



Informatica® PowerCenter
10.4.0

Guia de Transformações

© Copyright Informatica LLC 1999, 2020

Este software e a documentação são fornecidos somente sob um contrato de licença separado, contendo restrições sobre uso e divulgação. Não está permitida de forma alguma a reprodução ou a transmissão de qualquer parte deste documento (seja por meio eletrônico, fotocópia, gravação ou quaisquer outros meios) sem o consentimento prévio da Informatica LLC.

Informatica, o logotipo Informatica e PowerCenter são marcas comerciais ou marcas registradas da Informatica LLC nos Estados Unidos e em muitas jurisdições por todo o mundo. Uma lista atual das marcas comerciais da Informatica está disponível na Internet em <https://www.informatica.com/trademarks.html>. Os nomes de outras companhias e produtos podem ser nomes ou marcas comerciais de seus respectivos proprietários.

Partes desta documentação e/ou software estão sujeitas a copyright de terceiros, incluindo sem limitação: Copyright DataDirect Technologies. Todos os direitos reservados. Copyright © Sun Microsystems. Todos os direitos reservados. Copyright © RSA Security Inc. Todos os direitos reservados. Copyright © Ordinal Technology Corp. Todos os direitos reservados. Copyright © Aandacht c.v. Todos os direitos reservados. Copyright Genivia, Inc. Todos os direitos reservados. Copyright Isomorphic Software. Todos os direitos reservados. Copyright © Meta Integration Technology, Inc. Todos os direitos reservados. Copyright © Intalio. Todos os direitos reservados. Copyright © Oracle. Todos os direitos reservados. Copyright © Adobe Systems Incorporated. Todos os direitos reservados. Copyright © DataArt, Inc. Todos os direitos reservados. Copyright © ComponentSource. Todos os direitos reservados. Copyright © Microsoft Corporation. Todos os direitos reservados. Copyright © Rogue Wave Software, Inc. Todos os direitos reservados. Copyright © Teradata Corporation. Todos os direitos reservados. Copyright © Yahoo! Inc. Todos os direitos reservados. Copyright © Glyph & Cog, LLC. Todos os direitos reservados. Copyright © Thinkmap, Inc. Todos os direitos reservados. Copyright © Clearpace Software Limited. Todos os direitos reservados. Copyright © Information Builders, Inc. Todos os direitos reservados. Copyright © OSS Nokalva, Inc. Todos os direitos reservados. Copyright Edifecs, Inc. Todos os direitos reservados. Copyright Cleo Communications, Inc. Todos os direitos reservados. Copyright © International Organization for Standardization 1986. Todos os direitos reservados. Copyright © ej-technologies GmbH. Todos os direitos reservados. Copyright © Jaspersoft Corporation. Todos os direitos reservados. Copyright © International Business Machines Corporation. Todos os direitos reservados. Copyright © yWorks GmbH. Todos os direitos reservados. Copyright © Lucent Technologies. Todos os direitos reservados. Copyright © University of Toronto. Todos os direitos reservados. Copyright © Daniel Veillard. Todos os direitos reservados. Copyright © Unicode, Inc. Copyright IBM Corp. Todos os direitos reservados. Copyright © MicroQuill Software Publishing, Inc. Todos os direitos reservados. Copyright © PassMark Software Pty Ltd. Todos os direitos reservados. Copyright © LogiXML, Inc. Todos os direitos reservados. Copyright © 2003-2010 Lorenzi Davide, todos os direitos reservados. Copyright © Red Hat, Inc. Todos os direitos reservados. Copyright © The Board of Trustees of the Leland Stanford Junior University. Todos os direitos reservados. Copyright © EMC Corporation. Todos os direitos reservados. Copyright © Flexera Software. Todos os direitos reservados. Copyright © Jinfonet Software. Todos os direitos reservados. Copyright © Apple Inc. Todos os direitos reservados. Copyright © Telerik Inc. Todos os direitos reservados. Copyright © BEA Systems. Todos os direitos reservados. Copyright © PDFlib GmbH. Todos os direitos reservados. Copyright © Orientation in Objects GmbH. Todos os direitos reservados. Copyright © Tanuki Software, Ltd. Todos os direitos reservados. Copyright © Ricebridge. Todos os direitos reservados. Copyright © Sencha, Inc. Todos os direitos reservados. Copyright © Scalable Systems, Inc. Todos os direitos reservados. Copyright © jQWidgets. Todos os direitos reservados. Copyright © Tableau Software, Inc. Todos os direitos reservados. Copyright © MaxMind, Inc. Todos os direitos reservados. Copyright © TMatte Software s.r.o. Todos os direitos reservados. Copyright © MapR Technologies Inc. Todos os direitos reservados. Copyright © Amazon Corporate LLC. Todos os direitos reservados. Copyright © Highsoft. Todos os direitos reservados. Copyright © Python Software Foundation. Todos os direitos reservados. Copyright © BeOpen.com. Todos os direitos reservados. Copyright © CNRI. Todos os direitos reservados.

Este produto inclui software desenvolvido pela Apache Software Foundation (<http://www.apache.org/>) e/ou outros softwares licenciados nas várias versões da Licença Apache (a "Licença"). Você pode obter uma cópia dessas Licenças em <http://www.apache.org/licenses/>. A menos que exigido pela legislação aplicável ou concordado por escrito, o software distribuído em conformidade com estas Licenças é fornecido "NO ESTADO EM QUE SE ENCONTRA", SEM GARANTIA OU CONDIÇÃO DE QUALQUER TIPO, seja expressa ou implícita. Consulte as Licenças para conhecer as limitações e as permissões que regulam o idioma específico de acordo com as Licenças.

Este produto inclui software desenvolvido pela Mozilla (<http://www.mozilla.org/>), direitos autorais de software de The JBoss Group, LLC; todos os direitos reservados; software copyright © 1999-2006 de Bruno Lowagie e Paulo Soares e outros produtos de software licenciados sob a Licença Pública GNU Lesser General Public License Agreement, que pode ser encontrada em <http://www.gnu.org/licenses/lgpl.html>. Os materiais são fornecidos gratuitamente pela Informatica, no estado em que se encontram, sem garantia de qualquer tipo, explícita nem implícita, incluindo, mas não limitando-se, as garantias implicadas de comerciabilidade e adequação a um determinado propósito.

O produto inclui software ACE(TM) e TAO(TM) com copyright de Douglas C. Schmidt e seu grupo de pesquisa na Washington University, University of California, Irvine e Vanderbilt University, Copyright (©) 1993-2006, todos os direitos reservados.

Este produto inclui o software desenvolvido pelo OpenSSL Project para ser usado no kit de ferramentas OpenSSL (copyright The OpenSSL Project. Todos os direitos reservados) e a redistribuição deste software está sujeita aos termos disponíveis em <http://www.openssl.org> e <http://www.openssl.org/source/license.html>.

Este produto inclui o software Curl com o Copyright 1996-2013, Daniel Stenberg, <daniel@haxx.se>. Todos os direitos reservados. Permissões e limitações relativas a este software estão sujeitas aos termos disponíveis em <http://curl.haxx.se/docs/copyright.html>. É permitido usar, copiar, modificar e distribuir este software com qualquer objetivo, com ou sem taxa, desde que a nota de direitos autorais acima e esta nota de permissão apareçam em todas as cópias.

O produto inclui software copyright 2001-2005 (©) MetaStuff, Ltd. Todos os direitos reservados. Permissões e limitações relativas a este software estão sujeitas aos termos disponíveis em <http://www.dom4j.org/license.html>.

Este produto inclui o copyright de software © 1996-2006 Per Bothner. Todos os direitos reservados. O direito de usar tais materiais é estabelecido na licença que pode ser encontrada em <http://www.gnu.org/software/kawa/Software-License.html>.

Este produto inclui o software OSSP UUID com Copyright © 2002 Ralf S. Engelschall, Copyright © 2002 e OSSP Project Copyright © 2002 Cable & Wireless Deutschland. Permissões e limitações relativas a este software estão sujeitas aos termos disponíveis em <http://www.opensource.org/licenses/mit-license.php>.

Este produto inclui software desenvolvido pela Boost (<http://www.boost.org/>) ou sob a licença de software Boost. Permissões e limitações relativas a este software estão sujeitas aos termos disponíveis em http://www.boost.org/LICENSE_1_0.txt.

Este produto inclui software copyright © 1997-2007 University of Cambridge. Permissões e limitações relativas a este software estão sujeitas aos termos disponíveis em <http://www.pcre.org/license.txt>.

Este produto inclui o copyright de software © 2007 The Eclipse Foundation. Todos os direitos reservados. As permissões e as limitações relativas a este software estão sujeitas aos termos disponíveis em <http://www.eclipse.org/org/documents/epl-v10.php> e em <http://www.eclipse.org/org/documents/edl-v10.php>.

Este produto inclui softwares licenciados de acordo com os termos disponíveis em <http://www.tcl.tk/software/tcltk/license.html>, <http://www.bosrup.com/web/overlib/?License>, <http://www.stlport.org/doc/license.html>, <http://asm.ow2.org/license.html>, <http://www.cryptix.org/LICENSE.TXT>, <http://hsqldb.org/web/hsqldbLicense.html>, <http://httpunit.sourceforge.net/doc/license.html>, <http://jung.sourceforge.net/license.txt>, http://www.gzip.org/zlib/zlib_license.html, <http://www.openldap.org/software/release/license.html>, <http://www.libssh2.org>, <http://slf4j.org/license.html>, <http://www.sente.ch/software/OpenSourceLicense.html>, <http://fusesource.com/downloads/license-agreements/fuse-message-broker-v-5-3-license-agreement>, <http://antlr.org/license.html>, <http://aopalliance.sourceforge.net/>, <http://www.bouncycastle.org/license.html>, <http://www.jgraph.com/jgraphdownload.html>, <http://www.jcraft.com/jsch/LICENSE.txt>, http://jotm.objectweb.org/bsd_license.html, <http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231>, <http://www.slf4j.org/license.html>, <http://nanoxml.sourceforge.net/orig/copyright.html>, <http://www.json.org/license.html>, <http://forge.ow2.org/projects/javaservice/>, <http://www.postgresql.org/about/licence.html>, <http://www.sqlite.org/copyright.html>, <http://www.tcl.tk/software/tcltk/license.html>, <http://www.jaxen.org/faq.html>, <http://www.jdom.org/docs/faq.html>, <http://www.slf4j.org/license.html>;

<http://www.iodbc.org/dataspace/iodbc/wiki/IODBC/License>; <http://www.keplerproject.org/md5/license.html>; <http://www.toedter.com/en/jcalendar/license.html>; <http://www.edankert.com/bounce/index.html>; <http://www.net-snmp.org/about/license.html>; <http://www.openmdx.org/#FAQ>; http://www.php.net/license/3_01.txt; <http://srp.stanford.edu/license.txt>; <http://www.schneier.com/blowfish.html>; <http://www.jmock.org/license.html>; <http://xsom.java.net>; <http://benalman.com/about/license/>; <https://github.com/CreateJS/EaselJS/blob/master/src/easeljs/display/Bitmap.js>; <http://www.h2database.com/html/license.html#summary>; <http://jsoncpp.sourceforge.net/LICENSE>; <http://jdbc.postgresql.org/license.html>; <http://protobuf.googlecode.com/svn/trunk/src/google/protobuf/descriptor.proto>; <https://github.com/rantav/hector/blob/master/LICENSE>; <http://web.mit.edu/Kerberos/krb5-current/doc/mitK5license.html>; <http://jibx.sourceforge.net/jibx-license.html>; <https://github.com/lyokato/libgeohash/blob/master/LICENSE>; <https://github.com/hjiang/jsonxx/blob/master/LICENSE>; <https://code.google.com/p/lz4/>; <https://github.com/jedisct1/libsodium/blob/master/LICENSE>; <http://one-jar.sourceforge.net/index.php?page=documents&file=license>; <https://github.com/EsotericSoftware/kryo/blob/master/license.txt>; <http://www.scala-lang.org/license.html>; <https://github.com/tinkerpop/blueprints/blob/master/LICENSE.txt>; <http://gee.cs.oswego.edu/dl/classes/EDU/oswego/cs/dl/util/concurrent/intro.html>; <https://aws.amazon.com/asl/>; <https://github.com/twbs/bootstrap/blob/master/LICENSE>; <https://sourceforge.net/p/xmlunit/code/HEAD/tree/trunk/LICENSE.txt>.

Este produto inclui software licenciado de acordo com a Academic Free License (<http://www.opensource.org/licenses/afl-3.0.php>), a Common Development and Distribution License (<http://www.opensource.org/licenses/cddl1.php>), a Common Public License (<http://www.opensource.org/licenses/cpl1.0.php>), a Sun Binary Code License Agreement Supplemental License Terms, a BSD License (<http://www.opensource.org/licenses/bsd-license.php>), a nova BSD License (<http://opensource.org/licenses/BSD-3-Clause>), a MIT License (<http://www.opensource.org/licenses/mit-license.php>), a Artistic License (<http://www.opensource.org/licenses/artistic-license-1.0>) e a Initial Developer's Public License Version 1.0 (<http://www.firebirdsql.org/en/initial-developer-s-public-license-version-1-0/>).

Este produto inclui copyright do software © 2003-2006 Joe Walnes, 2006-2007 XStream Committers. Todos os direitos reservados. Permissões e limitações relativas a este software estão sujeitas aos termos disponíveis em <http://xstream.codehaus.org/license.html>. Este produto inclui software desenvolvido pelo Indiana University Extreme! Lab. Para obter mais informações, visite <http://www.extreme.indiana.edu/>.

Este produto inclui software Copyright © 2013 Frank Balluffi e Markus Moeller. Todos os direitos reservados. As permissões e limitações relativas a este software estão sujeitas aos termos da licença MIT.

Consulte as patentes em <https://www.informatica.com/legal/patents.html>.

ISENÇÃO DE RESPONSABILIDADE: a Informatica LLC fornece esta documentação no estado em que se encontra, sem garantia de qualquer tipo, expressa ou implícita, incluindo, mas não limitando-se, as garantias implícitas de não infração, comercialização ou uso para um determinado propósito. A Informatica LLC não garante que este software ou documentação não contenha erros. As informações fornecidas neste software ou documentação podem incluir imprecisões técnicas ou erros tipográficos. As informações deste software e documentação estão sujeitas a alterações a qualquer momento sem aviso prévio.

AVISOS

Este produto da Informatica (o "Software") traz determinados drivers (os "drivers da DataDirect") da DataDirect Technologies, uma empresa em funcionamento da Progress Software Corporation ("DataDirect"), que estão sujeitos aos seguintes termos e condições:

1. OS DRIVERS DA DATADIRECT SÃO FORNECIDOS NO ESTADO EM QUE SE ENCONTRAM, SEM GARANTIA DE QUALQUER TIPO, EXPRESSA OU IMPLÍCITA, INCLUINDO, MAS NÃO LIMITANDO-SE, AS GARANTIAS IMPLÍCITAS DE COMERCIALIZAÇÃO, ADEQUAÇÃO A UMA FINALIDADE ESPECÍFICA E NÃO INFRAÇÃO.
2. EM NENHUM CASO, A DATADIRECT OU SEUS FORNECEDORES TERCEIRIZADOS SERÃO RESPONSÁVEIS, EM RELAÇÃO AO CLIENTE FINAL, POR QUAISQUER DANOS DIRETOS, INDIRETOS, INCIDENTAIS, ESPECIAIS, CONSEQUENCIAIS OU DEMAIS QUE POSSAM ADVIR DO USO DE DRIVERS ODBC, SENDO OU NÃO ANTERIORMENTE INFORMADOS DAS POSSIBILIDADES DE TAIS DANOS. ESTAS LIMITAÇÕES SE APLICAM A TODAS AS CAUSAS DE AÇÃO, INCLUINDO, SEM LIMITAÇÕES, QUEBRA DE CONTRATO, QUEBRA DE GARANTIA, NEGLIGÊNCIA, RESPONSABILIDADE RIGOROSA, DETURPAÇÃO E OUTROS ATOS ILÍCITOS.

As informações contidas neste documento estão sujeitas a alteração sem aviso prévio. Se você encontrar quaisquer problemas nesta documentação, informe-os em infa_documentation@informatica.com.

Os produtos Informatica apresentam garantias segundo os termos e condições dos acordos em que são fornecidos. A INFORMATICA FORNECE AS INFORMAÇÕES NESTE DOCUMENTO "COMO ESTÃO" SEM GARANTIA DE QUALQUER TIPO, EXPRESSA OU IMPLÍCITA, INCLUINDO, SEM QUAISQUER GARANTIAS DE COMERCIALIZAÇÃO, ADEQUAÇÃO A UM DETERMINADO FIM E QUALQUER GARANTIA OU CONDIÇÃO DE NÃO-VIOLAÇÃO.

Data da Publicação: 2020-02-24

Conteúdo

Prefácio.....	22
Recursos da Informatica.	22
Rede da Informatica.	22
Base de Dados de Conhecimento da Informatica.	22
Documentação da Informatica.	23
Matrizes de Disponibilidade de Produto da Informatica.	23
Informatica Velocity.	23
Informatica Marketplace.	23
Suporte Global a Clientes da Informatica.	23
 Capítulo 1: Trabalhando com Transformações.....	24
Visão geral de transformações.	24
Transformações ativas.	25
Transformações passivas.	25
Transformações Desconectadas.	25
Transformações Nativas e Não Nativas.	25
Descrições de Transformações.	26
Criando uma transformação.	28
Configurando transformações.	29
Renomeando Transformações.	29
Portas de Transformação.	29
Criar Portas.	30
Configurar Portas.	30
Vinculando portas.	30
Transformações de vários grupos.	31
Trabalhando com expressões.	31
Usando o Editor de Expressão.	33
Avaliar expressões.	35
Avaliando uma expressão.	35
Restrições ao avaliar expressão.	36
Variáveis Locais.	36
Temporariamente armazenar dados e simplificar expressões complexas.	37
Armazenar Valores nas Linhas.	37
Capturar valores de procedimentos armazenados.	38
Diretrizes para configurar portas variáveis.	38
Valores Padrão para Portas.	39
Valores Padrão Definidos pelo Usuário.	40
Valores de Entrada Padrão Definidos pelo Usuário.	42
Valores de Saída Padrão Definidos pelo Usuário	43
Regras gerais para valores padrão.	46

Validação do Valor Padrão.	46
Configurando o nível de rastreamento em transformações.	47
Transformações Reutilizáveis.	47
Instâncias e alterações herdadas.	48
Mapeando variáveis em expressões.	48
Criando Transformações Reutilizáveis.	48
Promovendo Transformações Não Reutilizáveis.	49
Criando Instâncias Não Reutilizáveis de Transformações Reutilizáveis.	49
Adicionando Transformações Reutilizáveis a Mapeamentos.	50
Modificando uma transformação reutilizável.	50

Capítulo 2: Transformação Agregador..... 51

Visão Geral da Transformação Agregador.	51
Componentes da Transformação Agregador.	52
Configuração das Propriedades da Transformação Agregador.	52
Configuração das Portas da Transformação Agregador.	53
Configuração de caches agregados.	53
Expressões agregadas.	54
Funções de agregação.	54
Funções de Agregação Aninhadas.	54
Cláusulas condicionais.	55
Funções de não agregação.	55
Valores Null em funções de agregação.	55
Agrupar por Portas.	55
Expressões Não Agregadas.	57
Valores padrão.	57
Usando Entrada Classificada.	57
Condições de Entrada Classificadas.	58
Classificando Dados.	58
Criação de uma Transformação Agregador.	59
Dicas para Transformações Agregador.	59
Solução de Problemas de Transformações Agregador.	60

Capítulo 3: Transformação Personalizada..... 61

Visão Geral de Transformação Personalizada	61
Trabalhando com Transformações Criadas na Transformação Personalizada.	62
Compatibilidade de Página de Código.	62
Distribuindo Procedimentos de Transformação Personalizada.	63
Criando Transformações Personalizadas.	63
Regras e Diretrizes para Transformações personalizadas.	64
Componentes de Transformações Personalizadas.	64
Trabalhando com Grupos e Portas.	64
Criando Grupos e Portas.	64

Editando Grupos e Portas	65
Definindo Relações de Portas.	65
Trabalhando com Atributos de Porta.	66
Editando Valores de Atributo de Porta.	66
Propriedades de Transformação Personalizada.	67
Definindo a Estratégia de Atualização.	69
Trabalhando com Código de Procedimento Específico do Thread	69
Trabalhando com o Controle de Transação.	70
Escopo de Transformação.	70
Gerar Transação.	70
Trabalhando com Limites da Transação.	70
Bloqueando Dados de Entrada.	71
Gravando o Código do Procedimento para Bloquear Dados.	71
Configurando Transformações Personalizadas como Transformações de Bloqueio.	72
Validando Mapeamentos com Transformações Personalizadas.	72
Trabalhando com Propriedades de Procedimento.	73
Criando Procedimentos de Transformação Personalizada.	74
Etapa 1. Criar a transformação personalizada.	74
Etapa 2. Gerar os Arquivos C.	76
Etapa 3. Preencher o código com a lógica de transformação.	77
Etapa 4. Criar o Módulo.	82
Etapa 5. Criando um Mapeamento.	83
Etapa 6. Executar a Sessão em um Fluxo de Trabalho.	84
 Capítulo 4: Funções de Transformação Personalizada.	 85
Visão geral das Funções de Transformação Personalizada.	85
Trabalhando com Identificadores.	85
Referência de Função.	86
Trabalhando com Linhas.	89
Regras e Diretrizes para Modo de Acesso a Dados Baseado em Linhas e Baseado em Matriz.	90
Funções Geradas.	90
Funções de Inicialização.	91
Funções de Notificação.	92
Funções de Desinicialização.	94
Funções API.	96
Função de Definição do Modo de Acesso aos Dados.	97
Funções de Navegação.	97
Funções de Propriedade.	100
Funções de Religação de Tipo de Dados.	107
Funções de manipulação de dados (Modo Baseado em Linhas).	109
Função Configurar Porta de Passagem	112
Função de Notificação de Saída.	112
Função de Notificação de Saída de Limite de Dados.	113

Funções de Erro.	113
Funções de Mensagem do Log da Sessão.	114
Função de Incremento de Contagem de Erros.	114
Função Encerrado.	115
Funções de Bloqueio.	115
Funções de Ponteiro.	116
Função Alterar modo de string.	117
Função Definir Página de Código de Dados.	118
Funções de Estratégia de Linha (Modo Baseado em Linha).	119
Função Alterar Estratégia de Linha Padrão.	120
Funções de API baseadas em matriz.	121
Funções de Número Máximo de Linhas.	121
Funções de Número de Linhas.	122
Função Válido Para Linhas.	123
Funções de Manipulação de Dados (Modo Baseado em Matriz).	123
Funções de Estratégia de Linha (Modo Baseado em Matriz).	126
Funções Definir Linha de Erros de Entrada.	127
 Capítulo 5: Transformação Mascaramento de Dados.....	130
transformação de Mascaramento de Dados.	130
Propriedades de mascaramento.	131
Local.	131
Tipos de mascaramento.	132
Saída Repetível.	132
Semente.	132
Parâmetros de Mapeamento.	132
P/S Associada.	133
Mascaramento de Chave.	133
Mascaramento de Valores de String.	134
Mascaramento de Valores Numéricos.	135
Mascarando Valores de Data e Hora.	135
Mascaramento de Substituição.	135
Dicionários.	136
Tabelas de Armazenamento.	137
Propriedades do Mascaramento de Substituição.	138
Dicionário Relacional.	138
Requisitos de conexão.	139
Regras e Diretrizes para Mascaramento de Substituição.	139
Mascaramento dependente.	139
Exemplo de mascaramento dependente.	140
Mascaramento Dependente Repetível.	141
Mascaramento Aleatório.	141
Mascaramento de Valores Numéricos.	141

Mascaramento de Valores de String.	142
Mascaramento de Valores de Data	142
Aplicando Regras de Mascaramento.	143
Formato de Máscara.	143
Caracteres da String de Origem.	144
Caracteres de Substituição da String de Resultados.	145
Intervalo.	146
Embaçamento.	146
Mascaramento de Expressão.	147
Mascaramento Repetível de Expressão.	148
Regras e Diretrizes para Mascaramento de Expressão.	149
Formatos Especiais de Máscara.	149
Mascaramento de CPF.	150
Formato de Número da Previdência Social.	150
Requisito de Código de Área.	150
Mascaramento de CPF Repetível.	151
Mascaramento de Número de Cartão de Crédito.	151
Mascaramento de Número de Telefone.	152
Mascaramento de Endereço de E-mail.	152
Mascaramento de E-mail Avançado.	152
Mascaramento de Número de Previdência Social.	154
Dígito de Início SIN.	154
Números SIN Repetíveis.	154
Mascaramento de Endereço IP.	154
Mascaramento de Endereço de URL.	155
Arquivo de valor padrão.	155
Propriedades da sessão de Transformação de Mascaramento de Dados.	155
Regras e Diretrizes para Transformações Mascaramento de Dados.	156
Capítulo 6: Exemplos de Mascaramento de Dados.....	157
Arquivos de Pesquisa de Nome e Endereço.	157
Substituindo Dados com a Transformação Pesquisa.	157
Mascarando Dados com a Transformação Expressão.	161
Capítulo 7: Transformação Expressão.....	163
Visão geral da transformação de Expressão.	163
Componentes da Transformação Expressão.	163
Configurando portas.	164
Cálculo de Valores.	164
Criação de uma Transformação de Expressão.	165
Capítulo 8: Transformação Procedimento Externo.....	166
Visão geral da Transformação Procedimento Externo.	166

Compatibilidade de Página de Código.	167
Procedimentos Externos e Transformações de Procedimentos Externos.	167
Propriedades da Transformação Procedimento Externo.	167
Procedimentos Externos de COM Versus Informatica.	168
O Exemplo do Banksoft.	168
Configurando Propriedades da Transformação Procedimento Externo.	168
Desenvolvendo Procedimentos COM.	170
Etapas para Criar um Procedimento COM.	170
Tipo de Serviço de Procedimento Externo COM.	170
Usando o Visual C++ para Desenvolver Procedimentos COM.	171
Desenvolvendo procedimentos COM com o Visual Basic.	176
Desenvolvendo Procedimentos Externos da Informatica.	178
Etapa 1. Criar a Transformação de Procedimento Externo.	178
Etapa 2. Gerar os Arquivos C++.	180
Etapa 3. Preencher o Stub de Método com Implementação.	182
Etapa 4. Construindo o Módulo.	183
Etapa 5. Criando um Mapeamento.	184
Etapa 6. Executar a Sessão.	185
Distribuindo Procedimentos Externos.	186
Distribuindo Procedimentos COM.	186
Distribuindo Módulos Informatica.	187
Observações de Desenvolvimento.	187
Tipos de Dados COM.	187
Procedimentos no Nível de Linha.	188
Valores de Retorno de Procedimentos.	188
Exceções em Chamadas de Procedimentos.	188
Gerenciamento de Memória para Procedimentos.	189
Classes Wrapper para Funções do VB ou Bibliotecas C/C++ Pré-existentes.	189
Gerando Mensagens de Erro e Rastreamento.	189
Transformações Procedimento Externo Não Conectadas.	191
Inicializando os Módulos COM e Informatica.	191
Outros Arquivos Distribuídos e Usados em TX.	193
Variáveis do Processo de Serviço nas Propriedades de Inicialização.	193
Interfaces do Procedimento Externo.	194
Função Distribuir.	194
Função de Procedimento Externo.	195
Funções de Acesso das Propriedades.	195
Funções de Acesso a Parâmetros.	196
Funções de Acesso à Página de Código.	198
Funções de Acesso ao Nome da Transformação.	198
Funções de Acesso a Procedimentos.	199
Funções Relacionadas a Partições.	199

Função de Nível de Rastreamento.	199
Capítulo 9: Transformação Filtros.	201
Visão Geral da Transformação Filtro.	201
Componentes da Transformação Filtro.	202
Configuração das Portas da Transformação Filtro.	202
Condição de filtro.	202
Filtrando Linhas com Valores Nulos	203
Etapas para Criar uma Transformação de Filtro.	203
Dicas para Transformações de Filtro.	204
Capítulo 10: Transformação HTTP.	205
Visão Geral da Transformação de HTTP.	205
Autenticação.	206
Conexão ao Servidor HTTP.	206
Criação de uma Transformação HTTP.	206
Configuração da Guia Propriedades.	207
Configuração da guia HTTP.	208
Selecionando um Método.	208
Configurando Grupos e Portas.	208
Configuração de uma URL.	213
Exemplos.	215
Exemplo de GET.	215
Exemplo de POST.	216
Exemplo SIMPLE POST.	216
Exemplo de SIMPLE PATCH.	217
Exemplo de SIMPLE PUT.	218
Exemplo de SIMPLE DELETE.	219
Capítulo 11: Transformação de Resolução de Identidade	221
Visão Geral da Transformação de Resolução de Identidade.	221
Criar e Configurar a Transformação.	221
Conexão do Servidor de Pesquisa.	222
Configuração do Sistema e da Pesquisa.	222
Exibir Seleção.	223
Guias da Transformação de Resolução de Identidade	224
Grupos e Portas.	225
Grupos e Portas de Entrada	225
Grupos e Portas de Saída.	225
Capítulo 12: Transformação Java.	226
Visão Geral de Transformações Java.	226
Etapas para Definir uma Transformação Java.	227

Transformações Java Ativas e Passivas.	227
Conversão de Tipo de Dados.	227
Usando a Guia Código Java.	229
Configurando portas.	230
Criando grupos e portas.	230
Definindo Valores Padrão para Portas.	231
Configuração das Propriedades da Transformação Java.	232
Trabalhando com o Controle de Transação.	233
Definindo a Estratégia de Atualização.	234
Desenvolvimento do Código Java.	234
Criando trechos de código Java.	235
Importando pacotes Java.	236
Definindo Passiva.	237
Guia Na Linha de Entrada.	238
Guia No Final dos Dados.	238
Guia Sobre a Recepção de Transações.	239
Usando Código Java para Analisar um Arquivo Simples.	239
Configurando definições da transformação Java.	240
Configuração do Classpath.	240
Habilitando Alta Precisão.	242
Processando Subsegundos.	242
Compilando uma transformação Java.	242
Corrigindo erros de compilação.	243
Localizando a Origem de Erros de Compilação.	243
Identificando a origem de erros de compilação.	243

Capítulo 13: Referência da API da transformação Java..... 245

Visão geral dos métodos API da transformação Java.	245
commit.	246
defineJExpression.	247
failSession.	248
generateRow.	248
getInRowType.	249
getMetadata.	249
incrementErrorCount.	250
invokeJExpression.	251
isNull.	252
logError.	252
logInfo.	253
resetNotification.	253
rollback.	254
setNull.	255
setOutRowType.	255

storeMetadata.	256
------------------------	-----

Capítulo 14: Expressões Java..... 258

Visão Geral das Expressões Java.	258
Tipos de Função de Expressão.	259
Usando a caixa de diálogo Definir ExpressãoDefinir Função para definir uma expressão.	259
Etapa 1. Configurar a Função.	260
Etapa 2. Criar e Validar a Expressão.	260
Etapa 3. Gerar o Código Java para a Expressão.	260
Criando uma expressão e gerando código Java usando a caixa de diálogo Definir Expressão Definir Função.. . . .	260
Modelos da expressão Java.	261
Trabalhando com a Interface Simples.	261
invokeJExpression.	262
Exemplo de Interface Simples.	262
Trabalhando com a Interface Avançada.	263
Invocando uma expressão com a Interface Avançada.	263
Regras e Diretrizes para Trabalhar com a Interface Avançada.	264
Classe EDataType.	264
Classe JExprParamMetadata.	265
defineJExpression.	265
Classe JExpression.	266
Exemplo de Interface Avançada.	266
Referência de API da classe JExpression.	267
getBytes.	268
getDouble.	268
getInt.	268
getLong.	268
getResultDataType.	268
getResultMetadata.	269
getStringBuffer.	269
invoke.	269
isResultNull.	270

Capítulo 15: Exemplo de Transformação Java..... 271

Visão geral do exemplo de transformação Java.	271
Etapa 1. Importar o Mapeamento.	272
Etapa 2. Criar Transformação e Configurar Portas.	272
Etapa 3. Inserir Código Java.	273
Guia Importar Pacotes.	273
Guia Código Auxiliar.	274
Guia Na Linha de Entrada.	274
Etapa 4. Compilar o Código Java.	276

Etapa 5. Criar uma Sessão e Fluxo de Trabalho.	276
Dados de Exemplo.	276
Capítulo 16: Transformação Joiner.	278
Visão Geral da Transformação Joiner.	278
Trabalhando com a Transformação Joiner.	279
Propriedades da Transformação Joiner.	279
Definição de uma Condição de Associação.	281
Definindo o Tipo de Associação.	281
Associação Normal.	282
Associação Externa Mestre.	283
Associação Externa Detalhada.	283
Associação Externa Completa.	283
Usando Entrada Classificada.	284
Configuração da Ordem de Classificação.	285
Adição de transformações ao mapeamento.	285
Configurando a Transformação Joiner.	286
Definindo a Condição de Associação.	286
Unindo dados a partir de uma única origem.	287
Unindo duas ramificações do mesmo pipeline.	287
Unindo duas instâncias da mesma origem.	288
Diretrizes para unir dados a partir de uma única origem.	289
Bloqueio dos pipelines de origem.	289
Transformação Joiner Não Classificada.	289
Transformação Joiner Classificada.	289
Trabalhando com Transações.	290
Preservando os Limites de Transação de um Único Pipeline.	291
Preservando Limites da Transação no Pipeline de Detalhes.	291
Descarte dos Limites de Transação para Dois Pipelines.	291
Criando uma transformação Joiner.	292
Dicas para Transformações Joiner.	292
Capítulo 17: Transformação Pesquisa.	294
Visão Geral de Transformação Pesquisa.	294
Tipos de Origem da Pesquisa.	295
Pesquisas Relacionais.	295
Pesquisas de arquivos simples.	296
Pesquisas no Pipeline.	297
Pesquisas Conectadas e Não Conectadas.	298
Pesquisas Conectadas.	299
Pesquisas Não Conectadas.	300
Componentes da Pesquisa.	301
Origem da Pesquisa.	301

Portas de Pesquisa.	301
Propriedades de pesquisa.	302
Condição de pesquisa.	302
Propriedades de pesquisa.	303
Configurando Propriedades de Pesquisa em uma Sessão.	308
Consulta de Pesquisa.	310
Consulta de Pesquisa Padrão.	310
Substituindo a Consulta de Pesquisa.	310
Substituição SQL para Pesquisas sem Cache.	312
Filtro da Origem de Pesquisa.	313
Condição de pesquisa.	314
Cache Estático ou Sem Cache.	315
Cache Dinâmico.	315
Manuseando Várias Correspondências.	315
Caches de Pesquisa.	316
Retornar Várias Linhas.	317
Regras e Diretrizes para Retornar Várias Linhas.	317
Configurando Transformações de Pesquisa Não Conectadas.	318
Etapa 1. Adicionar Portas de Entrada.	318
Etapa 2. Adicionar a Condição de Pesquisa.	318
Etapa 3. Designar um valor de retorno.	319
Etapa 4. Chamar a Pesquisa Através de uma Expressão.	319
Resiliência de Deadlock de Banco de Dados.	320
Criando uma transformação Pesquisa.	321
Criando uma Transformação Pesquisa de Pipeline Reutilizável.	322
Criando uma Transformação Pesquisa de Pipeline Não Reutilizável.	322
Dicas para Transformações de Pesquisa.	322
Capítulo 18: Caches de Pesquisa.	324
Visão geral de Caches de Pesquisa.	324
Comparação de Cache.	326
Construindo Caches de Pesquisas Conectadas.	326
Caches Sequenciais.	327
Caches Simultâneos.	327
Usando um Cache de Pesquisa Persistente.	328
Usando um Cache Não Persistente.	328
Usando um Cache Persistente.	328
Recriando o Cache de Pesquisa.	329
Trabalhando com pesquisa sem cache ou com cache estático.	330
Compartilhando o Cache de Pesquisa.	330
Compartilhando um Cache de Pesquisa Sem Nome.	330
Compartilhando um Cache de Pesquisa Nomeado.	333
Dicas para Caches de Pesquisa.	337

Capítulo 19: Cache de Pesquisa Dinâmica.....	338
Visão geral do Cache de Pesquisa Dinâmica.	338
Propriedades da Pesquisa Dinâmica.	339
NewLookupRows.	340
Expressão associada.	340
Valores nulos.	341
Ignorar Portas em Comparação.	342
Substituição SQL.	343
Valores de Transformação Pesquisa.	343
Valores de Cache Iniciais.	344
Valores de Entrada.	344
Valores de Pesquisa.	345
Valores de Saída.	345
Atualizações do Cache de Pesquisa Dinâmica.	346
Inserir ou Atualizar.	347
Atualizar ou Inserir.	347
Mapeamentos com pesquisas dinâmicas.	348
Configurando a Transformação de Estratégia de Atualização Upstream.	348
Configurando transformações Downstream.	349
Configurando sessões com um cache de pesquisa dinâmica.	350
Atualizações de Cache Dinâmico Condicional.	351
Processamento de Sessão.	351
Configuração de Pesquisa de Cache Dinâmico Condicional.	351
Atualização de cache dinâmico com resultados de expressões.	352
Valores nulos da expressão.	352
Processamento de Sessão.	352
Configurando uma Expressão para Atualizações de Cache Dinâmico.	353
Sincronizando o Cache com a Origem da Pesquisa.	353
NewLookupRow.	354
Configuração da sincronização de cache dinâmico.	354
Exemplo de Cache de Pesquisa Dinâmica.	354
Regras e Diretrizes para Cache de Pesquisa Dinâmica.	355
 Capítulo 20: Transformação Normalizador.....	 357
Visão Geral de Transformação Normalizador.	357
Componentes da Transformação Normalizador.	358
Guia Portas.	358
Guia Propriedades.	359
Guia Normalizador.	360
Chaves geradas pela transformação Normalizador.	361
Armazenando Valores-chave Gerados.	361
Alterando os Valores-chave Gerados.	362

Transformação de Normalizador VSAM.	362
Guia Portas Normalizadoras VSAM.	364
Guia Normalizador VSAM.	365
Etapas para Criar uma Transformação Normalizador de VSAM.	366
Transformação de Normalizador de Pipeline.	366
Guia Portas do Normalizador do Pipeline	368
Guia Normalizador de Pipeline.	368
Etapas para Criar uma Transformação de Normalizador de Pipeline.	369
Usando uma Transformação de Normalizador em um Mapeamento.	370
Gerando valores-chave.	372
Solução de Problemas de Transformações Normalizador.	374
 Capítulo 21: Transformação Classificação.....	375
Visão Geral da Transformação Classificação.	375
Classificando Valores de String.	376
Caches de Classificação.	376
Propriedades da Transformação Classificação.	376
Portas em uma Transformação Classificação.	377
Índice de Classificação.	377
Definição de Grupos.	378
Criando uma transformação Classificação.	378
 Capítulo 22: Transformação Roteador.....	381
Visão Geral da Transformação Roteador.	381
Trabalhando com Grupos.	383
Grupo de Entrada.	383
Grupos de Saída.	383
Usando Condições do Filtro de Grupo.	383
Adição de Grupos.	385
Trabalhando com Portas.	385
Conexão das Transformações Roteador em um Mapeamento.	386
Criando uma Transformação Roteador.	386
 Capítulo 23: Transformação de Gerador de Sequência.....	387
Visão Geral da Transformação de Gerador de Sequência.	387
Portas do Gerador de Sequência.	388
Portas de Passagem.	388
Porta NEXTVAL.	388
CURRVAL.	392
Propriedades da Transformação Gerador de Sequência.	394
Valor Inicial.	396
Incrementar em.	396
Valor final.	396

Valor de Incremento.	396
Percorrer um Intervalo de Valores.	397
Valor Atual.	397
Número de Valores Guardados em Cache.	398
Geradores de Sequência Não-Reutilizáveis.	398
Geradores de Sequência Reutilizáveis.	399
Redefinir.	399
Manter Ordem das Linhas.	400
Objeto de Dados de Sequência.	400
Criando um objeto de dados de sequência.	400
Criando uma Transformação de Gerador de Sequência.	402
Criando uma Transformação de Gerador de Sequência.	403
Perguntas frequentes.	404
Transformação de Gerador de Sequência em um ambiente não nativo.	405
Transformação de Gerador de Sequência no mecanismo Blaze.	405
Transformação de Gerador de Sequência no mecanismo Spark.	405
Capítulo 24: Transformação Classificador.	406
Visão Geral da Transformação Classificador.	406
Classificando Dados.	406
Propriedades da Transformação de Classificador.	408
Tamanho do Cache do Classificador.	408
Distinção Entre Maiúsculas e Minúsculas.	408
Diretório de Trabalho.	408
Linhas de Saída Distintas.	409
Nível de rastreamento.	409
Nulo Tratado como Baixo.	409
Escopo de Transformação.	409
Criando uma Transformação Classificador.	410
Capítulo 25: Transformação de Qualificador de Origem.	411
Visão Geral da Transformação de Qualificador de Origem.	411
Tipos de Dados da Transformação.	412
Ordem de Carregamento de Destino.	412
Valores de data e hora.	412
Parâmetros e Variáveis.	413
Propriedades da Transformação de Qualificador de Origem.	414
Consulta Padrão.	415
Exibindo a Consulta Padrão.	415
Substituindo a Consulta Padrão.	416
Unindo Dados de Origem.	416
Associação Padrão.	416
Associações Personalizadas.	417

Associações Heterogêneas.	417
Criando Relações de Chaves.	418
Adicionando uma Consulta SQL.	418
Inserindo uma Associação Definida pelo Usuário.	420
Suporte à Associação Externa.	420
Sintaxe de Associação Informatica.	421
Criando uma Associação Externa.	426
Restrições Comuns de Sintaxe de Banco de Dados.	427
Inserindo um Filtro de Origem.	428
Usando Portas Classificadas.	429
Selecionar Distintos.	430
Substituindo Selecionar Distintos na Sessão.	430
Adicionando Comandos SQL Pré e Pós-Sessão.	431
Criando uma Transformação Qualificador de Origem.	431
Criando uma Transformação Qualificador de Origem Manualmente.	431
Configurando Opções da Transformação Qualificador de Origem.	432
Solucionando Problemas de Transformações de Qualificador de Origem.	432
Capítulo 26: Transformação SQL.....	434
Visão Geral da Transformação SQL.	434
Modo de Script.	435
Exemplo.	435
Regras e Diretrizes do Modo de Script.	436
Modo de Consulta.	436
Usando Consultas SQL Estáticas.	437
Utilizando Consultas SQL Dinâmicas.	438
Configuração de Portas de Passagem.	440
Configuração do Modo Passivo.	441
Regras e Diretrizes do Modo de Consulta.	441
Conectando com Bancos de Dados.	442
Usando uma Conexão de Banco de Dados Estática.	442
Passando uma Conexão Lógica de Banco de Dados	442
Passando Informações de Conexão Completa.	443
Regras e Diretrizes para Conexões com Bancos de Dados.	445
Processamento de Sessão.	446
Controle de transação.	446
Alta disponibilidade.	447
Log de Consultas SQL.	449
Cardinalidade de linha de entrada para linha de saída.	449
Processamento de Instruções de Consulta.	449
Número de Linhas Afetadas.	450
Contagem Máxima das Linhas de Saída.	451
Compreendendo Linhas de Erro.	451

Continuando no Erro de SQL	453
Propriedades da Transformação SQL.	453
Guia Propriedades.	454
Guia Configurações SQL.	455
Guia Portas SQL.	456
Instruções SQL.	457
Criação de uma Transformação SQL.	458
Capítulo 27: Usando a Transformação SQL em um Mapeamento.....	460
Visão Geral do Exemplo da Transformação SQL.	460
Exemplo de atualização dinâmica.	460
Definição do Arquivo de Origem.	461
Criação de uma Definição de Destino.	462
Criação da Tabela de Banco de Dados.	462
Configuração da Transformação Expressão.	463
Definindo a Transformação SQL.	463
Configurando os Atributos da Sessão.	465
Resultados de Dados de Destino.	465
Exemplo de conexão dinâmica.	465
Definição do Arquivo de Origem.	466
Criação de uma Definição de Destino.	467
Criação de tabelas de banco de dados.	467
Criação de Conexões ao Banco de Dados.	467
Configuração da Transformação Expressão	467
Definindo a Transformação SQL.	468
Configurando os Atributos da Sessão.	469
Resultados de Dados de Destino.	470
Capítulo 28: Transformação Procedimento Armazenado.....	471
Visão Geral da Transformação Procedimento Armazenado.	471
Dados de Entrada e Saída.	472
Conectada ou desconectada.	473
Especificando Quando o Procedimento Armazenado é Executado.	474
Usando um Procedimento Armazenado em um Mapeamento.	475
Gravando um Procedimento Armazenado.	476
Exemplo de Procedimento Armazenado.	476
Criação de uma Transformação Procedimento Armazenado.	478
Importando procedimentos armazenados.	479
Criando manualmente transformações de Procedimento Armazenado.	480
Definindo Opções Para o Procedimento Armazenado.	481
Alterando o Procedimento Armazenado.	482
Configurando uma transformação conectada.	482
Configurando uma transformação desconectada.	483

Chamando um Procedimento Armazenado de Uma Expressão.	483
Chamada de um Procedimento Armazenado Pré ou Pós-sessão.	486
Tratamento de erros.	487
Erros Pré-sessão.	487
Erros Pós-sessão.	488
Erros de Sessão.	488
Bancos de Dados Suportados.	488
Declaração SQL.	488
Tipos de Parâmetro.	488
Porta de entrada/saída no mapeamento.	489
Tipo de Valor de Retorno Suportado.	489
Regras de Expressão.	489
Dicas para Transformações Procedimento Armazenado.	490
Solução de Problemas de Transformações Procedimento Armazenado.	490
Capítulo 29: Transformação Controle de Transações.	492
Visão Geral da Transformação Controle de Transação.	492
Propriedades da Transformação Controle de Transação.	493
Guia Propriedades.	493
Exemplo.	494
Usando a Transformação Controle de Transação em Mapeamentos.	495
Mapeamentos Controle de Transação de Exemplo com Vários Destinos.	496
Mapeando Orientações e Validação.	497
Criando uma Transformação Controle de Transação.	497
Capítulo 30: Transformação União.	499
Visão Geral da Transformação de União.	499
Regras e Diretrizes para Transformações de União.	499
Componentes da Transformação de União.	500
Trabalhando com Grupos e Portas.	500
Criação de uma Transformação União.	500
Usando uma Transformação de União em um Mapeamento.	501
Capítulo 31: Transformação Dados Não Estruturados.	502
Visão Geral do Data Transformation não estruturado.	502
Configuração da opção de dados não estruturados.	503
Configuração do diretório do repositório do Data Transformation.	504
Tipos de Serviço de Data Transformation.	504
Componentes da Transformação de Dados Não Estruturados.	505
Guia Propriedades.	505
Guia Configurações de UDT.	506
Exibindo Mensagens de Rastreamento de Status.	507
Portas da Transformação de Dados Não Estruturados.	507

Tipos de entrada e saída.	508
Portas adicionais da transformação de dados não estruturados.	509
Criando Portas a partir de um serviço de Data Transformation.	509
Nomes de Serviço da Transformação de Dados não Estruturados.	510
Hierarquias Relacionais.	510
Exportando o Esquema de Hierarquia.	511
Mapeamentos.	511
Analisando Documentos de Word para Tabelas Relacionais.	512
Criando uma Planilha do Excel a Partir de um XML.	512
Saída de Arquivo XML Dividido.	513
Regras e Diretrizes para Mapeamentos de Dados Não Estruturados.	513
Criar uma transformação de dados não estruturados.	514
Capítulo 32: Transformação Estratégia de Atualização.....	515
Visão Geral da Transformação de Estratégia de Atualização.	515
Definindo a Estratégia de Atualização.	516
Sinalizando Linhas Dentro de um Mapeamento.	516
Encaminhamento de Linhas Rejeitadas.	516
Expressões de Estratégia de Atualização.	517
Transformações de Agregador e de Estratégia de Atualização.	517
Transformações Pesquisa e Estratégia de Atualização.	518
Definindo a Estratégia de Atualização para uma Sessão.	518
Especificando uma Operação para Todas as Linhas.	518
Especificando Operações para Tabelas de Destino Individuais.	519
Lista de Verificação da Estratégia de Atualização.	520
Capítulo 33: Transformações XML.....	521
Transformação de Qualificador de Origem XML.	521
Transformação de Analisador de XML.	521
Transformação Gerador de XML.	522
Índice.....	523

Prefácio

Consulte o *Guia de Transformação do PowerCenter®* para saber mais sobre a configuração, as diretrizes, o uso e o comportamento em tempo de execução das transformações da Informatica. Transformações são objetos de repositório que representam as operações que o Serviço de Integração executa nos dados.

Consulte o guia para realizar operações de dados para gerar, alterar ou transferir dados. Visualize o suporte para cada transformação, dependendo de onde e como você planeja executar o mapeamento. Os dados passam pelas portas de transformação para destinos e outras transformações vinculadas em um mapeamento ou mapplet. Você pode inserir expressões usando o Editor de Expressão nas transformações ativas. Uma transformação pode ter vários grupos de entrada e saída, um conjunto de portas que definem uma linha de dados recebidos ou enviados.

Recursos da Informatica

A Informatica oferece uma variedade de recursos de produtos através da Rede da Informatica e outros portais on-line. Use os recursos para obter o máximo de seus produtos e soluções da Informatica e para aprender com outros usuários da Informatica e especialistas no assunto.

Rede da Informatica

A Rede da Informatica é a porta de entrada para muitos recursos, incluindo a Base de Dados de Conhecimento da Informatica e o Suporte Global a Clientes da Informatica. Para acessar a Rede da Informatica, visite <https://network.informatica.com>.

Como membro da Rede da Informatica, você tem as seguintes opções:

- Pesquisar por recursos do produto na Base de Dados de Conhecimento.
- Visualizar informações sobre disponibilidade de produtos.
- Criar e revisar seus casos de suporte.
- Encontrar a sua Rede de Grupo de Usuários da Informatica local e colaborar com seus colegas.

Base de Dados de Conhecimento da Informatica

Use a Base de Dados de Conhecimento da Informatica para encontrar recursos de produtos, como artigos de instruções, práticas recomendadas, tutoriais em vídeo e respostas a perguntas frequentes.

Para pesquisar na Base de Dados de Conhecimento, visite <https://search.informatica.com>. Em caso de dúvidas, comentários ou ideias sobre a Base de Dados de Conhecimento, entre em contato com a equipe da Base de Dados de Conhecimento da Informatica em KB_Feedback@informatica.com.

Documentação da Informatica

Use o Portal de Documentação da Informatica para explorar uma extensa biblioteca de documentação para versões de produtos atuais e recentes. Para explorar o Portal de Documentação, visite <https://docs.informatica.com>.

Em caso de dúvidas, comentários ou ideias sobre a documentação do produto, entre em contato com a equipe da Documentação da Informatica em infa_documentation@informatica.com.

Matrizes de Disponibilidade de Produto da Informatica

As Matrizes de Disponibilidade de Produto (PAMs) indicam as versões dos sistemas operacionais, os bancos de dados e tipos de fontes e destinos de dados com os quais uma versão de produto é compatível. Veja as PAMs da Informatica em <https://network.informatica.com/community/informatica-network/product-availability-matrices>.

Informatica Velocity

O Informatica Velocity é uma coleção de dicas e práticas recomendadas desenvolvidas pelos Serviços Profissionais da Informatica e baseada em experiências reais de centenas de projetos de gerenciamento de dados. O Informatica Velocity representa o conhecimento coletivo dos consultores da Informatica que trabalham com organizações em todo o mundo para planejar, desenvolver, implantar e manter soluções de gerenciamento de dados bem-sucedidas.

Encontre os recursos do Informatica Velocity em <http://velocity.informatica.com>. Se você tiver dúvidas, comentários ou ideias sobre o Informatica Velocity, entre em contato com os Serviços Profissionais da Informatica em ips@informatica.com.

Informatica Marketplace

O Informatica Marketplace é um fórum onde você pode encontrar soluções que ampliam e aprimoram suas implementações da Informatica. Aproveite as centenas de soluções dos desenvolvedores e parceiros da Informatica no Marketplace para melhorar sua produtividade e agilizar o tempo de implementação em seus projetos. Encontre o Informatica Marketplace em <https://marketplace.informatica.com>.

Suporte Global a Clientes da Informatica

Você pode entrar em contato com um Centro de Suporte Global por telefone ou por meio da Rede da Informatica.

Para descobrir o número de telefone local do Suporte Global a Clientes da Informatica, visite o site da Informatica no seguinte link: <https://www.informatica.com/services-and-training/customer-success-services/contact-us.html>.

Para encontrar recursos de suporte on-line na Rede da Informatica, visite <https://network.informatica.com> e selecione a opção eSupport.

CAPÍTULO 1

Trabalhando com Transformações

Este capítulo inclui os seguintes tópicos:

- [Visão geral de transformações, 24](#)
- [Criando uma transformação, 28](#)
- [Configurando transformações, 29](#)
- [Portas de Transformação, 29](#)
- [Transformações de vários grupos, 31](#)
- [Trabalhando com expressões, 31](#)
- [Variáveis Locais, 36](#)
- [Valores Padrão para Portas, 39](#)
- [Configurando o nível de rastreamento em transformações, 47](#)
- [Transformações Reutilizáveis, 47](#)

Visão geral de transformações

Uma transformação é um objeto de repositório que gera, modifica ou passa dados. O Designer proporciona um conjunto de transformações que realizam funções específicas. Por exemplo, uma transformação Agregador realiza cálculos em grupos de dados.

As transformações em um mapeamento representam as operações feitas pelo Serviço de Integração nos dados. Os dados passam através de portas de transformação vinculadas a um mapeamento ou mapplet.

As transformações podem ser dos seguintes tipos:

- Ativa ou Passiva
- Conectada ou Não Conectada
- Nativa ou Não Nativa

Transformações ativas

Uma transformação ativa é capaz de efetuar quaisquer das ações a seguir:

- **Alterar o número de linhas que têm passagem pela transformação.** Por exemplo, a transformação de Filtro é ativa porque remove linhas que não atendem à condição de filtro. Todas as transformações de vários grupos estão ativas porque podem mudar o número de linhas que passam pela transformação.
- **Altere o limite de transação.** Por exemplo, a Transformação Controle de Transação está ativa porque define uma transação de confirmação ou reversão com base em uma expressão avaliada para cada linha.
- **Alterar o tipo de linha.** Por exemplo, a transformação Estratégia de Atualização está ativa porque sinaliza linhas para inserção, exclusão, atualização ou rejeição.

O Designer não permite que se conecte várias transformações ativas, ou uma transformação ativa e outra passiva, à mesma transformação de fluxo para baixo ou transformação em grupos, pois o Serviço de Integração pode não conseguir concatenar as linhas transferidas pelas transformações ativas. Por exemplo, uma ramificação do mapeamento contém uma transformação de Estratégia de Atualização que sinaliza uma linha para que seja excluída. Outra ramificação contém uma transformação de Estratégia de Atualização que sinaliza uma linha para inserção. Se você conectar essas transformações a um único grupo de entrada de transformação, o Serviço de Integração não será capaz de combinar as operações de exclusão e inserção da linha.

A transformação do Gerador de Sequência é uma exceção à regra. O Designer irá permitir que você conecte uma transformação de Gerador de Sequência e uma transformação ativa à mesma transformação de fluxo para baixo ou grupo de entrada de transformação. Uma transformação de Gerador de Sequência não recebe dados. Ela gera valores numéricos exclusivos. Consequentemente, o Serviço de Integração não encontra problemas ao concatenar linhas transferidas por uma transformação de Gerador de Sequência com uma transformação ativa.

Transformações passivas

Uma transformação passiva não altera o número de linhas que passam pela transformação e mantém o limite de transação e o tipo de linha.

Você pode conectar várias transformações à mesma transformação downstream ou ao mesmo grupo de entrada de transformação quando todas as transformações nas ramificações upstream são passivas. A transformação que origina a ramificação pode ser ativa ou passiva.

Transformações Desconectadas

As transformações podem ser conectadas ou desconectadas do fluxo de dados. Uma transformação desconectada não está conectada a outras transformações no mapeamento. Uma transformação desconectada é chamada de outra transformação e retorna um valor para essa transformação.

Transformações Nativas e Não Nativas

As transformações nativas são um conjunto de transformações que o Designer oferece. As transformações não nativas são aquelas que você cria usando a transformação Personalizada. O Designer também oferece algumas transformações não nativas, como de Java, SQL e União. As regras que se aplicam às transformações Personalizadas também se aplicam às transformações não nativas que são criadas usando a transformação Personalizada.

Descrições de Transformações

A tabela a seguir fornece uma rápida descrição de cada transformação:

Transformação	Tipo	Descrição
Agregador	<ul style="list-style-type: none">- Ativa- Conectada- Nativa	Executa cálculos de agregação.
Qualificador de Origem de Aplicativo	<ul style="list-style-type: none">- Ativa- Conectada- Não Nativa	Representa as linhas que o Serviço de Integração lê em um aplicativo, como uma origem ERP, quando ele executa uma sessão.
Personalizada	<ul style="list-style-type: none">- Ativa ou Passiva- Conectada- Não Nativa	Chama um procedimento em uma biblioteca compartilhada ou DLL.
Mascaramento de Dados	<ul style="list-style-type: none">- Passiva- Conectada- Não Nativa	Substitui dados de produção confidenciais por dados de teste realísticos para ambientes em que não há produção.
Expressão	<ul style="list-style-type: none">- Passiva- Conectada- Nativa	Calcula um valor.
Procedimento Externo	<ul style="list-style-type: none">- Passiva- Conectada ou Não Conectada- Nativa	Chama um procedimento em uma biblioteca compartilhada ou na camada COM do Windows.
Filtro	<ul style="list-style-type: none">- Ativa- Conectada- Nativa	Filtra dados.
HTTP	<ul style="list-style-type: none">- Passiva- Conectada- Não Nativa	Conecta-se a um servidor HTTP para ler ou atualizar dados.
Entrada	<ul style="list-style-type: none">- Passiva- Conectada- Nativa	Define linhas de entrada de mapplet. Disponível no Mapplet Designer.
Java	<ul style="list-style-type: none">- Ativa ou Passiva- Conectada- Não Nativa	Executa uma lógica de usuário codificada em Java. O código de bytes da lógica do usuário está armazenado no repositório.
Associador	<ul style="list-style-type: none">- Ativa- Conectada- Nativa	Reúne dados de diferentes bancos de dados ou de sistemas de arquivos simples.

Transformação	Tipo	Descrição
Pesquisa	<ul style="list-style-type: none"> - Ativa ou Passiva - Conectada ou Não Conectada - Nativa 	Pesquisa e retorna dados de um arquivo simples, uma tabela relacional, uma exibição ou um sinônimo.
Normalizador	<ul style="list-style-type: none"> - Ativa - Conectada - Nativa 	Qualificador de origem para origens de COBOL. Também pode ser usado no pipeline para normalizar dados de origens relacionais ou de arquivo simples.
Saída	<ul style="list-style-type: none"> - Passiva - Conectada - Nativa 	Define linhas de saída de mapplet. Disponível no Mapplet Designer.
Classificação	<ul style="list-style-type: none"> - Ativa - Conectada - Nativa 	Limita registros a uma faixa superior ou inferior.
Roteador	<ul style="list-style-type: none"> - Ativa - Conectada - Nativa 	Roteia dados em várias transformações com base nas condições do grupo.
Gerador de Sequência	<ul style="list-style-type: none"> - Passiva - Conectada - Nativa 	Gera chaves primárias.
Classificador	<ul style="list-style-type: none"> - Ativa - Conectada - Nativa 	Classifica dados com base em uma chave de classificação.
Qualificador de Origem	<ul style="list-style-type: none"> - Ativa - Conectada - Nativa 	Representa as linhas que o Serviço de Integração lê em uma origem de arquivo simples ou relacional quando ele executa uma sessão.
SQL	<ul style="list-style-type: none"> - Ativa ou Passiva - Conectada - Não Nativa 	Executa consultas SQL em um banco de dados.
Procedimento Armazenado	<ul style="list-style-type: none"> - Passiva - Conectada ou Não Conectada - Nativa 	Chama um procedimento armazenado.
Controle de Transação	<ul style="list-style-type: none"> - Ativa - Conectada - Nativa 	Define transações de confirmação e reversão.
União	<ul style="list-style-type: none"> - Ativa - Conectada - Não Nativa 	Mescla dados de diferentes bancos de dados ou sistemas de arquivos simples.

Transformação	Tipo	Descrição
Dados Não Estruturados	- Ativa ou Passiva - Conectada - Não Nativa	Transforma dados em formatos não estruturados ou semiestruturados.
Estratégia de Atualização	- Ativa - Conectada - Nativa	Determina se é para inserir, excluir, atualizar ou rejeitar linhas.
Gerador de XML	- Ativa - Conectada - Nativa	Lê dados de uma ou mais portas de entrada e gera o XML por meio de uma única porta de saída.
Analisador de XML	- Ativa - Conectada - Nativa	Lê XML de uma porta de entrada e gera dados para uma ou mais portas de saída.
Qualificador de Origem XML	- Ativa - Conectada - Nativa	Representa as linhas que o Serviço de Integração lê em uma origem XML ao executar uma sessão.

Quando constrói um mapeamento, você adiciona transformações e configura-as para tratar dados de acordo com uma finalidade comercial. Execute as tarefas a seguir para incorporar uma transformação a um mapeamento:

1. **Crie a transformação.** Crie-a no Mapping Designer como parte de um mapeamento, no Mapplet Designer como parte de um mapplet ou no Transformation Developer como uma transformação reutilizável.
2. **Configure a transformação.** Cada tipo de transformação tem um conjunto exclusivo de opções que você pode configurar.
3. **Vincule a transformação a outras transformações e a definições de destino.** Arraste uma porta até outra para vinculá-las no mapeamento ou no mapplet.

Criando uma transformação

É possível criar transformações usando as seguintes ferramentas do Designer:

- **Mapping Designer.** Crie transformações que conectam origens a destinos. Transformações em um mapeamento não podem ser usadas em outros mapeamentos, a menos que sejam configuradas para serem reutilizáveis.
- **Transformation Developer.** Crie transformações individuais, chamadas de transformações reutilizáveis, que são usadas em mapeamentos múltiplos.
- **Mapplet Designer.** Crie e configure um conjunto de transformações, chamadas de mapplets, que são usadas em mapeamentos múltiplos.

Use o mesmo processo para criar uma transformação no Mapping Designer, Transformation Developer e Mapplet Designer.

Para criar uma transformação:

1. Abra a ferramenta apropriada do Designer.

2. No Mapping Designer, abra ou crie um Mapeamento. No Mapplet Designer, abra ou crie um Mapplet.
3. Clique em Transformação > Criar e selecione o tipo de transformação que deseja criar.
4. Arraste pela porção do mapeamento na qual deseja colocar a transformação.

A nova transformação aparece no espaço de trabalho. Em seguida, é necessário configurar a transformação adicionando quaisquer novas portas a ela e definindo outras propriedades.

Configurando transformações

Depois de criar uma transformação, é possível configurá-la. Todas as transformações contêm as seguintes guias comuns:

- **Transformação.** Nomeie a transformação ou adicione uma descrição.
- **Porta.** Adicionar e configurar portas.
- **Propriedades.** Configure propriedades que sejam exclusivas à transformação.

Algumas transformações podem apresentar outras guias, como a guia Condição, na qual se inserem condições em uma transformação Joiner ou Normalizador.

Ao configurar as transformações, é possível concluir as seguintes tarefas:

- **Adicionar portas.** Definir as colunas de dados que se movem para dentro e para fora da transformação.
- **Adicionar grupos.** Em determinadas transformações, defina grupos de entrada ou de saída que definem uma linha de dados que entram ou saem da transformação.
- **Inserir expressões.** Insira expressões semelhantes a SQL em determinadas transformações que transformam dados.
- **Definir variáveis locais.** Defina variáveis locais em determinadas transformações que armazenem dados temporariamente.
- **Substituir valores padrão.** Configure os valores padrão para tratar de erros de valores nulos de entrada e de transformação de saída.
- **Insira níveis de rastreamento.** Escolha a quantidade de detalhes que o Serviço de Integração grava no log de sessão acerca de uma transformação.

Renomeando Transformações

Para renomear transformações, clique no botão Renomear e insira um nome descritivo para a transformação. Em seguida, clique em OK.

Portas de Transformação

Depois de criar uma transformação, defina as portas. Crie as portas e defina suas propriedades.

Ao criar algumas transformações, você não precisa criar todas as portas manualmente. Por exemplo, você pode criar uma transformação de Pesquisa e fazer referência a uma tabela de pesquisa. Se você exibir as portas de transformação, poderá ver que a transformação tem uma porta de saída para cada coluna na tabela que você referenciou. Você não precisa definir essas portas.

Criar Portas

Ao criar algumas transformações, você não precisa criar todas as portas manualmente. Por exemplo, você pode criar uma transformação de Pesquisa e fazer referência a uma tabela de pesquisa. Se você exibir as portas de transformação, poderá ver que a transformação tem uma porta de saída para cada coluna na tabela que você referenciou. Você não precisa definir essas portas.

Crie uma porta das maneiras a seguir:

- **Arraste uma porta de outra transformação.** Quando você arrasta uma porta de outra transformação, o Designer criará uma porta com as mesmas propriedades e vincula as duas portas. Clique em Layout > Copiar Colunas para ativar a cópia das portas.
- **Clique no botão Adicionar na guia Portas.** O Designer criará uma porta vazia que você pode configurar.

Configurar Portas

Ao definir as portas de transformação, você define as propriedades de porta. As propriedades de porta incluem o nome da porta, o tipo de dados, o tipo de porta e o valor padrão.

Configure as seguintes propriedades de porta:

- **Nome da porta.** O nome da porta. Use as seguintes convenções ao nomear portas:
 - Comece com uma letra de bytes simples ou duplos, ou sublinhado de bytes duplos (_).
 - Os nomes de porta podem conter um dos seguintes caracteres de bytes simples ou duplos: uma letra, um número, o sublinhado (_), \$, #, ou @.
- **Tipo de dados, precisão e escala.** Se você planejar inserir uma expressão ou uma condição, verifique se o tipo de dados corresponde ao valor de retorno da expressão.
- **Tipo de porta.** As transformações podem conter uma combinação de tipos de porta de entrada, de saída, de entrada/saída e variável.
- **Valor padrão.** Atribua um valor padrão a uma porta que contenha valores nulos ou um erro de transformação de saída. Você pode substituir o valor padrão em algumas portas.
- **Descrição.** Uma descrição da porta.
- **Outras propriedades.** Algumas transformações têm propriedades específicas, como expressões ou propriedades de agrupamento.

Vinculando portas

Depois de adicionar e configurar uma transformação em um mapeamento, você a vincula aos destinos e outras transformações. Você vincula os objetos de mapeamento através das portas. Os dados são transferidos para dentro/fora de um mapeamento através das portas a seguir:

- **Portas de entrada.** Recebem dados.
- **Portas de saída.** Transferem dados.
- **Portas de entrada/saída.** Recebem dados e os transferem inalterados.

Para vincular portas, arraste entre as portas nos diferentes objetos de mapeamento. O Designer valida o link e o cria somente quando ele cumpre os requisitos de validação.

Transformações de vários grupos

Uma transformação pode ter vários grupos de entrada e de saída. Um grupo é um conjunto de portas que define uma linha de dados de entrada ou de saída.

Um grupo é análogo a uma tabela em uma definição de origem ou de destino relacional. A maioria das transformações tem um grupo de entrada e um de saída. No entanto, algumas têm vários grupos de entrada, de saída ou ambos. Um grupo é a representação de uma linha de dados que entram ou saem de uma transformação.

Todas as transformações de vários grupos são transformações ativas. Você não pode conectar várias transformações ativas ou uma transformação ativa e uma passiva à mesma transformação downstream ou ao mesmo grupo de entrada da transformação.

Algumas transformações de vários grupos de entrada exigem que o Serviço de Integração bloqueie os dados em um grupo de entrada enquanto espera por uma linha de um grupo de entrada diferente. Uma transformação de bloqueio é uma transformação de vários grupos de entrada que bloqueia dados de entrada. As seguintes transformações são transformações de bloqueio:

- Transformação Personalizar com a propriedade Entradas Podem Bloquear ativada
- Transformação Separador configurada para a entrada não classificada

Quando você salva ou valida um mapeamento, alguns dos que contêm transformações ativas ou de bloqueio podem não ser válidos.

Trabalhando com expressões

Você pode inserir expressões usando o Editor de Expressão em algumas transformações. Crie expressões com as seguintes funções:

- **Funções de linguagem de transformação.** Funções do tipo SQL criadas para manipular expressões comuns.
- **Funções definidas pelo usuário.** Funções criadas por você no PowerCenter com base em funções de linguagem de transformação.
- **Funções personalizadas.** Funções criadas por você com a API de Função Personalizada.

Insira uma expressão em uma porta que use o valor dos dados de uma porta de entrada ou de entrada e saída. Por exemplo, você tem uma transformação com uma porta de entrada IN_SALARY que contém os salários de todos os funcionários. Você pode usar mais tarde os valores da coluna IN_SALARY no mapeamento e o total e a média de salários calculados através dessa transformação. Por isso, o Designer exige que você crie uma porta de saída separada para cada valor calculado.

A tabela a seguir lista as transformações nas quais você pode inserir expressões:

Transformação	Expressão	Valor de Retorno
Agregador	Realiza um cálculo agregado com base em todos os dados que passam pela transformação. Você também pode especificar um filtro para registros no cálculo agregado para excluir certos tipos de registro. Por exemplo, você pode encontrar o número total e o salário médio de todos os funcionários de uma filial usando essa transformação.	Resultado de um cálculo agregado para uma porta.
Mascaramento de dados	Realiza um cálculo com base no valor das portas de entrada ou saída para uma linha. Uma expressão é um método para mascarar dados de produção na transformação de Mascaramento de Dados.	Resultado de um cálculo no nível de linhas usando portas de entrada ou saída.
Expressão	Realiza um cálculo com base em valores de uma única linha. Por exemplo, com base no preço e quantidade de determinado item, você pode calcular o preço total da compra do item em um pedido.	Resultado de um cálculo no nível de linha para uma porta.
Filtro	Especifica uma condição usada para filtrar linhas que passam por essa transformação. Por exemplo, se quiser gravar dados de clientes com saldo pendente na tabela BAD_DEBT, você poderá usar a transformação Filtro para filtrar dados do cliente.	TRUE ou FALSE, dependendo de se a linha atende à condição especificada. Somente linhas que retornam TRUE passam por essa transformação. A transformação aplica esse valor a cada linha que passa por ela.
Classificação	Define condições para as linhas incluídas em uma classificação. Por exemplo, você pode classificar os 10 maiores vendedores que trabalham com a empresa.	Resultado de um cálculo ou condição para uma porta.
Roteador	Roteia dados em várias transformações com base em uma expressão de grupo. Por exemplo, use essa transformação para comparar salários de funcionários em três níveis diferentes de pagamento. Para tanto, você pode criar três grupos na transformação Roteador. Por exemplo, crie uma expressão de grupo para cada intervalo salarial.	TRUE ou FALSE, dependendo de a linha atender a expressão de grupo especificada. Somente linhas que retornam TRUE passam por cada grupo definido pelo usuário nessa transformação. Linhas que retornam FALSE passam pelo grupo padrão.

Transformação	Expressão	Valor de Retorno
Estratégia de Atualização	Sinaliza uma linha para atualizar, excluir ou rejeitar. Você usará essa transformação quando quiser controlar atualizações para um destino com base em condições que você aplicar. Por exemplo, você pode usar a transformação Estratégia para sinalizar todas as linhas de um cliente para atualização quando o endereço de correspondência for alterado ou rejeitar todas as linhas de funcionários que não trabalhem mais para a empresa.	Código numérico para atualizar, inserir, excluir ou rejeitar. A transformação aplica esse valor a cada linha que passa por ela.
Controle de transação	Especifica uma condição para determinar a ação realizada pelo Serviço de Integração: confirmar, reverter ou não alterar a transação. Você usará essa transformação quando quiser controlar transações de confirmação e reversão com base em uma linha ou conjunto de linhas que passam pela transformação. Por exemplo, use essa transformação para confirmar um conjunto de linhas com base na data de entrada de um pedido.	Uma das variáveis incorporadas, dependendo de se uma linha atende à condição especificada: <ul style="list-style-type: none"> - TC_CONTINUE_TRANSACTION - TC_COMMIT_BEFORE - TC_COMMIT_AFTER - TC_ROLLBACK_BEFORE - TC_ROLLBACK_AFTER O Serviço de Integração realiza ações com base no valor retornado.

Usando o Editor de Expressão

Use o Editor de Expressão para compilar instruções do tipo SQL. Embora você possa inserir expressões manualmente, use o método apontar-e-clicar. Selecione funções, portas, variáveis e operadores na interface de apontar e clicar para minimizar erros ao construir expressões. O número máximo de caracteres que você pode incluir em uma expressão é 32.767.

Você pode avaliar expressões configuradas no Editor de Expressão de uma transformação de Expressão.

Ao testar uma expressão, você pode inserir dados de amostra e depois avaliar a expressão. Avalie expressões em transformações reutilizáveis no Designer de Transformação. Avalie expressões em transformações não reutilizáveis no Mapping Designer. Algumas funções não oferecem suporte à avaliação de uma expressão quando você define o tipo de dados de porta como Binary ou Date/Time.

Inserindo nomes de portas em uma expressão

Para transformações conectadas, se você usar nomes de porta em uma expressão, o Designer atualiza essa expressão quando você alterar os nomes de porta na transformação. Por exemplo, você escreve uma expressão válida que determina a diferença entre duas datas, Date_Promised e Date_Delivered. Mais tarde, se você alterar o nome da porta Date_Promised para Due_Date, o Designer altera o nome da porta Date_Promised para Due_Date na expressão.

Nota: Você pode propagar o nome Due_Date para outras transformações não reutilizáveis que dependem desta porta no mapeamento.

Adicionando comentários

É possível adicionar comentários a uma expressão para fornecer informações descritivas sobre a expressão ou para especificar um URL válido para acessar documentações corporativas sobre a expressão.

Você pode adicionar comentários em qualquer uma das seguintes formas:

- Para adicionar comentários dentro da expressão, use os indicadores de comentário -- ou //.
- Para adicionar comentários em uma caixa de diálogo, clique no botão Comentários.

Validando expressões

Use o botão Validar para validar uma expressão. Se você não validar uma expressão, o Designer o fará quando você fechar o Editor de Expressão. Se a expressão for inválida, o Designer exibirá um aviso. Você poderá salvar a expressão inválida ou modificá-la. Não é possível executar uma sessão em um mapeamento com expressões inválidas.

Exibição do Editor de Expressão

O Editor de Expressão pode exibir expressões de sintaxe em cores diferentes para melhor legibilidade. Caso você tenha o controle mais atual do Rich Edit, o riched20.dll, instalado em seu sistema, o Editor de Expressão exibirá funções de expressão em azul, comentários em cinza e strings citadas em verde.

Você pode redimensionar o Editor de Expressão. Amplie a caixa de diálogo arrastando a partir das bordas. O Designer salva o novo tamanho da caixa de diálogo na qualidade de configuração de cliente.

Adicionando Expressões a uma Porta

Na Transformação Mascaramento de Dados, é possível adicionar uma expressão a uma porta de entrada. No caso de todas as outras transformações, adicione a expressão a uma porta de saída.

Realize as seguintes etapas para adicionar uma expressão a uma porta:

1. Na transformação, selecione a porta e abra o Editor de Expressão.
2. Digite a expressão.
Use as guias de Funções e de Portas e as teclas de operação.
3. Adicione comentários à expressão.
Use indicadores de comentários -- ou //.
4. Valide a expressão.
Use o botão Validar para validar a expressão.

Definindo Strings de Expressões em Arquivos de Parâmetros

O Serviço de Integração expande os parâmetros e variáveis de mapeamento em uma expressão depois que ele analisa a expressão. Se você tem uma expressão que muda frequentemente, pode definir a string da expressão em um arquivo de parâmetro, para que não tenha que atualizar os mapeamentos que usam a expressão quando ela mudar.

Para definir uma string de expressão em um arquivo de parâmetro, você cria um parâmetro ou variável de mapeamento para armazenar esta cadeia, e configura o parâmetro ou variável para a mesma no arquivo de parâmetro. O parâmetro ou variável que você cria deve ter IsExprVar configurado como verdadeiro. Quando IsExprVar é verdadeiro, o Serviço de Integração expande o parâmetro ou variável antes de analisar a expressão.

Por exemplo, para definir a expressão `IIF(color='red',5)` em um arquivo de parâmetro, execute as etapas a seguir:

1. No mapeamento que usa a expressão, crie um parâmetro de mapeamento \$\$Exp. Configure IsExprVar como verdadeiro e o tipo de dados como string.

2. No Editor de Expressão, configure a expressão para o nome do parâmetro de mapeamento conforme segue:

`$$Exp`

3. Configure a sessão ou fluxo de trabalho para usar um arquivo de parâmetro.
4. No arquivo de parâmetro, configure o valor `$$Exp` para a string de expressão conforme segue:

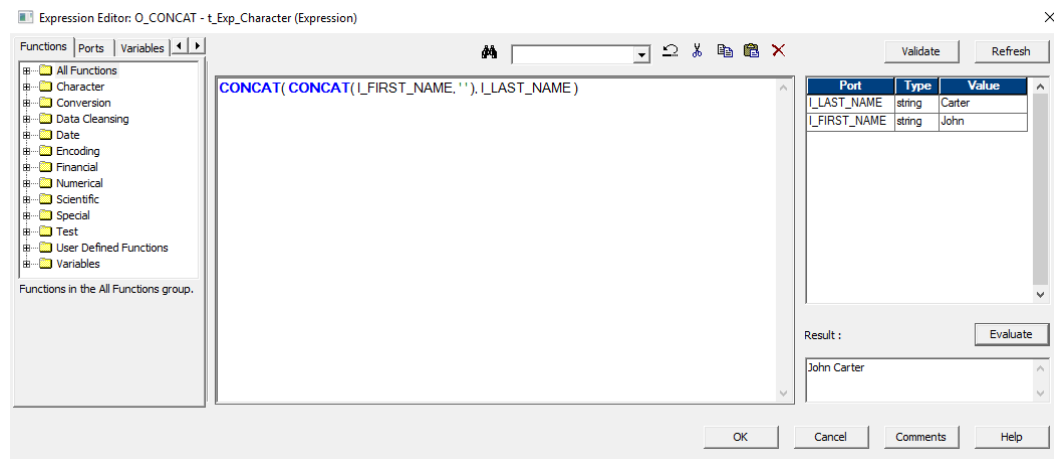
`$$Exp=IIF(color='red',5)`

Avaliar expressões

Você pode avaliar expressões configuradas no Editor de Expressão de uma transformação de Expressão. Ao testar uma expressão, você pode inserir dados de amostra e depois avaliar a expressão.

Se você editar uma condição de expressão, deverá selecionar Atualizar no painel de teste antes de poder avaliar os dados de teste. Se você não inserir uma expressão válida, o painel de teste não conseguirá preencher as portas. Quando você especifica parâmetros definidos pelo sistema, o valor inserido não é um valor em tempo de execução.

A imagem a seguir mostra como avaliar uma expressão de exemplo em um Editor de Expressão de uma transformação de Expressão:



Nota: Não é possível avaliar expressões dos tipos de dados Date/Time ou Binary.

Exemplo

Você precisa calcular ofertas promocionais com base no número total de pedidos recebidos para cada cliente antes de carregar todos os resultados no destino. Você desenvolve um mapeamento para calcular ofertas, definindo uma expressão dentro da transformação de Expressão. Você avalia a expressão para verificar o resultado antes que o operador execute o mapeamento.

Avaliando uma expressão

Você pode testar e verificar expressões no Editor de Expressão de uma transformação de Expressão.

1. No Editor de Expressão de uma transformação de Expressão, insira uma expressão.
2. Para validar a expressão, clique em Validar.
3. Para refletir as alterações mais recentes da condição de expressão na seção da expressão de teste no painel direito, clique em Atualizar.
4. No painel direito, insira valores de exemplo para as portas de entrada usadas na expressão.

5. Para avaliar a expressão, clique em Avaliar.

Restrições ao avaliar expressão

Certas restrições são aplicáveis quando você avalia uma expressão.

Considere as seguintes restrições ao avaliar expressões:

Restrições de transformação

É possível avaliar expressões nas seguintes transformações:

- Agregador
- Mascaramento de Dados
- Filtro
- Classificação
- Roteador
- Procedimento Armazenado
- Controle de Transações
- Atualizar Estratégia

Restrições de tipo de dados

Você não poderá avaliar uma expressão se o tipo de porta de entrada ou a função de retorno contiver um tipo de dados Binary ou Date/Time.

Restrições de portas

Você não pode avaliar uma expressão que usa a porta de Pesquisa ou Procedimento Armazenado.

Variáveis Locais

Use variáveis locais em transformações de Agregador, Expressão e Classificação para melhorar o desempenho. Você pode fazer referência a variáveis em uma expressão ou usá-las para armazenar dados temporariamente.

Você pode usar variáveis para realizar as seguintes tarefas:

- Armazenar dados temporariamente.
- Simplificar expressões complexas.
- Armazenar valores de linhas anteriores.
- Capturar vários valores de retorno de um procedimento armazenado.
- Comparar valores.
- Armazenar os resultados de uma transformação Pesquisa não concatenada.

Temporariamente armazenar dados e simplificar expressões complexas

As variáveis aumentam o desempenho quando você insere várias expressões relacionadas na mesma transformação. Você pode definir componentes como variáveis, em vez de analisar e validar os componentes da mesma expressão várias vezes na transformação.

