loader, especifique o arquivo de informações de configuração XML. Por exemplo:

```
PASyncDocumentURL = PASyncInfo.xml
```

- No novo arquivo de propriedades do personalizador do Loader, ative a sincronização. Por exemplo:

```
PASyncEnabled = true
```

- Utilize o método importar SQL ou excluir com o comando Load. Aqui está um exemplo de entrada XML apropriada ao 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:** A desativação da sincronização do espaço de pesquisa do Consultor de Produto fornece melhor desempenho do Loader; portanto, utilize este recurso somente quando ele for necessário.

## Personalizando Sincronização do Espaço de Procura do Consultor de Produto

O Loader package permite personalizar a sincronização do espaço de pesquisa do Consultor de Produto modificando o arquivo de propriedades do personalizador do Loader para fazer o seguinte:

- **Ativar ou desativar a sincronização**

Você pode ativar ou desativar a sincronização especificando true ou false como o valor para a seguinte propriedade no arquivo de propriedades do personalizador do Loader:

```
PASyncEnabled = true
```

- **Especificar o arquivo de informações da configuração para a sincronização**

Você pode especificar qual arquivo de informações de configuração XML a sincronização utiliza definindo o valor para a seguinte propriedade no arquivo de propriedades do personalizador do Loader:

```
PASyncDocumentURL = PASyncInfo.xml
```

- **Especificar o comprimento da consulta da programação**

Você pode especificar o comprimento da consulta de programação definindo o valor para a seguinte propriedade no arquivo de propriedades do personalizador do Loader:

```
PAScheduleQueryLength = 30
```

O valor para esta propriedade deve estar no intervalo de 20 a 900.

- **Especificar o horário de abertura programado**

Você pode especificar o horário de abertura programado especificando uma data e hora absolutas, uma data e hora atuais ou uma data e hora atuais com duração como o valor para a propriedade PAScheduledStartTime no arquivo de propriedades do personalizador do Loader.

**Nota:** O formato de data e hora deve ser apropriado para seu banco de dados.

Segue um exemplo para o DB2 que executará o job programado 5 minutos após o carregamento:

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



A seguir está um exemplo para o banco de dados Oracle que executará o job imediatamente:

```
PAScheduledStartTime = SYSDATE
```

---

## Determinando como Processar Dados ao Utilizar o Loader

O Loader oferece as seguintes opções para processar dados utilizando o comando Load:

- Carregando
- Importando
- Utilizando o recursos de importação SQL

Antes de carregar dados, você deve determinar qual método de processamento produziria os melhores resultados.

### Escolhendo o Método Load

Considere o método load nas seguintes situações:

- Se souber que os dados estão limpos e se o banco de dados não contém dados
- Se souber que os dados estão limpos e se souber que o banco de dados não contém os dados que estão sendo carregados
- Se souber que os dados estão limpos, se as tabelas de destino não contêm chaves principais e se souber que o banco de dados não contém os dados que estão sendo carregados

- Se o tempo de carregamento for sua maior preocupação
- Se o banco de dados for um banco de dados DB2 local

**400** Com o método load, os dados são carregados no banco de dados. Se os dados já existirem, o comando falha com um resultado de erro de chave duplicada e é exibida uma mensagem de erro de duplicata.

## Escolhendo o Método Importar

**NT** **2000** **AIX** **Solaris** **Linux** Com o método import para DB2, os dados também são carregados no banco de dados. Se os dados já existirem, eles não são excluídos, mas atualizados com novos valores. Considere esse método em qualquer uma das seguintes situações:

- Se o sistema de gerenciamento de banco de dados for DB2
- Se você sabe que os dados estão limpos
- Se tiver que atualizar grandes conjuntos de dados homogêneos em um nível de coluna
- Se o tempo de carregamento não for sua principal preocupação
- Se a tabela para a qual dados são importados tiver chaves principais

**400** Com o método import, os dados também são carregados no banco de dados. Se os dados já existirem, eles não são excluídos, mas atualizados com novos valores. Considere esse método em qualquer uma das seguintes situações:

- Se você sabe que os dados estão limpos
- Se os dados já existem no banco de dados
- Se o tempo de carregamento não for sua principal preocupação
- Se a tabela para a qual dados são importados tiver chaves principais

## Escolhendo o Método SQL Import

Com o método import de SQL, instruções JDBC ou SQL são utilizadas para atualizar ou inserir dados no banco de dados. Os dados são inseridos se já não existirem e os dados existentes serão atualizados. Considere esse método em qualquer uma das seguintes situações:

- Se você estiver atualizando dados existentes e precisar de atualizações no nível de coluna
  - O relato de erros de violações de limitação e de erros de tipo de dados é melhor com esse método.
- Se você souber que alguns dados não estão limpos
- Se a integridade do banco de dados for sua principal preocupação
- Se o banco de dados não for local
- Se você estiver utilizando sincronização de espaço de procura do Consultor de Produto

## Outras Considerações

- **Restrições para a utilização do método load**
  - O método load não pode inserir ou atualizar dados em campos de dados de bits.

**DB2** Com o método load, somente os registros novos são inseridos no banco de dados; os registros existentes não são atualizados.

**DB2** O método load pode ser utilizado apenas para bancos de dados DB2 locais, e não remotos.

- **Restrições para a utilização do método importar**

O sistema de gerenciamento do banco de dados deve ser o DB2 para a utilização do método import.

O método importar não pode inserir ou atualizar dados em campos de dados de bits.

Com o método import, o Loader somente insere ou atualiza tabelas que tenham chaves principais definidas nelas; o método import não pode inserir ou atualizar dados em tabelas que não tenham uma chave principal. Se o registro de entrada tem valores somente para as colunas principais, o registro é rejeitado.

- **Comparação dos métodos import de SQL e load**

O método import de SQL verifica a consistência dos dados, incluindo referências externas, e permite atualizar dados existentes. O método load não.

- **Comparação dos métodos import e import de SQL**

Os métodos import e import de SQL executam funções similares. O método import é geralmente mais rápido, mas requer espaço em disco para arquivos temporários.

O método import somente pode inserir ou atualizar tabelas que tenham chaves principais definidas nelas, enquanto que o método SQL import não requer que tabelas tenham chaves principais.

- **Comparação de métodos com base no produto do banco de dados utilizado**

Os métodos import e load utilizam utilitários nativos que são otimizados para o DB2, enquanto que o método import de SQL utiliza chamadas JDBC (que são genéricas para muitos produtos de banco de dados).

- **Considerações adicionais**

O método delete é utilizado para excluir dados que se encontram no documento XML de entrada do banco de dados. O elemento precisa conter os valores para a chave principal ou o índice exclusivo para a tabela. Se os dados que estão sendo excluídos tiverem dependências de dados em outra tabela com "cascade on delete" ativado, os dados dependentes também serão excluídos.

Se você está utilizando a sincronização de espaço de pesquisa do Consultor de Produto, deve executar o método import de SQL para carregar dados.

---

## Carregando Documentos Grandes

Ao utilizar os utilitários do Loader package para carregar grandes documentos em um banco de dados, considere os seguintes itens:

- **Tamanho de heap do JVM (Java Virtual Machine)**

Por padrão, o valor máximo de memória alocada para o heap de JVM é 64 MB. Se não for aumentado, a JVM pode, eventualmente, ficar sem memória durante o processo de carregamento. A quantidade máxima de memória alocada para o heap de Java pode ser variada com a utilização da opção -mx de JVM no comando Java.

- **Registro de rastreamento**

O logger de rastreamento pode exaurir o heap de JVM ao carregar um documento XML grande. As informações de rastreamento são utilizadas principalmente para depurar uma execução se a execução falhar. Se o

rastreamento do processo de carregamento não for necessário, o rastreamento deve ser desativado. Há um ganho de desempenho significativo quando o rastreamento é desativado. O rastreio é desativado modificando o documento XML de configuração de registro.

O arquivo de configuração de log padrão é `WCALoggerConfig.xml`. Para desativar o log de rastreamento, altere a configuração do logger de rastreamento do Loader de:

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

para:

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

Para obter mais informações sobre como modificar o arquivo `WCALoggerConfig.xml`, consulte “Personalizando o log para o Loader package” na página 68.

- **Contagem de consolidação**

A contagem de consolidação padrão do Loader quando está operando em modo de importação de SQL é 1. Portanto, por padrão, as transações são consolidadas para toda atualização ou inserção no banco de dados. Para melhorar o desempenho do Loader para documentos grandes, a contagem de consolidação deve ser aumentada. É sugerido o valor "100", mas pode ser maior, dependendo da quantidade de memória física no servidor, do tamanho do log de transação DBMS, etc.

A contagem de consolidações para o Loader é alterada utilizando-se a opção `-commitcount count` do comando Load (em que `count` é o número de instruções executadas antes da transação ser consolidada).

## Dica de Resolução de Problemas

Se o progresso for incomumente lento ao carregar dados, o log do Loader pode ter uma rotina de tratamento de arquivos configurada incorretamente. Isto pode resultar de uma das seguintes situações:

- O usuário que está chamando o Loader não tem permissão para gravar no diretório ou atualizar o arquivo especificado no documento de configuração do registro.
- O diretório especificado como a localização do arquivo no documento de configuração do registro não existe.
- A unidade especificada como a localização do arquivo no documento de configuração do registro não tem espaço suficiente.

Quando corrigir qualquer um destes problemas, talvez seja necessário alterar a localização especificada do arquivo, modificando o documento de configuração do registro (`WCALoggerConfig.xml` por padrão). Para obter mais informações sobre rotinas de tratamento de arquivos e o arquivo `WCALoggerConfig.xml`, consulte “Personalizando o log para o Loader package” na página 68.

---

## Capítulo 7. Extrair Dados

Para extrair dados de um banco de dados utilizando o Extractor, é necessário especificar os dados que deseja extrair do banco de dados utilizando um arquivo de filtro de extração. O filtro de extração utilizado depende do tipo de dados que deseja extrair.

O exemplo a seguir extrai dados do Subsistema de membros de um banco de dados, utilizando MemberSubsystemFilter.xml como um filtro de extração.

- ▶ NT ▶ 2000 ▶ AIX ▶ Solaris ▶ Linux
  - ▶ 400
- ```
java com.ibm.wca.MassExtract.Extract -filter MemberSubsystemFilter.xml
-outfile MemberSubsystemExtracted.xml -dbname mall -dbuser myname
-dbpwd mypassword -customizer MemberSubsystemCustomizer

QWEBCOMM/EXTWCSDTA FILTER(MemberSubsystemFilter.xml)
OUTFILE(MemberSubsystemExtracted.xml) DATABASE(database_name)
SCHEMA(mall) INSTRROOT(/QIBM/UserData/WebCommerce/instances/mser)
PASSWD(mypassword)
```

---

### Criando um Filtro de Extração

O exemplo a seguir de um filtro de extração extrai informações de categorias e de produtos das tabelas 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>

```

---

## Configurando o Extractor

Você pode alterar os drivers do banco de dados utilizados pelo Extractor fazendo o seguinte:

1. Crie um novo arquivo de propriedades do personalizador do Extractor.



`DB2ConnectionCustomizer.properties` está localizado no archive `MassExtract.zip`. Extraia esse arquivo, renomeie-o mas mantenha a extensão `.properties` e coloque-o em um diretório que esteja no classpath.

**Importante:** Não remova nem modifique o arquivo `DB2ConnectionCustomizer.properties` existente.



`ISeries_EXTWCSDTA_Customizer.properties` está localizado no diretório `/QIBM/ProdData/WebCommerce/properties`. Copie este arquivo para o diretório `/instroot/xml`, renomeie o novo arquivo, mas mantenha a extensão `.properties`; em seguida, faça as alterações necessárias no novo arquivo.

**Importante:** Não remova ou modifique o arquivo `ISeries_EXTWCSDTA_Customizer.properties` original.

2. Modifique os valores do driver do banco de dados no novo arquivo.
3. Especifique o nome do novo arquivo como o valor do parâmetro `customizer` do comando `Extract`.

A seguir está um resumo do arquivo de propriedades do personalizador de Extractor:

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

em que:

- `DBVendorName` é utilizado para selecionar o tipo de banco de dados.  
As opções são as seguintes:
  - DB2 Universal Database para iSeries (DB2/iSeries)
  - DB2 para outros sistemas operacionais (DB2)
  - Banco de dados Oracle (Oracle)
- `DBDriverName` é utilizado para selecionar o driver JDBC.  
As opções são as seguintes:
  - DB2 Universal Database para iSeries (`com.ibm.db2.jdbc.app.DB2Driver`)
  - DB2 para outros sistemas operacionais (`COM.ibm.db2.jdbc.app.DB2Driver`)
  - Banco de dados Oracle (`oracle.jdbc.driver.OracleDriver`)
- `DBURL` é utilizado para especificar a URL para acessar o banco de dados.  
As opções são as seguintes:
  - DB2 Universal Database para iSeries (`jdbc:db2://`)
  - DB2 para outros sistemas operacionais (`jdbc:db2:`)
  - Banco de dados Oracle (`jdbc:oracle:oci8:@`)

---

## Capítulo 8. Utilizando o Logger do Loader Package

Cada utilitário no Loader package cria mensagens para indicar êxito, falha e erros, bem como para fornecer informações de rastreamento do programa.

Os utilitários no Loader package fazem referência ao arquivo `WCALoggerConfig.xml`.

- 

Este arquivo existe em um diretório especificado na variável de ambiente `classpath` do sistema. Ele também pode ser especificado pela propriedade de sistemas Java `com.ibm.wca.logging.configFile`.

- 

Este arquivo está localizado no diretório `/instroot/xml`.

`WCALoggerConfig.xml` determina quais informações de registro cada utilitário fornece e onde as informações são exibidas ou armazenadas. Você pode personalizar esse arquivo e especificar os tipos de logs que são criados e os tipos de mensagens que são registrados em log.

---

### Configurando o Log em seu Ambiente para Sistemas Windows NT, Windows 2000, AIX, Linux e Solaris

Para configurar o log em seu ambiente, você deve definir a variável de ambiente do sistema `classpath` para incluir o arquivo `WCALoggerConfig.xml` ou especificar a propriedade do sistema `com.ibm.wca.logging.configFile`.

#### Exemplo de Definição da Variável `classpath`

Se o arquivo `WCALoggerConfig.xml` estiver no diretório `d:\WebSphere\CommerceServer\xml\loader` em uma máquina do Windows NT, por exemplo, você poderá utilizar a seguinte instrução para definir a variável `classpath`:

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

#### Exemplo de Especificação da Propriedade do Sistema `com.ibm.wca.logging.configFile`

Para especificar a propriedade do sistema `com.ibm.wca.logging.configFile`, utilize a opção `-D` ao chamar o interpretador Java. Segue-se um exemplo:

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

---

## Personalizando o log para o Loader package

Para personalizar o log para o Loader package, utilize o arquivo WCALoggerConfig.xml.

-     







Este arquivo existe em um diretório especificado na variável de ambiente classpath do sistema. Ele também pode ser especificado pela propriedade de sistemas Java com.ibm.wca.logging.configFile.

- 

Este arquivo está localizado no diretório */instroot/xml*.


WCALoggerConfig.xml contém uma ou mais tags de componentes, <component name="DTDGenerator"> por exemplo. Em cada uma dessas tags, você pode incluir loggers e rotinas de tratamento. Você *não* deve alterar as tags do utilitário e logger fornecidas com o sistema, mas você pode alterar as tags de rotinas de tratamento para os loggers. Para obter informações sobre o que você pode incluir neste arquivo, consulte o arquivo WCALogger.dtd.

Os logs do Loader package estão localizados no arquivo messages.txt nos seguintes diretórios:

-  unidade:\WebSphere\CommerceServer\instances\*instance\_name*\logs
-  unidade:\Arquivos de Programas\WebSphere\CommerceServer\instances\*instance\_name*\logs
-  /usr/WebSphere/CommerceServer/instances/*instance\_name*/logs
-   /opt/WebSphere/CommerceServer/instances/*instance\_name*/logs
-  /QIBM/UserData/WebCommerce/instances/*instance\_name*/logs

## Rotinas de Tratamento

Para incluir uma rotina de tratamento em um logger, especifique o tipo de rotina de tratamento no arquivo WCALoggerConfig.xml. Você pode incluir mais de uma rotina de tratamento em um logger. Observe que cada rotina de tratamento tem seus próprios atributos e tags subordinadas que não se aplicam necessariamente a outras rotinas de tratamento. Os tipos de rotinas de tratamento incluem o seguinte:

Tipo de rotina de tratamento	Descrição e atributos
console	Envia mensagens para saída padrão, geralmente a linha de comandos
file	<p>Armazena mensagens em um arquivo de texto</p> <p>Você deve incluir "&lt;filePath&gt;caminho do log&lt;/filePath&gt;" nesta rotina de tratamento como tag subordinada.</p>
multifile	<p>Cria um log de arquivos circular</p> <p>Você deve especificar "&lt;filePath&gt;caminho do log&lt;/filePath&gt;". São criados os arquivos de log 1 a <i>n</i>. Você pode incluir os seguintes atributos:</p> <p><b>MaxFiles</b> Um inteiro que indica quantos arquivos de log deve-se utilizar antes de se apagar o primeiro log</p> <p><b>MaxKBFileSize</b> Um inteiro que indica o número máximo de kbytes a armazenar em cada arquivo de log</p>
 <p>database</p>	<p>Armazena mensagens em uma tabela do DB2 em um log circular. Você pode incluir os seguintes atributos:</p> <p><b>brand</b> Nome da marca do banco de dados. DB2 é o único banco de dados suportado atualmente.</p> <p><b>maxRows</b> O número máximo de registros a serem armazenados em uma tabela antes de se apagar a entrada mais antiga</p> <p>Você pode incluir "&lt;jdbc/&gt;" como tag subordinada e incluir os seguintes atributos:</p> <p><b>url</b> URL utilizada em JDBC para acessar um banco de dados (por exemplo, "Jdbc:db2:"<i>wcm</i>"" , em que "<i>wcm</i>" é o nome do banco de dados). O banco de dados deve existir antes que você execute o utilitário.</p> <p><b>table</b> Nome da tabela do banco de dados onde as mensagens serão registradas. Ela deve ser criada com a seguinte instrução do DB2:</p> <pre>"CREATE TABLE"nome_tabela (KEY char(13) FOR BIT DATA NOT NULL, COMPONENTNAME VARCHAR(30), ENTRY VARCHAR(2000), PRIMARY KEY(key))"</pre> <p><b>userid</b> Nome do usuário do banco de dados. O usuário deve ter permissões para atualizar a tabela. A seguinte instrução do DB2 fará isto:</p> <pre>"GRANT SELECT, UPDATE, INSERT, DELETE ON TABLE"tablename TO USER userid"</pre> <p><b>password</b> Senha do banco de dados para o nome do usuário especificado</p>

O exemplo a seguir inclui uma rotina de tratamento do tipo "database" em um logger:

```

<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>

```

## Filtros

Filtros podem ser incluídos ou removidos de rotinas de tratamento para incluir e excluir tipos de mensagens. Se um logger não tiver filtros, nenhuma mensagem será registrada. Cada tag de filtro tem uma tag messageType subordinada que lista o tipo de mensagem, que geralmente é um dos seguintes:

- INFO
- ERROR
- FATAL
- WARNING

Outros tipos de mensagens estão listados no arquivo WCALogger.dtd, mas a maioria geralmente não é utilizada com o Loader package.

Os tipos de filtro incluem o seguinte:

Tipo de filtro	Descrição e atributos
Any	Inclui no arquivo de log qualquer mensagem sinalizada como um dos tipos messageType especificados  Por exemplo, se a lista messageType inclui ERROR e o aplicativo gera uma mensagem de tipo ERROR, a mensagem é registrada.
All	Requer que uma mensagem tenha todos os atributos de tipo messageType especificados antes de serem incluídos em um log
Exclui	Registra todas as mensagens não especificadas na lista de tags messageType

O exemplo a seguir de incluir um filtro em uma rotina de tratamento permite que os tipos de mensagens FATAL e ERROR sejam registradas e que outras mensagens sejam ignoradas:

```

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

```

## Formatos

Você pode especificar um entre dois tipos de formatador para a formatação da mensagem:

Tipo de formatador	Descrição e atributos
--------------------	-----------------------

safe (padrão)	Evita que seja definida uma exceção se não for possível encontrar uma mensagem em um arquivo de propriedades  Esse formatador cria uma mensagem indicando que o recurso está faltando.
xml	Formata a mensagem no formato XML  Se não for possível encontrar uma mensagem, este formatador também grava uma mensagem ao invés de definir uma exceção.

## Exemplo: 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

```



```
constants
-->
<!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)>
```



---

## Capítulo 9. Utilizando o Relator de Erros do Loader Package

O Loader e o ID Resolver incluem um relator de erros que gerará um documento de exceção se ocorrer um erro.

Por padrão, o documento de exceção é gravado no seguinte diretório:

-      o diretório em que o documento de entrada reside
-  `/instroot/logs`

Para especificar o diretório em que o documento de exceção foi gravado, utilize a propriedade Java `com.ibm.wcm.ErrorReporterDir`. Um exemplo do Loader em um ambiente Windows NT começaria da seguinte forma:

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

**Nota:** O usuário deve ter permissão para gravar no diretório especificado.

A seguir está um DTD de exemplo (`store-all-error.dtd`) para o relator de erros:

```
<!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  
  taxcgr_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  
  mfpartment     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
>

```

A seguir está um documento de relatório de erros de exemplo do Loader:

```







<?xml version="1.0"?>
<!DOCTYPE store-asset SYSTEM "store-all-error.dtd">
<store-asset>
  <error
    locus="Parser"
    id="SAXParseFatalError" >
    <message>
      Erro A cadeia "--" não é permitida em comentários. : 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>
        Foi recebida uma Exceção SQL [IBM][CLI Driver][DB2/NT] SQL0530N
        O valor de insert ou update da FOREIGN KEY
        "JANTONY.CALRULE.F_CALRULE4" não é igual a qualquer valor
        da chave pai da tabela pai. SQLSTATE=23503
      </message>
    </error>
  </calrule>
  <catentry
    catentry_id="10118"
    member_id="-2001"
    partnumber="1254"
    mfpartnumber="sku-163"
    mfname="InFashion"
    markfordelete="0"
    buyable="1"
    field1="abc" >
    <error
      locus="Formatter"
      id="FormattingError" >
      <message>
        Erro ao formatar o valor para CATENTRY.FIELD1 : abc com erro
        [class java.lang.NumberFormatException(abc)].
      </message>
    </error>
  </catentry>
</store-asset>

```

---

## Capítulo 10. Configurando Comandos e Scripts do Loader Package

Para lançar o Loader package e executar seus comandos, utilize os scripts ou comandos fornecidos no diretório do WebSphere Commerce:

-  *unidade:\WebSphere\CommerceServer\bin*
-  *unidade:\Arquivos de Programas\WebSphere\CommerceServer\bin*
-  */usr/WebSphere/CommerceServer/bin*
-   */opt/WebSphere/CommerceServer/bin*
-  Biblioteca nativa QWEBCOMM

O scripts e comandos são os seguintes:



### GENWCSDTD

Comando DTD Generate

### RESWCSID

Comando Resolução de IDs

### EXTWCSDTA

Comando Extract

### LODWCSDTA

Comando Load

### TRNWCSTXT

Comando Text Transform

### TRNWCXML

Comando XML Transform

### **dtdgen.cmd**

Comando DTD Generate

### **idresgen.cmd**

Comando Resolução de IDs

### **massextract.cmd**

Comando Extract

### **massload.cmd**

Comando Load

### **txttransform.cmd**

Comando Text Transform

### **xmltransform.cmd**

Comando XML Transform



**dtdgen.sh**

Script de shell DTD Generate

**idresgen.sh**

Script de shell ID Resolve

**massextract.sh**

Script de shell Extract

**massload.sh**

Script de shell Load

**txttransform.sh**

Script de shell de Text Transform

**xmltransform.sh**

Script de shell XML Transform

---

## Parte 3. Utilizando o Editor da Web

Esta seção descreve como administrar e utilizar o Editor da Web do Catalog Manager.