Por exemplo, se uma transformação de Agregador usar a mesma condição de filtro antes de calcular somas e médias, você poderá definir essa condição como variável e reutilizá-la nos cálculos agregados.

Você pode simplificar as expressões complexas. Se um Agregador incluir o mesmo cálculo em diversas expressões, você poderá aumentar o desempenho criando uma variável para armazenar os resultados dos cálculos.

Por exemplo, você pode criar as seguintes expressões para encontrar o salário médio e o salário total usando os mesmos dados:

```
AVG( SALARY, ( ( JOB_STATUS = 'Full-time' ) AND (OFFICE_ID = 1000 ) ) )
SUM( SALARY, ( ( JOB_STATUS = 'Full-time' ) AND (OFFICE_ID = 1000 ) ) )
```

Em vez de inserir os mesmos argumentos para os dois cálculos, você pode criar uma porta variável para cada condição no cálculo e modificar a expressão para usar as variáveis.

A tabela a seguir mostra como usar variáveis para simplificar expressões complexas e armazenar dados temporariamente:

Porta	Valor
V_CONDITION1	JOB_STATUS = 'Full-time'
V_CONDITION2	OFFICE_ID = 1000
AVG_SALARY	AVG(SALARY, (V_CONDITION1 E V_CONDITION2))
SUM_SALARY	SUM(SALARY, (V_CONDITION1 AND V_CONDITION2))

Armazenar Valores nas Linhas

Você pode configurar variáveis em transformações para armazenar dados de linhas de origem. Você pode utilizar as variáveis em expressões de transformação.

Por exemplo, um arquivo de origem contendo as seguintes linhas:

```
California
California
California
Hawaii
Hawaii
New Mexico
New Mexico
New Mexico
```

Cada linha contém um estado. Você deve contar o número de linhas e retornar a contagem de linhas para cada estado:

```
California,3
Hawaii      ,2
New Mexico,3
```

Você pode configurar uma transformação do Agregador para agrupar as linhas de origem por estado e contar o número de linhas em cada grupo. Configure uma variável na transformação do Agregador para armazenar a contagem de linhas. Defina outra variável para armazenar o nome do estado da linha anterior.

A transformação do Agregador tem as seguintes portas:

Porta	Tipo de Porta	Expressão	Descrição
Estado	Passagem	n/a	O nome de um estado. As linhas de origem são agrupadas por nome do estado. A transformação de Agregador retorna uma linha para cada estado.
State_Count	Variável	<code>IIF (PREVIOUS_STATE = STATE, STATE_COUNT +1, 1)</code>	O número de linhas do Estado atual. Quando o valor da coluna do Estado atual for igual ao da coluna Previous_State, o Serviço de Integração incrementará State_Count. Caso contrário, ele redefinirá State_Count como 1.
Previous_State	Variável	Estado	O valor da coluna Estado na linha anterior. Quando o Serviço de Integração processa uma linha, ele transfere o valor de Estado para Previous_State.
State_Counter	Saída	State_Count	O número de linhas processadas pela transformação de Agregador para um estado. O Serviço de Integração retorna State_Counter uma vez para cada estado.

Capturar valores de procedimentos armazenados

As variáveis fornecem uma maneira de capturar várias colunas de valores de retorno em procedimentos armazenados.

Diretrizes para configurar portas variáveis

Considere os seguintes fatores quando configurar portas variáveis em uma transformação:

- **Ordem da porta.** O Serviço de Integração avalia as portas por dependência. A ordem das portas em uma transformação deve corresponder à ordem de avaliação: portas de entrada, portas variáveis, portas de saída.
- **Tipo de dados.** O tipo de dados que você escolhe reflete o valor retornado da expressão inserida.
- **Inicialização de variáveis.** O Serviço de Integração configura valores iniciais em portas variáveis, onde você pode criar contadores.

Ordem de portas

O Serviço de Integração avalia primeiro as portas de entrada. Em seguida, ele avalia as portas variáveis e, por último, as portas de saída.

O Serviço de Integração avalia portas na seguinte ordem:

1. **Portas de entrada.** O Serviço de Integração avaliará todas as portas antes, uma vez que não depende de nenhuma outra. Portanto, você pode criar portas de entrada em qualquer ordem. O Serviço de Integração não classifica as portas de entrada porque elas não fazem referência a outras portas.

2. **Portas variáveis.** Portas variáveis podem fazer referência a portas de entrada e portas variáveis, mas não a portas de saída. Como as portas variáveis podem fazer referência a portas de entrada, o Serviço de Integração as avalia após as portas de entrada. As variáveis podem fazer referência a outras variáveis, portanto, a ordem de exibição das portas variáveis é a mesma na qual o Serviço de Integração avalia cada variável.

Por exemplo, se você calcular o valor original de uma construção e ajustá-lo por depreciação, poderá criar o cálculo do valor original como uma porta variável. Essa porta variável deve aparecer antes da porta que ajusta por depreciação.

3. **Portas de saída.** O Serviço de Integração avalia as portas de saída por último, porque elas podem fazer referência a portas de entrada e portas variáveis. A ordem de exibição das portas de saída não importa, já que elas não podem fazer referência a outras portas de saída. Certifique-se de que as portas de saída são exibidas no final da lista de portas.

Tipo de Dados

Ao configurar uma porta como variável, é possível inserir qualquer expressão ou condição nela. O tipo de dados que você escolhe para a porta reflete o valor de retorno da expressão inserida. Se você especificar uma condição por meio da porta variável, qualquer tipo de dados numérico retornará os valores para TRUE (não zero) e FALSE (zero).

Inicialização de Variáveis

O Serviço de Integração não define o valor inicial das variáveis como NULL.

Ele usa as seguintes diretrizes para definir os valores iniciais das variáveis:

- zero para portas numéricas
- strings vazias para portas de string
- 01/01/0001 para portas de Data/Hora

Portanto, use variáveis como contadores, que precisam de valor inicial. Por exemplo, você pode criar uma variável numérica com a seguinte expressão:

```
VAR1 + 1
```

Essa expressão conta o número de linhas na porta VAR1. Se o valor inicial da variável fosse definido como NULL, a expressão seria sempre avaliada como NULL. Por isso o valor inicial é definido como zero.

Valores Padrão para Portas

Todas as transformações usam valores padrão que determinam como o Serviço de Integração trata os valores nulos de entrada e os erros de transformação de saída.

As portas de entrada, saída e entrada/saída têm um valor padrão do sistema que, às vezes, você pode substituir por um valor padrão definido pelo usuário. Os valores padrão contêm diferentes funções em diferentes tipos de portas:

- **Porta de entrada.** O valor padrão do sistema para portas de entrada nula é NULL. O valor padrão aparece como uma lacuna na transformação. Se um valor de entrada for NULL, o Serviço de Integração irá deixá-lo como NULL.

- **Porta de saída.** O valor padrão do sistema para erros de transformação de saída é ERROR. O valor padrão aparece na transformação como ERROR('transformation error'). Se ocorrer um erro de transformação, o Serviço de Integração ignorará a linha. O Serviço de Integração anota todas as linhas de entrada que a função ERROR ignora no arquivo de log.

Os seguintes erros são erros de transformação:

- Erros de conversão de dados, como passar um número para uma função de dados.
- Erros de avaliação de expressão, como dividir por zero.
- Chamadas para uma função ERROR.
- **Porta de passagem.** O valor padrão do sistema para a entrada nula é o mesmo que para as portas de entrada, NULL. O valor padrão do sistema aparece como uma lacuna na transformação. O valor padrão dos erros de transformação de saída é o mesmo que das portas de saída. O valor padrão dos erros de transformação de saída não aparece na transformação.

Nota: A Transformação Java converte tipos de dados PowerCenter® em tipos de dados Java, com base no tipo da porta da Transformação Java. Os valores padrão para a entrada nula variam de acordo com o tipo de dados Java.

A tabela a seguir mostra os valores padrão do sistema para portas em transformações conectadas:

Tipo de porta	Valor padrão	Comportamento do Serviço de Integração	Valor padrão com suporte definido pelo usuário
Entrada, Passagem	NULL	O Serviço de Integração transmite todos os valores de entrada nulos como NULL.	Entrada, Entrada/Saída
Saída, Passagem	ERROR	O Serviço de Integração chama a função ERROR para erros de transformação de porta de saída. O Serviço de Integração ignora linhas com erros e grava os dados de entrada e a mensagem de erro no arquivo do log.	Saída

Portas com variáveis não suportam valores padrão. O Serviço de Integração inicializa as portas com variáveis de acordo com o tipo de dados.

Você pode substituir alguns dos valores padrão para alterar o modo como o Serviço de Integração se comporta quando encontra valores de entrada nulos e erros de transformação de saída.

Valores Padrão Definidos pelo Usuário

Você pode substituir os valores padrão do sistema por valores padrão definidos pelo usuário para as portas compatíveis de entrada, de passagem e de saída em uma transformação conectada.

Use as seguintes regras e diretrizes para substituir os valores padrão do sistema por portas:

- **Portas de entrada.** Você pode inserir valores padrão definidos pelo usuário para as portas de entrada se não desejar que o Serviço de Integração trate os valores nulos como NULL. Se NULL for passado para a porta de entrada, o Serviço de Integração substituirá NULL pelo valor padrão.
- **Portas de saída.** Você poderá inserir valores padrão definidos pelo usuário para as portas de saída se não desejar que o Serviço de Integração ignore a linha ou se desejar que ele grave uma mensagem específica com a linha ignorada no log. Se você definir um valor padrão na porta de saída, o Serviço de Integração substituirá a linha pelo valor padrão quando a porta de saída tiver um erro de transformação.

- **Portas de passagem.** Você pode inserir valores padrão definidos pelo usuário para as portas de passagem se não desejar que o Serviço de Integração trate os valores nulos como NULL. Você não pode inserir valores padrão definidos pelo usuário para os erros de transformação de saída em uma porta de passagem.

Nota: O Serviço de Integração ignora valores padrão definidos pelo usuário para as transformações desconectadas. Por exemplo, se você chamar uma transformação de Pesquisa ou de Procedimento Armazenado por meio de uma expressão, o Serviço de Integração ignorará qualquer valor padrão definido pelo usuário e aplicará o valor padrão do sistema.

Use as seguintes opções para inserir valores padrão definidos pelo usuário:

- **Valor constante.** Use qualquer constante (numérica ou texto), incluindo NULL.
- **Expressão constante.** Você pode incluir uma função de transformação com parâmetros constantes.
- **ERROR.** Gere um erro de transformação. Grave a linha e uma mensagem no log da sessão ou no log de erro de linha.
- **ERROR.** Gere um erro de transformação. Grave a linha e uma mensagem no log do mapeamento ou no log de erro de linha.
- **ABORT.** Anule a sessão.
- **ABORT.** Anule o mapeamento.

Valores Constantes

Você pode inserir qualquer valor constante como um valor padrão. O valor constante deve corresponder ao tipo de dados da porta.

Por exemplo, um valor padrão para uma porta numérica deve ser uma constante numérica. Dentre alguns valores constantes, temos:

```
0
9999
NULL
'Unknown Value'
'Null input data'
```

Expressões Constantes

Uma expressão constante é qualquer expressão que utilize funções de transformação (à exceção de funções de agregação) para registrar expressões constantes. Não é possível usar valores de portas de entrada, entrada/saída ou variáveis em uma expressão constante.

Eis algumas expressões constantes válidas:

```
500 * 1.75
TO_DATE('January 1, 1998, 12:05 AM','MONTH DD, YYYY, HH:MI AM')
ERROR ('Null not allowed')
ABORT('Null not allowed')
SESSSTARTTIME
```

Você não pode usar valores de portas na expressão, pois o Serviço de Integração atribui valores padrão para todo o mapeamento ao iniciar a sessão.

Você não pode usar valores de portas na expressão, pois o Serviço de Integração atribui valores padrão para todo o mapeamento ao iniciá-lo.

Os seguintes exemplos não são válidos porque usam valores de portas:

```
AVG(IN_SALARY)
IN_PRICE * IN_QUANTITY
:LKP(LKP_DATES, DATE_SHIPPED)
```

Nota: Não é possível chamar um procedimento armazenado ou tabela de pesquisa a partir de uma expressão de valor padrão.

Funções ERROR e ABORT

Use as funções ERROR e ABORT para os valores padrão da porta de entrada e saída, e os valores de entrada para as portas de entrada/saída. O Serviço de Integração ignora a linha quando encontra a função ERROR. Ele anula a sessão quando encontra a função ABORT.

Use as funções ERROR e ABORT para os valores padrão da porta de entrada e saída, e os valores de entrada para as portas de entrada/saída. O Serviço de Integração ignora a linha quando encontra a função ERROR. Ele anula o mapeamento quando encontra a função ABORT.

Valores de Entrada Padrão Definidos pelo Usuário

Você poderá inserir um valor de entrada padrão definido pelo usuário se não desejar que o Serviço de Integração trate os valores nulos como NULL.

Para substituir valores nulos, conclua uma das seguintes tarefas:

- Substitua o valor nulo por um valor constante ou expressão constante.
- Ignore o valor nulo com uma função ERROR.
- Anule o mapeamento com a função ABORT.
- Anular a sessão com a função ABORT.

A tabela a seguir resume como o Serviço de Integração manipula a entrada nula para portas de entrada e entrada/saída:

Valor Padrão	Tipo do Valor Padrão	Descrição
NULL (exibe em branco)	Sistema	O Serviço de Integração transmite NULL.
Constante ou expressão Constante	Definidos pelo usuário	O Serviço de Integração substitui o valor nulo pelo valor da constante ou da expressão constante.
ERROR	Definidos pelo usuário	O Serviço de Integração trata isto como um erro de transformação: <ul style="list-style-type: none">- Aumenta a contagem de erros de transformação em 1.- Ignora a linha e grava a mensagem de erro no arquivo de log ou no log de erros de linha. O Serviço de Integração não grava linhas no arquivo rejeitado.
ABORT	Definidos pelo usuário	A sessão é anulada quando o Serviço de Integração encontra um valor de entrada nulo. O Serviço de Integração não aumenta a contagem de erros ou grava linhas no arquivo rejeitado. O mapeamento é anulado quando o Serviço de Integração encontra um valor de entrada nulo. O Serviço de Integração não aumenta a contagem de erros ou grava linhas no arquivo rejeitado.

Substituir Valores Nulos

Use um valor ou uma expressão constante para substituir um valor especificado para valores nulos em uma porta.

Por exemplo, se o nome de uma porta de string de entrada for DEPT_NAME e você quiser substituir os valores nulos pela string 'UNKNOWN DEPT', defina o valor padrão como 'UNKNOWN DEPT'. Dependendo da transformação, o Serviço de Integração passa 'UNKNOWN DEPT' para uma expressão ou variável dentro da transformação ou para a próxima transformação no fluxo de dados.

Por exemplo, o Serviço de Integração substitui todos os valores nulos em uma porta pela string 'UNKNOWN DEPT'.

DEPT_NAME	REPLACED VALUE
Housewares	Housewares
NULL	UNKNOWN DEPT
Produce	Produce

Ignorar Registros Nulos

Utilize a função ERROR como valor padrão quando não quiser que valores nulos passem para uma transformação. Por exemplo, talvez você queira ignorar uma linha quando o valor de entrada de DEPT_NAME for NULO. Você poderia usar a seguinte expressão como valor padrão:

```
ERROR('Error. DEPT is NULL')
```

Quando você usa a função ERROR como um valor padrão, o Serviço de Integração ignora a linha com o valor nulo. O Serviço de Integração grava todas as linhas ignoradas pela função ERROR no arquivo de log. Ele não grava essas linhas no arquivo rejeitado.

DEPT_NAME	RETURN VALUE
Housewares	Housewares
NULL	'Error. DEPT is NULL' (Row is skipped)
Produce	Produce

O seguinte log mostra onde o Serviço de Integração ignora a linha com o valor nulo:

```
TE_11019 Port [DEPT_NAME]: Default value is: ERROR(<<Transformation Error>> [error]:
Error. DEPT is NULL
... error('Error. DEPT is NULL')
).
CMN_1053 EXPTRANS: : ERROR: NULL input column DEPT_NAME: Current Input data:
CMN_1053 Input row from SRCTRANS: Rowdata: ( RowType=4 Src Rowid=2 Targ Rowid=2
  DEPT_ID (DEPT_ID:Int): "2"
  DEPT_NAME (DEPT_NAME:Char.25): "NULL"
  MANAGER_ID (MANAGER_ID:Int): "1"
)
```

Anular a Sessão

Use a função ABORT para interromper a sessão quando o Serviço de Integração encontrar valores de entrada nulos.

Valores de Saída Padrão Definidos pelo Usuário

Você pode criar valores padrão definidos pelo usuário para substituir os valores padrão do sistema para portas de saída.

Você poderá inserir valores padrão definidos pelo usuário para as portas de saída se não desejar que o Serviço de Integração ignore linhas com erros ou se desejar que ele grave uma mensagem específica com a

linha ignorada no log. Você pode inserir valores padrão para concluir as seguintes funções quando o Serviço de Integração encontrar erros na transformação de saída:

- Substituir o erro por um valor constante ou expressão constante. O Serviço de Integração não ignora a linha.
- Anular a sessão com a função ABORT.
- Anule o mapeamento com a função ABORT.
- Gravar mensagens específicas no log para os erros de transformação.

Você não pode inserir valores de saída padrão definidos pelo usuário, para as portas de entrada/saída.

A tabela a seguir resume como o Serviço de Integração manipula os erros de transformação da porta de saída e os valores padrão nas transformações:

Valor Padrão	Tipo do Valor Padrão	Descrição
Erro de Transformação	Sistema	Quando um erro de transformação ocorrer e você não tiver substituído o valor padrão, o Serviço de Integração executará as seguintes tarefas: <ul style="list-style-type: none">- Aumenta a contagem de erros de transformação em 1.- Ignora a linha e grava o erro e a linha de entrada no arquivo de log da sessão ou no log de erro de linha, dependendo da configuração da sessão. O Serviço de Integração não grava a linha no arquivo rejeitado.
Constante ou Expressão Constante	Definidos pelo usuário	O Serviço de Integração substitui o erro pelo valor padrão. O Serviço de Integração não aumenta a contagem de erros e nem grava uma mensagem no log.
ABORT	Definidos pelo usuário	A sessão é anulada e o Serviço de Integração grava uma mensagem no log da sessão. O mapeamento é anulado e o Serviço de Integração grava uma mensagem no log. O Serviço de Integração não aumenta a contagem de erros ou grava linhas no arquivo rejeitado.

Substituir Erros

Se você não quiser que o Serviço de Integração ignore uma linha quando ocorrer um erro de transformação, use uma constante ou uma expressão constante como valor padrão de uma porta de saída.

Por exemplo, se você tiver uma porta de saída numérica chamada NET_SALARY e quiser usar o valor constante '9999' quando ocorrer um erro de transformação, atribua o valor padrão 9999 à porta NET_SALARY. Se houver um erro de transformação (como uma divisão por zero) durante a computação do valor de NET_SALARY, o Serviço de Integração usará o valor padrão 9999.

Abortando a sessão

Use a função ABORT como valor padrão em uma porta de saída se você não quiser permitir erros de transformação.

Gravar Mensagens no Log da Sessão ou em Logs de Erros de Linha

Você pode inserir um valor padrão definido pelo usuário na porta de saída se quiser que o Serviço de Integração grave uma mensagem específica no log da sessão com a linha ignorada. O padrão do sistema é ERROR ('transformation error'). O Serviço de Integração grava a mensagem 'transformation error' no log da sessão junto com a linha ignorada. Se quiser gravar outra mensagem, você poderá substituir a mensagem 'transformation error'.

Quando você habilita o log de erros de linha, o Serviço de Integração grava mensagens de erro no log de erros, em vez do log da sessão. O Serviço de Integração não registra os erros de reversão ou de confirmação da transformação Controle de Transações. Para gravar linhas no log da sessão, além do log de erros de linha, habilite o rastreamento detalhado de dados.

Funções ERROR em Expressões da Porta de Saída

Se você inserir uma expressão que usa a função ERROR, o valor padrão definido pelo usuário para a porta de saída poderia substituir essa função.

Por exemplo, insira a expressão a seguir que instrui o Serviço de Integração a usar o valor 'Venda Negativa' quando ele encontrar um erro:

```
IIF( TOTAL_SALES>0, TOTAL_SALES, ERROR ('Negative Sale'))
```

Os exemplos a seguir mostram como os valores padrão definidos pelo usuário podem substituir a função ERROR na expressão:

- **Valor de constante ou expressão.** O valor de constante ou expressão substitui a função ERROR na expressão da porta de saída.
Por exemplo, se você inserir '0' como o valor padrão, o Serviço de Integração substitui a função ERROR na expressão da porta de saída. Ele passa o valor 0 quando encontra um erro. Ele não ignora a linha nem grava 'Venda Negativa' no log.
- **ABORT.** A função ABORT substitui a função ERROR na expressão de porta de saída.
Se você usar a função ABORT como o valor padrão, o Serviço de Integração será anulado quando ocorrer um erro de transformação. A função ABORT substitui a função ERROR na expressão de porta de saída.
- **ERROR.** Se você usar a função ERROR como o valor padrão, o Serviço de Integração incluirá a seguinte informação no log:
 - Mensagem de erro do valor padrão
 - Mensagem de erro indicada na função ERROR na expressão de porta de saída
 - Linha ignorada

Por exemplo, você pode substituir o valor padrão pela função ERROR a seguir:

```
ERROR('No default value')
```

O Serviço de Integração ignora a linha e inclui ambas as mensagens de erro no log.

```
TE_7007 Transformation Evaluation Error; current row skipped...
TE_7007 [<<Transformation Error>> [error]: Negative Sale
... error('Negative Sale')
]
Sun Sep 20 13:57:28 1998
TE_11019 Port [OUT_SALES]: Default value is: ERROR(<<Transformation Error>> [error]:
No default value
... error('No default value')
```

Regras gerais para valores padrão

Use as seguintes regras e diretrizes ao criar valores padrão:

- O valor padrão deve ser um NULL, um valor constante, uma expressão constante, uma função ERROR ou uma função ABORT.
- Para portas de entrada/saída, o Serviço de Integração usa os valores padrão para manipular os valores de entrada nulos. O valor padrão de saída das portas de entrada/saída é sempre ERROR('Transformation Error').
- Portas variáveis não usam valores padrão.
- Você pode atribuir valores padrão ao grupo por portas, nas transformações de Agregador e Classificação.
- Nem todos os tipos de porta em todas as transformações usam os valores padrão definidos pelo usuário. Se uma porta não permite valores padrão definidos pelo usuário, o campo do valor padrão é desativado.
- Nem todas as transformações permitem valores padrão definidos pelo usuário.
- Se uma transformação não for conectada ao fluxo de dados do mapeamento, o Serviço de Integração ignora os valores padrão definidos pelo usuário.
- Se qualquer porta de entrada estiver desconectada, presume-se que o seu valor seja NULL e o Serviço de Integração usa o valor padrão para essa porta de entrada.
- Se o valor padrão da porta de entrada contém a função ABORT e o valor de entrada é NULL, o Serviço de Integração interrompe a sessão imediatamente. Use a função ABORT como valor padrão para restringir valores de entrada nulos. O primeiro valor nulo em uma porta de entrada interrompe a sessão.
- Se o valor padrão da porta de saída contém a função ABORT e qualquer erro de transformação ocorrer para essa porta, a sessão é interrompida imediatamente. Use a função ABORT como valor padrão para reforçar as regras estritas para os erros de transformação. O primeiro erro de transformação para esta porta interrompe a sessão.
- Se um valor padrão da porta de entrada contiver a função ABORT e o valor de entrada for NULL, o Serviço de Integração interromperá o mapeamento imediatamente. Use a função ABORT como valor padrão para restringir valores de entrada nulos. O primeiro valor nulo em uma porta de entrada interrompe o mapeamento.
- Se o valor padrão da porta de saída contiver a função ABORT e algum erro de transformação ocorrer para essa porta, o mapeamento será interrompido imediatamente. Use a função ABORT como valor padrão para reforçar as regras estritas para os erros de transformação. O primeiro erro de transformação para essa porta interrompe o mapeamento.
- A função ABORT, os valores constantes e as expressões constantes substituem as funções ERROR configuradas nas expressões da porta de saída.

Validação do Valor Padrão

Você pode validar os valores padrão à medida que os insere. O Designer inclui um botão Validar, para que você possa garantir valores padrão válidos. Uma mensagem aparece, indicando se o padrão é válido.

O Designer também valida os valores padrão quando você salva um mapeamento. Se você inserir um valor padrão inválido, o Designer marcará o mapeamento como não válido.

A ferramenta Developer valida os valores padrão à medida que você os insere.

A ferramenta Developer valida os valores padrão quando você salva um mapeamento. Se você inserir um valor padrão que não é válido, a ferramenta Developer marcará o mapeamento como não válido.

Configurando o nível de rastreamento em transformações

Quando você configura uma transformação, pode configurar a quantidade de detalhes que o Serviço de Integração escreve no log da sessão.

A tabela a seguir descreve os níveis de rastreamento do log da sessão:

Nível de rastreamento	Descrição
Normal	O Serviço de Integração registra as informações de inicialização e status, os erros encontrados e as linhas ignoradas devido a erros na linha de transformação. Resume os resultados da sessão, mas não no nível de linhas individuais.
Conciso	O Serviço de Integração registra mensagens de informações e erro de inicialização e a notificação de dados rejeitados.
Inicialização Detalhada	Além do rastreamento normal, o Serviço de Integração registra detalhes adicionais da inicialização, nomes de índice e arquivos de dados usados e estatísticas de transformação detalhadas.
Dados Detalhados	<p>Além do rastreamento da inicialização detalhado, o Serviço de Integração registra cada linha que transfere no mapeamento. Ele também nota onde o Serviço de Integração trunca dados de string para ajustar a precisão de uma coluna, e fornece estatísticas de transformação detalhadas.</p> <p>Permite que o Serviço de Integração escreva erros no log da sessão e no log de erros, quando você ativa o registro de erros de linha.</p> <p>Quando você configura o nível de rastreamento para dados detalhados, o Serviço de Integração escreve dados de linha para todas as linhas em um bloco, ao processar uma transformação.</p>

Por padrão, o nível de rastreamento para cada transformação é Normal. Altere o nível de rastreamento para uma configuração Detalhada apenas quando precisar depurar uma transformação que não está se comportando conforme o esperado. Para adicionar um leve aumento no desempenho, você pode também configurar o nível de rastreamento para Conciso, escrevendo o mínimo de detalhes no log da sessão ao executar um fluxo de trabalho que contém a transformação.

Quando você configura uma sessão, pode substituir os níveis de rastreamento para transformações individuais por um único nível de rastreamento para todas as transformações na sessão.

Transformações Reutilizáveis

Os mapeamentos podem conter transformações reutilizáveis e não reutilizáveis. As transformações não reutilizáveis existem dentro de um único mapeamento. As transformações reutilizáveis podem ser usadas em vários mapeamentos.

Por exemplo, você poderia criar uma transformação Expressão que calculasse o imposto sobre valor agregado das vendas no Canadá, o que seria útil para analisar o custo de fazer negócios nesse país. Em vez de fazer o mesmo trabalho toda vez, você pode criar uma transformação reutilizável. Quando precisar incorporar a transformação a um mapeamento, você adicionará uma instância dela ao mapeamento. Mais tarde, se você mudar a definição da transformação, todas as instâncias dela herdarão as alterações.

O Designer armazena cada transformação reutilizável como metadados separados de qualquer mapeamento que a utilize. Se você analisar o conteúdo de uma pasta no Navegador, verá a lista de todas as transformações reutilizáveis nessa pasta.

Cada transformação reutilizável enquadra-se dentro de uma categoria de transformações disponíveis no Designer. Por exemplo, você pode criar uma transformação Agregador reutilizável para executar os mesmos cálculos agregados em vários mapeamentos, ou uma transformação Procedimento Armazenado reutilizável para chamar o mesmo procedimento armazenado em vários mapeamentos.

Você pode criar a maioria das transformações como não reutilizáveis ou reutilizáveis. No entanto, você só pode criar a transformação Procedimento Externo como uma transformação reutilizável.

Ao adicionar instâncias de uma transformação reutilizável a mapeamentos, tome cuidado para que as alterações feitas na transformação não invalidem o mapeamento nem gerem dados inesperados.

Instâncias e alterações herdadas

Ao adicionar uma transformação reutilizável a um mapeamento, você adiciona uma instância da transformação. A definição da transformação ainda existe fora do mapeamento, ao passo que uma instância da transformação aparece dentro do mapeamento.

Uma vez que a instância de uma transformação reutilizável serve de ponteiro a essa transformação, quando você altera a transformação no Transformation Developer, suas instâncias refletem essas alterações. Em vez de atualizar a mesma transformação em cada mapeamento que a utilize, você pode atualizar a transformação reutilizável uma vez, e todas as instâncias da transformação herdam a mudança. Observe que as instâncias não herdam alterações feitas nas configurações de propriedades, apenas modificações em portas, expressões e no nome da transformação.

Mapeando variáveis em expressões

Use os parâmetros e variáveis de mapeamento nas expressões da transformação reutilizável. Quando o Designer valida o parâmetro ou variável, ele o trata como um tipo de dados Inteiro. Quando você usa a transformação em um mapplet ou mapeamento, o Designer valida a expressão novamente. Se o parâmetro ou variável de mapeamento não existir no mapplet ou mapeamento, o Designer registra um erro.

Criando Transformações Reutilizáveis

Você pode criar uma transformação reutilizável usando os seguintes métodos:

- **Projete-a no Transformation Developer.** No Transformation Developer, você pode criar novas transformações reutilizáveis.
- **Promova uma transformação não reutilizável do Mapping Designer.** Depois de adicionar uma transformação a um mapeamento, você pode promovê-la para o status de transformação reutilizável. A transformação projetada no mapeamento se torna uma instância de uma transformação reutilizável mantida em outro local do repositório.

Se você promove uma transformação para o status reutilizável, não pode rebaixá-la. No entanto, você pode criar uma instância não reutilizável da mesma.

Nota: As transformações Gerador de sequência devem ser reutilizáveis nos mapplets. Você não pode rebaixar transformações Gerador de Sequência reutilizáveis para não reutilizáveis em um mapplet.

Para criar uma transformação reutilizável:

1. No Designer, troque para o Transformation Developer.

2. Clique no botão na barra de ferramentas da Transformação correspondente ao tipo de transformação que você deseja criar.
3. Arraste dentro da pasta de trabalho para criar a transformação.
4. Clique duas vezes na barra de título da transformação para abrir o diálogo que exibe suas propriedades.
5. Clique no botão Renomear e insira um nome descritivo para a transformação e clique em OK.
6. Clique na guia Portas, e em seguida adicione qualquer porta de entrada e saída necessária para esta transformação.
7. Configure as outras propriedades da transformação e clique em OK.

Estas propriedades variam de acordo com a transformação criada. Por exemplo, se você criar uma transformação Expressão, precisa inserir uma expressão para uma ou mais portas de saída da transformação. Se você criar uma transformação Procedimento Armazenado, precisa identificar o procedimento armazenado que será chamado.

Promovendo Transformações Não Reutilizáveis

A outra técnica para criar uma transformação reutilizável é promover uma transformação existente em um mapeamento. Marque a opção Tornar Reutilizável na caixa de diálogo Editar Transformações para instruir o Designer a promover a transformação e criar uma instância no mapeamento.

Para promover uma transformação não reutilizável:

1. No Designer, abra um mapeamento e clique duas vezes na barra de título da transformação que quer promover.
2. Selecione a opção Tornar Reutilizável.
3. Quando for perguntado se tem certeza de que quer promover a transformação, clique em Sim.
4. Clique em OK para retornar ao mapeamento.

Agora, a transformação recém-promovida aparecerá na lista de transformações reutilizáveis na pasta em que você está trabalhando.

Criando Instâncias Não Reutilizáveis de Transformações Reutilizáveis

É possível criar uma instância não reutilizável de uma transformação reutilizável em um mapeamento. A transformação reutilizável deve ser tornada não reutilizável na mesma pasta. Se desejar ter uma instância não reutilizável de uma transformação reutilizável em uma pasta diferente, primeiramente é necessário fazer uma instância não reutilizável da transformação na pasta de origem, depois copiá-la para a pasta de destino.

Para criar uma instância não reutilizável de uma transformação reutilizável:

1. No Designer, abra um mapeamento.
2. No Navegador, selecione uma transformação atual e arraste-a ao espaço de trabalho de mapeamento. Pressione e segure a tecla Ctrl antes de liberar a transformação.

A barra de status exibe a mensagem a seguir:

`Make a non-reusable copy of this transformation and add it to this mapping.`

3. Libere a transformação.

O Designer cria uma instância não reutilizável da transformação reutilizável atual.

Adicionando Transformações Reutilizáveis a Mapeamentos

Depois de criar uma transformação reutilizável, você pode adicioná-la aos mapeamentos.

Para adicionar uma transformação reutilizável:

1. No Designer, troque para o Mapping Designer.
2. Abra ou crie um mapeamento.
3. Na lista de objetos de repositório, faça uma busca detalhada até encontrar a transformação reutilizável que deseja ter na seção Transformações de uma pasta.
4. Arraste a transformação do Navegador para o mapeamento.

Uma cópia (ou instância) da transformação reutilizável é exibida.

5. Vincule a nova transformação a outras transformações ou definições de destino.

Modificando uma transformação reutilizável

As alterações em uma transformação reutilizável, que você insere através do Transformation Developer, são imediatamente refletidas em todas as instâncias dessa transformação. Embora este recurso seja uma sólida maneira de salvar o trabalho e impor padrões, você corre o risco de invalidar o mapeamento quando modifica uma transformação reutilizável.

Para ver quais mapeamentos, mapplets ou atalhos podem ser afetados pelas alterações efetuadas em uma transformação, selecione a transformação no espaço de trabalho ou no Navegador, clique com o botão direito e selecione Exibir Dependências.

Se você fizer uma das seguintes alterações à transformação reutilizável, os mapeamentos que utilizam suas instâncias podem ser invalidados:

- Quando você exclui uma porta ou múltiplas portas de uma transformação, desconecta a instância do fluxo de dados parcial ou total através do mapeamento.
- Quando você altera um tipo de dados de porta, impossibilita o mapeamento de dados dessa porta para outra, usando um tipo de dados incompatível.
- Quando você altera um nome de porta, as expressões que se referem à porta deixam de ser válidas.
- Quando você insere uma expressão inválida na transformação reutilizável, os mapeamentos que usam a transformação deixam de ser válidos. O Serviço de Integração não pode executar sessões com base em mapeamentos inválidos.

Revertendo para a Transformação Reutilizável Original

Se você alterar as propriedades de uma transformação reutilizável em um mapeamento, poderá voltar para as propriedades da transformação reutilizável original, clicando no botão Reverter.

CAPÍTULO 2

Transformação Agregador

Este capítulo inclui os seguintes tópicos:

- [Visão Geral da Transformação Agregador, 51](#)
- [Componentes da Transformação Agregador, 52](#)
- [Configuração de caches agregados, 53](#)
- [Expressões agregadas, 54](#)
- [Agrupar por Portas, 55](#)
- [Usando Entrada Classificada, 57](#)
- [Criação de uma Transformação Agregador, 59](#)
- [Dicas para Transformações Agregador, 59](#)
- [Solução de Problemas de Transformações Agregador, 60](#)

Visão Geral da Transformação Agregador

A transformação Agregador executa cálculos agregados, como médias e somas. O Serviço de Integração executa cálculos agregados à medida que lê e armazena grupos de dados e dados de linha em um cache agregado. A transformação de Agregador é uma transformação ativa.

A transformação de Agregador é diferente da transformação de Expressão, porque você usa a transformação de Agregador para executar cálculos em grupos. A transformação de Expressão permite executar cálculos linha a linha.

Quando você usa a linguagem da transformação para criar expressões agregadas, pode usar cláusulas condicionais para filtrar linhas, o que permite mais flexibilidade que a linguagem SQL.

Depois de criar uma sessão que inclua uma transformação Agregador, você pode ativar a opção de sessão Agregação Incremental. Quando o Serviço de Integração executa a agregação incremental, ele transfere os dados de origem através do mapeamento e usa os dados de cache históricos para executar os cálculos de agregação de maneira incremental.

Componentes da Transformação Agregador

O Agregador é uma transformação ativa que altera o número de linhas no pipeline. A transformação do Agregador apresenta os seguintes componentes e opções:

- **Cache agregado.** O Serviço de Integração armazena dados no cache agregado até que os cálculos agregados sejam concluídos. O Serviço de Integração armazena valores de grupo em um cache de índice e armazena dados em linha no cache de dados.
- **Expressão agregada.** Insira uma expressão em uma porta de saída. A expressão pode conter expressões não agregadas e orações condicionais.
- **Agrupamento por porta.** Indica como criar grupos. É possível configurar uma porta de entrada, entrada/saída, saída ou variável para o grupo. Ao agrupar dados, a transformação do Agrupador efetua a saída da última linha de cada grupo, salvo indicação em contrário.
- **Entrada classificada.** Selecione esta opção para aperfeiçoar o desempenho da sessão. Para usar a entrada classificada, é necessário passar dados à transformação do Agregador classificada por grupo e por porta, em ordem crescente ou decrescente.

É possível configurar os componentes e opções da transformação do Agregador na guia Propriedades e Portas.

Configuração das Propriedades da Transformação Agregador

Modifique as propriedades da Transformação Agregador na guia Propriedades.

Configure as seguintes opções:

Configuração do Agregador	Descrição
Diretório de Cache	Diretório local em que o Serviço de Integração cria o índice e os arquivos do cache de dados. Por padrão, o Serviço de Integração usa o diretório inserido no Workflow Manager para a variável de processo \$PMCacheDir. Se você inserir um novo diretório, certifique-se de que ele exista e contenha espaço em disco suficiente para os caches agregados. Se você ativou a agregação incremental, o Serviço de Integração cria um backup dos arquivos sempre que você executa a sessão. O diretório de cache deve conter espaço em disco suficiente para dois conjuntos de arquivos.
Nível de Rastreamento	Volume de detalhes exibidos no log de sessão para esta transformação.
Entrada Classificada	Indica que os dados de entrada são pré-classificados por grupos. Selecione essa opção apenas se o mapeamento transferir dados classificados para a transformação de Agregador.
Tamanho de Cache de Dados do Agregador	Tamanho do cache de dados para a transformação. O tamanho padrão do cache é 2.000.000 bytes. Se o tamanho total do cache de sessão configurado for 2 GB (2.147.483.648 bytes) ou mais, você deverá executar a sessão em um Serviço de Integração de 64 bits. É possível usar um valor numérico para o cache, usar um valor de cache proveniente de um arquivo de parâmetros ou configurar o Serviço de Integração para definir o tamanho do cache usando a configuração Automático. Se você configurar o Serviço de Integração para determinar o tamanho do cache, também poderá especificar a quantidade máxima de memória alocada ao cache pelo Serviço de Integração.

Configuração do Agregador	Descrição
Tamanho de cache de índice do agregador	Tamanho do cache de índice para a transformação. O tamanho padrão do cache é 1.000.000 bytes. Se o tamanho total do cache de sessão configurado for 2 GB (2.147.483.648 bytes) ou mais, você deverá executar a sessão em um Serviço de Integração de 64 bits. É possível usar um valor numérico para o cache, usar um valor de cache proveniente de um arquivo de parâmetros ou configurar o Serviço de Integração para definir o tamanho do cache usando a configuração Automático. Se você configurar o Serviço de Integração para determinar o tamanho do cache, também poderá especificar a quantidade máxima de memória alocada ao cache pelo Serviço de Integração.
Escopo da Transformação	Especifica como o Serviço de Integração aplica a lógica de transformação aos dados de entrada: <ul style="list-style-type: none"> - Transação. Aplica a lógica de transformação a todas as linhas de uma transação. Escolha Transação quando uma linha de dados depender de todas as linhas na mesma transação, mas não de linhas em outras transações. - Todas as Entradas. Aplica a lógica de transformação em todos os dados de entrada. Quando você escolhe Todas as Entradas, o PowerCenter descarta os limites de transação de entrada. Escolha Todas as entradas quando uma linha de dados depender de todas as linhas na origem.

Configuração das Portas da Transformação Agregador

Para configurar portas na Transformação Agregador, complete as seguintes tarefas:

- Insira uma expressão em qualquer porta de saída, usando cláusulas condicionais ou funções não agregadas na porta.
- Crie múltiplas portas de saída de agregação.
- Configure qualquer porta de entrada, entrada/saída, saída ou variável como um grupo por porta.
- Melhore o desempenho, conectando somente as portas de entrada/saída necessárias para as transformações subsequentes, reduzindo o tamanho do cache de dados.
- Use portas variáveis para as variáveis locais.
- Crie conexões para outras transformações à medida que você insere uma expressão.

Configuração de caches agregados

Quando você executa uma sessão que usa uma transformação Agregador, o Serviço de Integração cria o índice e os caches de dados na memória, a fim de processar a transformação. Se o Serviço de Integração precisar de mais espaço, ele armazena os valores de estouro nos arquivos de cache.

É possível usar um valor numérico para o cache, usar um valor de cache proveniente de um arquivo de parâmetros ou configurar o Serviço de Integração para definir o tamanho do cache usando a configuração Automático.

Nota: O Serviço de Integração usa a memória para processar uma transformação Agregador com portas classificadas. O Serviço de Integração não usa a memória de cache. Você não precisa configurar a memória de cache para as transformações Agregador que usam portas classificadas.

Expressões agregadas

O Designer permite as expressões agregadas apenas na transformação do Agregador. Uma expressão agregada pode incluir cláusulas condicionais e funções não agregadas. A expressão também pode incluir uma função agregada dentro de outra, como:

```
MAX( COUNT( ITEM ) )
```

O resultado de uma expressão agregada varia com base no grupo por portas na transformação. Por exemplo, quando o Serviço de Integração calcula a seguinte expressão agregada sem um grupo por portas definido, ele encontra a quantidade total de itens vendidos:

```
SUM( QUANTITY )
```

No entanto, se você usa a mesma expressão e agrupa pela porta ITEM, o Serviço de Integração retorna a quantidade total de itens vendidos, por item.

Você pode criar uma expressão agregada em uma porta de saída e usar múltiplas portas agregadas em uma transformação.

TÓPICOS RELACIONADOS:

- [“Trabalhando com expressões” na página 31](#)

Funções de agregação

Use as seguintes funções de agregação dentro de uma transformação Agregador. É possível aninhar uma função de agregação dentro de outra função de agregação.

A linguagem de transformação inclui as seguintes funções de agregação:

- AVG
- COUNT
- FIRST
- LAST
- MAX
- MEDIAN
- MIN
- PERCENTILE
- STDDEV
- SUM
- VARIANCE

Ao usar qualquer uma dessas funções, é necessário usá-las em uma expressão dentro de uma transformação Agregador.

Funções de Agregação Aninhadas

É possível incluir funções de nível único múltiplas ou funções aninhadas múltiplas em diferentes portas de saída em uma transformação de Agregador. Entretanto, não é possível incluir as funções de nível único e aninhada em uma transformação de Agregador. Logo, se uma transformação de Agregador contiver uma função de nível único em uma porta de saída, não será possível usar uma função aninhada em nenhuma outra porta nessa transformação. Quando você inclui funções de nível único e aninhadas na mesma

transformação de Agregador, o Designer marca o mapeamento ou mapplet como inválido. Se você precisar criar funções aninhadas e de nível único, crie transformações de Agregador separadas.

Cláusulas condicionais

Use cláusulas condicionais na expressão agregada para reduzir o número de linhas usadas na agregação. A cláusula condicional pode ser qualquer cláusula que avalie para TRUE ou FALSE.

Por exemplo, use a seguinte expressão para calcular o total de comissões dos funcionários que excederam sua cota trimestral:

```
SUM( COMMISSION, COMMISSION > QUOTA )
```

Funções de não agregação

Você também pode usar as funções de não agregação na expressão de agregação.

A seguinte expressão retorna o número mais alto de itens vendidos para cada item (agrupados por item). Se nenhum item foi vendido, a expressão retorna 0.

```
IIF( MAX( QUANTITY ) > 0, MAX( QUANTITY ), 0 )
```

Valores Null em funções de agregação

Ao configurar o Serviço de Integração, você pode escolher como deseja que o Serviço de Integração trate os valores nulos em funções de agregação. Você pode optar por tratar os valores nulos nas funções de agregação como NULL ou zero. Por padrão, o Serviço de Integração trata valores nulos como NULL em funções de agregação.

Agrupar por Portas

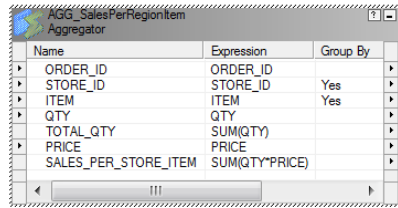
A transformação Agregador permite definir os grupos para agregações, em vez de executar a Agregação em todos os dados de entrada. Por exemplo, em vez de encontrar as vendas totais da empresa, você pode encontrar as vendas totais agrupadas por região.

Para definir um grupo para a expressão agregada, selecione as portas apropriadas de entrada, entrada/saída, saída e variáveis na transformação Agregador. Você pode selecionar múltiplos grupos por portas, a fim de criar um novo grupo para cada combinação exclusiva. Em seguida, o Serviço de Integração executa a agregação definida para cada grupo.

Quando você agrupa os valores, o Serviço de Integração produz uma linha para cada grupo. Se você não agrupar os valores, o Serviço de Integração retorna uma linha para todas as linhas de entrada. O Serviço de Integração tipicamente retorna a última linha de cada grupo (ou a última linha recebida) com o resultado da agregação. No entanto, se você especificar uma linha específica para ser retornada (por exemplo, usando a função FIRST), o Serviço de Integração retorna então a linha especificada.

Ao selecionar múltiplos grupos por portas na transformação Agregador, o Serviço de Integração usa a ordem da porta para determinar a ordem do agrupamento. Uma vez que a ordem de agrupamento pode afetar os resultados, faça-a por portas para garantir o agrupamento apropriado. Por exemplo, os resultados do agrupamento por ITEM_ID e depois QUANTITY podem ser diferentes de quando você agrupa por ITEM_ID e depois QUANTITY, porque os valores numéricos para a quantidade não são necessariamente exclusivos.

A seguinte transformação Agregador agrupa primeiro por STORE_ID e depois por ITEM:



Se você enviar os seguintes dados por esta transformação Agregador:

STORE_ID	ITEM	QTY	PRICE
101	'battery'	3	2.99
101	'battery'	1	3.19
101	'battery'	2	2.59
101	'AAA'	2	2.45
201	'battery'	1	1.99
201	'battery'	4	1.59
301	'battery'	1	2.45

O Serviço de Integração executa o cálculo agregado nos grupos exclusivos a seguir:

STORE_ID	ITEM
101	'battery'
101	'AAA'
201	'battery'
301	'battery'

Em seguida, o Serviço de Integração transfere a última linha recebida, junto com os resultados da agregação, conforme segue:

STORE_ID	ITEM	TOTAL_QTY	SALES_PER_STORE_ITEM
101	'AAA'	2	4.90
101	'battery'	6	17.34
201	'battery'	5	8.35
301	'battery'	1	2.45

Expressões Não Agregadas

Use as expressões não agregadas no grupo por portas, para modificar ou substituir grupos. Por exemplo, se você deseja substituir 'bateria AAA' antes de agrupar, você pode criar um novo grupo por porta de saída, nomeado como CORRECTED_ITEM, usando a expressão a seguir:

```
IIF( ITEM = 'AAA battery', battery, ITEM )
```

Valores padrão

Defina um valor padrão para cada porta do grupo para substituir os valores de entrada nulos. Isso permite que o Serviço de Integração inclua grupos de itens nulos na agregação.

TÓPICOS RELACIONADOS:

- [“Valores Padrão para Portas” na página 39](#)

Usando Entrada Classificada

Você pode usar a opção de entrada classificada para aumentar o desempenho de transformações do Agregador. Quando você usar entrada classificada, o Serviço de Integração assume todos os dados classificados por grupo e realiza cálculos enquanto lê as linhas de um grupo. Quando necessário, armazena informações do grupo na memória. Para usar a opção Entrada Classificada, você deverá passar dados classificados à transformação do Agregador. Você pode aumentar o desempenho com portas classificadas ao configurar a sessão com diversas partições.

Quando você não usa a entrada classificada, o Serviço de Integração realiza cálculos agregados enquanto lê. Como os dados não são classificados, o Serviço de Integração armazena dados de cada grupo até ler toda a origem, para garantir que todos os cálculos agregados sejam precisos.

Por exemplo, uma transformação do Agregador tem STORE_ID e ITEM agrupados por portas, com a opção de entrada classificada selecionada. Quando você passar os dados a seguir pelo Agregador, o Serviço de Integração realizará uma agregação para as três linhas no grupo 101/bateria tão logo encontrar o novo grupo, 201/bateria:

STORE_ID	ITEM	QTY	PRICE
101	'battery'	3	2.99
101	'battery'	1	3.19
101	'battery'	2	2.59
201	'battery'	4	1.59
201	'battery'	1	1.99

Se você usar uma entrada classificada e não pré-classificar os dados corretamente, receberá resultados inesperados.

Condições de Entrada Classificadas

Não utilize uma entrada classificada se qualquer uma das condições a seguir for verdadeira:

- A expressão agregada utiliza funções de agregação aninhadas.
- A sessão utiliza uma agregação incremental.

Se você usar uma entrada classificada e não classificar os dados corretamente, a sessão falhará.

Classificando Dados

Para usar entrada classificada, passe dados classificados pelo Agregador.

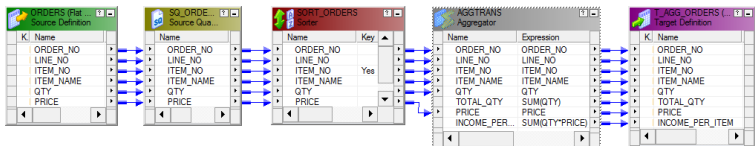
Os dados devem ser classificados das seguintes formas:

- Agrupados pelo Agregador por portas, nas ordem em que aparecem na transformação do Agregador.
- Usando a mesma ordem de classificação para a sessão. Se os dados não estiverem em ordem ascendente ou descendente estrita, com base na ordem de classificação da sessão, a sessão do Serviço de Integração fechará. Por exemplo, se você configurar uma sessão para usar uma ordem de classificação em francês, os dados que passarem pela transformação do Agregador devem estar classificados pela ordem de classificação em francês.

Para origens relacionais e de arquivo, use a transformação Classificador para classificar dados no mapeamento, antes de passá-los à transformação do Agregador. Você poderá colocar a transformação Classificador em qualquer lugar antes do Agregador no mapeamento, se nenhuma transformação alterar a ordem dos dados classificados. O grupo por colunas na transformação do Agregador deve estar na mesma ordem em que aparece na transformação Classificador.

Se a sessão usar origens relacionais, você também poderá usar a opção Número de Portas Classificadas na transformação Qualificador de origem para classificar o grupo por colunas no banco de dados de origem. O grupo por colunas deve ter a mesma ordem nas transformações do Agregador e Qualificador de origem.

O mapeamento a seguir mostra uma transformação Classificador, configurada para classificar os dados de origem ascendente por ITEM_NO:



A transformação classifica os dados como a seguir:

ITEM_NO	ITEM_NAME	QTY	PRICE
345	Soup	4	2.95
345	Soup	1	2.95
345	Soup	2	3.25
546	Cereal	1	4.49
546	Cereal	2	5.25

Com a entrada classificada, o transformação do Agregador retorna estes resultados:

ITEM_NO	ITEM_NAME	TOTAL_QTY	INCOME_PER_ITEM
345	Soup	7	21.25
546	Cereal	3	14.99

Criação de uma Transformação Agregador

Para usar uma transformação Agregador em um mapeamento, adicione a transformação Agregador ao mapeamento. Em seguida, configure a transformação com uma expressão agregada e agrupe por portas.

Para criar uma transformação Agregador:

1. No Mapping Designer, clique em Transformação > Criar. Selecione a transformação Agregador.
2. Insira um nome para o Agregador e clique em Criar. Em seguida, clique em Concluído.
O Designer cria a transformação Agregador.
3. Arraste as portas para a transformação Agregador.
O Designer cria portas de entrada/saída para cada porta que você incluir.
4. Clique duas vezes na barra de título da transformação para abrir a caixa de diálogo Editar transformações.
5. Selecione a guia Portas.
6. Clique na opção agrupar por, em cada coluna que você deseja que o Aggregator use ao criar grupos.
Ou então, insira um valor padrão para substituir grupos nulos.
7. Clique em Adicionar para adicionar uma porta de expressão.
A porta de expressão deve ser uma porta de saída. Transforme a porta em uma porta de saída desmarcando Entrada (I).
8. Como opção, adicione valores padrão para portas específicas.
Se o banco de dados de destino não manipula valores nulos e é provável que certas portas contenham valores nulos, especifique um valor padrão.
9. Configure as propriedades na guia Propriedades.

Dicas para Transformações Agregador

Utilize a entrada classificada para diminuir o uso de caches agregados.

A entrada classificada reduz o volume de dados armazenados no cache durante a sessão e melhora o desempenho da sessão. Utilize esta opção com a transformação Classificador, a fim de passar dados classificados para a transformação Agregador.

A transformação de Agregador pode não fornecer uma saída classificada.

Para classificar a saída de uma transformação de Agregador, use uma transformação de Classificador.

Limite as portas de entrada/saída ou de saída conectadas.

Limite o número de portas de entrada/saída ou de saída conectadas, para reduzir o volume de dados que a transformação Agregador armazena no cache de dados.

Filtre os dados antes de agregá-los.

Se você utilizar uma transformação Filtro no mapeamento, coloque-a antes da transformação Agregador, a fim de reduzir uma agregação desnecessária.

Solução de Problemas de Transformações Agregador

Selecionei uma entrada classificada, mas o fluxo de trabalho assume o mesmo período de tempo que antes.

Você não poderá utilizar uma entrada classificada se qualquer uma das condições a seguir for verdadeira:

- A expressão de agregação contém funções de agregação aninhadas.
- A sessão utiliza uma agregação incremental.
- Os dados de origem são controlados por dados.

Quando uma dessas condições for verdadeira, o Serviço de Integração processará a transformação como se você não usasse uma entrada classificada.

Uma sessão que utiliza uma transformação Agregador provoca um desempenho lento.

Pode ser que o Serviço de Integração esteja fazendo uma paginação no disco durante o fluxo de trabalho. Você pode melhorar o desempenho da sessão aumentando o tamanho do cache de dados e do índice nas propriedades de transformação.

Inseri um diretório de cache de substituição na transformação Agregador, mas o Serviço de Integração salva os arquivos de agregação incremental da sessão em outro local.

Você pode substituir o diretório do cache de transformação em um nível de sessão. O Serviço de Integração registra o diretório de cache no log da sessão. Você também pode verificar as propriedades da sessão para um diretório de cache de substituição.

CAPÍTULO 3

Transformação Personalizada

Este capítulo inclui os seguintes tópicos:

- [Visão Geral de Transformação Personalizada , 61](#)
- [Criando Transformações Personalizadas, 63](#)
- [Trabalhando com Grupos e Portas, 64](#)
- [Trabalhando com Atributos de Porta, 66](#)
- [Propriedades de Transformação Personalizada, 67](#)
- [Trabalhando com o Controle de Transação, 70](#)
- [Bloqueando Dados de Entrada, 71](#)
- [Trabalhando com Propriedades de Procedimento, 73](#)
- [Criando Procedimentos de Transformação Personalizada, 74](#)

Visão Geral de Transformação Personalizada

As transformações Personalizadas funcionam em conjunção com os procedimentos que você cria fora da interface do Designer, para ampliar a funcionalidade do PowerCenter. Você pode criar uma transformação Personalizada e a vincular a um procedimento que você desenvolve usando as funções da transformação Personalizada. A transformação Personalizada pode ser uma transformação ativa ou passiva.

Use a transformação Personalizada para criar aplicativos de transformação, como classificação e agregação, que exigem que todas as linhas de entrada sejam processadas antes de gerar qualquer linha de saída. Para suportar este processo, as funções de entrada e saída ocorrem separadamente nas transformações Personalizadas, comparadas às transformações Procedimento Externo.

O Serviço de Integração transfere os dados de entrada para o procedimento usando uma função de entrada. A função de saída é uma função separada que você deve inserir no código de procedimento, a fim de transferir os dados de saída para o Serviço de Integração. Por outro lado, na transformação Procedimento Externo, uma função de procedimento externo gera a entrada e a saída e seus parâmetros consistem em todas as portas da transformação.

Você também pode usar a transformação Personalizada para criar uma transformação que exige múltiplos grupos de entrada, de saída ou ambos. Um grupo é a representação de uma linha de dados que entram ou saem de uma transformação. Por exemplo, você pode criar uma transformação Personalizada com um grupo de entrada e vários grupos de saída que analisam os dados XML. Ou então, pode criar uma transformação Personalizada com dois grupos de entrada e um de saída que mescla dois fluxos de dados de entrada em um fluxo de dados de saída.

Trabalhando com Transformações Criadas na Transformação Personalizada

Você pode criar transformações usando a transformação Personalizada. Algumas transformações do PowerCenter são criadas usando a transformação Personalizada. As transformações a seguir, enviadas com produtos Informatica, são transformações nativas e *não são criadas* usando a transformação Personalizada:

- Transformação Agregador
- Transformação Expressão
- Transformação Procedimento Externo
- Transformação Filtro
- Transformação Joiner
- Transformação Pesquisa
- Transformação Normalizador
- Transformação Classificação
- Transformação Roteador
- Transformação Gerador de Sequência
- Transformação Classificador
- Transformação Qualificador de Fonte
- Transformação Procedimento Armazenado
- Transformação Controle de Transação
- Transformação Estratégia de Atualização

Todas as outras transformações são criadas com a Transformação personalizada. As regras que se aplicam a transformações Personalizadas, como regras de bloqueio, também se aplicam a transformações criadas com transformações Personalizadas. Por exemplo, ao conectar uma transformação Personalizada a um mapeamento, você deve verificar se os dados podem fluir de todas as origens em um grupo de ordem de carregamento de destino para destinos, sem que o Serviço de Integração bloqueie todas as origens. De modo semelhante, você deve também verificar isso em transformações criadas com uma transformação Personalizada.

Compatibilidade de Página de Código

Quando o Serviço de Integração executa no modo ASCII, ele transfere dados para o procedimento de transformação personalizada no ASCII. Quando o Serviço de Integração executa o modo Unicode, ele transfere dados ao procedimento em UCS-2.

Use as funções `NFA_CTChangeStringMode()` e `INFA_CTSetDataCodePageID()` no código de procedimento de transformação personalizada para solicitar os dados em um formato diferente ou em uma página de código diferente.

As funções que você pode usar dependem do modo de movimento dos dados do Serviço de Integração:

- **Modo ASCII.** Use a função `INFA_CTChangeStringMode()` para solicitar os dados em UCS-2. Quando você usa esta função, o procedimento deve transferir somente caractere ASCII no formato UCS-2 para o Serviço de Integração. Você não pode usar a função `INFA_CTSetDataCodePageID()` para alterar a página de código quando o Serviço de Integração executa no modo ASCII.

- **Modo Unicode.** Use a função `INFA_CTChangeStringMode()` para solicitar os dados em MBCS (conjunto de caracteres multibyte). Quando o procedimento solicita os dados no MBCS, o Serviço de Integração transfere os dados na própria página de código. Use a função `INFA_CTSetDataCodePageID()` para solicitar os dados em uma página de código diferente da página de código do Serviço de Integração. A página de código que você especifica na função `INFA_CTSetDataCodePageID()` deve ter compatibilidade bidirecional com a página de código do Serviço de Integração.

Nota: Você também pode usar a função `INFA_CTRebindInputDataType()` para alterar o formato de uma porta específica na transformação Personalizada.

Distribuindo Procedimentos de Transformação Personalizada

Você pode copiar uma transformação Personalizada de um repositório para outro. Quando você copia uma transformação Personalizada entre repositórios, deve verificar se a máquina do Serviço de Integração usada pelo repositório de destino contém o procedimento de transformação Personalizada.

Criando Transformações Personalizadas

Você pode criar transformações Personalizadas no Transformation Developer, e adicionar instâncias da transformação ao mapeamento. Você pode criar transformações Personalizadas não reutilizáveis no Mapping Designer ou no Mapplet Designer.

Cada transformação Personalizada especifica um módulo e o nome de um procedimento. Você pode criar uma transformação Personalizada com base em uma biblioteca compartilhada existente ou DLL que contenha o procedimento, ou então pode criar a transformação Personalizada como base para a criação do procedimento. Quando você cria uma transformação Personalizada para usar com uma biblioteca compartilhada existente ou DLL, certifique-se de definir o módulo e o nome do procedimento correto.

Quando você criar uma transformação Personalizada como base para a criação do procedimento, selecione a transformação e gere o código. O Designer usa as propriedades de transformação quando gera o código de procedimento. Ele gera código em um único diretório para todas as transformações que compartilham um nome de módulo comum.

O Designer gera os arquivos a seguir:

- **m_<module_name>.c.** Define o módulo. Este arquivo inclui uma função de inicialização, `m_<module_name>_moduleInit()`, que lhe permite escrever o código que você deseja que o Serviço de Integração execute ao carregar o módulo. Da mesma forma, este arquivo inclui uma função de desinicialização, `m_<module_name>_moduleDeinit()`, que lhe permite escrever o código que você deseja que o Serviço de Integração execute antes de descarregar o módulo.
- **p_<procedure_name>.c.** Define o procedimento no módulo. Esse arquivo contém o código que implementa a lógica do procedimento, como a limpeza de dados ou dados de mesclagem.
- **makefile.aix, makefile.aix64, makefile.hpparisc64, makefile.linux, makefile.sol, and makefile.sol64.** Cria arquivos para plataformas UNIX, exceto zLinux. Use `makefile.aix64` para plataformas AIX de 64 bits e `makefile.sol64` para plataformas Solaris de 64 bits.

Nota: Para o zLinux, você precisa atualizar o `makefile.linux`. Adicionar `-m64` à seção `FLAGS`. Por exemplo, `FLAGS=-Wall -fPIC -DUNIX -m64`.

Regras e Diretrizes para Transformações personalizadas

Utilize as seguintes regras e diretrizes ao criar uma Transformação personalizada:

- As transformações personalizadas são transformações conectadas. Você não pode fazer referência a uma transformação personalizada em uma expressão.
- Você pode incluir vários procedimentos em um módulo. Por exemplo, você pode incluir um procedimento de gravador XML e um procedimento de analisador XML no mesmo módulo.
- Você poderá vincular uma DLL ou uma biblioteca compartilhada a várias instâncias da transformação personalizada se gravar o código de procedimento para tratar de várias instâncias da transformação personalizada.
- Quando grava o código do procedimento, você precisa certificar-se de que ele não viole regras básicas de mapeamento.
- A transformação personalizada envia e recebe decimais de alta precisão como decimais de alta precisão.
- Utilize um código multithread nos procedimentos de transformação personalizada.

Componentes de Transformações Personalizadas

Ao configurar uma Transformação personalizada, você define os seguintes componentes:

- **Guia Transformação.** Você pode renomear a transformação e adicionar uma descrição na guia Transformação.
- **Guia Portas.** Você pode adicionar e editar portas e grupos para uma transformação Personalizada. Você também pode definir as portas de entrada das quais depende uma porta de saída.
- **Guia Definições de Atributo de Porta.** Você pode criar atributos de porta definidos pelo usuário para as portas da transformação Personalizada.
- **Guia Propriedades.** Você pode definir propriedades de transformação, como módulo e identificadores de função, propriedades de transação e o local do tempo de execução.
- **Guia Propriedades de Inicialização.** Você pode definir as propriedades que o procedimento externo usa no tempo de execução, como durante a inicialização.
- **Guia Extensões de Metadados.** Você pode criar extensões de metadados para definir as propriedades que o procedimento usa no tempo de execução, como durante inicialização.

Trabalhando com Grupos e Portas

Uma transformação personalizada tem grupos de entrada e saída. Ela também pode ter portas de entrada, portas de saída e portas de entrada/saída. Você cria e edita grupos e portas na guia Portas da transformação personalizada. Você também pode definir a relação entre as portas de entrada e saída na guia Portas.

Criando Grupos e Portas

Você pode criar múltiplos grupos de entrada e saída em uma transformação Personalizada. Você deve criar pelo menos um grupo de entrada e um de saída. Para criar um grupo de entrada, clique no ícone Criar Grupo de Entrada. Para criar um grupo de saída, clique no ícone Criar Grupo de Saída. É possível alterar nomes de grupo existentes digitando no cabeçalho do grupo. Quando você cria uma transformação Personalizada passiva, pode criar somente um grupo de entrada e um de saída.

Ao criar uma porta, o Designer a adiciona abaixo da linha ou grupo que estiver selecionado. Uma porta pode pertencer ao grupo de entrada e ao grupo de saída que aparece imediatamente acima dela. Qualquer porta de entrada/saída que aparecer abaixo do grupo de entrada também faz parte do grupo. Qualquer porta de entrada/saída que aparecer abaixo do grupo de saída também faz parte do grupo de entrada.

Grupos que compartilham portas são chamados de um grupo acoplado. Grupos adjacentes do tipo oposto podem compartilhar portas. Um grupo pode fazer parte de mais de um grupo acoplado. Por exemplo, na figura de [“Etapa 1. Criar a transformação personalizada” na página 74](#), InputGroup1 e OutputGroup1 é um grupo acoplado que compartilha ORDER_ID1

Se a transformação possui uma guia Definições do Atributo de Porta, você pode editar os atributos para cada porta.

Editando Grupos e Portas

Use as seguintes regras e diretrizes ao editar portas e grupos em uma Transformação personalizada:

- É possível alterar nomes de grupos digitando no cabeçalho do grupo.
- Somente é possível inserir caracteres ASCII para nome do grupo e de portas.
- Depois que você criar um grupo, não será possível alterar o tipo de grupo. Caso seja necessário alterar o tipo de grupo, exclua o grupo e adicione um novo grupo.
- Ao excluir um grupo, o Designer exclui todas as portas do mesmo tipo nesse grupo. Entretanto, todas as portas de entrada/saída permanecem na transformação, pertencem ao grupo que está acima delas e são alteradas para as portas de entrada ou portas de saída, dependendo do tipo de grupo que você exclui. Por exemplo, um grupo de saída contém portas de saída e portas de entrada/saída. Você exclui o grupo de saída. O Designer exclui as portas de saída. Ele muda as portas de entrada/saída para portas de entrada. Essas portas de entrada pertencem ao grupo de entrada com o cabeçalho diretamente acima deles.
- Para mover um grupo para cima ou para baixo, selecione o cabeçalho do grupo e clique no botão Mover Porta para Cima ou Mover Porta para Baixo. As portas acima e abaixo do cabeçalho do grupo permanecem as mesmas, mas os grupos aos quais pertencem podem mudar.
- Para criar uma porta de entrada/saída, a transformação deve ter um grupo de entrada e um grupo de saída.

Definindo Relações de Portas

Por padrão, uma porta de saída em uma transformação Personalizada depende de todas as portas de entrada. No entanto, você pode definir a relação entre as portas de entrada e saída em uma transformação Personalizada. Quando você faz isto, pode exibir caminhos de link em um mapeamento que contém uma transformação Personalizada, e pode ver de quais portas de entrada uma porta de saída depende. Você também pode exibir as dependências da coluna de origem para as portas de destino em um mapeamento que contém uma transformação Personalizada.

Para definir a relação entre portas em uma transformação Personalizada, crie uma dependência de porta. Uma dependência de porta é a relação entre uma porta de saída ou entrada/saída e uma ou mais portas de entrada ou entrada/saída. Quando você cria uma dependência de porta, deve baseá-la na lógica de procedimento no código.

Para criar uma dependência de porta, clique em Transformação Personalizada na guia Portas e escolha Dependências de Porta.

Por exemplo, crie um procedimento externo que analise dados XML. Você criará uma transformação Personalizada com um grupo de entrada que contém uma porta de entrada, e múltiplos grupos de saída contendo múltiplas portas de saída. De acordo com a lógica do procedimento externo, todas as portas de

saída dependem da porta de entrada. Você pode definir esta relação na transformação Personalizada, criando uma dependência de porta para cada porta de saída. Defina cada dependência de porta, de forma que a porta de saída dependa de uma porta de entrada.

Para criar uma dependência de porta:

1. Na guia Portas, clique em Transformação Personalizada e escolha Dependências de Porta.
2. Na caixa de diálogo Dependências da Porta de Saída, selecione uma porta de saída ou entrada/saída no campo Porta de Saída.
3. No painel Portas de Entrada, selecione uma porta de entrada ou entrada/saída da qual a porta de saída ou a de entrada/saída depende.
4. Clique em Adicionar.
5. Repita as etapas [3](#) a [4](#) para incluir mais portas de entrada ou entrada/saída na dependência de porta.
6. Para criar outra dependência de porta, repita as etapas [2](#) a [5](#).
7. Clique em OK.

Trabalhando com Atributos de Porta

As portas possuem atributos, como tipo de dados e precisão. Ao criar uma transformação personalizada, você pode criar atributos de porta definidos pelo usuário. Os atributos de porta definidos pelo usuário aplicam-se a todas as portas em uma transformação personalizada.

Por exemplo, você pode criar um procedimento externo para analisar dados XML. Você pode criar um atributo de porta denominado “XML path”, em que é possível definir a posição de um elemento na hierarquia XML.

Crie atributos de porta e atribua valores padrão na guia Definições de Atributos de Porta da transformação personalizada. Você pode definir um valor de atributo de porta específico para cada porta na guia Portas.

Ao criar um atributo de porta, defina as seguintes propriedades:

- **Nome.** O nome do atributo de porta.
- **Tipo de dados.** O tipo de dados do valor do atributo de porta. Você pode escolher Booleano, Numérico ou String.
- **Valor.** O valor padrão do atributo de porta. Essa propriedade é opcional. Quando você insere um valor aqui, ele se aplica a todas as portas na transformação personalizada. Você pode substituir o valor do atributo de cada porta na guia Portas.

Você define atributos de porta para cada transformação personalizada. Você não pode copiar um atributo de porta de uma transformação personalizada para outra.

Editando Valores de Atributo de Porta

Depois de criar atributos de porta, é possível editar os valores de atributo de porta de cada porta na transformação. Para editar os valores de atributo de porta, clique em Transformação Personalizada na guia Portas e escolha Editar Atributo de Porta.

É possível alterar o valor de atributo de porta de uma determinada porta, clicando no botão Abrir. Com isso, você abrirá a caixa de diálogo Valor Padrão para Editar Atributos de Porta. Outra opção é inserir um novo valor, digitando diretamente na coluna Valor.

É possível filtrar as portas listadas na caixa de diálogo Editar Atributos de Nível de Porta, escolhendo um grupo no campo Selecionar Grupo.

Propriedades de Transformação Personalizada

As propriedades da transformação Personalizada se aplicam ao procedimento e à transformação. Configure as Propriedades de Transformação Personalizada na guia Propriedades da Transformação Personalizada.

A tabela a seguir descreve as Propriedades de Transformação Personalizada:

Opção	Descrição
Linguagem	Linguagem usada para o código de procedimento. Você define a linguagem quando cria a transformação Personalizada. Se você precisar alterar a linguagem, crie uma nova transformação Personalizada.
Identificador do Módulo	Nome do módulo. Aplica-se aos procedimentos de transformação Personalizada desenvolvidos usando C ou C++. Insira somente caracteres ASCII neste campo. Você não pode inserir caracteres multibyte. Esta propriedade é o nome básico da DLL ou biblioteca compartilhada que contém o procedimento. O Designer usa este nome para criar o arquivo C quando você gera o código de procedimento externo.
Identificador da função	Nome do procedimento no módulo. Aplica-se aos procedimentos de Transformação personalizada desenvolvidos usando C. Insira somente caracteres ASCII neste campo. Você não pode inserir caracteres multibyte. O Designer usa esse nome para criar o arquivo C onde você insere o código de procedimento.
Nome de Classe	Nome da classe do procedimento de transformação personalizada. Aplica-se aos procedimentos de Transformação personalizada desenvolvidos usando C++ ou Java. Insira somente caracteres ASCII neste campo. Você não pode inserir caracteres multibyte.
Local do Tempo de Execução	Local que contém a DLL ou a biblioteca compartilhada. O padrão é \$PMExtProcDir. Insira um caminho relativo ao nó do Serviço de Integração que executa a sessão de Transformação personalizada. Se esta propriedade estiver em branco, o Serviço de Integração usará a variável de ambiente definida no nó do Serviço de Integração para localizar a DLL ou a biblioteca compartilhada. Copie todas as DLLs ou bibliotecas compartilhadas para o local do tempo de execução ou para a variável de ambiente definida no nó do Serviço de Integração. O Serviço de Integração não carrega o procedimento quando não consegue localizar a DLL, a biblioteca compartilhada ou um arquivo especificado.
Nível de rastreamento	Quantidade de detalhes exibidos no log de sessão dessa transformação. O padrão é Normal.

Opção	Descrição
É Particionável	<p>Indica se você pode criar múltiplas partições em um pipeline, usando essa transformação:</p> <ul style="list-style-type: none"> - Não. A transformação não pode ser particionada. A transformação e outras transformações no mesmo pipeline estão limitadas a uma partição. - Localmente. A transformação pode ser particionada, mas o Serviço de Integração deve executar todas as partições da pipeline no mesmo nó. Escolha Local quando diferentes partições da Transformação personalizada devem compartilhar objetos na memória. - Na Grade. A transformação pode ser particionada e o Serviço de Integração pode distribuir cada partição para nós diferentes. <p>O padrão é Não.</p>
Entradas Devem Bloquear	Indica se o procedimento associado à transformação deve ser capaz de bloquear os dados de entrada. O padrão é ativado.
Está Ativo	<p>Indica se essa transformação é uma transformação ativa ou passiva.</p> <p>Não é possível alterar essa propriedade depois de criar a Transformação personalizada. Se você precisar alterar esta propriedade, crie uma nova transformação Personalizada e selecione o valor de propriedade correto.</p>
Transformação Estratégia de Atualização	Indica se essa transformação define a estratégia de atualização para linhas de saída. O padrão é desativado. Você pode ativar essa opção para transformações Personalizadas ativas.
Escopo de Transformação	<p>Indica como o Serviço de Integração aplica a lógica de transformação aos dados de entrada:</p> <ul style="list-style-type: none"> - Linha - Transação - Todas as Entradas <p>Quando a transformação é passiva, esta propriedade sempre é Linha. Quando a transformação é ativa, esta propriedade é Todas as Entradas, por padrão.</p>
Gerar Transação	<p>Indica se esta transformação pode gerar transações. Quando uma transformação Personalizada gera transações, ela as gera para todos os grupos de saída.</p> <p>O padrão é desativado. Você só pode ativar essa opção para Transformações personalizadas ativas.</p>
Saída é repetível	<p>Indica se a <i>ordem</i> dos dados de saída é consistente entre as execuções de sessão.</p> <ul style="list-style-type: none"> - Nunca. A ordem dos dados de saída é inconsistente entre as execuções de sessão. Esse é o padrão para transformações ativas. - Baseada na Ordem de Entrada. A ordem de saída é consistente entre as execuções de sessão quando a ordem dos dados de entrada é consistente entre elas. Esse é o padrão para transformações passivas. - Sempre. A ordem dos dados de saída será consistente entre as execuções de sessão mesmo se a ordem de entrada for inconsistente entre elas.
Requer Thread Único por Partição	Indica se o Serviço de Integração processa cada partição em um procedimento com um thread. Quando você ativa esta opção, o código do procedimento pode usar operações específicas da linha. O padrão é ativado.
Saída é determinística	Indica se a transformação gera dados de saída consistentes entre as execuções de sessão. Ative esta propriedade para executar a recuperação em sessões que usam essa transformação.

Aviso: Se você configurar uma transformação como repetível e determinística, será sua responsabilidade garantir que os dados sejam repetíveis e determinísticos. Se você tentar recuperar uma sessão com transformações que não gerem os mesmos dados entre a sessão e a recuperação, o processo de recuperação poderá resultar em dados corrompidos.

Definindo a Estratégia de Atualização

Use uma transformação Personalizada ativa para definir a estratégia de atualização de mapeamentos nestes níveis:

- **Dentro do procedimento.** Você pode gravar o código de procedimento externo para definir a estratégia de atualização para linhas de saída. O procedimento externo pode sinalizar linhas para inserir, atualiza, excluir ou rejeitar.
- **Dentro do mapeamento.** Use a Transformação personalizada em um mapeamento para sinalizar linhas para inserção, atualização, exclusão ou rejeição. Selecione a propriedade da transformação Estratégia de Atualização para a transformação Personalizada.
- **Dentro da sessão.** Configure a sessão para tornar as linhas de origem controladas por dados.

Se você não configurar a transformação Personalizada para definir a estratégia de atualização ou não configurar a sessão como controlada por dados, o Serviço de Integração não usará o código de procedimento externo para sinalizar as linhas de saída. Em vez disso, se a transformação Personalizada for ativa, o Serviço de Integração sinalizará as linhas de saída para inserir. Se a transformação Personalizada for passiva, o Serviço de Integração manterá o tipo de linha. Por exemplo, quando uma linha sinalizada para atualizar entra em uma transformação Personalizada passiva, o Serviço de Integração mantém o tipo de linha e retorna a linha como atualizada.

Trabalhando com Código de Procedimento Específico do Thread

Procedimentos da transformação personalizada podem incluir operações específicas do thread. Uma operação específica do thread é um código que executa uma ação com base no thread que está processando o procedimento.

Você pode configurar a transformação personalizada, de modo que o Serviço de Integração utilize um thread para processar a transformação personalizada de cada partição, usando a propriedade Exige um Único Thread por Partição.

Quando você configura uma transformação personalizada para processar cada partição com um thread, o Serviço de Integração chama as seguintes funções com o mesmo thread para cada partição:

- `p_<proc_name>_partitionInit()`
- `p_<proc_name>_partitionDeinit()`
- `p_<proc_name>_inputRowNotification()`
- `p_<proc_name>_dataBdryRowNotification()`
- `p_<proc_name>_eofNotification()`

Você pode incluir operações específicas do thread nessas funções, porque o Serviço de Integração utiliza o mesmo thread para processar essas funções para cada partição. Por exemplo, você poderia anexar e desanexar threads para uma Java Virtual Machine.

Nota: Quando você configura uma transformação personalizada para processar cada partição com um thread, o Workflow Manager adiciona pontos da partição, dependendo da configuração de mapeamento.

Trabalhando com o Controle de Transação

Você pode definir um controle de transação para transformações personalizadas usando as seguintes propriedades de transformação:

- **Escopo de Transformação.** Determina como o Serviço de Integração aplica a lógica de transformação aos dados de entrada.
- **Gerar Transação.** Indica que o procedimento gera linhas de transação e envia-as para os grupos de saída.

Escopo de Transformação

Você pode configurar a maneira como o Serviço de Integração aplica a lógica de transformação aos dados de entrada. Você pode escolher um destes valores:

- **Linha** Aplica a lógica de transformação a uma linha de dados de cada vez. Escolha Linha se o resultado do procedimento depender de uma única linha de dados. Por exemplo, você pode escolher Linha quando um procedimento analisa uma linha que contém um arquivo XML.
- **Transação.** Aplica a lógica de transformação a todas as linhas de uma transação. Escolha Transação se os resultados do procedimento dependerem de todas as linhas na mesma transação, mas não de linhas em outras transações. Ao escolher Transação, você deve conectar todos os grupos de entrada ao mesmo ponto de controle de transação. Por exemplo, você pode escolher Transação quando o procedimento externo realizar cálculos agregados sobre dados em uma única transação.
- **Todas as Entradas.** Aplica a lógica de transformação a todos os dados de entrada. Se você escolher Todos de Entrada, o Serviço de Integração descartará limites de transação. Escolha Todos de Entrada se o resultado do procedimento depender de todas as linhas de dados na origem. Por exemplo, você pode escolher Todos de Entrada quando o procedimento externo classifica ou realiza cálculos agregados em todos os dados recebidos.

Gerar Transação

Você pode escrever o código de procedimento externo para as transações de saída, como linhas de confirmação e reversão. Quando o procedimento externo gera como saída linhas de confirmação e reversão, configure a transformação Personalizada para gerar transações. Selecione a propriedade de transformação Gerar Transação. Você pode ativar essa propriedade para transformações Personalizadas ativas.

Quando o procedimento externo gera como saída uma linha de confirmação ou reversão, ele gera ou reverte a linha para todos os grupos de saída.

Ao configurar a transformação para gerar transações, o Serviço de Integração trata a Transformação personalizada como uma Transformação Controle de Transação. A maioria das regras que se aplicam a uma Transformação Controle de Transação em um mapeamento também se aplicam à Transformação personalizada. Por exemplo, ao configurar uma Transformação personalizada para gerar transações, você não pode concatenar os pipelines ou ramificações de pipelines que contêm a transformação.

Ao editar ou criar uma sessão usando uma Transformação personalizada configurada para gerar transações, configure-a para a confirmação definida pelo usuário.

Trabalhando com Limites da Transação

O Serviço de Integração trata dos limites da transação entrando e saindo das transformações personalizadas, com base na configuração de mapeamento e nas propriedades dessas transformações.

A tabela a seguir mostra como o Serviço de Integração trata dos limites da transação em transformações personalizadas:

Escopo de Transformação	Opção Gerar Transações Habilitada	Operação Gerar Transações Desativada
Linha	<p>O Serviço de Integração descarta os limites da transação de entrada e não chama a função de notificação de limites de dados.</p> <p>Ele emite linhas de transação de acordo com a lógica do procedimento através de todos os grupos de saída.</p>	<p>Quando os dados recebidos de todos os grupos de entrada chegam do mesmo ponto de controle de transação, o Serviço de Integração preserva os limites da transação de entrada e os envia por todos os grupos de saída. No entanto, ele não chama a função de notificação de limites de dados.</p> <p>Quando dados de entrada dos grupos de entrada vêm de diferentes pontos de controle de transação, o Serviço de Integração descarta limites de transação de entrada. Ele não chama a função de notificação de limite de dados. O Serviço de Integração emite todas as linhas em uma transação aberta.</p>
Transação	<p>O Serviço de Integração preserva os limites da transação de entrada e chama a função de notificação de limites de dados.</p> <p>No entanto, ele emite linhas de transação de acordo com a lógica do procedimento em todos os grupos de saída.</p>	<p>O Serviço de Integração preserva os limites da transação de entrada e chama a função de notificação de limites de dados.</p> <p>Ele emite linhas de transação em todos os grupos de saída.</p>
Todas as Entradas	<p>O Serviço de Integração descarta os limites da transação de entrada e não chama a função de notificação de limites de dados. O Serviço de Integração emite linhas de transação de acordo com a lógica do procedimento em todos os grupos de saída.</p>	<p>O Serviço de Integração descarta os limites da transação de entrada e não chama a função de notificação de limites de dados. Ele emite todas as linhas em uma transação aberta.</p>

Bloqueando Dados de Entrada

Por padrão, o Serviço de Integração lê ao mesmo tempo as origens em um grupos de ordem de carregamento de destino. Entretanto, é possível escrever o código de procedimento externo para bloquear dados de entrada em determinados grupos de entrada. O bloqueio é a suspensão do fluxo de dados dentro de um grupo de entrada de uma transformação de vários grupos de entrada.

Para usar uma Transformação personalizada para bloquear os dados de entrada, é necessário escrever o código de procedimento para bloquear e desbloquear dados. Também é necessário ativar o bloqueio na guia Propriedades para a Transformação personalizada.

Gravando o Código do Procedimento para Bloquear Dados

Você pode gravar o procedimento para bloquear e desbloquear dados de entrada. Para bloquear dados de entrada, utilize a função `INFA_CTBlockInputFlow()`. Para desbloquear dados de entrada, use a função `INFA_CTUnblockInputFlow()`.

Talvez você queira bloquear dados de entrada se o procedimento externo precisar alternar a leitura de grupos de entrada. Sem a funcionalidade de bloqueio, você precisaria gravar o código do procedimento para armazenar dados de entrada no buffer. Você pode bloquear dados de entrada, em vez de armazená-los no buffer, o que normalmente melhora o desempenho da sessão.

Por exemplo, você precisa criar um procedimento externo com dois grupos de entrada. O procedimento externo lê uma linha do primeiro grupo de entrada e, em seguida, lê uma linha do segundo grupo de entrada. Se você usar o bloqueio, poderá gravar o código do procedimento externo para bloquear o fluxo de dados de um grupo de entrada enquanto processa os dados do outro grupo de entrada. Ao gravar o código do procedimento externo para bloquear dados, você melhora o desempenho porque o procedimento não precisa copiar os dados de origem em um buffer. No entanto, você poderia gravar o procedimento externo para alocar um buffer e copiar os dados de um grupo de entrada no buffer até ele estar pronto para processar os dados. A cópia de dados de origem em um buffer prejudica o desempenho.

TÓPICOS RELACIONADOS:

- [“Funções de Bloqueio” na página 115](#)

Configurando Transformações Personalizadas como Transformações de Bloqueio

Quando você cria uma transformação Personalizada, o Designer ativa a propriedade da transformação Entradas Devem Bloquear por padrão. Esta propriedade afeta a validação do fluxo de dados quando você salva ou valida um mapeamento. Quando você ativa esta propriedade, a transformação Personalizada é uma transformação de bloqueio. Quando você desmarca esta propriedade, a Transformação personalizada não é uma transformação de bloqueio.

Configure a transformação Personalizada como uma transformação de bloqueio, quando o código de procedimento externo deve ser capaz de bloquear os dados de entrada.

Você pode configurar a transformação Personalizada como uma transformação que não seja de bloqueio, quando uma das seguintes condições é verdadeira:

- O código de procedimento não inclui as funções de bloqueio.
- O código de procedimento inclui dois algoritmos, um que usa bloqueios e o outro que copia os dados da origem para um buffer alocado pelo procedimento, em vez de bloquear os dados. O código verifica se o Serviço de Integração permite ou não que a transformação Personalizada bloqueie os dados. O procedimento usa o algoritmo com as funções de bloqueio quando ele pode bloquear, e usa o outro algoritmo quando não pode bloquear. Você pode fazer isto para criar uma transformação Personalizada que será usada em múltiplas configurações de mapeamento.

Nota: Quando o procedimento bloqueia os dados e você configura a transformação Personalizada como uma transformação sem bloqueio, o Serviço de Integração provoca uma falha na sessão.

Validando Mapeamentos com Transformações Personalizadas

Quando você inclui uma transformação Personalizada em um mapeamento, o Designer e o Serviço de Integração validam o mapeamento. O Designer valida o mapeamento salvo por você ou valida e o Serviço de Integração valida o mapeamento quando você executa a sessão.

Validação no Tempo de Design

Quando você salva ou valida um mapeamento, o Designer realiza uma validação do fluxo de dados. Nessa validação, o Designer verifica se os dados podem fluir de todas as origens em um grupo de classificação de carregamento de destino até os destinos, sem bloquear transformações bloquear *todas* as origens. Alguns mapeamentos com transformações de bloqueio são inválidos.

Validando no Tempo de Execução

Quando você executa uma sessão, o Serviço de Integração valida o mapeamento em relação ao código de procedimento no tempo de execução. Quando o Serviço de Integração faz essa validação, ele determina se permitirá ou não que as transformações personalizadas bloqueiem dados:

- **Configure a transformação personalizada como uma transformação de bloqueio.** O Serviço de Integração sempre permite que a transformação personalizada bloqueie dados.
- **Configure a Transformação personalizada como uma transformação sem bloqueio.** O Serviço de Integração permite que a transformação personalizada bloqueie dados, dependendo da configuração do mapeamento. Se o Serviço de Integração puder bloquear dados na transformação personalizada sem bloquear todas as origens no grupo de classificação para carregamento de destino simultaneamente, ele permitirá que a transformação personalizada bloqueie dados.

Você pode gravar o código de procedimento para verificar se o Serviço de Integração permitirá que uma transformação personalizada bloqueie dados. Utilize a função `INFA_CT_getInternalProperty()` para acessar a ID da propriedade `INFA_CT_TRANS_MAY_BLOCK_DATA`. O Serviço de Integração retorna `VERDADEIRO` quando a transformação personalizada pode bloquear dados e `FALSO` quando ela não pode bloqueá-los.

Trabalhando com Propriedades de Procedimento

Você pode definir pares de valor e nome da propriedade na transformação personalizada que o procedimento possa usar quando o Serviço de Integração executá-lo, por exemplo, durante o tempo de inicialização. Você pode criar propriedades definidas pelo usuário nas seguintes guias da Transformação personalizada:

- **Extensões de Metadados.** Você pode especificar nome, tipo de dados, precisão e valor da propriedade. Use extensões de metadados para passar informações ao procedimento.
- **Propriedades de Inicialização.** Você pode especificar o nome e o valor da propriedade.

Você pode definir propriedades em ambas as guias da transformação personalizada, mas a guia Extensões de Metadados proporciona mais detalhes à propriedade. Use extensões de metadados para passar informações ao procedimento.

Por exemplo, você cria um procedimento externo da transformação personalizada que classifica dados antes de transformá-los. Você pode criar uma extensão de metadados booleana chamada `Classificar_Ascendente`. Ao usar a transformação personalizada em um mapeamento, você pode escolher `Verdadeiro` ou `Falso` para a extensão de metadados, dependendo de como quiser que o procedimento classifique os dados.

Ao definir uma propriedade na transformação personalizada, use as funções `get all property names`, como `INFA_CTGetAllPropertyNamesM()`, para acessar os nomes de todas as propriedades definidas nas guias Propriedades de Inicialização e Extensões de Metadados. Use as funções `get external property`, como `INFA_CT_getExternalPropertyM()`, para acessar o nome de propriedade e valor de uma ID de propriedade especificada.

Nota: Se você definir uma propriedade de inicialização e uma extensão de metadados com o mesmo nome, as funções da propriedades só retornarão informações para a extensão de metadados.

Criando Procedimentos de Transformação Personalizada

Você pode criar procedimentos de Transformação personalizada executados em máquinas do Serviço de Integração de 32 ou 64 bits. Use as seguintes etapas como diretrizes ao criar um procedimento de transformação Personalizada:

1. No Transformation Developer, crie uma Transformação personalizada reutilizável. Ou, no Mapplet Designer ou Mapping Designer, crie uma transformação personalizada não reutilizável.
2. Gere o código modelo para o procedimento.
Quando você gera o código de procedimento, o Designer usa as informações da transformação Personalizada para criar os arquivos e makefiles do código de origem C.
3. Modifique os arquivos C para adicionar a lógica de procedimento.
4. Utilize um compilador do C/C++ para compilar e vincular os arquivos do código de origem em uma biblioteca de DLL ou compartilhada e copiá-la para a máquina do Serviço de Integração.
5. Crie um mapeamento com a Transformação personalizada.
6. Execute a sessão em um fluxo de trabalho.

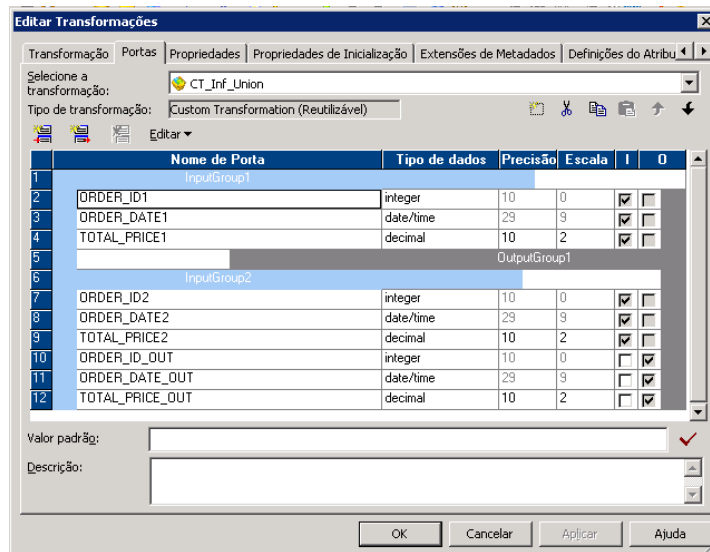
Esta seção inclui um exemplo para demonstrar este processo. As etapas desta seção criam uma transformação Personalizada que contém dois grupos de entrada e um de saída. O procedimento de transformação Personalizada verifica se a transformação Personalizada usa dois grupos de entrada e um de saída. Também verifica se o número de portas em todos os grupos é igual e se os tipos de dados da porta são os mesmos para todos os grupos. O procedimento captura linhas de dados de cada grupo de entrada e produz uma saída de todas as linhas para o grupo de saída.

Etapa 1. Criar a transformação personalizada

Para criar uma transformação personalizada:

1. No Transformation Developer, clique em Transformação > Criar.
2. Na caixa de diálogo Criar Transformação, selecione Transformação personalizada, digite um nome para a transformação e clique em Criar.
No exemplo União, digite `CT_Inf_Union` como nome da transformação.
3. Na caixa de diálogo Ativa ou Passiva, crie a transformação como passiva ou ativa e clique em OK.
No exemplo União, selecione Ativa.
4. Clique em Concluído para fechar a caixa de diálogo Criar Transformação.
5. Abra a transformação e clique na guia Portas. Crie grupos e portas.
Você pode editar os grupos e portas mais tarde, se necessário.

A figura a seguir mostra um exemplo de uma transformação de união com dois grupos:



No exemplo União, crie os grupos InputGroup1 e Inputgroup2. Crie as seguintes portas para InputGroup1:

Nome da Porta	Tipo de dados	Precisão	Escala
ORDER_ID1	integer	10	0
ORDER_DATE1	data/hora	19	0
TOTAL_PRICE1	decimal	10	2

Crie as seguintes portas para Inputgroup2:

Nome da Porta	Tipo de dados	Precisão	Escala	Entrada/Saída
ORDER_ID2	integer	10	0	Entrada
ORDER_DATA2	data/hora	19	0	Entrada
TOTAL_PRICE2	decimal	10	2	Entrada
ORDER_ID_OUT	integer	10	0	Saída
ORDER_DATE_OUT	data/hora	19	0	Saída
TOTAL_PRICE_OUT	decimal	10	2	Saída

- Selecione a guia Propriedades e insira um identificador de módulo e função e a localização do tempo de execução. Edite os outros atributos da propriedade de transformação, como os valores e caixas de seleção: Nível de Rastreamento, Se É Particionável, Se as Entradas Devem Bloquear, Se Está Ativo, a Transformação de Estratégia de Atualização, o Escopo de Transformação e Gerar Transação.

No exemplo União, configure as seguintes propriedades:

Nome da Propriedade	Valor
Module Identifier	UnionDemo
Function Identifier	União
Localização de Tempo de Execução	\$PMEExtProcDir
Nível de Rastreamento	Normal
É Particionável	Não
Entrada Deve Bloquear	Não
Está Ativo	Sim
Transformação de Estratégia de Atualização	Não
Escopo de Transformação	Todas as entradas
Gerar Transação	Não

7. Clique na guia Extensões de Metadados para inserir extensões de metadados, como propriedades de que o procedimento possa precisar para a inicialização.
No exemplo União, não crie extensões de metadados.
8. Clique na guia Definições de Atributos da Porta para criar atributos de porta, se necessário.
No exemplo União, não crie atributos de porta.

Depois de criar a Transformação personalizada que chama o procedimento, a próxima etapa é gerar os arquivos C.

Etapa 2. Gerar os Arquivos C

Depois de criar uma transformação Personalizada, você gera os arquivos de código de origem. O Designer gera nomes de arquivos em minúsculas.

Para gerar o código de procedimento de transformação personalizada:

1. No Transformation Developer, selecione a transformação e clique em Transformação > Gerar Código.
2. Selecione o procedimento recém-criado. O Designer lista os procedimentos como <nome_módulo>.<nome_procedimento>.

No exemplo União, selecione UnionDemo.Union.

3. Especifique o diretório em que deseja gerar os arquivos e clique em Gerar.

No exemplo União, selecione <diretório_instalação_cliente>/TX.

O Designer cria um subdiretório, <module_name>, no diretório especificado. No exemplo União, o Designer cria <diretório_instalação_cliente>/TX/UnionDemo. Cria também os seguintes arquivos:

- m_UnionDemo.c
- m_UnionDemo.h

- p_Union.c
- p_Union.h
- makefile.aix (32 bits), makefile.aix64 (64 bits), makefile.hp (32 bits), makefile.hp64 (64 bits), makefile.hpparisc64, makefile.linux (32 bits), e makefile.sol (32 bits).

Etapa 3. Preencher o código com a lógica de transformação

Você deve codificar o arquivo C do procedimento. Você pode também codificar o arquivo C do módulo. No exemplo União, preencha apenas o arquivo C do procedimento. Não é necessário preencher o arquivo C do módulo.

Para codificar o arquivo de procedimento C:

1. Abra p_<procedure_name>.c para o procedimento.

No exemplo União, abra p_Union.c.

2. Insira o código C do procedimento.

3. Salve o arquivo modificado.

No exemplo União, use este código:

```

/*****
 * Custom Transformation p_union Procedure File
 *
 * This file contains code that functions that will be called by the main
 * server executable.
 *
 * for more information on these files,
 * see $(INFA_HOME)/ExtProc/include/Readme.txt
 *****/

/*
 * INFORMATICA 'UNION DEMO' developed using the API for custom
 * transformations.

 * File Name: p_Union.c
 *
 * An example of a custom transformation ('Union') using PowerCenter
 *
 * The purpose of the 'Union' transformation is to combine pipelines with the
 * same row definition into one pipeline (i.e. union of multiple pipelines).
 * [ Note that it does not correspond to the mathematical definition of union
 * since it does not eliminate duplicate rows.]
 *
 * This example union transformation allows N input pipelines ( each
 * corresponding to an input group) to be combined into one pipeline.
 *
 * To use this transformation in a mapping, the following attributes must be
 * true:
 * a. The transformation must have >= 2 input groups and only one output group.
 * b. In the Properties tab set the following properties:
 *     i. Module Identifier: UnionDemo
 *     ii. Function Identifier: Union
 *     iii. Inputs May Block: Unchecked
 *     iv. Is Active: Checked
 *     v. Update Strategy Transformation: Unchecked *
 *     vi. Transformation Scope: All
 *     vii. Generate Transaction: Unchecked *
 *
 * * This version of the union transformation does not provide code for
 * * changing the update strategy or for generating transactions.
 * c. The input groups and the output group must have the same number of ports
 * and the same datatypes. This is verified in the initialization of the
 * module and the session is failed if this is not true.

```

```

* d. The transformation can be used in multiple number of times in a Target
*   Load Order Group and can also be contained within multiple partitions.
*/

/*****
                                Includes
*****/

include <stdlib.h>
#include "p_union.h"

/*****
                                Forward Declarations
*****/
INFA_STATUS validateProperties(const INFA_CT_PARTITION_HANDLE* partition);

/*****
                                Functions
*****/

/*****
                                Function: p_union_procInit

Description: Initialization for the procedure. Returns INFA_SUCCESS if
procedure initialization succeeds, else return INFA_FAILURE.

Input: procedure - the handle for the procedure
Output: None
Remarks: This function will get called once for the session at
initialization time. It will be called after the moduleInit function.
*****/

INFA_STATUS p_union_procInit( INFA_CT_PROCEDURE_HANDLE procedure)
{
    const INFA_CT_TRANSFORMATION_HANDLE* transformation = NULL;
    const INFA_CT_PARTITION_HANDLE* partition = NULL;
    size_t nTransformations = 0, nPartitions = 0, i = 0;

    /* Log a message indicating beginning of the procedure initialization */
    INFA_CTLogMessageM( eESL_LOG,
                        "union_demo: Procedure initialization started ..." );

    INFA_CTChangeStringMode( procedure, eASM_MBCS );

    /* Get the transformation handles */
    transformation = INFA_CTGetChildrenHandles( procedure,
                                                &nTransformations,
                                                TRANSFORMATIONTYPE);

    /* For each transformation verify that the 0th partition has the correct
     * properties. This does not need to be done for all partitions since rest
     * of the partitions have the same information */
    for (i = 0; i < nTransformations; i++)
    {
        /* Get the partition handle */
        partition = INFA_CTGetChildrenHandles(transformation[i],
                                                &nPartitions, PARTITIONTYPE );

        if (validateProperties(partition) != INFA_SUCCESS)
        {
            INFA_CTLogMessageM( eESL_ERROR,
                                "union_demo: Failed to validate attributes of "
                                "the transformation");
            return INFA_FAILURE;
        }
    }

    INFA_CTLogMessageM( eESL_LOG,
                        "union_demo: Procedure initialization completed." );
}

```

```

        return INFA_SUCCESS;
    }

/*****
    Function: p_union_procDeinit

    Description: Deinitialization for the procedure. Returns INFA_SUCCESS if
    procedure deinitialization succeeds, else return INFA_FAILURE.

    Input: procedure - the handle for the procedure
    Output: None
    Remarks: This function will get called once for the session at
    deinitialization time. It will be called before the moduleDeinit
    function.
    *****/

INFA_STATUS p_union_procDeinit( INFA_CT_PROCEDURE_HANDLE procedure,
                               INFA_STATUS sessionStatus )
{
    /* Do nothing ... */
    return INFA_SUCCESS;
}

/*****
    Function: p_union_partitionInit

    Description: Initialization for the partition. Returns INFA_SUCCESS if
    partition deinitialization succeeds, else return INFA_FAILURE.

    Input: partition - the handle for the partition
    Output: None
    Remarks: This function will get called once for each partition for each
    transformation in the session.
    *****/

INFA_STATUS p_union_partitionInit( INFA_CT_PARTITION_HANDLE partition )
{
    /* Do nothing ... */
    return INFA_SUCCESS;
}

/*****
    Function: p_union_partitionDeinit

    Description: Deinitialization for the partition. Returns INFA_SUCCESS if
    partition deinitialization succeeds, else return INFA_FAILURE.

    Input: partition - the handle for the partition
    Output: None
    Remarks: This function will get called once for each partition for each
    transformation in the session.
    *****/

INFA_STATUS p_union_partitionDeinit( INFA_CT_PARTITION_HANDLE partition )
{
    /* Do nothing ... */
    return INFA_SUCCESS;
}

/*****
    Function: p_union_inputRowNotification

    Description: Notification that a row needs to be processed for an input
    group in a transformation for the given partition. Returns INFA_ROWSUCCESS
    if the input row was processed successfully, INFA_ROWFAILURE if the input
    row was not processed successfully and INFA_FATALERROR if the input row
    causes the session to fail.

    Input: partition - the handle for the partition for the given row

```

```

        group - the handle for the input group for the given row
Output: None
Remarks: This function is probably where the meat of your code will go,
as it is called for every row that gets sent into your transformation.
*****/

INFA_ROWSTATUS p_union_inputRowNotification( INFA_CT_PARTITION_HANDLE partition,
                                             INFA_CT_INPUTGROUP_HANDLE inputGroup )

{
    const INFA_CT_OUTPUTGROUP_HANDLE* outputGroups = NULL;
    const INFA_CT_INPUTPORT_HANDLE* inputGroupPorts = NULL;
    const INFA_CT_OUTPUTPORT_HANDLE* outputGroupPorts = NULL;
    size_t nNumInputPorts = 0, nNumOutputGroups = 0,
          nNumPortsInOutputGroup = 0, i = 0;

    /* Get the output group port handles */
    outputGroups = INFA_CTGetChildrenHandles(partition,
                                             &nNumOutputGroups,
                                             OUTPUTGROUPTYPE);

    outputGroupPorts = INFA_CTGetChildrenHandles(outputGroups[0],
                                             &nNumPortsInOutputGroup,
                                             OUTPUTPORTTYPE);

    /* Get the input groups port handles */
    inputGroupPorts = INFA_CTGetChildrenHandles(inputGroup,
                                             &nNumInputPorts,
                                             INPUTPORTTYPE);

    /* For the union transformation, on receiving a row of input, we need to
     * output that row on the output group. */
    for (i = 0; i < nNumInputPorts; i++)
    {
        INFA_CTSetData(outputGroupPorts[i],
                      INFA_CTGetDataVoid(inputGroupPorts[i]));

        INFA_CTSetIndicator(outputGroupPorts[i],
                          INFA_CTGetIndicator(inputGroupPorts[i]) );

        INFA_CTSetLength(outputGroupPorts[i],
                        INFA_CTGetLength(inputGroupPorts[i]) );
    }

    /* We know there is only one output group for each partition */
    return INFA_CTOutputNotification(outputGroups[0]);
}

/*****
Function: p_union_eofNotification

Description: Notification that the last row for an input group has already
been seen. Return INFA_FAILURE if the session should fail as a result of
seeing this notification, INFA_SUCCESS otherwise.

Input: partition - the handle for the partition for the notification
       group - the handle for the input group for the notification
Output: None
*****/

INFA_STATUS p_union_eofNotification( INFA_CT_PARTITION_HANDLE partition,
                                     INFA_CT_INPUTGROUP_HANDLE group)
{
    INFA_CTLogMessageM( eESL_LOG,
                      "union_demo: An input group received an EOF notification");

    return INFA_SUCCESS;
}

/*****/

```



```

Function: p_union_dataBdryNotification

Description: Notification that a transaction has ended. The data
boundary type can either be commit or rollback.
Return INFA_FAILURE if the session should fail as a result of
seeing this notification, INFA_SUCCESS otherwise.

Input: partition - the handle for the partition for the notification
       transactionType - commit or rollback
Output: None
*****/

INFA_STATUS p_union_dataBdryNotification ( INFA_CT_PARTITION_HANDLE partition,
                                           INFA_CT_DATABDRY_TYPE transactionType)
{
    /* Do nothing */
    return INFA_SUCCESS;
}

/* Helper functions */

/*****
Function: validateProperties

Description: Validate that the transformation has all properties expected
by a union transformation, such as at least one input group, and only
one output group. Return INFA_FAILURE if the session should fail since the
transformation was invalid, INFA_SUCCESS otherwise.

Input: partition - the handle for the partition
Output: None
*****/

INFA_STATUS validateProperties(const INFA_CT_PARTITION_HANDLE* partition)
{
    const INFA_CT_INPUTGROUP_HANDLE* inputGroups = NULL;
    const INFA_CT_OUTPUTGROUP_HANDLE* outputGroups = NULL;
    size_t nNumInputGroups = 0, nNumOutputGroups = 0;
    const INFA_CT_INPUTPORT_HANDLE** allInputGroupsPorts = NULL;
    const INFA_CT_OUTPUTPORT_HANDLE* outputGroupPorts = NULL;
    size_t nNumPortsInOutputGroup = 0;
    size_t i = 0, nTempNumInputPorts = 0;

    /* Get the input and output group handles */
    inputGroups = INFA_CTGetChildrenHandles(partition[0],
                                           &nNumInputGroups,
                                           INPUTGROUPTYPE);

    outputGroups = INFA_CTGetChildrenHandles(partition[0],
                                           &nNumOutputGroups,
                                           OUTPUTGROUPTYPE);

    /* 1. Number of input groups must be >= 2 and number of output groups must
    * be equal to one. */
    if (nNumInputGroups < 1 || nNumOutputGroups != 1)
    {
        INFA_CTLogMessageM( eESL_ERROR,
                           "UnionDemo: There must be at least two input groups "
                           "and only one output group");
        return INFA_FAILURE;
    }

    /* 2. Verify that the same number of ports are in each group (including
    * output group). */
    outputGroupPorts = INFA_CTGetChildrenHandles(outputGroups[0],
                                           &nNumPortsInOutputGroup,
                                           OUTPUTPORTTYPE);

    /* Allocate an array for all input groups ports */
    allInputGroupsPorts = malloc(sizeof(INFA_CT_INPUTPORT_HANDLE*) *

```

```

        nNumInputGroups);

for (i = 0; i < nNumInputGroups; i++)
{
    allInputGroupsPorts[i] = INFA_CTGetChildrenHandles(inputGroups[i],
                                                         &nTempNumInputPorts,
                                                         INPUTPORTTYPE);

    if ( nNumPortsInOutputGroup != nTempNumInputPorts)
    {
        INFA_CTLogMessageM( eESL_ERROR,
                            "UnionDemo: The number of ports in all input and "
                            "the output group must be the same.");
        return INFA_FAILURE;
    }
}

free(allInputGroupsPorts);

/* 3. Datatypes of ports in input group 1 must match data types of all other
 * groups.
 * TODO:*/

return INFA_SUCCESS;
}

```

Etapa 4. Criar o Módulo

Você pode construir o módulo em uma plataforma Windows ou UNIX.

A seguinte tabela lista os nomes de arquivo de biblioteca de cada plataforma quando você cria o módulo:

Plataforma	Nome do Arquivo de Módulo
Windows	<module_identifier>.dll
AIX	lib<module_identifier>.a
Linux	lib<module_identifier>.so
Solaris	lib<module_identifier>.so

Criando o módulo no Windows

No Windows, use o Microsoft Visual C++ para criar o módulo.

Para criar o módulo no Windows:

1. Inicie o Visual C++.
2. Clique em Arquivo > Novo.
3. Na caixa de diálogo Avançar, clique na guia Propriedades e selecione a opção Biblioteca de link dinâmico do Win32.
4. Insira sua localização.
No exemplo de União, insira <client_installation_directory>/TX/UnionDemo.
5. Digite o nome do projeto.
Você deve usar o nome do módulo especificado para a função personalizada como o nome do projeto.
No exemplo União, insira UnionDemo.

6. Clique em OK.
O Visual C++ cria um assistente para ajudar você a definir os componentes do projeto.
7. No assistente, selecione qualquer projeto de DLL vazio e clique em Concluir. Clique em OK na caixa de diálogo Novo Projeto.
O Visual C++ cria os arquivos do projeto no diretório especificado.
8. Clique em Projeto > Adicionar a Projeto > Arquivos.
9. Navegue até um nível do diretório. Esse diretório contém os arquivos de procedimento que você criou. Selecione todos os arquivos .c e clique em OK.
No exemplo da União, adicione os seguintes arquivos:
 - m_UnionDemo.c
 - p_Union.c
10. Clique em Projeto > Configurações.
11. Clique na guia C/C++ e selecione Pré-processador no campo Categoria.
12. No campo adicional Incluir Diretórios, digite o seguinte caminho e clique em OK:


```
..; <PowerCenter_install_dir>\extproc\include\ct
```
13. Clique em Compilar > Compilar <module_name>.dll ou pressione F7 para compilar o projeto.
O Visual C++ cria o DLL e o coloca no diretório de depuração ou de versão no diretório do projeto.

Criando o Módulo no UNIX

No UNIX, use qualquer compilador C para criar o módulo.

Para criar o módulo no UNIX:

1. Copie todos os arquivos C e makefiles gerados pelo Designer para a máquina UNIX.
Nota: Se você criar a biblioteca compartilhada em uma máquina que não seja a do Serviço de Integração, deve também copiar os arquivos no seguinte diretório para a máquina da criação:
<PowerCenter_install_dir>\ExtProc\include\ct
No exemplo União, copie todos os arquivos de <client_installation_directory>/TX/UnionDemo.
2. Defina a variável de ambiente INFA_HOME como o diretório de instalação do Serviço de Integração.
Nota: Se você especificar um caminho de diretório incorreto para a variável de ambiente INFA_HOME, o Serviço de Integração não poderá ser iniciado.
3. Insira um comando da seguinte tabela para criar o projeto:

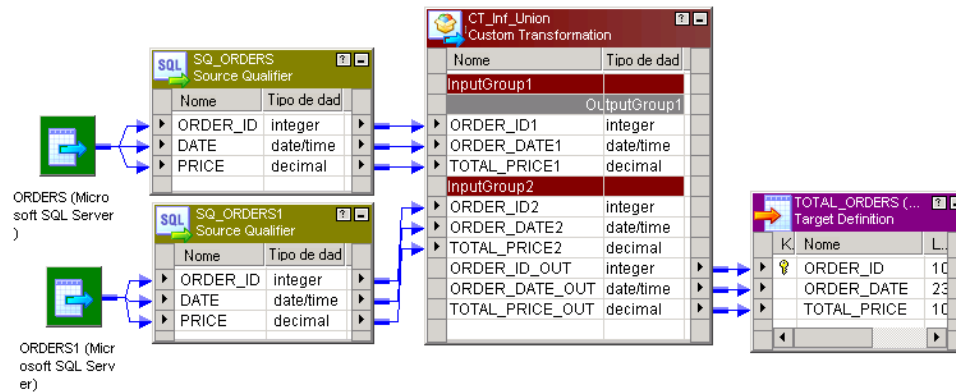
Versão do UNIX	Comando
AIX (32 bits)	make -f makefile.aix
AIX (64 bits)	make -f makefile.aix64
Linux	make -f makefile.linux
Solaris	make -f makefile.sol

Etapa 5. Criando um Mapeamento

No Mapping Designer, crie um mapeamento que utilize a Transformação personalizada.

Neste mapeamento, duas origens com as mesmas portas e tipos de dados conectam-se aos dois grupos de entrada na transformação personalizada. A transformação personalizada obtém as linhas das duas origens e as emite pelo seu grupo de saída. O grupo de saída tem as mesmas portas e tipos de dados que os grupos de entrada.

No exemplo União, crie um mapeamento semelhante ao da seguinte figura:



Etapa 6. Executar a Sessão em um Fluxo de Trabalho

Quando você executa a sessão, o Serviço de Integração procura a biblioteca compartilhada ou a DLL no local do tempo de execução especificado na transformação personalizada.

Para executar uma sessão em um fluxo de trabalho:

1. No Workflow Manager, crie um fluxo de trabalho.
2. Crie uma sessão para este mapeamento no fluxo de trabalho.
3. Copie a biblioteca compartilhada ou a DLL no diretório do local de tempo de execução.
4. Execute o fluxo de trabalho que contém a sessão.

Quando o Serviço de Integração carrega uma transformação personalizada associada a um procedimento, ele carrega a DLL ou a biblioteca compartilhada e chama o procedimento que você definir.

CAPÍTULO 4

Funções de Transformação Personalizada

Este capítulo inclui os seguintes tópicos:

- [Visão geral das Funções de Transformação Personalizada, 85](#)
- [Referência de Função, 86](#)
- [Trabalhando com Linhas, 89](#)
- [Funções Geradas, 90](#)
- [Funções API, 96](#)
- [Funções de API baseadas em matriz, 121](#)

Visão geral das Funções de Transformação Personalizada

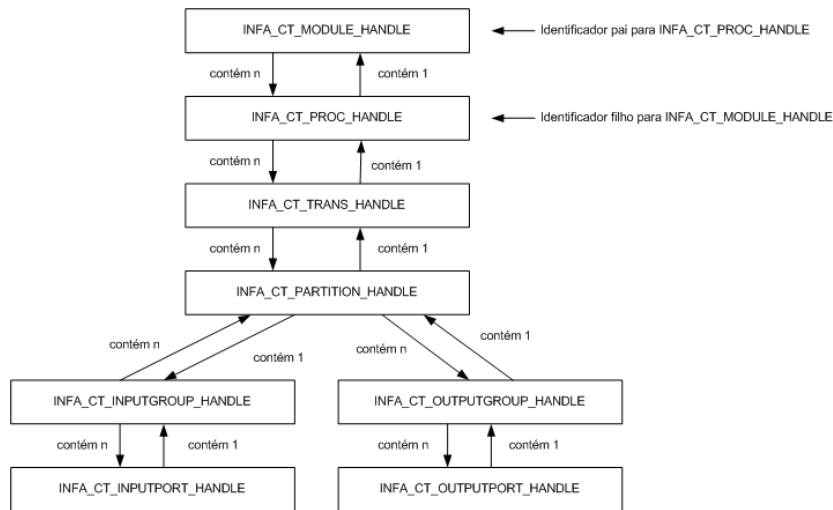
As transformações Personalizadas funcionam em conjunção com os procedimentos que você cria fora do Designer, para ampliar a funcionalidade do PowerCenter. As funções da transformação Personalizada permitem desenvolver a lógica da transformação em um procedimento que você associa com uma transformação Personalizada. O PowerCenter fornece dois conjuntos de funções denominadas geradas e API. O Serviço de Integração usa funções geradas para estabelecer uma interface com o procedimento. Quando você cria uma transformação Personalizada e gera os arquivos de código de origem, o Designer inclui as funções geradas nos arquivos. Use as funções API no código do procedimento para desenvolver a lógica de transformação.

Quando você escreve o código de procedimento, pode configurá-lo para receber um bloco de linhas do Serviço de Integração ou uma única linha de cada vez. Você pode aumentar o desempenho do procedimento quando recebe e processa um bloco de linhas.

Trabalhando com Identificadores

A maioria das funções está associada a um identificador, como INFA_CT_PARTITION_HANDLE. O primeiro parâmetro dessas funções é o identificador que a função afeta. Os identificadores de transformação personalizada têm entre si um relacionamento hierárquico. Um identificador pai tem um relacionamento 1:n com seu identificador filho.

A figura a seguir mostra identificadores da transformação Personalizada:



A tabela a seguir descreve identificadores da transformação Personalizada:

Nome do Identificador	Descrição
INFA_CT_MODULE_HANDLE	Representa a biblioteca compartilhada ou DLL. O procedimento externo só pode acessar o identificador de módulo em sua própria biblioteca ou DLL compartilhada. Ele não pode acessar o identificador de módulo em nenhuma outra biblioteca compartilhada ou DLL.
INFA_CT_PROC_HANDLE	Representa um procedimento específico no DLL ou biblioteca compartilhada. Você pode usar esse identificador quando precisar gravar uma função que afete um procedimento com referências de várias transformações Personalizadas.
INFA_CT_TRANS_HANDLE	Representa uma instância de Transformação personalizada na sessão.
INFA_CT_PARTITION_HANDLE	Representa uma partição específica em uma instância específica da transformação Personalizada.
INFA_CT_INPUTGROUP_HANDLE	Representa um grupo de entrada em uma partição.
INFA_CT_INPUTPORT_HANDLE	Representa uma porta de entrada em um grupo de entrada em uma partição.
INFA_CT_OUTPUTGROUP_HANDLE	Representa um grupo de saída em uma partição.
INFA_CT_OUTPUTPORT_HANDLE	Representa uma porta de saída em um grupo de saída em uma partição.

Referência de Função

As funções de Transformação personalizada incluem funções geradas e de API.

A tabela a seguir lista as funções geradas pela Transformação personalizada:

Função	Descrição
m_<module_name>_moduleInit()	Função de inicialização de módulo.
p_<proc_name>_procInit()	Função de inicialização de procedimento.
p_<proc_name>_partitionInit()	Função de inicialização de partição.
p_<proc_name>_inputRowNotification()	Função de notificação da linhas de entrada.
p_<proc_name>_dataBdryNotification()	Função de notificação de limite de dados.
p_<proc_name>_eofNotification()	Função de notificação de final de arquivo.
p_<proc_name>_partitionDeinit()	Função de desinicialização de partição.
p_<proc_name>_procedureDeinit()	Função de desinicialização de procedimento.
m_<module_name>_moduleDeinit()	Função de desinicialização de módulo.

A tabela a seguir lista as funções de API da Transformação personalizada:

Função	Descrição
INFA_CTSetDataAccessMode()	Função Definir modo de acesso aos dados.
INFA_CTGetAncestorHandle()	Função Obter tratamento de ancestral.
INFA_CTGetChildrenHandles()	Função Obter Identificadores de Filhos.
INFA_CTGetInputPortHandle()	Função Obter tratamento de porta de entrada.
INFA_CTGetOutputPortHandle()	Função Obter manipulador de porta de saída.
INFA_CTGetInternalProperty<datatype>()	Função Obter propriedades internas.
INFA_CTGetAllPropertyNamesM()	Obter todos os nomes de propriedade na função de modo MBCS.
INFA_CTGetAllPropertyNamesU()	Obter todos os nomes de propriedade na função de modo Unicode.
INFA_CTGetExternalProperty<datatype>M()	Obter propriedade externa na função MBCS.
INFA_CTGetExternalProperty<datatype>U()	Obter propriedade externa na função Unicode.
INFA_CTRebindInputDataType()	Reassociar a função de tipos de dados da porta de entrada.
INFA_CTRebindOutputDataType()	Reassociar a função de tipos de dados da porta de saída.
INFA_CTGetData<datatype>()	Funções obter dados.
INFA_CTSetData()	Definir funções de dados.
INFA_CTGetIndicator()	Obter funções de indicador.

Função	Descrição
INFA_CTSetIndicator()	Definir funções de indicador.
INFA_CTGetLength()	Obter funções de extensão.
INFA_CTSetLength()	Definir funções de extensão.
INFA_CTSetPassThruPort()	Função Definir porta de passagem.
INFA_CTOutputNotification()	Função de notificação de saída.
INFA_CTDataBdryOutputNotification()	Função de notificação de saída do limite de dados.
INFA_CTGetErrorMsgU()	Obter mensagem de erro na função Unicode.
INFA_CTGetErrorMsgM()	Obter mensagem de erro na função MBCS.
INFA_CTLogMessageU()	Mensagem de log no log de sessão na função Unicode.
INFA_CTLogMessageM()	Mensagem de log no log de sessão na função MBCS.
INFA_CTIncrementErrorCount()	Função de incremento de contagem de erros.
INFA_CTIstTerminateRequested()	É desativação da função solicitada.
INFA_CTBlockInputFlow()	Função de bloqueio de grupos de entrada.
INFA_CTUnblockInputFlow()	Função de bloqueio de grupos de entrada.
INFA_CTSetUserDefinedPtr()	Configurar função de ponteiro definida pelo usuário.
INFA_CTGetUserDefinedPtr()	Função Obter ponteiro definido pelo usuário.
INFA_CTChangeStringMode()	Alterar a função do modo de string.
INFA_CTSetDataCodePageID()	Função de definição de ID da página de código de dados.
INFA_CTGetRowStrategy()	Obter a função de estratégia de linha.
INFA_CTSetRowStrategy()	Função de definição de estratégia de linha.
INFA_CTChangeDefaultRowStrategy()	Alterar a estratégia de linha padrão de uma transformação.

A tabela a seguir lista as funções baseadas em matriz pela Transformação personalizada:

Função	Descrição
INFA_CTGetInputRowMax()	Obter número máximo de função de linhas de entrada.
INFA_CTGetOutputRowMax()	Obter número máximo de função de linhas de saída.
INFA_CTASetOutputRowMax()	Definir número máximo de função de linhas de saída.

Função	Descrição
INFA_CTAGetNumRows()	Função Obter número de linhas.
INFA_CTASetNumRows()	Função Definir números de linha.
INFA_CTAIsRowValid()	É uma função válida para a linha.
INFA_CTAGetData<datatype>()	Funções obter dados.
INFA_CTAGetIndicator()	Obter funções de indicador.
INFA_CTASetData()	Definir funções de dados.
INFA_CTAGetRowStrategy()	Obter a função de estratégia de linha.
INFA_CTASetRowStrategy()	Função de definição de estratégia de linha.
INFA_CTASetInputErrorRowM()	Definir função da linha de erro de entrada para MBCS.
INFA_CTASetInputErrorRowU()	Definir função da linha de erro de entrada para Unicode.

Trabalhando com Linhas

O Serviço de Integração pode passar uma única linha ao procedimento de transformação personalizada ou um bloco de linhas em uma matriz. Você pode especificar no código do procedimento para especificar se este receberá uma linha ou um bloco de linhas. O desempenho pode aumentar quando o procedimento recebe blocos de linhas:

- Você pode diminuir o número de chamadas de função feitas pelo Serviço de Integração e pelo procedimento. O Serviço de Integração e o procedimento chamam as funções de notificação da linha de entrada e de notificação de saída, respectivamente, menos vezes.
- Você pode aumentar a localidade do espaço de acesso à memória para os dados.
- Você pode gravar o código do procedimento para executar algoritmos em blocos de dados, em vez de em cada linha de dados.

Por padrão, o procedimento recebe uma linha de dados de cada vez. Para receber blocos de linhas, inclua a função `INFA_CTSetDataAccessMode()` para que o modo de acesso aos dados tenha base em matriz. Quando o modo de acesso aos dados tem base em matriz, use versões de funções de estratégia de linha e de manipulação de dados com base em matriz para acessar e fazer a saída de dados. Quando o modo de acesso a dados é baseado em linha, você deve usar a manipulação de dados baseada em linha, bem como funções de estratégia de linha para acessar e emitir os dados.

Todas as funções com base em matriz usam o prefixo `INFA_CTA`. Todas as outras funções usam o prefixo `INFA_CT`.

Use estas etapas para fazer com que o código do procedimento acesse um bloco de linhas:

1. Chame `INFA_CTSetDataAccessMode()` durante o processo de inicialização para que o modo de acesso aos dados tenha base em matriz.

2. Ao criar uma transformação Personalizada passiva, você também pode chamar `INFA_CTSetPassThruPort()` durante o processo de inicialização, para passar dados para portas de entrada e saída.

Quando o bloco de dados chega ao procedimento de transformação personalizada, o Serviço de Integração chama `p_<proc_name>_inputRowNotification()` para cada bloco de dados. Execute o restante das etapas dentro dessa função.

3. Chame `INFA_CTGetNumRows()` usando o controle do grupo de entrada na função de notificação da linha de entrada, para descobrir o número de linhas no bloco.
4. Chame uma das funções `INFA_CTGetData<datatype>()` usando o controle da porta de entrada para obter os dados de determinada linha no bloco.
5. Chame `INFA_CTSetData` para que as linhas de um bloco saiam.
6. Antes de chamar `INFA_CTOutputNotification()`, chame `INFA_CTSetNumRows()` para informar ao Serviço de Integração o número de linhas no bloco que o procedimento faz sair.
7. Chame `INFA_CTOutputNotification()`.

Regras e Diretrizes para Modo de Acesso a Dados Baseado em Linhas e Baseado em Matriz

Utilize as regras e diretrizes a seguir quando gravar o código do procedimento para usar um modo de acesso a dados baseado em linhas ou baseado em matriz.

- No modo baseado em linhas, você pode retornar `INFA_ROWERROR` na função de notificação de linha de entrada para indicar que a função encontrou um erro para a linha de dados na entrada. O Serviço de Integração incrementa a contagem de erros internos.
- No modo baseado em matriz, não retorne `INFA_ROWERROR` na função de notificação de linha de entrada. O Serviço de Integração trata essa função como um erro fatal. Se você precisar indicar uma linha em um bloco que contenha um erro, chame a função `INFA_CTSetInputErrorRowM()` ou `INFA_CTSetInputErrorRowU()`.
- No modo baseado em linhas, o Serviço de Integração passa somente linhas válidas para o procedimento.
- No modo baseado em matriz, um bloco de entrada pode conter linhas inválidas, como linhas descartadas, filtradas ou de erro. Chame `INFA_CTAIsRowValid()` para determinar se uma linha em um bloco é válida.
- No modo baseado em matriz, não chame `INFA_CTSetNumRows()` para uma transformação personalizada passiva. Você pode chamar essa função para transformações personalizadas ativas.
- No modo baseado em matriz, chame `INFA_CTOutputNotification()` uma vez.
- No modo baseado em matriz, você pode chamar `INFA_CTSetPassThruPort()` somente para transformações personalizadas passivas.
- No modo baseado em matriz para transformações personalizadas passivas, você deve emitir todas as linhas em um bloco de saída, inclusive as linhas de erro.

Funções Geradas

Ao usar o Designer para gerar o código de procedimento, o Designer inclui um conjunto de funções chamadas de funções geradas nos arquivos `m_<module_name>.c` e `p_<procedure_name>.c`. O Serviço de Integração usa as funções geradas para estabelecer uma interface com o procedimento. Ao executar uma

sessão, o Serviço de Integração chama essas funções geradas na seguinte ordem para cada grupo de ordem de carregamento de destino no mapeamento:

1. Funções de inicialização
2. Funções de notificação
3. Funções de desinicialização

Funções de Inicialização

Primeiro, o Serviço de Integração chama as funções de inicialização. Use as funções de inicialização para escrever processos que você deseja que o Serviço de Integração execute antes que ele passe os dados para a transformação Personalizada. Escrever o código nas funções de inicialização reduz a sobrecarga do processamento, porque o Serviço de Integração executa esses processos apenas uma vez para um módulo, procedimento ou partição.

O Designer gera as seguintes funções de inicialização:

- m_<module_name>_moduleInit()
- p_<proc_name>_proclnit()
- p_<proc_name>_partitionInit()

Função de Inicialização de Módulo

O Serviço de Integração chama a função m_<module_name>_moduleInit() durante a inicialização da sessão, antes de executar as tarefas pré-sessão. Ele chama essa função, uma vez para um módulo, antes de todas as outras funções.

Se você deseja que o Serviço de Integração execute um processo específico ao carregar o módulo, deve incluí-lo nesta função. Por exemplo, você pode escrever o código para criar estruturas globais que serão acessadas pelos procedimentos dentro deste módulo.

Utilize a seguinte sintaxe:

```
INFA_STATUS m_<module_name>_moduleInit(INFA_CT_MODULE_HANDLE module);
```

A tabela a seguir descreve os argumentos para esta função:

Argumento	Tipo de Dados	Entrada/ Saída	Descrição
módulo	INFA_CT_MODULE_HANDLE	Entrada	Manipulador de módulo.

O tipo de dados do valor de retorno é INFA_STATUS. Use INFA_SUCCESS e INFA_FAILURE para o valor de retorno. Quando a função retorna INFA_FAILURE, o Serviço de Integração interrompe a sessão.

Função de Inicialização do Procedimento

O Serviço de Integração chama a função p_<proc_name>_proclnit() durante a inicialização da sessão, antes de executar as tarefas pré-sessão e depois de executar a função do módulo de inicialização. O Serviço de Integração chama essa função uma vez para cada procedimento no módulo.

Grave códigos nessa função quando quiser que o Serviço de Integração execute um processo para certo procedimento. Você também pode inserir algumas funções da API na função de inicialização do procedimento, como funções de navegação e propriedades.

Utilize a seguinte sintaxe:

```
INFA_STATUS p_<proc_name>_procInit(INFA_CT_PROCEDURE_HANDLE procedure);
```

A tabela a seguir descreve os argumentos para esta função:

Argumento	Tipo de Dados	Entrada/ Saída	Descrição
procedimento	INFA_CT_PROCEDURE_HANDLE	Entrada	Manipulador de procedimento.

O tipo de dados do valor de retorno é INFA_STATUS. Use INFA_SUCCESS e INFA_FAILURE para o valor de retorno. Quando a função retorna INFA_FAILURE, o Serviço de Integração interrompe a sessão.

Função de Inicialização da Partição

O Serviço de Integração chama a função p_<proc_name>_partitionInit() antes de ela passar dados para a transformação personalizada. O Serviço de Integração chama essa função uma vez para cada partição em uma instância de transformação personalizada.

Se quiser que o Serviço de Integração execute determinado processo antes de passar dados através de uma partição da transformação personalizada, você deverá incluí-la nessa função.

Utilize a seguinte sintaxe:

```
INFA_STATUS p_<proc_name>_partitionInit(INFA_CT_PARTITION_HANDLE transformation);
```

A tabela a seguir descreve os argumentos para esta função:

Argumento	Tipo de Dados	Entrada/ Saída	Descrição
transformação	INFA_CT_PARTITION_HANDLE	Entrada	Manipulador de partição.

O tipo de dados do valor de retorno é INFA_STATUS. Use INFA_SUCCESS e INFA_FAILURE para o valor de retorno. Quando a função retorna INFA_FAILURE, o Serviço de Integração interrompe a sessão.

Nota: Quando a transformação personalizada requer um thread para cada partição, você pode incluir operações específicas do thread na função de inicialização da partição.

Funções de Notificação

O Serviço de Integração chama as funções de notificação quando transfere uma linha de dados para a transformação Personalizada.

O Designer gera as seguintes funções de notificação:

- p_<proc_name>_inputRowNotification()
- p_<proc_name>_dataBdryRowNotification()
- p_<proc_name>_eofNotification()

Nota: Quando a Transformação personalizada requer um thread para cada partição, você pode incluir operações específicas do thread nas funções de notificação.

Função de Notificação da Linhas de Entrada

O Serviço de Integração chama a função `p_<proc_name>_inputRowNotification()` quando passa por uma linha ou por um bloco de linhas para a Transformação personalizada. Ela observa qual grupo e partição de entrada recebe dados por meio do controle do grupo de entrada e do controle de partição.

Utilize a seguinte sintaxe:

```
INFA_ROWSTATUS p_<proc_name>_inputRowNotification(INFA_CT_PARTITION_HANDLE Partition,
INFA_CT_INPUTGROUP_HANDLE group);
```

A tabela a seguir descreve os argumentos para esta função:

Argumento	Tipo de Dados	Entrada/ Saída	Descrição
partição	INFA_CT_PARTITION_HANDLE	Entrada	Manipulador de partição.
grupo	INFA_CT_INPUTGROUP_HANDLE	Entrada	Manipulador de grupo de entrada.

O tipo de dados do valor retornado é `INFA_ROWSTATUS`. Use os seguintes valores para o valor retornado:

- **INFA_ROWSUCCESS.** Indica a função que processou com êxito a linha de dados.
- **INFA_ROWERROR.** Indica a função que encontrou um erro na linha de dados. O Serviço de Integração incrementa a contagem de erros internos. Retorne apenas esse valor quando o modo de acesso aos dados for linha.

Se a função de notificação da linha de entrada fornecer `INFA_ROWERROR` em modo baseado em matriz, o Serviço de Integração tratará isso como um erro fatal. Se você precisar indicar uma linha em um bloco que contenha um erro, chame a função `INFA_CTASetInputErrorRowM()` ou `INFA_CTASetInputErrorRowU()`.
- **INFA_FATALERROR.** Indica a função que encontrou um erro fatal na linha de dados ou no bloco de dados. O Serviço de Integração causa falha na sessão.

Função de Notificação de Limite de Dados

O Serviço de Integração chama a função `p_<proc_name>_dataBdryNotification()` quando transfere uma linha de confirmação ou reversão para uma partição.

Utilize a seguinte sintaxe:

```
INFA_STATUS p_<proc_name>_dataBdryNotification(INFA_CT_PARTITION_HANDLE transformation,
INFA_CTDataBdryType dataBoundaryType);
```

A tabela a seguir descreve os argumentos para esta função:

Argumento	Tipo de Dados	Entrada/ Saída	Descrição
transformação	INFA_CT_PARTITION_HANDLE	Entrada	Manipulador de partição.
dataBoundaryType	INFA_CTDataBdryType	Entrada	O Serviço de Integração usa um dos seguintes valores para o parâmetro <code>dataBoundaryType</code> : <ul style="list-style-type: none">- <code>eBT_COMMIT</code>- <code>eBT_ROLLBACK</code>

O tipo de dados do valor de retorno é INFA_STATUS. Use INFA_SUCCESS e INFA_FAILURE para o valor de retorno. Quando a função retorna INFA_FAILURE, o Serviço de Integração interrompe a sessão.

Função de Notificação de Final de Arquivo

O Serviço de Integração chama a função p_<proc_name>_eofNotification() depois de transferir a última linha para uma partição em um grupo de entrada.

Utilize a seguinte sintaxe:

```
INFA_STATUS p_<proc_name>_eofNotification(INFA_CT_PARTITION_HANDLE transformation,  
INFA_CT_INPUTGROUP_HANDLE group);
```

A tabela a seguir descreve os argumentos para esta função:

Argumento	Tipo de Dados	Entrada/ Saída	Descrição
transformação	INFA_CT_PARTITION_HANDLE	Entrada	Manipulador de partição.
grupo	INFA_CT_INPUTGROUP_HANDLE	Entrada	Manipulador de grupo de entrada.

O tipo de dados do valor de retorno é INFA_STATUS. Use INFA_SUCCESS e INFA_FAILURE para o valor de retorno. Quando a função retorna INFA_FAILURE, o Serviço de Integração interrompe a sessão.

Funções de Desinicialização

O Serviço de Integração chama as funções de desinicialização após processar dados para a transformação Personalizada. Use as funções de desinicialização para registrar processos que você queira que o Serviço de Integração execute após passar todas as linhas de dados à transformação Personalizada.

O Designer gera as seguintes funções de não inicialização:

- p_<proc_name>_partitionDeinit()
- p_<proc_name>_procDeinit()
- m_<module_name>_moduleDeinit()

Nota: Quando a Transformação personalizada exigir um thread para cada partição, você pode incluir operações específicas do thread nas funções de inicialização e não inicialização.

Função de Anulação da Inicialização da Partição

O Serviço de Integração chama a função p_<proc_name>_partitionDeinit() depois da função p_<proc_name>_eofNotification() ou da função p_<proc_name>_abortNotification(). O Serviço de Integração chama essa função uma vez para cada partição da Transformação personalizada.

Utilize a seguinte sintaxe:

```
INFA_STATUS p_<proc_name>_partitionDeinit(INFA_CT_PARTITION_HANDLE partition);
```

A tabela a seguir descreve os argumentos para esta função:

Argumento	Tipo de Dados	Entrada/ Saída	Descrição
partição	INFA_CT_PARTITION_HANDLE	Entrada	Manipulador de partição.

O tipo de dados do valor de retorno é INFA_STATUS. Use INFA_SUCCESS e INFA_FAILURE para o valor de retorno. Quando a função retorna INFA_FAILURE, o Serviço de Integração interrompe a sessão.

Nota: Quando a Transformação personalizada requer um thread para cada partição, você pode incluir operações específicas do thread na função de inicialização da partição.

Função de Anulação da Inicialização do Procedimento

O Serviço de Integração chama a função `p_<proc_name>_procDeinit()` depois de chamar a função `p_<proc_name>_partitionDeinit()` para todas as partições de cada instância da transformação personalizada que usa esse procedimento no mapeamento.

Utilize a seguinte sintaxe:

```
INFA_STATUS p_<proc_name>_procDeinit(INFA_CT_PROCEDURE_HANDLE procedure, INFA_STATUS sessionStatus);
```

A tabela a seguir descreve os argumentos para esta função:

Argumento	Tipo de Dados	Entrada/ Saída	Descrição
procedimento	INFA_CT_PROCEDURE_HANDLE	Entrada	Manipulador de procedimento.
sessionStatus	INFA_STATUS	Entrada	O Serviço de Integração usa um dos seguintes valores para o parâmetro sessionStatus: <ul style="list-style-type: none">- INFA_SUCCESS. Indica que a sessão teve êxito.- INFA_FAILURE. Indica que a sessão falhou.

O tipo de dados do valor de retorno é INFA_STATUS. Use INFA_SUCCESS e INFA_FAILURE para o valor de retorno. Quando a função retorna INFA_FAILURE, o Serviço de Integração interrompe a sessão.

Função de Desinicialização de Módulo

O Serviço de Integração chama a função `m_<module_name>_moduleDeinit()` depois de executar as tarefas pós-sessão. Ele chama esta função, uma vez para um módulo, depois de todas as outras funções.

Utilize a seguinte sintaxe:

```
INFA_STATUS m_<module_name>_moduleDeinit(INFA_CT_MODULE_HANDLE module, INFA_STATUS sessionStatus);
```

A tabela a seguir descreve os argumentos para esta função:

Argumento	Tipo de Dados	Entrada/ Saída	Descrição
módulo	INFA_CT_MODULE_HANDLE	Entrada	Manipulador de módulo.
sessionStatus	INFA_STATUS	Entrada	O Serviço de Integração usa um dos seguintes valores para o parâmetro sessionStatus: <ul style="list-style-type: none">- INFA_SUCCESS. Indica que a sessão teve êxito.- INFA_FAILURE. Indica que a sessão falhou.

O tipo de dados do valor de retorno é INFA_STATUS. Use INFA_SUCCESS e INFA_FAILURE para o valor de retorno. Quando a função retorna INFA_FAILURE, o Serviço de Integração interrompe a sessão.

Funções API

O PowerCenter fornece um conjunto de funções API, que você usa para desenvolver a lógica da transformação. Quando o Designer gera os arquivos do código de origem, ele inclui as funções geradas no código de origem. Adicione funções API ao código, para implementar a lógica da transformação. O procedimento usa as funções API para formar interface com o Serviço de Integração. Você deve codificar funções de API no arquivo C do procedimento. Você pode também codificar o arquivo C do módulo.

A Informatica fornece os seguintes grupos de funções API:

- Configurar o modo de acesso aos dados
- Navegação
- Propriedade
- Religar tipos de dados
- Manipulação de dados (modo baseado em linha)
- Configurar a porta de passagem
- Notificação de saída
- Notificação de saída do limite de dados
- Erro
- Mensagem do log da sessão
- Incrementar contagem do erro
- Está terminado
- Bloqueio
- Ponteiro
- Alterar modo de string
- Definir página de código de dados
- Estratégia de linhas (modo baseado em linha)
- Alterar estratégia da linha padrão

A Informatica também fornece Funções de API baseadas em matriz.

Função de Definição do Modo de Acesso aos Dados

Por padrão, o Serviço de Integração passa uma linha de cada vez ao procedimento de transformação personalizada. Entretanto, use a função `INFA_CTSetDataAccessMode()` para que o modo de acesso aos dados tenha base em matriz. Quando você define a base em matriz para o modo de acesso aos dados, o Serviço de Integração passa ao procedimento várias linhas como um bloco em uma matriz.

Ao definir a base em matriz para o modo de acesso aos dados, você deve usar versões de funções de estratégia de linha e de manipulação de dados com base em matriz. Ao alternar funções de estratégia de linha e manipulação de dados com base em linhas para o modo com base em matriz, você terá resultados inesperados. Por exemplo, a biblioteca compartilhada ou de DLL pode falhar.

Você só pode usar essa função na função de inicialização do procedimento.

Se você não usar essa função no código do procedimento, o modo de acesso aos dados terá base em linhas. Contudo, se você quiser que o modo de acesso aos dados tenha base em linhas, inclua essa função e defina o modo de acesso com base em linhas.

Utilize a seguinte sintaxe:

```
INFA_STATUS INFA_CTSetDataAccessMode( INFA_CT_PROCEDURE_HANDLE procedure,
INFA_CT_DATA_ACCESS_MODE mode );
```

A tabela a seguir descreve os argumentos para esta função:

Argumento	Tipo de Dados	Entrada/ Saída	Descrição
procedimento	INFA_CT_PROCEDURE_HANDLE	Entrada	Nome do procedimento.
modo	INFA_CT_DATA_ACCESS_MODE	Entrada	Modo de acesso a dados. Use os seguintes valores para o parâmetro de modo: - eDA_ROW - eDA_ARRAY

Funções de Navegação

Use as funções de navegação quando desejar que o procedimento navegue na hierarquia da manipulação.

O PowerCenter fornece as seguintes funções de navegação:

- `INFA_CTGetAncestorHandle()`
- `INFA_CTGetChildrenHandles()`
- `INFA_CTGetInputPortHandle()`
- `INFA_CTGetOutputPortHandle()`

Função Obter tratamento de ancestral.

Use a função `INFA_CTGetAncestorHandle()` se quiser que o procedimento acesse um identificador pai de um determinado indicador.

Utilize a seguinte sintaxe:

```
INFA_CT_HANDLE INFA_CTGetAncestorHandle(INFA_CT_HANDLE handle, INFA_CTHandleType
returnHandleType);
```

A tabela a seguir descreve os argumentos para esta função:

Argumento	Tipo de Dados	Entrada/ Saída	Descrição
manipulador	INFA_CT_HANDLE	Entrada	Nome do manipulador.
returnHandleType	INFA_CTHandleType	Entrada	Tipo de manipulador de retorno. Use os seguintes valores para o parâmetro returnHandleType: <ul style="list-style-type: none">- PROCEDURETYPE- TRANSFORMATIONTYPE- PARTITIONTYPE- INPUTGROUPTYPE- OUTPUTGROUPTYPE- INPUTPORTTYPE- OUTPUTPORTTYPE

O parâmetro do manipulador especifica o manipulador cujo pai você deseja que o procedimento acesse. O Serviço de Integração retorna INFA_CT_HANDLE se você especificar um manipulador válido na função. Do contrário, ele retorna um valor nulo.

Para evitar erros de compilação, você deve codificar o procedimento para definir um nome de manipulador para o valor retornado.

Por exemplo, você pode inserir o código a seguir:

```
INFA_CT_MODULE_HANDLE module = INFA_CTGetAncestorHandle(procedureHandle,  
INFA_CT_HandleType);
```

Função Obter Identificadores de Filhos

Use a função INFA_CTGetChildrenHandles() quando desejar que o procedimento acesse os identificadores de filhos de um determinado identificador.

Utilize a seguinte sintaxe:

```
INFA_CT_HANDLE* INFA_CTGetChildrenHandles(INFA_CT_HANDLE handle, size_t*  
pnChildrenHandles, INFA_CTHandleType returnHandleType);
```

A tabela a seguir descreve os argumentos para esta função:

Argumento	Tipo de Dados	Entrada/ Saída	Descrição
manipulador	INFA_CT_HANDLE	Entrada	Nome do manipulador.
pnChildrenHandles	size_t*	Saída	O Serviço de Integração retorna uma matriz de manipuladores de filhos. O parâmetro pnChildrenHandles indica o número de manipuladores de filhos na matriz.
returnHandleType	INFA_CTHandleType	Entrada	Use os seguintes valores para o parâmetro returnHandleType: <ul style="list-style-type: none">- PROCEDURETYPE- TRANSFORMATIONTYPE- PARTITIONTYPE- INPUTGROUPTYPE- OUTPUTGROUPTYPE- INPUTPORTTYPE- OUTPUTPORTTYPE

O parâmetro de identificador especifica o identificador com o qual você deseja produzir os filhos para fins de acesso. O Serviço de Integração retorna INFA_CT_HANDLE* quando você especifica um identificador válido na função. Do contrário, ele retorna um valor nulo.

Para evitar erros de compilação, é necessário codificar o procedimento para definir um nome de identificador para o valor retornado.

Por exemplo, você pode inserir o código a seguir:

```
INFA_CT_PARTITION_HANDLE partition = INFA_CTGetChildrenHandles(procedureHandle,  
pnChildrenHandles, INFA_CT_PARTITION_HANDLE_TYPE);
```

Funções Obter Tratamento de Porta

O Serviço de Integração associa o INFA_CT_INPUTPORT_HANDLE com as portas de entrada e entrada/saída, e o INFA_CT_OUTPUTPORT_HANDLE com as portas de saída e entrada/saída.

O PowerCenter fornece as seguintes funções de manipulador da porta de obtenção:

- **INFA_CTGetInputPortHandle()**. Use esta função quando o procedimento conhece o manipulador da porta de saída para uma porta de entrada/saída e precisa do manipulador da porta de entrada.

Utilize a seguinte sintaxe:

```
INFA_CT_INPUTPORT_HANDLE INFA_CTGetInputPortHandle(INFA_CT_OUTPUTPORT_HANDLE  
outputPortHandle);
```

A tabela a seguir descreve o argumento para esta função:

Argumento	Tipo de Dados	Entrada/ Saída	Descrição
outputPortHandle	INFA_CT_OUTPUTPORT_HANDLE	entrada	Manipulador de porta de saída.

- **INFA_CTGetOutputPortHandle()**. Use esta função quando o procedimento conhece o manipulador da porta de entrada para uma porta de entrada/saída e precisa do manipulador da porta de saída.

Utilize a seguinte sintaxe:

```
INFA_CT_OUTPUTPORT_HANDLE INFA_CTGetOutputPortHandle(INFA_CT_INPUTPORT_HANDLE
inputPortHandle);
```

A tabela a seguir descreve o argumento para esta função:

Argumento	Tipo de Dados	Entrada/ Saída	Descrição
inputPortHandle	INFA_CT_INPUTPORT_HANDLE	entrada	Manipulador de porta de entrada.

O Serviço de Integração retorna NULL quando você usa as funções do manipulador da porta de obtenção com portas de entrada ou saída.

Funções de Propriedade

Use as funções de propriedade quando quiser que o procedimento acesse as propriedades de transformação personalizadas. As funções de propriedade acessam propriedades nas seguintes guias da transformação personalizada:

- Portas
- Propriedades
- Propriedades de inicialização
- Extensões de Metadados
- Definições de Atributo de Porta

Use as seguintes funções de propriedade nas funções de inicialização:

- INFA_CTGetInternalProperty<datatype>()
- INFA_CTGetAllPropertyNamesM()
- INFA_CTGetAllPropertyNamesU()
- INFA_CTGetExternalProperty<datatype>M()
- INFA_CTGetExternalProperty<datatype>U()

Função Obter Propriedade Interna

O PowerCenter fornece funções para acessar os atributos de porta especificados na guia de portas, e as propriedades especificadas para os atributos na guia Propriedades da transformação Personalizada.

O Serviço de Integração associa cada porta e atributo de propriedade a uma ID de propriedade. Você deve especificar a ID de propriedade no procedimento para acessar os valores especificados para os atributos. Para o parâmetro do manipulador, especifique um nome de manipulador na respectiva hierarquia. O Serviço de Integração provoca a falha da sessão se o nome do manipulador for inválido.

Use as seguintes funções quando quiser que o procedimento acesse as propriedades:

- **INFA_CTGetInternalPropertyStringM()**. Acessa um valor de string de tipo no MBCS para uma determinada ID de propriedade.

Utilize a seguinte sintaxe:

```
INFA_STATUS INFA_CTGetInternalPropertyStringM( INFA_CT_HANDLE handle, size_t propId,
const char** psPropValue );
```

- **INFA_CTGetInternalPropertyStringU()**. Acessa um valor de string de tipo em Unicode para uma determinada ID de propriedade.

Utilize a seguinte sintaxe:

```
INFA_STATUS INFA_CTGetInternalPropertyStringU( INFA_CT_HANDLE handle, size_t propId,
const INFA_UNICHAR** psPropValue );
```

- **INFA_CTGetInternalPropertyInt32()**. Acessa um valor de inteiro de tipo para uma determinada ID de propriedade.

Utilize a seguinte sintaxe:

```
INFA_STATUS INFA_CTGetInternalPropertyInt32( INFA_CT_HANDLE handle, size_t propId,
INFA_INT32* pnPropValue );
```

- **INFA_CTGetInternalPropertyBool()**. Acessa um valor de tipo Booleano para uma determinada ID de propriedade.

Utilize a seguinte sintaxe:

```
INFA_STATUS INFA_CTGetInternalPropertyBool( INFA_CT_HANDLE handle, size_t propId,
INFA_BOOLEAN* pbPropValue );
```

- **INFA_CTGetInternalPropertyINFA_PTR()**. Acessa um ponteiro para um valor para uma determinada ID de propriedade.

Utilize a seguinte sintaxe:

```
INFA_STATUS INFA_CTGetInternalPropertyINFA_PTR( INFA_CT_HANDLE handle, size_t propId,
INFA_PTR* pvPropValue );
```

O tipo de dados do valor de retorno é INFA_STATUS. Use INFA_SUCCESS e INFA_FAILURE para o valor de retorno.

IDs de Propriedade da Porta e de Atributos de Propriedade

A tabela a seguir lista as IDs de propriedade da porta e atributos de propriedade na transformação personalizada. Cada tabela lista um identificador de transformação personalizada e as IDs de propriedade que você pode acessar com o identificador em uma função de propriedade.

A tabela a seguir lista IDs da propriedade INFA_CT_MODULE_HANDLE:

ID de propriedade do identificador	Tipo de dados	Descrição
INFA_CT_MODULE_NAME	String	Especifica o nome do módulo.
INFA_CT_SESSION_INFA_VERSION	String	Especifica a versão do Informatica.
INFA_CT_SESSION_CODE_PAGE	Número inteiro	Especifica o código de página do Serviço de Integração.
INFA_CT_SESSION_DATAMOVEMENT_MODE	Número inteiro	Especifica o modo de movimento dos dados. O Serviço de Integração retorna um dos seguintes valores: <ul style="list-style-type: none">- eASM_MBCS- eASM_UNICODE
INFA_CT_SESSION_VALIDATE_CODEPAGE	Boolean	Especifica se o Serviço de Integração impõe a validação da página de código.

ID de propriedade do identificador	Tipo de dados	Descrição
INFA_CT_SESSION_PROD_INSTALL_DIR	String	Especifica o diretório de instalação do Serviço de Integração.
INFA_CT_SESSION_HIGH_PRECISION_MODE	Boolean	Especifica se a sessão está configurada para alta precisão.
INFA_CT_MODULE_RUNTIME_DIR	String	Especifica o diretório de tempo de execução para a DLL ou a biblioteca compartilhada.
INFA_CT_SESSION_IS_UPD_STR_ALLOWED	Boolean	Especifica se a propriedade da transformação Estratégia de Atualização está selecionada na transformação.
INFA_CT_TRANS_OUTPUT_IS_REPEATABLE	Número inteiro	Especifica se a transformação personalizada produz dados na mesma ordem em cada sessão executada. O Serviço de Integração retorna um dos seguintes valores: <ul style="list-style-type: none"> - eOUTREPEAT_NEVER = 1 - eOUTREPEAT_ALWAYS = 2 - eOUTREPEAT_BASED_ON_INPUT_ORDER = 3
INFA_CT_TRANS_FATAL_ERROR	Boolean	Especifica se a transformação personalizada causou um erro fatal. O Serviço de Integração retorna um dos seguintes valores: <ul style="list-style-type: none"> - INFA_TRUE - INFA_FALSE

A tabela a seguir lista IDs da propriedade INFA_CT_PROC_HANDLE:

ID de propriedade do identificador	Tipo de dados	Descrição
INFA_CT_PROCEDURE_NAME	String	Especifica o nome do procedimento de transformação personalizada.

A tabela a seguir lista IDs da propriedade INFA_CT_TRANS_HANDLE:

ID de propriedade do identificador	Tipo de dados	Descrição
INFA_CT_TRANS_INSTANCE_NAME	String	Especifica o nome da instância de Transformação personalizada.
INFA_CT_TRANS_TRACE_LEVEL	Número inteiro	Especifica o nível de rastreamento. O Serviço de Integração retorna um dos seguintes valores: <ul style="list-style-type: none"> - eTRACE_TERSE - eTRACE_NORMAL - eTRACE_VERBOSE_INIT - eTRACE_VERBOSE_DATA

ID de propriedade do identificador	Tipo de dados	Descrição
INFA_CT_TRANS_MAY_BLOCK_DATA	Boolean	Especifica se o Serviço de Integração permite que o procedimento bloqueie dados de entrada na sessão atual.
INFA_CT_TRANS_MUST_BLOCK_DATA	Boolean	Especifica se a propriedade de transformação personalizada Entradas Devem Bloquear está selecionada.
INFA_CT_TRANS_ISACTIVE	Boolean	Especifica se a transformação personalizada é uma transformação ativa ou passiva.
INFA_CT_TRANS_ISPARTITIONABLE	Boolean	Especifica se você pode particionar sessões que usem essa transformação personalizada.
INFA_CT_TRANS_IS_UPDATE_STRATEGY	Boolean	Especifica se a transformação personalizada se comporta como uma transformação Estratégia de Atualização.
INFA_CT_TRANS_DEFAULT_UPDATE_STRATEGY	Número inteiro	Especifica a estratégia de atualização padrão. <ul style="list-style-type: none"> - eDUS_INSERT - eDUS_UPDATE - eDUS_DELETE - eDUS_REJECT - eDUS_PASSTHROUGH
INFA_CT_TRANS_NUM_PARTITIONS	Número inteiro	Especifica o número de partições nas sessões que usam essa transformação personalizada.
INFA_CT_TRANS_DATACODEPAGE	Número inteiro	Especifica a página de código em que o Serviço de Integração passa dados para a transformação personalizada. Use a função da página de código de dados definida se quiser que a transformação personalizada acesse dados em outra página de código.
INFA_CT_TRANS_TRANSFORM_SCOPE	Número inteiro	Especifica o escopo na transformação personalizada. O Serviço de Integração retorna um dos seguintes valores: <ul style="list-style-type: none"> - eTS_ROW - eTS_TRANSACTION - eTS_ALLINPUT
INFA_CT_TRANS_GENERATE_TRANSACTIONS	Boolean	Especifica se a propriedade da transação Gerar está ativada. O Serviço de Integração retorna um dos seguintes valores: <ul style="list-style-type: none"> - INFA_TRUE - INFA_FALSE

ID de propriedade do identificador	Tipo de dados	Descrição
INFA_CT_TRANS_OUTPUT_IS_REPEATABLE	Número inteiro	<p>Especifica se a transformação personalizada produz dados na mesma ordem em cada sessão executada. O Serviço de Integração retorna um dos seguintes valores:</p> <ul style="list-style-type: none"> - eOUTREPEAT_NEVER = 1 - eOUTREPEAT_ALWAYS = 2 - eOUTREPEAT_BASED_ON_INPUT_ORDER = 3
INFA_CT_TRANS_FATAL_ERROR	Boolean	<p>Especifica se a transformação personalizada causou um erro fatal. O Serviço de Integração retorna um dos seguintes valores:</p> <ul style="list-style-type: none"> - INFA_TRUE - INFA_FALSE

A tabela a seguir lista IDs das propriedades INFA_CT_INPUT_GROUP_HANDLE e INFA_CT_OUTPUT_GROUP_HANDLE:

ID de propriedade do identificador	Tipo de dados	Descrição
INFA_CT_GROUP_NAME	String	Especifica o nome do grupo.
INFA_CT_GROUP_NUM_PORTS	Número inteiro	Especifica o número de portas no grupo.
INFA_CT_GROUP_ISCONNECTED	Boolean	Especifica se todas as portas em um grupo estão conectadas a outra transformação.
INFA_CT_PORT_NAME	String	Especifica o nome da porta.
INFA_CT_PORT_CDATATYPE	Número inteiro	<p>Especifica o tipo de dados da porta. O Serviço de Integração retorna um dos seguintes valores:</p> <ul style="list-style-type: none"> - eINFA_CTYPE_SHORT - eINFA_CTYPE_INT32 - eINFA_CTYPE_CHAR - eINFA_CTYPE_RAW - eINFA_CTYPE_UNICHAR - eINFA_CTYPE_TIME - eINFA_CTYPE_FLOAT - eINFA_CTYPE_DOUBLE - eINFA_CTYPE_DECIMAL18_FIXED - eINFA_CTYPE_DECIMAL28_FIXED - eINFA_CTYPE_INFA_CTDATETIME
INFA_CT_PORT_PRECISION	Número inteiro	Especifica a precisão da porta.
INFA_CT_PORT_SCALE	Número inteiro	Especifica a escala da porta, se aplicável.

ID de propriedade do identificador	Tipo de dados	Descrição
INFA_CT_PORT_IS_MAPPED	Boolean	Especifica se a porta está vinculada a outras transformações no mapeamento.
INFA_CT_PORT_STORAGESIZE	Número inteiro	Especifica o tamanho do armazenamento interno dos dados de uma porta. O tamanho do armazenamento depende do tipo de dados da porta.
INFA_CT_PORT_BOUNDDATATYPE	Número inteiro	Especifica o tipo de dados da porta. Utilize, em vez de usar INFA_CT_PORT_CDATATYPE, se você reassociar a porta e especificar um tipo de dados diferente do padrão.

A tabela a seguir lista IDs das propriedades INFA_CT_INPUTPORT_HANDLE e INFA_CT_OUTPUT_HANDLE:

ID de propriedade do identificador	Tipo de dados	Descrição
INFA_CT_PORT_NAME	String	Especifica o nome da porta.
INFA_CT_PORT_CDATATYPE	Número inteiro	Especifica o tipo de dados da porta. O Serviço de Integração retorna um dos seguintes valores: <ul style="list-style-type: none"> - eINFA_CTYPE_SHORT - eINFA_CTYPE_INT32 - eINFA_CTYPE_CHAR - eINFA_CTYPE_RAW - eINFA_CTYPE_UNICHAR - eINFA_CTYPE_TIME - eINFA_CTYPE_FLOAT - eINFA_CTYPE_DOUBLE - eINFA_CTYPE_DECIMAL18_FIXED - eINFA_CTYPE_DECIMAL28_FIXED - eINFA_CTYPE_INFA_CTDATETIME
INFA_CT_PORT_PRECISION	Número inteiro	Especifica a precisão da porta.
INFA_CT_PORT_SCALE	Número inteiro	Especifica a escala da porta, se aplicável.
INFA_CT_PORT_IS_MAPPED	Boolean	Especifica se a porta está vinculada a outras transformações no mapeamento.
INFA_CT_PORT_STORAGESIZE	Número inteiro	Especifica o tamanho do armazenamento interno dos dados de uma porta. O tamanho do armazenamento depende do tipo de dados da porta.
INFA_CT_PORT_BOUNDDATATYPE	Número inteiro	Especifica o tipo de dados da porta. Utilize, em vez de usar INFA_CT_PORT_CDATATYPE, se você reassociar a porta e especificar um tipo de dados diferente do padrão.

Obter Todos os Nomes de Propriedades Externas (MBCS ou Unicode)

O PowerCenter fornece duas funções para acessar os nomes de propriedade definidos na guia Extensões de Metadados, guia Propriedades de Inicialização e guia Definições de Atributo de Porta da transformação Personalizada.

Use as seguintes funções quando desejar que o procedimento acesse os nomes de propriedade:

- **INFA_CTGetAllPropertyNamesM()**. Acessa os nomes de propriedade no MBCS.

Utilize a seguinte sintaxe:

```
INFA_STATUS INFA_CTGetAllPropertyNamesM(INFA_CT_HANDLE handle, const char*const**  
paPropertyNames, size_t* pnProperties);
```

A tabela a seguir descreve os argumentos para esta função:

Argumento	Tipo de Dados	Entrada/ Saída	Descrição
manipulador	INFA_CT_HANDLE	Entrada	Especifique o nome do manipulador.
paPropertyNames	const char*const**	Saída	Especifica o nome da propriedade. O Serviço de Integração retorna uma série de nomes de propriedade no MBCS.
pnProperties	size_t*	Saída	Indica o número de propriedades na série.

- **INFA_CTGetAllPropertyNamesU()**. Acessa os nomes de propriedade em Unicode.

Utilize a seguinte sintaxe:

```
INFA_STATUS INFA_CTGetAllPropertyNamesU(INFA_CT_HANDLE handle, const  
INFA_UNICHAR*const** pasPropertyNames, size_t* pnProperties);
```

A tabela a seguir descreve os argumentos para esta função:

Argumento	Tipo de Dados	Entrada/ Saída	Descrição
manipulador	INFA_CT_HANDLE	Entrada	Especifique o nome do manipulador.
paPropertyNames	const INFA_UNICHAR*const**	Saída	Especifica o nome da propriedade. O Serviço de Integração retorna uma série de nomes de propriedade em Unicode.
pnProperties	size_t*	Saída	Indica o número de propriedades na série.

O tipo de dados do valor de retorno é INFA_STATUS. Use INFA_SUCCESS e INFA_FAILURE para o valor de retorno.

Obter Propriedades Externas (MBCS ou Unicode)

O PowerCenter fornece funções para acessar os valores das propriedades definidas na guia Extensões de Metadados, guia Propriedades de Inicialização, ou guia Definições de Atributo de Porta da Transformação personalizada.

Você deve especificar os nomes de propriedade nas funções se deseja que o procedimento acesse os valores. Use as funções INFA_CTGetAllPropertyNamesM() ou INFA_CTGetAllPropertyNamesU() para acessar

os nomes de propriedade. Para o parâmetro do manipulador, especifique um nome de manipulador na respectiva hierarquia. O Serviço de Integração provoca a falha da sessão se o nome do manipulador for inválido.