O Editor da Web permite criar, excluir e fazer alterações nos dados do catálogo, por meio de um navegador da Web. Os formulários de entrada de dados para exibir e atualizar informações são centrais ao Editor da Web. No caso mais simples, os formulários correspondem a tabela no banco de dados do WebSphere Commerce. O administrador pode escolher utilizar os formulários padrão fornecidos ou personalizar os formulários disponíveis.

**Nota:** O editor da Web utiliza o Internet Explorer 5 e posterior.











---

## Capítulo 11. Configurando o Editor da Web







O administrador pode configurar o editor da Web para contribuir com dados para o WebSphere Commerce, incluindo extensões e personalizações para o esquema do WebSphere Commerce. O Editor da Web não foi projetado como um conjunto de formulários de entrada da Web específicos de uma instância particular do WebSphere Commerce. O Editor da Web foi projetado para ser flexível e personalizável, a fim de suportar as necessidades e funções específicas de diversas empresas.

Uma exibição do banco de dados é uma consulta armazenada em uma ou mais tabelas no banco de dados. Durante a instalação do Catalog Manager, um arquivo de amostra para criar uma exibição lógica de um produto do WebSphere Commerce é colocado em cada subdiretório específico do sistema (db2, oracle e os400), do seguinte diretório:

-  unidade:\WebSphere\CommerceServer\schema
-  unidade:\Arquivos de Programas\WebSphere\CommerceServer\schema
-  /usr/WebSphere/CommerceServer/schema
-   /opt/WebSphere/CommerceServer/schema
-  /QIBM/ProdData/WebCommerce/schema

Este arquivo de amostra, `wcs.view.sql`, contém uma exibição do produto criada para combinar as informações sobre produtos de várias tabelas. Ele contém a definição de dados de SQL da exibição do produto. Os administradores podem estudar este arquivo para planejar o desenvolvimento de suas próprias exibições de um banco de dados.

O arquivo de descrição de formulário XML padrão (`forms51_be.xml`) contém formulários projetados para facilitar a inclusão, edição e exclusão de dados em um banco de dados do WebSphere Commerce. Uma cópia deste arquivo está localizada no seguinte diretório:

-  unidade:\WebSphere\CommerceServer\xml\wcwebeditor\xml
-  unidade:\Arquivos de Programas\WebSphere\CommerceServer\xml\wcwebeditor\xml
-  /usr/WebSphere/CommerceServer/xml/wcwebeditor/xml
-   /opt/WebSphere/CommerceServer/xml/wcwebeditor/xml
-  /instroot/xml/wcwebeditor/xml

Este arquivo de configuração pode ser utilizado como está, ou pode ser alterado e aperfeiçoado pelo administrador que configura o Editor da Web. Para personalizar o arquivo de descrição de formulário XML, consulte as instruções incluídas posteriormente nesta seção.

**Nota:** O DTD Generator pode criar automaticamente formulários a serem utilizados pelo Editor da Web.




---

## Configurando o Editor da Web

Esta seção fornece informações sobre como configurar o Editor da Web. Embora o processo de instalação trate esta configuração inicialmente, um administrador pode utilizar estas informações para fazer coisas como reconfigurar o Editor da Web para utilizar outro banco de dados.

### Editando o Arquivo `webeditor.properties`

O Editor da Web tem alguns parâmetros de aplicativos que estão definidos no arquivo `webeditor.properties`. Ele está localizado no seguinte diretório:

-  `unidade:\WebSphere\AppServer\installedApps\WC_Enterprise_App_demo.ear\wcwebeditor.war\WEB-INF\classes\webeditor.properties`
-  `unidade:\Arquivos de Programas\WebSphere\AppServer\installedApps\WC_Enterprise_App_demo.ear\wcwebeditor.war\WEB-INF\classes\webeditor.properties`
-  `/usr/WebSphere/AppServer/installedApps/WC_Enterprise_App_demo.ear/wcwebeditor.war/WEB-INF/classes/webeditor.properties`
-   `/opt/WebSphere/AppServer/installedApps/WC_Enterprise_App_demo.ear/wcwebeditor.war/WEB-INF/classes/webeditor.properties`
-  `/QIBM/UserData/WEBASADV4/was_instance_name/installedApps/WC_Enterprise_App_wcs_instance_name.ear/wcwebeditor.war/WEN-INF/classes`

Ao editar o arquivo `webeditor.properties`, um administrador pode fazer coisas como alterar o arquivo utilizado para descrever os formulários que são exibidos no Editor da Web. Aqui está um exemplo do conteúdo de um arquivo `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=mall

(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)	'







## Alterando a Localização de Arquivos Temporários

A localização de arquivos temporários pode ser alterada removendo a marca de comentário e incluindo um valor na propriedade temp.dir no arquivo webeditor.properties.

Como alternativa, a propriedade Java java.io.tmpdir é utilizada para determinar onde os arquivos temporários são criados.

## Criando um Arquivo de Descrição de Formulário XML Utilizando o DTD Generator

O arquivo `forms51_be.xml` é um exemplo de um arquivo de descrição de formulário XML. Ele fornece um conjunto de formulários a serem utilizados pelo editor da Web. Uma cópia deste arquivo está localizada no seguinte diretório:

-  `unidade:\WebSphere\CommerceServer\xml\wcwebeditor\xml`
-  `unidade:\Arquivos de Programas\WebSphere\CommerceServer\xml\wcwebeditor\xml`
-  `/usr/WebSphere/CommerceServer/xml/wcwebeditor/xml`
-   `/opt/WebSphere/CommerceServer/xml/wcwebeditor/xml`
-  `/instroot/xml/wcwebeditor/xml`

As etapas a seguir descrevem como o administrador do sistema pode utilizar o DTD Generator para incluir novos formulários XML.

**Nota:** Antes de executar o procedimento descrito abaixo, renomeie o arquivo `forms51_be.xml` existente no diretório do editor da Web. Como alternativa, crie um arquivo de saída durante o seguinte procedimento com um novo nome e, em seguida, reconfigure o editor da Web para utilizar o arquivo criado recentemente. Consulte a seção anterior para obter instruções sobre como fazer isto.

Para criar formulários XML, execute o comando DTD Generate.

As etapas a seguir descrevem como criar os formulários XML:

1. Crie um arquivo temporário chamado `tables.txt` contendo os nomes das tabelas que deseja utilizar nos formulários.

Digite o nome de cada tabela em uma única linha como no exemplo a seguir:

```
catentry
catentdesc
catentship
inventory
```

2. Salve `tables.txt` no diretório em que o comando DTD Generate está localizado. Consulte Capítulo 10, “Configurando Comandos e Scripts do Loader Package” na página 77 para obter a localização instalada deste comando.
3. No prompt de comandos do sistema operacional, vá para o diretório no qual o comando DTD Generate está localizado.

4. Execute o comando DTD Generate digitando o seguinte:

▶ NT ▶ 2000

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

▶ AIX ▶ Solaris ▶ Linux

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

▶ 400

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

A chave de descrição de tabela (-xmlTableDesc ou XMLTABDESC) faz o DTD Generator criar uma nova descrição de formulário das tabelas, além de um DTD.

▶ NT ▶ 2000 ▶ AIX ▶ Solaris ▶ Linux

A opção -propFile especifica o nome de um arquivo de propriedades externo no qual o texto da ajuda, os valores padrão e as informações de descrição de campo podem ser armazenados.

5. Reconfigure o editor da Web para utilizar os arquivos criados recentemente, conforme descrito na seção anterior.
6. Reinicie o editor da Web no WebSphere Advanced Administrative Console. Para fazer isso, siga estes passos:
  - a. Expanda **Domínio Administrativo do WebSphere**.
  - b. Expanda **Aplicativos Corporativos**.
  - c. Clique com o botão direito do mouse em **WebSphere Commerce Enterprise Application - demo** e selecione **Parar**.
  - d. Aguarde até que a mensagem indique que o aplicativo foi parado.
  - e. Clique com o botão direito do mouse em **WebSphere Commerce Enterprise Application - demo** e selecione **Iniciar**.
  - f. Aguarde até que a mensagem indique que o aplicativo corporativo foi iniciado.
7. Para ver o novo formulário em um navegador da Web, abra a seguinte URL:

```
https://host_name:8000/wcm/webeditor
```

em que *host\_name* é o nome completo do host HTTP de seu WebSphere Application Server.

O editor da Web é exibido na janela do navegador com uma lista de todos os nomes de tabelas.

## Personalizando a Descrição de Formulário XML

Esta seção descreve como o administrador pode aperfeiçoar os formulários exibidos pelo editor da Web.

A descrição do formulário XML pode ser personalizada e aperfeiçoada definindo atributos e valores no próprio arquivo de descrição de formulário XML ou em um arquivo de propriedades separado.

**Nota:** O nome deste arquivo de propriedades deve ser especificado como o valor do atributo `resourcePackage` na tag `formList`. Se o nome do arquivo aparecer em um subdiretório de um diretório no caminho da classe, ele deve utilizar a especificação de pacote(`dot`).

A tabela a seguir lista os atributos de campos de formulários do editor da Web que podem ser alterados por um administrador.

Atributo de campo	Descrição
Currency	Faz os valores serem exibidos com um separador numérico específico de locale (como uma vírgula como um separador de milhar para os Estados Unidos)
DbColumn	Utilizado para mapear o nome do campo para a chave do arquivo de propriedades  Se os arquivos de propriedades específicos de locale estiverem sendo utilizados, isto indica que o valor nesta entrada deve corresponder ao que foi digitado no arquivo de propriedades. O DTD Generator anexa um esquema a esta entrada.
DefaultValue	Especifica um valor que é exibido no formulário de entrada de dados quando um usuário preenche um novo formulário  Este atributo pode ser definido pelo DTD Generator como o valor padrão do banco de dados. Ele pode ser uma cadeia estática, mas também pode incluir uma consulta escalar SQL em uma tabela com uma única linha. Para recuperar dados externos, utilize uma função definida pelo usuário na consulta. Por exemplo:  DefaultValue="SELECT CURRENT TIMETAMP FROM EXEC"  em que EXEC é definido e ocupado como:  CREATE TABLE EXEC (A CHAR(1)); INSERT INTO A VALUES('A');
dynamicSqlSelectionList	Faz um menu drop-down ser reconstruído para cada formulário

Atributo de campo	Descrição
FieldDescription	<p data-bbox="863 239 1443 296">Fornece uma descrição que é exibida junto ao campo de entrada no formulário</p> <p data-bbox="863 321 1443 436">O DTD Generator utilizará os comentários na coluna se houver algum durante a criação deste atributo. Se não houver comentários na coluna, o padrão será o nome da coluna.</p> <p data-bbox="863 462 1443 600">Este atributo pode ser definido no arquivo de propriedades específico de locale ou no arquivo de descrição de formulário XML. Se for especificado um valor no arquivo de propriedades, ele terá precedência.</p>