Nota: Se você definir uma propriedade de inicialização com o mesmo nome que uma extensão de metadados, o Serviço de Integração retorna o valor da extensão de metadados.

Use as seguintes funções quando desejar que o procedimento acesse os valores das propriedades:

- **INFA_CTGetExternalProperty<datatype>M().** Acessa o valor da propriedade no MBCS.

A tabela a seguir mostra a sintaxe:

Sintaxe	Tipo de Dados da Propriedade
<code>INFA_STATUS INFA_CTGetExternalPropertyStringM(INFA_CT_HANDLE handle, const char* sPropName, const char** psPropValue);</code>	String
<code>INFA_STATUS INFA_CTGetExternalPropertyINT32M(INFA_CT_HANDLE handle, const char* sPropName, INFA_INT32* pnPropValue);</code>	Número inteiro
<code>INFA_STATUS INFA_CTGetExternalPropertyBoolM(INFA_CT_HANDLE handle, const char* sPropName, INFA_BOOLEAN* pbPropValue);</code>	Booleano

- **INFA_CTGetExternalProperty<datatype>U().** Acessa o valor da propriedade em Unicode.

A tabela a seguir mostra a sintaxe:

Sintaxe	Tipo de Dados da Propriedade
<code>INFA_STATUS INFA_CTGetExternalPropertyStringU(INFA_CT_HANDLE handle, INFA_UNICHAR* sPropName, INFA_UNICHAR** psPropValue);</code>	String
<code>INFA_STATUS INFA_CTGetExternalPropertyStringU(INFA_CT_HANDLE handle, INFA_UNICHAR* sPropName, INFA_INT32* pnPropValue);</code>	Número inteiro
<code>INFA_STATUS INFA_CTGetExternalPropertyStringU(INFA_CT_HANDLE handle, INFA_UNICHAR* sPropName, INFA_BOOLEAN* pbPropValue);</code>	Booleano

O tipo de dados do valor de retorno é INFA_STATUS. Use INFA_SUCCESS e INFA_FAILURE para o valor de retorno.

Funções de Religação de Tipo de Dados

Você pode religar uma porta com tipo de dados diferente do tipo padrão com o PowerCenter. Use as funções de religação de tipo de dados se quiser que o procedimento acesse dados em um tipo de dados diferente do tipo padrão. Você deve religar a porta a um tipo de dados compatível.

Você só pode usar essas funções nas funções de inicialização.

Leve estas regras em consideração ao religar o tipo de dados de uma porta de entrada e saída:

- Você deve usar as funções de manipulação de dados para definir os dados e o indicador para a porta. Use as funções `INFA_CTSetData()` e `INFA_CTSetIndicator()` em modo com base em linha e a função `INFA_CTASetData()` em modo com base em matriz.
- Não chame a função `INFA_CTSetPassThruPort()` para a porta de saída.

A tabela a seguir lista tipos de dados compatíveis:

Tipo de Dados Padrão	Compatível com
Caractere	Unichar
Unichar	Caractere
Date	INFA_DATETIME Utilize a seguinte sintaxe: <pre>struct INFA_DATETIME { int nYear; int nMonth; int nDay; int nHour; int nMinute; int nSecond; int nNanoSecond; }</pre>
Dec18	Caractere, Unichar
Dec28	Caractere, Unichar

O PowerCenter fornece as seguintes funções de reassociação de tipo de dados:

- **INFA_CTRebindInputDataType()**. Reassocia a porta de entrada. Utilize a seguinte sintaxe:

```
INFA_STATUS INFA_CTRebindInputDataType(INFA_CT_INPUTPORT_HANDLE portHandle,
INFA_CDATATYPE datatype);
```
- **INFA_CTRebindOutputDataType()**. Religa a porta de saída. Utilize a seguinte sintaxe:

```
INFA_STATUS INFA_CTRebindOutputDataType(INFA_CT_OUTPUTPORT_HANDLE portHandle,
INFA_CDATATYPE datatype);
```

A tabela a seguir descreve os argumentos para esta função:

Argumento	Tipo de Dados	Entrada/ Saída	Descrição
portHandle	INFA_CT_OUTPUTPORT_HANDLE	Entrada	Manipulador de porta de saída.
tipo de dados	INFA_CDATATYPE	Entrada	<p>O tipo de dados com o qual você vincula novamente a porta. Use os seguintes valores para o parâmetro de tipo de dados:</p> <ul style="list-style-type: none"> - eINFA_CTYPE_SHORT - eINFA_CTYPE_INT32 - eINFA_CTYPE_CHAR - eINFA_CTYPE_RAW - eINFA_CTYPE_UNICHAR - eINFA_CTYPE_TIME - eINFA_CTYPE_FLOAT - eINFA_CTYPE_DOUBLE - eINFA_CTYPE_DECIMAL18_FIXED - eINFA_CTYPE_DECIMAL28_FIXED - eINFA_CTYPE_INFA_CTDATETIME

O tipo de dados do valor de retorno é INFA_STATUS. Use INFA_SUCCESS e INFA_FAILURE para o valor de retorno.

Funções de manipulação de dados (Modo Baseado em Linhas)

Quando o Serviço de Integração chama a função de notificação de linha de entrada, ela notifica o procedimento de que ele pode acessar uma linha ou bloco de dados. Entretanto, para obter dados da porta de entrada, modificá-los e definir os dados na porta de saída, é necessário usar as funções de manipulação de dados na função de notificação da linha de entrada. Quando o modo de acesso a dados é baseado em linha, use as funções de manipulação de dados baseados em linha.

Acrescente a função INFA_CTGetData<datatype>() para obter os dados provenientes da porta de entrada e a função INFA_CTSetData() para definir os dados na porta de saída. Acrescente a função INFA_CTGetIndicator() ou INFA_CTGetLength() se desejar que o procedimento verifique se a porta apresenta um valor nulo ou uma string vazia antes de você obter os dados.

O PowerCenter fornece as seguintes funções de manipulação de dados:

- INFA_CTGetData<datatype>()
- INFA_CTSetData()
- INFA_CTGetIndicator()
- INFA_CTSetIndicator()
- INFA_CTGetLength()
- INFA_CTSetLength()

Funções Obter Dados (Modo Baseado em Linhas)

Use as funções INFA_CTGetData<datatype>() para recuperar dados da porta especificada pela função.

É necessário modificar o nome da função, dependendo do tipo de dados da porta que você deseja que o procedimento acesse.

A tabela a seguir lista a sintaxe da função INFA_CTGetData<datatype>() e o tipo de dados do valor de retorno:

Sintaxe	Tipo de dados de Valor Retornado
<code>void* INFA_CTGetDataVoid(INFA_CT_INPUTPORT_HANDLE dataHandle);</code>	Ponteiro void de dados relacionado ao valor de retorno
<code>char* INFA_CTGetDataStringM(INFA_CT_INPUTPORT_HANDLE dataHandle);</code>	String (MBCS)
<code>IUNICHAR* INFA_CTGetDataStringU(INFA_CT_INPUTPORT_HANDLE dataHandle);</code>	String (Unicode)
<code>INFA_INT32 INFA_CTGetDataINT32(INFA_CT_INPUTPORT_HANDLE dataHandle);</code>	Número inteiro
<code>double INFA_CTGetDataDouble(INFA_CT_INPUTPORT_HANDLE dataHandle);</code>	Duplo
<code>INFA_CT_RAWDATE INFA_CTGetDataDate(INFA_CT_INPUTPORT_HANDLE dataHandle);</code>	Data bruta
<code>INFA_CT_RAWDEC18 INFA_CTGetDataRawDec18(INFA_CT_INPUTPORT_HANDLE dataHandle);</code>	BLOB decimal (precisão 18)
<code>INFA_CT_RAWDEC28 INFA_CTGetDataRawDec28(INFA_CT_INPUTPORT_HANDLE dataHandle);</code>	BLOB decimal (precisão 28)
<code>INFA_CT_DATETIME INFA_CTGetDataDateTime(INFA_CT_INPUTPORT_HANDLE dataHandle);</code>	Datetime

Função Definir Dados (Modo Baseado em Linha)

Use a função INFA_CTSetData() quando quiser que o procedimento passe um valor para uma porta de saída.

Utilize a seguinte sintaxe:

```
INFA_STATUS INFA_CTSetData(INFA_CT_OUTPUTPORT_HANDLE dataHandle, void* data);
```

O tipo de dados do valor de retorno é INFA_STATUS. Use INFA_SUCCESS e INFA_FAILURE para o valor de retorno.

Nota: Se você usar a função INFA_CTSetPassThruPort() em uma porta de entrada/saída, não utilize os dados nem o indicador dessa porta.

Funções do Indicador (Modo Baseado em Linhas)

Use as funções de indicador quando quiser que o procedimento obtenha o indicador para uma porta de entrada ou para definir o indicador para uma porta de saída. O indicador para a porta mostra se os dados são válidos, nulos ou truncados.

O PowerCenter fornece as seguintes funções de indicador:

- **INFA_CTGetIndicator()**. Obtém o indicador para uma porta de entrada. Utilize a seguinte sintaxe:

```
INFA_INDICATOR INFA_CTGetIndicator(INFA_CT_INPUTPORT_HANDLE dataHandle);
```

O tipo de dados do valor de retorno é INFA_INDICATOR. Use os seguintes valores para INFA_INDICATOR:

- **INFA_DATA_VALID**. Indica que os dados são válidos.
 - **INFA_NULL_DATA**. Indica um valor nulo.
 - **INFA_DATA_TRUNCATED**. Indica que os dados foram truncados.
- **INFA_CTSetIndicator()**. Define o indicador para uma porta de saída. Utilize a seguinte sintaxe:

```
INFA_STATUS INFA_CTSetIndicator(INFA_CT_OUTPUTPORT_HANDLE dataHandle, INFA_INDICATOR indicator);
```

A tabela a seguir descreve os argumentos para esta função:

Argumento	Tipo de Dados	Entrada/saída	Descrição
dataHandle	INFA_CT_OUTPUTPORT_HANDLE	Entrada	Manipulador de porta de saída.
indicador	INFA_INDICATOR	Entrada	O valor do indicador para a porta de saída. Use um destes valores: <ul style="list-style-type: none">- INFA_DATA_VALID. Indica que os dados são válidos.- INFA_NULL_DATA. Indica um valor nulo.- INFA_DATA_TRUNCATED. Indica que os dados foram truncados.

O tipo de dados do valor de retorno é INFA_STATUS. Use INFA_SUCCESS e INFA_FAILURE para o valor de retorno.

Nota: Se você usar a função INFA_CTSetPassThruPort() em uma porta de entrada/saída, não defina os dados ou o indicador para essa porta.

Funções de Extensão

Use as funções de extensão quando quiser que o procedimento acesse a extensão de uma porta de entrada de string ou binária, ou para definir a extensão de uma porta de saída de string ou binária.

Use as seguintes funções de extensão:

- **INFA_CTGetLength()**. Use essa função somente para portas binárias e de string. O Serviço de Integração retorna a extensão como o número de caracteres, o que inclui os espaços em branco. Utilize a seguinte sintaxe:

```
INFA_UINT32 INFA_CTGetLength(INFA_CT_INPUTPORT_HANDLE dataHandle);
```

O tipo de dados do valor retornado é INFA_UINT32. Use um valor entre zero e 2 GB para o valor de retorno.

- **INFA_CTSetLength()**. Quando a Transformação Personalizada contém uma porta de saída binária ou de string, é necessário usar esta função para definir a extensão dos dados, inclusive os espaços em branco. Verifique se a extensão que você define para portas binárias e de string não é maior que a precisão da porta. Se você definir o comprimento maior que a precisão da porta, obterá resultados inesperados. Por exemplo, pode ser que a sessão falhe.

Utilize a seguinte sintaxe:

```
INFA_STATUS INFA_CTSetLength(INFA_CT_OUTPUTPORT_HANDLE dataHandle, IUINT32 length);
```

O tipo de dados do valor de retorno é INFA_STATUS. Use INFA_SUCCESS e INFA_FAILURE para o valor de retorno.

Função Configurar Porta de Passagem

Use a função INFA_CTSetPassThruPort() quando quiser que o Serviço de Integração passe dados de uma porta de entrada para uma porta de saída sem modificá-los. Quando você usa a função INFA_CTSetPassThruPort(), o Serviço de Integração passará os dados à porta de saída depois de chamar a função de notificação da linha de entrada.

Considere as regras e diretrizes a seguir quando usar a função configurar porta de passagem:

- Só use essa função em uma função de inicialização.
- Se o procedimento incluir essa função, não inclua as funções INFA_CTSetData(), INFA_CTSetLength, INFA_CTSetIndicator() nem INFA_CTASetData() para passar dados à porta de saída.
- No modo com base em linhas, só se poderá incluir essa função se o escopo da transformação for Linha. Se o escopo da transformação for Transação ou Todos de entrada, a função retornará INFA_FAILURE.
- No modo com base em linhas, se você usar essa função para calcular várias linhas para determinada linha de entrada, cada linha de saída conterá os dados passados da porta de entrada.
- No modo baseado em matriz, você só pode usar essa função para transformações personalizadas passivas.

Verifique se o tipo de dados, precisão e escala são os mesmos para as portas de entrada e saída. O Serviço de Integração encerrará a sessão caso o tipo de dados, precisão ou escala não forem os mesmos para as porta de entrada e saída especificadas na função INFA_CTSetPassThruPort().

Utilize a seguinte sintaxe:

```
INFA_STATUS INFA_CTSetPassThruPort(INFA_CT_OUTPUTPORT_HANDLE outputport,  
INFA_CT_INPUTPORT_HANDLE inputport)
```

O tipo de dados do valor de retorno é INFA_STATUS. Use INFA_SUCCESS e INFA_FAILURE para o valor de retorno.

Função de Notificação de Saída

Quando desejar que o procedimento efetue a saída de uma linha no Serviço de Integração, use a função INFA_CTOutputNotification(). Inclua essa função apenas para transformações Personalizadas ativas. Para Transformações personalizadas passivas, o procedimento gera uma linha para o Serviço de Integração quando a função de notificação de linha de entrada fornece um valor de retorno. Se o procedimento chamar esta função para uma Transformação personalizada passiva, o Serviço de Integração ignora a função.

Nota: Quando o escopo de transformação for Linha, você poderá incluir essa função somente na função de notificação da linha de entrada. Se incluir em algum outro lugar, retornará uma falha.

Utilize a seguinte sintaxe:

```
INFA_ROWSTATUS INFA_CTOutputNotification(INFA_CT_OUTPUTGROUP_HANDLE group);
```

A tabela a seguir descreve o argumento para esta função:

Argumento	Tipo de Dados	Entrada/ Saída	Descrição
grupo	INFA_CT_OUTPUT_GROUP_HANDLE	Entrada	Manipulador de grupo de saída.

O tipo de dados do valor retornado é INFA_ROWSTATUS. Use os seguintes valores para o valor retornado:

- **INFA_ROWSUCCESS.** Indica a função que processou com êxito a linha de dados.
- **INFA_ROWERROR.** Indica a função que encontrou um erro na linha de dados. O Serviço de Integração incrementa a contagem de erros internos.
- **INFA_FATALERROR.** Indica que a função encontrou um erro fatal relativo à linha de dados. O Serviço de Integração causa falha na sessão.

Nota: Quando o código de procedimento chama a função INFA_CTOutputNotification(), é necessário verificar que todos os ponteiros de um identificador de porta de saída indicam dados válidos. Quando um ponteiro não apontar para dados válidos, o Serviço de Integração poderá encerrar inesperadamente.

Função de Notificação de Saída de Limite de Dados

Inclua a função INFA_CTDataBdryOutputNotification() quando quiser gerar o procedimento que efetua a saída de uma transação de comprometimento ou reversão.

Ao usar esta função, é necessário selecionar a propriedade de Transações Gerais para esta transformação Personalizada. Se esta propriedade não for selecionada, o Serviço de Integração falha na sessão.

Utilize a seguinte sintaxe:

```
INFA_STATUS INFA_CTDataBdryOutputNotification(INFA_CT_PARTITION_HANDLE handle,  
INFA_CTDataBdryType dataBoundaryType);
```

A tabela a seguir descreve os argumentos para esta função:

Argumento	Tipo de Dados	Entrada/ Saída	Descrição
manipulador	INFA_CT_PARTITION_HANDLE	Entrada	Nome do manipulador.
dataBoundaryType	INFA_CTDataBdryType	Entrada	O tipo de transação. Use os seguintes valores para o parâmetro dataBoundaryType: - eBT_COMMIT - eBT_ROLLBACK

O tipo de dados do valor de retorno é INFA_STATUS. Use INFA_SUCCESS e INFA_FAILURE para o valor de retorno.

Funções de Erro

Use as funções de erro para acessar erros de procedimento. O Serviço de Integração retorna os erros mais recentes.

O PowerCenter fornece as seguintes funções de erro:

- **INFA_CTGetErrorMsgM().** Obtém a mensagem de erro em MBCS. Utilize a seguinte sintaxe:

```
const char* INFA_CTGetErrorMsgM();
```
- **INFA_CTGetErrorMsgU().** Obtém a mensagem de erro em Unicode. Utilize a seguinte sintaxe:

```
const IUNICHAR* INFA_CTGetErrorMsgU();
```

Funções de Mensagem do Log da Sessão

Utilize as funções de mensagem do log da sessão quando desejar que o procedimento registre uma mensagem no log da sessão em Unicode ou MBCS.

O PowerCenter fornece as seguintes funções de mensagem do log de sessão:

- **INFA_CTLogMessageU()**. Registra uma mensagem em Unicode.

Utilize a seguinte sintaxe:

```
void INFA_CTLogMessageU(INFA_CT_ErrorSeverityLevel errorSeverityLevel, INFA_UNICHAR* msg)
```

A tabela a seguir descreve os argumentos para esta função:

Argumento	Tipo de Dados	Entrada/Saída	Descrição
errorSeverityLevel	INFA_CT_ErrorSeverityLevel	Entrada	O nível de gravidade da mensagem de erro que o Serviço de Integração deverá gravar no log da sessão. Use os seguintes valores para o parâmetro errorSeverityLevel: <ul style="list-style-type: none">- eESL_LOG- eESL_DEBUG- eESL_ERROR
msg	INFA_UNICHAR*	Entrada	Insira o texto da mensagem em Unicode entre aspas.

- **INFA_CTLogMessageM()**. Registra uma mensagem em MBCS.

Utilize a seguinte sintaxe:

```
void INFA_CTLogMessageM(INFA_CT_ErrorSeverityLevel errorSeverityLevel, char* msg)
```

A tabela a seguir descreve os argumentos para esta função:

Argumento	Tipo de Dados	Entrada/Saída	Descrição
errorSeverityLevel	INFA_CT_ErrorSeverityLevel	Entrada	O nível de gravidade da mensagem de erro que o Serviço de Integração deverá gravar no log da sessão. Use os seguintes valores para o parâmetro errorSeverityLevel: <ul style="list-style-type: none">- eESL_LOG- eESL_DEBUG- eESL_ERROR
msg	caractere*	Entrada	Insira o texto da mensagem em MBCS entre aspas.

Função de Incremento de Contagem de Erros

Use a função `INFA_CTIncrementErrorCount()` quando deseja aumentar a contagem de erros para a sessão.

Utilize a seguinte sintaxe:

```
INFA_STATUS INFA_CTIncrementErrorCount(INFA_CT_PARTITION_HANDLE transformation, size_t nErrors, INFA_STATUS* pStatus);
```

A tabela a seguir descreve os argumentos para esta função:

Argumento	Tipo de Dados	Entrada/ Saída	Descrição
transformação	INFA_CT_PARTITION_HANDLE	Entrada	Manipulador de partição.
nErrors	size_t	Entrada	O Serviço de Integração incrementa a contagem de erros em nErrors para a instância de transformação específica.
pStatus	INFA_STATUS*	Entrada	O Serviço de Integração usa INFA_FAILURE para o parâmetro pStatus quando a contagem de erros excede o limite e provoca a falha da sessão.

O tipo de dados do valor de retorno é INFA_STATUS. Use INFA_SUCCESS e INFA_FAILURE para o valor de retorno.

Função Encerrado

Use a função INFA_CTIsTerminated() quando deseja que o procedimento verifique se o Cliente do PowerCenter solicitou ao Serviço de Integração para interromper a sessão. Você pode chamar esta função se o procedimento incluir um processo demorado.

Utilize a seguinte sintaxe:

```
INFA_CTTerminateType INFA_CTIsTerminated(INFA_CT_PARTITION_HANDLE handle);
```

A tabela a seguir descreve o argumento para esta função:

Argumento	Tipo de Dados	Entrada/ Saída	Descrição
manipulador	INFA_CT_PARTITION_HANDLE	entrada	Manipulador de partição.

O tipo de dados do valor retornado é INFA_CTTerminateType. O Serviço de Integração retorna um dos seguintes valores:

- **eTT_NOTTERMINATED.** Indica que o Cliente do PowerCenter não solicitou a interrupção da sessão.
- **eTT_ABORTED.** Indica que o Serviço de Integração abortou a sessão.
- **eTT_STOPPED.** Indica que o Serviço de Integração falhou na sessão.

Funções de Bloqueio

Quando a transformação Personalizada contém múltiplos grupos de entrada, você pode escrever o código para bloquear os dados de entrada em um grupo de entrada.

Considere as seguintes regras quando usar as funções de bloqueio:

- Você pode bloquear no máximo $n-1$ grupos de entrada.

- Você não pode bloquear um grupo de entrada que já esteja bloqueado.
- Você não pode bloquear um grupo de entrada quando ele recebe dados da mesma origem que outro grupo de entrada.
- Você não pode desbloquear um grupo de entrada que já esteja desbloqueado.

O PowerCenter fornece as seguintes funções de bloqueio:

- **INFA_CTBlockInputFlow()**. Permite que o procedimento bloqueie um grupo de entrada.

Utilize a seguinte sintaxe:

```
INFA_STATUS INFA_CTBlockInputFlow(INFA_CT_INPUTGROUP_HANDLE group);
```

- **INFA_CTUnblockInputFlow()**. Permite que o procedimento desbloqueie um grupo de entrada.

Utilize a seguinte sintaxe:

```
INFA_STATUS INFA_CTUnblockInputFlow(INFA_CT_INPUTGROUP_HANDLE group);
```

A tabela a seguir descreve o argumento para esta função:

Argumento	Tipo de Dados	Entrada/ Saída	Descrição
grupo	INFA_CT_INPUTGROUP_HANDLE	Entrada	Manipulador de grupo de entrada.

O tipo de dados do valor de retorno é INFA_STATUS. Use INFA_SUCCESS e INFA_FAILURE para o valor de retorno.

Verificar Bloqueio

Ao usar as funções INFA_CTBlockInputFlow() e INFA_CTUnblockInputFlow() no código de procedimento, certifique-se de que o procedimento verifica se o Serviço de Integração permite a transformação Personalizada bloquear dados recebidos. Para tanto, verifique o valor de INFA_CT_TRANS_MAY_BLOCK_DATA propID usando a função INFA_CTGetInternalPropertyBool().

Quando o valor de INFA_CT_TRANS_MAY_BLOCK_DATA propID for FALSE, o procedimento não deve usar as funções bloqueadas, ou deve retornar um erro fatal e interromper a sessão.

Se o código do procedimento usar as funções bloqueadas quando o Serviço de Integração não permitir a transformação Personalizada bloquear dados, o Serviço de Integração poderá encerrar a sessão.

Funções de Ponteiro

Use as funções de ponteiro quando quiser que o Serviço de Integração crie e acesse ponteiros para um objeto ou uma estrutura.

O PowerCenter fornece as seguintes funções de ponteiro:

- **INFA_CTGetUserDefinedPtr()**. Permite ao procedimento acessar um objeto ou estrutura durante o tempo de execução.

Utilize a seguinte sintaxe:

```
void* INFA_CTGetUserDefinedPtr(INFA_CT_HANDLE handle)
```

A tabela a seguir descreve o argumento para esta função:

Argumento	Tipo de Dados	Entrada/ Saída	Descrição
manipulador	INFA_CT_HANDLE	Entrada	Nome do manipulador.

- **INFA_CTSetUserDefinedPtr()**. Permite ao procedimento associar um objeto ou uma estrutura a qualquer indicador que o Serviço de Integração fornecer. Para reduzir a sobrecarga de processamento, inclua essa função nas funções de inicialização.

Utilize a seguinte sintaxe:

```
void INFA_CTSetUserDefinedPtr(INFA_CT_HANDLE handle, void* pPtr)
```

A tabela a seguir descreve os argumentos para esta função:

Argumento	Tipo de Dados	Entrada/ Saída	Descrição
manipulador	INFA_CT_HANDLE	Entrada	Nome do manipulador.
pPtr	void*	Entrada	Ponteiro do usuário.

Você deve substituir um indicador válido por INFA_CT_HANDLE.

Função Alterar modo de string

Quando o Serviço de Integração executa o modo Unicode, ele transfere dados ao procedimento em UCS-2 por padrão. Quando é executado em modo ASCII, ele transfere dados em ASCII por padrão. Use a função `INFA_CTChangeStringMode()` se desejar alterar o modo de string padrão para o procedimento. Ao alterar o modo padrão de string para MBCS, o Serviço de Integração transfere dados para a página de códigos do Serviço de Integração. Use a função `INFA_CTSetDataCodePageID()` se desejar alterar a página de códigos.

Quando um procedimento incluir a função `INFA_CTChangeStringMode()`, o Serviço de Integração altera o modo de string para todas as portas em cada transformação Personalizada que utilize este procedimento em especial.

Use a função de modo de alteração de string nas funções de inicialização.

Utilize a seguinte sintaxe:

```
INFA_STATUS INFA_CTChangeStringMode(INFA_CT_PROCEDURE_HANDLE procedure,  
INFA_CTStringMode stringMode);
```

A tabela a seguir descreve os argumentos para esta função:

Argumento	Tipo de Dados	Entrada/ Saída	Descrição
procedimento	INFA_CT_PROCEDURE_HANDLE	Entrada	Nome do manipulador de procedimento.
stringMode	INFA_CTStringMode	Entrada	Especifica o modo de string que você deseja que o Serviço de Integração use. Use os seguintes valores para o parâmetro stringMode: <ul style="list-style-type: none"> - eASM_UNICODE. Use isto quando o Serviço de Integração for executado em modo ASCII e você deseja que o procedimento acesse dados em Unicode. - eASM_MBCS. Use esta opção quando o Serviço de Integração for executado no modo Unicode e você deseja que o procedimento acesse dados em MBCS.

O tipo de dados do valor de retorno é INFA_STATUS. Use INFA_SUCCESS e INFA_FAILURE para o valor de retorno.

Função Definir Página de Código de Dados

Use INFA_CTSetDataCodePageID() quando quiser que o Serviço de Integração passe dados à transformação Personalizada em uma página de código diferente da página de código do Serviço de Integração.

Use a função definir página de código de dados na função de inicialização do procedimento.

Utilize a seguinte sintaxe:

```
INFA_STATUS INFA_CTSetDataCodePageID(INFA_CT_TRANSFORMATION_HANDLE transformation, int dataCodePageID);
```

A tabela a seguir descreve os argumentos para esta função:

Argumento	Tipo de Dados	Entrada/ Saída	Descrição
transformação	INFA_CT_TRANSFORMATION_HANDLE	Entrada	Nome do manipulador de transformação.
dataCodePageID	inteiro	Entrada	Especifica a página de código na qual você quer que o Serviço de Integração passe dados. Para obter os valores válidos do parâmetro dataCodePageID, consulte "Páginas de Código" no <i>Guia do Administrador</i> .

O tipo de dados do valor de retorno é INFA_STATUS. Use INFA_SUCCESS e INFA_FAILURE para o valor de retorno.

Funções de Estratégia de Linha (Modo Baseado em Linha)

As funções de estratégia baseadas em linhas permitem que você acesse e configure a estratégia de atualização para cada linha.

O PowerCenter fornece as seguintes funções de estratégia de linha:

- **INFA_CTGetRowStrategy()**. Permite que o procedimento obtenha a estratégia de atualização de uma linha.

Utilize a seguinte sintaxe:

```
INFA_STATUS INFA_CTGetRowStrategy(INFA_CT_INPUTGROUP_HANDLE group,  
INFA_CTUpdateStrategy updateStrategy);
```

A tabela a seguir descreve os argumentos para esta função:

Argumento	Tipo de Dados	Entrada/ Saída	Descrição
grupo	INFA_CT_INPUTGROUP_HANDLE	Entrada	Manipulador de grupo de entrada.
updateStrategy	INFA_CT_UPDATESTRATEGY	Entrada	Estratégia de atualização da porta de entrada. O Serviço de Integração utiliza os seguintes valores: <ul style="list-style-type: none">- eUS_INSERT = 0- eUS_UPDATE = 1- eUS_DELETE = 2- eUS_REJECT = 3

- **INFA_CTSetRowStrategy()**. Define a estratégia de atualização de cada linha. Substitui a função INFA_CTChangeDefaultRowStrategy.

Utilize a seguinte sintaxe:

```
INFA_STATUS INFA_CTSetRowStrategy(INFA_CT_OUTPUTGROUP_HANDLE group,  
INFA_CT_UPDATESTRATEGY updateStrategy);
```

A tabela a seguir descreve os argumentos para esta função:

Argumento	Tipo de Dados	Entrada/ Saída	Descrição
grupo	INFA_CT_OUTPUTGROUP_HANDLE	Entrada	Manipulador de grupo de saída.
updateStrategy	INFA_CT_UPDATESTRATEGY	Entrada	Estratégia de atualização que você quer definir para a porta de saída. Use um destes valores: <ul style="list-style-type: none">- eUS_INSERT = 0- eUS_UPDATE = 1- eUS_DELETE = 2- eUS_REJECT = 3

O tipo de dados do valor de retorno é INFA_STATUS. Use INFA_SUCCESS e INFA_FAILURE para o valor de retorno.

Função Alterar Estratégia de Linha Padrão

Por padrão, a estratégia de linha de uma transformação Personalizada é de passagem quando o escopo de transformação for Bruto. Quando o escopo de transformação for Transação ou Entrada de Todos, a estratégia de linha terá, por padrão, o mesmo valor que a propriedade de sessão Tratar Linhas de Origem Como.

Por exemplo, em um mapeamento, há uma Transformação Estratégia de Atualização seguida de transformação personalizada com escopo de transformação de linha. A Transformação Estratégia de Atualização sinaliza as linhas a serem atualizadas, inseridas ou excluídas. Quando o Serviço de Integração transfere uma linha à transformação personalizada, a transformação Personalizada retém a sinalização, já que sua estratégia de linha é de passagem.

Entretanto, é possível alterar a estratégia de linha de transformação personalizada com o PowerCenter. Use a função `INFA_CTChangeDefaultRowStrategy()` para alterar a estratégia de linha padrão no nível de transformação. Por exemplo, ao alterar a estratégia de linha padrão de uma transformação personalizada a ser inserida, o Serviço de Integração sinaliza todas as linhas que passam por esta transformação para inserção.

Nota: O Serviço de Integração retorna `INFA_FAILURE` se a sessão não estiver em modo controlado por dados.

Utilize a seguinte sintaxe:

```
INFA_STATUS INFA_CTChangeDefaultRowStrategy(INFA_CT_TRANSFORMATION_HANDLE  
transformation, INFA_CT_DefaultUpdateStrategy defaultUpdateStrategy);
```

A tabela a seguir descreve os argumentos para esta função:

Argumento	Tipo de Dados	Entrada/ Saída	Descrição
transformação	INFA_CT_TRANSFORMATION_HANDLE	Entrada	Manipulador de transformação.
defaultUpdateStrategy	INFA_CT_DefaultUpdateStrategy	Entrada	Especifica a estratégia de linha que você deseja que o Serviço de Integração use para a transformação Personalizada. <ul style="list-style-type: none">- <code>eDUS_PASSTHROUGH</code>. Sinaliza a linha para passagem.- <code>eDUS_INSERT</code>. Sinaliza linhas para inserção.- <code>eDUS_UPDATE</code>. Sinaliza linhas para atualização.- <code>eDUS_DELETE</code>. Sinaliza linhas para exclusão.

O tipo de dados do valor de retorno é `INFA_STATUS`. Use `INFA_SUCCESS` e `INFA_FAILURE` para o valor de retorno.

Funções de API baseadas em matriz

As funções baseadas em matriz são funções API que são utilizadas ao alterar o modo de acesso aos dados para baseado em matriz.

A Informatica fornece os seguintes grupos de funções API baseadas em matriz:

- Número máximo de linhas
- Número de linhas
- É válido para linhas
- Manipulação de dados (modo baseado em matriz)
- Estratégia de linha
- Definir linha de erro de entrada

Funções de Número Máximo de Linhas

Por padrão, o Serviço de Integração permite um número máximo de linhas em um bloco de entrada e de saída. No entanto, você pode alterar o número máximo de linhas permitidas em um bloco de saída.

Use as funções do `INFA_CTAGetInputNumRowsMax()` e `INFA_CTAGetOutputNumRowsMax()` para determinar o número máximo de linhas em blocos de entrada e saída. use os valores que essas funções retornam para determinar o tamanho do buffer se o procedimento necessita de um buffer.

Você pode definir um número máximo de linhas de bloco de saída usando a função `INFA_CTASetOutputRowMax()`. Você pode usar essa função se você deseja que o processo utilizar um buffer maior ou menor.

Essas funções só podem ser chamadas em uma função de inicialização.

O PowerCenter fornece as seguintes funções para determinar e definir o número máximo de linhas em blocos:

- **`INFA_CTAGetInputNumRowsMax()`**. Use esta função para determinar o número máximo de linhas permitidas em um bloco de entrada.

Utilize a seguinte sintaxe:

```
IINT32 INFA_CTAGetInputRowMax( INFA_CT_INPUTGROUP_HANDLE inputgroup );
```

A tabela a seguir descreve o argumento para esta função:

Argumento	Tipo de Dados	Entrada/ Saída	Descrição
inputgroup	INFA_CT_INPUTGROUP_HANDLE	Entrada	Manipulador de grupo de entrada.

- **`INFA_CTAGetOutputNumRowsMax()`**. Use essa função para determinar o número máximo de linhas permitidas em um bloco de saída.

Utilize a seguinte sintaxe:

```
IINT32 INFA_CTAGetOutputRowMax( INFA_CT_OUTPUTGROUP_HANDLE outputgroup );
```

A tabela a seguir descreve o argumento para esta função:

Argumento	Tipo de Dados	Entrada/ Saída	Descrição
outputgroup	INFA_CT_OUTPUTGROUP_HANDLE	Entrada	Manipulador de grupo de saída.

- **INFA_CTASetOutputRowMax()**. Use essa função para definir o número máximo de linhas permitidas em um bloco de saída.

Utilize a seguinte sintaxe:

```
INFA_STATUS INFA_CTASetOutputRowMax( INFA_CT_OUTPUTGROUP_HANDLE outputgroup,
INFA_INT32 nRowMax );
```

A tabela a seguir descreve os argumentos para esta função:

Argumento	Tipo de Dados	Entrada/ Saída	Descrição
outputgroup	INFA_CT_OUTPUTGROUP_HANDLE	Entrada	Manipulador de grupo de saída.
nRowMax	INFA_INT32	Entrada	O número máximo de linhas permitidas em um bloco de saída. Você deve digitar um número positivo. A função retorna um erro fatal ao usar um número não positivo, incluindo zero.

Funções de Número de Linhas

Use as funções de número de linhas para determinar os número de linhas em um bloco de entrada ou para definir o número de linhas em um bloco de saída para o grupo de entrada ou de saída especificado.

O PowerCenter fornece a seguinte série de funções de linha:

- **INFA_CTAGetNumRows()**. É possível determinar o número de linhas em um bloco de entrada.

Utilize a seguinte sintaxe:

```
INFA_INT32 INFA_CTAGetNumRows( INFA_CT_INPUTGROUP_HANDLE inputgroup );
```

A tabela a seguir descreve o argumento para esta função:

Argumento	Tipo de Dados	Entrada/ Saída	Descrição
inputgroup	INFA_CT_INPUTGROUP_HANDLE	Entrada	Manipulador de grupo de entrada.

- **INFA_CTASetNumRows()**. É possível determinar o número de linhas em um bloco de saída. Chame esta função antes de chamar a função de notificação de saída.

Utilize a seguinte sintaxe:

```
void INFA_CTASetNumRows( INFA_CT_OUTPUTGROUP_HANDLE outputgroup, INFA_INT32 nRows );
```

A tabela a seguir descreve os argumentos para esta função:

Argumento	Tipo de Dados	Entrada/ Saída	Descrição
outputgroup	INFA_CT_OUTPUTGROUP_HANDLE	Entrada	Manipulador de porta de saída.
nRows	INFA_INT32	Entrada	Número de linhas que você deseja definir no bloco de saída. Você deve digitar um número positivo. O Serviço de Integração não consegue efetuar a função de notificação de saída ao especificar um número não positivo.

Função Válido Para Linhas

Algumas linhas em um bloco podem ser de descarte, filtro ou erro. Use a função `INFA_CTIsRowValid()` para determinar se uma linha em um bloco é válida. Esta função retorna `INFA_TRUE` quando uma linha é válida.

Utilize a seguinte sintaxe:

```
INFA_BOOLEAN INFA_CTIsRowValid( INFA_CT_INPUTGROUP_HANDLE inputgroup, INFA_INT32 iRow);
```

A tabela a seguir descreve os argumentos para esta função:

Argumento	Tipo de Dados	Entrada/ Saída	Descrição
inputgroup	INFA_CT_INPUTGROUP_HANDLE	Entrada	Manipulador de grupo de entrada.
iRow	INFA_INT32	Entrada	Número de índice da linha no bloco. O índice baseia-se em zero. Verifique se o procedimento passa somente um número de índice que existe no bloco de dados. Se você passar um valor inválido, o Serviço de Integração será fechado inesperadamente.

Funções de Manipulação de Dados (Modo Baseado em Matriz)

Quando o Serviço de Integração chama a função `p_<proc_name>_inputRowNotification()`, ele informa ao procedimento que ele pode acessar uma linha ou bloco de dados. Entretanto, para obter dados da porta de entrada, modificá-los e definir os dados na porta de saída em modo baseado em matriz, é necessário usar as funções de manipulação de dados baseada em matriz na função de notificação da linha de entrada.

Inclua a função `INFA_CTGetData<datatype>()` para obter os dados da porta de entrada e a função `INFA_CTSetData()` para definir os dados na porta de saída. Inclua a função `INFA_CTGetIndicator()` se desejar que o procedimento verifique se a porta apresenta um valor nulo ou uma string vazia antes de você obter os dados.

O PowerCenter fornece as seguintes funções de manipulação de dados para o modo de acesso a dados baseado em matriz:

- `INFA_CTGetData<datatype>()`

- INFA_CTGetIndicator()
- INFA_CTSetData()

Funções Obter Dados (Modo Baseado em Matriz)

Use as funções INFA_CTGetData<datatype>() para recuperar dados da porta especificada pela função. É necessário modificar o nome da função, dependendo do tipo de dados da porta que você deseja que o procedimento acesse. O Serviço de Integração transfere o comprimento dos dados nas funções de obtenção de dados baseadas em matriz.

A tabela a seguir lista a sintaxe da função INFA_CTGetData<datatype>() e o tipo de dados do valor de retorno:

Sintaxe	Tipo de dados de Valor Retornado
<code>void* INFA_CTGetDataVoid(INFA_CT_INPUTPORT_HANDLE inputport, INFA_INT32 iRow, INFA_UINT32* pLength);</code>	Ponteiro void de dados relacionado ao valor de retorno
<code>char* INFA_CTGetDataStringM(INFA_CT_INPUTPORT_HANDLE inputport, INFA_INT32 iRow, INFA_UINT32* pLength);</code>	String (MBCS)
<code>IUNICHAR* INFA_CTGetDataStringU(INFA_CT_INPUTPORT_HANDLE inputport, INFA_INT32 iRow, INFA_UINT32* pLength);</code>	String (Unicode)
<code>INFA_INT32 INFA_CTGetDataINT32(INFA_CT_INPUTPORT_HANDLE inputport, INFA_INT32 iRow);</code>	Número inteiro
<code>double INFA_CTGetDataDouble(INFA_CT_INPUTPORT_HANDLE inputport, INFA_INT32 iRow);</code>	Duplo
<code>INFA_CT_RAWDATETIME INFA_CTGetDataRawDate(INFA_CT_INPUTPORT_HANDLE inputport, INFA_INT32 iRow);</code>	Data bruta
<code>INFA_CT_DATETIME INFA_CTGetDataDateTime(INFA_CT_INPUTPORT_HANDLE inputport, INFA_INT32 iRow);</code>	Datetime
<code>INFA_CT_RAWDEC18 INFA_CTGetDataRawDec18(INFA_CT_INPUTPORT_HANDLE inputport, INFA_INT32 iRow);</code>	BLOB decimal (precisão 18)
<code>INFA_CT_RAWDEC28 INFA_CTGetDataRawDec28(INFA_CT_INPUTPORT_HANDLE inputport, INFA_INT32 iRow);</code>	BLOB decimal (precisão 28)

Função Obter Indicador (Modo Baseado em Matriz)

Use a função de indicador de obtenção quando desejar que o procedimento verifique se a porta de entrada tem um valor nulo.

Utilize a seguinte sintaxe:

```
INFA_INDICATOR INFA_CTGetIndicator( INFA_CT_INPUTPORT_HANDLE inputport, INFA_INT32 iRow );
```

A tabela a seguir descreve os argumentos para esta função:

Argumento	Tipo de Dados	Entrada/ Saída	Descrição
inputport	INFA_CT_INPUTPORT_HANDLE	Entrada	Manipulador de porta de entrada.
iRow	INFA_INT32	Entrada	Número de índice da linha no bloco. O índice baseia-se em zero. Verifique se o procedimento passa somente um número de índice que existe no bloco de dados. Se você passar um valor inválido, o Serviço de Integração será fechado inesperadamente.

O tipo de dados do valor de retorno é INFA_INDICATOR. Use os seguintes valores para INFA_INDICATOR:

- **INFA_DATA_VALID.** Indica que os dados são válidos.
- **INFA_NULL_DATA.** Indica um valor nulo.
- **INFA_DATA_TRUNCATED.** Indica que os dados foram truncados.

Função Definir Dados (Modo Baseado em Matriz)

Utilize a função Definir Dados quando quiser que o procedimento passe um valor para uma porta de saída. Você pode definir os dados, o tamanho deles (se aplicável) e o indicador da porta de saída que especificar. Você não utiliza funções separadas para definir o tamanho ou o indicador da porta de saída.

Utilize a seguinte sintaxe:

```
void INFA_CTASetData( INFA_CT_OUTPUTPORT_HANDLE outputport, INFA_INT32 iRow, void* pData, INFA_UINT32 nLength, INFA_INDICATOR indicator);
```

A tabela a seguir descreve os argumentos para esta função:

Argumento	Tipo de Dados	Entrada/ Saída	Descrição
outputport	INFA_CT_OUTPUTPORT_HANDLE	Entrada	Manipulador de porta de saída.
iRow	INFA_INT32	Entrada	Número de índice da linha no bloco. O índice baseia-se em zero. Verifique se o procedimento passa somente um número de índice que existe no bloco de dados. Se você passar um valor inválido, o Serviço de Integração será fechado inesperadamente.
pData	void*	Entrada	Ponteiro para os dados.

Argumento	Tipo de Dados	Entrada/ Saída	Descrição
nLength	INFA_UINT32	Entrada	Comprimento da porta. Utilize somente para portas binárias e de string. Verifique se a função passa o comprimento correto dos dados. Se a função passar outro comprimento, a função de notificação de saída retornará falha para essa porta. Verifique se o comprimento que você definir para portas binárias e de string não é maior que a precisão da porta. Se você definir o tamanho maior que a precisão da porta, obterá resultados inesperados. Por exemplo, pode ser que a sessão falhe.
indicador	INFA_INDICATOR	Entrada	Valor do indicador da porta de saída. Use um destes valores: <ul style="list-style-type: none"> - INFA_DATA_VALID. Indica que os dados são válidos. - INFA_NULL_DATA. Indica um valor nulo. - INFA_DATA_TRUNCATED. Indica que os dados foram truncados.

Funções de Estratégia de Linha (Modo Baseado em Matriz)

As funções de estratégia de linha baseadas em matriz permitem que você acesse e configure a estratégia de atualização de cada linha de um bloco.

O PowerCenter fornece as seguintes funções de estratégia de linha:

- **INFA_CTGetRowStrategy()**. Permite que o procedimento obtenha a estratégia de atualização de uma linha em um bloco.

Utilize a seguinte sintaxe:

```
INFA_CT_UPDATESTRATEGY INFA_CTGetRowStrategy( INFA_CT_INPUTGROUP_HANDLE inputgroup,
INFA_INT32 iRow);
```

A tabela a seguir descreve os argumentos para esta função:

Argumento	Tipo de Dados	Entrada/ Saída	Descrição
inputgroup	INFA_CT_INPUTGROUP_HANDLE	Entrada	Manipulador de grupo de entrada.
iRow	INFA_INT32	Entrada	Número de índice da linha no bloco. O índice baseia-se em zero. Verifique se o procedimento passa somente um número de índice que existe no bloco de dados. Se você passar um valor inválido, o Serviço de Integração será fechado inesperadamente.

- **INFA_CTSetRowStrategy()**. Define a estratégia de atualização de uma linha em um bloco.

Utilize a seguinte sintaxe:

```
void INFA_CTASetRowStrategy( INFA_CT_OUTPUTGROUP_HANDLE outputgroup, INFA_INT32 iRow,  
INFA_CT_UPDATESTRATEGY updateStrategy );
```

A tabela a seguir descreve os argumentos para esta função:

Argumento	Tipo de Dados	Entrada/ Saída	Descrição
outputgroup	INFA_CT_OUTPUTGROUP_HANDLE	Entrada	Manipulador de grupo de saída.
iRow	INFA_INT32	Entrada	Número de índice da linha no bloco. O índice baseia-se em zero. Verifique se o procedimento passa somente um número de índice que existe no bloco de dados. Se você passar um valor inválido, o Serviço de Integração será fechado inesperadamente.
updateStrategy	INFA_CT_UPDATESTRATEGY	Entrada	Estratégia de atualização da porta. Use um destes valores: - eUS_INSERT = 0 - eUS_UPDATE = 1 - eUS_DELETE = 2 - eUS_REJECT = 3

Funções Definir Linha de Erros de Entrada

Quando você usa o modo de acesso com base na matriz, não é possível retornar INFA_ROWERROR na função de notificação da linha de entrada. Em vez disso, use as funções da linha de erros de entrada para notificar o Serviço de Integração sobre um erro em determinada linha de entrada.

O PowerCenter fornece as seguintes funções de definição de linhas de entrada no modo com base na matriz:

- **INFA_CTASetInputErrorRowM()**. Você pode notificar o Serviço de Integração sobre um erro em uma linha do bloco de entrada e para que ele emita uma mensagem de erro MBCS ao log da sessão.

Utilize a seguinte sintaxe:

```
INFA_STATUS INFA_CTASetInputErrorRowM( INFA_CT_INPUTGROUP_HANDLE inputGroup,  
INFA_INT32 iRow, size_t nErrors, INFA_MBCSCHAR* sErrMsg );
```

A tabela a seguir descreve os argumentos para esta função:

Argumento	Tipo de Dados	Entrada/ Saída	Descrição
inputGroup	INFA_CT_INPUTGROUP_HANDLE	Entrada	Manipulador de grupo de entrada.
iRow	INFA_INT32	Entrada	Número de índice da linha no bloco. O índice baseia-se em zero. Verifique se o procedimento passa somente um número de índice que existe no bloco de dados. Se você passar um valor inválido, o Serviço de Integração será fechado inesperadamente.
nErrors	size_t	Entrada	Use este parâmetro para especificar o número de erros causados por essa linha de entrada.
sErrMsg	INFA_MBCSCHAR*	Entrada	A string MBCS contendo a mensagem de erro que você quer que a função emita. Você deve digitar uma string terminada em nulo. Esse parâmetro é opcional. Quando você inclui esse argumento, o Serviço de Integração imprime a mensagem no log da sessão mesmo quando você ativa o log de erros de linha.

- **INFA_CTSetInputErrorRowU()**. Você pode notificar o Serviço de Integração de que uma linha do bloco de entrada contém um erro, e produzir uma mensagem de erro Unicode para o log de sessão.

Utilize a seguinte sintaxe:

```
INFA_STATUS INFA_CTSetInputErrorRowU( INFA_CT_INPUTGROUP_HANDLE inputGroup,
INFA_INT32 iRow, size_t nErrors, INFA_UNICHAR* sErrMsg );
```

A tabela a seguir descreve os argumentos para esta função:

Argumento	Tipo de Dados	Entrada/ Saída	Descrição
inputGroup	INFA_CT_INPUTGROUP_HANDLE	Entrada	Manipulador de grupo de entrada.
iRow	INFA_INT32	Entrada	Número de índice da linha no bloco. O índice baseia-se em zero. Verifique se o procedimento passa somente um número de índice que existe no bloco de dados. Se você passar um valor inválido, o Serviço de Integração será fechado inesperadamente.

Argumento	Tipo de Dados	Entrada/ Saída	Descrição
nErrors	size_t	Entrada	Use este parâmetro para especificar o número de erros causados por essa linha de saída.
sErrMsg	INFA_UNICHAR*	Entrada	<p>String Unicode que contém a mensagem de erro que a função deve gerar. Você deve digitar uma string terminada em nulo.</p> <p>Esse parâmetro é opcional. Quando você inclui esse argumento, o Serviço de Integração imprime a mensagem no log da sessão mesmo quando você ativa o log de erros de linha.</p>

CAPÍTULO 5

Transformação Mascaramento de Dados

Este capítulo inclui os seguintes tópicos:

- [transformação de Mascaramento de Dados, 130](#)
- [Propriedades de mascaramento, 131](#)
- [Mascaramento de Chave, 133](#)
- [Mascaramento de Substituição, 135](#)
- [Mascaramento dependente, 139](#)
- [Mascaramento Aleatório, 141](#)
- [Aplicando Regras de Mascaramento, 143](#)
- [Mascaramento de Expressão, 147](#)
- [Formatos Especiais de Máscara, 149](#)
- [Mascaramento de CPF, 150](#)
- [Mascaramento de Número de Cartão de Crédito, 151](#)
- [Mascaramento de Número de Telefone, 152](#)
- [Mascaramento de Endereço de E-mail, 152](#)
- [Mascaramento de Número de Previdência Social, 154](#)
- [Mascaramento de Endereço IP, 154](#)
- [Mascaramento de Endereço de URL, 155](#)
- [Arquivo de valor padrão, 155](#)
- [Propriedades da sessão de Transformação de Mascaramento de Dados, 155](#)
- [Regras e Diretrizes para Transformações Mascaramento de Dados, 156](#)

transformação de Mascaramento de Dados

Use a Transformação de Mascaramento de Dados para mudar dados de produção confidenciais para dados de teste realistas, para ambientes que não sejam de produção. A Transformação Mascaramento de Dados modifica dados de origem com base em regras de mascaramento que você configura para cada coluna.

Crie dados mascarados para desenvolvimento, testes e treinamento de software e mineração de dados. Você pode manter os relacionamentos de dados nos dados mascarados e manter a integridade referencial

entre tabelas de bancos de dados. A transformação de Mascaramento de Dados é uma transformação passiva.

A Transformação Mascaramento de Dados fornece regras de mascaramento com base no tipo de dados de origem e no tipo de mascaramento que você configura para uma porta. No caso de cadeias, é possível restringir os caracteres de uma cadeia a serem substituídos e os caracteres a serem aplicados na máscara. No caso de números e datas, é possível fornecer uma variedade de números para os dados mascarados. É possível configurar uma variedade fixa ou de diferença percentual do número original. O Serviço de Integração substitui caracteres com base na localidade que você configura com as regras de mascaramento.

É possível aplicar os seguintes tipos de mascaramento com a Transformação Mascaramento de Dados:

- **Mascaramento de chaves.** Gera resultados deterministas para os mesmos dados de origem, regras de mascaramento e valor de semente.
- **Mascaramento de substituição.** Substitui uma coluna de dados por dados semelhantes, mas não relacionados, de um dicionário.
- **Mascaramento dependente.** Substitui os valores de uma coluna de origem com base nos valores de outra coluna de origem.
- **Mascaramento aleatório.** Substitui os valores de uma coluna de dados com outros valores da mesma coluna.
- **Mascaramento aleatório.** Gera resultados aleatórios e que não se repetem para os mesmos dados de origem e regras de mascaramento.
- **Mascaramento de expressão.** Aplica uma expressão a uma porta para alterar os dados ou criar dados.
- **Formatos especiais de máscara.** Aplica formatos especiais de máscara para tipos comuns de dados confidenciais. Você pode mascarar números de previdência social, números de cartão de crédito, números de telefone, URLs, endereços de e-mail, ou endereços IP.

Para usar a transformação de Mascaramento de Dados, você precisa da licença apropriada.

TÓPICOS RELACIONADOS:

- [“Valores Padrão para Portas” na página 39](#)
- [“Aplicando Regras de Mascaramento” na página 143](#)
- [“Arquivo de valor padrão” na página 155](#)

Propriedades de mascaramento

Defina as portas de entrada e configure as propriedades da máscara para cada porta na guia Propriedades da Máscara. O tipo de máscara que você pode selecionar é baseado no tipo de dados da porta. Quando você escolhe um tipo de máscara, o Designer exibe as respectivas regras da máscara.

Local

O local identifica o idioma e a região dos caracteres nos dados. Escolha um local na lista. A transformação de Mascaramento de Dados mascara os dados com caracteres do local que você escolher. Os dados de origem devem conter caracteres compatíveis com o local que você selecionar.

Se a localidade não está na lista, selecione uma localidade que tem uma página de código similar ou correspondente. Não é possível selecionar Unicode para a localidade.

Tipos de mascaramento

O tipo de mascaramento é o tipo de mascaramento de dados a ser aplicado à coluna selecionada. Selecione um dos tipos de mascaramento a seguir:

- Mascaramento de chaves. Gera resultados deterministas para os mesmos dados de origem, regras de mascaramento e valor de semente.
- Mascaramento de substituição. Substitui uma coluna de dados por dados semelhantes, mas não relacionados, de um dicionário.
- Mascaramento dependente. Substitui os valores de uma coluna de origem com base nos valores de outra coluna de origem.
- Mascaramento aleatório. Gera resultados aleatórios para os mesmos dados de origem e regras de mascaramento.
- Mascaramento de expressão. Aplica uma expressão a uma porta para alterar os dados ou criar dados.
- Formatos especiais de máscara. Aplica formatos especiais de máscara para tipos comuns de dados confidenciais. Você pode mascarar números de previdência social, números de cartão de crédito, números de telefone, URLs, endereços de e-mail, ou endereços IP.
- Substituição. Substitui uma coluna de dados por dados semelhantes, mas não relacionados, de um dicionário.
- Sem mascaramento. A transformação de Mascaramento de Dados não altera os dados de origem.

O padrão é Sem Mascaramento.

Saída Repetível

Saída repetível é o conjunto de valores consistentes retornados pela transformação de Mascaramento de Dados.

A saída repetível retorna valores determinísticos. Por exemplo, você configura a saída repetível para uma coluna de nomes. A transformação de Mascaramento de Dados retorna os mesmos valores mascarados toda vez que você inclui a transformação em um fluxo de trabalho.

Você pode configurar o mascaramento repetível para todos os tipos de mascaramento de dados. Para configurar o mascaramento repetível, clique em **Saída Repetível** e selecione o valor de **Semente**.

Semente

O valor de semente é um ponto inicial para gerar valores mascarados.

A transformação de Mascaramento de Dados cria um valor de semente padrão, que é um número aleatório entre 1 e 1.000. Você pode inserir outro valor de semente ou aplicar um valor de parâmetro de mapeamento. Aplique o mesmo valor de semente a uma coluna para retornar os mesmos valores de dados mascarados em outros dados de origem. Por exemplo, se você tiver a mesma coluna Cust_ID em quatro tabelas desejadas que todas emitam os mesmos valores mascarados, defina todas as quatro colunas com o mesmo valor de semente.

Parâmetros de Mapeamento

Você pode usar um parâmetro de mapeamento para definir um valor de semente. Crie um parâmetro de mapeamento para cada valor de semente que você deseja adicionar à transformação. O valor do parâmetro de mapeamento é um número entre 1 e 1.000.

Quando você configurar o mascaramento da dados para uma coluna, selecione Parâmetro de Mapeamento para a semente. O Designer exibe uma lista dos parâmetros de mapeamento. Selecione um parâmetro de mapeamento na lista. Antes de executar uma sessão, você pode alterar o valor do parâmetro de mapeamento em um arquivo de parâmetro para a sessão.

Crie os parâmetros de mapeamento antes de criar a transformação de Mascaramento de Dados. Se você decidir parametrizar o valor da semente e o mapeamento não tiver nenhum parâmetro de mapeamento, um erro aparece. Se você selecionar uma porta com uma regra de mascaramento que faz referência a um parâmetro de mapeamento excluído, o Designer gera um novo valor de semente aleatório para a porta. O valor da semente não é um parâmetro de mapeamento. Aparecerá a mensagem de que o parâmetro de mapeamento foi excluído e o Designer cria um novo valor de semente.

O Serviço de Integração aplica um valor de semente padrão nas circunstâncias a seguir:

- A opção de parâmetro de mapeamento é selecionada para uma coluna, mas a sessão não tem nenhum arquivo de parâmetro.
- Você exclui o parâmetro de mapeamento.
- Um valor de semente do parâmetro de mapeamento não é um número entre 1 e 1.000.

O Serviço de Integração aplica valores mascarados do arquivo de valores padrão. Você pode editar o arquivo de valor padrão para alterar os valores padrão.

O arquivo de valor padrão é um arquivo XML no local a seguir:

```
<PowerCenter Installation Directory>\infa_shared\SrcFiles\defaultValue.xml
```

O par nome-valor para a semente é

```
default_seed = "500".
```

Se o valor da semente no arquivo de valor padrão não estiver entre 1 e 1.000, o Serviço de Integração atribui o valor de 725 à semente e escreve uma mensagem no log da sessão.

TÓPICOS RELACIONADOS:

- [“Arquivo de valor padrão” na página 155](#)

P/S Associada

A P/S Associada é a porta de saída associada a uma porta de entrada. A Transformação de Mascaramento de Dados cria uma porta de saída para cada porta de entrada. A convenção de nomenclatura é out_<nome da porta>. A porta de saída associada é somente leitura.

Mascaramento de Chave

Uma coluna configurada para mascaramento de chaves retorna dados mascarados determinísticos sempre que o valor de origem e o valor de semente são iguais. A Transformação Mascaramento de Dados retorna valores exclusivos para a coluna.

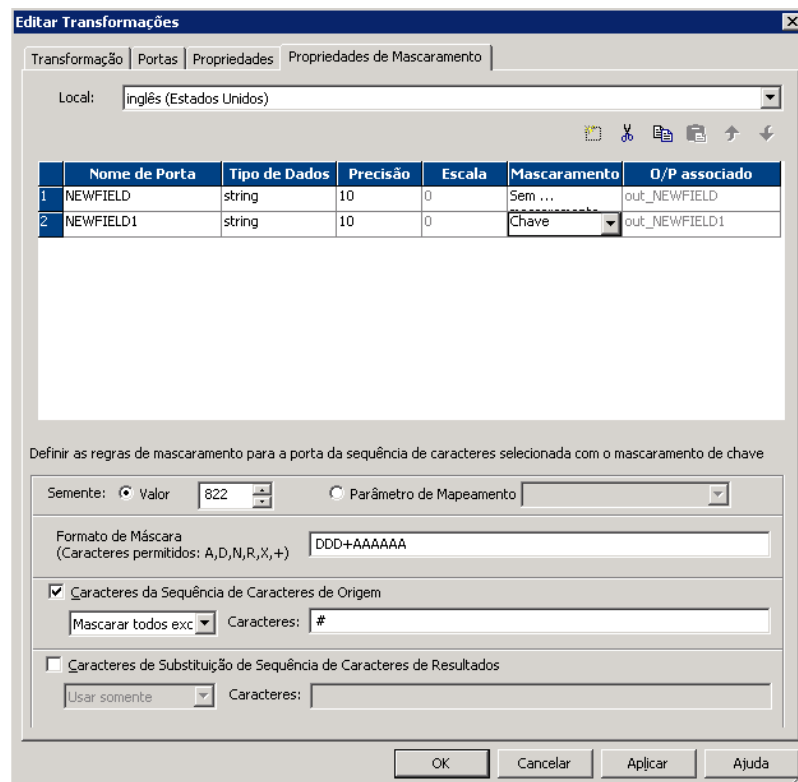
Quando você configura uma coluna para mascaramento de chaves, a transformação de Mascaramento de Dados cria um valor de semente para a coluna. É possível alterar o valor de semente para produzir dados repetíveis entre diferentes Transformações de Mascaramento de Dados. Por exemplo, configure mascaramento de chave para impor integridade referencial. Use o mesmo valor de semente para mascarar uma chave primária em uma tabela e o valor de chave externa em outra tabela.

Você pode definir regras de mascaramento que afetam o formato de dados retornado pela transformação de Mascaramento de Dados. Mascare valores de string e numéricos com mascaramento de chave.

Mascaramento de Valores de String

É possível configurar o mascaramento de chave para que as strings gerem saída repetível. Configure um formato de máscara para definir limitações para cada caractere na string de saída, para definir um formato de máscara. Configure os caracteres da string de origem e os caracteres de substituição de string de resultado que definem os caracteres origem para mascarar e os caracteres com os quais devem ser mascarados.

A figura a seguir mostra propriedades de mascaramento de chave para um tipo de dados de string:



É possível configurar as seguintes regras de mascaramento para valores de string de mascaramento de chave:

- **Semente.** Aplique um valor semente para gerar dados mascarados determinísticos para uma coluna. Selecione uma das seguintes opções:
 - **Valor.** Aceite o valor semente padrão ou insira um número entre 1 e 1.000.
 - **Parâmetro de Mapeamento.** Use um parâmetro de mapeamento para definir o valor semente. O Designer exibe uma lista dos parâmetros de mapeamento que você cria para o mapeamento. Escolha o parâmetro de mapeamento na lista para usar como o valor semente.
- **Formato de Máscara.** Para cada caractere dos dados de entrada, defina o tipo de caractere a ser substituído. É possível limitar cada caractere a um tipo de caractere alfabético, numérico ou alfanumérico.

- **Caracteres da String de Origem.** Defina os caracteres na string de origem que deseja mascarar. Por exemplo, mascare o caractere de sinal de número (#) sempre que ocorrer nos dados de entrada. A Transformação de Mascaramento de Dados mascara todos os caracteres de entrada quando os caracteres da string de origem estiverem em branco. A Transformação de Mascaramento de Dados nem sempre retorna dados exclusivos se o número de caracteres de strings de origem for inferior ao número de caracteres da string de resultado.
- **Caracteres da String de Resultado.** Substitua os caracteres na string de destino pelos caracteres que você definir nos Caracteres da String de Resultado. Por exemplo, insira os caracteres a seguir para configurar cada máscara para que contenha todos os caracteres alfabéticos maiúsculos:

ABCDEFGHIJKLMNOPQRSTUVWXYZ

Mascaramento de Valores Numéricos

Configure o mascaramento de chave para os dados de origem numéricos, para gerar a saída determinística. Quando você configurar uma coluna para o mascaramento de chaves numéricas, atribua um valor de semente aleatório para a coluna. Quando a transformação de Mascaramento de Dados mascara os dados de origem, ela aplica um algoritmo de mascaramento que exige a semente.

Você pode alterar o valor da semente para uma coluna, a fim de produzir resultados repetíveis se o mesmo valor de origem ocorrer em uma coluna diferente. Por exemplo, você deseja manter uma relação de chave externa primária entre duas tabelas. Em cada transformação de Mascaramento de Dados, insira o mesmo valor de semente para a coluna de chave primária, que para a coluna de chave externa. A transformação de Mascaramento de Dados produz resultados determinísticos para os mesmos valores numéricos. A integridade referencial é mantida entre as tabelas.

Mascarando Valores de Data e Hora

Quando você pode configurar o mascaramento de chaves para valores de data e hora, a transformação de Mascaramento de Dados requer um número aleatório como semente. Você pode alterar a semente para corresponder ao valor de semente de outra coluna, a fim de retornar valores datetime repetíveis entre as colunas.

A transformação de Mascaramento de Dados pode mascarar datas entre 1753 e 2400 com o mascaramento de chaves. Se o ano da origem for bissexto, a transformação de mascaramento de dados retorna um ano que também é bissexto. Se o mês da origem contém 31 dias, a transformação de mascaramento de dados retorna um mês que tenha 31 dias. Se o mês da origem for fevereiro, a transformação de mascaramento de dados retorna fevereiro.

A transformação de mascaramento de dados sempre gera datas válidas.

TÓPICOS RELACIONADOS:

- [“Intervalo de datas ” na página 146](#)

Mascaramento de Substituição

O mascaramento de substituição substitui uma coluna de dados por dados semelhantes, mas não relacionados. Use o mascaramento de substituição para substituir dados de produção por dados de teste

realistas. Quando você configurar o mascaramento de substituição, defina o dicionário que contém os valores substitutos.

A transformação de Mascaramento de Dados realizará uma pesquisa no dicionário que você configurar. A transformação de Mascaramento de Dados substitui dados da origem por dados do dicionário. Arquivos de dicionário podem conter dados de string, valores de data e hora, inteiros e números de ponto flutuante. Insira valores de data e hora no seguinte formato:

```
mm/dd/yyyy
```

Você pode substituir dados com valores repetíveis ou não repetíveis. Se você escolher valores repetíveis, a transformação de Mascaramento de Dados produzirá resultados deterministas para dados da mesma origem e valor de semente. Você deve configurar um valor de semente para substituir dados por resultados deterministas. O Serviço de Integração mantém uma tabela de armazenamento de valores de origem mascarados para mascaramento repetível.

Você pode substituir mais de uma coluna de dados por valores mascarados da mesma linha de dicionário. Configure o mascaramento de substituição para uma coluna de entrada. Configure o mascaramento de dados dependentes para as outras colunas que recebem dados mascarados da mesma linha do dicionário.

Dicionários

Um dicionário é um arquivo simples ou uma tabela relacional que contém os dados substitutos para cada linha no arquivo. A transformação de Mascaramento de Dados gera um número para recuperar uma linha de dicionário. A transformação de Mascaramento de Dados gera uma chave de hash para mascaramento de substituição repetível ou um número aleatório para mascaramento não repetível. Você pode configurar uma condição de pesquisa adicional.

Você pode configurar um dicionário para mascarar mais de uma porta na transformação de Mascaramento de Dados.

O exemplo a seguir mostra um dicionário de arquivo simples que contém nome e sexo:

```
SNO,GENDER,FIRSTNAME
1,M,Adam
2,M,Adeel
3,M,Adil
4,F,Alice
5,F,Alison
```

Neste dicionário, o primeiro campo da linha é o número de série e o segundo é o gênero. Você pode adicionar o sexo como uma condição de pesquisa. O Serviço de Integração recupera uma linha do dicionário usando uma chave hash, e encontra uma linha com um sexo que corresponde ao sexo nos dados da origem.

Use as seguintes regras e diretrizes ao criar um dicionário:

- A primeira linha de um dicionário de arquivo simples deve ter rótulos de coluna para identificar os campos em cada registro. Os campos estão separados por vírgulas. Se a primeira linha não contiver rótulos de coluna, o Serviço de Integração usa os valores dos campos na primeira linha como nomes de coluna.
- Um dicionário de arquivo simples deve estar no diretório do arquivo de pesquisa \$PMLookupFileDir. Por padrão, esse diretório está no seguinte local:

```
<PowerCenter_Installation_Directory>\server\infa_shared\LkpFiles
```

- Se você criar um dicionário de arquivo simples no Windows e copiá-lo para uma máquina UNIX, verifique se o formato do arquivo está correto para o UNIX. Por exemplo, o Windows e o UNIX usam caracteres diferentes para o marcador de final de linha.
- Se você configurar o mascaramento da substituição para mais de uma porta, todos os dicionários relacionais devem estar no mesmo esquema do banco de dados.

- O comprimento do buffer sequencial em linha de um dicionário de arquivo simples deve ser menor que ou igual a 600 caracteres.
- Você não pode alterar o tipo ou o nome do Dicionário de substituição nas propriedades da sessão.

Tabelas de Armazenamento

A transformação de Mascaramento de Dados mantém tabelas de armazenamento para substituição repetida entre sessões. Uma linha da tabela de armazenamento contém a coluna de origem e um par de valores mascarados. Toda vez que a transformação de Mascaramento de Dados mascara um valor com um valor substituto repetível, ela pesquisa a tabela de armazenamento por nome de dicionário, local, nome de coluna, valor de entrada e semente. Se ela encontrar uma linha, ela retorna o valor mascarado da tabela de armazenamento. Se a transformação de Mascaramento de Dados não encontrar uma linha, ela recuperará uma linha do dicionário com uma chave de hash.

O formato de nome do dicionário na tabela de armazenamento não é o mesmo para um dicionário de arquivo simples e um dicionário relacional. O nome do dicionário de um arquivo simples é identificado pelo nome do arquivo. Já o nome do dicionário relacional tem a seguinte sintaxe:

```
<Connection object>_<dictionary table name>
```

O Informatica fornece scripts que você pode executar para criar uma tabela de armazenamento relacional. Os scripts estão no seguinte local:

```
<PowerCenter Client installation directory>\client\bin\Extensions\DataMasking
```

O diretório contém um script para bancos de dados Sybase, Microsoft SQL Server, IBM DB2 e Oracle. Cada script recebe o nome Substitution_<tipo do banco de dados>. Você poderá criar uma tabela em um banco de dados diferente se configurar as instruções SQL e as restrições de chaves primárias.

É necessário criptografar as tabelas de armazenamento para mascaramento de substituição quando você tiver dados descriptografados no armazenamento e usar o mesmo valor de semente e dicionário para criptografar as mesmas colunas.

Criptografando Tabelas de Armazenamento para Mascaramento de Substituição

Você pode usar funções de codificação de linguagem de transformação para criptografar tabelas de armazenamento. É necessário criptografar tabelas de armazenamento quando você tiver ativado criptografia de armazenamento.

1. Crie um mapeamento com a tabela de armazenamento IDM_SUBSTITUTION_STORAGE como origem.
2. Crie uma transformação de Mascaramento de Dados.
3. Aplique a técnica de mascaramento de substituição nas portas de valor de entrada e de valor mascarado.
4. Use a seguinte expressão na porta INPUTVALUE:

```
Enc_Base64(AES_Encrypt(INPUTVALUE, Key))
```
5. Use a seguinte expressão na porta MASKEDVALUE:

```
Enc_Base64(AES_Encrypt(MASKEDVALUE, Key))
```
6. Vincule as portas ao destino.

Propriedades do Mascaramento de Substituição

Você pode configurar estas regras de mascaramento para o mascaramento de substituição:

- **Saída Repetível.** Retorna resultados deterministas entre sessões. A transformação de Mascaramento de Dados armazena valores mascarados na tabela de armazenamento.
- **Semente.** Aplique um valor semente para gerar dados mascarados determinísticos para uma coluna. Selecione uma das seguintes opções:
 - **Valor.** Aceite o valor semente padrão ou insira um número entre 1 e 1.000.
 - **Parâmetro de Mapeamento.** Use um parâmetro de mapeamento para definir o valor semente. O Designer exibe uma lista dos parâmetros de mapeamento que você cria para o mapeamento. Escolha o parâmetro de mapeamento na lista para usar como o valor semente.
 - **Saída Exclusiva.** Força a transformação de Mascaramento de Dados a criar valores de saída exclusivos para valores de entrada exclusivos. Não há dois valores de entrada mascarados para o mesmo valor de saída. O dicionário deve ter linhas exclusivas suficientes para ativar a saída exclusiva. Quando você desabilita a saída exclusiva, a Transformação de Mascaramento de Dados poderá não mascarar valores de entrada para valores de saída exclusivos. O dicionário pode conter menos linhas.
- **Informações do Dicionário.** Configure o arquivo simples ou a tabela relacional que contém os valores de dados substituídos.
 - **Tabela Relacional.** Selecione Tabela Relacional se o dicionário estiver em uma tabela do banco de dados. Clique em Selecionar Tabela para configurar o banco de dados e a tabela.
 - **Arquivo simples.** Selecione Arquivo Simples se o dicionário estiver em um arquivo simples delimitado por vírgulas. Clique em Selecionar Arquivo Simples para procurar e escolher um arquivo.
 - **Nome do Dicionário.** Exibe o nome do arquivo simples ou da tabela relacional selecionada.
 - **Coluna Saída.** Escolha a coluna para retornar à transformação de Mascaramento de Dados.
- **Condição de pesquisa.** Configure uma condição de pesquisa para qualificar ainda mais a linha do dicionário a ser usada no mascaramento de substituição. A condição de pesquisa é semelhante à cláusula WHERE em uma consulta SQL. Ao configurar uma condição de pesquisa, você compara o valor de colunas na origem ao de colunas no dicionário.

Por exemplo, você quer mascarar o nome. Os dados na origem e no dicionário têm uma coluna de nomes e outra de sexo. Você pode adicionar uma condição segundo a qual cada nome feminino seja substituído por outro nome feminino do dicionário. A condição de pesquisa compara o gênero na origem ao gênero no dicionário.

 - **Porta de entrada.** Coluna de dados da origem a ser usada na pesquisa.
 - **Coluna do Dicionário.** A coluna do dicionário a ser comparada à porta de entrada.

Dicionário Relacional

Ao escolher uma tabela relacional como um dicionário, configure a tabela de banco de dados que contém o dicionário. Ao definir a tabela, o Designer conecta-se à tabela e exibe as colunas.

Para selecionar um banco de dados, clique em Selecionar tabela na guia Propriedades de mascaramento da transformação de Mascaramento de Dados.

Configure os campos a seguir na caixa de diálogo Selecionar tabela:

- **Origem de dados ODBC.** Origem de dados ODBC que se conecta ao banco de dados que contém o dicionário.
- **Nome de usuário** Um nome de usuário de banco de dados para se conectar ao banco de dados. O nome de usuário deve ter as permissões de banco de dados para exibir as tabelas.

- **Nome do proprietário.** Se o nome de usuário não for o proprietário da tabela de dicionário, digite o proprietário da tabela.
- **Senha** A senha para o nome de usuário do banco de dados.
- **Pesquisa de tabelas nomeadas.** Opcional. Limita o número de tabelas que aparecem na caixa de diálogo.

Ao clicar em Conectar, o Designer se conecta à origem de dados e exibe as tabelas. Role a lista de tabelas e selecione a que você deseja usar para o dicionário. O nome da tabela aparece no campo Nome do Dicionário da guia Propriedades de mascaramento.

Requisitos de conexão

Você deve configurar uma conexão de banco de dados para a tabela de armazenamento e o dicionário relacional, se usá-los com o mascaramento de substituição.

Configure as conexões do banco de dados no Workflow Manager. Selecione um objeto de conexão relacional para a conexão IDM_Dictionary e a IDM_Storage na guia Mapeamento das propriedades da sessão.

Regras e Diretrizes para Mascaramento de Substituição

Use as seguintes regras e diretrizes para mascaramento de substituição:

- Se uma tabela de armazenamento não existir para uma máscara de substituição repetível exclusiva, a sessão falhará.
- Se o dicionário não contiver linhas, a transformação de Mascaramento de Dados retorna uma mensagem de erro.
- Quando a transformação de Mascaramento de Dados encontra um valor de entrada com o local, o dicionário e a semente na tabela de armazenamento, ela recupera o valor mascarado, mesmo que a linha não esteja mais no dicionário.
- Se você excluir um objeto de conexão ou modificar o dicionário, trunque a tabela de armazenamento. Caso contrário, talvez obtenha resultados inesperados.
- Se o número de valores no dicionário for menor que o número de valores exclusivos nos dados de origem, a transformação de Mascaramento de Dados não poderá mascarar os dados com valores repetíveis exclusivos. A transformação de Mascaramento de Dados retorna uma mensagem de erro.

Mascaramento dependente

O mascaramento dependente substitui dados de várias colunas de dados de origem por dados da mesma linha de dicionário.

Quando a transformação de Mascaramento de Dados executa o mascaramento de substituição em várias colunas, os dados mascarados podem conter combinações de campos irreais. Você pode configurar o mascaramento dependente para substituir os dados por várias colunas de entrada da mesma linha de dicionário. Os dados mascarados recebem combinações válidas, como "Nova York, Nova York" ou "Chicago, Illinois".

Antes de configurar o mascaramento dependente, você deve configurar uma coluna de entrada para o mascaramento de substituição. Configure outras colunas de entrada para serem dependentes dessa coluna de substituição. Por exemplo, escolha a coluna de CEP para mascaramento de substituição e as colunas de Cidade e Estado para serem dependentes da coluna de CEP. O mascaramento dependente garante que os valores substituídos de cidade e estado sejam válidos para o valor de CEP substituído.

Nota: Você só pode configurar uma coluna para mascaramento dependente depois de configurar uma coluna para mascaramento de substituição.

Configure as seguintes regras de mascaramento ao configurar uma coluna para mascaramento dependente:

Coluna de Dependentes

O nome da coluna de entrada que você configurou para o mascaramento de substituição. A transformação de Mascaramento de Dados recupera dados substitutos de um dicionário usando as regras de mascaramento para aquela coluna. A coluna configurada para mascaramento de substituição se torna a coluna principal para recuperação de dados mascarados do dicionário.

Coluna de Saída

O nome da coluna do dicionário que contém o valor da coluna que está sendo configurada com mascaramento dependente.

Exemplo de mascaramento dependente

Um dicionário de mascaramento de dados pode conter linhas de endereço com os seguintes valores:

```
SNO, STREET, CITY, STATE, ZIP, COUNTRY  
1, 32 Apple Lane, Chicago, IL, 61523, US  
2, 776 Ash Street, Dallas, TX, 75240, US  
3, 2229 Big Square, Atleeville, TN, 38057, US  
4, 6698 Cowboy Street, Houston, TX, 77001, US
```

É necessário mascarar os dados de origem com combinações válidas de cidade, estado e CEP do dicionário Endereço.

Configure a porta ZIP para mascaramento de substituição.

Insira as seguintes regras de mascaramento para a porta de CEP:

Regra	Valor
Tipo de Arquivo de Dicionário	Arquivo simples ou tabela relacional
Nome do Dicionário	Address.dic
Coluna de Saída	CEP

Configure a porta Cidade para mascaramento dependente.

Insira as seguintes regras de mascaramento para a porta Cidade:

Regra	Valor
Coluna de Dependentes	CEP
Coluna de Saída	Cidade

Configure a porta Estado para mascaramento dependente.

Insira as seguintes regras de mascaramento para a porta Estado:

Regra	Valor
Coluna de Dependentes	CEP
Coluna de Saída	Estado

Quando a transformação de Mascaramento de Dados mascara o CEP, ela retorna da linha de dicionário a cidade e o estado corretos para o CEP em questão.

Mascaramento Dependente Repetível

Quando configura vários campos de entrada para receber dados mascarados da mesma linha de dicionário, você precisa determinar qual campo configurar para mascaramento de substituição. Com o mascaramento repetível, selecione uma coluna que identifique de forma exclusiva os dados de origem. A transformação de Mascaramento de Dados usa o campo de chave para determinar se uma linha já existe no armazenamento para mascaramento repetível.

Nota: Quando os dados de origem têm uma chave primária, você pode configurar a coluna de chave primária para mascaramento de substituição. Configure outra coluna para mascaramento dependente.

Por exemplo, se mascarar dados de funcionários, você poderá configurar EmployeeID para mascaramento de substituição e FirstName e LastName para mascaramento dependente.

Quando você configura o mascaramento repetível, as seguintes linhas recebem dados mascarados diferentes:

EmployeeID	FirstName	LastName
111	John	Jones
222	Radhika	Jones

Se você escolher LastName como campo de chave para mascaramento dependente repetível, todas as linhas com o sobrenome Jones receberão os mesmos dados de mascaramento.

Mascaramento Aleatório

O mascaramento aleatório gera dados mascarados aleatórios não determinísticos. A transformação de Mascaramento de Dados retorna diferentes valores quando o mesmo valor de origem ocorre em diferentes linhas. Você pode definir regras de mascaramento que afetam o formato de dados retornado pela transformação de Mascaramento de Dados. Mascare valores numéricos, de string e de data com um mascaramento aleatório.

Mascaramento de Valores Numéricos

Ao mascarar dados numéricos, é possível configurar uma série de valores de saída de uma coluna. A Transformação de Mascaramento de Dados retorna um valor entre os valores mínimo e máximo do intervalo, dependendo da precisão da porta. Para definir a série, configure os intervalos mínimo e máximo ou um intervalo de embaçamento com base em uma variância a partir do valor de origem primário.

Você pode configurar os seguintes parâmetros de mascaramento para dados numéricos:

Faixa

Defina um intervalo de valores de saída. A Transformação de Mascaramento de Dados retorna dados numéricos entre os valores mínimo e máximo.

Faixa de Embaçamento

Defina um intervalo de valores de saída que estejam dentro de uma variância fixa ou uma variância de porcentagem dos dados de origem. A Transformação de Mascaramento de Dados retorna dados numéricos que estão próximos ao valor dos dados de origem. É possível configurar um intervalo e um intervalo de embaçamento.

Mascaramento de Valores de String

Configure mascaramento aleatório para gerar saídas aleatórias para colunas de string. Para configurar os limites de cada caractere na string de saída, configure um formato de máscara. Configure caracteres de filtro para definir quais caracteres de origem devem ser mascarados e os caracteres com os quais se mascara esses caracteres de origem.

É possível aplicar as seguintes regras de mascaramento para uma porta de string:

Faixa

Configure a extensão mínima e máxima da string. A Transformação de Mascaramento de Dados retorna uma string de caracteres aleatórios entre as extensões mínima e máxima de string.

Formato de Máscara

Para cada caractere dos dados de entrada, defina o tipo de caractere a ser substituído. É possível limitar cada caractere a um tipo de caractere alfabético, numérico ou alfanumérico.

Caracteres da String de Origem

Defina os caracteres na string de origem que você deseja mascarar. Por exemplo, mascare o caractere de sinal de número (#) sempre que ocorrer nos dados de entrada. A Transformação de Mascaramento de Dados mascara todos os caracteres de entrada quando os caracteres da string de origem estiverem em branco.

Caracteres de Substituição da String de Resultados

Substitua os caracteres na string de destino pelos caracteres que você definir em Caracteres da String de Resultado. Por exemplo, insira os caracteres a seguir para configurar cada máscara para que contenha caracteres alfabéticos maiúsculos de A a Z:

ABCDEFGHIJKLMNOPQRSTUVWXYZ

Mascaramento de Valores de Data

Para mascarar valores de data com o mascaramento aleatório, configure um intervalo de datas de saída ou escolha uma variância. Quando configurar uma variância, escolha uma parte da data para desfocar. Escolha o ano, mês, dia, hora, minuto ou segundo. A transformação de Mascaramento de Dados retorna uma data que esteja dentro do intervalo que você configurou.

Você pode configurar as seguintes regras de mascaramento quando mascarar um valor datetime:

Faixa

Configura os valores mínimo e máximo para retornar, em referência ao valor datetime selecionado.

Embaçamento

Mascara uma data com base em uma variância que você aplica a uma unidade da data. A Transformação de Mascaramento de Dados retorna uma data dentro de uma variância. Você pode embaçar o ano, mês, dia, hora, minuto ou segundo. Escolha uma variância baixa e alta para aplicar.

Aplicando Regras de Mascaramento

Aplique regras de mascaramento com base no tipo de dados de origem. Quando você clica em uma propriedade de coluna na guia Propriedades de Mascaramento, o Designer exibe as regras de mascaramento com base no tipo de dados da porta.

A tabela a seguir descreve as regras de mascaramento que você pode configurar, com base no tipo de mascaramento e no tipo de dados de origem:

Tipos de mascaramento	Tipo de dados de origem:	Regras de Mascaramento	Descrição
Aleatório e Chave	String	Formato de Máscara	Máscara que limita cada caractere em uma string de saída para um caractere alfabético, numérico ou alfanumérico.
Aleatório e Chave	String	Caracteres da String de Origem	Conjunto de caracteres de origem para mascarar ou excluir do mascaramento.
Aleatório e Chave	String	Caracteres de Substituição da String de Resultados	Um conjunto de caracteres para incluir ou excluir em uma máscara.
Aleatório	Numérico String Data/Hora	Intervalo	Uma variedade de valores de saída. <ul style="list-style-type: none">- Numérico A Transformação de Mascaramento de Dados retorna dados numéricos entre os valores mínimo e máximo.- String. Retorna uma string de caracteres aleatórios entre os tamanhos de string mínimo e máximo.- Data e hora Retorna uma data e hora dentro do datetime mínimo e máximo.
Aleatório	Numérico Data/Hora	Embaçamento	Intervalo de valores de saída com uma variância fixa ou percentual dos dados de origem. A Transformação de Mascaramento de Dados retorna dados que estão próximos ao valor dos dados de origem. As colunas Datetime exigem uma variância fixa.

Formato de Máscara

Configure um formato de máscara para limitar cada caractere da coluna de saída a um caractere alfabético, numérico ou alfanumérico. Use os caracteres a seguir para definir um formato de máscara:

A, D, N, X, +, R

Nota: O formato de máscara contém caracteres em maiúsculas. Quando você insere um caractere de máscara em minúscula, a transformação de Mascaramento de Dados converte o caractere em letra maiúscula.

A tabela a seguir descreve os caracteres do formato de máscara:

Caractere	Descrição
A	Caracteres alfabéticos. Por exemplo, caracteres em ASCII de a a z e de A a Z.
Dia	Dígitos de 0 a 9. A transformação de Mascaramento de Dados retorna um "X" para caracteres diferentes dos dígitos de 0 a 9.
N	Caracteres alfanuméricos. Por exemplo, caracteres em ASCII de a a z, de A a Z e de 0 a 9.
X	Qualquer caractere. Por exemplo, alfanumérico ou símbolo.
+	Sem mascaramento.
R	Caracteres restantes. R especifica que os caracteres restantes na string podem ser qualquer tipo de caractere. R deve aparecer como o último caractere da máscara.

Por exemplo, um nome de departamento tem o formato a seguir:

```
nnn-<department_name>
```

É possível configurar uma máscara para forçar os três primeiros caracteres para serem numéricos, o nome do departamento para ser alfanumérico e o traço para permanecer na saída. Configure o seguinte formato de máscara:

```
DDD+AAAAAAAAAAAAAAAA
```

A Transformação Mascaramento de Dados substitui os três primeiros caracteres por caracteres numéricos. O quarto caractere não é substituído. A transformação de Mascaramento de Dados substitui os caracteres restantes por caracteres alfabéticos.

Se você não definir um formato de máscara, a Transformação Mascaramento de Dados substituirá cada caractere de origem por qualquer caractere. Se o formato de máscara for mais extenso do que a string de entrada, a Transformação Mascaramento de Dados ignora os caracteres a mais no formato de máscara. Se o formato de máscara for menor que a string de origem, a transformação de Mascaramento de Dados mascarará os caracteres restantes com formato R.

Nota: Não é possível configurar um formato de máscara com a opção de intervalo.

Caracteres da String de Origem

Caracteres da string de origem são caracteres que podem ser ou não mascarados. A posição dos caracteres na string de origem não importa. Os caracteres de origem diferenciam maiúsculas e minúsculas.

Você pode configurar qualquer número de caracteres. Se Caracteres estiver em branco, a transformação de Mascaramento de Dados substituirá todos os caracteres de origem na coluna.

Selecione uma das seguintes opções para caracteres da string de origem:

Apenas máscara

A transformação de Mascaramento de Dados mascara caracteres que você tenha configurado como caracteres da string de origem. Por exemplo, se você inserir os caracteres A, B e c, a transformação de Mascaramento de Dados substituirá A, B ou c por outro caractere quando ocorrer nos dados de origem.

Outros caracteres de origem diferentes de A, B ou c não serão alterados. A máscara diferencia maiúsculas e minúsculas.

Mascarar tudo exceto

Mascara todos os caracteres, exceto os caracteres da sequência de origem que ocorram na string de origem. Por exemplo, se você inserir o caractere de origem do filtro "-" e selecionar Mascarar Todos Exceto, a transformação de Mascaramento de Dados não substituirá o caractere "-" quando ele ocorrer na sequência de origem. Os outros caracteres de origem serão alterados.

Exemplo de String de Origem

Um arquivo de origem apresenta uma coluna denominada Dependentes. A coluna Dependentes contém mais de um nome separados por vírgulas. É necessário mascarar a coluna Dependentes e manter a vírgula nos dados de teste para delimitar os nomes.

No caso da coluna de Dependências, selecione Caracteres da String de Origem. Escolha Não Mascarar e digite "," para o caractere de origem a ignorar. Não digite aspas.

A Transformação Mascaramento de Dados substitui todos os caracteres na string de origem, exceto a vírgula.

Caracteres de Substituição da String de Resultados

Caracteres de substituição da string de resultados são escolhidos por você como caracteres substitutos nos dados mascarados. Quando você configura caracteres de substituição da string de resultados, a transformação de Mascaramento de Dados substitui caracteres na string de origem por caracteres de substituição da string de resultados. Para não gerar a mesma saída para valores de entrada diferentes, configure uma variedade grande de caracteres substitutos ou mascare poucos caracteres da origem. A posição de cada caractere na string não é importante.

Selecione uma das seguintes opções para caracteres de substituição da string de resultados:

Usar Apenas

Mascare a origem só com os caracteres que você definir como caracteres de substituição da string de resultados. Por exemplo, se você inserir os caracteres A, B e c, a transformação de Mascaramento de Dados substituirá cada caractere na coluna de origem por A, B ou c. A palavra "casa" poderá ser substituída por "BAcA".

Usar tudo exceto

Mascare a origem com quaisquer caracteres, exceto aqueles que você definir como caracteres de substituição da string de resultados. Por exemplo, se você inserir os caracteres de substituição da string de resultados A, B e c, os dados mascarados nunca conterão os caracteres A, B e c.

Exemplo de caracteres de substituição da string de resultados

Para substituir todas as vírgulas da coluna Dependentes por pontos e vírgulas, realize as seguintes tarefas:

1. Configure as vírgulas como caracteres de string de origem e selecione Apenas Máscara.
A Transformação Mascaramento de Dados mascara apenas as vírgulas quando essas ocorrem na coluna Dependentes.
2. Configure os pontos-e-vírgulas como caracteres de substituição de string de resultado e selecione Apenas Usar.
A Transformação Mascaramento de Dados substitui cada vírgula da coluna Dependentes por um ponto-e-vírgula.

Intervalo

Defina um intervalo para dados numéricos, de data ou de string. Quando você define um intervalo para valores numéricos ou de data, a transformação de Mascaramento de Dados mascara os dados de origem com um valor entre os valores mínimo e máximo. Quando configura um intervalo para uma string, você configura um intervalo de tamanhos de string.

Intervalo de Strings

Quando você configura um mascaramento de string aleatório, a transformação de Mascaramento de Dados gera strings com comprimento variável em relação ao comprimento da string de origem. Se preferir, você pode configurar uma largura de string máxima e uma mínima. Os valores que você inserir como largura máxima ou mínima deverão ser números inteiros positivos. Cada largura deverá ser menor ou igual à precisão da porta.

Intervalo numérico

Defina os valores mínimo e máximo para uma coluna numérica. O valor máximo deve ser menor que ou igual à precisão da porta. O intervalo padrão vai desde um até o comprimento da precisão da porta.

Intervalo de datas

Configure os valores mínimo e máximo para o valor datetime. Os campos mínimos e máximos contêm as datas mínima e máxima padrão. O formato padrão de datetime é MM/DD/AAAA HH24:MI:SS. O datetime máximo deve ser posterior ao mínimo.

Embaçamento

O Embaçamento cria um valor de saída dentro de uma variância fixa ou percentual a partir do valor dos dados de origem. Configure o embaçamento para retornar um valor aleatório próximo do valor original. Você pode embaçar valores numéricos e de data.

Embaçando Valores Numéricos

Selecione uma variância fixa ou percentual para embaçar um valor de origem numérico. O valor baixo de embaçamento é uma variância que está abaixo do valor de origem. O valor alto de embaçamento é uma variância que está acima do valor de origem. Os valores baixo e alto devem ser maiores ou iguais a zero. Quando a Transformação Mascaramento de Dados retorna dados mascarados, os dados numéricos ficam no intervalo que você define.

A tabela a seguir descreve os resultados de mascaramento dos valores de intervalo de embaçamento quando o valor de origem de entrada é 66:

Tipo de Embaçamento	Inferior	Alto	Resultado
Determinado	0	10	Entre {66} e {76}
Determinado	10	0	Entre {56} e {66}
Determinado	10	10	Entre 56 e 76
Por cento	0	50	Entre 66 e 99

Tipo de Embaçamento	Inferior	Alto	Resultado
Por cento	50	0	Entre 33 e 66
Por cento	50	50	Entre 33 e 99

Embaçando Valores de Data

Mascare uma data como variância da data de origem, configurando o embaçamento. Selecione uma unidade da data na qual aplicar a variância. É possível selecionar o ano, o mês, o dia ou a hora. Insira os limites mínimo e máximo para definir uma variância que esteja acima e abaixo da unidade da data de origem. A Transformação de Mascaramento de Dados aplica a variância e retorna uma data que esteja dentro da variância.

Por exemplo, para restringir a data mascarada para uma data que esteja a dois anos da data de origem, selecione o ano como unidade. Insira 2 como o limite mínimo e máximo. Se uma data de origem for 02/02/2006, a Transformação de Mascaramento de Dados retornará uma data entre 02/02/2004 e 02/02/2008.

Por padrão, a unidade de embaçamento é o ano.

Mascaramento de Expressão

O mascaramento de expressão aplica uma expressão a uma porta para alterar os dados ou criar novos dados. Ao configurar o mascaramento de expressão, crie uma expressão no Editor de Expressão. Selecione portas de entrada e de saída, funções, variáveis e operadores para criar expressões.

É possível concatenar dados de várias portas para criar um valor para outra porta. Por exemplo, é necessário criar um nome de logon. A origem tem colunas de nome e de sobrenome. Mascare o nome e o sobrenome dos arquivos de pesquisa. Na Transformação de Mascaramento de Dados, crie outra porta chamada Logon. Para a porta de Logon, configure uma expressão para concatenar a primeira letra do nome com a primeira letra do sobrenome:

```
SUBSTR (FIRSTNM, 1, 1) || LASTNM
```

Ao configurar um mascaramento de expressão para uma porta, o nome da porta aparece como a expressão por padrão. Para acessar o Editor de Expressão, clique no botão abrir. O Editor de Expressão exibe portas de entrada e as portas de saída que não estão configuradas para mascaramento de expressão. Você não pode usar a saída de uma expressão como entrada de outra. Se você adicionar o nome da porta de saída manualmente à expressão, pode obter resultados inesperados.

Selecione funções, portas, variáveis e operadores na interface de apontar e clicar para minimizar erros ao construir expressões.

Ao criar uma expressão, verifique se ela retorna um valor que combine com o tipo de dados da porta. A Transformação de Mascaramento de Dados retorna zero se o tipo de dados da porta de expressão for numérico e o tipo de dados da expressão não for o mesmo. A Transformação de Mascaramento de Dados retorna um valor nulo se o tipo de dados da porta de expressão for uma string e o tipo de dados da expressão não for o mesmo.

Mascaramento Repetível de Expressão

Configure o mascaramento repetível de expressão quando uma coluna de origem ocorrer em mais de uma tabela e você precisar mascarar-la com o mesmo valor.

Quando você configura o mascaramento de expressão repetível, a transformação de Mascaramento de Dados salva os resultados de uma expressão em uma tabela de armazenamento. Se a coluna ocorrer em outra tabela de origem, a transformação de Mascaramento de Dados retornará o valor mascarado da tabela de armazenamento e não da expressão.

Nome do Dicionário

Ao configurar um mascaramento de expressão repetível, é necessário inserir um nome de dicionário. O nome do dicionário é uma chave que permite que várias Transformações de Mascaramento de Dados gerem os mesmos valores mascarados a partir dos mesmos valores de origem. Defina o mesmo nome de dicionário em cada Transformação Mascaramento de Dados. O nome do dicionário pode ser qualquer texto.

Tabela de Armazenamento

A tabela de armazenamento contém os resultados do mascaramento de expressão repetível entre sessões. Uma linha da tabela de armazenamento contém a coluna de origem e um par de valores mascarados. A tabela de armazenamento para mascaramento de expressão é separada da tabela de armazenamento para mascaramento de substituição.

Toda vez que a transformação de Mascaramento de Dados mascara um valor com uma expressão repetível, ela pesquisa a tabela de armazenamento pelo nome de dicionário, local, nome de coluna e valor de entrada. Se ela encontrar uma linha na tabela de armazenamento, ela retorna o valor mascarado da tabela de armazenamento. Se a transformação de Mascaramento de Dados não encontrar uma linha, ela gerará um valor mascarado da expressão para a coluna.

É necessário criptografar as tabelas de armazenamento para mascaramento de expressão quando você tiver dados descriptografados no armazenamento e usar o mesmo nome de dicionário como chave.

Criptografando Tabelas de Armazenamento para Mascaramento de Expressão

Você pode usar funções de codificação de linguagem de transformação para criptografar tabelas de armazenamento. É necessário criptografar tabelas de armazenamento quando você tiver ativado criptografia de armazenamento.

1. Crie um mapeamento com a tabela de armazenamento IDM_EXPRESSION_STORAGE como origem.
2. Crie uma transformação de Mascaramento de Dados.
3. Aplique a técnica de mascaramento de expressão nas portas de valor mascarado.
4. Use a seguinte expressão na porta MASKEDVALUE:

```
Enc_Base64(AES_Encrypt(MASKEDVALUE, Key))
```
5. Vincule as portas ao destino.

Exemplo

Por exemplo, a tabela Funcionários contém as colunas a seguir:

```
FirstName  
LastName  
LoginID
```

Na transformação de Mascaramento de Dados, mascare LoginID com uma expressão que combine FirstName e LastName. Configure a máscara da expressão para ser repetível. Insira um nome de dicionário como uma chave de mascaramento repetível.

A tabela Computer_Users contém uma coluna LoginID, mas não a FirstName ou LastName:

```
Dept
LoginID
Password
```

Para mascarar LoginID em Computer_Users com o mesmo LoginID que Funcionários, configure o mascaramento de expressão para a coluna LoginID. Ative o mascaramento repetível e insira o mesmo nome de dicionário que você definiu para a tabela Funcionários em LoginID. O Serviço de Integração recupera os valores LoginID da tabela de armazenamento.

Crie uma expressão padrão para usar quando o Serviço de Integração não puder encontrar uma linha na tabela de armazenamento para LoginID. A tabela Computer_Users não tem as colunas FirstName ou LastName, portanto a expressão cria um LoginID menos significativo.

Script de Tabela de Armazenamento

A Informatica fornece scripts que podem ser executados para criar a tabela de armazenamento. Os scripts estão no seguinte local:

```
<PowerCenter installation directory>\client\bin\Extensions\DataMasking
```

O diretório contém um script para bancos de dados Sybase, Microsoft SQL Server, IBM DB2 e Oracle. Cada script recebe o nome <Expression_<database type>.

Regras e Diretrizes para Mascaramento de Expressão

Use as seguintes regras e diretrizes para mascaramento de expressão:

- Você não pode usar a saída de uma expressão como entrada de outra. Se você adicionar o nome da porta de saída manualmente à expressão, pode obter resultados inesperados.
- Utilize o método de apontar e clicar para construir expressões. Selecione funções, portas, variáveis e operadores na interface de apontar e clicar para minimizar erros ao construir expressões.
- Se a transformação de Mascaramento de Dados estiver configurada para mascaramento repetido, e a tabela de armazenamento não existir, o Serviço de Integração substituirá os dados de origem por valores padrão.

TÓPICOS RELACIONADOS:

- [“Usando o Editor de Expressão” na página 33](#)

Formatos Especiais de Máscara

Os tipos de máscara a seguir retêm o formato original dos dados:

- Cadastros de Pessoas Físicas
- Números de cartão de crédito
- Números de telefone
- URLs

- Endereços de e-mail
- Endereços IP
- CPF

A transformação de Mascaramento de Dados substitui dados confidenciais por valores mascarados em um formato realista. Por exemplo, quando você mascara um CPF, a transformação de Mascaramento de Dados retorna um CPF com formato correto, mas que não é válido.

Quando o tipo ou o formato dos dados de origem for inválido para uma máscara, o Serviço de Integração aplicará uma máscara padrão para os dados. O Serviço de Integração aplica valores mascarados do arquivo de valores padrão. Você pode editar o arquivo de valor padrão para alterar os valores padrão.

TÓPICOS RELACIONADOS:

- [“Arquivo de valor padrão” na página 155](#)

Mascaramento de CPF

A transformação de Mascaramento de Dados gera um número de previdência social que não é válido com base na lista da Previdência Social. A High Group List contém números válidos emitidos pela Administração do Seguro Social. A transformação de Mascaramento de Dados acessa a High Group List mais recente neste local:

```
<Installation Directory>\infa_shared\SrcFiles\highgroup.txt
```

A transformação de Mascaramento de Dados gera números de CPF que não estão na High Group List. A Administração do Seguro Social atualiza a High Group List todo mês. Baixe a versão mais recente da lista no seguinte local:

```
http://www.socialsecurity.gov/employer/ssns/highgroup.txt
```

Formato de Número da Previdência Social

A transformação de Mascaramento de Dados aceita qualquer formato SSN que contenha nove dígitos. Os dígitos podem ser delimitados por qualquer conjunto de caracteres. Por exemplo, a transformação de Mascaramento de Dados aceita o formato a seguir: `+54-*9944$#789-,*()"`.

Requisito de Código de Área

A transformação de Mascaramento de Dados retorna um número de previdência social que não é válido com o mesmo formato da origem. Os primeiros três dígitos do SSN definem o código de área. A Transformação Mascaramento de Dados não mascara o código de área. Ela mascara o número do grupo e o número de série. O SSN de origem deve conter um código de área válido. A transformação mascaramento de dados localiza o código de área na Lista de Grupos Altos e determina um intervalo de novos números que ela pode aplicar como dados mascarados. Se o SSN não for válido, a transformação mascaramento de dados não mascara os dados de origem.

Mascaramento de CPF Repetível

Você pode configurar mascaramentos repetíveis para números de CPF. Para configurar o mascaramento repetível de números de CPF, clique em Saída Repetível e escolha Valor de Semente ou Parâmetro de Mapeamento.

Quando você escolher Valor de Semente, o Designer atribuirá um número aleatório como semente. Para produzir o mesmo número de CPF em diferentes dados de origem, altere o valor de semente em cada transformação de Mascaramento de Dados para coincidir com o valor de semente do número de CPF em outras transformações. Se tiver definido a transformação de Mascaramento de Dados em um mapeamento, você poderá configurar um parâmetro de mapeamento para o valor de semente.

A transformação de Mascaramento de Dados retorna números de CPF deterministas com mascaramento repetível. A transformação de Mascaramento de Dados não pode retornar todos os números de CPF exclusivos porque não pode retornar números de CPF válidos emitidos pela Administração do Seguro Social.

Mascaramento de Número de Cartão de Crédito

O mascaramento de cartão de crédito aplica um formato de máscara interno para disfarçar números de cartão de crédito. Você pode inserir vários formatos de número de cartão de crédito. Opcionalmente, você pode substituir o emissor do cartão de crédito.

O Serviço de Integração do PowerCenter gera um número de cartão de crédito logicamente válido, quando mascara um número de cartão de crédito válido. O comprimento do número do cartão de crédito de origem deve ser 13 a 19 dígitos. O número do cartão de crédito de entrada deve ter uma soma de verificação válida e baseada nas regras do setor de cartão de crédito.

O número do cartão de crédito de origem pode conter números, espaços e hífen. Se o cartão de crédito tiver caracteres incorretos ou o comprimento errado, o Serviço de Integração escreve um erro no log da sessão. O Serviço de Integração do PowerCenter aplica uma máscara padrão de número do cartão de crédito quando os dados de origem são inválidos.

Você pode configurar o mascaramento de cartão de crédito para manter ou substituir os primeiras seis dígitos de um número de cartão de crédito. Juntos, essas dígitos identificam o emissor do cartão de crédito. Se substituir o emissor do cartão de crédito, você poderá especificar outro emissor de cartão de crédito. Você pode especificar os seguintes emissores de cartão de crédito:

- American Express
- Descobrir
- JCB
- MasterCard
- Visa
- Qualquer

Se você selecionar Qualquer, a transformação de Mascaramento de Dados retornará uma combinação de todos os emissores de cartão de crédito. Se você mascarar o emissor de cartão de crédito, a transformação de Mascaramento de Dados poderá retornar valores não exclusivos.

Por exemplo, a tabela CUSTOMER tem os seguintes números de cartão de crédito:

```
2131 0000 0000 0008
5500 0000 0000 0004
6334 0000 0000 0004
```

Se você selecionar um único emissor de cartão de crédito, os números de cartão de crédito compartilharão todos os dígitos, exceto o último. Para gerar números de cartão de crédito válidos, a transformação de Mascaramento de Dados define o último dígito de cada cartão de crédito para o mesmo valor.

Mascaramento de Número de Telefone

A transformação de Mascaramento de Dados mascara um número de telefone sem alterar o formato do número de telefone original. Por exemplo, a transformação de Mascaramento de Dados pode mascarar o número de telefone (607)382 0658 como (408)256 3106.

Os dados de origem podem conter números, espaços, hífen e parênteses. O Serviço de Integração não mascara caracteres alfabéticos nem caracteres especiais.

A transformação de Mascaramento de Dados pode mascarar dados do tipo string, integer e bigint.

Mascaramento de Endereço de E-mail

Use a transformação de Mascaramento de Dados para mascarar o endereço de e-mail que contém o valor da string. A transformação de Mascaramento de Dados pode mascarar um endereço de e-mail com caracteres ASCII aleatórios ou substituir o endereço de e-mail com um endereço realista de e-mail.

Você pode aplicar os seguintes tipos de mascaramento com o endereço de e-mail:

Mascaramento de e-mail padrão

A transformação de Mascaramento de Dados retorna caracteres ASCII aleatórios quando mascara um endereço de e-mail. Por exemplo, a transformação de Mascaramento de Dados pode mascarar Georgesmith@yahoo.com como KtrlupQAPyk@vdSKh.BIC. É o padrão.

Mascaramento de e-mail avançado

A transformação de Mascaramento de Dados mascara o endereço de e-mail com outro endereço realista de e-mail derivado das portas de saída da transformação ou das colunas de dicionário.

Mascaramento de E-mail Avançado

Com o mascaramento de e-mail avançado, você pode mascarar o endereço de e-mail com outro endereço realista de e-mail. A transformação de Mascaramento de Dados cria o endereço de e-mail das colunas de dicionário ou das portas de saída de transformação.

Você pode criar a parte local no endereço de e-mail do mapeamento de portas de saída. Ou você pode criar a parte local no endereço de e-mail da tabela relacional ou das colunas de arquivos simples.

A transformação de Mascaramento de Dados pode criar o nome de domínio para o endereço de e-mail de um valor constante ou de um valor aleatório no dicionário de domínio.

Você pode criar um mascaramento de e-mail avançado com base nas seguintes opções:

Endereço de E-mail com Base em um Mapeamento

Você pode criar um endereço de e-mail com base nas portas de saída da transformação de Mascaramento de Dados. Selecione as portas de saída de transformação para as colunas de nome e

sobrenome. A transformação de Mascaramento de Dados mascara o nome, o sobrenome ou ambos, com base nos valores especificados para o tamanho do nome e do sobrenome.

Endereço de E-mail com Base em um Dicionário

Você pode criar um endereço de e-mail com base nas colunas de um dicionário. O dicionário pode ser uma tabela relacional ou um arquivo simples.

Selecione as colunas de dicionário para o nome e o sobrenome. A transformação de Mascaramento de Dados mascara o nome, o sobrenome ou ambos, com base nos valores especificados para o tamanho do nome e do sobrenome. Você também pode configurar uma expressão para mascarar o endereço de e-mail. Se você configurar uma expressão, a transformação de Mascaramento de Dados não mascarará com base no tamanho da coluna do nome ou do sobrenome.

Parâmetros de Configuração para um Tipo de Mascaramento de Endereço de E-mail Avançado

Especifique os parâmetros de configuração quando você configurar o mascaramento de endereço de e-mail avançado.

Você pode especificar os seguintes parâmetros de configuração:

Delimitador

Você pode selecionar um delimitador, como um ponto, hífen ou sublinhado, para separar o nome e o sobrenome no endereço de e-mail. Se você não deseja separar o nome e o sobrenome no endereço de e-mail, deixe o delimitador em branco.

Coluna de FirstName

Selecione uma porta de saída da transformação de Mascaramento de Dados ou uma coluna de dicionário para mascarar o nome no endereço de e-mail.

Coluna de LastName

Selecione uma porta de saída da transformação de Mascaramento de Dados ou uma coluna de dicionário para mascarar o sobrenome no endereço de e-mail.

Tamanho das colunas de FirstName ou LastName

Restringe o comprimento de caracteres para mascarar as colunas de nome e de sobrenome. Por exemplo, os dados de entrada são Timothy para nome e Smith para sobrenome. Selecione 5 como o tamanho da coluna de nome. Selecione 1 como o tamanho da coluna de sobrenome e um ponto como o delimitador. A transformação de Mascaramento de Dados gera o seguinte endereço de e-mail:

```
timot.s@<domain_name>
```

DomainName

Você pode usar um valor constante, como gmail.com, para o nome de domínio. Ou você pode especificar um outro arquivo de dicionário que contém uma lista de nomes de domínio. O dicionário de domínio pode ser um arquivo simples ou uma tabela relacional.

Expressões para o Tipo de Mascaramento de Endereço de E-mail Avançado

Você pode usar funções de expressão quando você seleciona colunas de dicionário para criar um endereço de e-mail.

Para configurar uma expressão para o tipo de mascaramento de e-mail avançado, selecione colunas para compor o endereço de e-mail do dicionário. Você pode configurar a expressão com as colunas de dicionário, as portas de transformação, ou com ambos.

As portas de dicionário estão listadas na sintaxe a seguir no Editor de Expressão:

```
<Porta de mapeamento de string de entrada>_Dictionary_<nome de coluna de dicionário>
```

Mascaramento de Número de Previdência Social

A transformação de Mascaramento de Dados mascara um CPF de nove dígitos. Os dígitos podem ser delimitados por qualquer conjunto de caracteres.

Se o número não contiver delimitadores, o número mascarado também não conterá delimitadores. Caso contrário, o número mascarado terá o seguinte formato:

```
xxx-xxx-xxx
```

Dígito de Início SIN

Você pode definir o primeiro dígito do SIN mascarado.

Ative **Dígito de Início** e insira o dígito. A transformação de Mascaramento de Dados cria números SIN mascarados que começam com o número que você inseriu.

Números SIN Repetíveis

Você pode configurar a transformação de Mascaramento de Dados para retornar valores SIN repetíveis. Quando você configura uma porta para mascaramento SIN repetível, a transformação de Mascaramento de Dados retorna dados mascarados determinísticos sempre que o valor SIN de origem e o valor de semente são iguais.

Para retornar números SIN repetíveis, ative **Valores Repetíveis** e insira um número de semente. A transformação de Mascaramento de Dados retorna valores exclusivos para cada SIN.

Mascaramento de Endereço IP

A transformação de Mascaramento de Dados mascara um endereço IP, transformando-o em outro endereço IP, dividindo-o em quatro números separados por um ponto final. O primeiro número é a rede. A Transformação Mascaramento de Dados mascara o número de rede dentro dos limites da rede.

A Transformação Mascaramento de Dados mascara um endereço IP de Classe A como um endereço IP de Classe A e um endereço 10.x.x.x como um endereço 10.x.x.x. A Transformação Mascaramento de Dados não mascara o endereço de rede de classe e privado. Por exemplo, a Transformação Mascaramento de Dados pode mascarar 11.12.23.34 como 75.32.42.52. e 10.23.24.32 como 10.61.74.84.

Nota: Quando você mascara vários endereços IP, a transformação de Mascaramento de Dados pode retornar valores não únicos, pois não mascara a classe ou a rede privada dos endereços IP.

Mascaramento de Endereço de URL

A transformação de Mascaramento de Dados procura a string '://' e analisa a subsequência à direita para analisar uma URL. A URL de origem deve conter a string '://'. A URL de origem pode conter caracteres alfabéticos e numéricos.

A transformação de Mascaramento de Dados não mascara o protocolo da URL. Por exemplo, se a URL for `http://www.yahoo.com`, a transformação de Mascaramento de Dados poderá retornar `http://MgLaHjCa.VsD/`. A transformação de Mascaramento de Dados pode gerar uma URL que não é válida.

Nota: A transformação de Mascaramento de Dados sempre retorna caracteres ASCII para uma URL.

Arquivo de valor padrão

Quando o tipo ou o formato dos dados de origem for inválido para uma máscara, o Serviço de Integração aplicará uma máscara padrão para os dados. O Serviço de Integração aplica valores mascarados do arquivo de valores padrão. Você pode editar o arquivo de valor padrão para alterar os valores padrão.

O arquivo de valor padrão é um arquivo XML no local a seguir:

```
<PowerCenter Installation Directory>\infa_shared\SrcFiles\defaultValue.xml
```

O arquivo `defaultValue.xml` contém os seguintes pares de nome-valor:

```
<?xml version="1.0" standalone="yes" ?>
<defaultValue
  default_char = "X"
  default_digit = "9"
  default_date = "11/11/1111 00:00:00"
  default_email = "abc@xyz.com"
  default_ip = "99.99.9.999"
  default_url = "http://www.xyz.com"
  default_phone = "999999999"
  default_ssn = "999-99-9999"
  default_cc = "9999 9999 9999 9999"
  default_seed = "500"
/>
```

Propriedades da sessão de Transformação de Mascaramento de Dados

Você pode configurar as propriedades de sessão da transformação de Mascaramento de Dados para melhorar o desempenho.

Configure as seguintes propriedades de sessão:

Tamanho do Cache

O tamanho do cache do dicionário na memória principal. Aumente o tamanho de memória para melhorar o desempenho. Mínimo de tamanho recomendado é 32 MB para 100.000 registros. O padrão é 8 MB.

Diretório de Cache

A localização do cache do dicionário. Você deve ter permissões de gravação para o diretório. O padrão é \$PMCacheDir.

Tabela de armazenamento compartilhado

Habilita o compartilhamento de tabela de armazenamento entre instâncias de transformação de Mascaramento de Dados. Ative a tabela de armazenamento compartilhado quando duas instâncias da transformação de Mascaramento de Dados usarem a mesma coluna do dicionário para a conexão de banco de dados, valor de semente e localidade. Você também pode habilitar a tabela de armazenamento compartilhado quando duas portas na mesma transformação de Mascaramento de Dados usarem a mesma coluna do dicionário para a conexão, semente, e a localidade. Desative a tabela de armazenamento compartilhado quando as Transformações de Mascaramento de Dados ou as portas não compartilharem a coluna do dicionário. Por padrão, essa opção fica desativada.

Intervalo de confirmação de armazenamento

O número de linhas a serem confirmadas de uma vez para a tabela de armazenamento. Aumente o valor para melhorar o desempenho. Configure o intervalo de confirmação quando não configurar a tabela de armazenamento compartilhado. O padrão é 100.000.

Criptografar o Armazenamento

Criptografa as tabelas de armazenamento, tais como IDM_SUBSTITUTION_STORAGE e IDM_EXPRESSION_STORAGE. Verifique se você criptografou dados em tabelas de armazenamento antes de ativar a propriedade de criptografia de armazenamento. Desmarque essa opção se você não deseja criptografar as tabelas de armazenamento. Por padrão, essa opção fica desativada.

Chave de Criptografia de Armazenamento

A transformação de Mascaramento de Dados criptografa o armazenamento com base na chave de criptografia de armazenamento. Use a mesma chave de criptografia para cada execução de sessão da mesma instância da transformação de Mascaramento de Dados.

Regras e Diretrizes para Transformações Mascaramento de Dados

Utilize as seguintes regras e diretrizes ao configurar uma transformação de Mascaramento de Dados:

- A transformação de Mascaramento de Dados não mascara valores nulos. Se os dados de origem contiverem valores nulos, a transformação de Mascaramento de Dados retornará valores nulos. Para substituir valores nulos, adicione uma transformação upstream que permita valores padrão definidos pelo usuário para portas de entrada.
- Quando o tipo ou o formato dos dados de origem for inválido para uma máscara, o Serviço de Integração aplicará uma máscara padrão para os dados. O Serviço de Integração aplica valores mascarados de um arquivo de valores padrão.
- A transformação de Mascaramento de Dados retorna um número de Previdência Social inválido com o mesmo formato e código de área que a origem. Se a Administração do Seguro Social tiver emitido mais da metade dos números para uma área, pode ser que a transformação de Mascaramento de Dados não consiga retornar números de Previdência Social inválidos exclusivos com o mascaramento de chave.

CAPÍTULO 6

Exemplos de Mascaramento de Dados

Este capítulo inclui os seguintes tópicos:

- [Arquivos de Pesquisa de Nome e Endereço, 157](#)
- [Substituindo Dados com a Transformação Pesquisa, 157](#)
- [Mascarando Dados com a Transformação Expressão, 161](#)

Arquivos de Pesquisa de Nome e Endereço

A instalação do Mascaramento de Dados traz consigo diversos arquivos simples que contêm nomes, endereços e nomes de empresa para mascarar dados. É possível configurar uma transformação de Pesquisa para que recupere, desses arquivos, nomes próprios, sobrenomes e endereços, todos aleatórios.

Ao instalar o componente de serviços de Mascaramento de Dados, o instalador coloca os seguintes arquivos na pasta `server\infa_shared\LkpFiles`:

- **Address.dic.** Endereços de arquivos simples. Cada registro tem um número de série, rua, cidade, estado, código postal e país.
- **Company_names.dic.** Arquivo simples de nomes de empresa. Cada registro tem um número de série e um nome de empresa.
- **Firstnames.dic.** Arquivo simples de nomes. Cada registro tem um número de série, um nome e um código de gênero.
- **Surnames.dic.** Arquivo simples de sobrenomes. Cada registro tem um número de série e um sobrenome.

Substituindo Dados com a Transformação Pesquisa

Você pode substituir uma coluna de dados por dados semelhantes, mas não relacionados. A substituição é uma maneira eficiente de mascarar dados confidenciais com dados que têm uma aparência realística.

O exemplo a seguir mostra como configurar várias transformações Pesquisa para recuperar dados de teste e substituí-los para dados de origem. Crie um mapeamento Mascaramento de Dados para mascarar os campos confidenciais na tabela CUSTOMERS_PROD.

O exemplo inclui os tipos de mascaramento a seguir:

- Substituição de nome e endereço nas tabelas de pesquisa
- Mascaramento de chaves
- Embaçamento
- Formatos especiais de máscara

Nota: Este exemplo é o mapeamento M_CUSTOMERS_MASKING.xml que você pode importar para seu repositório a partir da pasta `client\samples`.

A tabela de banco de dados de cliente denominada Customers_Prod contém dados confidenciais. Você quer usar os dados do cliente em um cenário de teste, mas deseja manter a segurança. Mascare os dados em cada coluna e grave os dados de teste em uma tabela de destino denominada Customers_Test.

A tabela Customers_Prod contém as seguintes colunas:

Coluna	Tipo de Dados
CustID	Número inteiro
FullName	String
Endereço	String
Telefone	String
Fax	String
CreateDate	Data
E-mail	String
SSN	String
CreditCard	String

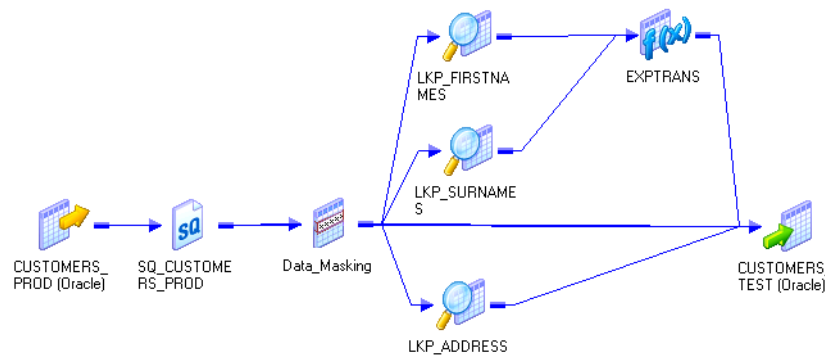
Você pode criar um mapeamento que pesquise valores substitutos em arquivos de dicionário. A transformação de Mascaramento de Dados mascara os dados do cliente com valores dos arquivos de dicionário. Os arquivos incluem um arquivo de nomes, um de sobrenomes e um de endereços.

A seguinte tabela lista os arquivos que estão localizados na pasta `server\\infa_shared\LkpFiles`:

Arquivo	Número de Registros	Campos	Descrição
Firstnames.dic	21,000	SNO, Gênero, Nome	Lista alfabética de nomes. O número de série vai de 1 a 21000. Gênero indica se o nome é masculino ou feminino.
Surnames.dic	81,000	SNO, Sobrenome	Lista alfabética de sobrenomes. O número de série vai de 1 a 81,000.
Address.dic	13,000	SNO, Rua, Cidade, Estado, CEP, País	Lista completa de endereços. O número de série vai de 1 a 13,000.

Nota: O Informatica inclui a coluna de gênero no arquivo Firstnames.dic para que você possa criar arquivos de origem de pesquisa separados por gênero. Você poderá usar a coluna de gênero em uma condição de pesquisa se precisar mascarar um nome masculino com um nome masculino e um nome feminino com um nome feminino.

A figura a seguir mostra o mapeamento que você pode importar:



O mapeamento contém as seguintes transformações junto com uma origem e um destino:

- **Qualificador de Origem.** Passa dados do cliente para a transformação de Mascaramento de Dados. Ele passa a coluna CustID para várias portas na transformação:
 - **CustID.** Número do cliente.
 - **Randid1.** Gerador de números aleatórios para pesquisas de nomes.
 - **Randid2.** Gerador de números aleatórios para pesquisas de sobrenomes.
 - **Randid3.** Gerador de números aleatórios para pesquisas de endereços.
- **Transformação de Mascaramento de Dados.** Cria números aleatórios para pesquisar o nome, o sobrenome e o endereço de substituição. Aplica formatos especiais de máscara para número de telefone, número de fax, endereço de e-mail e número de cartão de crédito. A transformação de Mascaramento de Dados mascara as seguintes colunas:

Porta de entrada	Tipo de Mascaramento	Regras de Mascaramento	Descrição	Destino de Saída
CustID	Chave	Semente = 934	CustID é a coluna de chave primária. Ela precisa ser mascarada com um número aleatório que seja repetido e determinístico.	Customers_Test
Randid1	Aleatório	Intervalo Mínimo = 0 Máximo = 21000	O número aleatório da pesquisa de nome na transformação LKUP_Firstnames.	LKUP_Firstnames
Randid2	Aleatório	Intervalo Mínimo = 0 Máximo = 13000	O número aleatório da pesquisa de sobrenome na transformação LKUP_Surnames.	LKUP_Surnames

Porta de entrada	Tipo de Mascaramento	Regras de Mascaramento	Descrição	Destino de Saída
Randid3	Aleatório	Intervalo Mínimo = 0 Máximo = 81000	O número aleatório da pesquisa de endereço na transformação LKUP_Address.	LKUP_Address
Telefone	Telefone	-	O número de telefone tem o mesmo formato que o número de telefone de origem.	Customers_Test
Fax	Telefone	-	O número de telefone tem o mesmo formato que o número de telefone de origem.	Customers_Test
CreatedDate	Aleatório	Embaçamento Unidade = Ano Limite inferior = 1 Limite Superior = 1	A data aleatória que está dentro de um ano do ano de origem.	Customers_Test
E-mail	Endereço de E-mail	-	Endereço de e-mail que tem o mesmo formato que o original.	Customers_Test
SSN	SSN	-	SSN não está no arquivo highgroup.txt.	Customers_Test
CreditCard	Cartão de Crédito	-	O cartão de crédito tem os mesmos seis primeiros dígitos que a origem e tem uma soma de verificação válida.	Customers_Test

- **LKUP_Firstnames.** Faz uma pesquisa de arquivo simples em Firstnames.dic. A transformação recupera o registro com o número de série igual ao número aleatório Randid1. Esta é a condição de pesquisa:

```
SNO = out_RANDID1
```

A transformação LKUP_Firstnames passa o nome mascarado para a transformação Expressão Exptrans.

- **LKUP_Surnames.** Faz uma pesquisa de arquivo simples no arquivo Surnames.dic. Ela recupera o registro com o número de série igual a Randid2. A transformação LKUP_Firstnames passa um sobrenome mascarado para a transformação Expressão Exptrans.
- **Exptrans.** Combina o nome e o sobrenome e retorna um número completo. A transformação Expressão passa o nome completo para o destino Customers_Test.

Esta é a expressão para combinar o nome e o sobrenome:

```
FIRSTNAME || ' ' || SURNAME
```

- **LKUP_Address.** Faz uma pesquisa de arquivo plano no arquivo Address.dic. Recupera o registro de endereço com o número de série igual a Randid3. A transformação Pesquisa passa as colunas no endereço para o destino.

Você pode usar a tabela Customer_Test em um ambiente de teste.

Mascarando Dados com a Transformação Expressão

Use a transformação Expressão com a transformação de Mascaramento de Dados para manter uma relação entre duas colunas depois de mascarar uma delas.

Por exemplo, quando você mascara informações de conta que contêm datas iniciais e finais para apólices de seguro, é conveniente manter o comprimento da apólice em cada registro mascarado. Use uma transformação de Mascaramento de Dados para mascarar todos os dados, exceto a data final. Use uma transformação Expressão para calcular o comprimento da apólice adicioná-lo à data inicial mascarada.

Esse exemplo inclui os seguintes tipos de mascaramento:

- Chave
- Embaçamento da data
- Embaçamento do número
- Formatação da máscara

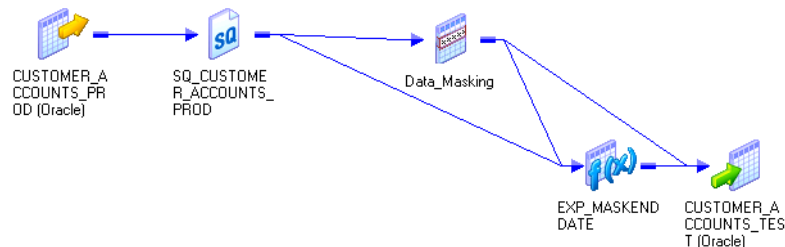
Nota: Este exemplo é o mapeamento M_CUSTOMER_ACCOUNTS_MASKING.xml que você pode importar para seu repositório a partir da pasta `client\samples`.

A tabela de banco de dados de cliente denominada Customers_Prod contém dados confidenciais. Mascare os dados em cada coluna e grave os dados de teste em uma tabela de destino denominada Customers_Test.

Mascare as seguintes colunas Customer_Accounts_Prod:

Coluna	Tipo de Dados
AcctID	String
CustID	Número Inteiro
Saldo	Double
StartDate	Data e hora
EndDate	Data e hora

A figura a seguir mostra o mapeamento que você pode importar:



O mapeamento contém as seguintes transformações junto com uma origem e um destino:

- **Qualificador de Origem.** Transfere AcctID, CustID, Equilíbrio e Start_Date à transformação de Mascaramento de Dados. Transfere as colunas Start_Date e End_Date para uma transformação Expressão.

- **Transformação Mascaramento de Dados.** Mascara todas as colunas exceto End_Date. A transformação de Mascaramento de Dados transfere as colunas mascaradas para o destino. Ela transfere a data inicial da apólice, a data final e a data inicial mascarada para a transformação Expressão.

A Transformação Mascaramento de Dados mascara as seguintes colunas:

Porta de entrada	Tipo de Mascaramento	Regras de Mascaramento	Descrição	Destino de Saída
AcctID	Aleatório	Formato de máscara AA+DDDDD Caracteres de Substituição da String de Resultados ABCDEFGHIJKLMNOPQRSTUVWXYZ	Os primeiros dois caracteres são caracteres alfabéticos em maiúscula. O terceiro caractere é um traço e não é mascarado. Os últimos cinco caracteres são números.	Destino Customer_Account_Test
CustID	Chave	Semente = 934	A semente é 934. A máscara CustID é determinística.	Destino Customer_Account_Test
Saldo	Aleatório	Embaçamento Por cento Limite inferior = 10 Limite superior = 10	O equilíbrio mascarado está dentro de dez por cento do equilíbrio de origem.	Destino Customer_Account_Test
Start_Date	Aleatório	Embaçamento Unidade = Ano Limite inferior = 2 Limite Superior = 2	O start_date mascarado está dentro de dois anos da data de origem.	Destino Customer_Account_Test Exp_MaskEndDatetransformation

- **Transformação Expressão.** Calcula a data final mascarada. Calcula o tempo entre as datas inicial e final. Adiciona o tempo à data inicial mascarada, para determinar a data final mascarada.

As expressões para gerar a data final mascarada são:

```
DIFF = DATE_DIFF(END_DATE, START_DATE, 'DD')
out_END_DATE = ADD_TO_DATE(out_START_DATE, 'DD', DIFF)
```

A transformação Expressão transfere out_END_DATE para o destino.

CAPÍTULO 7

Transformação Expressão

Este capítulo inclui os seguintes tópicos:

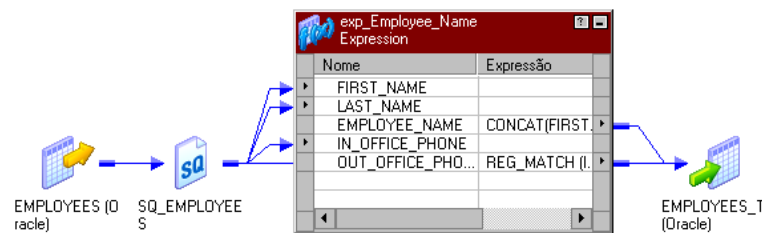
- [Visão geral da transformação de Expressão, 163](#)
- [Componentes da Transformação Expressão, 163](#)
- [Configurando portas, 164](#)
- [Criação de uma Transformação de Expressão, 165](#)

Visão geral da transformação de Expressão

Use a transformação da Expressão para calcular valores em uma única linha. Por exemplo, você pode precisar ajustar os salários do funcionário, concatenar o nome e o sobrenome ou converter cadeias de caracteres em números. Você também pode usar a transformação de Expressão para testar instruções condicionais antes de transferir os resultados para um destino ou outras transformações. A transformação de Expressão é uma transformação passiva.

Use a transformação da Expressão para realizar cálculos não agregados. Para realizar cálculos envolvendo múltiplas linhas, como somas ou médias, use a transformação do Agregador.

A figura a seguir mostra um mapeamento simples com uma transformação da Expressão usada para concatenar o nome e o sobrenome dos funcionários a partir da tabela FUNCIONÁRIOS:



Você pode avaliar expressões configuradas no Editor de Expressão.

Componentes da Transformação Expressão

Você pode criar uma transformação de Expressão no Transformation Developer ou Mapping Designer.

Uma transformação expressão contém as seguintes guias:

- **Transformação.** Digite o nome e a descrição da transformação. A convenção de nomenclatura para uma transformação de Expressão é `EXP_TransformationName`. Você também pode tornar a transformação reutilizável.
- **Portas.** Crie e configure portas.
- **Propriedades** Configure o nível de rastreamento para determinar a quantidade de detalhes de transação relatados no arquivo de registro da sessão.
- **Extensões de Metadados.** Especifique o nome, o tipo de dados, a precisão e o valor da extensão. Também é possível criar extensões de metadados reutilizáveis.

Configurando portas

Você pode criar e modificar portas na guia Portas.

Configure os componentes a seguir na guia Portas:

- **Nome da porta.** Nome da porta.
- **Tipo de dados, precisão e escala.** Configure o tipo de dados e defina a precisão e a escala de cada porta.
- **Tipo de porta.** Uma porta pode ser de entrada, saída, entrada/saída ou variável. As portas de entrada recebem dados e as portas de saída transferem dados. As portas de entrada/saída transferem dados não alterados. As portas variáveis armazenam dados temporariamente e têm a capacidade de armazenar valores pelas linhas.
- **Expressão.** Use o Editor de Expressão para inserir expressões. As expressões utilizam a linguagem de transformação, que conta com funções semelhantes a SQL, para realizar cálculos.
- **Valores padrão e descrição.** Defina o valor padrão para portas e adicione a descrição.

Cálculo de Valores

Para calcular valores para uma única linha usando a transformação Expressão, você deve incluir as portas a seguir:

- **Portas de entrada ou entrada/saída.** Fornece os valores usados em um cálculo. Por exemplo, se você precisa calcular o preço total de um pedido, crie duas portas de entrada ou entrada/saída. Uma porta fornece o preço da unidade e a outra informa a quantidade solicitada.
- **Portas de saída.** Fornece o valor de retorno da expressão. Você insere a expressão como uma opção de configuração para a porta de saída. Você também pode configurar um valor padrão para cada porta.

Você pode inserir múltiplas expressões em uma única transformação Expressão, criando uma expressão para cada porta de saída. Por exemplo, você pode calcular tipos diferentes de impostos retidos na origem a partir do pagamento de cada funcionário, como imposto de renda local e federal, previdência e INSS. Uma vez que todos estes cálculos exigem o salário do funcionário, a categoria de retenção, e podem exigir a taxa tributária correspondente, você pode criar portas de entrada/saída para o salário e a categoria de retenção e uma porta de saída separada para cada cálculo.

Criação de uma Transformação de Expressão

Use o seguinte procedimento para criar uma transformação de Expressão:

1. No Mapping Designer, abra uma transformação Qualificador de origem.
2. Clique em Transformação > Criar. Selecione a transformação de Expressão.
3. Insira um nome e clique em Concluído.
4. Selecione e arraste as portas do qualificador de origem ou outras transformações para adicionar à transformação de Expressão.
Você também pode abrir a transformação e criar as portas manualmente.
5. Clique duas vezes na barra de título e clique na guia Portas. Você pode criar portas de saída e de variável dentro da transformação.
6. Atribua o tipo de dados de porta, a precisão e a escala para corresponder ao valor de retorno da expressão.
7. Na seção Expressão de uma porta de saída ou variável, abra o Editor de Expressão.
8. Insira uma expressão.
9. Clique em Validar para verificar a sintaxe da expressão.
10. Para refletir as alterações mais recentes da condição de expressão na seção da expressão de teste no painel direito, clique em Atualizar.
11. No painel direito, insira valores de exemplo para as portas de entrada usadas na expressão.
12. Para avaliar a expressão, clique em Avaliar.
13. Clique em OK.
14. Crie transformações reutilizáveis na guia Transformação.
Nota: Depois de tornar a transformação reutilizável, você não pode copiar portas do qualificador de origem ou outras transformações. Você *pode* criar portas manualmente dentro da transformação.
15. Configure o nível de rastreamento na guia Propriedades.
16. Adicione extensões de metadados na guia Extensões de Metadados.
17. Clique em OK.
18. Conecte as portas de saída a um destino ou transformação downstream.

CAPÍTULO 8

Transformação Procedimento Externo

Este capítulo inclui os seguintes tópicos:

- [Visão geral da Transformação Procedimento Externo, 166](#)
- [Configurando Propriedades da Transformação Procedimento Externo, 168](#)
- [Desenvolvendo Procedimentos COM, 170](#)
- [Desenvolvendo Procedimentos Externos da Informatica, 178](#)
- [Distribuindo Procedimentos Externos, 186](#)
- [Observações de Desenvolvimento, 187](#)
- [Variáveis do Processo de Serviço nas Propriedades de Inicialização, 193](#)
- [Interfaces do Procedimento Externo, 194](#)

Visão geral da Transformação Procedimento Externo

As transformações Procedimento Externo funcionam em conjunção com os procedimentos que você cria fora da interface do Designer, para ampliar a funcionalidade do PowerCenter.

Embora as transformações padrão lhe proporcionem uma ampla variedade de opções, há ocasiões em que você pode estender a funcionalidade fornecida com o PowerCenter. Por exemplo, a série de transformações padrão, como Expressão e Filtro, pode não fornecer a funcionalidade de que você precisa. Se você for um programador experiente, pode desejar desenvolver funções complexas dentro de uma biblioteca de link dinâmico (DLL) ou biblioteca compartilhada do UNIX, em vez de criar as transformações Expressão necessárias em um mapeamento.

Para obter este tipo de extensibilidade, use a interface de invocação dinâmica Troca de Transformação (TX) criada no PowerCenter. Usando o TX, você pode criar uma transformação Procedimento Externo do Informatica e vinculá-la a um procedimento externo que você desenvolveu. Você pode vincular transformações Procedimento Externo a dois tipos de procedimentos externos:

- Procedimentos externos COM (disponível somente no Windows)
- Procedimentos externos do Informatica (disponível no Windows, AIX, Linux e Solaris)

Para usar o TX, você deve ser um programador experiente em C, C++ ou Visual Basic.

Utilize um código multithread nos procedimentos externos.

Compatibilidade de Página de Código

Quando o Serviço de Integração executa no modo ASCII, o procedimento externo pode processar dados no ASCII de 7 bits. Quando o Serviço de Integração executa no modo Unicode, o procedimento externo pode processar dados compatíveis de maneira bidirecional com a página de código do Serviço de Integração.

Configure o Serviço de Integração para executar no modo Unicode se a DLL ou biblioteca compartilhada do procedimento externo contém caracteres multibytes. Os procedimentos externos devem usar a mesma página de código que o Serviço de Integração para interpretar cadeias de caracteres de entrada do Serviço de Integração e criar cadeias de saída que contenham caracteres multibytes.

Configure o Serviço de Integração para executar no modo ASCII ou Unicode se a DLL ou biblioteca compartilhada do procedimento externo contém apenas caracteres ASCII.

Procedimentos Externos e Transformações de Procedimentos Externos

Há dois componentes do TX: *procedimentos externos* e *transformações de Procedimentos Externos*.

Um *procedimento externo* está separado do Serviço de Integração. Consiste de código em C, C++ ou Visual Basic escrito por um usuário para definir uma transformação. Este código é compilado e vinculado em uma DLL ou biblioteca compartilhada, que é carregada pelo Serviço de Integração em tempo de execução. Um procedimento externo é “acoplado” a uma transformação de Procedimento Externo.

Uma *transformação de Procedimento Externo* é criada no Designer. É um objeto que reside no repositório Informatica e serve a diversos propósitos:

1. Contém os metadados que descrevem o seguinte procedimento externo. É por meio desses metadados que o Serviço de Integração reconhece a “assinatura” (número e tipos de parâmetros e tipos de valor de retorno, se houver) do procedimento externo.
2. Permite que um procedimento externo tenha referência em um mapeamento. Acrescentando uma instância de uma transformação de Procedimento Externo a um mapeamento, chama-se o acoplamento do procedimento externo a essa transformação.

Nota: É possível criar um Procedimento Externo conectado ou não.

3. Ao desenvolver procedimentos externos da Informatica, a transformação de Procedimentos Externos fornece as informações necessárias para gerar stubs de procedimentos externos da Informatica.

Propriedades da Transformação Procedimento Externo

Crie transformações reutilizáveis de Procedimento externo no Transformation Developer, e adicione instâncias da transformação ao mapeamento. Você não pode criar transformações de Procedimento externo no Mapping Designer ou no Mapplet Designer.

As transformações de Procedimento externo retornam uma ou nenhuma linha de saída para cada linha de entrada.

Na guia Propriedades da transformação de Procedimento externo, insira apenas caracteres ASCII nos campos Identificador de módulo/programático e Nome do procedimento. Você não pode inserir caracteres multibytes nestes campos. Na guia Portas da transformação de Procedimento externo, insira somente caracteres ASCII para os nomes de porta. Você não pode inserir caracteres multibytes para os nomes de porta da transformação de Procedimento externo.

Procedimentos Externos de COM Versus Informatica

A tabela a seguir descreve as diferenças entre procedimentos externos da COM e da Informatica:

	COM	Informatica
Tecnologia	Usa tecnologia da COM	Usa tecnologia proprietária da Informatica
Sistema operacional	É executado somente em Windows	É executado em todas as plataformas compatíveis com o Serviço de Integração: Windows, AIX, HP, Linux, Solaris
Linguagem	C, C++, VC++, VB, Perl, VJ++	Apenas C++

O Exemplo do Banksoft

As seções a seguir utilizam um exemplo denominado BankSoft para ilustrar como desenvolver procedimentos COM e Informatica. O exemplo do BankSoft utiliza uma função financeira, VF, para ilustrar como desenvolver e chamar um procedimento externo. O procedimento VF calcula o valor futuro de um investimento com base nos pagamentos regulares e em uma taxa de juros constante.

Configurando Propriedades da Transformação Procedimento Externo

Configure as propriedades da transformação na guia Propriedades.

A seguinte tabela descreve as propriedades da transformação Procedimento Externo:

Propriedade	Descrição
Tipo	Tipo de procedimento externo. Use os seguintes tipos: <ul style="list-style-type: none">- COM- Informatica O padrão é Informatica.
Identificador do Módulo/ Programático	Um módulo é um nome básico da DLL (no Windows) ou do objeto compartilhado (no UNIX) que contém os procedimentos externos. Ele determina o nome da DLL ou objeto compartilhado no sistema operacional. Insira apenas caracteres ASCII. Um identificador programático, ou ProgID, é o nome lógico de uma classe. No Designer, você faz referência a classes COM por meio de ProgIDs. Internamente, classes são identificadas por CLSIDs numéricas. Por exemplo: {33B17632-1D9F-11D1-8790-0000C044ACF9} O formato padrão de um ProgID é <i>Projeto.Classe[.Versão]</i> . Insira apenas caracteres ASCII.
Nome do Procedimento	Nome do procedimento externo. Insira apenas caracteres ASCII.

Propriedade	Descrição
Local em tempo de execução	<p>Localização que contém a DLL ou biblioteca compartilhada. Insira um caminho relativo ao nó do Serviço de Integração que executa a sessão de Procedimento Externo. Se você inserir \$PMExtProcDir, o Serviço de Integração examinará o diretório especificado pelo variável de processo \$PMExtProcDir para localizar a biblioteca.</p> <p>Se essa propriedade estiver em branco, o Serviço de Integração usará a variável de ambiente definida no nó do Serviço de Integração para localizar a DLL ou a biblioteca compartilhada.</p> <p>Você pode codificar um caminho como a Localização do Tempo de Execução. Isso não é recomendado, pois o caminho é específico somente para uma máquina.</p> <p>É necessário copiar todas as DLLs ou bibliotecas compartilhadas para a localização de tempo de execução ou para a variável de ambiente definida no nó do Serviço de Integração. O Serviço de Integração não consegue carregar o procedimento quando não consegue localizar a DLL, a biblioteca compartilhada ou um arquivo de referência.</p> <p>O padrão é \$PMExtProcDir.</p>
Nível de Rastreamento	<p>Quantidade de detalhes de transações relatados no arquivo de log da sessão. Use os seguintes níveis de rastreamento:</p> <ul style="list-style-type: none"> - Conciso - Normal - Inicialização Detalhada - Dados Detalhados <p>O padrão é Normal.</p>
É Particionável	<p>Indica se você pode criar múltiplas partições em um pipeline que usa essa transformação. Use os seguintes valores:</p> <ul style="list-style-type: none"> - Não. A transformação não pode ser particionada. A transformação e outras transformações no mesmo pipeline estão limitadas a uma partição. - Localmente. A transformação pode ser particionada, mas o Serviço de Integração deve executar todas as partições do pipeline no mesmo nó. Escolha Local quando diferentes partições da transformação BAPI/RFC devem compartilhar objetos na memória. - Na Grade. A transformação pode ser particionada, e o Serviço de Integração pode distribuir cada partição em nós diferentes. <p>O padrão é Não.</p>
A Saída é Repetível	<p>Indica se a transformação gerará linhas na mesma ordem entre execuções da sessão. O Serviço de Integração poderá retomar uma sessão do último ponto de verificação quando a saída for repetível e determinista. Use os seguintes valores:</p> <ul style="list-style-type: none"> - Sempre. A ordem dos dados de saída será consistente entre as execuções de sessão mesmo se a ordem de entrada for inconsistente entre elas. - Com Base na Ordem de Entrada. A transformação produzirá dados repetíveis entre execuções da sessão quando a ordem dos dados de saída de todos os grupos de entrada for consistente entre as execuções da sessão. Se os dados de entrada de qualquer grupo de entrada não estiverem ordenados, a saída não será ordenada. - Nunca. A ordem dos dados de saída é inconsistente entre as execuções de sessão. Não será possível configurar a retomada da recuperação do último ponto de verificação se uma transformação não produzir dados repetíveis. <p>O padrão é Baseada na Ordem de Entrada.</p>
Saída é Determinística	<p>Indica se a transformação gera dados de saída consistentes entre as execuções de sessão. É necessário ativar esta propriedade para executar a recuperação nas sessões que usam esta transformação.</p> <p>Por padrão, essa opção fica desativada.</p>

Aviso: Se você configurar uma transformação como repetível e determinística, será sua responsabilidade garantir que os dados sejam repetíveis e determinísticos. Se você tentar recuperar uma sessão com transformações que não gerem os mesmos dados entre a sessão e a recuperação, o processo de recuperação poderá resultar em dados corrompidos.

A tabela a seguir descreve as variáveis de ambiente que o Serviço de Integração utiliza para localizar a DLL ou objeto compartilhado nas várias plataformas do local do tempo de execução:

Tabela 1. Variáveis de ambiente

Sistema Operacional	Variável de ambiente
Windows	PATH
AIX	LIBPATH
HPUX	SHLIB_PATH
Linux	LD_LIBRARY_PATH
Solaris	LD_LIBRARY_PATH

Desenvolvendo Procedimentos COM

É possível desenvolver procedimentos externos de COM com Microsoft Visual C++ ou Visual Basic. As seções a seguir descrevem como criar procedimentos externos COM usando Visual C++ e como criar os mesmos procedimentos usando Visual Basic.

Etapas para Criar um Procedimento COM

Para criar um procedimento externo COM, execute as etapas a seguir:

1. Usando o Microsoft Visual C++ ou o Visual Basic, crie um projeto.
2. Defina uma classe com uma interface do *IDispatch*.
3. Adicione um método à interface. Esse método é o procedimento externo que será invocado de dentro do Serviço de Integração.
4. Compile e vincule a classe a uma biblioteca de vínculo dinâmico.
5. Registre a classe no Registro do Windows local.
6. Importe o procedimento COM no Transformation Developer.
7. Crie um mapeamento com o procedimento COM.
8. Crie uma sessão usando o mapeamento.

Tipo de Serviço de Procedimento Externo COM

O Serviço de Integração somente suporta servidores COM do processo que tenham *Tipo de Servidor: Biblioteca de link dinâmico*. Isto é feito para melhorar o desempenho. É mais eficiente ao processar grandes quantidades de dados, a fim de processar os dados no mesmo processo, em vez de encaminhá-los para um processo separado na mesma máquina ou em uma máquina remota.

Usando o Visual C++ para Desenvolver Procedimentos COM

Desenvolvedores C++ podem usar o Visual C++ versão 5.0 ou superior para desenvolver procedimentos COM. A primeira tarefa é criar um projeto.

TÓPICOS RELACIONADOS:

- [“Distribuindo Procedimentos Externos” na página 186](#)
- [“Classes Wrapper para Funções do VB ou Bibliotecas C/C++ Pré-existentes” na página 189](#)

Etapa 1. Criar um Projeto ATL COM AppWizard

1. Inicie o Visual C++ e clique em Arquivo > Novo.
2. Na caixa de diálogo que é exibida, selecione a guia Projetos.
3. Digite o nome e a localização do projeto.
No exemplo do BankSoft, insira *COM_VC_Banksoft* como o nome do projeto e *c:\COM_VC_Banksoft* como o diretório.
4. Selecione a opção *ATL COM AppWizard* na caixa de listagem de projetos e clique em OK.
Aparecerá um assistente que é usado para criar projetos COM no Visual C++.
5. Defina o Tipo do Servidor como Biblioteca de Vínculo Dinâmico (DLL), selecione a opção *Suporte a MFC* e clique em Concluir.
A última página do assistente aparecerá.
6. Clique em OK para retornar ao Visual C++.
7. Adicione uma classe ao novo projeto.
8. Na próxima página do assistente, clique no botão OK. O Developer Studio criará os arquivos básicos do projeto.

Etapa 2. Adicionar um Objeto ATL a um Projeto

1. Na janela Espaço de Trabalho, selecione a guia Modo de Exibição de Classe, clique com o botão direito do mouse nas classes *COM_VC_BankSoft.BSoftFin* do item da árvore e escolha Novo Objeto ATL no menu local que aparecerá.
2. Realce o item Objetos na caixa de listagem do lado esquerdo e selecione *Objeto Simples* na lista de tipos de objeto.
3. Clique em Avançar.
4. No campo Nome Curto, insira um nome curto para a classe que deseja criar.
No exemplo do BankSoft, utilize o nome *BSoftFin*, já que você está desenvolvendo uma função financeira para a empresa fictícia BankSoft. À medida que você digitar no campo Nome Curto, o assistente preencherá nomes sugeridos nos outros campos.
5. Insira o identificador programático da classe.
No exemplo do BankSoft, altere o campo ProgID (identificador programático) para *COM_VC_BankSoft.BSoftFin*.
Um identificador programático, ou ProgID, é o nome legível de uma classe. Internamente, as classes são identificadas por CLSIDs numéricas. Por exemplo:

```
{33B17632-1D9F-11D1-8790-0000C044ACF9}
```

O formato padrão de um ProgID é *Projeto.Classe[.Versão]*. No Designer, você faz referência a classes COM por meio de ProgIDs.

6. Selecione a guia Atributos e defina o modelo de threading como *Livre*, a interface como *Dual* e a configuração de agregação como *Não*.
7. Clique em OK.

Agora que você tem uma definição de classe básica, poderá adicionar um método a ela.

Etapa 3. Adicionar os Métodos Obrigatórios à Classe

1. Retorne à guia Exibir Classes da janela Espaço de Trabalho.
2. Expanda a exibição de árvore.
No exemplo do Banksoft, expanda *COM_VC_BankSoft*.
3. Clique com o botão direito na classe recém-adicionada.
No exemplo do Banksoft, clique com o botão direito no item da árvore *IBSoftFin*.
4. Clique no item de menu Adicionar Método e insira o nome do método.
No exemplo BankSoft, você insere *FV*.
5. No campo Parâmetros, insira a assinatura do método.

Para FV, insira o seguinte:

```
[in] double Rate,  
[in] long nPeriods,  
[in] double Payment,  
[in] double PresentValue,  
[in] long PaymentType,  
[out, retval] double* FV
```

Essa assinatura é expressa nos termos da MIDL (Linguagem de Descrição de Interface Microsoft). Para ler uma descrição completa da MIDL, consulte a MIDL referência de linguagem. Observe que:

- **[in]** indica que o parâmetro é um parâmetro de entrada.
- **[out]** indica que o parâmetro é um parâmetro de saída.
- **[out, retval]** indica que o parâmetro é o valor de retorno do método.

Além disso, todos os parâmetros [out] são passados por referência. No exemplo do BankSoft, o parâmetro FV é dobrado.

6. Clique em OK.
O Developer Studio adiciona ao projeto um stub para o método adicionado.

Etapa 4. Preencher o Stub de Método com uma Implementação

1. No exemplo BankSoft, retorne à guia Modo de Exibição de Classe na janela Espaço de Trabalho e expanda o item de classes *COM_VC_BankSoft*.
2. Expanda o item *CBSOFTFin*.
3. Expanda o item *IBSOFTFin* abaixo do item acima.
4. Clique com o botão direito do mouse no item *VF* e escolha *Ir para Definição*.
5. Posicione o cursor na janela de edição na linha depois do comentário *TODO* e adicione o código a seguir:

```
double v = pow((1 + Rate), nPeriods);  
*FV = -(  
    (PresentValue * v) +
```

```

        (Payment * (1 + (Rate * PaymentType))) * ((v - 1) / Rate)
    );

```

Como você faz referência à função `pow`, deverá adicionar a seguinte instrução de pré-processador depois de todas as outras instruções de inclusão no início do arquivo:

```
#include <math.h>
```

A última etapa é construir a DLL. Ao construí-la, você registra o procedimento COM com o registro do Windows.

Etapa 5. Construir o Projeto

1. Clique no menu suspenso **Compilar**.
2. Selecione **Recompilar Tudo**.

Ao compilar o projeto, o Developer Studio gera estas saídas:

```

-----Configuration: COM_VC_BankSoft - Win32 Debug-----

Performing MIDL step

Microsoft (R) MIDL Compiler Version 3.01.75

Copyright (c) Microsoft Corp 1991-1997. All rights reserved.

Processing .\COM_VC_BankSoft.idl

COM_VC_BankSoft.idl

Processing C:\msdev\VC\INCLUDE\oidl.idl

oidl.idl

Processing C:\msdev\VC\INCLUDE\objidl.idl

objidl.idl

Processing C:\msdev\VC\INCLUDE\unknwn.idl

unknwn.idl

Processing C:\msdev\VC\INCLUDE\wtypes.idl

wtypes.idl

Processing C:\msdev\VC\INCLUDE\ocidl.idl

ocidl.idl

Processing C:\msdev\VC\INCLUDE\oleidl.idl

oleidl.idl

Compiling resources...

Compiling...

StdAfx.cpp

Compiling...

COM_VC_BankSoft.cpp

BSoftFin.cpp

Generating Code...

```

```

Linking...

Creating library Debug/COM_VC_BankSoft.lib and object Debug/COM_VC_BankSoft.exp

Registering ActiveX Control...

RegSvr32: DllRegisterServer in .\Debug\COM_VC_BankSoft.dll succeeded.
COM_VC_BankSoft.dll - 0 error(s), 0 warning(s)

```

Perceba que o Visual C++ compila os arquivos no projeto, vincula-os à biblioteca de vínculo dinâmico (DLL) chamada COM_VC_BankSoft.DLL e registra a classe COM (ActiveX) COM_VC_BankSoft.BSoftFin no registro local.

Depois de registrado, o componente poderá ser acessado pelo Serviço de Integração em execução nesse host.

Etapa 6. Registrar um Procedimento COM com o Repositório

1. Abra o Transformation Developer.
2. Clique em Transformação > Importar Procedimento Externo.
A caixa de diálogo Importar Método COM Externo aparecerá.
3. Clique no botão Procurar.
4. Selecione o DLL COM que você criou e clique em OK.
No exemplo do Banksoft, selecione *COM_VC_Banksoft.DLL*.
5. Na exibição da árvore Selecionar Método, expanda o nó da classe (neste exemplo, BSoftFin).
6. Expanda Métodos.
7. Selecione o método desejado (neste exemplo, FV) e pressione OK.
O Designer criará uma transformação Procedimento Externo.
8. Abra a transformação Procedimento Externo e selecione a guia Propriedades.
Insira os campos Módulo/Identificador Programático e Nome do Procedimento.
9. Clique na guia Portas.
10. Insira os nomes das portas.
11. Clique em OK.

Etapa 7. Criar uma Origem e um Destino para um Mapeamento

Use as instruções SQL a seguir para criar uma tabela de origem e preenchê-la com dados de amostra:

```

create table FVInputs(
    Rate float,
    nPeriods int,
    Payment float,
    PresentValue float,
    PaymentType int
)

insert into FVInputs values (.005,10,-200.00,-500.00,1)
insert into FVInputs values (.01,12,-1000.00,0.00,0)
insert into FVInputs values (.11/12,35,-2000.00,0.00,1)
insert into FVInputs values (.005,12,-100.00,-1000.00,1)

```

Use a instrução SQL a seguir para criar uma tabela de destino:

```

create table FVOutputs(
    FVin_ext_proc float,
)

```

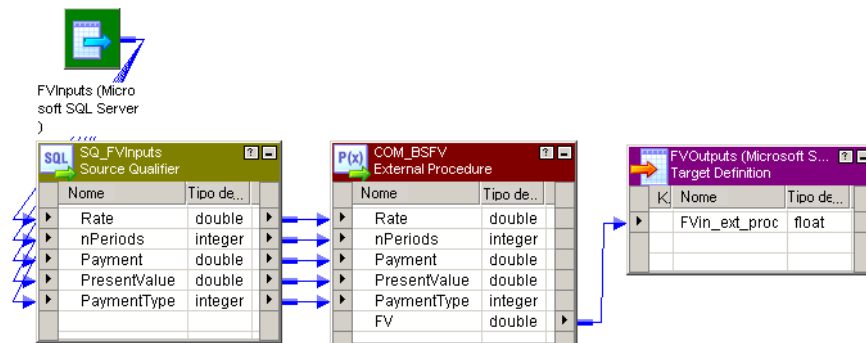
Utilize o Source Analyzer e o Target Designer para importar FVInputs e FVOutputs para a mesma pasta em que você criou a transformação COM_BSFV.

Etapa 8. Criar um Mapeamento Para Testar a Transformação de Procedimento Externo

Agora crie um mapeamento para testar a transformação de Procedimento Externo:

1. No Mapping Designer, crie um nome de mapeamento denominado Test_BSFV.
2. Arraste a tabela de origem FVInputs até o mapeamento.
3. Arraste a tabela de origem FVOutputs até o mapeamento.
4. Arraste a transformação COM_BSFV até o mapeamento.
5. Conecte as portas da transformação Qualificador de origem às portas da transformação Procedimento Externo, conforme apropriado.
6. Conecte a porta VF na transformação Procedimento Externo à coluna de destino FVin_ext_proc.
7. Valide e salve o mapeamento.

A figura a seguir mostra o mapeamento completo:



Etapa 9. Iniciar o Serviço de Integração

Inicie o Serviço de Integração. O serviço deve ser iniciado no mesmo host em que o componente COM foi registrado.

Etapa 10. Executar um Fluxo de Trabalho para Testar o Mapeamento

Quando o Serviço de Integração executa uma sessão em um fluxo de trabalho, realiza uma das seguintes funções:

- Usa o recurso de tempo de execução COM para carregar o DLL e criar uma instância da classe.
- Usa a interface COM do IDispatch para chamar o procedimento externo definido uma vez para cada linha que passa pelo mapeamento.

Nota: É possível definir várias classes, cada uma com vários métodos, dentro de um único projeto. Cada um desses métodos pode ser invocado como procedimento externo.

Para executar um fluxo de trabalho para testar o mapeamento:

1. No Workflow Manager, crie a sessão s_Test_BSFV do mapeamento Test_BSFV.
2. Crie um fluxo de trabalho com a sessão s_Test_BSFV.

3. Executar o fluxo de trabalho. O serviço de Integração procura a entrada para a classe COM_VC_BankSoft.BSoftFin no registro. Essa entrada tem informações que permitem ao Serviço de Integração determinar a localização do DLL que contém a classe. O Serviço de Integração carrega o DLL, cria uma instância da classe e invoca a função VF para cada linha na tabela de origem.

Quando o fluxo de trabalho terminar, a tabela FVOutputs deverá conter estes resultados:

FVIn_ext_proc

2581.403374

12682.503013

82846.246372

2301.401830

Desenvolvendo procedimentos COM com o Visual Basic

O Microsoft Visual Basic oferece um ambiente de desenvolvimento diferente para criar procedimentos COM. Embora a linguagem do Basic tenha sintaxe e convenções diferentes, o procedimento de desenvolvimento tem as mesmas estruturas de tópicos gerais que os procedimentos COM no Visual C++.

TÓPICOS RELACIONADOS:

- [“Distribuindo Procedimentos Externos” na página 186](#)
- [“Classes Wrapper para Funções do VB ou Bibliotecas C/C++ Pré-existentes” na página 189](#)

Etapa 1. Criar um Projeto do Visual Basic com uma Classe Única

1. Inicie o Visual Basic e clique em Arquivo > Novo Projeto.
2. Na caixa de diálogo exibida, selecione ActiveX DLL como tipo do projeto e clique em OK.
O Visual Basic criará um novo projeto chamado *Projeto1*.
Se a janela Projeto não for exibida, digite Ctrl+R ou clique em Exibir > Explorador de Projeto.
Se a janela Propriedades não for exibida, pressione F4 ou clique em Exibir > Propriedades.
3. Na janela Explorador de Projeto do novo projeto, clique com o botão direito no projeto e escolha Propriedades do Projeto1 no menu que for exibido.
4. Insira o nome do novo projeto.
Na janela Projeto, selecione *Projeto1* e altere o nome para *COM_VB_BankSoft* na janela Propriedades.

Etapa 2. Alterar os Nomes do Projeto e da Classe

1. No Explorador de Projeto, selecione o item “Projeto – Projeto1”, que deve ser o item raiz no controle da árvore. As propriedades de projeto aparecem na janela Propriedades.
2. Selecione a guia Ordem Alfabética na janela Propriedades e altere a propriedade Nome para COM_VB_BankSoft. Isso renomeará o item raiz no Explorador de Projeto para COM_VB_BankSoft (COM_VB_BankSoft).
3. Expanda o item COM_VB_BankSoft (COM_VB_BankSoft) no Explorador de Projeto.
4. Expanda o item Módulos de Classe.
5. Selecione o item Classe1 (Classe1). As propriedades da classe aparecem na janela Propriedades.

6. Selecione a guia Ordem Alfabética na Janela Propriedades e altere a propriedade Nome para BSoftFin.

Ao alterar o nome do projeto e da classe, você especifica que o identificador programático da classe criada é "COM_VB_BankSoft.BSoftFin". Use esse identificador programático para fazer referência a essa classe no Designer.

Etapa 3. Adicionar um Método à Classe

Posicione o ponteiro dentro da janela do código e insira o texto a seguir:

```
Public Function FV( _  
    Rate As Double, _  
    nPeriods As Long, _  
    Payment As Double, _  
    PresentValue As Double, _  
    PaymentType As Long _  
) As Double  
    Dim v As Double  
  
    v = (1 + Rate) ^ nPeriods  
  
    FV = -( _  
        (PresentValue * v) + _  
        (Payment * (1 + (Rate * PaymentType))) * ((v - 1) / Rate) _  
    )  
End Function
```

Esta função FV do Visual Basic realiza a mesma operação que a função FV do C++ em ["Desenvolvendo procedimentos COM com o Visual Basic" na página 176](#).

Etapa 4. Construir o Projeto

Para construir o projeto:

1. No menu Arquivo, selecione Criar COM_VB_BankSoft.DLL. Uma caixa de diálogo solicita que você insira a localização do arquivo.
2. Insira a localização do arquivo e clique em OK.

O Visual Basic compila o código-fonte e cria o arquivo COM_VB_BankSoft.DLL no local especificado. Ele também registra a classe COM_VB_BankSoft.BSoftFin no registro local.

Depois de registrado, o componente poderá ser acessado pelo Serviço de Integração em execução nesse host.