Atributo de campo	Descrição
FieldHelp	<p>Fornece uma breve descrição da ajuda do campo a ser exibido na barra de mensagens inferior do navegador quando o campo está em foco no formulário.</p> <p>Por padrão, ele contém uma mensagem simples para digitar dados no campo fornecido juntamente com o tipo de coluna dos dados.</p> <p>Este atributo pode ser definido no arquivo de propriedades específico de locale ou no arquivo de descrição de formulário XML. Se for especificado um valor no arquivo de propriedades, ele terá precedência.</p>
formatNumber	<p>Utilizado para instruir o editor da Web a não processar um número de maneira nenhuma</p> <p>Defina o atributo formatNumber como "false" para tratar o valor digitado como uma cadeia, exceto durante as consultas de extratos (nas quais o valor não é colocado entre aspas como seria uma cadeia). O padrão deste atributo é "true."</p>
Hidden	<p>Indica que o valor não será exibido no formulário mas ainda ficará disponível como um campo HTML oculto</p>
HideOnCreate	<p>Indica que o campo está disponível quando é composto um novo formulário</p> <p>Semelhante a showInCreateMode="false"; mas inclui o nome do campo como tipo oculto.</p>
Maxlength	<p>Especifica o comprimento da coluna do banco de dados</p> <p>É utilizado para assegurar que o usuário não digitará um valor com um comprimento maior do que o que pode ser armazenado no banco de dados.</p>
MinOccurs	<p>Indica se o campo é obrigatório</p> <p>Um valor de "1" significa obrigatório, um valor de "0" significa opcional.</p>
Name	<p>Especifica o nome da coluna do banco de dados.</p>
readOnly readOnlyForCreate readOnlyForEdit	<p>Controla quando um campo está disponível para edição e quando ele pode apenas ser lido pelo usuário</p> <p>readOnly="true" significa que um campo está sempre no modo somente leitura. readOnlyForCreate="true" significa que um campo está no modo somente leitura quando um novo formulário está sendo composto. readOnlyForEdit="true" significa que um campo está no modo somente leitura quando um formulário existente está sendo editado.</p>



Atributo de campo	Descrição
ShowColumnInList	<p data-bbox="865 237 1377 321">Especifica que o campo é uma das colunas que constituem uma exibição de dados com vários registros quando definido como "true"</p> <p data-bbox="865 342 1409 405">O DTD Generator define as primeiras seis colunas como "true."</p>

Atributo de campo	Descrição
showInCreateMode	Utilizado para ocultar um campo do formulário criado  Defina o atributo showInCreateMode como "false" para ocultar o campo no formulário criado exibido. O padrão deste atributo é "true."
ShowInSearchMode	Utilizado para ocultar determinadas colunas da página de critérios de pesquisa  Se este atributo estiver definido como "false," o campo fornecido não será exibido no formulário de pesquisa.
SqlSelectionList	Cria um menu drop-down no qual um usuário pode selecionar ao compor um novo formulário  A consulta deve retornar um conjunto de resultados de uma ou duas colunas. A primeira coluna é a lista de rótulos entre os quais o usuário pode fazer escolhas; a segunda coluna é a lista de valores reais a serem armazenados na tabela do banco de dados. Se for utilizada somente uma coluna, o valor exibido é armazenado na tabela. Este recurso é muito útil para reforçar relacionamentos entre chaves externas.
Type	Indica o tipo de coluna do banco de dados e quais tipos de verificações de validação devem ser executados nos dados  Por exemplo, se o tipo for "integer," isto indica que o aplicativo assegura que somente um inteiro válido será digitado neste campo.
UniqueKey	Indica que o campo é uma chave principal para uma tabela  Se este atributo estiver definido como "true," isto indica que o aplicativo reforçará que somente uma limitação única nos dados será digitada nesta coluna. Esta verificação é executada apenas em registros carregados no aplicativo. Ela não se estende aos registros no banco de dados.
ValidateInput	Utilizado para desativar a verificação de validação  Este atributo permite que o usuário digite texto em um campo numérico que será tratado pelo ID Resolver ou pelo XML Transformer.

## Editando Nomes de Formulários

A tag de formulário na descrição de formulário XML tem os atributos name e displayName.

- O atributo name deve ser definido como o nome da tabela do banco de dados ou exibição que ele representa.
- O atributo displayName fornece um nome fácil de utilizar no aplicativo editor da Web. Como alternativa, um atributo displayName para um formulário pode

ser definido no arquivo de propriedades especificado no atributo `resourcePackage` da tag `formList`. (O atributo `resourcePackage` é utilizado para suporte multilíngüe).

**Nota:** Se houver uma entrada no arquivo de propriedades, ela terá precedência sobre o que foi digitado no arquivo de descrição de formulário XML.

## Alterando a Descrição de um Campo

Aqui está um exemplo de descrição de campo em um arquivo de descrição de formulário 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=""
.
.
```

Talvez você queira alterá-lo para o seguinte:

```
<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=""
.
.
```

## Incluindo um Menu Drop-down

Você pode incluir uma lista de seleção na forma de um menu drop-down em um formulário do editor da Web para facilitar o preenchimento do formulário e limitar as seleções do usuário a um conjunto válido de opções.

Para atualizar as listas de seleção, o administrador pode reiniciar o editor da Web. Existe também um atributo do campo de descrição de formulário,

dynamicSqlSelectionList, que o administrador pode definir como "true" para assegurar que um menu drop-down seja sempre recarregado.

A consulta de uma lista de seleção pode ter um conjunto de resultados que retorna uma ou duas colunas. Se o resultado retornar duas colunas, a segunda coluna conterá os valores reais armazenados e a primeira coluna conterá os rótulos do usuário.

Uma lista de seleção é criada digitando uma consulta SQL no atributo sqlSelectionList da tag de campo ou criando uma enumeração no arquivo de descrição de formulário XML. Os dois métodos são mostrados no exemplo a seguir:

```
<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>
>
```

Existem campos nas tabelas do banco de dados do WebSphere Commerce que utilizam 1 e 0 para true e false. Para criar campos mais intuitivos ao usuário, é possível estabelecer enumerações para esses campos. O Catalog Manager fornece um script SQL para criar uma tabela de enumeração chamada "NUMDESC." Este script é chamado "createEnum.sql." Um administrador pode editar o atributo sqlSelectionList para criar uma lista de seleção utilizando a tabela ENUMDESC conforme exibido no exemplo abaixo:

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

## Incluindo Ajuda para Campos

A ajuda para campo é exibida na parte inferior da janela do navegador da Web para o campo atual em foco. O atributo fieldHelp na tag de campo no arquivo de descrição de formulário XML ou a chave fieldHelp no arquivo de propriedades pode ser utilizado para definir este valor.

Por exemplo, o arquivo de propriedades pode conter as seguintes especificações de ajuda para campo:

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

Se nenhum valor for especificado, será criada uma mensagem padrão "Digite um valor para *field\_name* aqui *field\_type*".

**Nota:** Se houver um arquivo de propriedades contendo informações do formulário, ele terá precedência sobre as entradas no arquivo de descrição de formulário XML.













## Personalizando Resultados da Pesquisa e a Lista de Sessões de Trabalho

O conjunto de colunas exibido na página de resultados da pesquisa pode ser personalizado. Existe um atributo showColumnInList da tag de campo no arquivo de descrição de formulário XML. Para incluir um campo entre os resultados da pesquisa, defina este atributo como "true"; caso contrário, ele não aparecerá como uma das colunas na exibição de resultados da pesquisa. O campo aparece quando o formulário é exibido.

---

## Editando o Arquivo weProcessList

O arquivo weProcessList permite que um administrador personalize os utilitários do Catalog Manager executados quando uma sessão de trabalho do editor da Web é processada.

-  O arquivo weProcessList.xml está localizado no seguinte diretório:
  -  unidade:\WebSphere\CommerceServer\xml\wcwebeditor\xml
  -  unidade:\Arquivos de Programas\WebSphere\CommerceServer\xml\wcwebeditor\xml
  -  /usr/WebSphere/CommerceServer/xml/wcwebeditor/xml
  -   /opt/WebSphere/CommerceServer/xml/wcwebeditor/xml
-  O arquivo weProcessListOracle.xml está localizado no seguinte diretório:
  -  unidade:\WebSphere\CommerceServer\xml\wcwebeditor\xml
  -  unidade:\Arquivos de Programas\WebSphere\CommerceServer\xml\wcwebeditor\xml
  -  /usr/WebSphere/CommerceServer/xml/wcwebeditor/xml
  -   /opt/WebSphere/CommerceServer/xml/wcwebeditor/xml

- 400 O arquivo `weProcessListAS400.xml` está localizado no seguinte diretório:
  - `/instroot/xml/wcwebeditor/xml`

Este arquivo contém modelos de envelopes para os diversos utilitários. Ele também pode conter referências a aplicativos personalizados que o administrador deseja executar.

Existe um conjunto de variáveis do sistema que podem ser utilizadas neste arquivo. A variável do sistema `%-dbname%`, por exemplo, faz o nome do banco de dados ser inserido no envelope que é gerado para determinada chamada de um utilitário como o Loader. O arquivo de descrição de formulário XML contém referências a estes processos que indicam quais devem ser chamados para inclusão, edição ou exclusão.

Aqui está um exemplo de um arquivo `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="%tempFileURI1%"/>
    </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"/>
    <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:** As referências do arquivo fazem distinção entre maiúsculas e minúsculas.

A tabela a seguir inclui uma lista de variáveis de substituição válidas reconhecidas pelo aplicativo.



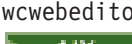



Variável de substituição %	Retorna
%-dbname%	Nome do banco de dados atual

%-dbuser%	Nome do usuário do banco de dados
%-dtdname%	Localização de URI do DTD para os arquivos XML
%-dbpwd%	Senha para o nome do usuário do banco de dados
%tempFilePath%	Caminho completo para um arquivo temporário
%tempFilePath1%	Estes são nomes de arquivos temporários únicos. Eles podem ser colocados na sintaxe de modelo de uma definição de envelope ou linha de comandos. Por exemplo, se %tempFilePath% for colocado no atributo de valor do parâmetro -infile em um modelo de envelope, o editor da Web gravará os dados da sessão de trabalho na localização do arquivo temporário.
%tempFilePath2%	
Em SubSystem	URI Temporário
%tempFileURI%	Os URIs temporários são URI para os mesmos arquivos representados por %tempFilePath%...%tempFilePath2%.Este não é um conjunto de arquivos adicionais, mas uma forma de recuperar o mesmo arquivo temporário gerado retornado com uma sintaxe diferente.
%tempFileURI1%	
%tempFileURI2%	
%previousOutFileAsURI%	Fornecer uma representação de URI das tarefas anteriores - parâmetro outfile como um URI
%webEditorDir%	Localização da instalação do editor da Web



## Editando o Arquivo `webeditor.xml`

O editor da Web utiliza o arquivo `webeditor.xml` como a página de estilo XSL padrão. Ele está localizado no seguinte diretório:

-  `unidade:\WebSphere\CommerceServer\xml\wcwebeditor\xsl`
-  `unidade:\Arquivos de Programas\WebSphere\CommerceServer\xml\wcwebeditor\xsl`
-  `/usr/WebSphere/CommerceServer/xml/wcwebeditor/xsl`
-   `/opt/WebSphere/CommerceServer/xml/wcwebeditor/xsl`
-  `/instroot/xml/wcwebeditor/xsl`

Editando o arquivo `webeditor.xml`, um administrador pode alterar o formato da saída do editor da Web. Aqui está um exemplo do conteúdo de um arquivo `webeditor.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>
```



---

## Capítulo 12. Trabalhando com Catálogos

Os procedimentos a seguir descrevem como trabalhar com catálogos utilizando o editor da Web para incluir, modificar e excluir registros nas tabelas do banco de dados que contêm dados do catálogo.

---

### Incluindo um Registro em uma Tabela Utilizando o Editor da Web

Para incluir um registro em uma tabela utilizando o editor da Web, siga estas etapas:

1. Em um navegador da Web, abra a seguinte URL:

```
https://host_name:8000/wcm/webeditor
```

em que *host\_name* é o nome completo do host HTTP de seu WebSphere Application Server.

A janela de logon do banco de dados do editor da Web é exibida.

2. Digite o nome do usuário e a senha do banco de dados e, em seguida, clique em **Logon**.

O editor da Web é exibido na janela do navegador com uma lista de nomes de tabelas na barra de menus esquerda.

3. Clique no hyperlink apropriado no submenu Incluir da barra de menus esquerda.

É exibido o formulário apropriado.

4. Digite todos os dados necessários no formulário.

5. Clique em **Mover para a sessão de trabalho**.

São exibidos os resultados da sessão de trabalho para o formulário. Estes resultados contêm todas as edições, inclusões e exclusões que foram feitas, mas não removidas ou processadas durante esta sessão de trabalho do editor da Web.

6. Se desejar remover as alterações do registro da sessão de trabalho, selecione a caixa na frente de cada alteração de registro que deseja remover e clique em **Limpar selecionado**.

7. Clique em **Processar sessão de trabalho** para submeter as alterações selecionadas ao banco de dados.

Uma página de status exibe uma mensagem indicando que o processo foi bem-sucedido.

8. Navegue para o site na Web, clique nos hyperlinks apropriados e verifique se suas alterações foram feitas.

---

## Modificando um Registro em uma Tabela Utilizando o Editor da Web

Para modificar um registro em uma tabela utilizando o editor da Web, siga estas etapas:

1. Em um navegador da Web, abra a seguinte URL:

`https://host_name:8000/wcm/webeditor`

em que *host\_name* é o nome completo do host HTTP de seu WebSphere Application Server.

A janela de logon do banco de dados do editor da Web é exibida.

2. Digite o nome do usuário e a senha do banco de dados e, em seguida, clique em **Logon**.  
O editor da Web é exibido na janela do navegador com uma lista de nomes de tabelas na barra de menus esquerda.
3. Clique no hyperlink apropriado no submenu Pesquisar da barra de menus esquerda.  
É exibida a página de pesquisa apropriada.
4. Especifique seus critérios de pesquisa fazendo o seguinte:
  - a. Selecione a caixa de opção ao lado de cada atributo que deseja especificar em sua pesquisa.
  - b. Utilize o menu drop-down apropriado para cada atributo selecionado para selecionar a lógica que deseja utilizar na pesquisa.
  - c. No próximo campo para cada atributo selecionado, digite ou selecione o valor que deseja utilizar na pesquisa.
5. Clique em **Localizar**.  
Isto submete os critérios de pesquisa ao editor da Web.  
É exibida uma página de status com o número de registros que atendem aos seus critérios de pesquisa.
6. Execute um dos seguintes procedimentos:
  - Clique em **Carregar dados** para ver uma lista dos registros encontrados.  
O editor da Web exibe uma lista de registros recuperados da consulta.  
Vá para a Etapa 6.
  - Clique em **Nova pesquisa** para retornar à página de pesquisa.  
Retorne à Etapa 3.
7. Selecione o registro que deseja editar.  
É exibido o formulário apropriado.
8. Role para baixo no campo que deseja editar e altere seu conteúdo.
9. Clique em **Mover para a sessão de trabalho**.  
São exibidos os resultados da sessão de trabalho para o formulário. Estes resultados contêm todas as edições, inclusões e exclusões que foram feitas, mas não removidas ou processadas durante esta sessão de trabalho do editor da Web.
10. Se desejar remover as alterações do registro da sessão de trabalho, selecione a caixa na frente de cada alteração de registro que deseja remover e clique em **Limpar selecionado**.
11. Clique em **Processar sessão de trabalho** para submeter as alterações selecionadas ao banco de dados.

Uma página de status exibe uma mensagem indicando que o processo foi bem-sucedido.

12. Navegue para o site na Web, clique nos hyperlinks apropriados e verifique se suas alterações foram feitas.

---

## Excluindo um Registro de uma Tabela Utilizando o Editor da Web

Para excluir um registro de uma tabela utilizando o editor da Web, siga estas etapas:

1. Em um navegador da Web, abra a seguinte URL:

`https://host_name:8000/wcm/webeditor`

em que *host\_name* é o nome completo do host HTTP de seu WebSphere Application Server.

A janela de logon do banco de dados do editor da Web é exibida.

2. Digite o nome do usuário e a senha do banco de dados e, em seguida, clique em **Logon**.  
O editor da Web é exibido na janela do navegador com uma lista de nomes de tabelas na barra de menus esquerda.
3. Clique no hyperlink apropriado no submenu Pesquisar da barra de menus esquerda.  
É exibida a página de pesquisa apropriada.
4. Especifique seus critérios de pesquisa fazendo o seguinte:
  - a. Selecione a caixa de opção ao lado de cada atributo que deseja especificar em sua pesquisa.
  - b. Utilize o menu drop-down apropriado para cada atributo selecionado para selecionar a lógica que deseja utilizar na pesquisa.
  - c. No próximo campo para cada atributo selecionado, digite ou selecione o valor que deseja utilizar na pesquisa.
5. Clique em **Localizar**.  
Isto submete os critérios de pesquisa ao editor da Web.  
É exibida uma página de status com o número de registros que atendem aos seus critérios de pesquisa.
6. Execute um dos seguintes procedimentos:
  - Clique em **Carregar dados** para ver uma lista dos registros encontrados.  
O editor da Web exibe uma lista de registros recuperados da consulta.  
Vá para a Etapa 6.
  - Clique em **Nova pesquisa** para retornar à página de pesquisa.  
Retorne à Etapa 3.
7. Selecione a caixa de opção ao lado de cada registro que deseja excluir.
8. Clique em **Mover para a lista de exclusão**.
9. Clique no hyperlink apropriado no submenu Sessão de Trabalho da barra de menus esquerda.  
São exibidos os resultados da sessão de trabalho para o formulário. Estes resultados contêm todas as edições, inclusões e exclusões que foram feitas, mas não removidas ou processadas durante esta sessão de trabalho do editor da Web.
10. Se desejar remover as alterações do registro da sessão de trabalho, selecione a caixa na frente de cada alteração de registro que deseja remover e clique em **Limpar selecionado**.
11. Clique em **Processar sessão de trabalho** para submeter as alterações ao banco de dados.

Uma página de status exibe uma mensagem indicando que o processo foi bem-sucedido.

12. Navegue para o site na Web, clique nos hyperlinks apropriados e verifique se suas alterações foram feitas.





---

## Parte 4. Referência de Comandos



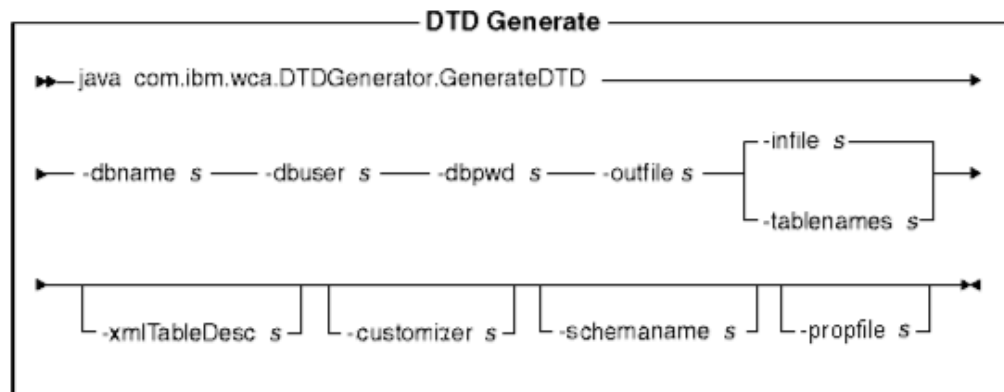
---

## Capítulo 13. Comando DTD Generate

Este comando cria arquivos DTD e esquema para serem utilizados com o Loader package.

---

### Comando DTD Generate para Sistemas Windows, AIX, Linux e Solaris



#### Notas:

1. O diagrama acima serve a princípio como uma referência para os parâmetros do comando. O script ou arquivo de comandos fornecido para esse comando e listado sob Capítulo 10, “Configurando Comandos e Scripts do Loader Package” na página 77 age como um wrapper para o comando Java real e aceita os mesmos parâmetros; portanto, é recomendado que o script ou arquivo de comandos seja utilizado em vez de chamar o comando Java diretamente.
2. Nomes de arquivos especificados como parâmetros para esse comando podem ser precedidos de caminhos relativos ou absolutos.

#### Valores de parâmetros:

##### **-dbname**

Nome do banco de dados de destino

##### **-dbuser**

Nome do usuário que se conecta ao banco de dados

##### **-dbpwd**

Senha do usuário que se conecta ao banco de dados

##### **-outfile**

Nome do arquivo DTD de saída

**-infile** Nome do arquivo de entrada que contém o nome da tabela do banco de dados em cada linha

##### **-tablenames**

Nomes de tabelas, separados por vírgulas

##### **-xmlTableDesc**

Caminho do arquivo de esquema a ser criado

##### **-customizer**

Nome do arquivo de propriedade do personalizador a ser utilizado.

DB2ConnectionCustomizer.properties é o arquivo padrão. O arquivo de propriedades do personalizador pode ser especificado conforme mostrado no seguinte exemplo:

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

Se este arquivo existir em um diretório especificado na variável de ambiente classpath do sistema, o mesmo arquivo poderá ser especificado conforme mostrado no seguinte exemplo:

```
-customizer dttdgen
```

**-schemaname**

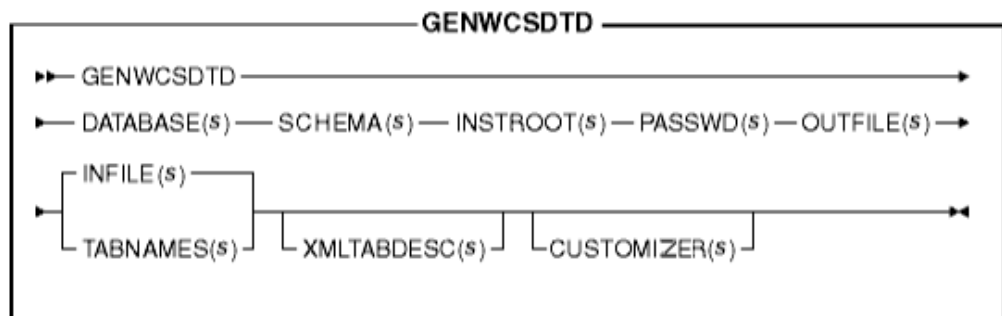
Nome do esquema do banco de dados de destino

**-propFile**

Arquivo que contém propriedades como um arquivo de propriedades externo no qual o texto da ajuda, valores padrão e as informações de descrição de campos podem ser armazenados para uma descrição de formulário do editor da Web

---

## Comando DTD Generate para Sistemas iSeries



**Nota:** Nomes de arquivos especificados como parâmetros para esse comando podem ser precedidos de caminhos relativos ou absolutos.

**Valores de parâmetros:**

**DATABASE**

Nome do banco de dados de destino, conforme exibido no diretório do banco de dados relacional

**SCHEMA**

Nome do esquema de banco de dados de destino; é o mesmo que o nome da instância

**INSTROOT**

Nome completo do caminho da raiz da instância do WebSphere Commerce, como /QIBM/UserData/WebCommerce/instances/*instance\_name*

**PASSWD**

Senha para a instância do WebSphere Commerce

**OUTFILE**

Nome do arquivo DTD de saída

**INFILE**

Nome do arquivo de entrada que contém o nome da tabela do banco de dados em cada linha

**TABNAMES**

Nomes de tabelas, separados por vírgulas

**XMLTABDESC**

Caminho do arquivo do arquivo de esquema a ser criado. Este parâmetro é opcional.

**CUSTOMIZER**

Nome do arquivo de propriedade do personalizador a ser utilizado. O arquivo padrão é `ISeries_GENWCSDTD_Customizer.properties`. O arquivo de propriedades do personalizador pode ser especificado conforme mostrado no seguinte exemplo:

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

Se este arquivo existir em um diretório especificado na variável de ambiente `classpath` do sistema, o mesmo arquivo poderá ser especificado conforme mostrado no seguinte exemplo:

```
CUSTOMIZER(dtdgen)
```



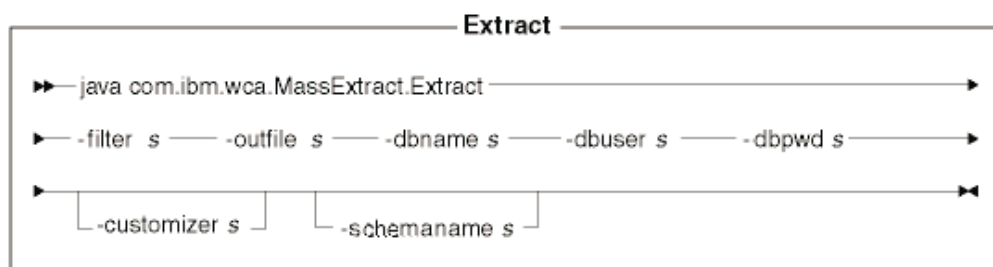
---

## Capítulo 14. Comando Extract

Esse comando extrai um sub-conjunto selecionado de dados de um banco de dados sob a forma de um arquivo XML.