Desenvolvendo Procedimentos Externos da Informatica

Você pode criar procedimentos externos que executam em máquinas do Serviço de Integração de 32 ou 64 bits. Conclua as seguintes etapas para criar um procedimento externo no estilo Informatica:

1. No Transformation Developer, crie uma transformação Procedimento Externo.
A transformação Procedimento Externo define a assinatura do procedimento. Os nomes das portas, tipos de dados e tipo de porta (entrada ou saída) devem corresponder à assinatura do procedimento externo.
2. Gere o código do modelo para o procedimento externo.
Quando você executa este comando, o Designer usa as informações da transformação Procedimento Externo para criar vários arquivos de código de origem C++ e um makefile. Um destes arquivos de código de origem contém um "stub" para a função cuja assinatura você definiu na transformação.
3. Modifique o código para adicionar a lógica de procedimento. Preencha o stub com uma implementação e use um compilador do C++ para compilar e vincular os arquivos de código de origem em uma biblioteca de link dinâmico ou compartilhada.
Quando o Serviço de Integração encontra uma transformação Procedimento Externo ligada a um procedimento do Informatica, ele carrega a DLL ou biblioteca compartilhada e chama o procedimento externo que você definiu.
4. Crie a biblioteca e copie para a máquina do Serviço de Integração.
5. Crie um mapeamento com a transformação Procedimento Externo.
6. Execute a sessão em um fluxo de trabalho.

O exemplo do BankSoft ilustra como implementar este recurso.

Etapa 1. Criar a Transformação de Procedimento Externo

1. Abra o Transformation Developer e crie uma transformação Procedimento Externo.
2. Abra a transformação e insira um nome para ela.
No exemplo BankSoft, insira `EP_extINF_BSFV`.
3. Na guia Portas, crie uma porta para cada argumento passado para o procedimento que você planeja definir.

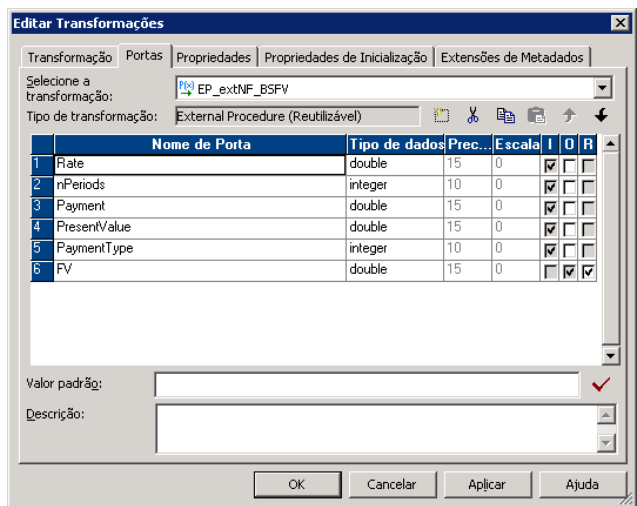
Certifique-se de utilizar os tipos de dados corretos.

A seguinte tabela descreve as portas:

Nome da Porta	Tipo de dados	Precisão	Escala	Entrada/Saída	Reutilizável
Taxa	duplo	15	0	Entrada	Não
nPeriods	integer	10	0	Entrada	Não
Pagamento	duplo	15	0	Entrada	Não
ValorAtual	duplo	15	0	Entrada	Não

Nome da Porta	Tipo de dados	Precisão	Escala	Entrada/Saída	Reutilizável
PaymentType	integer	10	0	Entrada	Não
FV	duplo	15	0	Saída	Sim

O seguinte exemplo do BankSoft mostra um exemplo de uma transformação Procedimento Externo:

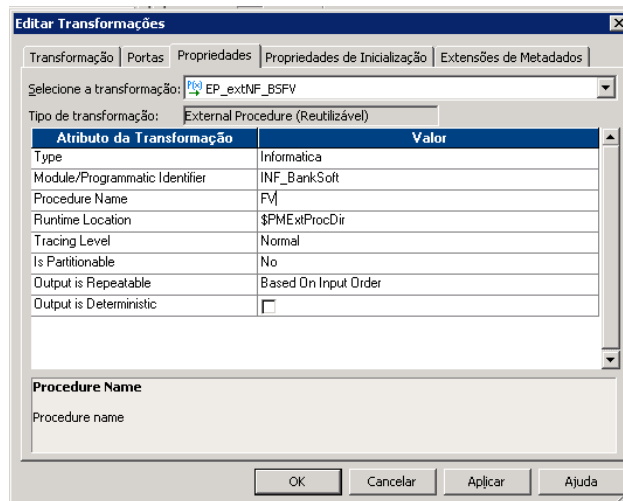


A última porta, FV, captura o valor de retorno do procedimento.

- Selecione a guia Propriedades e configure o procedimento como um procedimento Informatica. No exemplo do BankSoft, configure as seguintes propriedades:

Atributo de Transformação	Valor
Tipo	Informatica
Identificador do Módulo/Programático	INF_BankSoft
Nome do procedimento	FV
Localização de Tempo de Execução	\$PMEExtProcDir
Nível de Rastreamento	Normal
É Particionável	Não
Saída é Repetível	Baseada na Ordem de Entrada.
Saída é Determinística	Não

O seguinte exemplo do BankSoft mostra um exemplo de um procedimento Informatica:



Nota: no Identificador do Módulo/Programático:

A tabela a seguir explica como o nome do módulo determina o nome da DLL ou do objeto compartilhado nas várias plataformas:

Sistema operacional	Identificador do Módulo	Nome do Arquivo de Biblioteca
Windows	INF_BankSoft	INF_BankSoft.DLL
AIX	INF_BankSoft	libINF_BankSoftshr.a
HPUX	INF_BankSoft	libINF_BankSoft.sl
Linux	INF_BankSoft	libINF_BankSoft.so
Solaris	INF_BankSoft	libINF_BankSoft.so.1

5. Clique em OK.

Depois de criar a transformação Procedimento Externo que chama o procedimento, a próxima etapa é gerar os arquivos C++.

Etapa 2. Gerar os Arquivos C++

Depois de criar uma transformação Procedimento Externo, você pode gerar o código. O Designer gera nomes de arquivo com letras minúsculas porque os arquivos criados nas unidades mapeadas para UNIX estão sempre em minúsculas. As regras a seguir aplicam-se aos arquivos gerados:

- **Nomes de arquivos.** O prefixo 'tx' é usado para arquivos de módulo TX.
- **Nomes de classes de módulos.** O código gerado contém declarações de classe para o módulo que contém os procedimentos TX. O prefixo Tx é usado para classes de módulos TX. Por exemplo, se uma transformação Procedimento Externo tiver um módulo denominado Mymod, o nome da classe será TxMymod.

Para gerar o código de um procedimento externo:

1. Selecione a transformação e clique em Transformação > Gerar Código.

2. Marque a caixa de seleção ao lado do nome do procedimento que você acabou de criar.

No exemplo BankSoft, selecione *INF_BankSoft.FV*.

3. Especifique o diretório em que deseja gerar os arquivos e clique em Gerar.

O Designer cria um subdiretório, *INF_BankSoft*, no diretório especificado.

Cada transformação Procedimento Externo criada no Designer deve especificar um módulo e um nome de procedimento. O Designer gera código em um único diretório para todas as transformações que compartilham um nome de módulo comum. A criação do código em um diretório criará uma única biblioteca compartilhada.

O Designer gera os arquivos a seguir:

- **tx<Nomedomódulo>.h**. Define a classe de módulo do procedimento externo. Essa classe é derivada de uma classe base *TINFExternalModule60*. Não são definidos membros de dados para essa classe no código gerado. No entanto, você pode adicionar novos membros de dados e métodos aqui.
- **tx<Nomedomódulo>.cpp**. Implementa a classe de módulo do procedimento externo. Você pode expandir o método *InitDerived()* para incluir a inicialização de quaisquer novos membros de dados que adicionar. O Serviço de Integração chama o método *InitDerived()* da classe derivada somente quando completa com sucesso o método *Init()* da classe base.

Esse arquivo define as assinaturas de todas as transformações Procedimento Externo no módulo.

Qualquer modificação dessas assinaturas provoca inconsistência com as transformações Procedimento Externo definidas no Designer. Portanto, você não deve alterá-las.

Esse arquivo também inclui uma função C *CreateExternalModuleObject*, que cria um objeto da classe do módulo do procedimento externo usando o construtor definido no arquivo. O Serviço de Integração chama *CreateExternalModuleObject*, em vez de chamar o construtor diretamente.

- **<procedureName>.cpp**. O Designer gera um desses arquivos para cada procedimento externo nesse módulo. Esse arquivo contém o código que implementa a lógica do procedimento, como a limpeza e a filtragem de dados. Para a limpeza de dados, crie um código para ler valores das portas de entrada e gerar valores para portas de saída. Para filtragem, crie um código para suprimir a geração de linhas de saída, retornando *INF_NO_OUTPUT_ROW*.
- **stdafx.h**. Arquivo stub usado para construir sistemas UNIX. Os vários arquivos *.cpp incluem esse arquivo. Nos sistemas Windows, o Visual Studio gera um arquivo *stdafx.h*, que deve ser usado em substituição ao arquivo gerado pelo Designer.
- **version.cpp**. Este é um pequeno arquivo que carrega o número de versão dessa implementação. Nas versões anteriores, a implementação do procedimento externo era tratada de maneira diferente. Esse arquivo permite que o Serviço de Integração determine a versão do módulo do procedimento externo.
- **makefile.aix, makefile.aix64, makefile.hp, makefile.hp64, makefile.hpparisc64, makefile.linux, makefile.sol**. Crie arquivos para plataformas UNIX. Utilize *makefile.aix*, *makefile.hp*, *makefile.linux* e *makefile.sol* para plataformas de 32 bits. Utilize *makefile.aix64* para plataformas AIX de 64 bits e *makefile.hp64* para plataformas HP-UX (Itanium) de 64 bits.

Exemplo 1

No exemplo do BankSoft, o Designer gera os seguintes arquivos:

- **txinf_banksoft.h**. Contém declarações para a classe de módulo *TxINF_BankSoft* e para o procedimento externo *FV*.
- **txinf_banksoft.cpp**. Contém código para a classe de módulo *TxINF_BankSoft*.
- **fv.cpp**. Contém código para o procedimento *FV*.
- **version.cpp**. Retorna a versão *TX*.
- **stdafx.h**. Necessário para a compilação em UNIX. No Windows, *stdafx.h* é gerado pelo Visual Studio.

- **leiame.txt.** Contém informações gerais de ajuda.

Exemplo 2

Se você criar duas transformações Procedimento Externo com nomes de procedimento 'Myproc1' e 'Myproc2', ambos com o nome de módulo Mymod, o Designer gera os arquivos a seguir:

- **txmymod.h.** Contém declarações para a classe de módulo TxMymod e os procedimentos externos Myproc1 e Myproc2.
- **txmymod.cpp.** Contém código para a classe de módulo TxMymod.
- **myproc1.cpp.** Contém código para o procedimento Myproc1.
- **myproc2.cpp.** Contém código para o procedimento Myproc2.
- **version.cpp.**
- **stdafx.h.**
- **leiame.txt.**

Etapas 3. Preencher o Stub de Método com Implementação

A etapa final é codificar o procedimento.

1. Abra o arquivo stub <Nome_Procedimento>.cpp gerado para o procedimento.
No exemplo do BankSoft, abra o fv.cpp para codificar o procedimento TxINF_BankSoft::FV.
2. Insira o código C++ para o procedimento.
O código a seguir implementa o procedimento FV:

```
INF_RESULT TxINF_BankSoft::FV()
{
    // Input port values are mapped to the m_pInParamVector array in
    // the InitParams method. Use m_pInParamVector[i].IsValid() to check
    // if they are valid. Use m_pInParamVector[i].GetLong or GetDouble,
    // etc. to get their value. Generate output data into m_pOutParamVector.
    // TODO: Fill in implementation of the FV method here.
    ostream ss;
    char* s;
    INF_BOOLEAN bVal;
    double v;
    TINFParm* Rate = &m_pInParamVector[0];
    TINFParm* nPeriods = &m_pInParamVector[1];
    TINFParm* Payment = &m_pInParamVector[2];
    TINFParm* PresentValue = &m_pInParamVector[3];
    TINFParm* PaymentType = &m_pInParamVector[4];
    TINFParm* FV = &m_pOutParamVector[0];
    bVal =
        INF_BOOLEAN(
            Rate->IsValid() &&
            nPeriods->IsValid() &&
            Payment->IsValid() &&
            PresentValue->IsValid() &&
            PaymentType->IsValid()
        );
    if (bVal == INF_FALSE)
    {
        FV->SetIndicator(INF_SQL_DATA_NULL);
        return INF_SUCCESS;
    }
    v = pow((1 + Rate->GetDouble()), (double)nPeriods->GetLong());
```

```

FV->SetDouble(
    - (
        (PresentValue->GetDouble() * v) +
        (Payment->GetDouble() *
         (1 + (Rate->GetDouble() * PaymentType->GetLong())) *
         ((v - 1) / Rate->GetDouble()))
    );
ss << "The calculated future value is: " << FV->GetDouble() <<ends;
s = ss.str();
(*m_pfnMessageCallback)(E_MSG_TYPE_LOG, 0, s);
(*m_pfnMessageCallback)(E_MSG_TYPE_ERR, 0, s);
delete [] s;
return INF_SUCCESS;
}

```

O Designer gera o perfil de função, incluindo argumentos e valor de retorno. Você deve inserir o código real na função, como indicado nos comentários. Como você referenciou a função POW e definiu uma variável `ostrstream`, deve incluir também as instruções do pré-processador:

No Windows:

```

#include <math.h>
#include <strstream> using namespace std;

```

No UNIX, as instruções incluídas devem ser as seguintes:

```

#include <math.h>
#include <strstream.h>

```

3. Salve o arquivo modificado.

Etapa 4. Construindo o Módulo

No Windows, utilize Visual C++ para compilar a DLL.

Criando o módulo no Windows

Para criar um DLL no Windows:

1. Inicie o Visual C++.
2. Clique em Arquivo > Novo.
3. Na caixa de diálogo Avançar, clique na guia Propriedades e selecione a opção MFC AppWizard (DLL).
4. Insira sua localização.

No exemplo da BankSoft, você insere `c:\pmclient\tx\INF_BankSoft`, supondo-se que tenha gerado os arquivos em `c:\pmclient\tx`.

5. Digite o nome do projeto.

Deve ser igual ao nome do módulo inserido para a transformação Procedimento Externo. No exemplo do BankSoft, é `INF_BankSoft`.

6. Clique em OK.

Agora o Visual C++ o orienta através de um assistente que define todos os componentes do projeto.

7. No assistente, clique em DLL da Extensão MFC (usando o MFC DLL compartilhado).
8. Clique em Concluir.

O assistente gera vários arquivos.

9. Clique em Projeto > Adicionar a Projeto > Arquivos.

10. Navegue até um nível do diretório. Esse diretório contém os arquivos de procedimento externo que você criou. Selecione todos os arquivos `.cpp`.

No exemplo do BankSoft, adicione os seguintes arquivos:

- fv.cpp
- txinf_banksoft.cpp
- version.cpp

11. Clique em Projeto > Configurações.
12. Clique na guia C/C++ e selecione Pré-processador no campo Categoria.
13. No campo Diretórios de Inclusão Adicionais, insira `..; <pmserver install dir>\extproc\include`.
14. Clique na guia Link e selecione Geral no campo Categoria.
15. Insira `<pmserver install dir>\bin\pmtx.lib` no campo Módulos de Objeto/Biblioteca.
16. Clique em OK.
17. Clique em Compilar > Compilar INF_BankSoft.dll ou pressione F7 para compilar o projeto.
Agora, o compilador cria o DLL e o coloca no diretório de depuração ou de versão no diretório do projeto.

Criando o Módulo no UNIX

Para criar bibliotecas compartilhadas no UNIX:

1. Se você não pode acessar as ferramentas do Cliente do PowerCenter diretamente, copie todos os arquivos necessários para a biblioteca compartilhada na máquina UNIX onde planeja executar a construção.
Por exemplo, no procedimento do BankSoft, use o ftp ou outro mecanismo para copiar tudo do diretório INF_BankSoft para a máquina UNIX.
2. Defina a variável de ambiente INFA_HOME como o diretório de instalação do PowerCenter.
Aviso: Se você especificar um caminho de diretório incorreto para a variável de ambiente INFA_HOME, o Serviço de Integração não poderá ser iniciado.
3. Insira o comando para criar o projeto.

O comando depende da versão do UNIX, conforme resumido abaixo:

Versão do UNIX	Comando
AIX (32 bits)	make -f makefile.aix
AIX (64 bits)	make -f makefile.aix64
Linux	make -f makefile.linux
Solaris	make -f makefile.sol

Etapa 5. Criando um Mapeamento

No Mapping Designer, crie um mapeamento que use esta transformação de Procedimento Externo.

Etapa 6. Executar a Sessão

Quando você executa a sessão, o Serviço de Integração procura a biblioteca (DLL) criada na etapa 4 no diretório especificado como Localização do Tempo de Execução. O valor padrão da propriedade Localização do Tempo de Execução nas propriedades da sessão é \$PMExtProcDir.

Para executar uma sessão:

1. No Workflow Manager, crie um fluxo de trabalho.
2. Crie uma sessão para este mapeamento no fluxo de trabalho.

Sugestão: Você também pode criar uma sessão reutilizável no Desenvolvedor de Tarefas e utilizá-la no fluxo de trabalho.

3. Copie a biblioteca (DLL) para o diretório Local do Tempo de Execução.
4. Execute o fluxo de trabalho que contém a sessão.

Executando uma Sessão com a Versão de Depuração do Módulo no Windows

O Informatica inclui o PowerCenter no Windows com a compilação de versão (pmtx.dll) e a compilação de depuração (pmtxdbg.dll) da biblioteca da transformação Procedimento Externo. Essas bibliotecas são instaladas no diretório bin do servidor.

Se você criar uma versão de compilação do módulo na Etapa 4, execute a sessão em um fluxo de trabalho para usar a compilação de versão (pmtx.dll) da biblioteca da transformação Procedimento Externo. Você não precisa realizar a tarefa a seguir.

Se você criar uma versão de depuração do módulo na Etapa 4, execute o procedimento a seguir para usar a compilação de depuração (pmtxdbg.dll) da biblioteca da transformação Procedimento Externo.

Para executar uma sessão usando uma versão de depuração do módulo:

1. No Workflow Manager, crie um fluxo de trabalho.
2. Crie uma sessão para este mapeamento no fluxo de trabalho.
Você também pode criar uma sessão reutilizável no Desenvolvedor de Tarefas e utilizá-la no fluxo de trabalho.
3. Copie a biblioteca (DLL) para o diretório Local do Tempo de Execução.
4. Para usar uma compilação de depuração da biblioteca da transformação Procedimento Externo:
 - Preserve pmtx.dll, renomeando-o ou movendo-o do diretório bin do servidor.
 - Renomeie pmtxdbg.dll como pmtx.dll.
5. Execute o fluxo de trabalho que contém a sessão.
6. Para reverter a compilação de versão da biblioteca da transformação Procedimento Externo de volta para a biblioteca padrão:
 - Renomeie pmtx.dll de volta para pmtxdbg.dll.
 - Retorne/renomeie o arquivo pmtx.dll original para o diretório bin do servidor.

Nota: Se executar um fluxo de trabalho contendo essa sessão com a versão de depuração do módulo no Windows, você deverá retornar o arquivo pmtx.dll original ao seu nome e local originais para poder executar uma sessão de não depuração.

Distribuindo Procedimentos Externos

Partamos do pressuposto de que você desenvolveu um conjunto de procedimentos externos e deseja torná-los disponíveis em diversos servidores, cada um dos quais executando o Serviço de Integração. Os métodos para realizar isso dependem do tipo de procedimento externo e do sistema operacional no qual você o cria.

Também é possível utilizar esses procedimentos para distribuir procedimentos externos a clientes externos.

Distribuindo Procedimentos COM

O Visual Basic e o Visual C++ registram classes COM no registro local quando você compila o projeto. Assim que o registro for efetuado, essas classes estarão acessíveis ao Serviço de Integração executado na máquina na qual você compilou a DLL. Por exemplo, se você compilar um projeto em HOST1, todas as classes do projeto serão incluídas no registro HOST1 e estarão acessíveis ao Serviço de Integração em execução no HOST1. Entretanto, suponhamos que você também queira que as classes sejam acessíveis ao Serviço de Integração que está sendo executado no HOST2. Para que isso ocorra, as classes devem ser registradas no registro do HOST2.

O Visual Basic fornece um utilitário para criar um programa de instalação capaz de instalar classes COM em uma máquina com Windows e registrar essas classes no registro da máquina. Enquanto nenhum utilitário está disponível no Visual C++, é fácil você mesmo registrar a classe.

Distribuindo um Procedimento Visual Basic COM

Para distribuir um procedimento Visual Basic COM:

1. Depois de criar a DLL, saia do Visual Basic e inicie o assistente Configuração do Aplicativo Visual Basic.
2. Ignore o primeiro painel do assistente.
3. No segundo painel, especifique o local do projeto e selecione a opção *Criar um Programa de Configuração*.
4. No terceiro painel, selecione o método de distribuição que você planeja usar.
5. No próximo painel, especifique o diretório em que deseja escrever os arquivos de configuração.

Para componentes simples do ActiveX, você pode continuar até o painel final do assistente. Caso contrário, você pode precisar adicionar mais informações, dependendo do tipo de arquivo e do método de distribuição.

6. Clique em Concluir no painel final.

Em seguida, o Visual Basic cria o programa de configuração para a DLL. Execute este programa de configuração em qualquer máquina do Windows em que o Serviço de Integração está executando.

Distribuindo um Procedimento Visual Basic COM Manualmente

Para distribuir manualmente um procedimento COM Visual C++/Visual Basic:

1. Copie a DLL para o diretório, na nova máquina Windows, onde deseja salvá-la.
2. Faça o logon nesta máquina Windows e abra um prompt do DOS.
3. Navegue até o diretório que contém a DLL e execute o comando a seguir:

```
REGSVR32 project_name.DLL
```

project_name é o nome da DLL que você criou. No exemplo do BankSoft, o nome do projeto é *COM_VC_BankSoft.DLL*. ou *COM_VB_BankSoft.DLL*.

Este programa de linha de comando registra a DLL e todas as classes COM contidas nela.

Distribuindo Módulos Informatica

Você pode distribuir procedimentos externos entre repositórios.

Para distribuir procedimentos externos entre repositórios:

1. Mova a DLL ou objeto compartilhado que contém o procedimento externo para um diretório em uma máquina que o Serviço de Integração possa acessar.
2. Copie a transformação Procedimento Externo do repositório original para o repositório de destino, usando a ferramenta do cliente Designer.

Você também pode exportar a transformação Procedimento Externo para um arquivo XML e importá-la no repositório de destino.

Observações de Desenvolvimento

Esta sessão apresenta mais algumas orientações e informações acerca do desenvolvimento de procedimentos externos COM e Informatica.

Tipos de Dados COM

Quando usar o Visual C++ ou Visual Basic para desenvolver procedimentos COM, você precisa usar tipos de dados COM que correspondam aos tipos de dados internos que o Serviço de Integração usa ao ler e transformar os dados. Estas correspondências do tipo de dados são importantes quando o Serviço de Integração tenta mapear tipos de dados entre as portas de uma transformação Procedimento Externo e argumentos (ou valores de retorno) do procedimento chamado pela transformação.

A tabela a seguir compara os tipos de dados do Visual C++ e da transformação:

Tipo de dados Visual C++ COM	Tipo de dados da transformação
VT_I4	Número inteiro
VT_UI4	Número inteiro
VT_R8	Duplo
VT_BSTR	String
VT_DECIMAL	Decimal
VT_DATE	Data/Hora

A seguinte tabela compara os tipos de dados do Visual Basic e da transformação:

Tipo de dados Visual Basic COM	Tipo de dados da transformação
Longa	Número inteiro
Duplo	Duplo

Tipo de dados Visual Basic COM	Tipo de dados da transformação
String	String
Decimal	Decimal
Data	Data/Hora

Se você não corresponder corretamente os tipos de dados, o Serviço de Integração pode tentar uma conversão. Por exemplo, se você atribuir o tipo de dados Inteiro a uma porta, mas o tipo de dados para o argumento correspondente for BSTR, o Serviço de Integração tenta converter o valor Inteiro em um BSTR.

Procedimentos no Nível de Linha

Todas as transformações Procedimento Externo chamam procedimentos usando valores de uma linha que passa pela transformação. Não é possível usar valores de várias linhas em uma única chamada de procedimento. Por exemplo, não é possível codificar o equivalente às funções de agregação SUM ou AVG em uma chamada de procedimento. Nesse sentido, nenhum procedimento externo deve ter *monitoração de estado*.

Valores de Retorno de Procedimentos

Quando você chama um procedimento, o Serviço de Integração captura um valor de retorno adicional, independentemente do valor de retorno codificado no procedimento. Esse valor adicional indica se o Serviço de Integração chamou o procedimento com sucesso.

Para procedimentos COM, esse valor de retorno utiliza o tipo HRESULT.

Os procedimentos Informatica utilizam o tipo INF_RESULT. Se o valor retornado for S_OK/INF_SUCCESS, isso indica que o Serviço de Integração chamou o procedimento com sucesso. Você deve retornar o valor apropriado para indicar o sucesso ou a falha do procedimento externo. Os procedimentos Informatica retornam quatro valores:

- **INF_SUCCESS.** O procedimento externo processou a linha com sucesso. O Serviço de Integração passa a linha para a próxima transformação no mapeamento.
- **INF_NO_OUTPUT_ROW.** O Serviço de Integração não grava a linha atual devido à lógica do procedimento externo. Isso não é um erro. Quando você usa INF_NO_OUTPUT_ROW para filtrar linhas, a transformação Procedimento Externo se comporta da mesma maneira que a transformação Filtro.

Nota: Quando usar INF_NO_OUTPUT_ROW no procedimento externo, certifique-se de vincular a transformação Procedimento Externo a outra transformação que receba linhas somente da transformação Procedimento Externo.

- **INF_ROW_ERROR.** Equivalente a um erro de transformação. O Serviço de Integração descarta a linha atual, mas pode processar a linha seguinte, a menos que você configure a sessão para parar em *n* erros.
- **INF_FATAL_ERROR.** Equivalente a uma chamada da função ABORT(). O Serviço de Integração anula a sessão e não processa mais nenhuma outra linha.

Exceções em Chamadas de Procedimentos

O Serviço de Integração captura a maioria das exceções que ocorrem quando ele chama um procedimento COM ou do Informatica através de uma transformação Procedimento Externo. Por exemplo, se a chamada de procedimento cria um erro de divisão por zero, o Serviço de Integração captura a exceção.

Em alguns casos, o Serviço de Integração não pode capturar erros gerados por chamadas de procedimento. Uma vez que o Serviço de Integração suporta apenas servidores COM dentro do processo, e uma vez que todos os procedimentos do Informatica são armazenados em bibliotecas compartilhadas e DLLs, o código que executa os procedimentos externos que existem no mesmo espaço de endereço da memória que o Serviço de Integração. Portanto, é possível que o código de procedimento externo substitua a memória do Serviço de Integração, causando sua interrupção. Se os procedimentos COM ou do Informatica causarem essa interrupção, revise o código da origem quanto a problemas de acesso à memória.

Gerenciamento de Memória para Procedimentos

Uma vez que todos os tipos de dados usados nos procedimentos da Informatica têm comprimento fixo, não existem problemas de gerenciamento de memória para os procedimentos externos da Informatica. Para os procedimentos COM, você precisará alocar memória apenas se um parâmetro [out] de um procedimento utilizar o tipo de dados BSTR. Neste caso, você precisa alocar memória a cada chamada para este procedimento. Durante uma sessão, o Serviço de Integração libera a memória depois de chamar a função.

Classes Wrapper para Funções do VB ou Bibliotecas C/C++ Pré-existent

Suponha que o BankSoft tenha uma biblioteca de funções C ou C++ e queira conectar essas funções ao Serviço de Integração. Em especial, a biblioteca contém a própria implementação do BankSoft da função FV, denominada PreExistingFV. O método geral para fazer isso é o mesmo usado para procedimentos externos COM e Informatica. Uma solução semelhante está disponível no Visual Basic. Basta você fazer chamadas para funções do Visual Basic pré-existent ou para métodos em objetos que estejam acessíveis a ele.

Gerando Mensagens de Erro e Rastreamento

A implementação do procedimento externo Informatica TxINF_BankSoft::FV no [“Etapa 4. Construindo o Módulo” na página 183](#) contém as seguintes linhas de código:

```
ostream ss;

char* s;

...

ss << "The calculated future value is: " << FV->GetDouble() << ends;

s = ss.str();

(*m_pfnMessageCallback)(E_MSG_TYPE_LOG, 0, s);

(*m_pfnMessageCallback)(E_MSG_TYPE_ERR, 0, s);

delete [] s;
```

Quando o Serviço de Integração cria um objeto do tipo Tx<MODNAME>, ele transfere ao seu construtor um ponteiro para uma função de retorno de chamada, que pode ser usada para escrever mensagens de erro ou depuração no log da sessão. (O código para construtor de Tx<MODNAME> está no arquivo Tx<MODNAME>.cpp.) Este ponteiro é armazenado no membro Tx<MODNAME> variável m_pfnMessageCallback. O tipo deste ponteiro é definido em um typedef no arquivo \$PMExtProcDir/include/infemmsg.h:

```
typedef void (*PFN_MESSAGE_CALLBACK) (

    enum E_MSG_TYPE eMsgType,

    unsigned long Code,
```

```
char* Message
);
```

Também definida nesse arquivo está a enumeração E_MSG_TYPE:

```
enum E_MSG_TYPE {
    E_MSG_TYPE_LOG = 0,
    E_MSG_TYPE_WARNING,
    E_MSG_TYPE_ERR
};
```

Se você especificar o eMsgType da função de retorno de chamada como E_MSG_TYPE_LOG, tal função escreverá uma mensagem *log* no log da sessão. Se você especificar E_MSG_TYPE_ERR, a função de retorno de chamada escreve uma mensagem de *erro* no log da sessão. Se você especificar E_MSG_TYPE_WARNING, a função de retorno de chamada gravará uma mensagem *aviso* no log da sessão. Use estas mensagens para permitir uma capacidade de depuração simples nos procedimentos externos Informatica.

Para depurar procedimentos externos COM, você pode usar as instalações de saída disponíveis dentro de uma classe do Visual Basic ou C++. Por exemplo, no Visual Basic use um MsgBox para imprimir o resultado de um cálculo para cada linha. Obviamente, você deseja fazer isto somente em amostras pequenas de dados durante a depuração, e não deixe de remover o MsgBox antes de fazer uma execução de produção.

Nota: Antes de tentar usar qualquer instalação de saída dentro de uma classe Visual Basic ou C++, adicione o seguinte valor ao registro:

1. Adicione a seguinte entrada ao registro do Windows:

```
\HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\PowerMart\Parameters\MiscInfo
\RunInDebugMode=Yes
```

Esta opção inicia o Serviço de Integração como um aplicativo regular, não um serviço. Você pode depurar o Serviço de Integração sem alterar os privilégios de depuração deste serviço enquanto ele está executando.

2. Inicie o Serviço de Integração a partir da linha de comando, usando o comando PMSERVER.EXE.

O Serviço de Integração está agora executando no modo de depuração.

Quando terminar a depuração, certifique-se de remover esta entrada do registro ou configurar o RunInDebugMode como Não. Do contrário, quando você tentar iniciar o PowerCenter como um serviço, ele não iniciará.

1. Interrompa o Serviço de Integração e altere a entrada de registro que você adicionou anteriormente à seguinte configuração:

```
\HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\PowerMart\Parameters\MiscInfo
\RunInDebugMode=No
```

2. Reinicie o Serviço de Integração como um serviço do Windows.

Os Indicadores e a Classe TINFParm

O método <PROCNAME> acessa parâmetros de entrada e saída usando duas matrizes de parâmetro e cada um dos elementos da matriz é do tipo de dados TINFParm. O tipo de dados TINFParm é uma classe C++ que serve como estrutura de dados "variante" que pode reter qualquer tipo de dados internos da Informatica. Os dados reais em um parâmetro do tipo TINFParm* são acessados através de funções de membro do formulário Obter<Tipo> e Definir<Tipo>, em que <Tipo> é um tipo de dados internos da Informatica. TINFParm também tem métodos para obter e definir o indicador de cada parâmetro.

Você é responsável por verificar a entrada do procedimento externo nesses indicadores e por configurá-los na saída. Na entrada, os indicadores de todos os parâmetros de saída são definidos explicitamente como

INF_SQL_DATA_NULL, portanto, se você não redefini-los antes de retornar do procedimento externo, obterá apenas NULL para todos os parâmetros de saída. A classe TINFParm também suporta funções para obter os metadados de determinado parâmetro. Para obter uma descrição completa de todas as funções de membro da classe TINFParm, consulte o arquivo infemdef.h incluído no diretório tx/include.

Uma das maiores vantagens dos procedimentos externos da Informatica sobre procedimentos externos COM é que aqueles dão suporte direto à manipulação de indicadores. Ou seja, você pode verificar um parâmetro de entrada para ver se é NULL e pode definir um parâmetro de saída como NULL. A COM não fornece suporte a indicadores. Assim, se uma linha inserida em um procedimento externo COM tiver NULLs, a linha não poderá ser processada. Use o recurso de valor padrão no Designer para superar essa deficiência. Entretanto, não é possível passar NULLs de uma função COM.

Transformações Procedimento Externo Não Conectadas

Quando adiciona uma instância de uma transformação Procedimento Externo a um mapeamento, você pode optar por conectá-la como parte do pipeline ou deixá-la desconectada. As transformações Procedimento Externo não conectadas chamam o procedimento COM ou Informativa toda vez que uma linha passa pela transformação.

Para obter valores de retorno a partir de uma transformação Procedimento Externo não conectada, chame-a em uma expressão usando a sintaxe a seguir:

```
:EXT.transformation_name(arguments)
```

Quando uma linha passa pela transformação que contém a expressão, o Serviço de Integração chama o procedimento associado à transformação Procedimento Externo. A expressão captura o valor de retorno do procedimento através da porta de retorno da transformação Procedimento Externo, que deve estar com a opção Resultado (R) selecionada.

Inicializando os Módulos COM e Informatica

Alguns procedimentos externos devem ser configurados no momento da inicialização. Esta inicialização assume uma de duas formas, dependendo do tipo de procedimento externo:

- **Inicialização de procedimentos externos no estilo Informatica.** A classe Tx<MODNAME>, que contém o procedimento externo, também contém a função de inicialização Tx<MODNAME>::InitDerived. A assinatura dessa função de inicialização é bem conhecida do Serviço de Integração e apresenta três parâmetros:
 - **nInitProps.** Este parâmetro informa à função de inicialização sobre quantas propriedades de inicialização estão sendo atribuídas a ela.
 - **Propriedades.** Este parâmetro é uma matriz de strings nInitProp que representa os nomes das propriedades de inicialização.
 - **Valores.** Este parâmetro é uma matriz de strings nInitProp que representa os valores das propriedades de inicialização.

Primeiro, o Serviço de Integração chama a função Init() na classe base. Quando a função Init() for concluída com êxito, a classe base chama a função Tx<MODNAME>::InitDerived().

O Serviço de Integração cria o objeto Tx<MODNAME> e, em seguida, chama a função de inicialização. O desenvolvedor externo de procedimentos tem a responsabilidade de fornecer essa parte da função Tx<MODNAME>::InitDerived() que interprete as propriedades de inicialização e as utilize para inicializar o procedimento externo. Assim que o objeto tiver sido criado e inicializado, o Serviço de Integração pode chamar o procedimento externo para o objeto de cada linha.

- **Inicialização de procedimentos externos no estilo COM.** O objeto que contém o procedimento externo (ou objeto PE) não apresenta uma função de inicialização. Ao invés disso, um outro objeto (o objeto FC) serve de classe de fábrica ao objeto PE. O objeto FC apresenta um método que pode criar um objeto PE.

A assinatura do método do objeto FC é determinada a partir de sua biblioteca de tipos. O Serviço de Integração cria o objeto FC e, em seguida, chama o método para que auxilie a criar o objeto PE, que passa por este método, independente dos parâmetros necessários. Para isso, é necessário que a assinatura do método consista em um conjunto de parâmetros de entrada, cujos tipos podem ser determinados a partir da biblioteca de tipos, seguidos de um único parâmetro de saída, que é um `IUnknown**` ou um `IDispatch**`, ou ainda um `VARIANT*`, que aponta a um `IUnknown*` ou a um `IDispatch*`.

Os parâmetros de entrada detêm os valores necessários para inicializar o objeto EP e o parâmetro de saída recebe o objeto inicializado. O parâmetro de saída pode ter ou o atributo `[out]` ou `[out, retval]`. Ou seja, o objeto inicializado pode ser retornado como um parâmetro de saída ou como o valor retornado do método. Os tipos de dados suportados para o parâmetro de entrada são:

- Tipo COM VC
- VT_UI1
- VT_BOOL
- VT_I2
- VT_UI2
- VT_I4
- VT_UI4
- VT_R4
- VT_R8
- VT_BSTR
- VT_CY
- VT_DATE

Definindo Propriedades de Inicialização no Designer

Insira propriedades de inicialização do procedimento externo na guia Propriedades de Inicialização da caixa de diálogo Editar Transformações. A guia exibe diferentes campos, dependendo de o procedimento externo ter o estilo COM ou Informatica.

As transformações Procedimento Externo com estilo COM contêm os seguintes campos na guia Propriedades de Inicialização:

- **Identificador Programático para Fábrica de Classes.** Insira o identificador programático da fábrica de classes.
- **Construtor.** Especifique o método da fábrica de classes que cria o objeto EP.

Você pode inserir um número ilimitado de propriedades de inicialização, a fim de passar para o método construtor, tanto para transformações Procedimento Externo com estilo COM quanto Informatica.

Para adicionar uma nova propriedade de inicialização, clique no botão Adicionar. Insira o nome do parâmetro na coluna Propriedade e o valor do parâmetro na coluna Valor. Por exemplo, você pode inserir os seguintes parâmetros:

Parâmetro	Valor
Param1	abc
Param2	100
Param3	3.17

Nota: Crie uma relação um para um entre as propriedades de inicialização que você definir no Designer e os parâmetros de entrada do método construtor da fábrica de classes. Por exemplo, se o construtor tiver n parâmetros e o último deles for o parâmetro de saída que recebe o objeto inicializado, você deverá definir $n - 1$ propriedades de inicialização no Designer, uma para cada parâmetro de entrada no método construtor.

Você também pode processar variáveis em propriedades de inicialização.

Outros Arquivos Distribuídos e Usados em TX

A seguir estão os arquivos de cabeçalho localizados no caminho `$PMExtProcDir/include` que são necessários para compilar procedimentos externos:

- `infconfig.h`
- `infem60.h`
- `infemdef.h`
- `infemmsg.h`
- `infparam.h`
- `infsigtr.h`

A seguir estão os arquivos de biblioteca localizados no caminho `<PMInstallDir>` que são necessários para vincular procedimentos externos e executar a sessão:

- `libpmtx.a` (AIX)
- `libpmtx.so` (Linux)
- `libpmtx.so` (Solaris)
- `pmtx.dll` e `pmtx.lib` (Windows)

Variáveis do Processo de Serviço nas Propriedades de Inicialização

O PowerCenter dá suporte a variáveis de processo interno na lista de propriedades de inicialização da transformação Procedimento Externo. Se os valores da propriedade contiverem variáveis de processo interno, o Serviço de Integração irá expandi-los antes de passá-los à biblioteca de procedimentos externos. Isso pode ser muito útil para gravar transformações Procedimento Externo portáteis.

A seguinte tabela descreve as propriedades de inicialização e os valores para uma transformação de Procedimento Externo na guia Propriedades de Inicialização das propriedades de transformação:

Tabela 2. Propriedades de Inicialização de Procedimento Externo

Propriedade	Valor	Valor Expandido Passado à Biblioteca de Procedimentos Externos
mytempdir	\$PMTempDir	/tmp
memorysize	5000000	5000000
input_file	\$PMSourceFileDir/file.in	/data/input/file.in
output_file	\$PMTargetFileDir/file.out	/data/output/file.out
extra_var	\$some_other_variable	\$some_other_variable

Quando você executa o fluxo de trabalho, o Serviço de Integração expande a lista de propriedades e a passa à função de inicialização do procedimento externo. Supondo que os valores da variável de processo interno \$PMTempDir seja */tmp*, de \$PMSourceFileDir seja */data/input* e de \$PMTargetFileDir seja */data/output*, a última coluna em [“Variáveis do Processo de Serviço nas Propriedades de Inicialização” na página 193](#) conterá as informações de valor expandido e propriedades. O Serviço de Integração não expande a última propriedade “\$some_other_variable” pois não é uma variável de processo interno.

Interfaces do Procedimento Externo

O Serviço de Integração usa as seguintes funções principais com os Procedimentos Externos:

- Expedição
- Procedimento externo
- Acesso de propriedade
- Acesso de parâmetro
- Acesso da página de código
- Acesso do nome da transformação
- Acesso do procedimento
- Particionamento relacionado
- Nível de rastreamento

Função Distribuir

O Serviço de Integração chama a função distribuir para transferir cada linha de entrada para o módulo de procedimento externo. A função distribuir, por sua vez, chama a função de procedimento externo que você especificar.

Os procedimentos externos acessam as portas na transformação, usando diretamente a variável do membro `m_pInParamVector` para portas de entrada e `m_pOutParamVector` para portas de saída.

Assinatura

A função distribuir tem uma assinatura fixa que inclui um parâmetro de índice.

```
virtual INF_RESULT Dispatch(unsigned long ProcedureIndex) = 0
```

Função de Procedimento Externo

A função de procedimento externo é o principal ponto de entrada no módulo de procedimento externo e é um atributo da transformação de Procedimento Externo. A função de distribuição chama a função de procedimento externo para cada linha de entrada. Para as transformações de Procedimento Externo, use a função de procedimento externo para a entrada e a saída do módulo de procedimento externo. A função pode acessar os valores de porta IN e IN-OUT para cada linha de entrada e pode configurar os valores de porta OUT e IN-OUT. A função de procedimento externo contém toda a lógica de processamento de entrada e saída.

Assinatura

A função do procedimento externo não possui parâmetros. O array do parâmetro de entrada já passou pelo método InitParams() e está armazenado na variável de membro m_pInParamVector. Cada entrada no array corresponde às portas IN e IN-OUT correspondentes da transformação Procedimento Externo, na mesma ordem. O Serviço de Integração preenche este vetor antes de chamar a função de distribuição.

Utilize a variável de membro m_pOutParamVector para passar a linha de saída antes de retornar a função Dispatch().

Para a transformação Procedimento MyExternal, a função de procedimento externo é a seguinte. Nela, os parâmetros de entrada estão na variável de membro m_pInParamVector e os valores de saída estão na variável de membro m_pOutParamVector:

```
INF_RESULT Tx<ModuleName>::MyFunc()
```

Funções de Acesso das Propriedades

As funções de acesso das propriedades fornecem informações sobre as propriedades de inicialização associadas à transformação Procedimento Externo. Quando você edita a transformação Procedimento Externo, os nomes e valores das propriedades de inicialização aparecem na guia Propriedades de Inicialização.

O Informatica fornece funções de acesso de propriedades na classe base e na classe TINFConfigEntriesList. Utilize as funções GetConfigEntryName() e GetConfigEntryValue() na classe TINFConfigEntriesList para acessar respectivamente o nome e o valor das propriedades de inicialização.

Assinatura

A Informatica fornece as seguintes funções na classe base:

```
TINFConfigEntriesList* TINFBaseExternalModule60::accessConfigEntriesList();  
const char* GetConfigEntry(const char* LHS);
```

A Informatica fornece as seguintes funções na classe TINFConfigEntriesList:

```
const char* TINFConfigEntriesList::GetConfigEntryValue(const char* LHS);  
const char* TINFConfigEntriesList::GetConfigEntryValue(int i);  
const char* TINFConfigEntriesList::GetConfigEntryName(int i);  
const char* TINFConfigEntriesList::GetConfigEntry(const char* LHS)
```

Nota: Na classe TINFConfigEntriesList, use as funções de acesso a propriedades GetConfigEntryName() e GetConfigEntryValue() para acessar os nomes e valores da propriedade de inicialização.

Você pode chamar essas funções de um programa TX. O programa TX então converte o valor da string em um número usando `atoi` ou `sscanf`, por exemplo. No exemplo a seguir, "addFactor" é uma Propriedade de Inicialização. `accessConfigEntriesList()` é uma variável de membro da classe base TX e não precisa ser definida.

```
const char* addFactorStr = accessConfigEntriesList()-> GetConfigEntryValue("addFactor");
```

Funções de Acesso a Parâmetros

As funções de acesso de parâmetro são específicas do tipo de dados. Use a função de acesso de parâmetro `GetDataType` para retornar o tipo de dados de um parâmetro. Em seguida, use uma função de acesso de parâmetro que corresponda a este tipo de dados, para retornar informações sobre o parâmetro.

Um parâmetro transferido a um procedimento externo pertence ao tipo de dados `TINFParm*`. O arquivo de cabeçalho `infparam.h` define as funções de acesso relacionadas. O Designer gera um código stub que inclui comentários indicando o tipo de dados do parâmetro. Você também pode determinar o tipo de dados de um parâmetro na transformação Procedimento Externo correspondente no Designer.

Assinatura

Um parâmetro passado a um procedimento externo é um ponteiro para um objeto da classe `TINFParm`. Essa função de assinatura fixa é um método da classe e retorna o tipo de dados do parâmetro como um valor de enum.

Os tipos de dados válidos são:

- `INF_DATATYPE_LONG`
- `INF_DATATYPE_STRING`
- `INF_DATATYPE_DOUBLE`
- `INF_DATATYPE_RAW`
- `INF_DATATYPE_TIME`

A tabela a seguir descreve algumas funções de acesso a parâmetros:

Função de Acesso a Parâmetros	Descrição
<code>INF_DATATYPE GetDataType(void);</code>	Obtém o tipo de dados de um parâmetro. Use o tipo de dados do parâmetro para determinar qual função específica a ele deve usar ao acessar valores de parâmetro.
<code>INF_BOOLEAN IsValid(void);</code>	Verifica se os dados de entrada são válidos. Retorna <code>FALSE</code> se o parâmetro contiver dados truncados para a string ou tipos de dados brutos. Também retorna <code>FALSE</code> quando os dados de entrada excedem a precisão ou são um valor nulo.
<code>INF_BOOLEAN IsNULL(void);</code>	Verifica se os dados de entrada são <code>NULL</code> .
<code>INF_BOOLEAN IsInputMapped (void);</code>	Verifica se a porta de entrada que passa dados a esse parâmetro está conectada a uma transformação.
<code>INF_BOOLEAN IsOutput Mapped (void);</code>	Verifica se a porta de saída que recebe dados desse parâmetro está conectada a uma transformação.
<code>INF_BOOLEAN IsInput(void);</code>	Verifica se o parâmetro corresponde a uma porta de entrada.
<code>INF_BOOLEAN IsOutput(void);</code>	Verifica se o parâmetro corresponde a uma porta de saída.

Função de Acesso a Parâmetros	Descrição
INF_BOOLEAN GetName(void);	Obtém o nome do parâmetro.
SQLIndicator GetIndicator(void);	Obtém o valor do indicador de um parâmetro. As funções IsValid e ISNULL são casos especiais dessa função. Essa função também pode retornar INF_SQL_DATA_TRUNCATED.
void SetIndicator(SQLIndicator Indicator);	Define um indicador de parâmetro de saída, como inválido ou truncado.
long GetLong(void);	Obtém o valor de um parâmetro cujo tipo de dados é Long ou Integer. Só chame essa função se você souber que o tipo de dados do parâmetro é Integer ou Long. Essa função não converte dados de outro tipo de dados em Long.
double GetDouble(void);	Obtém o valor de um parâmetro cujo tipo de dados é Float ou Double. Chame essa função somente se você souber que o tipo de dados do parâmetro é Float ou Double. Essa função não converte outro tipo de dados em Double.
char* GetString(void);	Obtém o valor de um parâmetro como uma string não terminada em nulo. Chame essa função somente se você souber que o tipo de dados do parâmetro é String. Essa função não converte outro tipo de dados em String. O valor no ponteiro mudará quando a linha seguinte de dados for lida. Se você quiser armazenar o valor de uma linha para uso posterior, copie explicitamente a string para o seu próprio buffer.
char* GetRaw(void);	Obtém o valor de um parâmetro como uma matriz de bytes não terminada em nulo. Chame essa função somente se você souber que o tipo de dados do parâmetro é Raw. Essa função não converte outro tipo de dados em Raw.
unsigned long GetActualDataLen(void);	Obtém a atual extensão da matriz retornada por GetRaw.
TINFTime GetTime(void);	Obtém o valor de um parâmetro cujo tipo de dados é Date/Time. Chame essa função somente se você souber que o tipo de dados do parâmetro é Date/Time. Essa função não converte outro tipo de dados em Date/Time.
void SetLong(long lVal);	Define o valor de um parâmetro de saída com tipo de dados Long.
void SetDouble(double dblVal);	Define o valor de um parâmetro de saída cujo tipo de dados é Double.
void SetString(char* sVal);	Define o valor de um parâmetro de saída cujo tipo de dados é String.
void SetRaw(char* rVal, size_t ActualDataLen);	Define uma matriz de bytes não terminada em nulo.
void SetTime(TINFTime timeVal);	Define o valor de um parâmetro de saída cujo tipo de dados é Date/Time.

Só use a função SetInt32 ou GetInt32 quando executar o procedimento externo em um Serviço de Integração de 64 bits. Não use nenhuma destas funções:

- GetLong
- SetLong

- GetpLong
- GetpDouble
- GetpTime

Passe os parâmetros usando duas listas de parâmetros.

A tabela a seguir lista as variáveis de membro da classe base do procedimento externo:

Variável	Descrição
m_nInParamCount	Número de parâmetros de entrada.
m_pInParamVector	Matriz real de parâmetros de entrada.
m_nOutParamCount	Número de parâmetros de saída.
m_pOutParamVector	Matriz real de parâmetros de saída.
Nota: Portas definidas como entrada e saída mostradas nas duas listas de parâmetros.	

Funções de Acesso à Página de Código

A Informatica fornece duas funções de acesso à página de códigos que retornam a página de códigos do Serviço de Integração e duas que retornam a página de códigos dos dados que o procedimento externo processa. Quando o Serviço de Integração é executado em modo Unicode, os dados de string que são transferidos ao programa de procedimento externo podem conter caracteres de vários bytes. A página de códigos determina como o procedimento externo interpreta uma string de vários bytes. Quando o Serviço de Integração é executado em modo Unicode, os dados processados pelo programa de procedimentos externos devem ter compatibilidade bilateral com a página de códigos do Serviço de Integração.

Assinatura

Use as funções a seguir para obter a página de código do Serviço de Integração por meio do programa de procedimento externo. As duas funções retornam informações equivalentes.

```
int GetServerCodePageID() const;
const char* GetServerCodePageName() const;
```

Use as funções a seguir para obter a página de código dos dados que o procedimento externo processa por meio do programa de procedimento externo. As duas funções retornam informações equivalentes.

```
int GetDataCodePageID(); // returns 0 in case of error
const char* GetDataCodePageName() const; // returns NULL in case of error
```

Funções de Acesso ao Nome da Transformação

A Informatica fornece duas funções de acesso ao nome da transformação que retornam o nome da transformação Procedimento Externo. A função GetWidgetName() retorna o nome da transformação e a função GetWidgetInstanceName() retorna o nome da instância da transformação no mapplet ou mapeamento.

Assinatura

O `char*` retornado pelas funções de acesso do nome da transformação é uma string MBCS na página de código do Serviço de Integração. Não fica na página de código dos dados.

```
const char* GetWidgetInstanceName() const;
const char* GetWidgetName() const;
```

Funções de Acesso a Procedimentos

A Informatica fornece duas funções de acesso a procedimentos que fornecem informações sobre procedimentos externos associados à transformação Procedimento Externo. A função `GetProcedureName()` retorna o nome do procedimento externo especificado no campo Nome do Procedimento da transformação Procedimento Externo. A função `GetProcedureIndex()` retorna o índice do procedimento externo.

Assinatura

Use a função a seguir para obter o nome do procedimento externo associado à transformação Procedimento Externo:

```
const char* GetProcedureName() const;
```

Use a seguinte função para obter o índice do procedimento externo associado à transformação de Procedimento Externo:

```
inline unsigned long GetProcedureIndex() const;
```

Funções Relacionadas a Partições

Use funções relacionadas a partições para procedimentos externos em sessões com diversas partições. Quando você particiona uma sessão que contém transformações Procedimento Externo, o Serviço de Integração cria instâncias dessas transformações para cada partição. Por exemplo, se você definir cinco partições para uma sessão, o Serviço de Integração criará cinco instâncias de cada procedimento externo no tempo de execução da sessão.

Assinatura

Use a função a seguir para obter o número de partições em uma sessão:

```
unsigned long GetNumberOfPartitions();
```

Use a função a seguir para obter o índice da partição que chamou esse procedimento externo:

```
unsigned long GetPartitionIndex();
```

Função de Nível de Rastreamento

A função de nível de rastreamento retorna o nível de rastreamento da sessão. Por exemplo:

```
typedef enum
{
    TRACE_UNSET = 0,
    TRACE_TERSE = 1,
    TRACE_NORMAL = 2,
    TRACE_VERBOSE_INIT = 3,
    TRACE_VERBOSE_DATA = 4
} TracingLevelType;
```

Assinatura

Use a função a seguir para retornar o nível de rastreamento da sessão:

```
TracingLevelType GetSessionTraceLevel();
```


CAPÍTULO 9

Transformação Filtros

Este capítulo inclui os seguintes tópicos:

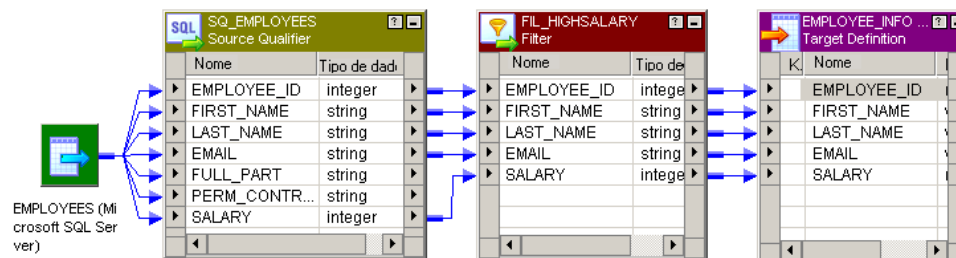
- [Visão Geral da Transformação Filtro, 201](#)
- [Componentes da Transformação Filtro, 202](#)
- [Condição de filtro, 202](#)
- [Etapas para Criar uma Transformação de Filtro, 203](#)
- [Dicas para Transformações de Filtro, 204](#)

Visão Geral da Transformação Filtro

Use a transformação Filtro para filtrar as linhas em um mapeamento. Como uma transformação ativa, a transformação Filtro pode alterar o número de linhas que passam por ela. A transformação Filtro permite a transferência de linhas que cumpram a condição de filtro especificada. Ela descarta linhas que não cumprem a condição. Você pode filtrar os dados com base em uma ou mais condições. A transformação de Filtro é uma transformação ativa.

Uma condição de filtro retorna TRUE ou FALSE para cada linha que o Serviço de Integração avalia, dependendo do fato de a linha cumprir ou não a condição especificada. Para cada linha que retorna TRUE, o Serviço de Integração transfere a transformação. Para cada linha que retorna FALSE, o Serviço de Integração descarta e escreve uma mensagem no log da sessão.

O seguinte mapeamento transfere as linhas de uma tabela de recursos humanos que contém os dados do funcionário, através de uma transformação Filtro. O filtro permite a transferência das linhas de funcionários que ganham salários de US\$30.000 ou mais.



Você não pode concatenar portas de mais de uma transformação na transformação Filtro. As portas de entrada para o filtro devem vir de uma única transformação.

Sugestão: Coloque a transformação Filtro o mais próximo possível das origens no mapeamento, a fim de maximizar o desempenho da sessão. Em vez de passar linhas que você queira descartar pelo mapeamento, você pode filtrar dados indesejáveis logo no fluxo de dados entre as origens e os destinos.

Componentes da Transformação Filtro

Você pode criar uma transformação de Filtro no Transformation Developer ou Mapping Designer. Uma transformação Filtro contém as guias a seguir:

- **Transformação.** Digite o nome e a descrição da transformação. A convenção de nomenclatura para uma transformação Filtro é `FIL_TransformationName`. Você também pode tornar a transformação reutilizável.
- **Portas.** Crie e configure portas.
- **Propriedades** Configure a condição de filtro para filtrar linhas. Use o Editor de Expressão para inserir a condição de filtro. Você também pode configurar o nível de rastreamento para determinar a quantidade de detalhes de transação relatados no arquivo de registro da sessão.
- **Extensões de Metadados.** Crie uma extensão de metadados não reutilizável para estender os metadados da transformação. Configure o nome, o tipo de dados, a precisão e o valor da extensão. Também é possível promover extensões de metadados para serem reutilizáveis para disponibilizá-las para todas as transformações.

Configuração das Portas da Transformação Filtro

Você pode criar e modificar portas na guia Portas.

Você pode configurar as seguintes propriedades na guia Portas:

- **Nome da porta.** Nome da porta.
- **Tipo de dados, precisão e escala.** Configure o tipo de dados e defina a precisão e a escala de cada porta.
- **Tipo de porta.** Todas as portas são de entrada/saída. As portas de entrada recebem dados e as portas de saída transferem dados.
- **Valores padrão e descrição.** Defina o valor padrão para portas e adicione a descrição.

Condição de filtro

A condição de filtro é uma expressão que retorna TRUE ou FALSE. Insira as condições usando o Editor de Expressão disponível na guia Propriedades.

Qualquer expressão de retorne um valor único pode ser usada como filtro. Por exemplo, se você deseja filtrar para *fora* as linhas dos funcionários cujos salários sejam inferior a US\$30.000, deve inserir a seguinte condição:

```
SALARY > 30000
```

Você pode especificar vários componentes da condição, usando os operadores lógicos AND e OR. Se deseja filtrar para fora os funcionários que ganham menos de US\$30.000 e mais de US\$100.000, deve inserir a seguinte condição:

```
SALARY > 30000 AND SALARY < 100000
```

Você também pode inserir uma constante para a condição de filtro. O equivalente numérico de FALSE é zero (0). Qualquer valor que não seja zero é o equivalente de TRUE. Por exemplo, a transformação contém uma porta com o nome NUMBER_OF_UNITS e um tipo de dados numérico. Você configura uma condição de filtro para retornar FALSE se o valor de NUMBER_OF_UNITS for igual a zero. Caso contrário, a condição retorna TRUE.

Não é necessário especificar TRUE ou FALSE como valores na expressão. TRUE e FALSE são valores de retorno implícitos a partir de qualquer condição que você configura. Se a condição de filtro avaliar como NULL, a linha é tratada como FALSE.

Nota: A condição de filtro faz distinção entre maiúsculas e minúsculas.

Filtrando Linhas com Valores Nulos

Para filtrar as linhas que contêm valores nulos ou espaços, use as funções ISNULL e IS_SPACES para testar o valor da porta. Por exemplo, se deseja filtrar as linhas que contêm um valor NULL na porta FIRST_NAME, use a seguinte condição:

```
IIF (ISNULL (FIRST_NAME) , FALSE, TRUE)
```

Esta condição afirma que se a porta FIRST_NAME é NULL, o valor de retorno é FALSE e a linha deve ser descartada. Do contrário, a linha é transferida para a próxima transformação.

Etapas para Criar uma Transformação de Filtro

Use o seguinte procedimento para criar uma transformação de Filtro:

1. No Mapping Designer, abra uma transformação Qualificador de origem.
2. Clique em Transformação > Criar. Selecione a transformação Filtro.
3. Insira um nome para a transformação. Clique em Criar e em Concluído.
4. Selecione e arraste todas as portas de um qualificador de origem ou outra transformação para adicioná-las à transformação Filtro.
5. Clique duas vezes na barra de título e clique na guia Portas. Você também pode criar porta manualmente na transformação.
6. Clique na guia Propriedades para configurar a condição de filtro e o nível de rastreamento.
7. Na seção Valor da condição de filtro, abra o Editor de Expressão.
8. Digite as condições de filtro a serem aplicadas. A condição padrão retorna TRUE.

Use valores de uma das portas de entrada na transformação como parte da condição. Mas você pode ainda usar valores de portas de saída em outras transformações.

9. Insira uma expressão. Clique em validar para verificar a sintaxe das condições inseridas.
10. Selecione o nível de rastreamento.
11. Adicione extensões de metadados na guia Extensões de Metadados.

Dicas para Transformações de Filtro

Use a transformação Filtro fornecida no mapeamento.

Para aumentar o desempenho da sessão, mantenha a transformação Filtro o mais próximo possível das origens no mapeamento. Em vez de passar linhas que você queira descartar pelo mapeamento, você pode filtrar dados indesejáveis logo no fluxo de dados entre as origens e os destinos.

Use a transformação de Qualificador de Origem para filtrar.

A transformação Qualificador de origem fornece um modo alternativo de filtrar linhas. Em vez de filtrar linhas de um mapeamento, a transformação Qualificador de origem filtra linhas quando lê a partir de uma origem. A principal diferença é que o qualificador de origem limita *o conjunto de linhas extraído de uma origem*, e a transformação Filtro limita *o conjunto de linhas enviados a um destino*. Como reduz o número de linhas usadas no mapeamento, o qualificador de origem proporciona um desempenho melhor.

Contudo, a transformação Qualificador de origem só permite filtrar linhas de origens relacionais e a transformação Filtro filtra linhas de qualquer tipo de origem. Observe ainda que, se for executada no banco de dados, certifique-se de que a condição de filtro na transformação Qualificador de origem use apenas SQL padrão. A transformação Filtro pode definir uma condição usando qualquer função de transformação ou instrução que retorne um valor TRUE ou FALSE.

CAPÍTULO 10

Transformação HTTP

Este capítulo inclui os seguintes tópicos:

- [Visão Geral da Transformação de HTTP, 205](#)
- [Criação de uma Transformação HTTP, 206](#)
- [Configuração da Guia Propriedades, 207](#)
- [Configuração da guia HTTP, 208](#)
- [Exemplos, 215](#)

Visão Geral da Transformação de HTTP

A transformação de HTTP traz a possibilidade de se conectar a um servidor de HTTP para usar seus serviços e aplicativos. A transformação de HTTP é uma transformação passiva. Ao executar uma sessão com uma transformação de HTTP, o Serviço de Integração se conecta ao servidor de HTTP e emite uma solicitação para recuperar ou atualizar dados no servidor HTTP, dependendo de como você configura a transformação:

- **Ler os dados do servidor HTTP.** Quando o Serviço de Integração lê dados de um servidor de HTTP, ele recupera os dados do servidor de HTTP e transfere os dados ao destino ou a uma transformação de fluxo para baixo no mapeamento. Por exemplo, é possível se conectar a um servidor de HTTP para ler dados atuais de inventário, efetuar cálculos com base nos dados durante a sessão do PowerCenter e transferir os dados ao destino.
- **Atualizar dados no servidor de HTTP.** Quando o Serviço de Integração efetua registros em um servidor de HTTP, ele posta dados no servidor de HTTP e transfere respostas do servidor de HTTP à transformação de destino ou de fluxo para baixo no mapeamento. Por exemplo, é possível postar dados fornecendo informações de agendamento de transformações contra a corrente ao servidor de HTTP durante uma sessão.

O Serviço de Integração transfere dados de transformações contra a corrente ou da origem à transformação de HTTP, lê um URL configurado na transformação ou conexão de aplicativo de HTTP e envia uma solicitação de HTTP ao servidor de HTTP para fins de leitura ou atualização de dados.

As solicitações contêm informações do cabeçalho e podem conter informações de corpo. O cabeçalho contém informações como parâmetros de autenticação, comandos para ativar programas ou serviços da Web que residem no servidor de HTTP, além de outras informações que se aplicam à totalidade da solicitação de HTTP. O corpo contém os dados que o Serviço de Integração envia ao servidor de HTTP.

Quando o Serviço de Integração envia uma solicitação de leitura de dados, o servidor de HTTP envia de volta uma resposta de HTTP com os dados solicitados. O Serviço de Integração envia os dados solicitados para transformações de fluxo para baixo ou para o destino.

Quando o Serviço de Integração envia uma solicitação para atualizar dados, o servidor de HTTP grava os dados que recebe e envia de volta uma resposta em HTTP informando que a atualização teve sucesso. A transformação de HTTP considera os códigos de resposta 200, 201 e 202 como códigos de êxito. A transformação de HTTP considera todos os outros códigos de resposta como falhas. O registro da sessão exibe um erro quando um servidor de HTTP transfere um código de resposta considerado uma falha para a transformação de HTTP. Em seguida, o Serviço de Integração envia a resposta em HTTP para transformações de fluxo para baixo ou para o destino.

É possível configurar a transformação de HTTP para os cabeçalhos de respostas de HTTP. Os dados do corpo de resposta de HTTP passam pela porta de saída HTTPOUT.

Autenticação

A Transformação HTTP usa as seguintes formas de autenticação:

- **Básica.** Baseada em um nome de usuário e senha não criptografados.
- **Avançada.** Baseada em um nome de usuário e senha criptografados.
- **NTLM.** Baseada em um nome de usuário, senha e domínio criptografados.

Conexão ao Servidor HTTP

Quando você configura uma transformação HTTP, pode configurar a URL para a conexão. Você também pode criar um objeto de conexão HTTP no Workflow Manager. Configure uma conexão de aplicativo HTTP nas circunstâncias a seguir:

- O servidor HTTP requer autenticação.
- Você deseja configurar o tempo limite da conexão.
- Você deseja substituir a URL básica na transformação HTTP.

Criação de uma Transformação HTTP

Crie transformações HTTP no Transformation Developer ou Mapping Designer. Uma transformação HTTP possui as seguintes guias:

- **Transformação.** Configure o nome e a descrição da transformação.
- **Portas.** Visualize as portas de entrada e saída da transformação. Você não pode adicionar nem editar portas na guia Portas. O Designer cria portas na guia Portas, quando você adiciona portas ao grupo do cabeçalho na guia HTTP.
- **Properties.** Configure as propriedades para a transformação HTTP na guia Propriedades.
- **HTTP.** Configure o método, as portas e a URL na guia HTTP.

Para criar uma transformação HTTP:

1. No Transformation Developer ou Mapping Designer, clique em Transformação > Criar.
2. Selecione transformação HTTP.
3. Insira um nome para a transformação.
4. Clique em Criar.
A transformação HTTP é exibida no espaço de trabalho.
5. Clique em Concluído.

6. Configure as guias na transformação.

Configuração da Guia Propriedades

A transformação HTTP foi desenvolvida usando a Transformação personalizada. Algumas propriedades da transformação Personalizada não se aplicam à transformação HTTP ou não são configuráveis.

A seguinte tabela descreve as propriedades da transformação HTTP que você pode configurar:

Opção	Descrição
Local do Tempo de Execução	<p>Local que contém a DLL ou a biblioteca compartilhada. O padrão é \$PMExtProcDir. Insira um caminho relativo ao nó do Serviço de Integração que executa a sessão de transformação HTTP.</p> <p>Se esta propriedade estiver em branco, o Serviço de Integração usará a variável de ambiente definida no Serviço de Integração para localizar a DLL ou a biblioteca compartilhada.</p> <p>Copie todas as DLLs ou bibliotecas compartilhadas para o local do tempo de execução ou para a variável de ambiente definida no nó do Serviço de Integração. O Serviço de Integração não carrega o procedimento quando não consegue localizar a DLL, a biblioteca compartilhada ou um arquivo especificado.</p>
Nível de rastreamento	Quantidade de detalhes exibidos no log de sessão dessa transformação. O padrão é Normal.
É Particionável	<p>Indica se você pode criar múltiplas partições em um pipeline, usando essa transformação:</p> <ul style="list-style-type: none">- Não. A transformação não pode ser particionada. A transformação e outras transformações no mesmo pipeline estão limitadas a uma partição.- Localmente. A transformação pode ser particionada, mas o Serviço de Integração deve executar todas as partições da pipeline no mesmo nó. Escolha Local quando diferentes partições da Transformação personalizada devem compartilhar objetos na memória.- Na Grade. A transformação pode ser particionada e o Serviço de Integração pode distribuir cada partição para nós diferentes. <p>O padrão é Não.</p>
Saída é repetível	<p>Indica se a ordem dos dados de saída é consistente entre as execuções de sessão.</p> <ul style="list-style-type: none">- Nunca. A ordem dos dados de saída é inconsistente entre as execuções de sessão.- Baseada na Ordem de Entrada. A ordem de saída é consistente entre as execuções de sessão quando a ordem dos dados de entrada é consistente entre elas.- Sempre. A ordem dos dados de saída será consistente entre as execuções de sessão mesmo se a ordem de entrada for inconsistente entre elas. <p>O padrão é Baseada na Ordem de Entrada.</p>
Requer Thread Único por Partição	Indica se o Serviço de Integração processa cada partição em um procedimento com um thread.
Saída é determinística	<p>Indica se a transformação gera dados de saída consistentes entre as execuções de sessão.</p> <p>Ative esta propriedade para executar a recuperação em sessões que usam essa transformação. O padrão é ativado.</p>

Aviso: Se você configurar uma transformação como repetível e determinística, será sua responsabilidade garantir que os dados sejam repetíveis e determinísticos. Se você tentar recuperar uma sessão com transformações que não gerem os mesmos dados entre a sessão e a recuperação, o processo de recuperação poderá resultar em dados corrompidos.

Configuração da guia HTTP

Na guia HTTP, você pode configurar a transformação para ler ou escrever os dados do servidor HTTP. Configure as seguintes informações na guia HTTP:

- **Selecione o método.** Selecione o método GET, POST, SIMPLE POST, SIMPLE PATCH, SIMPLE PUT ou SIMPLE DELETE com base em se você deseja ler ou gravar dados em um servidor HTTP, executar atualização parcial ou substituir todo o bloco de dados ou excluir dados do servidor HTTP.
- **Configure grupos e portas.** Gerencie o corpo da solicitação/resposta do HTTP e os detalhes do cabeçalho, configurando as portas de entrada e saída. Você também pode configurar nomes de porta com caracteres especiais.
- **Configure uma URL básica.** Configure a URL básica para o servidor HTTP ao qual você deseja se conectar.

Selecionando um Método

Os grupos e portas definidos em uma transformação dependem do método selecionado. Para ler dados de um servidor HTTP, selecione o método GET. Para gravar dados em um servidor HTTP, selecione o método POST ou SIMPLE POST.

A tabela a seguir explica os diferentes métodos:

Método	Descrição
GET	Lê dados de um servidor HTTP.
POST	Grava dados de diversas portas de entrada no servidor HTTP.
SIMPLE POST	Grava dados de uma porta de entrada como um único bloco de dados em um servidor HTTP.
SIMPLE PATCH	Atualiza dados parciais de uma porta de entrada como um patch para o recurso.
SIMPLE PUT	Substitui ou grava um recurso.
SIMPLE DELETE	Exclui um recurso do servidor HTTP.

Configurando Grupos e Portas

As portas que você adicionar a uma Transformação HTTP dependem do método e do grupo escolhidos. Uma Transformação HTTP usa os seguintes grupos:

- **Saída.** Contém dados do corpo para a resposta HTTP. Transfere respostas do servidor de HTTP para transformações de fluxo para baixo ou para o destino. Por padrão, contém uma porta de saída, HTTPOUT. Não é possível adicionar portas ao grupo de saída. É possível modificar a precisão da porta de saída HTTPOUT.
- **Entrada.** Contém dados do corpo para a solicitação HTTP. Também contém metadados que o Designer usa para formular um URL para conectar o servidor HTTP. Para gravar dados em um servidor HTTP, o grupo de entrada transfere informações de corpo ao servidor HTTP. Por padrão, contém uma porta de entrada.

- **Cabeçalho** Contém dados de cabeçalho para a solicitação e a resposta. Transfere informações do cabeçalho ao servidor HTTP quando o Serviço de Integração envia uma solicitação de HTTP. As portas que são acrescentadas ao grupo do cabeçalho transferem dados aos cabeçalhos HTTP. Ao adicionar portas ao grupo de cabeçalho, o Designer adiciona portas aos grupos de entrada e de saída na guia Portas. Por padrão, não contém portas. Para todos os métodos, você pode usar o grupo de cabeçalhos para informações de cabeçalho da solicitação HTTP.

Nota: Os dados que são transferidos por meio de uma transformação HTTP devem ser do tipo de dados String. Dados String abarcam qualquer linguagem de marcação que seja comum em comunicação por HTTP, como HTML e XML.

Método GET

Lê dados de um servidor HTTP. Para definir os metadados para a solicitação HTTP, use o grupo de entrada para adicionar portas de entrada que o Designer usa para construir a URL final para o servidor HTTP.

A tabela a seguir descreve os grupos e portas do método GET:

Solicitação/ Resposta:	Grupo	Descrição
REQUEST	Entrada	O Designer usa os nomes e valores das portas de entrada para formular a URL final.
REQUEST	Cabeçalho	É possível configurar portas de entrada e de entrada/saída para solicitações HTTP. O Designer adiciona portas aos grupos de entrada e de saída com base nas portas que você adiciona ao grupo de cabeçalho: <ul style="list-style-type: none"> - Grupo de entrada. Cria portas de entrada com base em portas de entrada e de entrada/saída a partir do grupo do cabeçalho. - Grupo de saída. Cria portas de saída com base em portas de entrada/saída a partir do grupo do cabeçalho.
RESPONSE	Cabeçalho	É possível configurar portas de saída e de entrada/saída para respostas HTTP. O Designer adiciona portas aos grupos de entrada e de saída com base nas portas que você adiciona ao grupo de cabeçalho: <ul style="list-style-type: none"> - Grupo de entrada. Cria portas de entrada com base em portas de entrada/saída no grupo do cabeçalho. - Grupo de saída. Cria portas de saída com base em portas de saída e de entrada/saída no grupo do cabeçalho.
RESPONSE	Saída	Todos os dados de corpo para uma resposta HTTP são transferidos por meio de uma porta de saída HTTPOUT.

Método POST

Grava dados de diversas portas de entrada no servidor HTTP. Para definir os metadados para a solicitação HTTP, use o grupo de entrada para os dados que definem o corpo da solicitação HTTP.

A tabela a seguir descreve as portas para o método POST:

Solicitação/ Resposta:	Grupo	Descrição
REQUEST	Entrada	É possível adicionar várias portas ao grupo de entrada. Os dados de corpo para uma solicitação HTTP podem ser transferidos por meio de uma ou mais portas de entrada com base no que é adicionado ao grupo de cabeçalho.
REQUEST	Cabeçalho	É possível configurar portas de entrada e de entrada/saída para solicitações HTTP. O Designer adiciona portas aos grupos de entrada e de saída com base nas portas que você adiciona ao grupo de cabeçalho: <ul style="list-style-type: none"> - Grupo de entrada. Cria portas de entrada com base em portas de entrada e de entrada/saída a partir do grupo do cabeçalho. - Grupo de saída. Cria portas de saída com base em portas de entrada/saída a partir do grupo do cabeçalho.
RESPONSE	Cabeçalho	É possível configurar portas de saída e de entrada/saída para respostas HTTP. O Designer adiciona portas aos grupos de entrada e de saída com base nas portas que você adiciona ao grupo de cabeçalho: <ul style="list-style-type: none"> - Grupo de entrada. Cria portas de entrada com base em portas de entrada/saída no grupo do cabeçalho. - Grupo de saída. Cria portas de saída com base em portas de saída e de entrada/saída no grupo do cabeçalho.
RESPONSE	Saída	Todos os dados de corpo para uma resposta HTTP são transferidos por meio de uma porta de saída HTTPOUT.

Método SIMPLE POST

Uma versão simplificada do método POST. Grava dados de uma porta de entrada como um único bloco de dados em um servidor HTTP. Para definir os metadados para a solicitação HTTP, use o grupo de entrada para os dados que definem o corpo da solicitação HTTP.

A seguinte tabela descreve as portas do método SIMPLE POST:

Solicitação/ Resposta	Grupo	Descrição
REQUEST	Entrada	É possível adicionar uma porta de entrada. Os dados de corpo para uma solicitação HTTP podem ser transferidos por uma porta de entrada.
REQUEST	Cabeçalho	É possível configurar portas de entrada e de entrada/saída para solicitações HTTP. O Designer adiciona portas aos grupos de entrada e de saída com base nas portas que você adiciona ao grupo de cabeçalho: <ul style="list-style-type: none"> - Grupo de entrada. Cria portas de entrada com base em portas de entrada e de entrada/saída a partir do grupo do cabeçalho. - Grupo de saída. Cria portas de saída com base em portas de entrada/saída a partir do grupo do cabeçalho.

Solicitação/ Resposta	Grupo	Descrição
RESPONSE	Cabeçalho	É possível configurar portas de saída e de entrada/saída para respostas HTTP. O Designer adiciona portas aos grupos de entrada e de saída com base nas portas que você adiciona ao grupo de cabeçalho: <ul style="list-style-type: none"> - Grupo de entrada. Cria portas de entrada com base em portas de entrada/saída no grupo do cabeçalho. - Grupo de saída. Cria portas de saída com base em portas de saída e de entrada/saída no grupo do cabeçalho.
RESPONSE	Saída	Todos os dados de corpo para uma resposta HTTP são transferidos por meio de uma porta de saída HTTPOUT.

Método SIMPLE PATCH

Atualiza dados parciais de uma porta de entrada como um patch para o recurso. Grava dados de uma porta de entrada como um bloco de dados completo ou parcial em um servidor HTTP. Para definir os metadados para a solicitação HTTP, use o grupo de entrada para os dados que definem o corpo da solicitação HTTP.

A seguinte tabela descreve as portas do método SIMPLE PATCH:

Solicitação/ Resposta	Grupo	Descrição
REQUEST	Entrada	É possível adicionar uma porta de entrada. Os dados de corpo para uma solicitação HTTP podem ser transferidos por uma porta de entrada.
REQUEST	Cabeçalho	É possível configurar portas de entrada e de entrada/saída para solicitações HTTP. O Designer adiciona portas aos grupos de entrada e de saída com base nas portas que você adiciona ao grupo de cabeçalho: <ul style="list-style-type: none"> - Grupo de entrada. Cria portas de entrada com base em portas de entrada e de entrada/saída a partir do grupo do cabeçalho. - Grupo de saída. Cria portas de saída com base em portas de entrada/saída a partir do grupo do cabeçalho.
RESPONSE	Cabeçalho	É possível configurar portas de saída e de entrada/saída para respostas HTTP. O Designer adiciona portas aos grupos de entrada e de saída com base nas portas que você adiciona ao grupo de cabeçalho: <ul style="list-style-type: none"> - Grupo de entrada. Cria portas de entrada com base em portas de entrada/saída no grupo do cabeçalho. - Grupo de saída. Cria portas de saída com base em portas de saída e de entrada/saída no grupo do cabeçalho.
RESPONSE	Saída	Todos os dados de corpo para uma resposta HTTP são transferidos por meio de uma porta de saída HTTPOUT.

Método SIMPLE PUT

Substitui ou grava um recurso. Grava dados de uma porta de entrada como um único bloco de dados em um servidor HTTP.

Se os dados não existirem, o método SIMPLE PUT os publicará. Se os dados existirem, o método SIMPLE PUT atualizará os dados de uma porta de entrada como um único bloco de dados para o servidor HTTP.

Para definir os metadados para a solicitação HTTP, use o grupo de entrada para os dados que definem o corpo da solicitação HTTP.

A seguinte tabela descreve as portas do método SIMPLE PUT:

Solicitação/ Resposta	Grupo	Descrição
REQUEST	Entrada	É possível adicionar uma porta de entrada. Os dados de corpo para uma solicitação HTTP podem ser transferidos por uma porta de entrada.
REQUEST	Cabeçalho	É possível configurar portas de entrada e de entrada/saída para solicitações HTTP. O Designer adiciona portas aos grupos de entrada e de saída com base nas portas que você adiciona ao grupo de cabeçalho: <ul style="list-style-type: none"> - Grupo de entrada. Cria portas de entrada com base em portas de entrada e de entrada/saída a partir do grupo do cabeçalho. - Grupo de saída. Cria portas de saída com base em portas de entrada/saída a partir do grupo do cabeçalho.
RESPONSE	Cabeçalho	É possível configurar portas de saída e de entrada/saída para respostas HTTP. O Designer adiciona portas aos grupos de entrada e de saída com base nas portas que você adiciona ao grupo de cabeçalho: <ul style="list-style-type: none"> - Grupo de entrada. Cria portas de entrada com base em portas de entrada/saída no grupo do cabeçalho. - Grupo de saída. Cria portas de saída com base em portas de saída e de entrada/saída no grupo do cabeçalho.
RESPONSE	Saída	Todos os dados de corpo para uma resposta HTTP são transferidos por meio de uma porta de saída HTTPOUT.

Método SIMPLE DELETE

Exclui um recurso do servidor HTTP. Se o corpo da solicitação for necessário, SIMPLE DELETE excluirá os dados de uma porta de entrada como um único bloco de dados para o servidor HTTP. Para definir os metadados para a solicitação HTTP, use o grupo de entrada para adicionar portas de entrada que o Designer usa para construir a URL final para o servidor HTTP.

A seguinte tabela descreve as portas do método SIMPLE DELETE:

Solicitação/ Resposta:	Grupo	Descrição
REQUEST	Entrada	O Designer usa os nomes e valores das portas de entrada para formular a URL final.
REQUEST	Cabeçalho	É possível configurar portas de entrada e de entrada/saída para solicitações HTTP. O Designer adiciona portas aos grupos de entrada e de saída com base nas portas que você adiciona ao grupo de cabeçalho: <ul style="list-style-type: none"> - Grupo de entrada. Cria portas de entrada com base em portas de entrada e de entrada/saída a partir do grupo do cabeçalho. - Grupo de saída. Cria portas de saída com base em portas de entrada/saída a partir do grupo do cabeçalho.