---

### Comando Extract para Sistemas Windows, AIX, Linux e Solaris



#### Notas:

1. O diagrama acima serve a princípio como uma referência para os parâmetros do comando. O script ou arquivo de comandos fornecido para esse comando e listado sob Capítulo 10, "Configurando Comandos e Scripts do Loader Package" na página 77 age como um wrapper para o comando Java real e aceita os mesmos parâmetros; portanto, é recomendado que o script ou arquivo de comandos seja utilizado em vez de chamar o comando Java diretamente.
2. Nomes de arquivos especificados como parâmetros para esse comando podem ser precedidos de caminhos relativos ou absolutos.

#### Valores de parâmetros:

**-filter** Nome do arquivo do filtro de extração

**-outfile**

Nome do arquivo XML de saída no qual os dados extraídos serão armazenados

**-dbname**

Nome do banco de dados do qual estão sendo extraídos dados

**-dbuser**

Nome do usuário do banco de dados para o banco de dados do qual os dados estão sendo extraídos

**-dbpwd**

Senha associada ao nome do usuário para o banco de dados do qual os dados estão sendo extraídos

**-customizer**

Nome do arquivo de propriedade do personalizador a ser utilizado. `DB2ConnectionCustomizer.properties` é o arquivo padrão. O arquivo de propriedades do personalizador pode ser especificado conforme mostrado no seguinte exemplo:

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

Se este arquivo existir em um diretório especificado na variável de ambiente classpath do sistema, o mesmo arquivo poderá ser especificado conforme mostrado no seguinte exemplo:

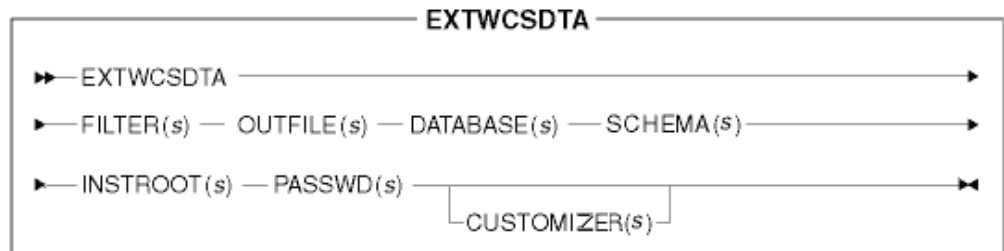
```
-customizer extract
```

**-schemaname**

Nome do esquema do banco de dados de destino

---

## Comando Extract para Sistemas iSeries



**Nota:** Nomes de arquivos especificados como parâmetros para esse comando podem ser precedidos de caminhos relativos ou absolutos.

**Valores de parâmetros:**

**FILTER**

Nome do arquivo do filtro de extração

**OUTFILE**

Nome do arquivo XML de saída no qual os dados extraídos serão armazenados

**DATABASE**

Nome do banco de dados do qual estão sendo extraídos dados, conforme exibido no diretório do banco de dados relacional

**SCHEMA**

Nome do esquema de banco de dados a partir do qual dados estão sendo extraídos; é o mesmo que o nome da instância

**INSTROOT**

Nome completo do caminho da raiz da instância do WebSphere Commerce, como /QIBM/UserData/WebCommerce/instances/*instance\_name*

**PASSWD**

Senha para a instância do WebSphere Commerce

**CUSTOMIZER**

Nome do arquivo de propriedade do personalizador a ser utilizado. O arquivo padrão é ISeries\_EXTWCSDTA\_Customizer.properties. O arquivo de propriedades do personalizador pode ser especificado conforme mostrado no seguinte exemplo:

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

Se este arquivo existir em um diretório especificado na variável de ambiente classpath do sistema, o mesmo arquivo poderá ser especificado conforme mostrado no seguinte exemplo:

```
CUSTOMIZER(extract)
```



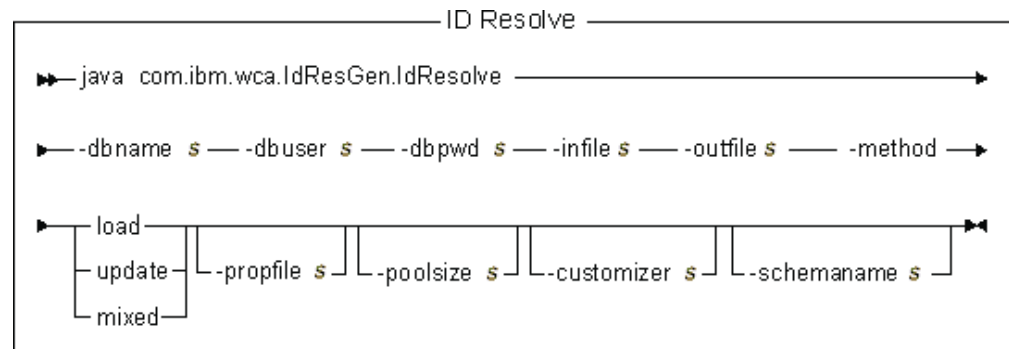
---

## Capítulo 15. Comando ID Resolve

Esse comando gera identificadores para elementos XML que os requerem antes de carregar em um banco de dados.

---

### Comando ID Resolve para Sistemas Windows, AIX, Linux e Solaris



#### Notas:

1. O diagrama acima serve a princípio como uma referência para os parâmetros do comando. O script ou arquivo de comandos fornecido para esse comando e listado sob Capítulo 10, “Configurando Comandos e Scripts do Loader Package” na página 77 age como um wrapper para o comando Java real e aceita os mesmos parâmetros; portanto, é recomendado que o script ou arquivo de comandos seja utilizado em vez de chamar o comando Java diretamente.
2. Nomes de arquivos especificados como parâmetros para esse comando podem ser precedidos de caminhos relativos ou absolutos.

#### Valores de parâmetros:

##### -dbname

Nome do banco de dados de destino

##### -dbuser

Nome do usuário que se conecta ao banco de dados

##### -dbpwd

Senha do usuário que se conecta ao banco de dados

**-infile** O nome do documento XML de entrada que contém descritores de registro de tabela

##### -outfile

Nome do arquivo XML de saída a ser produzido; este arquivo pode ser utilizado como entrada para o Loader.

##### -method

Método a ser utilizado para processar o arquivo de entrada

- Utilize o método **load** para processar o arquivo de entrada se **todos** os registros no arquivo **não existirem** no banco de dados.
- Utilize o método **update** para processar o arquivo de entrada se **todos** os registros no arquivo **existirem** no banco de dados.

- Utilize o método mixed para processar o arquivo de entrada se *apenas alguns* registros no arquivo *existirem* no banco de dados.

O método padrão é load.

**-profile**

O arquivo de texto que contém propriedades Java na forma de pares nome=valor. Esse arquivo é utilizado para definir os nomes de colunas a parte para uma consulta de identificadores de chave externa e o predicado de seleção para consultas de tabela principal (como CATEGORY e PRODUCT). Você pode omitir entradas nesse arquivo para tabelas que possuem um índice exclusivo definido que não inclui o identificador. O arquivo padrão é IdResolveKeys.properties. Apesar desse arquivo ter uma extensão .properties, não utilize a extensão quando especificar o valor.

**-poolsize**

Número de identificadores a serem reservados. O número padrão é 50.

**-customizer**

Nome do arquivo de propriedade do personalizador a ser utilizado. DB2ConnectionCustomizer.properties é o arquivo padrão. O arquivo de propriedades do personalizador pode ser especificado conforme mostrado em um dos seguintes exemplos:

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

Se este arquivo existir no diretório atual, o mesmo arquivo pode ser especificado como mostrado no exemplo a seguir:

```
-customizer idres.properties
```

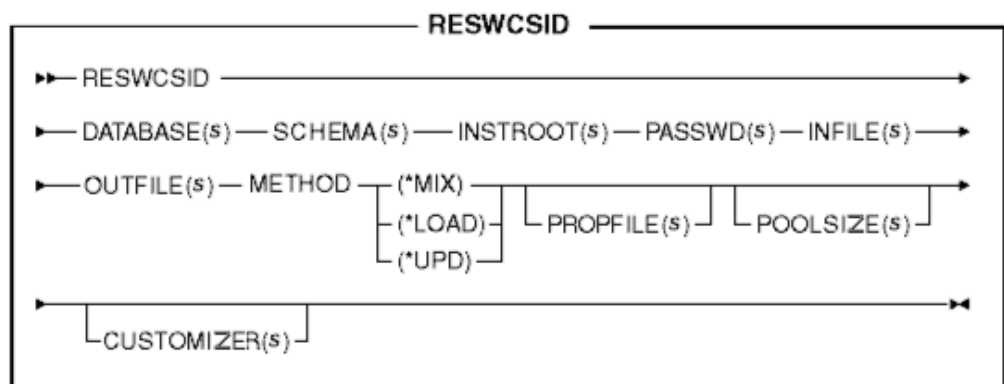
Se este arquivo existir em um diretório especificado na variável de ambiente classpath do sistema, o mesmo arquivo poderá ser especificado conforme mostrado no seguinte exemplo:

```
-customizer idres
```

**-schemaname**

Nome do esquema do banco de dados de destino

## Comando ID Resolve para Sistemas iSeries



**Nota:** Nomes de arquivos especificados como parâmetros para esse comando podem ser precedidos de caminhos relativos ou absolutos.

## Valores de parâmetros:

### DATABASE

Nome do banco de dados de destino, conforme exibido no diretório do banco de dados relacional

### SCHEMA

Nome do esquema de banco de dados de destino; é o mesmo que o nome da instância

### INSTROOT

Nome completo do caminho da raiz da instância do WebSphere Commerce, como `/QIBM/UserData/WebCommerce/instances/instance_name`

### PASSWD

Senha para a instância do WebSphere Commerce

### INFILE

O nome do documento XML de entrada que contém descritores de registro de tabela

### OUTFILE

Nome do arquivo XML de saída a ser produzido; este arquivo pode ser utilizado como entrada para o Loader.

### METHOD

Método a ser utilizado para processar o arquivo de entrada

- Utilize o método load (\*LOAD) para processar o arquivo de entrada se *nenhum* registro no arquivo *existir* no banco de dados.
- Utilize o método update (\*UPD) para processar o arquivo de entrada se *todos* os registros no arquivo *existirem* no banco de dados.
- Utilize o método mixed (\*MIX) para processar o arquivo de entrada se *apenas alguns* registros no arquivo *existirem* no banco de dados.

### PROFFILE

O arquivo de texto que contém propriedades Java na forma de pares nome=valor. Esse arquivo é utilizado para definir os nomes de colunas a parte para uma consulta de identificadores de chave externa e o predicado de seleção para consultas de tabela principal (como CATEGORY e PRODUCT). Você pode omitir entradas nesse arquivo para tabelas que possuem um índice exclusivo definido que não inclui o identificador. O arquivo padrão é `IdResolveKeys.properties`. Apesar desse arquivo ter uma extensão `.properties`, não utilize a extensão quando especificar o valor.

### POOLSIZE

Número de identificadores a serem reservados. O número padrão é 50.