Solicitação/ Resposta:	Grupo	Descrição
RESPONSE	Cabeçalho	É possível configurar portas de saída e de entrada/saída para respostas HTTP. O Designer adiciona portas aos grupos de entrada e de saída com base nas portas que você adiciona ao grupo de cabeçalho: <ul style="list-style-type: none"> - Grupo de entrada. Cria portas de entrada com base em portas de entrada/saída no grupo do cabeçalho. - Grupo de saída. Cria portas de saída com base em portas de saída e de entrada/saída no grupo do cabeçalho.
RESPONSE	Saída	Todos os dados de corpo para uma resposta HTTP são transferidos por meio de uma porta de saída HTTPOUT.

Adição de um nome HTTP

O Designer não permite caracteres especiais, como um hífen (-), nos nomes de portas. Se você precisar usar caracteres especiais no nome da porta, pode configurar um nome HTTP para substituir o nome da porta. Por exemplo, se deseja inserir uma porta com o nome Content-type, pode nomeá-la como ContentType e inserir Content-Type como o nome HTTP.

Configuração de uma URL

Depois de selecionar um método e configurar as portas de entrada e saída, você deve configurar uma URL. Insira uma URL básica e o Designer cria a URL final. Se você selecionar o método GET ou SIMPLE DELETE, a URL final conterá a URL básica e os parâmetros baseados nos nomes de porta no grupo de entrada. Se você selecionar os métodos SIMPLE PATCH, SIMPLE PUT, POST ou SIMPLE POST, a URL final será igual à de base.

Você pode usar um parâmetro ou variável de mapeamento para configurar a URL básica. Por exemplo, declare o parâmetro de mapeamento \$\$ParamBaseURL, insira o parâmetro de mapeamento \$ \$ParamBaseURL no campo da URL básica, e então defina \$\$ParamBaseURL no arquivo de parâmetro.

Você também pode especificar uma URL quando configura uma conexão do aplicativo HTTP. A URL básica especificada na conexão do aplicativo HTTP anula a URL básica especificada na transformação HTTP.

Nota: Um servidor HTTP pode redirecionar uma solicitação de HTTP para outro servidor HTTP. Quando isto ocorre, o servidor HTTP manda uma URL de volta ao Serviço de Integração, que então estabelece uma conexão ao outro servidor HTTP. O Serviço de Integração pode estabelecer no máximo cinco conexões adicionais.

Construção Final de URL para o Método GET

O Designer constrói a URL final para o método GET baseada na URL base e nos nomes das portas no grupo de entrada. Ele acrescenta argumentos de HTTP a URL de base para construir a URL final na forma de uma string de consulta de HTTP. Uma string de consulta consiste em um ponto de interrogação (?) seguido de pares de nome/valor. O Designer acrescenta o ponto de interrogação e os pares de nome/valor que correspondem aos nomes e valores das portas de entrada adicionadas ao grupo de entrada.

Ao selecionar o método GET e adicionar portas de entrada ao grupo de entrada, o Designer acrescenta as seguintes informações de grupo e de porta a URL de base para construir a URL final:

```
?<input group input port 1 name> = <input group input port 1 value>
```

Para cada porta de entrada que vem logo após a porta de entrada do grupo de entrada, o Designer acrescenta as seguintes informações de grupo e de porta:

```
& <input group input port n name> = $<input group input port n value>
```

onde *n* representa a porta de entrada.

Por exemplo, se você inserir `www.company.com` como a URL de base e adicionar as portas de entrada `ID`, `EmpName` e `Department` ao grupo de entrada, o Designer constrói a seguinte URL final:

```
www.company.com?ID=$ID&EmpName=$EmpName&Department=$Department
```

É possível editar a URL final para modificar ou adicionar operadores, variáveis ou outros argumentos. Para obter mais informações sobre solicitações de HTTP e string de consulta, acesse <http://www.w3c.org>.

Parametrizar URL base

Você pode parametrizar a URL base para os métodos GET e SIMPLE DELETE.

Por exemplo, você tem a seguinte URL base:

```
www.informatica.com
```

O Designer usa as informações na URL base e pode construir a URL final da seguinte maneira:

```
www.informatica.com?firstname=1&lastname=2
```

Ao marcar a opção da caixa de seleção para parametrizar a URL base no Designer para o método GET ou SIMPLE DELETE, você pode editar a URL base para usar o parâmetro ou a variável de mapeamento da seguinte maneira:

```
www.informatica.com/$firstname$lastname
```

Os Serviços de Integração enviam os valores do arquivo de origem à porta de entrada de nome e sobrenome da transformação de HTTP e enviam as solicitações HTTP ao servidor HTTP especificado na URL final.

A URL final pode aparecer da seguinte maneira:

```
www.informatica.com/12
```

Caracteres especiais na URL

Quando você configura uma URL, a URL final pode conter os caracteres especiais codificados no conjunto de caracteres UTF-8.

A tabela a seguir descreve como alguns caracteres especiais usados na URL são configurados no ambiente UTF-8:

Caracteres especiais	Codificado (UTF-8)
,	%27
(%28
)	%29
*	2A
espaço	%20

Se você usar o sinalizador personalizado `UseURLAsEnteredinCURLEncoding` no nível do Serviço de Integração ou na sessão, poderá transmitir a URL exatamente como digitada na URL Base (formato de codificação CURL).

Exemplos

Esta seção contém exemplos para cada tipo de método:

- GET
- POST
- SIMPLE POST
- SIMPLE PATCH
- SIMPLE PUT
- SIMPLE DELETE

Exemplo de GET

O arquivo de origem usado neste exemplo contém os seguintes dados:

```
78576
78577
78578
```

A seguinte figura mostra a guia HTTP da transformação HTTP para o exemplo GET:

Editar Transformações

Transformação | Portas | Propriedades | HTTP

Nome da transformação: HTTP

Método HTTP: GET

	Nome	Tipo de dados	Precisão	Escala	I	O
1						SAÍDA
2	HTTPOUT	string	65535	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>
3						ENTRADA
4	NEWFIELD	string	10	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5						CABEÇALHO

Valor Padrão:

Descrição:

Nome HTTP:

URL Base: `http://www.informatica.com`

URL Final: `http://www.informatica.com?NEWFIELD=$NEWFIELD`

OK Cancelar Aplicar Ajuda

O Designer acrescenta um ponto de interrogação (?), o nome da porta de entrada do grupo de entrada, um cifrão (\$) e, mais uma vez, o nome da porta de entrada do grupo de entrada para a URL básica para gerar a URL final:

```
http://www.informatica.com?CR=$CR
```

O Serviço de Integração envia os valores do arquivo de origem à porta de entrada CR da transformação HTTP e envia as seguintes solicitações de HTTP ao servidor HTTP:

```
http://www.informatica.com?CR=78576
http://www.informatica.com?CR=78577
http://www.informatica.com?CR=78578
```

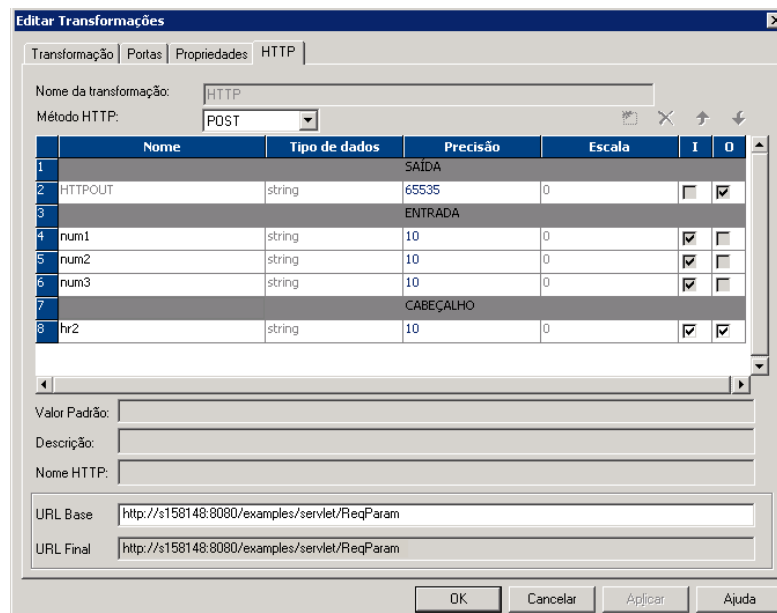
O servidor HTTP envia uma resposta HTTP de volta ao Serviço de Integração, que envia os dados por meio da porta de saída da transformação HTTP para o destino.

Exemplo de POST

O arquivo de origem usado neste exemplo contém os seguintes dados:

```
33,44,1
44,55,2
100,66,0
```

A figura a seguir mostra que cada campo do arquivo de origem contém uma porta de entrada correspondente:



O Serviço de Integração envia os valores dos três campos de cada linha pelas portas de entrada da transformação de HTTP e envia a solicitação HTTP ao servidor HTTP especificado na URL final.

Exemplo SIMPLE POST

O texto a seguir mostra o arquivo XML usado neste exemplo:

```
<?xml version="1.0" encoding="UTF-16LE"?>
<n4:Envelope xmlns:cli="http://localhost:8080/axis/Clienttest1.jws" xmlns:n4="http://
schemas.xmlsoap.org/soap/envelope/" xmlns:tns="http://schemas.xmlsoap.org/soap/
encoding/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance/" xmlns:xsd="http://
www.w3.org/2001/XMLSchema">
<n4:Header>
```



```

</n4:Header>
<n4:Body n4:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"><cli:smpsource>
<Metadatainfo xsi:type="xsd:string">smpsourceRequest.Metadatainfo106</Metadatainfo></
cli:smpsource>
</n4:Body>
<n4:Envelope>, capeconnect:Clienttest1services:Clienttest1#smpsource

```

A figura a seguir mostra a guia HTTP da transformação HTTP para o exemplo SIMPLE POST:

Editar Transformações

Transformação | Portas | Propriedades | **HTTP**

Nome da transformação:

Método HTTP:

	Nome	Tipo de dados	Precisão	Escala	I	O
1						SAÍDA
2	HTTPOUT	string	65535	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>
3						ENTRADA
4	SOAPRequest	string	65535	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5						CABEÇALHO
6	soapaction	string	100	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
7	host	string	100	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>
8	date	string	100	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>
9	server	string	100	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Valor Padrão:

Descrição:

Nome HTTP:

URL Base:

URL Final:

OK Cancelar Aplicar Ajuda

O Serviço de Integração envia o corpo do arquivo de origem pela porta de entrada e envia a solicitação HTTP ao servidor HTTP especificado na URL final.

Exemplo de SIMPLE PATCH

O arquivo de origem usado neste exemplo contém os seguintes dados:

```

{"firstname": "Raj"}

```

A figura a seguir mostra que cada campo do arquivo de origem contém uma porta de entrada correspondente:

Edit Transformations

Transformation Ports Properties **HTTP**

Static

Transformation Name: HTTP3

HTTP Method: SIMPLE PATCH

	Name	Datatype	Precision	Scale	I	O
1			OUTPUT			
2	HTTPOUT	string	65535	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>
3			INPUT			
4	update	string	1000	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5			HEADER			
6	ContentType	string	1000	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Default Value: 'application/json'

Description:

HTTP Name: Content-Type

Base URL: \$\$BASEURL1

Final URL: \$\$BASEURL1

☐ Parameterize Base URL

OK Cancel Apply Help

Inclua a variável de mapeamento \$\$BASEURL1 como a URL base e a URL final.

A variável de mapeamento \$\$BASEURL1 aponta para o seguinte link:

http://avengers5:8181/emp_all/1/

SIMPLE PATCH oferece suporte à atualização parcial de dados de uma porta de entrada para o servidor HTTP. O Serviço de Integração envia os valores da linha afetada pelas portas de entrada da transformação de HTTP e envia a solicitação HTTP ao servidor HTTP especificado na URL final.

Exemplo de SIMPLE PUT

O arquivo de origem usado neste exemplo contém os seguintes dados:

```
{"emp_id": 10,"firstname": "Raj","lastname": "Kumar","designation": "QA","department": "RND","age": 24}
```

A figura a seguir mostra que cada campo do arquivo de origem contém uma porta de entrada correspondente:

Edit Transformations

Transformation Ports Properties **HTTP**

Static

Transformation Name: HTTP

HTTP Method: SIMPLE PUT

	Name	Datatype	Precision	Scale	I	O
1			OUTPUT			
2	HTTPOUT	string	65535	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>
3			INPUT			
4	update	string	1000	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5			HEADER			
6	contentType	string	100	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Default Value:

Description:

HTTP Name:

Base URL:

Final URL:

☐ Parameterize Base URL

OK Cancel Apply Help

SIMPLE PUT grava dados de uma porta de entrada como um único bloco de dados em um servidor HTTP. O Serviço de Integração envia o corpo do arquivo de origem pela porta de entrada da transformação HTTP e envia a solicitação HTTP ao servidor HTTP especificado na URL final.

Exemplo de SIMPLE DELETE

O arquivo de origem usado neste exemplo contém os seguintes dados:

2
1

A figura a seguir mostra a guia HTTP da transformação de HTTP para o exemplo SIMPLE DELETE com a opção URL de Base Parametrizada selecionada:

Edit Transformations

Transformation Ports Properties **HTTP**

Static

Transformation Name:

HTTP Method:

	Name	Datatype	Precision	Scale	I	O
1			OUTPUT			
2	HITPOUT	string	65535	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>
3			INPUT			
4	id	string	10	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5			HEADER			
6	ContentType	string	10	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Default Value:

Description:

HTTP Name:

Base URL:

Final URL:

☒ Parameterize Base URL

OK Cancel Apply Help

Quando você marca a caixa de seleção Parametrizar URL Base, o Designer anexa o nome da porta de entrada do grupo de entrada à URL base para construir a URL final:

```
http://www.avengers5:8181/emp_all/$id/
```

O Serviço de Integração envia os valores do arquivo de origem à porta de entrada ID da transformação de HTTP e envia as seguintes solicitações de HTTP ao servidor HTTP:

```
http://www.avengers5:8181/emp_all/2
http://www.avengers5:8181/emp_all/1
```

O método SIMPLE DELETE exclui os dados necessários do servidor HTTP. O servidor HTTP envia uma resposta HTTP de volta ao Serviço de Integração, que envia os dados atualizados por meio da porta de saída da transformação de HTTP ao destino.

CAPÍTULO 11

Transformação de Resolução de Identidade

Este capítulo inclui os seguintes tópicos:

- [Visão Geral da Transformação de Resolução de Identidade, 221](#)
- [Criar e Configurar a Transformação, 221](#)
- [Guias da Transformação de Resolução de Identidade , 224](#)
- [Grupos e Portas, 225](#)

Visão Geral da Transformação de Resolução de Identidade

A transformação Resolução de Identidade é uma transformação ativa que você pode usar para pesquisar e corresponder dados na Resolução de Identidade da Informatica (IIR). O Serviço de Integração do PowerCenter usa a definição de pesquisa que você especifica na transformação Resolução de Identidade para pesquisar e corresponder dados residentes nas tabelas de IIR. As exibições de entrada e saída no sistema determinam as portas de entrada e saída da transformação.

Configure parâmetros de tolerância de correspondência e de amplitude de pesquisa na transformação Resolução de Identidade para definir o esquema de correspondência e o nível de pesquisa. Quando você executa uma sessão, o IIR retorna os registros candidatos à transformação Resolução de Identidade.

O IIR fornece pesquisa, correspondência, triagem, vinculação e descoberta duplicada on-line para todos os tipos de dados de identificação armazenados em tabelas Oracle, DB2/UDB e Microsoft SQL Server. O IIR usa o Servidor de Pesquisa do IIR para operações de pesquisa e correspondência.

Criar e Configurar a Transformação

Crie uma transformação Resolução de Identidade no Transformation Developer. Quando você cria uma transformação Resolução de Identidade, primeiro conecta-se ao Servidor de Pesquisa de Resolução de Identidade da Informatica. O Servidor de Pesquisa fornece acesso às tabelas IIR.

Após conectar-se, você configura o sistema e os parâmetros de pesquisa e correspondência. Configure as informações a seguir na janela **Configurar Transformação IR**:

- Conexão ao Servidor de Pesquisa de IR
- Configuração do sistema e da pesquisa
- Exibições de entrada e saída

Conexão do Servidor de Pesquisa

Para conexão com o Servidor de Pesquisa, você fornece um nome de host e um número de porta. Em seguida, você configura a string de conexão Rulebase. A Rulebase contém informações sobre sistemas que o IIR usa para pesquisar e fazer a correspondência de dados. A conexão Rulebase em uma transformação de Resolução de Identidade especifica o banco de dados que contém a Rulebase.

Nota: Você não pode alterar o nome do host ou a conexão da Rulebase depois de criar a transformação.

Nome do Host do IR

Nome de host da máquina em que é executado o Servidor de Resolução de Identidade da Informatica.

Porta

Número da porta do Servidor de Pesquisa.

String de Conexão da Rulebase

String de conexão ao Servidor de Pesquisa. Você pode especificar a conexão da Rulebase como ODBC ou SSA. Configure o ODBC e a SSA no host do IIR.

ODBC

A tabela a seguir descreve as informações de conexão do ODBC:

Propriedade de Conexão	Descrição
Número da Rulebase	O número designado para a Rulebase do IR.
Nome de usuário	Nome de usuário do banco de dados.
Senha	Senha do nome de usuário do banco de dados.
Nome do serviço	Nome do serviço definido no arquivo odbc.ini.

SSA

Para estabelecer conexão por meio da SSA, você precisa fornecer o alias. O alias oculta a string de conexão dos programas aplicativos. O IIR usa aliases para Rulebases, bancos de dados e nomes de origem.

Configuração do Sistema e da Pesquisa

Após conectar-se ao Servidor de Pesquisa, você escolhe o sistema e os parâmetros de pesquisa e correspondência.

Você pode alterar esses parâmetros na guia **Configurações do IR** após criar a transformação.

Sistema

Escolha o sistema na Rulebase. Um sistema é a definição lógica de nível mais alto em uma Rulebase do IIR.

Definição de Pesquisa

Escolha a definição de pesquisa que contenha as regras para o sistema. A definição de pesquisa reside em um sistema e contém as regras para pesquisar, fazer a correspondência de registros e exibi-los.

Amplitude da Pesquisa

Escolha a amplitude de pesquisa que determine o escopo da pesquisa que você deseja conduzir.

A tabela a seguir descreve as opções de amplitude de pesquisa que você pode escolher:

Amplitude da Pesquisa	Descrição
Restrita	Permite uma pesquisa restrita com um tempo de resposta rápido. Use essa opção quando houver um baixo risco associado à perda de uma correspondência, quando você não exigir uma alta precisão de correspondência ou quando o volume de dados for grande e o tempo de resposta for crítico.
Típico	Propicia um equilíbrio entre a qualidade e o tempo de resposta. Use essa opção para pesquisas de transação on-line ou em lotes. O padrão é Típica.
Abrangente	Permite uma pesquisa detalhada com um tempo de resposta estendido. Use essa opção quando houver um alto risco associado à perda de uma correspondência, quando a qualidade dos dados for uma prioridade ou quando o volume de dados for baixo e o tempo de resposta não for crítico.

Tolerância de Correspondência

Escolha a tolerância de correspondência que determine o grau de acurácia do esquema de correspondência para selecionar os registros candidatos.

A tabela a seguir descreve as opções de tolerância de correspondência que você pode escolher:

Tolerância de Correspondência	Descrição
Extrema	Mostra um candidato à correspondência com um conjunto de métodos possíveis para processamento. O tempo de resposta para processamento é maior. Use essa opção quando precisar encontrar uma correspondência mesmo que ela contenha erros e variações.
Conservadora	Mostra correspondências que não são excessivas ou extremas. Use essa opção para pesquisas de transação on-line ou em lotes.
Típica	Mostra correspondências precisas. Use em sistemas de lote quando precisar de uma correspondência acurada.
Ampla	Mostra correspondências com um alto grau de variação em relação ao nível típico. Use em sistemas nos quais o risco de perder uma correspondência é alto e quando você puder analisar os resultados.

Exibir Seleção

Após selecionar o sistema e a definição da pesquisa, você pode escolher a exibição de entrada e de saída.

Exibição de Entrada

Os campos da exibição de entrada determinam as portas de entrada da transformação. A exibição de entrada contém os campos usados para operações de pesquisa e correspondência. Se você não escolher uma exibição de entrada, a transformação usará o layout da Tabela de Identidade do IIR (IDT) para a exibição de entrada.

Nota: Se o sistema contiver a cláusula XFORM na definição de exibição para concatenar vários campos em um só campo, você não poderá usar a exibição na transformação de Resolução de Identidade. Contudo, você pode concatenar vários campos fora da transformação de Resolução de Identidade, e passar a string concatenada às portas de entrada da transformação de Resolução de Identidade.

Exibição de Saída

Os campos da exibição de saída determinam as portas de saída da transformação. A exibição de saída formata os resultados da pesquisa retornados pelo Servidor de Pesquisa. Se você não escolher uma exibição de saída, a transformação usará o layout da Tabela de Identidade do IIR (IDT) para a exibição de saída.

Guias da Transformação de Resolução de Identidade

A transformação de Resolução de Identidade tem as seguintes guias:

Transformação

Configure a descrição da transformação.

Portas

Veja os grupos e portas que representam as exibições de entrada e saída. Você não pode editar os grupos e portas.

Propriedades

A tabela a seguir descreve as opções que você pode configurar na guia **Propriedades**:

Atributo de Transformação	Descrição
Localização em Tempo de Execução	Local que contém a DLL ou biblioteca compartilhada. O padrão é \$PMEExtProcDir.
Nível de rastreamento	Detalhes exibidos no log de sessão para esta transformação. O padrão é Normal.
É Particionável	Indica que você pode criar várias partições em um pipeline que use esta transformação: <ul style="list-style-type: none">- Não A transformação não pode ser particionada. A transformação e as outras transformações no mesmo pipeline estão limitadas a uma partição.- Localmente. A transformação pode ser particionada, mas o Serviço de Integração deve executar todas as partições do pipeline no mesmo nó. Escolha Localmente quando diferentes partições da transformação Personalizada precisarem compartilhar objetos na memória.- Na Grade. A transformação pode ser particionada e o Serviço de Integração pode distribuir cada partição em nós diferentes. O padrão é Na Grade.

Configurações de IR

Veja as configurações que você definiu quando criou a transformação. Você pode alterar as informações do sistema e de pesquisa.

Grupos e Portas

Uma transformação de Resolução de Identidade contém um grupo de entrada e um grupo de saída. O grupo de entrada tem portas que representam campos na exibição de entrada da definição de pesquisa. O grupo de saída tem portas que representam campos na exibição de saída da definição de pesquisa além de portas que descrevem o resultado da pesquisa.

Você pode exibir grupos e portas na guia **Portas**. Você também pode exibir portas na guia **Configurações de IR**.

Grupos e Portas de Entrada

Uma transformação de Resolução de Identidade contém um grupo de entrada com portas de entrada. O número de portas do grupo de entrada corresponde ao número de campos da exibição de entrada selecionada. Vincule todas as portas de entrada requeridas pela pesquisa à transformação de Resolução de Identidade.

Grupos e Portas de Saída

O grupo de saída contém portas de saída e uma porta de entrada/saída. O grupo de saída contém portas da exibição de saída além de portas padrão para cada transformação de Resolução de Identidade.

As portas de saída que se baseiam na exibição de saída contêm registros de candidatos do Servidor de Pesquisa.

A tabela a seguir descreve as portas de saída padrão:

Portas de Saída Padrão	Descrição
IR_Search_Link	Porta do link de pesquisa de entrada/saída. Use essa porta de passagem para passar um link entre os dados de origem da entrada e os candidatos à saída resultantes.
IR_Score_Out	Porta de saída que contém a pontuação da operação de pesquisa conduzida pelo Servidor de Pesquisa. Os valores são números positivos de 0 a 100. Por exemplo, se você pesquisar um nome com a entrada "Rob", os resultados poderão conter registros candidatos com o nome "Rob" que têm uma pontuação de 100. Os resultados também podem conter registros candidatos com o nome "Bob" que têm uma pontuação de 90. A Resolução de Identidade da Informatica calcula a pontuação associada ao resultado.
IR_Result_Count	Porta de saída que contém o número de registros.

CAPÍTULO 12

Transformação Java

Este capítulo inclui os seguintes tópicos:

- [Visão Geral de Transformações Java, 226](#)
- [Usando a Guia Código Java, 229](#)
- [Configurando portas, 230](#)
- [Configuração das Propriedades da Transformação Java, 232](#)
- [Desenvolvimento do Código Java, 234](#)
- [Configurando definições da transformação Java, 240](#)
- [Compilando uma transformação Java, 242](#)
- [Corrigindo erros de compilação, 243](#)

Visão Geral de Transformações Java

Amplie a funcionalidade do PowerCenter com a transformação Java. A transformação Java fornece uma interface de programação nativa simples, para definir a funcionalidade de transformação com a linguagem de programação Java. Você pode usar a transformação Java para definir rapidamente a funcionalidade de transformação simples ou moderadamente complexa, sem conhecimento avançado da linguagem de programação Java ou um ambiente de desenvolvimento Java externo. A transformação Java pode ser uma transformação passiva ou ativa.

O Cliente do PowerCenter usa o Kit de Desenvolvimento de Java (JDK) para compilar o código Java e gerar o código de bytes para a transformação. O Cliente do PowerCenter armazena o código de bytes no repositório do PowerCenter.

O Serviço de Integração usa o Java Runtime Environment (JRE) para executar código de bytes gerado em tempo de execução. Quando o Serviço de Integração executa uma sessão com uma transformação Java, ele usa o JRE para executar o código de bytes e as linhas de entrada do processo e gerar linhas de saída.

Crie transformações Java escrevendo snippets de código Java que definem a lógica da transformação. Defina o comportamento para uma transformação Java, com base nos eventos a seguir:

- A transformação recebe uma linha de entrada.
- A transformação processou todas as linhas de entrada.
- A transformação recebe uma notificação de transação como confirmação ou reversão.

TÓPICOS RELACIONADOS:

- [“Referência da API da transformação Java” na página 245](#)
- [“Expressões Java” na página 258](#)

Etapas para Definir uma Transformação Java

Execute as etapas a seguir para gravar e compilar um código Java e corrigir erros de compilação em uma transformação Java:

1. Crie a transformação no Transformation Developer ou no Mapping Designer.
2. Configure portas de entrada e saída e grupos para a transformação. Utilize nomes de porta como variáveis nos trechos de código Java.
3. Configure as propriedades da transformação.
4. Utilize as guias de entrada de código na transformação para gravar e compilar o código Java para a transformação.
5. Localize e corrija erros de compilação no código Java para a transformação.

Transformações Java Ativas e Passivas

Ao criar uma transformação Java, você define o tipo como ativo ou passivo.

Depois de definir o tipo de transformação, não será possível alterá-la.

Uma transformação Java executa o código Java que você define na guia **Na Entrada Na Linha de Entrada** uma vez para cada linha de dados de entrada.

Uma transformação Java trata linhas de saída com base no tipo de transformação da seguinte forma:

- Uma transformação Java passiva gera uma linha de saída para cada linha de entrada na transformação após processar cada linha de entrada.
- Uma transformação Java ativa gera várias linhas de saída para cada linha de entrada na transformação.

Use o método `generateRow` para gerar cada linha de saída. Por exemplo, se a transformação contiver duas portas de entrada que representem uma data de início e uma data de término, você poderá usar o método `generateRow` para gerar uma linha de saída para cada data entre a data de início e a data de término.

Conversão de Tipo de Dados

Uma transformação Java converte os tipos de dados da Developer tool do PowerCenter em tipos de dados Java com base no tipo da porta da transformação Java.

Quando uma transformação Java lê linhas de entrada, ela converte tipos de dados da porta de entrada em tipos de dados Java.

Quando uma transformação Java grava as linhas de saída, ela converte os tipos de dados Java em tipos de dados da porta de saída.

Por exemplo, o seguinte processamento ocorre para uma porta de entrada com o tipo de dados `integer` em uma transformação Java:

1. A transformação Java converte o tipo de dados `integer` da porta de entrada para o tipo de dado `int` primitivo Java.

2. Na transformação, a transformação trata o valor da porta de entrada como o tipo de dado int primitivo Java.
3. Quando a transformação gera a linha de saída, ele converte o tipo de dado int primitivo Java para o tipo de dados integer.

A tabela a seguir mostra como a transformação Java mapeia os tipos de dados da Developer tool do PowerCenter para tipos de dados primitivos e complexos Java:

Tipo de Dados do PowerCenter	Tipos de dados Java
Caractere	String
Binary	byte[]
Long (INT32)	int
Double	double
Decimal	double BigDecimal
BIGINT	long
Date/Time	BigDecimal long (número de milissegundos desde 1 de janeiro de 1970 00:00:00.000 GMT)

Tipo de dados da Developer Tool	Tipos de dados Java
array*	java.util.List
bigint	long
binary	byte[]
date/time	Com o processamento de nanossegundos habilitado, BigDecimal com precisão de nanossegundo Com o processamento de nanossegundos desabilitado, long com precisão de milissegundos (o número de milissegundos desde 1 de janeiro de 1970 00:00:00.000 GMT)
decimal	Com processamento de alta precisão desabilitado, double com precisão 15 Com processamento de alta precisão habilitado, BigDecimal
double	duplo
número inteiro	int
map*	java.util.Map
string	String

Tipo de dados da Developer Tool	Tipos de dados Java
struct*	Classe JavaBean personalizada com getters e setters para os elementos do campo struct
text	String
* Com suporte no mecanismo Spark.	

Em Java, os tipos de dados `java.util.List`, `java.util.Map`, `String`, `byte []` e `BigDecimal` são tipos de dados complexos. Os tipos de dados `double`, `int` e `long` são tipos de dados primitivos.

Na Developer tool, os tipos de dados array, struct e map são tipos de dados complexos.

A transformação Java configura valores nulos nos tipos de dados primitivos como zero. Você pode usar os métodos API `isNull` e `setNull` na guia **Na Linha de Entrada** para definir valores nulos na porta de entrada para valores nulos na porta de saída. Para ver um exemplo, consulte [“setNull” na página 255](#).

Nota: O tipo de dados decimal é mapeado para `BigDecimal` quando a alta precisão está ativada. O `BigDecimal` não pode ser usado com alguns operadores, como o operador `+`. Se o código Java contiver uma expressão que use uma porta decimal e a porta for usada com um dos operadores, o código Java não conseguirá compilar.

Nota: O tipo de dados decimal é mapeado para `BigDecimal` quando a alta precisão está ativada. O `BigDecimal` não pode ser usado com alguns operadores, como o operador `+`. Se o código Java contiver uma expressão que use uma porta decimal ou uma porta complexa com um elemento de um tipo de dados decimal e a porta for usada com um dos operadores, o código Java não conseguirá compilar.

Usando a Guia Código Java

Utilize a guia Código Java para definir, compilar e corrigir erros de compilação no código Java. Crie trechos de código nas guias de entrada de código.

Ao definir o código Java, você pode executar a seguinte tarefa:

- Definir um código ou um bloco estático, variáveis de instância e métodos definidos pelo usuário.
- Definir expressões Java e uma lógica de transformação.
- Utilizar os métodos API de transformação Java e construções de linguagens Java padrão.
- Importar Java APIs de terceiros, pacotes Java internos ou pacotes Java personalizados. Você pode usar trechos de código Java de pacotes Java.

Depois de desenvolver trechos de código, você poderá compilar o código Java e ver os resultados da compilação na janela Saída ou exibir o código Java completo.

A guia Código Java mantém os seguintes componentes:

- **Navegador.** Adicionar portas de entrada ou saída ou APIs a um trecho de código. O Navegador lista as portas de entrada e saída da transformação, as APIs de transformação Java disponíveis e uma descrição da porta ou da função API. Para as portas de entrada e saída, a descrição inclui o nome da porta, o tipo, o tipo de dados, a precisão e a escala. Para as funções API, a descrição inclui a sintaxe e a utilização da função API.

O Navegador desabilita qualquer porta ou função API que não esteja disponível para a guia de entrada de código. Por exemplo, você não pode adicionar portas ou funções API de chamada usando a guia de entrada de código Importar Pacotes.
- **Janela do código.** Desenvolva o código Java para a transformação. A janela do código usa o realce básico de sintaxe Java.
- **Guias de entrada de código.** Defina o comportamento da transformação. Cada guia de entrada de código tem uma janela de código associada. Para inserir o código Java de uma guia de entrada de código, clique na guia e grave o código Java na janela de código.
- **Link Definir Expressão.** Abre a caixa de diálogo Definir Expressão que você utiliza para criar expressões Java.
- **Link Configurações.** Abre a caixa de diálogo Configurações. Utilize a caixa de diálogo Configurações para definir a classpath para pacotes Java de terceiros e personalizados, habilitar a alta precisão para tipos de dados decimais e processar dados em menos de um segundo. O Cliente do PowerCenter inclui arquivos dentro da classpath quando compila o código Java.
- **Link Compilar.** Compila o código Java da transformação. A saída do compilador Java, inclusive mensagens de erro e informativas, aparece na janela Saída.
- **Link Código Completo.** Abre a janela Código Completo para exibir o código de classe completo da transformação Java. O código completo da transformação inclui o código Java das guias de entrada de código adicionadas ao modelo de classe de transformação Java.
- **Janela Saída.** Exibe os resultados da compilação da classe de transformação Java. Você pode clicar com o botão direito do mouse em uma mensagem de erro na janela Saída, para localizar o erro no trecho de código ou no código completo da classe de transformação Java na janela Código Completo. Você também pode clicar duas vezes em um erro na janela Saída para localizar a origem do erro.

Configurando portas

Uma transformação Java pode ter portas de entrada, portas de saída e portas de entrada/saída. Crie e edite grupos e portas na guia Portas. Você pode especificar valores padrão para as portas. Depois de adicionar portas a uma transformação, use os nomes de porta como variáveis nos snippets do código Java.

Criando grupos e portas

Ao criar uma transformação Java, um grupo de entrada e um grupo de saída são incluídos.

Uma transformação Java sempre tem um grupo de entrada e um grupo de saída. A transformação não é válida se tiver vários grupos de entrada ou de saída. É possível alterar nomes de grupo existentes digitando no cabeçalho do grupo. Se você excluir um grupo, é possível adicionar um novo grupo clicando no ícone Criar Grupo de Entrada ou Criar Grupo de Saída.

Ao criar uma porta, a ferramenta DesignerDeveloper a adiciona abaixo da linha ou grupo selecionado. Qualquer porta de entrada/saída exibida abaixo do grupo de entrada também faz parte do grupo de saída. Qualquer porta de entrada/saída exibida abaixo do grupo de saída também faz parte do grupo de entrada.

Definindo Valores Padrão para Portas

Você pode definir valores padrão para portas em uma transformação Java.

A transformação Java inicializa variáveis de porta com o valor padrão para a porta dependendo do tipo de dados da porta.

Portas de Entrada e Saída

A transformação Java inicializa o valor das portas de entrada desconectadas ou das portas de saída que não são atribuídas a um valor nos trechos de código Java.

A transformação Java inicializa as portas com base nos seguintes tipos de dados Java:

Tipo de dados primitivo

Se você definir um valor padrão para a porta que não seja igual a nulo, a transformação inicializará o valor da variável de porta com o valor padrão. Caso contrário, ela inicializará o valor da variável de porta como 0.

Tipo de dados complexo

Se você definir um valor padrão para a porta, a transformação criará um novo objeto e o inicializará com o valor padrão. Caso contrário, a transformação inicializará a variável de porta para NULL. Por exemplo, se você definir um valor padrão para a porta string, a transformação criará um novo objeto String e o inicializará com o valor padrão.

Nota: Se você acessar uma variável de porta de entrada com um valor nulo no código Java, uma exceção `NullPointerException` ocorrerá.

É possível ativar uma porta de entrada como uma chave de partição e uma chave de classificação e também é possível atribuir uma direção de classificação. O Serviço de Integração de Dados particiona e classifica os dados em cada partição com base na chave e na direção de classificação. A Chave de Partição e a Chave de Classificação são válidas quando o escopo de transformação é definido como Todas as Entradas.

Use as seguintes propriedades para o particionamento e a classificação de dados:

Chave de Partição

Porta de entrada que determina as linhas de dados a serem agrupadas na mesma partição.

Chave de Classificação

Porta de entrada que determina os critérios de classificação em cada partição.

Direção

Ordem ascendente ou descendente. O padrão é decrescente.

Portas de entrada/saída

A transformação Java trata as portas de entrada/saída como portas de passagem. Se você não configurar um valor para a porta no código Java da transformação, o valor de saída será o mesmo que o de entrada. A transformação Java inicializa o valor de uma porta de entrada/saída no mesmo que inicializa uma porta de entrada.

Se você configurar o valor de uma variável de porta para uma porta de entrada/saída no código Java, a transformação Java usará esse valor quando gerar uma linha de saída. Se você não configurar o valor de uma porta de entrada/saída, a transformação Java configura o valor da variável da porta como 0 para tipos de dados simples e NULL para os complexos, quando gerar uma linha de saída.

Configuração das Propriedades da Transformação Java

A transformação Java inclui propriedades para o código de transformação e a transformação. Se você criar uma transformação Java no Transformation Developer, pode substituir as propriedades da transformação quando usá-la em um mapeamento.

A tabela a seguir descreve as propriedades da transformação Java:

Propriedade	Descrição
Linguagem	Linguagem usada para o código de transformação. Não é possível alterar esse valor.
Nome de Classe	Nome da classe Java para a transformação. Não é possível alterar esse valor.
Nível de rastreamento	Quantidade de detalhes exibidos no log de sessão dessa transformação. Use os seguintes níveis de rastreamento: <ul style="list-style-type: none">- Conciso- Normal- Inicialização Detalhada- Dados Detalhados O padrão é Normal.
É Particionável	Várias partições em um pipeline podem usar essa transformação. Use estas opções: <ul style="list-style-type: none">- Não. A transformação não pode ser particionada. A transformação e outras transformações no mesmo pipeline estão limitadas a uma partição. Você deve escolher Não se a transformação processa todos os dados de entrada em conjunto, como limpeza de dados.- Localmente. A transformação pode ser particionada, mas o Serviço de Integração deve executar todas as partições da pipeline no mesmo nó. Escolha Localmente quando diferentes partições da transformação precisarem compartilhar objetos na memória.- Na Grade. A transformação pode ser particionada e o Serviço de Integração pode distribuir cada partição para nós diferentes. O padrão é Não.
Entradas Devem Bloquear	O procedimento associado à transformação deve ser capaz de bloquear os dados de entrada. O padrão é ativado.
Está Ativo	A transformação pode gerar mais de uma linha de saída para cada linha de entrada. Não é possível alterar essa propriedade depois de criar a transformação Java. Se você precisar alterar esta propriedade, crie uma nova transformação Java.
Transformação Estratégia de Atualização	A transformação define a estratégia de atualização das linhas de saída. Você pode ativar esta propriedade para transformações Java ativas. O padrão é desativado.
Escopo de Transformação	O método pelo qual o Serviço de Integração aplica a lógica de transformação aos dados de entrada. Use estas opções: <ul style="list-style-type: none">- Linha- Transação- Todas as Entradas Essa propriedade é sempre Linha para transformações passivas. O padrão é Todas as Entradas para transformações ativas.

Propriedade	Descrição
Gerar Transação	A transformação gera linhas de transação. Você pode ativar esta propriedade para transformações Java ativas. O padrão é desativado.
Saída é repetida	A ordem dos dados de saída é consistente entre as execuções de sessão. <ul style="list-style-type: none"> - Nunca. A ordem dos dados de saída é inconsistente entre as execuções de sessão. - Baseada na Ordem de Entrada. A ordem de saída é consistente entre as execuções de sessão quando a ordem dos dados de entrada é consistente entre elas. - Sempre. A ordem dos dados de saída será consistente entre as execuções de sessão mesmo se a ordem de entrada for inconsistente entre elas. O padrão é Nunca para transformações ativas. O padrão é Baseado na Ordem de Entrada para transformações passivas.
Requer Thread Único por Partição	Um único thread processa os dados para cada partição. Não é possível alterar esse valor.
Saída é Determinística	A transformação gera dados de saída consistentes entre as execuções de sessão. Ative esta propriedade para executar a recuperação em sessões que usam essa transformação. O padrão é ativado.

Aviso: Se você configurar uma transformação como repetível e determinística, será sua responsabilidade garantir que os dados sejam repetíveis e determinísticos. Se você tentar recuperar uma sessão com transformações que não gerem os mesmos dados entre a sessão e a recuperação, o processo de recuperação poderá resultar em dados corrompidos.

Trabalhando com o Controle de Transação

Você pode definir um controle de transação para uma transformação Java usando as seguintes propriedades:

- **Escopo de Transformação.** Determina como o Serviço de Integração aplica a lógica de transformação aos dados de entrada.
- **Gerar Transação.** Indica que o código Java para a transformação gera linhas de transação e as passa para o grupo de saída.

Escopo de Transformação

Você pode configurar a maneira como o Serviço de Integração aplica a lógica de transformação aos dados de entrada. Você pode escolher um destes valores:

- **Linha.** Aplica a lógica de transformação a uma linha de dados de cada vez. Escolha Linha quando os resultados da transformação dependerem de uma única linha de dados. Você deve escolher Linha para transformações passivas.
- **Transação.** Aplica a lógica de transformação a todas as linhas de uma transação. Escolha Transação quando os resultados da transformação dependerem de todas as linhas na mesma transação, mas não de linhas em outras transações. Por exemplo, você pode escolher Transação quando o código Java fizer cálculos agregados nos dados em uma única transação.

- **Todas as Entradas.** Aplica a lógica de transformação a todos os dados de entrada. Se você escolher Todos de Entrada, o Serviço de Integração descartará limites de transação. Escolha Todas as Entradas quando os resultados da transformação dependerem de todas as linhas de dados na origem. Por exemplo, você pode escolher Todos de Entrada quando o código Java da transformação classifica todos os dados recebidos.

Gerar Transação

Você pode definir o código Java em uma transformação Java ativa para gerar linhas de transação, como linhas de confirmação e reversão. Para gerar linhas de transação, ative a transformação para gerá-las.

Quando você configura a transformação para gerar as linhas de transação, o Serviço de Integração trata transformação Java como uma transformação Controle de Transação. A maioria das regras que se aplicam a uma transformação Controle de Transação em um mapeamento também se aplicam à transformação Java. Por exemplo, quando você configura uma transformação Java para gerar linhas de transação, não pode concatenar os pipelines ou ramificações de pipelines que contêm a transformação.

Quando você editar ou criar uma sessão usando uma transformação Java configurada para gerar linhas de transação, configure-a para a confirmação definida pelo usuário.

TÓPICOS RELACIONADOS:

- [“commit” na página 246](#)
- [“rollBack” na página 254](#)

Definindo a Estratégia de Atualização

Use uma transformação Java ativa para definir a estratégia de atualização de mapeamentos. Você pode definir a estratégia de atualização nestes níveis:

- **Dentro do código Java.** Você pode gravar o código Java para definir a estratégia de atualização das linhas de saída. O código Java pode sinalizar linhas para inserir, atualizar, excluir ou rejeitar. Para obter mais informações sobre configurações da estratégia de atualização, consulte [“setOutRowType” na página 255](#).
- **Dentro do mapeamento.** Use a transformação Java em um mapeamento para sinalizar linhas para inserção, atualização, exclusão ou rejeição. Selecione a propriedade da Transformação de Estratégia de Atualização para a transformação Java.
- **Dentro da sessão.** Configure a sessão para tornar as linhas de origem controladas por dados.

Se você não configurar a transformação Java para definir a estratégia de atualização ou não configurar a sessão como controlada por dados, o Serviço de Integração não usará o código Java para sinalizar as linhas de saída. Em vez disso, o Serviço de Integração indica as linhas de saída como inserção.

Desenvolvimento do Código Java

Use as guias de inserção para inserir snippets de código Java que definem a funcionalidade da transformação Java. Você pode escrever o código Java usando as guias de inserção de código para importar pacotes Java, escrever o código auxiliar, definir expressões Java e escrever o código Java que defina o comportamento da transformação para eventos específicos. Você pode desenvolver snippets nas guias de inserção de código em qualquer ordem.

Insira o código Java nas seguintes guias de entrada de código:

- **Importar Pacotes.** Importar pacotes Java de terceiros, internos ou personalizados.
- **Código Auxiliar.** Defina variáveis e métodos disponíveis para todas as guias, exceto Importar Pacotes.
- **Na Fila de Entrada.** Defina o comportamento da transformação quando ela recebe uma linha de entrada.
- **No Final dos Dados.** Defina o comportamento da transformação depois que ela processou todos os dados de entrada.
- **Na Transação de Recepção.** Defina o comportamento da transformação quando ela recebe uma notificação de transação. Use com transformações Java ativas.
- **Expressões Java.** Defina expressões Java para chamar expressões do PowerCenter. Você pode usar as expressões Java nas guias de entrada de Código Auxiliar, Na Linha de Entrada, No Final dos Dados e Na Transação.

Acesse os dados de entrada e de saída configurados na guia Na Linha de Entrada. Para as transformações ativas, você também pode configurar os dados de saída nas guias No Final dos Dados e Na Transação de Recepção.

Criando trechos de código Java

Para criar trechos de código Java para definir o comportamento da transformação, use a janela **CódigoCódigo Java** na entrada na guia **Código Java** da entrada de código.

1. Clique na guia apropriada de entrada de código.

A tabela a seguir descreve as tarefas que você pode executar nas guias de entrada do código no modo de exibição **Java**:

Guia	Descrição
Importações	Importa pacotes Java personalizados e integrados de outras empresas para uma transformação Java passiva ou ativa. Depois de importar os pacotes, você pode usá-los nas outras guias de entrada de código.
Auxiliares	Declara variáveis e métodos definidos pelo usuário para a classe de transformação Java em uma transformação Java passiva ou ativa. Depois de declarar variáveis e métodos, você poderá usá-los em qualquer outra entrada de código, exceto o guia Importações .
Na Entrada	Define como uma transformação Java passiva ou ativa se comporta quando recebe uma linha de entrada. O código Java que você define nesta guia é executado uma vez para cada linha de entrada. Nessa guia, você também pode acessar e usar os dados, as variáveis e os métodos API da transformação Java da porta de entrada e saída.
No Final	Define como uma transformação Java passiva ou ativa se comporta depois de processar todos os dados de entrada. Nessa guia, você também pode definir os dados de saída para transformações ativas e chamar métodos API de transformação Java.

Guia	Descrição
Funções	Define as funções que invocam expressões em uma transformação Java com a linguagem de programação Java. Por exemplo, você pode definir uma função que invoca uma expressão que procura os valores das portas de entrada ou saída ou que procura os valores de variáveis da transformação Java. Na guia Funções , você pode definir manualmente as funções ou clicar em Nova Função para invocar a caixa de diálogo Definir Função , que permite definir facilmente uma função.
Interfaces do Otimizador	Define a otimização de filtro de seleção antecipada ou de push-into. Selecione o método de otimização no navegador. Atualize os trechos de código para ativar a otimização. Defina as portas de entrada e as portas de saída associadas para passar a lógica de filtro.
Código Completo	Somente leitura. Nessa guia, você pode exibir e compilar o código de classe completo para a transformação Java.

- Para acessar as variáveis da coluna de entrada ou saída no trecho, clique duas vezes no nome da porta do navegador, expanda a lista **Entrada** ou **Saída** no navegador e clique duas vezes no nome da porta.
- Para chamar uma API de transformação Java no trecho, clique duas vezes no nome da API no navegador. Se necessário, configure os valores de entrada apropriados da API expanda a lista **APIs que podem ser chamadas** no navegador e clique duas vezes no nome do método. Se necessário, configure os valores de entrada adequados para o método.
- Grave o código Java apropriado com base no tipo de guia da entrada de código do trecho de código. Exiba o código de classe completo para a transformação Java na janela **Código Completo** **Código Java** na guia **Código Completo**.

Importando pacotes Java

Na guia **Importar Pacotes** **Importações**, você pode importar pacotes Java para transformações Java ativas ou passivas.

Você pode importar pacotes Java personalizados e integrados de outras empresas. Depois de importar pacotes Java, você poderá usar os pacotes importados nas outras guias de entrada de código.

Nota: Na guia **Importar Pacotes** **Importações**, você não pode declarar nem usar variáveis estáticas, variáveis de instância nem métodos de usuário.

Na ferramenta Developer do Cliente do PowerCenter, quando você exporta ou importa metadados que contêm uma transformação Java, os arquivos JAR ou de classe que contêm os pacotes personalizados ou de terceiros exigidos pela transformação com Java não são incluídos na exportação ou importação.

Se você importar metadados que contêm uma transformação Java, será necessário copiar os arquivos JAR ou de classe que contêm os pacotes personalizados ou de terceiros para o nó Serviço de Integração e Cliente do PowerCenter nó Data Integration Service e cliente da ferramenta Developer.

Por exemplo, para importar o pacote Java de E/S, digite o seguinte código na guia **Importar Pacotes** **Importações**:

```
import java.io.*;
```

Ao importar pacotes Java não padrão, adicione o pacote ou classe ao classpath nas Configurações da transformação Java.

TÓPICOS RELACIONADOS:

- [“Configurando definições da transformação Java” na página 240](#)

Definindo Passiva

Na guia **Código AuxiliarAuxiliares**, você pode declarar variáveis e métodos definidos pelo usuário para a classe de transformação Java da transformação Java ativa ou passiva.

Depois de declarar variáveis e métodos na guia **Código Auxiliar Auxiliares**, você poderá usar as variáveis e os métodos em qualquer guia de entrada de código, exceto as guias **Importar Pacotes** e **Importações**.

Na guia **Código AuxiliarAuxiliares**, você pode declarar os seguintes tipos de código, variáveis e métodos:

- Código estático e variáveis estáticas.

Em um bloco estático, você pode declarar variáveis estáticas e código estático. Todas as instâncias de uma transformação Java reutilizável em um mapeamento e todas as partições na sessão compartilham o código estático e as variáveis. O código estático é executado antes de qualquer outro código em uma transformação Java.

Por exemplo, o código a seguir declara uma variável estática para armazenar o limite de erro para todas as instâncias de uma transformação de Java em um mapeamento:

```
static int errorThreshold;
```

Use esta variável para armazenar o limite de erros para a transformação e acessá-la a partir de todas as instâncias da transformação Java em um mapeamento e a partir de qualquer partição de uma sessão.

Nota: É necessário sincronizar variáveis estáticas em uma sessão de partição múltipla ou em uma transformação Java reutilizável.

- Variáveis de instância.

Você pode declarar variáveis de instância de nível de partição. Várias instâncias de uma transformação Java reutilizável em um mapeamento ou várias partições em uma sessão não compartilham variáveis de instância. Declare as variáveis de instância com um prefixo para evitar conflitos e inicializar variáveis de instância não primitivas.

Por exemplo, o código a seguir utiliza uma variável booleana para declarar se uma linha de saída vai ou não ser gerada:

```
// boolean to decide whether to generate an output row
// based on validity of input
private boolean generateRow;
```

- Métodos de instância ou estáticos definidos pelo usuário.

Estende a funcionalidade da transformação Java. Os métodos Java declarados na guia **Código AuxiliarAuxiliares** podem usar ou modificar variáveis de saída ou variáveis de instância declaradas localmente. Você não pode acessar as variáveis de entrada a partir dos métodos Java na guia **Código AuxiliarAuxiliares**.

Por exemplo, use o seguinte código na guia **Código AuxiliarAuxiliares** para declarar uma função que adiciona dois inteiros:

```
private int myTXAdd (int num1,int num2)
{
    return num1+num2;
}
```

Guia Na Linha de Entrada

Use a guia Na Linha de Entrada para definir o comportamento da transformação com Java quando receber uma linha de entrada. O código Java nesta guia é executado uma vez para cada linha de entrada. É possível acessar dados de linha de entrada somente na guia Na Linha de Entrada.

Acesse e use os seguintes dados, variáveis e métodos de porta de entrada e de saída a partir da guia Na Linha de Entrada:

- **Variáveis da porta de entrada e de saída.** Acesse os dados da porta de entrada e de saída como uma variável, usando o nome da porta como o nome da variável. Por exemplo, se "in_int" é uma porta de entrada de Inteiro, é possível acessar os dados dessa porta referindo-se como uma variável "in_int" com o inteiro de tipo de dados primitivos Java. Não é necessário declarar as portas de entrada e de saída como variáveis.

Não atribua um valor a uma variável de porta de entrada. Caso você atribua um valor a uma variável de entrada na guia Na Linha de Valor, não será possível obter os dados de entrada para a porta correspondente na linha atual.

- **Variáveis de instância e métodos definidos pelo usuário.** Use qualquer instância ou variável estática ou método definido pelo usuário declarado na guia Código Auxiliar.

Por exemplo, uma transformação com Java tem duas portas de entrada, BASE_SALARY e BONUSSES, com um tipo de dados inteiro, e uma única porta de saída, TOTAL_COMP, com um tipo de dados inteiro. Você cria um método definido pelo usuário na guia Código Auxiliar, myTXAdd, que adiciona dois inteiros e retorna o resultado. Use o seguinte código Java na guia Na Linha de Entrada para atribuir os valores totais das portas de entrada às portas de saída e gerar uma linha de saída:

```
TOTAL_COMP = myTXAdd (BASE_SALARY, BONUSSES);  
generateRow();
```

Quando a transformação com Java recebe uma linha de entrada, ela adiciona o valor das portas de entrada BASE_SALARY e BONUSSES, atribui o valor à porta de saída TOTAL_COMP e gera uma linha de saída.

- **Métodos API da transformação Java.** Você pode chamar métodos de API fornecidos pela transformação com Java.

Guia No Final dos Dados

Use a guia No final dos dados nas transformações ativas ou passivas Java para definir o comportamento da transformação Java depois que ela processar todos os dados de entrada. Para gerar linhas de saída na guia No final dos dados, configure o escopo da transformação como Transação ou Todas as entradas. Não é possível acessar ou definir o valor das variáveis de porta de entrada nesta guia.

Acesse e use as seguintes variáveis e métodos da guia No final dos dados:

- **Variáveis da porta de saída.** Use os nomes de portas de saída como variáveis para acessar ou definir dados de saída para transformações Java ativas.
- **Variáveis de instância e métodos definidos pelo usuário.** Use quaisquer variáveis de instância ou métodos definidos pelo usuário que você declarou na guia Código Auxiliar.
- **Métodos API da transformação Java.** Chame métodos de API fornecidos pela transformação com Java. Use os métodos de API de confirmação e reversão para gerar uma transação.

Por exemplo, use o seguinte código Java para escrever as informações no log da sessão, quando o final dos dados for atingido:

```
logInfo("Number of null rows for partition is: " + partCountNullRows);
```

Guia Sobre a Recepção de Transações

Use a guia Sobre a Recepção de Transações nas transformações ativas com Java para definir o comportamento de uma transformação ativa com Java ao receber uma notificação de transação. O snippet de código da guia Sobre a Recepção de Transações somente é executado se o Escopo de Transação para a transformação estiver definido para Transação. Não é possível acessar ou definir o valor das variáveis de porta de entrada nesta guia.

Acesse e use os seguintes dados de saída, variáveis e métodos a partir da guia Sobre a Recepção de Transações:

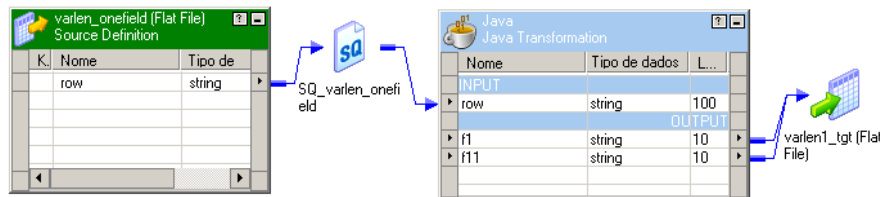
- **Variáveis da porta de saída.** Use os nomes de portas de saída como variáveis para acessar ou definir dados de saída.
- **Variáveis de instância e métodos definidos pelo usuário.** Use quaisquer variáveis de instância ou métodos definidos pelo usuário que você declarou na guia Código Auxiliar.
- **Métodos API da transformação Java.** Chame métodos de API fornecidos pela transformação com Java. Use os métodos de API de confirmação e reversão para gerar uma transação. Por exemplo, use o seguinte código de Java para gerar uma transação após a transformação receber uma transação:

```
commit();
```

Usando Código Java para Analisar um Arquivo Simples

Você pode desenvolver o código Java para analisar um arquivo plano. Use código Java para extrair colunas de dados específicas de um arquivo simples de esquemas variáveis ou uma mensagem JMS.

Por exemplo, você quer ler as duas primeiras colunas de dados de um arquivo simples delimitado. Crie um mapeamento que leia dados de um arquivo simples delimitado e passe dados para uma ou mais portas de saída. A figura a seguir mostra como a transformação Java lê linhas de dados e os passa para um destino:



O mapeamento contém os seguintes componentes:

- **Definição da origem.** A origem é um arquivo simples delimitado. Configure a origem para passar linhas de um arquivo simples para a transformação Java como strings. O arquivo de origem tem os seguintes dados:

```
1a, 2a, 3a, 4a, 5a, 6a, 7a, 8a, 9a, 10a
1b, 2b, 3b, 4b, 5b, 6b, 7b, 8b, 9b
1c, 2c, 3c, 4c, 5c, 6c, 7c
1d, 2d, 3d, 4d, 5d, 6d, 7d, 8d, 9d, 10d
```

- **Transformação Java.** Defina a funcionalidade da transformação Java na guia Código Java.

Use a guia Importar Pacotes em Código Java para importar o pacote Java. Insira o código a seguir na guia Importar Pacotes:

```
import java.util.StringTokenizer;
```

Use a guia Na Linha de Entrada em Código Java para ler cada linha de uma string e passar os dados para portas de saída. Insira o código a seguir na guia Na Linha de Entrada para recuperar as duas primeiras colunas de dados:

```
StringTokenizer st1 = new StringTokenizer(row, ",");  
f1 = st1.nextToken();  
f11= st1.nextToken();
```

- **Definição do destino.** Configure o destino para receber linhas de dados da transformação Java. Depois que você executar um fluxo de trabalho que contenha o mapeamento, o destino conterá duas colunas de dados. O arquivo de destino contém os seguintes dados:

```
1a,2a  
1b,2b  
1c,2c  
1d,2d
```

Configurando definições da transformação Java

É possível configurar as configurações das transformações com Java para definir o classpath para terceiros e pacotes de Java personalizados e habilitar grande precisão para tipos de dados decimais. Também é possível configurar a transformação para processar dados de subsegundos.

Configuração do Classpath

Quando você importar pacotes Java não padrão para a guia Importar pacote, configure o classpath para o Cliente do PowerCenter e o Serviço de Integração para cada arquivo JAR ou diretório de arquivo de classe associado ao pacote Java. No UNIX, use dois-pontos para separar as entradas de classpath. No Windows, use ponto-e-vírgula para separar as entradas de classpath. Os arquivos JAR ou de classe devem ser acessíveis no Cliente do PowerCenter e no nó do Serviço de Integração.

Por exemplo, se você importar o `converter` do pacote Java na guia Importar pacotes e define o pacote no `converter.jar`. Você deve adicionar o `converter.jar` ao classpath antes de compilar o código Java para a transformação Java.

Você não precisa configurar o classpath para os pacotes Java incorporados. Por exemplo, `java.io` é um pacote Java incorporado. Se você importar `java.io`, não precisará configurar o classpath para o `java.io`.

Configuração do Classpath para o Serviço de Integração

Você pode adicionar arquivos JAR ou diretórios de arquivo de classe ao classpath do Serviço de Integração. Para configurar o classpath no nó do Serviço de Integração, conclua uma das tarefas a seguir:

- **Configure a propriedade da sessão do Classpath Java.** Configure o classpath usando a propriedade da sessão do Classpath Java. Este classpath se aplica à sessão.
- **Configure o Classpath Java SDK.** Configure o Classpath Java SDK na guia Processos das propriedades do Serviço de Integração no Console de Administração. Esta configuração se aplica a todas as sessões executadas no Serviço de Integração.
- **Configure a variável do ambiente CLASSPATH.** Configure a variável do ambiente CLASSPATH no nó do Serviço de Integração. Reinicie o Serviço de Integração depois de configurar a variável de ambiente. Aplica-se a todas as sessões executadas no Serviço de Integração.

Configurando o Classpath para o Serviço de Integração em UNIX

Para configurar a variável de ambiente do CLASSPATH em UNIX:

- ▶ Em um ambiente de shell C de UNIX, digite:

```
setenv CLASSPATH <classpath>
```

Em um ambiente de shell Bourne de UNIX, digite:

```
CLASSPATH = <classpath>  
export CLASSPATH
```

Configurando o Classpath para o Serviço de Integração no Windows

Para configurar a variável de ambiente do CLASSPATH no Windows:

- ▶ Insira a variável do ambiente CLASSPATH e defina o valor para o classpath padrão.

Configurando o Classpath para a ferramenta Developer do PowerCenter

Você pode adicionar arquivos JAR ou diretórios de arquivo de classe ao classpath da ferramenta Developer do Cliente do PowerCenter.

Para definir o classpath para a máquina em que a ferramenta Developer do Cliente do PowerCenter é executada, conclua uma das seguintes tarefas:

- Configure a variável do ambiente CLASSPATH. Defina a variável de ambiente CLASSPATH na máquina cliente da ferramenta Developer do Cliente do PowerCenter. Isto se aplica a todos os processos Java executados na máquina.
- Para uma transformação Java não reutilizável, Configure o classpath nas propriedades avançadas das configurações da transformação Java. Isso se aplica a mapeamentos de sessões que incluem essa transformação Java. O Cliente do PowerCenter cliente da Developer tool inclui arquivos dentro do classpath ao compilar o código Java.

Para adicionar arquivos JAR ou diretórios de arquivo de classe ao classpath em uma transformação Java, conclua as seguintes etapas:

1. Na guia **Código Java Avançadas**, clique no link **Configurações** e clique no ícone de seta para baixo na coluna **Valor** ao lado de **Classpath**.
A caixa de diálogo **Configurações Editar Classpath** é exibida.
2. Para adicionar um classpath, conclua as seguintes etapas:
 - a. Clique em **Adicionar**.
A janela **Salvar Como** é exibida.
 - b. Na janela **Salvar Como**, navegue até o diretório em que o arquivo JAR está localizado.
 - c. Clique em **OK**.
O classpath é exibido na caixa de diálogo **Editar Classpath**.
3. Clique em Procurar em **Adicionar Classpath** para selecionar o arquivo JAR ou o diretório de arquivo de classe para o pacote importado. Clique em **OK**.
4. Clique em **Adicionar**.
O diretório JAR ou de arquivo de classe é exibido na lista de diretórios JAR e de arquivo de classe para a transformação.
5. Para remover um arquivo JAR ou diretório de arquivo de classe, selecione o diretório JAR ou de arquivo de classe e clique **Remover**.
O diretório desaparece da lista de diretórios.

Habilitando Alta Precisão

Por padrão, a transformação Java converte portas do tipo Decimal em tipos de dados duplos com uma precisão de 15. Se você deseja processar um tipo de dados Decimal com uma precisão maior que 15, ative a precisão alta para processar portas decimais, com a classe Java BigDecimal.

Quando você ativa a precisão alta, pode processar portas Decimais com precisão menor que 28 como o BigDecimal. A transformação Java converte os dados decimais com uma precisão maior que 28 para o tipo de dados Duplo. As expressões da transformação Java processam dados binários, inteiros, duplos e de string. As expressões da transformação Java não podem processar dados bigint.

Por exemplo, uma transformação Java tem uma porta de entrada do tipo Decimal que recebe um valor de 40012030304957666903. Se você ativar a precisão alta, o valor da porta é tratado conforme o exibido. Se você não ativar a precisão alta, o valor da porta é $4,00120303049577 \times 10^{19}$.

Nota: Se você ativar a precisão alta para uma transformação Java, ative-a também para a sessão que inclui a transformação Java. Se a precisão alta for ativada na Transformação Java e não na sessão, a sessão pode falhar com o erro a seguir:

```
[ERROR]Failed to bind column with index <index number> to datatype <datatype name>.
```

Processando Subsegundos

Você pode processar dados por subsegundo até nanossegundos no código Java. Quando você configura as definições para usar nanossegundos em valores de data e hora, o código Java gerado converte o tipo de dados Data e Hora da transformação em BigDecimal Java, que tem precisão de nanossegundos.

Por padrão, o código Java gerado converte o tipo de dados Data e Hora da transformação em Longo Java, que tem precisão de milissegundos.

Compilando uma transformação Java

A ferramenta Developer do Cliente do PowerCenter usa o compilador Java para compilar o código Java e gerar o código de bytes para a transformação.

O compilador Java compila o código Java e exibe os resultados da compilação na janela **Saída Resultados** nas propriedades **Compilação** nas guias de entrada de código. O compilador Java é instalado com a ferramenta Developer do Cliente do PowerCenter no diretório `java/bin`.

Para compilar o código completo para a transformação Java, clique em **Compilar** nas propriedades **Compilação** da guia **Código JavaCódigo Completo**.

Quando você cria uma transformação Java, ela contém uma classe Java que define a funcionalidade básica para uma transformação Java. O código completo da classe Java contém o código da classe de modelo para a transformação, além do código Java que você define nas guias de entrada de código.

Quando você compila uma transformação Java, a ferramenta Developer do Cliente do PowerCenter adiciona o código das guias de entrada de código à classe do modelo para a transformação a fim de gerar o código de classe completo para a transformação. Em seguida, a ferramenta Developer do Cliente do PowerCenter chama o compilador Java para compilar o código de classe completo. O compilador Java compila a transformação e gera o código de bytes para ela.

Os resultados da compilação são mostrados na janela **SaídaResultados**. Use os resultados da compilação para identificar e localizar erros de código Java.

Nota: A transformação Java também é compilada quando você clica em **OK** na transformação.

Corrigindo erros de compilação

Você pode identificar erros do código Java e localizar a origem desses erros para uma transformação Java na janela Saída. Os erros da transformação Java podem ocorrer como resultado de um erro na guia de entrada do código, ou de um erro no código completo para a classe da transformação Java.

Para solucionar os problemas de uma transformação Java:

- **Localize a origem do erro.** Você pode localizar a origem do erro no código do snippet Java, ou no código de classe completo para a transformação.
- **Identifique o tipo de erro.** Use os resultados da compilação na janela de saída e o local do erro, para identificar o tipo de erro.

Depois que identificar a origem e o tipo do erro, corrija o código Java na guia de entrada do código e compile a transformação novamente.

Localizando a Origem de Erros de Compilação

Ao compilar uma transformação de Java, a janela Saída exibe os resultados da compilação. Use os resultados da compilação para identificar os erros de compilação. Ao usar a janela Saída para localizar a origem de um erro, o Cliente do PowerCenter destaca a origem do erro em uma guia de entrada de códigos ou na janela Código Completo.

É possível localizar erros na janela Código Completo, mas não é possível editar códigos Java nessa mesma janela. Para corrigir erros localizados na janela Código Completo, modifique o código na guia de entrada de códigos apropriada. Pode ser necessário usar a janela Código Completo para visualizar erros ocasionados pela adição de códigos de usuário ao código de classe completa da transformação.

Use os resultados da compilação na janela Saída para identificar erros nos locais a seguir:

- Guias de entrada de código
- Janela Código Completo

Localizando erros nas guias Entrada de Código

Para localizar a origem de um erro nas guias de entrada de código, clique com o botão direito no erro na janela Saída e escolha Visualizar Erro no trecho de código, ou clique duas vezes no erro na janela Saída. O Cliente do PowerCenter destaca a origem do erro na guia de entrada de código apropriada.

Localizando erros na janela Código Completo

Para localizar a origem dos erros na janela Código completo, clique com o botão direito no erro na janela Saída e escolha Exibir erro no código completo, ou clique duas vezes no erro na janela Saída. O cliente PowerCenter realça a origem do erro na janela Código completo.

Identificando a origem de erros de compilação

Os erros de compilação podem surgir como resultado de erros no código do usuário.

Erros no código do usuário também podem gerar um erro no código de não usuário da classe. Erros de compilação ocorrem no código do usuário e não usuário da transformação Java.

Erros de Código do Usuário

Podem ocorrer erros no código do usuário nas guias de entrada de código. Os erros de código do usuário incluem erros de idioma e sintaxe Java padrão.

Também podem ocorrer erros de código do usuário quando a ferramenta Developer do Cliente do PowerCenter do adiciona o código do usuário das guias de entrada de código ao código de classe completo.

Por exemplo, uma transformação Java tem uma porta de entrada com o nome int1 e um tipo de dados inteiro. O código completo da classe declara a variável da porta de entrada com este código:

```
int int1;
```

No entanto, se você usar o mesmo nome da variável na guia **Na Linha de EntradaNa Entrada**, o compilador Java emitirá um erro de redeclaração de uma variável. Para corrigir o erro, renomeie a variável na guia **Na Linha de EntradaNa Entrada**.

Erros de código de não usuário

O código do usuário nas guias de entrada de código pode causar erros no código de não usuário.

Por exemplo, uma transformação Java tem uma porta de entrada e uma de saída, int1 e out1, com tipos de dados inteiro. Grave o seguinte código na guia de entrada de código **Na Linha de EntradaNa Entrada** para calcular os juros da porta de entrada int1 e atribuí-los à porta de saída out1:

```
int interest;  
interest = CallInterest(int1); // calculate interest  
out1 = int1 + interest;  
}
```

Ao compilar a transformação, a ferramenta Developer do Cliente do PowerCenter adiciona o código da guia de entrada de código **Na Linha de EntradaNa Entrada** ao código de classe completo da transformação. Quando o compilador Java compila o código Java, a chave incompatível causa a terminação precoce de um método no código de classe completo e o compilador Java emite um erro.

CAPÍTULO 13

Referência da API da transformação Java

Este capítulo inclui os seguintes tópicos:

- [Visão geral dos métodos API da transformação Java, 245](#)
- [commit, 246](#)
- [defineJExpression, 247](#)
- [failSession, 248](#)
- [generateRow, 248](#)
- [getInRowType, 249](#)
- [getMetadata, 249](#)
- [incrementErrorCount, 250](#)
- [invokeJExpression, 251](#)
- [isNull, 252](#)
- [logError, 252](#)
- [logInfo, 253](#)
- [resetNotification, 253](#)
- [rollBack, 254](#)
- [setNull, 255](#)
- [setOutRowType, 255](#)
- [storeMetadata, 256](#)

Visão geral dos métodos API da transformação Java

Na guia **Código Java** de uma transformação Java nas guias de entrada de código do modo de exibição **Javano** Editor, você pode adicionar métodos API ao código Java para definir o comportamento da transformação.

Para adicionar um método API ao código, expanda a lista **APIs que podem ser chamadas** no navegador da guia de entrada de código e clique duas vezes no nome do método que você deseja adicionar ao código.

Como alternativa, você pode arrastar o método do navegador até o trecho de código Java ou inserir manualmente o método API no trecho de código Java.

Você pode adicionar os seguintes métodos API ao código Java em uma transformação Java:

commit

Gera uma transação.

defineJExpression

Define uma expressão Java.

failSession

Lança uma exceção com uma mensagem de erro e apresenta falha no mapeamento da sessão.

generateRow

Gera uma linha de saída para transformações Java ativas.

getInRowType

Retorna o tipo de entrada da linha atual na transformação.

incrementErrorCount

Incrementa a contagem de erros do mapeamento da sessão.

invokeJExpression

Invoca uma expressão Java que você definiu usando o método `defineJExpression`.

isNull

Verifica se há um valor nulo em uma coluna de entrada.

logError

Grava uma mensagem de erro no log da sessão.

logInfo

Grava uma mensagem com informações no log da sessão.

resetNotification

Se a máquina do Data Integration Service estiver sendo executada no modo de reinicialização, redefine as variáveis que você usa no código Java depois de executar um mapeamento.

rollback

Gera uma transação de reversão.

setNull

Define como nulo o valor de uma coluna de saída em uma transformação Java ativa ou passiva.

setOutRowType

Define a estratégia de atualização de linhas de saída. Pode sinalizar linhas para inserção, atualização ou exclusão.

commit

Gera uma transação.

Use confirmação em qualquer guia, exceto as guias de entrada de código Importar Pacotes e Expressões Java. Só é possível usar confirmação em transformações ativas configuradas para gerar transações. Se

você usar confirmação em uma transformação ativa não configurada para gerar transações, o Serviço de Integração gerará um erro e a sessão será encerrada.

Utilize a seguinte sintaxe:

```
commit();
```

Use seguinte o código Java para gerar uma transação para cada 100 linhas processadas por uma transformação Java, e então configure o contador rowsProcessed como 0:

```
if (rowsProcessed==100) {  
    commit();  
    rowsProcessed=0;  
}
```

defineJExpression

Define uma expressão, incluindo a string da expressão e os parâmetros de entrada. Argumentos para o método defineJExpression incluem uma matriz de objetos JExprParamMetadata que contém os parâmetros de entrada e um valor de string que define a sintaxe de expressão.

Utilize a seguinte sintaxe:

```
defineJExpression(  
    String expression,  
    Object[] paramMetadataArray  
);
```

A tabela a seguir descreve os parâmetros:

Parâmetro	Tipo	Tipo de dados	Descrição
expressão	Entrada	String	String que representa a expressão.
paramMetadataArray	Entrada	Objeto	Matriz de objetos JExprParamMetadata que contém os parâmetros de entrada da expressão.

Você pode adicionar o método defineJExpression ao código Java em qualquer guia de entrada de código, exceto as guias **Importações** e **Funções**.

Para usar o método defineJExpression, é necessário representar uma matriz de objetos JExprParamMetadata que representam os parâmetros de entrada da expressão. Defina os valores de metadados para os parâmetros e transfira a matriz como um parâmetro para o método de defineJExpression.

Por exemplo, o seguinte código Java cria uma expressão para consultar o valor de duas strings:

```
JExprParamMetadata params[] = new JExprParamMetadata[2];  
params[0] = new JExprParamMetadata(EDataType.STRING, 20, 0);  
params[1] = new JExprParamMetadata(EDataType.STRING, 20, 0);  
defineJExpression(":lkp.mylookup(x1,x2)",params);
```

Nota: Você deve numerar os parâmetros transferidos para a expressão consecutivamente e iniciar os parâmetros com a letra x. Por exemplo, para transferir três parâmetros para uma expressão, nomeie os parâmetros x1, x2 e x3.

failSession

Lança uma exceção com uma mensagem de erro e apresenta falha no mapeamento da sessão.

Utilize a seguinte sintaxe:

```
failSession(String errorMessage);
```

A tabela a seguir descreve o parâmetro:

Parâmetro	Tipo de Parâmetro	Tipo de dados	Descrição
errorMessage	Entrada	String	String de mensagem de erro.

Use o método `failSession` para finalizar o mapeamento da sessão. Não use o método `failSession` em um bloco `try/catch` em uma guia de entrada de código.

Você pode adicionar o método `failSession` ao código Java em qualquer guia de entrada de código, exceto as guias **Importações**, **Importar Pacotes** e **Funções Expressões Java**.

O código Java a seguir mostra como testar a porta de entrada `input1` para um valor nulo e como encerrar o mapeamento da sessão se for nulo:

```
if(isNull("input1")) {  
    failSession("Cannot process a null value for port input1.");  
}
```

generateRow

Gera uma linha de saída para transformações Java ativas.

Utilize a seguinte sintaxe:

```
generateRow();
```

Ao chamar o método `generateRow`, a transformação Java gera uma linha de saída usando o valor atual das variáveis da porta de saída. Se desejar gerar linhas múltiplas que correspondam a uma linha de entrada, você poderá chamar o método `generateRow` mais de uma vez para cada linha de entrada. Se você não usar o método `generateRow` em uma transformação Java ativa, a transformação não gerará linhas de saída.

Você pode adicionar o método `generateRow` ao código Java em qualquer guia de entrada de código, exceto as guias **Importações**, **Importar Pacotes** e **Funções Expressões Java**.

Você pode chamar o método `generateRow` somente em transformações ativas. Se você chamar o método `generateRow` em uma transformação passiva, o `Data Integration Service` gerará um erro.

Use o seguinte código Java para gerar uma linha de saída, modificar os valores das portas de saída e gerar outra linha de saída:

```
// Generate multiple rows.  
if(!isNull("input1") && !isNull("input2"))  
{  
    output1 = input1 + input2;  
    output2 = input1 - input2;  
}  
generateRow();  
// Generate another row with modified values.  
output1 = output1 * 2;
```



```
output2 = output2 * 2;  
generateRow();
```

getInRowType

Retorna o tipo de entrada da linha atual na transformação. O método retorna um valor de inserção, atualização, exclusão ou rejeição.

Utilize a seguinte sintaxe:

```
rowType getInRowType();
```

A tabela a seguir descreve o parâmetro:

Parâmetro	Tipo de Parâmetro	Tipo de dados	Descrição
rowType	Saída	String	Retorna o tipo de estratégia de atualização, que é um dos seguintes valores: <ul style="list-style-type: none">- DELETE- INSERT- REJECT- UPDATE

Você pode adicionar o método `getInRowType` ao código Java na guia **Na entradaNa Linha de Entrada** de entrada de código.

Você só pode utilizar o método `getInRowType` em transformações ativas configuradas para definir a estratégia de atualização. Se você chamar esse método em uma transformação ativa que não está configurada para definir a estratégia de atualização, a sessão Data Integration Service gerará um erro.

Use o seguinte código Java para concluir as seguintes ações:

- Propagar o tipo de linha de entrada para a linha de saída.
- Se o valor da porta de entrada `input1` for maior que 100, defina o tipo de linha de saída como DELETE.

```
// Set the value of the output port.  
output1 = input1;  
// Get and set the row type.  
String rowType = getInRowType();  
setOutRowType(rowType);  
// Set row type to DELETE if the output port value is > 100.  
if(input1 > 100  
    setOutRowType(DELETE);
```

getMetadata

Recupera os metadados de transformação de Java no tempo de execução. O método `getMetadata` recupera os metadados que você salva com o método `storeMetadata`, tais como uma condição de filtro que o otimizador passa para a transformação de Java na função `pushFilter`.

Utilize a seguinte sintaxe:

```
getMetadata (String key);
```

A tabela a seguir descreve os parâmetros:

Parâmetro	Tipo de Parâmetro	Tipo de dados	Descrição
chave	Entrada	String	Identifica os metadados. O método <code>getMetadata</code> usa a chave para determinar quais metadados recuperar.

Você pode adicionar o método `getMetadata` ao código Java nas seguintes guias de entrada de código:

- Auxiliar
- Na Entrada
- No Final
- Interfaces do Otimizador
- Funções

Você pode configurar o método `getMetadata` para recuperar as condições de filtro para otimização de push-into. O método `getMetadata` pode recuperar cada condição de filtro que você armazenar no otimizador.

```
// Retrieve a filter condition
String mydata = getMetadata ("FilterKey");
```

incrementErrorCount

Incrementa a contagem de erros da sessão. Se a contagem de erros atingir o limite de erros para a sessão, o mapeamento da sessão apresentará falha.

Utilize a seguinte sintaxe:

```
incrementErrorCount(int nErrors);
```

A seguinte tabela descreve o parâmetro:

Parâmetro	Tipo de Parâmetro	Tipo de dados	Descrição
nErrors	Entrada	Número inteiro	Número para incrementar a contagem de erros da sessão.

Você pode adicionar o método `incrementErrorCount` para o código Java em qualquer guia de entrada de código, exceto as guias **Importações**, **Importar Pacotes** e **Funções**, **Expressões Java**.

O seguinte código Java mostra como incrementar a contagem de erros caso uma porta de entrada de uma transformação tenha um valor nulo:

```
// Check if input employee id and name is null.
if (isNull ("EMP_ID_INP") || isNull ("EMP_NAME_INP"))
{
    incrementErrorCount(1);
    // if input employee id and/or name is null, don't generate a output row for this
    input row
    generateRow = false;
}
```

invokeJExpression

Invoca uma expressão e retorna o valor para a expressão.

Utilize a seguinte sintaxe:

```
(datatype) invokeJExpression(  
    String expression,  
    Object[] paramMetadataArray);
```

Os parâmetros de entrada para o método invokeJExpression são um valor de string que representa a expressão e uma matriz de objetos que contém os parâmetros de entrada da expressão.

A tabela a seguir descreve os parâmetros:

Parâmetro	Tipo de Parâmetro	Tipo de dados	Descrição
expressão	Entrada	String	String que representa a expressão.
paramMetadataArray	Entrada	Objeto[]	Matriz de objetos que contém os parâmetros de entrada da expressão.

Você pode adicionar o método de invokeJExpression ao código Java em qualquer guia de entrada de código, exceto as guias **Importações** e **Funções** **Importar Pacotes** e **Expressões Java**.

Use as seguintes regras e diretrizes ao usar o método invokeJExpression:

- Tipo de dados de retorno. O tipo de dados de retorno do método invokeJExpression é um objeto. É necessário moldar o valor de retorno da função com o tipo de dados apropriado.
É possível retornar valores com os tipos de dados Inteiro, Double, String e byte[].
- Tipo de linha. O tipo de linha para valores de retorno do método invokeJExpression é INSERT.
Para usar um tipo de linha diferente para o valor de retorno, use a interface avançada.
- Valores nulos. Se você transferir um valor nulo como um parâmetro ou se o valor de retorno para o método invokeJExpression for NULL, o valor será tratado como um indicador nulo.
Por exemplo, se o valor de retorno de uma expressão for NULL e o tipo de dados retornado for String, uma string será retornada com um valor de nulo.
- Tipo de dados Data. Você deve converter os parâmetros de entrada com um tipo de dados Data para o tipo de dados String.
Para usar a string em uma expressão como tipo de dados Data, utilize a função to_date() para converter a string nesse tipo de dados.
Além disso, é necessário moldar o tipo de retorno de qualquer expressão que retorne um tipo de dados Data como uma String.