### CUSTOMIZER

Nome do arquivo de propriedade do personalizador a ser utilizado. O arquivo padrão é `ISeries_RESWCSID_Customizer.properties`. O arquivo de propriedades do personalizador pode ser especificado conforme mostrado em um dos seguintes exemplos:

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

```
CUSTOMIZER(/wc/prop/idres)
```

Se este arquivo existir no diretório atual, o mesmo arquivo pode ser especificado como mostrado no exemplo a seguir:

```
CUSTOMIZER(idres.properties)
```

Se este arquivo existir em um diretório especificado na variável de ambiente classpath do sistema, o mesmo arquivo poderá ser especificado conforme mostrado no seguinte exemplo:

```
CUSTOMIZER(idres)
```

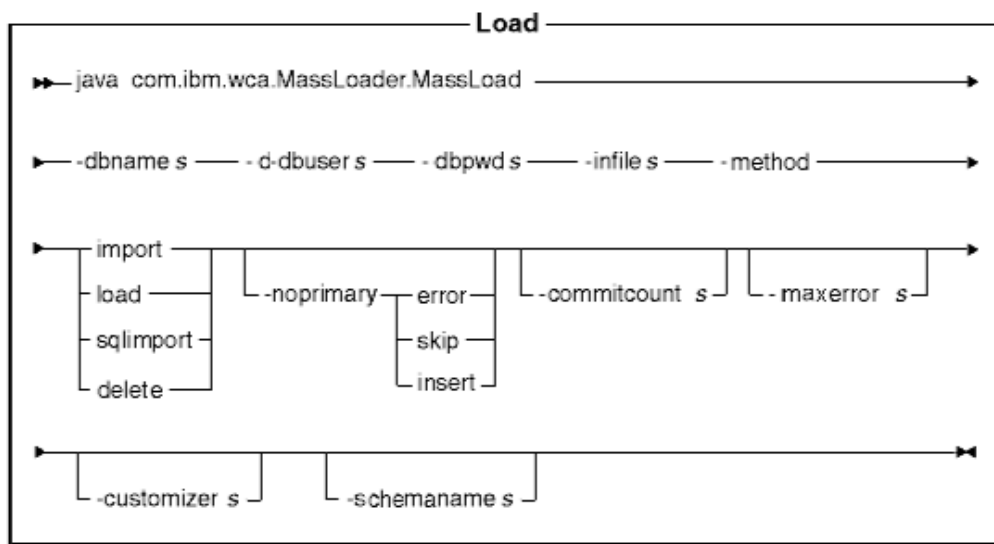
---

## Capítulo 16. Comando Load

Esse comando carrega um arquivo de entrada XML em um banco de dados de destino.

---

### Comando Load para Sistemas Windows, AIX, Linux e Solaris



#### Notas:

1. O diagrama acima serve a princípio como uma referência para os parâmetros do comando. O script ou arquivo de comandos fornecido para esse comando e listado sob Capítulo 10, "Configurando Comandos e Scripts do Loader Package" na página 77 age como um wrapper para o comando Java real e aceita os mesmos parâmetros; portanto, é recomendado que o script ou arquivo de comandos seja utilizado em vez de chamar o comando Java diretamente.
2. Nomes de arquivos especificados como parâmetros para esse comando podem ser precedidos de caminhos relativos ou absolutos.

#### Valores de parâmetros:

##### **-dbname**

Nome do banco de dados de destino

##### **-dbuser**

Nome do usuário que se conecta ao banco de dados

##### **-dbpwd**

Senha do usuário que se conecta ao banco de dados

**-infile** Nome do arquivo XML de entrada

##### **-method**

O modo de operação a ser utilizado pelo Loader ao inserir dados no banco de dados

- O método load utiliza o carregador nativo do fornecedor do banco de dados. Você pode utilizar o método load para bancos de dados Oracle locais e remotos; mas o método load pode ser utilizado somente para bancos de dados DB2 locais.
- Utilize o método import para carregar dados em bancos de dados DB2 remotos. O método import utiliza a opção import ou update se estiverem disponíveis no fornecedor do banco de dados. Se você especificar este método para um banco de dados no qual a opção import ou update não está disponível, como o Oracle, as instruções SQL utilizando JDBC serão utilizadas para atualizar o banco de dados.
- O método SQL import (sqlimport) pode ser utilizado com bancos de dados locais e remotos.
- O método delete exclui dados do banco de dados.

Se estiver utilizando a sincronização de espaço de procura do Consultor de Produto, será necessário utilizar o método sqlimport ou delete.

**-noprimary**

A ação que o Loader deve realizar quando a chave principal está ausente para um registro no arquivo de entrada. A opção error indica que ele deve relatar a chave principal ausente como um erro e terminar. A opção skip ignora os registros no arquivo de entrada que não possuem uma chave primária. A opção insert tenta processar (inserir ou excluir) os dados. A ação padrão é error.

**-commitcount**

Número de registros processados antes de ocorrer a consolidação do banco de dados durante a utilização do método de operação SQL update. O número padrão é 1.

**-maxerror**

Número de erros após o qual o Loader terminará no método de operação SQL update

**-customizer**

Nome do arquivo de propriedade do personalizador a ser utilizado. MassLoadCustomizer.properties é o arquivo padrão. O arquivo de propriedades do personalizador pode ser especificado conforme mostrado no seguinte exemplo:

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

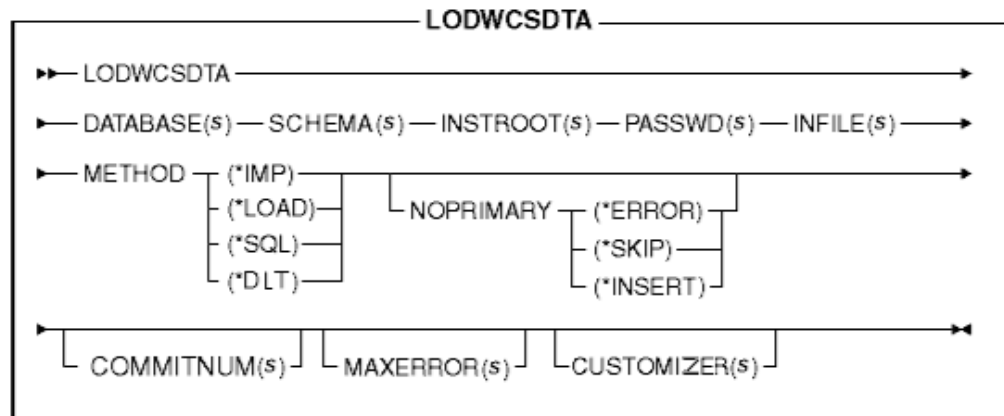
Se este arquivo existir em um diretório especificado na variável de ambiente classpath do sistema, o mesmo arquivo poderá ser especificado conforme mostrado no seguinte exemplo:

```
-customizer ml
```

**-schemaname**

Nome do esquema do banco de dados de destino

## Comando Load para Sistemas iSeries



**Nota:** Nomes de arquivos especificados como parâmetros para esse comando podem ser precedidos de caminhos relativos ou absolutos.

### Valores de parâmetros:

#### DATABASE

Nome do banco de dados de destino, conforme exibido no diretório do banco de dados relacional

#### SCHEMA

Nome do esquema de banco de dados de destino; é o mesmo que o nome da instância

#### INSTROOT

Nome completo do caminho da raiz da instância do WebSphere Commerce, como /QIBM/UserData/WebCommerce/instances/*instance\_name*

#### PASSWD

Senha para a instância do WebSphere Commerce

#### INFILE

Nome do arquivo XML de entrada

#### METHOD

O modo de operação a ser utilizado pelo Loader ao inserir dados no banco de dados

- O método load (\*LOAD) utiliza o carregador nativo no fornecedor do banco de dados. Você pode utilizar o método load (\*LOAD) para bancos de dados Oracle locais e remotos; mas o método load (\*LOAD) pode ser utilizado somente para bancos de dados DB2 locais.
- Utilize a opção import (\*IMP) para carregar dados em bancos de dados DB2 remotos. O método import (\*IMP) utiliza a opção import ou update se estiver disponível no fornecedor do banco de dados. Se a opção import ou update não estiver disponível, instruções SQL que utilizam JDBC serão utilizadas para atualizar o banco de dados.
- O método SQL import (\*SQL) pode ser utilizado com bancos de dados locais e remotos.
- O método delete (\*DLT) exclui dados do banco de dados.

#### NOPRIMARY

A ação que o Loader deve realizar quando a chave principal está ausente

para um registro no arquivo de entrada. A opção de erro (\*ERROR) indica que ele deve relatar a chave principal ausente como erro e encerrar. A opção skip (\*SKIP) ignora os registros no arquivo de entrada que não têm uma chave principal. A opção insert (\*INSERT) tenta processar (inserir ou excluir) os dados. A ação padrão é error.

**COMMITNUM**

Número de registros processados antes de ocorrer a consolidação do banco de dados durante a utilização do método de operação SQL update. O número padrão é 1.

**MAXERROR**

Número de erros após o qual o Loader terminará no método de operação SQL update

**CUSTOMIZER**

Nome do arquivo de propriedade do personalizador a ser utilizado. O arquivo padrão é ISeries\_LODWCSDTA\_Customizer.properties. O arquivo de propriedades do personalizador pode ser especificado conforme mostrado no seguinte exemplo:

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

Se este arquivo existir em um diretório especificado na variável de ambiente classpath do sistema, o mesmo arquivo poderá ser especificado conforme mostrado no seguinte exemplo:

```
CUSTOMIZER(ml)
```



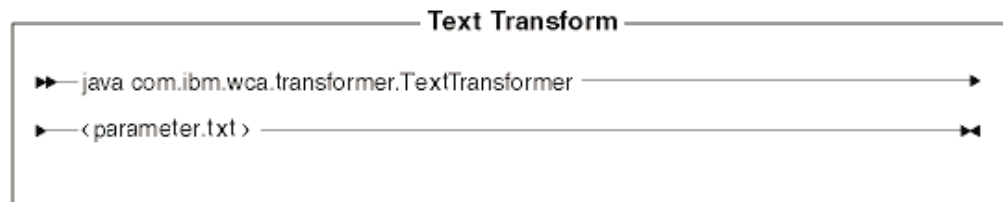
---

## Capítulo 17. Comando Text Transform

Este comando transforma dados entre um formato variável delimitado por caracteres e um formato XML.

---

### Comando Text Transform para Sistemas Windows, AIX, Linux e Solaris



**Nota:** O diagrama acima serve a princípio como uma referência para os parâmetros do comando. O script ou arquivo de comandos fornecido para esse comando e listado sob Capítulo 10, "Configurando Comandos e Scripts do Loader Package" na página 77 age como um wrapper para o comando Java real e aceita os mesmos parâmetros; portanto, é recomendado que o script ou arquivo de comandos seja utilizado em vez de chamar o comando Java diretamente.

#### Valores dos Parâmetros

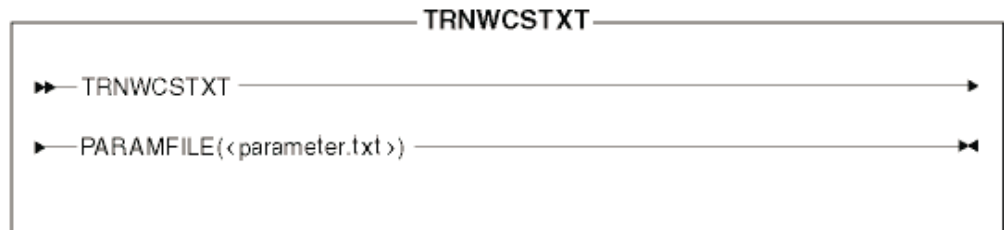
Os valores a seguir são especificados e separados por vírgulas em um arquivo de parâmetros (*parameter.txt*):

- arquivo de entrada  
Nome do arquivo a ser transformado
- arquivo esquema  
Nome do arquivo esquema XML a ser utilizado na transformação
- arquivo de saída  
Nome do arquivo de saída no qual os dados transformados serão armazenados
- método transformation  
Método a ser utilizado na inclusão de dados no arquivo de saída. Especifique **Criar** se um novo arquivo for ser criado ou **Anexar** se os dados de saída forem anexados a um arquivo de dados já existente.

**Nota:** Esse arquivo também é referido como arquivo "manifest" ou "command".

---

## Comando Text Transform para Sistemas iSeries



### Valores dos Parâmetros

Os valores a seguir são especificados e separados por vírgulas em um arquivo de parâmetros (*parameter.txt*):

- arquivo de entrada  
Nome do arquivo a ser transformado
- arquivo esquema  
Nome do arquivo esquema XML a ser utilizado na transformação
- arquivo de saída  
Nome do arquivo de saída no qual os dados transformados serão armazenados
- método transformation  
Método a ser utilizado na inclusão de dados no arquivo de saída. Especifique **Criar** se um novo arquivo for ser criado ou **Anexar** se os dados de saída forem anexados a um arquivo de dados já existente.

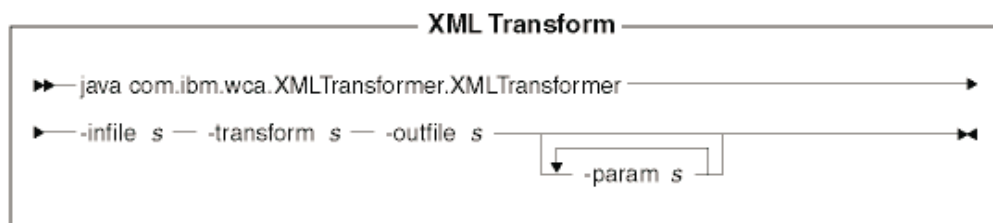
---

## Capítulo 18. Comando XML Transform

Esse comando converte um arquivo XML em um formato XML alternativo.

---

### Comando XML Transform para Sistemas Windows, AIX, Linux e Solaris



#### Notas:

1. O diagrama acima serve a princípio como uma referência para os parâmetros do comando. O script ou arquivo de comandos fornecido para esse comando e listado sob Capítulo 10, "Configurando Comandos e Scripts do Loader Package" na página 77 age como um wrapper para o comando Java real e aceita os mesmos parâmetros; portanto, é recomendado que o script ou arquivo de comandos seja utilizado em vez de chamar o comando Java diretamente.
2. Nomes de arquivos especificados como parâmetros para esse comando podem ser precedidos de caminhos relativos ou absolutos.

#### Valores de parâmetros:

**-infile** Nome do arquivo a ser transformado

**-transform**

Nome do arquivo de regras XSL de transformação

**-outfile**

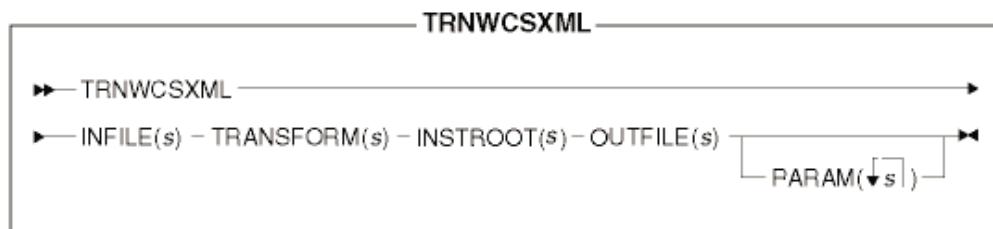
Nome do arquivo XML de saída no qual os dados transformados serão armazenados

**-param**

Parâmetro a ser transmitido ao arquivo de regras XSL. *Este parâmetro é opcional.* Este parâmetro pode ser especificado várias vezes para transmitir vários pares "name=value".

---

### Comando XML Transform para Sistemas iSeries



**Nota:** Nomes de arquivos especificados como parâmetros para esse comando podem ser precedidos de caminhos relativos ou absolutos.

**Valores de parâmetros:**

**INFILE**

Nome do arquivo a ser transformado

**TRANSFORM**

Nome do arquivo de regras XSL de transformação

**INSTROOT**

Nome completo do caminho da raiz da instância do WebSphere Commerce, como /QIBM/UserData/WebCommerce/instances/*instance\_name*

**OUTFILE**

Nome do arquivo XML de saída no qual os dados transformados serão armazenados

**PARAM**

Parâmetro a ser transmitido ao arquivo de regras XSL. *Este parâmetro é opcional.* A cadeia pode conter vários valores para transmitir vários pares "name=value".

---

## Avisos

Estas informações foram desenvolvidas para produtos e serviços oferecidos nos Estados Unidos. É possível que a IBM não ofereça os produtos, serviços ou recursos discutidos neste documento em outros países. Consulte um representante IBM local para obter informações sobre produtos e serviços disponíveis atualmente em sua área. Qualquer referência a produtos, programas ou serviços IBM não significa que apenas produtos, programas ou serviços IBM possam ser utilizados. Qualquer produto, programa ou serviço funcionalmente equivalente, que não infrinja quaisquer direitos de propriedade intelectual da IBM ou outros direitos legalmente protegidos, poderá ser utilizado em substituição a este produto, programa ou serviço. Entretanto, a avaliação e verificação da operação de qualquer produto, programa ou serviço não-IBM são de responsabilidade do Cliente.

A IBM pode ter patentes ou solicitações de patentes pendentes relativas a assuntos tratados nesta publicação. O fornecimento deste documento não garante ao Cliente nenhum direito sobre tais patentes. Pedidos de licença devem ser enviados, por escrito, à:

Gerência de Relações Comerciais e Industriais da IBM Brasil  
Av. Pasteur, 138/146  
Botafogo  
Rio de Janeiro - RJ  
CEP: 22290-240

Para pedidos de licença relacionados a informações de byte-duplo (DBCS), entre em contato com o Departamento de Propriedade Intelectual da IBM em seu país ou envie pedidos de licença, por escrito, para:

IBM World Trade Asia Corporation  
Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106-0032, Japan

**O parágrafo a seguir não se aplica a nenhum país em que tais disposições não estejam de acordo com a legislação local:** A INTERNATIONAL BUSINESS MACHINES CORPORATION FORNECE ESTA PUBLICAÇÃO “NO ESTADO EM QUE SE ENCONTRA”, SEM GARANTIA DE NENHUM TIPO, SEJA EXPRESSA OU IMPLÍCITA, INCLUINDO, MAS NÃO SE LIMITANDO ÀS GARANTIAS IMPLÍCITAS DE NÃO-VIOLAÇÃO, MERCADO OU ADEQUAÇÃO A UM DETERMINADO PROPÓSITO. Alguns países não permitem a exclusão de garantias expressas ou implícitas em certas transações; portanto, esta disposição pode não se aplicar ao Cliente.

Esta publicação pode conter imprecisões técnicas ou erros tipográficos. Periodicamente, são feitas alterações nas informações aqui contidas; tais alterações serão incorporadas em futuras edições desta publicação. A IBM pode, a qualquer momento, aperfeiçoar e/ou alterar os produtos e/ou programas descritos nesta publicação, sem aviso prévio.

Referências nestas informações a sites não-IBM na Web são fornecidas apenas por conveniência e não representam de forma alguma um endosso a esses sites na

Web. Os materiais contidos nestes sites da Web não fazem parte dos materiais deste produto IBM, e a utilização desses sites da Web é de inteira responsabilidade do Cliente.

A IBM pode utilizar ou distribuir as informações fornecidas da forma que julgar apropriada sem incorrer em qualquer obrigação para com o Cliente.

Licenciados deste programa que desejam obter informações sobre este assunto com o objetivo de permitir: (i) a troca de informações entre programas criados independentemente e outros programas (incluindo este) e (ii) a utilização mútua das informações trocadas, devem entrar em contato com:

Gerência de Relações Comerciais e Industriais da IBM Brasil  
Av. Pasteur, 138/146  
Botafogo  
Rio de Janeiro - RJ  
CEP: 22290-240

Tais informações podem estar disponíveis, sujeitas a termos e condições apropriadas, incluindo em alguns casos, o pagamento de uma taxa.

O programa licenciado descrito neste documento e todo o material licenciado disponível são fornecidos pela IBM sob os termos do Contrato com o Cliente IBM, do Contrato de Licença do Programa Internacional IBM ou de qualquer outro contrato equivalente.

Todos os dados de desempenho aqui contidos foram obtidos em um ambiente controlado. Portanto, os resultados obtidos em outros ambientes operacionais podem variar significativamente. Algumas medições podem ter sido efetuadas em sistemas em nível de desenvolvimento e não há garantias de que estas medições serão idênticas nos sistemas geralmente disponíveis. Além disso, algumas medições podem ter sido estimadas através de extrapolação. Os resultados reais podem variar. Os usuários deste documento devem verificar os dados aplicáveis para seu ambiente específico.

As informações relativas a produtos não-IBM foram obtidas junto aos fornecedores dos respectivos produtos, de seus anúncios publicados ou de outras fontes disponíveis publicamente. A IBM não testou estes produtos e não pode confirmar a precisão do desempenho, compatibilidade ou quaisquer outras alegações relacionadas a produtos não-IBM. Dúvidas sobre os recursos de produtos não-IBM devem ser encaminhadas diretamente a seus fornecedores.

Todas as declarações relacionadas aos objetivos e intenções futuras da IBM estão sujeitas a alterações ou cancelamento sem aviso prévio e representam apenas metas e objetivos.

Todos os preços IBM apresentados são preços de revenda sugeridos pela IBM, são atuais e estão sujeitos a alteração sem aviso prévio. Os preços do revendedor podem variar.

Estas informações foram projetadas apenas com o propósito de planejamento. As informações aqui contidas estão sujeitas a alterações antes que os produtos descritos fiquem disponíveis.

Estas informações contêm exemplos de dados e relatórios utilizados nas operações diárias de negócios. Para ilustrá-los da forma mais completa possível, os exemplos

podem incluir nomes de indivíduos, empresas, marcas e produtos. Todos estes nomes são fictícios e qualquer semelhança com nomes e endereços utilizados por uma empresa real é mera coincidência.

#### LICENÇA DE COPYRIGHT:

Estas informações contêm exemplos de programas aplicativos na linguagem fonte, ilustrando as técnicas de programação em diversas plataformas operacionais. Você pode copiar, modificar e distribuir estes exemplos de programas sem a necessidade de pagar a IBM, com objetivos de desenvolvimento, utilização, marketing ou distribuição de programas aplicativos em conformidade com a interface de programação do aplicativo para a plataforma operacional para a qual os programas de exemplo são criados. Estes exemplos não foram testados completamente em todas as condições. Portanto, a IBM não pode garantir ou implicar a confiabilidade, manutenção ou função destes programas. Você pode copiar, modificar e distribuir estes exemplos de programas de qualquer maneira sem pagamento à IBM, com objetivos de desenvolvimento, utilização, marketing ou distribuição de programas aplicativos em conformidade com interfaces de programação de aplicativos da IBM.

Cada cópia ou parte destes programas de exemplo ou qualquer trabalho derivado, deve incluir um aviso de copyright com os dizeres:

©Copyright International Business Machines Corporation 2001.  
Partes deste código são derivados do Exemplo de Programas da IBM Corp.  
©Copyright IBM Corp. 2000, 2001. Todos os direitos reservados.

Se você estiver exibindo estas informações em cópia eletrônica, as fotografias e ilustrações coloridas podem não aparecer.

---

## Marcas e Marcas de Serviço

Os termos a seguir são marcas da International Business Machines Corporation nos Estados Unidos e/ou em outros países:

AIX  
DB2  
DB2 Universal Database  
IBM  
iSeries  
OS/400  
WebSphere

Microsoft, Windows, Windows NT e Windows 2000 são marcas registradas da Microsoft Corporation nos Estados Unidos e/ou em outros países.

Oracle é uma marca registrada da Oracle Corporation nos Estados Unidos e/ou em outros países.

Solaris, Java e todas as marcas e logotipos baseados em Java são marcas ou marcas registradas da Sun Microsystems Inc. nos Estados Unidos e/ou em outros países.

Outros nomes de empresas, produtos e serviços podem ser marcas ou marcas de serviço de terceiros.







**IBM**