O exemplo a seguir concatena strings " John " e " Smith " e retorna a string " Paulo Silva ":

```
(String) invokeJExpression("concat(x1,x2)", new Object [] { "John ", "Smith" });
```

Nota: Você deve numerar os parâmetros que você transfere para a expressão consecutivamente e iniciar o parâmetro com a letra x. Por exemplo, para transferir três parâmetros para uma expressão, nomeie os parâmetros x1, x2 e x3.

isNull

Verifica o valor de uma coluna de entrada para um valor nulo.

Utilize a seguinte sintaxe:

```
Boolean isNull(String strColName);
```

A tabela a seguir descreve o parâmetro:

Parâmetros	Tipo de Parâmetro	Tipo de dados	Descrição
strColName	Entrada	String	Nome de uma coluna de entrada.

Você pode adicionar o método isNull ao código Java na guia de entrada de código **Na Entrada Na Linha de Entrada**.

O código Java a seguir mostra como verificar se o valor da coluna de entrada SALARY é NULL antes de adicioná-lo à variável de instância totalSalaries:

```
// if value of SALARY is not null
if (!isNull("SALARY")) {
    // add to totalSalaries
    TOTAL_SALARIES += SALARY;
}
```

Como alternativa, use o seguinte código Java para obter os mesmos resultados:

```
// if value of SALARY is not null
String strColName = "SALARY";
if (!isNull(strColName)) {
    // add to totalSalaries
    TOTAL_SALARIES += SALARY;
}
```

logError

Grava uma mensagem de erro no log da sessão.

Utilize a seguinte sintaxe:

```
logError(String msg);
```

A tabela a seguir descreve o parâmetro:

Parâmetro	Tipo de Parâmetro	Tipo de dados	Descrição
msg	Entrada	String	String de mensagem de erro.

Você pode adicionar o método logError ao código Java em qualquer guia de entrada de código, exceto as guias **Importações****Importar Pacotes** e **Funções****Expressões Java**.

O código Java a seguir mostra como registrar um erro quando a porta de entrada for nula:

```
// check BASE_SALARY
if (isNull("BASE_SALARY")) {
```

```
        logError("Cannot process a null salary field.");
    }
}
```

Quando o código é executado, a seguinte mensagem é exibida no log da sessão:

```
[JTX_1013] [ERROR] Cannot process a null salary field.
```

logInfo

Grava uma mensagem com informações no log da sessão.

Utilize a seguinte sintaxe:

```
logInfo(String msg);
```

A tabela a seguir descreve o parâmetro:

Parâmetro	Tipo de Parâmetro	Tipo de dados	Descrição
msg	Entrada	String	String de mensagem de informações.

Você pode adicionar o método `logInfo` ao código Java em qualquer guia de entrada de código, exceto as guias **Importações**, **Importar Pacotes** e **Funções Expressões Java**.

O seguinte código Java mostra como gravar uma mensagem no log da sessão depois que a transformação Java processar um limite de 1000 linhas para a mensagem:

```
if (numRowsProcessed == messageThreshold) {
    logInfo("Processed " + messageThreshold + " rows.");
}
```

resetNotification

Se a máquina do Data Integration Service estiver sendo executada no modo de reinicialização, redefina as variáveis que você usa no código Java depois de executar um mapeamento.

No modo de reinicialização, o Serviço de Integração de Dados não está desligado, mas é redefinido após uma solicitação, de forma que possa processar a próxima solicitação.

Para uma transformação Java, use o método `resetNotification` para redefinir as variáveis no código Java depois de executar um mapeamento.

Utilize a seguinte sintaxe:

```
public int resetNotification(IGroup group) {
    return EStatus.value;
}
```

A tabela a seguir descreve os parâmetros:

Parâmetro	Tipo de Parâmetro	Tipo de dados	Descrição
inteiro	Saída	Estatus. <i>Valor</i>	Valor de retorno, onde <i>valor</i> é um dos seguintes valores: <ul style="list-style-type: none">- SUCESSO Sucesso.- FAKHA Falha.- NOIMPL. Não implementado.
grupo	Entrada	IGroup	O grupo de entrada.

Você pode adicionar o método `resetNotification` ao código Java na guia de entrada de código na guia **Ajuda**.

O método `resetNotification` não aparece na lista de APIs de chamada (callable).

Por exemplo, supondo que o código Java declara uma variável estática chamada `out5_satatic` e a inicializa como 1, o seguinte código Java redefine a variável `out5_static` como 1, depois da próxima execução de mapeamento:

```
public int resetNotification(IGroup group) {  
    out5_static=1;  
    return EStatus.SUCCESS;  
}
```

Esse método não é necessário. No entanto, se o Data Integration Service é executado no modo de reinicialização e o mapeamento contém uma transformação Java que não implementa o método `resetNotification`, a mensagem de aviso JSDK_42075 é exibida no log.

rollBack

Gera uma transação de reversão.

Use `rollBack` em qualquer guia, exceto as guias de entrada de código Importar Pacotes ou Expressões Java. Só é possível usar reversão em transformações ativas configuradas para gerar transações. Se você usar reversão em uma transformação ativa não configurada para gerar transações, o Serviço de Integração gerará um erro e a sessão será encerrada.

Utilize a seguinte sintaxe:

```
rollBack();
```

Use o código a seguir para gerar uma transação de reversão e encerrar a sessão se uma linha de entrada tiver uma condição ilegal ou gerar uma transação, caso o número de linhas processadas for 100:

```
// If row is not legal, rollback and fail session.  
  
if (!isRowLegal()) {  
    rollback();  
    failSession("Cannot process illegal row.");  
} else if (rowsProcessed==100) {  
    commit();  
    rowsProcessed=0;  
}
```

setNull

Define o valor de uma coluna de saída como nulo em uma transformação Java passiva ou ativa.

Utilize a seguinte sintaxe:

```
setNull(String strColName);
```

A tabela a seguir descreve o parâmetro:

Parâmetro	Tipo de Parâmetro	Tipo de dados	Descrição
strColName	Entrada	String	Nome de uma coluna de saída.

O método setNull define como nulo o valor de uma coluna de saída em uma transformação Java passiva ou ativa. Depois de definir uma coluna de saída como nula, você não poderá modificar o valor até gerar uma linha de saída.

Você pode adicionar o método setNull ao código Java em qualquer guia de entrada de código, exceto as guias **Importações**, **Importar Pacotes** e **Funções**, **Expressões Java**.

O seguinte código Java mostra como verificar o valor de uma coluna de entrada e define o valor correspondente de uma coluna de saída como nulo:

```
// check value of Q3RESULTS input column
if(isNull("Q3RESULTS")) {
    // set the value of output column to null
    setNull("RESULTS");
}
```

Como alternativa, você pode usar o seguinte código Java para obter os mesmos resultados:

```
// check value of Q3RESULTS input column
String strColName = "Q3RESULTS";
if(isNull(strColName)) {
    // set the value of output column to null
    setNull(strColName);
}
```

setOutRowType

Define a estratégia de atualização de linhas de saída. O método setOutRowType pode sinalizar as linhas para inserção, atualização ou exclusão.

Você só pode usar setOutRowType na guia de entrada do código Na Linha de Entrada. Você só pode utilizar setOutRowType em transformações ativas configuradas para definir a estratégia de atualização. Se você usar setOutRowType em uma transformação ativa que não esteja configurada para definir a estratégia de atualização, a sessão gerará um erro e falhará.

Utilize a seguinte sintaxe:

```
setOutRowType(String rowType);
```

A tabela a seguir descreve o argumento para este método:

Argumento	Tipo de Dados	Entrada/Saída	Descrição
rowType	String	Entrada	Tipo de estratégia de atualização. O valor pode ser INSERT, UPDATE ou DELETE.

Utilize o código Java a seguir para propagar o tipo de entrada da linha atual se o tipo de linha for UPDATE ou INSERT e o valor da porta de entrada input1 for menor que 100 ou defina o tipo de saída como DELETE se o valor de input1 for maior que 100:

```
// Set the value of the output port.  
output1 = input1;  
  
// Get and set the row type.  
String rowType = getInRowType();  
setOutRowType(rowType);  
  
// Set row type to DELETE if the output port value is > 100.  
if(input1 > 100)  
    setOutRowType(DELETE);
```

storeMetadata

Armazena os metadados de transformação de Java que você pode recuperar no tempo de execução com o método `getMetadata`.

Utilize a seguinte sintaxe:

```
storeMetadata (String key String data);
```

A tabela a seguir descreve os parâmetros:

Parâmetro	Tipo de Parâmetro	Tipo de dados	Descrição
chave	Entrada	String	Identifica os metadados. O método <code>storeMetadata</code> requer uma chave para identificar os metadados. Defina a chave como qualquer string.
dados	Entrada	String	Os dados que você deseja armazenar como metadados de transformação de Java.

Você pode adicionar o método `storeMetadata` ao código Java para as seguintes guias de entrada de código:

- Auxiliar
- Na Entrada
- No Final
- Interfaces do Otimizador

- Funções

Você pode configurar o método `storeMetadata` em uma transformação ativa para aceitar as condições de filtro para otimização push-into. O método `storeMetadata` armazena uma condição de filtro que o otimizador passa do mapeamento para a transformação de Java.

```
// Store a filter condition
storeMetadata ("FilterKey", condition);
```

CAPÍTULO 14

Expressões Java

Este capítulo inclui os seguintes tópicos:

- [Visão Geral das Expressões Java, 258](#)
- [Usando a caixa de diálogo Definir ExpressãoDefinir Função para definir uma expressão, 259](#)
- [Trabalhando com a Interface Simples, 261](#)
- [Trabalhando com a Interface Avançada, 263](#)
- [Referência de API da classe JExpression, 267](#)

Visão Geral das Expressões Java

Você pode invocar expressões do PowerCenter em uma transformação Java com a linguagem de programação Java.

Use as expressões para estender a funcionalidade de uma transformação Java. Por exemplo, você pode invocar uma expressão em uma transformação Java para procurar os valores das portas de entrada ou saída ou procurar os valores das variáveis da transformação Java.

Para invocar expressões em uma transformação Java, você deve gerar o código Java ou usar os métodos APIs da transformação Java para invocar a expressão. Você invoca a expressão e usa o resultado da mesma na guia de entrada de código apropriada. Você pode gerar o código Java que invoca uma expressão ou usar métodos de API para escrevê-lo.

A tabela a seguir descreve os métodos que você pode usar para criar e invocar expressões em uma transformação Java:

Método	Descrição
Caixa de diálogo Definir ExpressãoDefinir Função	Permite criar uma função que invoca uma expressão e gerar o código de uma expressão.
Interface simples	Permite chamar um único método API para invocar uma expressão e obter o resultado da expressão.
Interface avançada	Permite definir a expressão, invocar a expressão e usar o resultado da expressão. Se você estiver familiarizado com a programação orientada a objeto e quiser ter mais controle ao invocar a expressão, use a interface avançada.

Tipos de Função de Expressão

Você pode criar expressões para uma transformação Java usando o **Editor de Expressão**, gravando a expressão na caixa de diálogo **Definir Expressão**, usando a caixa de diálogo **Definir Função** ou usando a interface simples ou avançada.

Você pode inserir expressões que usam variáveis de porta de entrada ou saída, ou variáveis no código Java como parâmetros de entrada.

Se você usar a caixa de diálogo **Definir Expressão**, poderá usar o **Editor de Expressão** para validar a expressão antes de usá-la na transformação Java.

Se você usar a caixa de diálogo **Definir Função**, poderá validar a expressão antes de usá-la em uma transformação Java.

Você pode invocar os seguintes tipos de funções de expressão em uma transformação Java:

Tipo de função de expressão	Descrição
Funções de linguagem de transformação	Funções do tipo SQL criadas para manipular expressões comuns.
Funções definidas pelo usuário	Funções que você cria na ferramenta Developer do PowerCenter com base em funções de linguagem de transformação.
Funções personalizadas.	Funções criadas por você com a API de Função Personalizada.

Você também pode usar transformações desconectadas e , variáveis internas, variáveis de mapeamento e de fluxo trabalho definidas pelo usuário e variáveis de fluxo de trabalho predefinidas em expressões. Por exemplo, você pode usar uma transformação de pesquisa desconectada em uma expressão.

Usando a caixa de diálogo Definir ExpressãoDefinir Função para definir uma expressão

Ao definir uma expressão Java, você configura a função, cria a expressão e gera o código que chama a expressão.

Você pode definir a função e criar a expressão na caixa de diálogo **Definir ExpressãoDefinir Função**.

Para criar uma função de expressão e usar a expressão em uma transformação Java, execute as tarefas de alto nível abaixo:

1. Configure a função que invoca a expressão, incluindo função, nome, descrição e parâmetros. Quando cria a expressão, você utiliza os parâmetros da função.
2. Crie a sintaxe de expressão e valide a expressão.
3. Gere o código Java que invoca a expressão.

O Designer coloca o código na guia de entrada de código **Expressões Java** do Transformation Developer.

O Developer coloca o código na guia de entrada de código **Funções**.

Depois de gerar o código Java, chame a função gerada na guia de entrada de código apropriada para chamar uma expressão ou obter um objeto JExpression, dependendo da interface que você usar, a simples ou a avançada.

Nota: Para validar uma expressão ao criá-la, use a caixa de diálogo **Definir Expressão** **Definir Função**.

Etapa 1. Configurar a Função

Você configura parâmetros de entrada, descrição e nome para a função Java que invoca a expressão.

Use as seguintes regras e diretrizes ao configurar a função:

- Use um nome de função exclusivo que não entre em conflito com funções Java na transformação, nem com palavras-chave Java reservadas.
- Você deve configurar tipo de dados, precisão, escala e nome do parâmetro. Os parâmetros de entrada são os valores passados quando você chama a função no código Java para a transformação.
- Para transferir um tipo de dados Data para uma expressão, use um tipo de dados String para o parâmetro de entrada.

Se uma expressão retornar um tipo de dados Data, você poderá usar o valor de retorno como tipo de dados String na interface simples e tipo de dados String ou Longo na interface avançada.

Etapa 2. Criar e Validar a Expressão

Ao criar a expressão, utilize os parâmetros que você configurou para a função.

Você também pode usar funções de linguagem da transformação, funções personalizadas ou outras funções definidas pelo usuário na expressão. Você pode criar e validar a expressão nas caixas de diálogo **Definir Função** **Definir Expressão** ou na caixa de diálogo **Editor de Expressão**.

Etapa 3. Gerar o Código Java para a Expressão

Depois de configurar a função e os parâmetros dela e definir e validar a expressão, você poderá gerar o código Java que invoca a expressão.

O DesignerDeveloper coloca o código Java gerado na guia de entrada de código **Expressões Java** **Funções**. Use o código Java gerado para chamar as funções que invocam a expressão nas guias de entrada de código no Transformation Developer. Você pode gerar o código Java simples ou avançado.

Depois de gerar o código Java que invoca uma expressão, você não poderá editar nem revalidar a expressão. Para modificar uma expressão depois de gerar o código, você precisa recriar a expressão.

Criando uma expressão e gerando código Java usando a caixa de diálogo Definir Expressão Definir Função.

Você pode criar uma função que invoca uma expressão na caixa de diálogo **Definir Expressão** **Definir Função**.

Conclua as seguintes etapas para criar uma função que invoca uma expressão:

1. No Transformation Developer, abra uma transformação Java ou crie uma nova.
2. Na guia **Código Java**, clique em **Nova Função** no link **Definir Expressão**.
A caixa de diálogo **Definir Expressão** **Definir Função** é exibida.
3. Digite um nome de função.
4. Se quiser, insira uma descrição para a expressão.

Digite até 2.000 caracteres.

5. Crie os argumentos de parâmetros para a função.

Ao criar os argumentos de parâmetros, configure o nome, o tipo de dado, a precisão e a escala do argumento de parâmetro.

6. Clique em **Iniciar Editor** para criar uma expressão com os parâmetros que você criou. Na guia **Expressão**, crie uma expressão com os argumentos que você criou.
7. Para validar a expressão, clique em **Validar**.
8. Como alternativa, insira a expressão na caixa de diálogo **Expressão**. Em seguida, clique em **Validar** para validar a expressão.
9. Para gerar o código Java usando a interface avançada, selecione a opção **Gerar Código Avançado**. Em seguida, clique em **Gerar**.

O Designer Developer gera a função para invocar a expressão na guia **Expressões Java Funções** da entrada de código.

Modelos da expressão Java

Você pode gerar o código Java para uma expressão usando o código Java simples ou avançado para uma expressão.

O código Java da expressão é gerado com base no modelo da expressão.

O exemplo a seguir mostra o modelo de uma expressão Java gerado para código Java simples:

```
Object function_name (Java datatype x1[,
                        Java datatype x2 ...] )
                        throws SDK Exception
{
    return (Object)invokeJExpression( String expression,
                                     new Object [] { x1[, x2, ... ]} );
}
```

O exemplo a seguir mostra o modelo de uma expressão Java gerado com a interface avançada:

```
JExpression function_name () throws SDKException
{
    JExprParamMetadata params[] = new JExprParamMetadata[number of parameters];
    params[0] = new JExprParamMetadata (
        EDataType.STRING, // data type
        20, // precision
        0 // scale
    );
    ...
    params[number of parameters - 1] = new JExprParamMetadata (
        EDataType.STRING, // data type
        20, // precision
        0 // scale
    );
    ...
    return defineJExpression(String expression,params);
}
```

Trabalhando com a Interface Simples

Utilize o método Java API `invokeJExpression` para invocar uma expressão na interface simples.

invokeJExpression

Invoca uma expressão e retorna o valor para a expressão.

Utilize a seguinte sintaxe:

```
(datatype) invokeJExpression(  
    String expression,  
    Object[] paramMetadataArray);
```

Os parâmetros de entrada para o método invokeJExpression são um valor de cadeia de caracteres que representa a expressão e uma matriz de objetos que contém os parâmetros de entrada da expressão.

A tabela a seguir descreve os parâmetros:

Parâmetro	Tipo de Parâmetro	Tipo de dados	Descrição
expressão	Entrada	String	String que representa a expressão.
paramMetadataArray	Entrada	Objeto[]	Matriz de objetos que contém os parâmetros de entrada da expressão.

Você pode adicionar o método de invokeJExpression ao código Java em qualquer guia de entrada de código, exceto as guias **Importações** e **Funções** **Importar Pacotes** e **Expressões Java**.

Use as seguintes regras e diretrizes ao usar o método invokeJExpression:

- Tipo de dados de retorno. O tipo de dados de retorno do método invokeJExpression é um objeto. É necessário moldar o valor de retorno da função com o tipo de dados apropriado.
É possível retornar valores com os tipos de dados Inteiro, Double, String e byte[].
- Tipo de linha. O tipo de linha para valores de retorno do método invokeJExpression é INSERT.
Para usar um tipo de linha diferente para o valor de retorno, use a interface avançada.
- Valores nulos. Se você transferir um valor nulo como um parâmetro ou se o valor de retorno para o método invokeJExpression for NULL, o valor será tratado como um indicador nulo.
Por exemplo, se o valor de retorno de uma expressão for NULL e o tipo de dados retornado for String, uma string será retornada com um valor de nulo.
- Tipo de dados Data. Você deve converter os parâmetros de entrada com um tipo de dados Data para o tipo de dados String.
Para usar a string em uma expressão como tipo de dados Data, utilize a função to_date() para converter a string nesse tipo de dados.
Além disso, é necessário moldar o tipo de retorno de qualquer expressão que retorne um tipo de dados Data como uma String.

Nota: Você deve numerar os parâmetros que você transfere para a expressão consecutivamente e iniciar o parâmetro com a letra x. Por exemplo, para transferir três parâmetros para uma expressão, nomeie os parâmetros x1, x2 e x3.

Exemplo de Interface Simples

Você pode definir e chamar expressões que usam o método API invokeJExpression nas guias de entrada de código **Código Auxiliar** **Auxiliares** e **Na Linha de Entrada** **Na Entrada**.

O exemplo a seguir mostra como concluir uma pesquisa nas portas de entrada NAME e ADDRESS em uma transformação Java e atribuir o valor de retorno à porta de saída COMPANY_NAME.

Digite o seguinte código na guia de entrada de código **Na Linha de Entrada**Na Entrada:

```
COMPANY_NAME = (String)invokeJExpression(":lkp.my_lookup(X1,X2)", new Object []  
{str1 ,str2} );  
generateRow();
```

Trabalhando com a Interface Avançada

Na interface avançada, você pode usar métodos API orientados a objeto para definir, chamar e obter o resultado de uma expressão.

A tabela a seguir descreve as classes e métodos API que estão disponíveis na interface avançada:

Classe ou método API	Descrição
Classe EDataType	Enumera os tipos de dados de uma expressão.
Classe JExprParamMetadata.	Contém os metadados de cada parâmetro em uma expressão. Os metadados de parâmetro incluem o tipo de dados, a precisão e a escala.
Método API defineJExpression	Define a expressão Inclui a string e os parâmetros da expressão do PowerCenter.
Método API invokeJExpression	Chama uma expressão.
Classe JExpression	Contém os métodos para criar, chamar, obter os metadados e o resultado da expressão e verificar o tipo de dados de retorno.

Invocando uma expressão com a Interface Avançada

Você pode definir, chamar e obter o resultado de uma expressão usando a interface avançada.

1. Na guia **Código Auxiliar** ou **Na Linha de Entrada Auxiliares** ou **Na Entrada** de entrada de código, crie uma instância de Classe JExprParamMetadata para cada argumento de parâmetro da expressão e defina o valor dos metadados. Como alternativa, você pode instanciar o objeto JExprParamMetadata no método defineJExpression.
2. Use o método defineJExpression para obter o objeto JExpression da expressão.
3. Na guia de entrada de código apropriada, invoque a expressão com o método invokeJExpression.
4. Verifique o resultado do valor de retorno com o método isResultNull.
5. Você pode obter o tipo de dados ou os metadados do valor de retorno com os métodos getResultDataType e getResultMetadata.
6. Obtenha o resultado da expressão usando o método API adequado. Você pode usar os métodos getInt, getDouble, getStringBuffer e getBytes.

Regras e Diretrizes para Trabalhar com a Interface Avançada

Ao trabalhar com a interface avançada, você deve estar ciente das regras e diretrizes.

Use as seguintes regras e diretrizes:

- Se você transferir um valor nulo como um parâmetro ou o resultado de uma expressão for nulo, o valor será tratado como um indicador nulo. Por exemplo, se o resultado de uma expressão for nulo e o tipo de dados de retorno for String, será retornada uma string com um valor nulo. Você pode verificar o resultado de uma expressão usando o método `isResultNull`.
- Você precisa converter os parâmetros de entrada com um tipo de dados Data para um tipo String antes de usá-los em uma expressão. Para usar a string em uma expressão como tipo de dados Data, utilize a função `to_date()` para converter a string nesse tipo de dados.

Você pode obter o resultado de uma expressão que retorna um tipo de dados Data como String ou Longo.

Para obter o resultado de uma expressão que retorna um tipo de dados Data como String, use o método `getStringBuffer`. Para obter o resultado de uma expressão que retorne um tipo de dados Data como Longo, use o método `getLong`.

Classe EDataType

Enumera os tipos de dados Java usados nas expressões. Obtém o tipo de dados de retorno de uma expressão ou atribui o tipo de dados para um parâmetro em um objeto `JExprParamMetadata`. Não é necessário instanciar a classe `EDataType`.

A tabela a seguir mostra os valores enumerados para os tipos de dados Java nas expressões:

Tipo de dados	Valor enumerado
INT	1
DOUBLE	2
STRING	3
BYTE_ARRAY	4
DATE_AS_LONG	5

O exemplo de código Java a seguir mostra como usar a classe `EDataType` para atribuir um tipo de dados String a um objeto `JExprParamMetadata`:

```
JExprParamMetadata params[] = new JExprParamMetadata[2];
params[0] = new JExprParamMetadata (
    EDataType.STRING, // data type
    20, // precision
    0 // scale
);
...
```


Classe JExprParamMetadata

Instancia um objeto que representa os parâmetros de uma expressão e define os metadados para os parâmetros.

Use uma matriz de objetos JExprParamMetadata como entrada para o método defineJExpression a fim de definir os metadados para os parâmetros de entrada. Você pode criar uma instância do objeto JExprParamMetadata na guia de entrada do código **Expressões JavaFunções** ou em defineJExpression.

Utilize a seguinte sintaxe:

```
JExprParamMetadata paramMetadataArray[] = new JExprParamMetadata[numberOfParameters];
paramMetadataArray[0] = new JExprParamMetadata(datatype, precision, scale);
...
paramMetadataArray[numberOfParameters - 1] = new JExprParamMetadata(datatype, precision,
scale);;
```

A tabela a seguir descreve os argumentos:

Argumento	Tipo de Argumento	Tipo de Dados do Argumento	Descrição
tipo de dados	Entrada	EDatatype	Tipo de dados do parâmetro.
precisão	Entrada	Número inteiro	Precisão do parâmetro.
escala	Entrada	Número inteiro	Escala do parâmetro.

Por exemplo, use o segundo código Java para instanciar uma matriz de dois objetos JExprParamMetadata com tipos de dados String, precisão 20 e escala 0:

```
JExprParamMetadata params[] = new JExprParamMetadata[2];
params[0] = new JExprParamMetadata(EDatatype.STRING, 20, 0);
params[1] = new JExprParamMetadata(EDatatype.STRING, 20, 0);
return defineJExpression(":LKP.LKP_addresslookup(X1,X2)",params);
```

defineJExpression

Define uma expressão, incluindo a cadeia de caracteres da expressão e os parâmetros de entrada.

Argumentos para o método defineJExpression incluem uma matriz de objetos JExprParamMetadata que contém os parâmetros de entrada e um valor de cadeia de caracteres que define a sintaxe de expressão.

Utilize a seguinte sintaxe:

```
defineJExpression(
    String expression,
    Object[] paramMetadataArray
);
```

A tabela a seguir descreve os parâmetros:

Parâmetro	Tipo	Tipo de dados	Descrição
expressão	Entrada	String	String que representa a expressão.
paramMetadataArray	Entrada	Objeto	Matriz de objetos JExprParamMetadata que contém os parâmetros de entrada da expressão.

Você pode adicionar o método defineJExpression ao código Java em qualquer guia de entrada de código, exceto as guias **Importações** e **Funções**.

Para usar o método `defineJExpression`, é necessário representar uma matriz de objetos `JExprParamMetadata` que representam os parâmetros de entrada da expressão. Defina os valores de metadados para os parâmetros e transfira a matriz como um parâmetro para o método de `defineJExpression`.

Por exemplo, o seguinte código Java cria uma expressão para consultar o valor de duas strings:

```
JExprParamMetadata params[] = new JExprParamMetadata[2];
params[0] = new JExprParamMetadata(EDataType.STRING, 20, 0);
params[1] = new JExprParamMetadata(EDataType.STRING, 20, 0);
defineJExpression(":lkp.mylookup(x1,x2)", params);
```

Nota: Você deve numerar os parâmetros transferidos para a expressão consecutivamente e iniciar os parâmetros com a letra x. Por exemplo, para transferir três parâmetros para uma expressão, nomeie os parâmetros x1, x2 e x3.

Classe JExpression

Contém métodos para criar e invocar uma expressão, retornar o valor de uma expressão e verificar o tipo de dados de retorno.

A tabela a seguir lista os métodos na classe `JExpression`:

Nome do Método	Descrição
<code>invoke</code>	Chama uma expressão.
<code>getResultDataType</code>	Retorna o tipo de dados do resultado da expressão.
<code>getResultMetadata</code>	Retorna metadados do resultado da expressão.
<code>isResultNull</code>	Verifica o valor de resultado de um resultado de expressão.
<code>getInt</code>	Retorna o valor de um resultado de expressão como um tipo de dados Inteiro.
<code>getDouble</code>	Retorna o valor de um resultado de expressão como um tipo de dados Double.
<code>getStringBuffer</code>	Retorna o valor de um resultado de expressão como um tipo de dados de String.
<code>getBytes</code>	Retorna o valor de um resultado de expressão como um tipo de dados byte[].

TÓPICOS RELACIONADOS:

- [“Referência de API da classe JExpression” na página 267](#)

Exemplo de Interface Avançada

Você pode usar a interface avançada para criar e invocar uma expressão de pesquisa em uma transformação Java.

O exemplo de código Java a seguir mostra como criar uma função que chama uma expressão e como invocar a expressão para obter o valor de retorno. Este exemplo transfere os valores de duas portas de entrada com um tipo de dados String, `NAME` e `COMPANY` à função `myLookup`. A função `myLookup` usa uma expressão de pesquisa para pesquisar o valor da porta de saída `ADDRESS`.

Nota: Este exemplo parte do pressuposto de que você tem uma transformação de pesquisa não conectada no mapeamento chamada LKP_addresslookup.

Use o seguinte código Java na guia **Código AuxiliarAuxiliares** do Transformation Developer:

```
JExpression addressLookup() throws SDKException
{
    JExprParamMetadata params[] = new JExprParamMetadata[2];
    params[0] = new JExprParamMetadata (
        EDataType.STRING,          // data type
        50,                        // precision
        0                          // scale
    );
    params[1] = new JExprParamMetadata (
        EDataType.STRING,          // data type
        50,                        // precision
        0                          // scale
    );
    return defineJExpression(":LKP.LKP_addresslookup(X1,X2)",params);
}
JExpression lookup = null;
boolean isJExprObjCreated = false;
```

Use o seguinte código Java na guia **Na Linha de EntradaNa Entrada** para invocar a expressão e retornar o valor da porta ADDRESS:

```
...
if(!isJExprObjCreated)
{
    lookup = addressLookup();
    isJExprObjCreated = true;
}
lookup = addressLookup();
lookup.invoke(new Object [] {NAME,COMPANY}, ERowType.INSERT);
EDataType addressDataType = lookup.getResultDataType();
if(addressDataType == EDataType.STRING)
{
    ADDRESS = (lookup.getStringBuffer()).toString();
} else {
    logError("Expression result datatype is incorrect.");
}
...
```

Referência de API da classe JExpression

A classe JExpression contém os métodos API que permitem a você criar e invocar uma expressão, retornar o valor de uma expressão e verificar o tipo de dados de retorno.

A classe JExpression contém os seguintes métodos de API:

- getBytes
- getDouble
- getInt
- getLong
- getResultDataType
- getResultMetadata
- getStringBuffer
- invoke

- isResultNull

getBytes

Retorna o valor de um resultado de expressão como um tipo de dados `byte[]`. Obtém o resultado de uma expressão que criptografa os dados com a função `AES _ ENCRYPT`.

Utilize a seguinte sintaxe:

```
objectName.getBytes();
```

Use o seguinte exemplo de código Java para obter o resultado de uma expressão que criptografa os dados binários usando a função `AES_ENCRYPT`, onde `JExprEncryptData` é um objeto `JExpression`:

```
byte[] newBytes = JExprEncryptData.getBytes();
```

getDouble

Retorna o valor de um resultado de expressão como um tipo de dados `Double`.

Utilize a seguinte sintaxe:

```
objectName.getDouble();
```

Use o seguinte exemplo de código Java para obter o resultado de uma expressão que retorna um valor de salário como um `double`, onde `JExprSalary` é um objeto `JExpression`:

```
double salary = JExprSalary.getDouble();
```

getInt

Retorna o valor de um resultado de expressão como um tipo de dados `Inteiro`.

Utilize a seguinte sintaxe:

```
objectName.getInt();
```

Por exemplo, use o seguinte código Java para obter o resultado de uma expressão que retorna um número de ID de funcionário como um inteiro, onde `findEmpID` é um objeto `JExpression`:

```
int empID = findEmpID.getInt();
```

getLong

Retorna o valor de um resultado de expressão como um tipo de dados `Longo`. Obtém o resultado de uma expressão que usa um tipo de dados `Data`.

Utilize a seguinte sintaxe:

```
objectName.getLong();
```

Por exemplo, use o seguinte código Java para obter o resultado de uma expressão que retorna um valor `Data` como um tipo de dados `Longo`, onde `JExprCurrentDate` é um objeto `JExpression`:

```
long currDate = JExprCurrentDate.getLong();
```

getResultDataType

Retorna o tipo de dados de um resultado da expressão. Retorna um valor `EDDataType`.

Utilize a seguinte sintaxe:

```
objectName.getResultDataType();
```

Use o seguinte exemplo de código Java para invocar uma expressão e atribuir o tipo de dados do resultado ao tipo de dados da variável:

```
myObject.invoke(new Object[] { NAME,COMPANY }, ERowType INSERT);
EDataType dataType = myObject.getResultDataType();
```

getResultMetadata

Retorna os metadados em um resultado de expressão. Você pode usar `getResultMetadata` para obter a precisão, a escala e o tipo de dados de um resultado de expressão. Você pode atribuir os metadados do valor de retorno de uma expressão a um objeto `JExprParamMetadata`. Use os métodos de objeto `getScale`, `getPrecision` e `getDataType` para recuperar os metadados do resultado.

Utilize a seguinte sintaxe:

```
objectName.getResultMetadata();
```

Use o seguinte código Java para atribuir a escala, a precisão e o tipo de dados do valor de retorno de `myObject` às variáveis:

```
JExprParamMetadata myMetadata = myObject.getResultMetadata();
int scale = myMetadata.getScale();
int prec = myMetadata.getPrecision();
int datatype = myMetadata.getDataType();
```

Nota: O método de objeto `getDataType` retorna o valor inteiro do tipo de dados, conforme enumerado em `EDataType`.

getStringBuffer

Retorna o valor de um resultado de expressão como um tipo de dados `String`.

Utilize a seguinte sintaxe:

```
objectName.getStringBuffer();
```

Use o código Java a seguir para obter o resultado de uma expressão que retorna duas strings concatenadas, onde `JExprConcat` é um objeto `JExpression`:

```
String result = JExprConcat.getStringBuffer();
```

invoke

Chama uma expressão. Os argumentos para invocação incluem um objeto que define os parâmetros de entrada e o tipo de linha. Você deve instanciar um objeto `JExpression` antes de usar o método de invocar. Para o tipo de linha, use `ERowType.INSERT`, `ERowType.DELETE` e `ERowType.UPDATE`.

Utilize a seguinte sintaxe:

```
objectName.invoke(
    new Object[] { param1[, ... paramN ]},
    rowType
);
```

A tabela a seguir descreve os argumentos:

Argumento	Tipo de dados	Entrada/ Saída	Descrição
objectName	JExpression	Entrada	Nome do objeto JExpression.
parâmetros	-	Entrada	Matriz de objetos que contém os valores de entrada para a expressão.

Por exemplo, você cria uma função na guia de entrada de código **Expressões JavaFunções** chamada `address_lookup()` que retorna um objeto JExpression que representa a expressão. Use o seguinte código para invocar a expressão que usa as portas de entrada NAME e COMPANY:

```
JExpression myObject = address_lookup();  
myObject.invoke(new Object[] { NAME,COMPANY }, ERowType INSERT);
```

isResultNull

Verifica o valor de um resultado de expressão.

Utilize a seguinte sintaxe:

```
objectName.isResultNull();
```

Use o seguinte código Java para invocar uma expressão e atribuir seu valor de retorno da expressão ao endereço da variável se o valor de retorno não for nulo:

```
JExpression myObject = address_lookup();  
myObject.invoke(new Object[] { NAME,COMPANY }, ERowType INSERT);  
if(!myObject.isResultNull()) {  
    String address = myObject.getStringBuffer();  
}
```

CAPÍTULO 15

Exemplo de Transformação Java

Este capítulo inclui os seguintes tópicos:

- [Visão geral do exemplo de transformação Java, 271](#)
- [Etapa 1. Importar o Mapeamento, 272](#)
- [Etapa 2. Criar Transformação e Configurar Portas, 272](#)
- [Etapa 3. Inserir Código Java, 273](#)
- [Guia Importar Pacotes, 273](#)
- [Guia Código Auxiliar, 274](#)
- [Guia Na Linha de Entrada, 274](#)
- [Etapa 4. Compilar o Código Java, 276](#)
- [Etapa 5. Criar uma Sessão e Fluxo de Trabalho, 276](#)

Visão geral do exemplo de transformação Java

É possível usar o código Java neste exemplo para criar e compilar uma transformação Java ativa. Importe um mapeamento de amostra e crie e compile a transformação Java. Em seguida, é possível criar e executar uma sessão e um fluxo de trabalho que contenham o mapeamento.

A transformação Java processa dados de funcionários para uma empresa fictícia. Ela lê linhas de entrada de uma origem de arquivos simples e grava as linhas de saída em um destino de arquivos simples. O arquivo de origem contém dados de funcionários, como o número de identificação, o nome, o cargo e o número de identificação do gerente do funcionário.

A transformação encontra o nome do gerente de um determinado funcionário com base no número de identificação do gerente e gera linhas de saída que contenham dados dos funcionários. Dentre os dados de saída, há o número de identificação, o nome, o cargo e o nome do gerente do funcionário. Se o funcionário não tiver um gerente nos dados de origem, a transformação partirá do pressuposto de que o funcionário está no topo da hierarquia do organograma da empresa.

Nota: A lógica da transformação parte do pressuposto de que os cargos dos funcionários são organizados em ordem decrescente no arquivo de origem.

Execute as etapas a seguir para importar o mapeamento de amostra, criar e compilar uma transformação Java e criar uma sessão e um fluxo de trabalho que contenham o mapeamento:

1. Importe o mapeamento de amostra.
2. Crie a transformação Java e configure suas portas.
3. Insira o código Java da transformação nas guias de entrada de código apropriadas.

4. Compile o código Java.
5. Crie e execute uma sessão e um fluxo de trabalho.

A instalação do Cliente do PowerCenter contém um mapeamento, m_jtx_hier_useCase.xml, e uma origem de arquivo simples, hier_data, que é possível usar com este exemplo.

TÓPICOS RELACIONADOS:

- [“Transformação Java” na página 226](#)

Etapa 1. Importar o Mapeamento

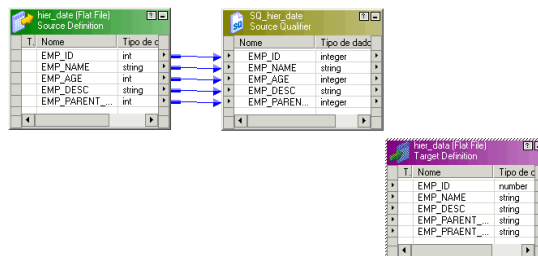
Importe os metadados do exemplo de mapeamento para o Designer. O exemplo de mapeamento contém os seguintes componentes:

- **Definição de origem e transformação de Qualificador de Origem.** Definição de origem do arquivo simples hier_input, que define os dados de origem da transformação.
- **Definição de destino.** Definição de destino do arquivo simples hier_data, que recebe os dados de saída da transformação.

Você pode importar os metadados para o mapeamento a partir deste local:

```
<PowerCenter Client installation directory>\client\bin\m_jtx_hier_useCase.xml
```

A figura a seguir mostra o exemplo de mapeamento:



Etapa 2. Criar Transformação e Configurar Portas

Você cria a transformação Java e configura as portas no Mapping Designer. Você pode usar nomes de portas de entrada e de saída como variáveis no código Java. Em uma transformação Java, crie portas de entrada e de saída em um grupo de entrada ou saída. Uma transformação Java só pode conter um grupo de entrada e um grupo de saída.

No Mapping Designer, crie uma transformação Java ativa e configure as portas. Neste exemplo, a transformação se chama jtx_hier_useCase.

Nota: Para usar o código Java neste exemplo, use os nomes exatos para as portas de entrada e de saída.

A tabela a seguir mostra as portas de entrada e de saída da transformação:

Nome da porta	Tipo de Porta	Tipo de dados	Precisão	Escala
EMP_ID_INP	Entrada	Número inteiro	10	0
EMP_NAME_INP	Entrada	String	100	0
EMP_AGE	Entrada	Número inteiro	10	0
EMP_DESC_INP	Entrada	String	100	0
EMP_PARENT_EMPID	Entrada	Número inteiro	10	0
EMP_ID_OUT	Saída	Número inteiro	10	0
EMP_NAME_OUT	Saída	String	100	0
EMP_DESC_OUT	Saída	String	100	0
EMP_PARENT_EMPNAME	Saída	String	100	0

Etapa 3. Inserir Código Java

Insira o código Java da transformação nas seguintes guias de entrada de código:

- **Importar Pacotes.** Importa os pacotes `java.util.Map` e `java.util.HashMap`.
- **Código Auxiliar.** Contém um objeto `Map`, um objeto de bloqueio e variáveis booleanas usadas para rastrear o estado de dados na transformação Java.
- **Na Fila de Entrada.** Define o comportamento da transformação Java quando ela recebe uma linha de entrada.

Guia Importar Pacotes

Importe os pacotes Java de terceiros, pacotes internos do Java ou pacotes personalizados do Java na guia Importar Pacotes. A transformação de exemplo usa os pacotes `Map` e `HashMap`.

Insira o seguinte código na guia Importar Pacotes:

```
import java.util.Map;  
import java.util.HashMap;
```

O Designer adiciona as instruções de importação ao código Java para a transformação.

TÓPICOS RELACIONADOS:

- [“Configurando definições da transformação Java” na página 240](#)

Guia Código Auxiliar

Declarar variáveis definidas pelo usuário e métodos para a transformação com Java na guia Código Auxiliar. A guia Código Auxiliar define as seguintes variáveis utilizadas pelo código Java na guia Na Linha de Entrada:

- **empMap.** Objeto de mapa que armazena o número de identificação e nome do funcionário a partir da origem.
- **lock.** Bloquear um objeto utilizado para sincronizar o acesso a empMap pelas partições.
- **generateRow.** Variável booleana utilizada para determinar se uma linha de saída deve ser gerada para a linha de entrada atual.
- **isRoot.** Variável booleana utilizada para determinar se um funcionário está na parte superior do organograma funcional (raiz).

Insira o código a seguir na guia Código Auxiliar:

```
// Static Map object to store the ID and name relationship of an
// employee. If a session uses multiple partitions, empMap is shared
// across all partitions.

private static Map <Integer, String> empMap = new HashMap <Integer, String> ();

// Static lock object to synchronize the access to empMap across
// partitions.

private static Object lock = new Object();

// Boolean to track whether to generate an output row based on validity
// of the input data.

private boolean generateRow;

// Boolean to track whether the employee is root.
private boolean isRoot;
```

Guia Na Linha de Entrada

A transformação com Java executa o código Java na guia Na Linha de Entrada quando a transformação recebe uma linha de entrada. Neste exemplo, a transformação pode ou não gerar uma linha de saída com base no valor da linha de entrada.

Insira o seguinte código na guia Na Fila de Entrada:

```
// Initially set generateRow to true for each input row.

generateRow = true;

// Initially set isRoot to false for each input row.
```

```

isRoot = false;

// Check if input employee id and name is null.
if (isNull ("EMP_ID_INP") || isNull ("EMP_NAME_INP"))
{
    incrementErrorCount(1);

    // If input employee id and/or name is null, don't generate a output
    // row for this input row.
    generateRow = false;
} else {
    // Set the output port values.
    EMP_ID_OUT = EMP_ID_INP;
    EMP_NAME_OUT = EMP_NAME_INP;
}

if (isNull ("EMP_DESC_INP"))
{
    setNull("EMP_DESC_OUT");
} else {
    EMP_DESC_OUT = EMP_DESC_INP;
}

boolean isParentEmpIdNull = isNull("EMP_PARENT_EMPID");

if(isParentEmpIdNull)
{
    // This employee is the root for the hierarchy.
    isRoot = true;
    logInfo("This is the root for this hierarchy.");
    setNull("EMP_PARENT_EMPNAME");
}

synchronized(lock)
{
    // If the employee is not the root for this hierarchy, get the
    // corresponding parent id.
    if(!isParentEmpIdNull)
        EMP_PARENT_EMPNAME = (String) (empMap.get(new Integer (EMP_PARENT_EMPID)));
    // Add employee to the map for future reference.

```

```

empMap.put (new Integer(EMP_ID_INP), EMP_NAME_INP);
}

// Generate row if generateRow is true.
if(generateRow)
    generateRow();

```

Etapa 4. Compilar o Código Java

Clique em **Compilar** no Transformation Developer para compilar o código Java para a transformação. A janela Saída exibe o status da compilação. Se o código Java não for compilado com sucesso, corrija os erros nas guias de entrada do código e recompile-o. Depois que a transformação for devidamente compilada, salve-a no repositório.

TÓPICOS RELACIONADOS:

- [“Compilando uma transformação Java” na página 242](#)
- [“Corrigindo erros de compilação” na página 243](#)

Etapa 5. Criar uma Sessão e Fluxo de Trabalho

Use o mapeamento m_jtx_hier_useCase para criar uma sessão e fluxo de trabalho para o mapeamento no Workflow Manager.

Quando configurar a sessão, você poderá usar o arquivo de exemplo da origem neste local:

```
<PowerCenter Client installation directory>\client\bin\hier_data
```

Dados de Exemplo

Os dados a seguir foram retirados do arquivo da origem de exemplo:

```

1,James Davis,50,CEO,
4,Elaine Masters,40,Vice President - Sales,1
5,Naresh Thiagarajan,40,Vice President - HR,1
6,Jeanne Williams,40,Vice President - Software,1
9,Geetha Manjunath,34,Senior HR Manager,5
10,Dan Thomas,32,Senior Software Manager,6
14,Shankar Rahul,34,Senior Software Manager,6
20,Juan Cardenas,32,Technical Lead,10
21,Pramodh Rahman,36,Lead Engineer,14
22,Sandra Patterson,24,Software Engineer,10

```

23,Tom Kelly,32,Lead Engineer,10
35,Betty Johnson,27,Lead Engineer,14
50,Dave Chu,26,Software Engineer,23
70,Srihari Giran,23,Software Engineer,35
71,Frank Smalls,24,Software Engineer,35

Os seguintes dados foram retirados de um arquivo de destino de exemplo:

1,James Davis,CEO,
4,Elaine Masters,Vice President - Sales,James Davis
5,Naresh Thiagarajan,Vice President - HR,James Davis
6,Jeanne Williams,Vice President - Software,James Davis
9,Geetha Manjunath,Senior HR Manager,Naresh Thiagarajan
10,Dan Thomas,Senior Software Manager,Jeanne Williams
14,Shankar Rahul,Senior Software Manager,Jeanne Williams
20,Juan Cardenas,Technical Lead,Dan Thomas
21,Pramodh Rahman,Lead Engineer,Shankar Rahul
22,Sandra Patterson,Software Engineer,Dan Thomas
23,Tom Kelly,Lead Engineer,Dan Thomas
35,Betty Johnson,Lead Engineer,Shankar Rahul
50,Dave Chu,Software Engineer,Tom Kelly
70,Srihari Giran,Software Engineer,Betty Johnson
71,Frank Smalls,Software Engineer,Betty Johnson

CAPÍTULO 16

Transformação Joiner

Este capítulo inclui os seguintes tópicos:

- [Visão Geral da Transformação Joiner, 278](#)
- [Propriedades da Transformação Joiner, 279](#)
- [Definição de uma Condição de Associação, 281](#)
- [Definindo o Tipo de Associação, 281](#)
- [Usando Entrada Classificada, 284](#)
- [Unindo dados a partir de uma única origem, 287](#)
- [Bloqueio dos pipelines de origem, 289](#)
- [Trabalhando com Transações, 290](#)
- [Criando uma transformação Joiner, 292](#)
- [Dicas para Transformações Joiner, 292](#)

Visão Geral da Transformação Joiner

Use a transformação Joiner para associar dados de origem de duas origens heterogêneas relacionadas que residam em diferentes locais ou sistemas de arquivos. Também é possível associar dados da mesma origem. A transformação Joiner associa origens a pelo menos uma coluna correspondente. A transformação de Associador utiliza uma condição que corresponde a um ou mais pares de colunas entre as duas origens. A transformação de Associador é uma transformação ativa.

Os dois pipelines de entrada contam com um pipeline mestre e um pipeline de detalhes, ou uma ramificação mestre e de detalhes. O pipeline mestre termina na transformação Joiner, ao passo que o pipeline de detalhes segue para o destino.

Para associar mais de duas origens em um mapeamento, associe a saída da transformação Joiner com outro pipeline de origem. Adicione transformações de Joiner ao mapeamento até ter associado todos os pipelines de origem.

A transformação Joiner aceita entrada da maioria das transformações. Entretanto, lembre das seguintes limitações dos pipelines que você conecta à transformação Joiner:

- Não é possível usar uma transformação Joiner quando qualquer pipeline de entrada contiver uma transformação de Estratégia de Atualização.
- Não é possível usar uma transformação Joiner caso se conecte uma transformação de Gerador de Sequência logo antes da transformação Joiner.

Trabalhando com a Transformação Joiner

Ao trabalhar com a transformação Joiner, você deverá configurar as propriedades da transformação, assim como o tipo e a condição de associação. Você pode configurar a transformação Joiner para entrada classificada, a fim de melhorar o desempenho do Serviço de Integração. Você também pode configurar o escopo da transformação para controlar o modo como o Serviço de Integração aplica a lógica da transformação. Para trabalhar com a transformação Joiner, execute as tarefas a seguir:

- **Configure as propriedades da transformação Joiner.** As propriedades da transformação Joiner identificam a localização do diretório de cache, o modo como o Serviço de Integração processa a transformação e o modo como ele trata do armazenamento no cache.
- **Configure a condição de associação.** A condição de associação contém portas das duas origens de entrada que precisam corresponder para que o Serviço de Integração junte duas linhas. Dependendo do tipo de associação selecionada, o Serviço de Integração adiciona a linha ao conjunto de resultados ou descarta-a.
- **Configure o tipo de associação.** Uma associação é um operador relacional que combina dados de várias tabelas em diferentes bancos de dados ou arquivos simples em um único conjunto de resultados. Você pode configurar a transformação Joiner para usar o tipo de associação Normal, Externa Mestre, Externa de Detalhe ou Externa Completa.
- **Configure a sessão para entrada classificada ou não classificada.** Você pode melhorar o desempenho da sessão configurando a transformação Joiner para usar a entrada classificada. Para configurar um mapeamento a fim de usar dados classificados, você estabelece e mantém uma ordem de classificação no mapeamento de modo que o Serviço de Integração possa usar os dados classificados quando processar a transformação Joiner.
- **Configure o escopo da transação.** Quando o Serviço de Integração processa uma transformação Joiner, ele pode aplicar a lógica da transformação a todos os dados em uma transação, a todos os dados de entrada ou a uma linha de dados por vez.

Se a opção de particionamento estiver disponível no PowerCenter, você poderá aumentar o número de partições em um pipeline para melhorar o desempenho da sessão.

Propriedades da Transformação Joiner

As propriedades da transformação Joiner identificam a localização do diretório de cache, o modo como o Serviço de Integração processa a transformação e o modo como ele trata do armazenamento no cache. As propriedades também determinam como o Serviço de Integração junta as tabelas e arquivos.

Quando você cria um mapeamento, especifica as propriedades para cada transformação Joiner. Quando você cria uma sessão, pode anular algumas propriedades, como o tamanho do índice e do cache de dados para cada transformação.

A tabela a seguir descreve as propriedades da transformação Joiner:

Opção	Descrição
Diretório de Cache	Especifica o diretório usado para armazenar em cache as linhas principal e de detalhe e o índice dessas linhas. Por padrão, os arquivos de cache são criados em um diretório especificado pela variável de processo \$PMCacheDir. Se você substituir o diretório, verifique se ele já existe e se possui espaço em disco suficiente para os arquivos de cache. O diretório pode ser uma unidade mapeada ou montada.
Tipo de associação	Especifica o tipo de associação: Normal, Externa Mestre, Externa Detalhada ou Externa Completa.
Ordenação dos Nulos no Mestre	Não é aplicável a esse tipo de transformação.
Ordenação dos Nulos no Detalhe	Não é aplicável a esse tipo de transformação.
Nível de Rastreamento	Volume de detalhes exibidos no log de sessão para esta transformação. As opções são Conciso, Normal, Dados Detalhados e Inicialização Detalhada.
Tamanho de Cache de Dados de Associador	Tamanho do cache de dados para a transformação. O tamanho padrão do cache é 2.000.000 bytes. Se o tamanho total do cache configurado for 2 GB ou mais, você deverá executar a sessão em um Serviço de Integração de 64 bits. É possível usar um valor numérico para o cache, usar um valor de cache proveniente de um arquivo de parâmetros ou configurar o Serviço de Integração para definir o tamanho do cache usando a configuração Automático. Se você configurar o Serviço de Integração para determinar o tamanho do cache, também poderá especificar a quantidade máxima de memória alocada ao cache pelo Serviço de Integração.
Tamanho de Cache de Índice de Associador	Tamanho do cache de índice para a transformação. O tamanho padrão do cache é 1.000.000 bytes. Se o tamanho total do cache configurado for 2 GB ou mais, você deverá executar a sessão em um Serviço de Integração de 64 bits. É possível usar um valor numérico para o cache, usar um valor de cache proveniente de um arquivo de parâmetros ou configurar o Serviço de Integração para definir o tamanho do cache usando a configuração Automático. Se você configurar o Serviço de Integração para determinar o tamanho do cache, também poderá especificar a quantidade máxima de memória alocada ao cache pelo Serviço de Integração.
Entrada Classificada	Especifica que os dados sejam classificados. Escolha Entrada Classificada para associar os dados classificados. O uso da entrada classificada pode melhorar o desempenho.
Ordem de Classificação Mestre	Especifica a ordem de classificação dos dados da origem mestra. Escolha Crescente se os dados da origem mestra estiverem na ordem crescente. Se você escolher Crescente, ative também a entrada classificada. O Padrão é Automático.
Escopo da Transformação	Especifica como o Serviço de Integração aplica a lógica de transformação aos dados de entrada. Você pode escolher Transação, Todas as Entradas ou Linha.

Definição de uma Condição de Associação

A condição de associação contém portas das duas origens de entrada que precisam corresponder para que o Serviço de Integração junte duas linhas. Dependendo do tipo de associação selecionada, o Serviço de Integração adiciona a linha ao conjunto de resultados ou descarta-a. A transformação Joiner produz conjuntos de resultados com base no tipo de associação, na condição e nas origens de dados de entrada.

Antes que você defina uma condição de associação, verifique se as origens mestre e detalhada foram configuradas para um desempenho ideal. Durante a sessão, o Serviço de Integração compara cada linha da origem mestre com a origem de detalhes. Para aumentar o desempenho de uma Transformação Joiner não ordenada, use a origem com menos linhas que a origem mestra. Para aumentar o desempenho de uma Transformação Joiner ordenada, use a origem com menos valores-chave duplicados como o mestre.

Por padrão, quando você adiciona portas a uma transformação Joiner, as portas do primeiro pipeline de origem são exibidas como origens detalhadas. A adição das portas do segundo pipeline de origem as configura como origens mestres. Para alterar estas configurações, clique na coluna M na guia Portas referente às portas que você deseja configurar como a origem mestre. Isto configura as portas desta origem como mestres, e as portas da outra como detalhadas.

Você define uma ou mais condições com base na igualdade entre as origens mestres e detalhadas especificadas. Por exemplo, se duas origens com tabelas nomeadas EMPLOYEE_AGE e EMPLOYEE_POSITION contêm números de ID de funcionário, a seguinte condição corresponde às linhas com funcionários listados em ambas as origens:

```
EMP_ID1 = EMP_ID2
```

Use uma ou mais portas das origens de entrada de uma transformação Joiner na condição de associação. Portas adicionais aumentam o tempo necessário para a associação de duas origens. A ordem das portas na condição pode afetar o desempenho da transformação Joiner. Se você usar múltiplas portas na condição de associação, o Serviço de Integração compara as portas na ordem que você especificar.

O Designer valida os tipos de dados em uma condição. Ambas as portas de uma condição devem ter o mesmo tipo de dados. Se você precisar usar duas portas na condição com tipos de dados que não correspondem, converta os tipos de dados para que correspondam.

Se você juntar os tipos de dados Char e Varchar, o Serviço de Integração conta qualquer espaço que preenche os valores Char como parte da string:

```
Char(40) = "abcd"  
Varchar(40) = "abcd"
```

O valor Char é preenchido por "abcd" com 36 espaços em branco e o Serviço de Integração não faz a associação dos dois campos porque o campo Char contém espaços à direita.

Nota: A transformação Joiner não corresponde valores nulos. Por exemplo, se EMP_ID1 e EMP_ID2 contiverem uma linha com um valor nulo, o Serviço de Integração não os considera uma correspondência e não junta as duas linhas. Para juntar linhas com valores nulos, substitua a entrada nula por valores padrão e, em seguida, junte os valores padrão.

Definindo o Tipo de Associação

No SQL, uma associação é um operador relacional que combina dados de diversas tabelas em um único conjunto de resultados. A transformação Joiner é semelhante a uma associação de SQL, mas os dados podem se originar em diferentes tipos de origem.

Você define o tipo de associação na guia Propriedades na transformação. A transformação Joiner permite os seguintes tipos de associação:

- Normal
- Externa Mestre
- Externa de Detalhes
- Externa Completa

Nota: Uma associação externa normal ou mestre trabalha mais rápido do que uma associação externa completa ou externa de detalhes.

Se um conjunto de resultados apresentar campos que não contenham dados em nenhuma das origens, a transformação Joiner preenche os campos abertos com valores nulos. Se souber que um campo retornará um valor NULL e não quiser inserir NULLs no destino, é possível definir um valor padrão na guia Portas para a porta correspondente.

Associação Normal

Com uma associação normal, o Serviço de Integração descarta todas as linhas de dados da origem mestre e de detalhes que não combinam com base na condição.

Por exemplo, é possível que haja duas origens de dados para peças de automóvel chamadas de PARTS_SIZE e PARTS_COLOR com os dados a seguir:

PARTS_SIZE (master source)

PART_ID1	DESCRIPTION	SIZE
1	Seat Cover	Large
2	Ash Tray	Small
3	Floor Mat	Medium

PARTS_COLOR (detail source)

PART_ID2	DESCRIPTION	COLOR
1	Seat Cover	Blue
3	Floor Mat	Black
4	Fuzzy Dice	Yellow

Para associar as duas tabelas combinando PART_IDs nas duas origens, defina as condições da seguinte forma:

```
PART_ID1 = PART_ID2
```

Ao associar essas tabelas com uma associação normal, o conjunto resultante apresenta os dados a seguir:

PART_ID	DESCRIPTION	SIZE	COLOR
1	Seat Cover	Large	Blue
3	Floor Mat	Medium	Black

O exemplo a seguir mostra a instrução SQL equivalente:

```
SELECT * FROM PARTS_SIZE, PARTS_COLOR WHERE PARTS_SIZE.PART_ID1 = PARTS_COLOR.PART_ID2
```

Associação Externa Mestre

Uma associação externa mestre mantém todas as linhas de dados a partir da origem do detalhe e as linhas correspondentes a partir da origem mestre. Ela descarta as linhas não correspondidas da origem mestre.

Quando você junta as tabelas de amostra com uma associação externa mestre e a mesma condição, o conjunto de resultados inclui os seguintes dados:

PART_ID	DESCRIPTION	SIZE	COLOR
1	Seat Cover	Large	Blue
3	Floor Mat	Medium	Black
4	Fuzzy Dice	NULL	Yellow

Como nenhum tamanho é especificado para o Fuzzy Dice, o Serviço de Integração preenche o campo com um NULL.

O exemplo a seguir mostra a instrução SQL equivalente:

```
SELECT * FROM PARTS_SIZE RIGHT OUTER JOIN PARTS_COLOR ON (PARTS_COLOR.PART_ID2 = PARTS_SIZE.PART_ID1)
```

Associação Externa Detalhada

Uma associação externa detalhada mantém todas as linhas de dados na origem mestre e as linhas correspondentes na origem do detalhe. Ela descarta as linhas não correspondidas da origem detalhada.

Ao associar as tabelas de amostra a uma associação externa detalhada e à mesma condição, o conjunto de resultados incluirá os seguintes dados:

PART_ID	DESCRIPTION	SIZE	COLOR
1	Seat Cover	Large	Blue
2	Ash Tray	Small	NULL
3	Floor Mat	Medium	Black

Como nenhuma cor é especificada para o Ash Tray, o Serviço de Integração preenche o campo com um NULL.

O exemplo a seguir mostra a instrução SQL equivalente:

```
SELECT * FROM PARTS_SIZE LEFT OUTER JOIN PARTS_COLOR ON (PARTS_SIZE.PART_ID1 = PARTS_COLOR.PART_ID2)
```

Associação Externa Completa

Uma associação externa completa mantém todas as linhas de dados a partir das origens mestre e detalhada.

Quando você junta as tabelas de amostra com uma associação externa completa e a mesma condição, o conjunto de resultados inclui:

PART_ID	DESCRIPTION	SIZE	Color
1	Seat Cover	Large	Blue
2	Ash Tray	Small	NULL
3	Floor Mat	Medium	Black
4	Fuzzy Dice	NULL	Yellow

Uma vez que nenhuma cor foi especificada para o Ash Tray e nenhum tamanho foi especificado para Fuzzy Dice, o Serviço de Integração preenche os campos com NULL.

O exemplo a seguir mostra a instrução SQL equivalente:

```
SELECT * FROM PARTS_SIZE FULL OUTER JOIN PARTS_COLOR ON (PARTS_SIZE.PART_ID1 =  
PARTS_COLOR.PART_ID2)
```

Usando Entrada Classificada

Você pode melhorar o desempenho da sessão configurando a transformação Joiner para usar a entrada classificada. Quando você configura a transformação Joiner para utilizar dados classificados, o Serviço de Integração melhora o desempenho, minimizando a entrada e a saída de disco. Você vê a máxima melhoria no desempenho quando trabalha com grandes conjuntos de dados.

Para configurar um mapeamento para usar dados classificados, você estabelece e mantém uma ordem de classificação no mapeamento de modo que o Serviço de Integração possa usar os dados classificados quando processar a transformação Joiner. Conclua as seguintes tarefas para configurar o mapeamento:

- **Configure a ordem de classificação.** Configure a ordem de classificação dos dados que deseja unir. Você pode unir arquivos simples classificados ou classificar dados relacionais usando uma transformação Qualificador de origem. Você também pode usar uma transformação Classificador.
- **Adicione transformações.** Use transformações para manter a ordem dos dados classificados.
- **Configure a transformação Joiner.** Configure a transformação Joiner para utilizar dados classificados e configure a condição de associação para usar as portas de origem de classificação. A origem de classificação representa a origem dos dados classificados.

Ao configurar a ordem de classificação em uma sessão, você pode selecionar uma ordem de classificação associada à página de código do Serviço de Integração. Quando você executa o Serviço de Integração no modo Unicode, ele utiliza a ordem de classificação da sessão selecionada para classificar dados de caractere. Quando você executa-o no modo ASCII, ele classifica todos os dados de caractere usando uma ordem de classificação binária. Para assegurar que os dados sejam classificados conforme o Serviço de Integração requer, a ordem de classificação do banco de dados deve ser igual à ordem de classificação da sessão definida pelo usuário.

Ao unir dados classificados a partir de pipelines particionados, você deve configurar as partições para manter a ordem dos dados classificados.

Configuração da Ordem de Classificação

Você deve configurar a ordem de classificação para garantir que o Serviço de Integração transfira dados classificados para a transformação Joiner.

Configure a ordem de classificação usando um dos métodos a seguir:

- **Use arquivos simples classificados.** Quando os arquivos simples contêm dados classificados, verifique se a ordem das colunas do tipo corresponde em cada arquivo de origem.
- **Use dados relacionais classificados.** Use portas classificadas na transformação Qualificador de Origem para classificar as colunas do banco de dados de origem. Configure a ordem das portas classificadas da mesma maneira em toda transformação Qualificador de origem.
- **Use as transformações Separador.** Use uma transformação Separador para classificar dados relacionais ou de arquivo simples. Coloque uma transformação Separador nos pipelines mestre e detalhado. Configure cada transformação Separador para usar a mesma ordem das portas de chave de classificação e a mesma direção da ordem de classificação.

Se você transferir dados não classificados (ou incorretamente classificados) para uma transformação Joiner configurada para usar dados classificados, a sessão falha e o Serviço de Integração registra o erro no arquivo de log da sessão.

Adição de transformações ao mapeamento

Quando você adiciona transformações entre a origem da classificação e a transformação Joiner, use as seguintes orientações para manter os dados classificados:

- Não coloque nenhuma das seguintes transformações entre a origem da classificação e a transformação Joiner:
 - Personalizar
 - Agregador Não Classificado
 - Normalizador
 - Classificação
 - Transformação União
 - Transformação Analisador de XML
 - Transformação Gerador XML
 - Maplet, se ele contiver uma das transformações mencionadas acima
- Você pode colocar uma transformação Agregador classificada entre a origem da classificação e a transformação Joiner, se usar as orientações a seguir:
 - Configure a transformação Agregador para a entrada classificada.
 - Use as mesmas portas para o grupo por colunas, na transformação Agregador, que as portas na origem da classificação.
 - O grupo por portas deve estar na mesma ordem que as portas na origem da classificação.
- Quando você junta o conjunto de resultados de uma transformação Joiner com outro pipeline, verifique se a saída de dados da primeira transformação Joiner é classificada.

Sugestão: Você pode colocar a transformação Joiner diretamente depois da origem da classificação, para manter os dados classificados.

Configurando a Transformação Joiner

Para configurar a transformação Joiner, complete as seguintes tarefas:

- Ative a Entrada Classificada na guia Propriedades.
- Defina a condição de associação para que receba dados classificados na mesma ordem que a origem de classificação.

Definindo a Condição de Associação

Configure a condição de associação para manter a ordem de classificação estabelecida na origem da classificação: o arquivo simples classificado, a transformação do Qualificador de Origem ou a transformação de Origem. Se você utilizar uma transformação de Agregador classificada entre a origem de classificação e a transformação Joiner, trate a transformação de Agregador classificada como origem de classificação ao definir a condição de associação. Use as seguintes orientações ao definir condições de associação:

- As portas usadas na condição de associação devem combinar com as portas na origem de classificação.
- Ao configurar várias condições de associação, as portas da primeira condição de associação devem combinar com as primeiras portas da origem de classificação.
- Ao configurar várias condições, a ordem das condições deve combinar com a ordem das portas da origem de classificação, e não se deve ignorar nenhuma porta.
- O número de portas classificadas da origem de classificação pode ser maior ou igual ao número de portas da condição de associação.

Exemplo de uma Condição de Associação

Por exemplo, você configura as transformações do Classificador nos pipelines mestre e de detalhes com as seguintes portas classificadas:

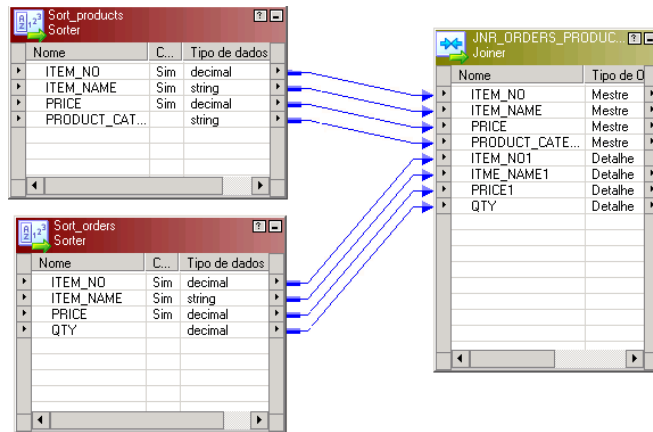
1. ITEM_NO
2. ITEM_NAME
3. PRICE

Ao configurar a condição de associação, use as seguintes orientações para manter a ordem de classificação:

- É necessário usar ITEM_NO na primeira condição de associação.
- Caso se adicione uma segunda condição de associação, será necessário usar ITEM_NAME.
- Caso você deseje usar PRICE em uma condição de associação, também será necessário usar ITEM_NAME na segunda condição de associação.

Caso você ignore o ITEM_NAME e a associação no ITEM_NO e PRICE, a ordem de classificação será perdida e o Serviço de Integração não obterá êxito na sessão.

A figura a seguir mostra um mapeamento configurado para classificação e associação nas portas ITEM_NO, ITEM_NAME e PRICE:



Ao usar a transformação Joiner para associar os pipelines mestre e de detalhes, é possível configurar qualquer uma das seguintes condições de associação:

ITEM_NO = ITEM_NO

ou

ITEM_NO = ITEM_NO1
ITEM_NAME = ITEM_NAME1

ou

ITEM_NO = ITEM_NO1
ITEM_NAME = ITEM_NAME1
PRICE = PRICE1

Unindo dados a partir de uma única origem

Convém unir dados da mesma origem se você quiser efetuar um cálculo em parte dos dados e unir os dados transformados aos dados originais. Ao unir os dados com este método, é possível manter os dados originais e transformar partes desses dados dentro de um mapeamento. É possível unir dados da mesma origem das seguintes maneiras:

- Uma duas ramificações da mesma pipeline.
- Uma duas instâncias da mesma origem.

Unindo duas ramificações do mesmo pipeline

Ao associar dados da mesma origem, é possível criar duas ramificações do pipeline. Ao ramificar um pipeline, é necessário adicionar uma transformação entre o qualificador de origem e a transformação Joiner em pelo menos uma ramificação do pipeline. É necessário associar dados classificados e configurar a transformação Joiner para entrada classificada.

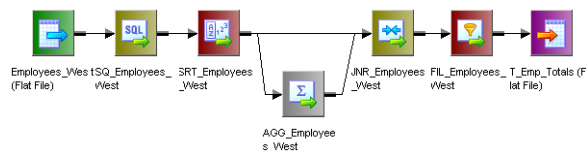
Por exemplo, há uma origem com as seguintes portas:

- Funcionário
- Departamento
- Venda total

No destino, você deseja visualizar os funcionários que geraram vendas maiores que as vendas médias de seus departamentos. Para fazer isso, crie um mapeamento com as seguintes transformações:

- **Transformação Classificador.** Classifica os dados.
- **Transformação Agregador Classificado** Efetua a média de dados e de grupo de vendas por departamento. Ao efetuar essa agregação, perde-se os dados para cada funcionário. Para manter os dados de funcionários, é necessário transferir uma ramificação do pipeline à transformação do Agregador e transferir uma ramificação com os mesmos dados à transformação Joiner para manter os dados originais. Ao associar as duas ramificações do pipeline, associa-se os dados agregados aos dados originais.
- **Transformação Joiner classificada.** Usa uma transformação Joiner classificada para associar os dados agregados classificados aos dados originais.
- **Transformação Filtro.** Compara os dados de vendas médias com os dados de vendas para cada funcionário e separa funcionários com vendas médias que sejam inferiores ao que está acima da média.

A seguinte figura mostra um mapeamento que une duas ramificações do mesmo pipeline:



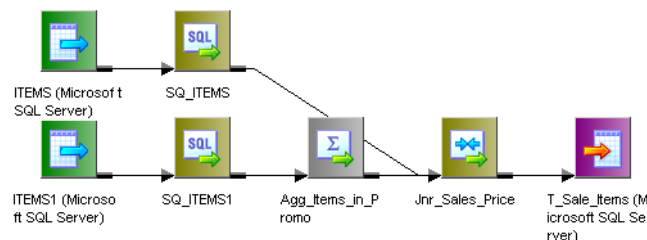
Nota: Também é possível associar dados de grupos de saída da mesma transformação, como a transformação Personalizada ou a transformação de Qualificador de Origem XML. Coloque uma transformação de Classificador entre cada grupo de saída e a transformação Joiner e configure a transformação Joiner para receber entrada classificada.

Associar duas ramificações pode causar impacto no desempenho se a transformação Joiner receber dados de uma ramificação muito mais tarde do que de outra ramificação. A transformação Joiner armazena em cache todos os dados da primeira ramificação e registra o cache em disco se ele ficar cheio. Logo, a transformação Joiner deve ler os dados do disco ao receber os dados da segunda ramificação. Isso pode desacelerar o processamento.

Unindo duas instâncias da mesma origem

Também é possível associar dados da mesma origem, criando uma segunda instância da origem. Depois de criar a instância da segunda origem, é possível associar os pipelines das duas instâncias de origem. Se desejar associar dados não associados, é necessário criar duas instâncias da mesma origem e associar os pipelines.

A figura a seguir mostra duas instâncias da mesma origem associadas com uma transformação Joiner:



Nota: Ao associar dados usando este método, o Serviço de Integração lê os dados de origem para cada instância de origem, de modo que o desempenho possa ser mais lento do que associar duas ramificações de um pipeline.

Diretrizes para unir dados a partir de uma única origem

Use as seguintes orientações ao decidir se vai associar ramificação de um pipeline ou associar duas instâncias de uma origem:

- Associe duas ramificações de um pipeline quando tiver uma grande origem ou se conseguir ler os dados de origem apenas uma vez. Por exemplo, é possível ler apenas uma vez os dados de origem de uma fila de mensagens.
- Associe duas ramificações de um pipeline ao usar dados classificados. Se os dados de origem não estiverem classificados e você usar uma transformação de Classificação para classificar os dados, ramifique o pipeline após classificar os dados.
- Associe duas instâncias de uma origem quando precisar adicionar uma transformação de bloqueio a um pipeline entre a origem e a transformação do Joiner.
- Associe duas instâncias de uma origem se um pipeline tiver a possibilidade de processar mais devagar do que outro pipeline.
- Associe duas instâncias de uma origem se precisar associar dados não classificados.

Bloqueio dos pipelines de origem

Quando você executa uma sessão com uma Transformação Joiner, o Serviço de Integração bloqueia e desbloqueia os dados de origem, com base na configuração do mapeamento e se você configurou a Transformação Joiner para inserção classificada.

Transformação Joiner Não Classificada

Quando o Serviço de Integração processa uma transformação não classificada Joiner, ele lê todas as linhas mestres antes de ler as linhas de detalhes. Para garantir a leitura de todas as linhas mestres antes das linhas de detalhe, o Serviço de Integração bloqueia a origem de detalhes enquanto armazena em cache as linhas da origem mestre. Assim que o Serviço de Integração lê e armazena em cache todas as linhas mestres, ele desbloqueia a origem de detalhes e lê as linhas de detalhe. Alguns mapeamentos com transformações Joiner violam a validação do fluxo de dados.

Transformação Joiner Classificada

Quando o Serviço de Integração processa uma transformação Joiner classificada, ele bloqueia dados com base na configuração de mapeamento. A lógica de bloqueio será possível se a entrada mestre e a de detalhes para a transformação Joiner tiverem origem em origens diferentes.

O Serviço de Integração usa a lógica de bloqueio para processar a transformação Joiner se puder fazê-lo sem bloquear todas as origens em um grupo de classificação de carregamento de destino ao mesmo tempo. Caso contrário, ele não usará a lógica de bloqueio. Em vez disso, armazenará mais linhas no cache.

Quando o Serviço de Integração puder usar a lógica de bloqueio para processar a transformação Joiner, ele armazenará menos linhas no cache, melhorando o desempenho.

Cache das Linhas Mestres

Quando o Serviço de Integração processa uma transformação Separador, ele lê as linhas de ambas as origens simultaneamente e cria o índice e o cache de dados com base nas linhas mestres. Em seguida, o Serviço de Integração executa a associação com base nos dados da origem detalhada e nos dados de

cache. O número de linhas que o Serviço de Integração armazena no cache depende do tipo de partição, dos dados de origem e de você ter ou não configurado a transformação de transformação de Associador para a entrada classificada. Para aumentar o desempenho de uma Transformação de Associador não ordenada, use a origem com menos linhas que a origem mestra. Para aumentar o desempenho de uma Transformação de Associador ordenada, use a origem com menos valores-chave duplicados como o mestre.

Trabalhando com Transações

Quando o Serviço de Integração processa uma transformação Joiner, ele pode aplicar a lógica da transformação a todos os dados em uma transação, a todos os dados de entrada ou a uma linha de dados por vez. O Serviço de Integração pode descartar ou preservar limites da transação, dependendo da configuração do mapeamento e do escopo da transformação. Você configura o modo como o Serviço de Integração aplica a lógica de transformação e trata dos limites da transação usando a propriedade de escopo da transformação.

Você configura valores de escopo da transformação com base na configuração de mapeamento e no fato de querer preservar ou descartar limites da transação.

Você pode preservar limites de transação ao unir as seguintes origens:

- **Você une duas ramificações do mesmo pipeline de origem.** Utilize o escopo da transformação Transação para preservar limites de transação.
- **Você une duas origens e deseja preservar limites de transação para a origem de detalhe.** Utilize o escopo da transformação Linha para preservar limites da transação no pipeline de detalhes.

Você pode descartar os limites de transação ao unir as seguintes origens:

- **Você une duas origens ou duas ramificações e deseja descartar limites da transação.** Utilize o escopo da transformação Todas as Entradas para aplicar a lógica da transformação a todos os dados de entrada e descartar limites da transação para os dois pipelines.

A tabela a seguir resume como preservar limites da transação usando escopos da transformação com a transformação Joiner:

Escopo de Transformação	Tipo de entrada	Comportamento do Serviço de Integração
Linha	Não classificada	Preserva os limites de transação na pipeline de detalhes.
Linha	Classificada	Falha na sessão.
Transação	Classificada	Preserva limites da transação quando mestre e detalhe têm origem no mesmo gerador de transação. A sessão falha quando mestre e detalhe não têm origem no mesmo gerador de transação
Transação	Não classificada	Falha na sessão.
Todas as entradas	Classificado, Não classificado	Descarta limites da transação.

Nota: As sessões falham se você usar dados em tempo real com escopos de transformação Todas as Entradas ou Transação.

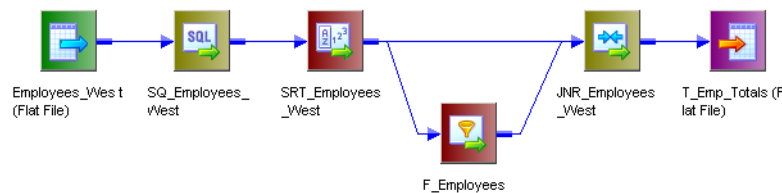
Preservando os Limites de Transação de um Único Pipeline

Ao unir dados da mesma origem, utilize o escopo da transformação Transação para preservar limites da transação de entrada para um pipeline. Utilize o escopo da transformação Transação quando a transformação Joiner juntar dados da mesma origem, seja de duas ramificações do mesmo pipeline, seja de dois grupos de saída de um gerador de transações. Utilize esse escopo de transformação com dados classificados e qualquer tipo de associação.

Ao usar o escopo da transformação Transação, verifique se os pipelines mestre e de detalhes têm origem no mesmo ponto de controle de transação e se você utiliza a entrada classificada. Por exemplo, em [“Preservando os Limites de Transação de um Único Pipeline” na página 291](#) a transformação Classificador é o ponto de controle de transação. Você não pode colocar outro ponto de controle de transação entre a transformação Classificador e a transformação Joiner. No mapeamento, as ramificações de pipeline mestre e de detalhes têm origem no mesmo ponto de controle de transação. O Serviço de Integração junta as ramificações de pipeline com a transformação Joiner, preservando limites da transação.

A figura a seguir mostra um mapeamento que une duas ramificações de um pipeline e preserva limites da transação:

Figura 1. Preservando Limites da Transação ao Unir Duas Ramificações de Pipeline



Preservando Limites da Transação no Pipeline de Detalhes

Para preservar os limites da transação no pipeline de detalhes, escolha o escopo da transformação Linha. O escopo da transformação Linha permite que o Serviço de Integração processe os dados, uma linha por vez. O Serviço de Integração armazena os dados mestre no cache e os compara com os dados de detalhe.

Quando os dados de origem são originados a partir de uma origem em tempo real, como IBM MQ Series, o Serviço de Integração compara os dados mestre armazenados no cache com cada mensagem à medida que lê a origem de detalhes.

Utilize o escopo da transformação Linha com os tipos de associação Normal e Externa Mestre que usam dados não classificados.

Descarte dos Limites de Transação para Dois Pipelines

Quando você deseja juntar dados a partir de duas origens ou duas ramificações e não precisa preservar os limites da transação, use o escopo de transformação Todas as Entradas. Quando você usa Todas as Entradas, o Serviço de Integração descarta os limites de transação de entrada para os dois pipelines e gera a saída de todas as linhas da transformação como uma transação aberta. Na transformação Joiner, os dados do pipeline mestre podem ser colocados em cache ou reunidos simultaneamente, dependendo de como você configura a ordem de classificação. Utilize esse escopo de transformação com dados classificados não classificados e qualquer tipo de associação.

Criando uma transformação Joiner

Para usar uma Transformação Joiner, adicione uma Transformação Joiner ao mapeamento, defina as origens de entrada e configure a transformação com uma condição e efetue a associação e a classificação de tipos.

Para criar uma Transformação Joiner:

1. No Mapping Designer, clique em Transformação > Criar. Selecione a Transformação Joiner. Digite um nome e clique em OK.

A convenção de nomenclatura da Transformação Joiner é `JNR_TransformationName`. Insira uma descrição para a transformação.

O Designer cria a transformação Joiner.

2. Arraste todas as portas de entrada/saída da primeira origem para dentro da Transformação Joiner.

O Designer cria portas de entrada/saída para os campos de origem na Transformação Joiner como campos de detalhe por padrão. É possível editar essa propriedade mais tarde.

3. Selecione e arraste todas as portas de entrada/saída da segunda origem para dentro da Transformação Joiner.

O Designer configura o segundo conjunto de campos de origem e os campos mestres por padrão.

4. Clique duas vezes na barra de títulos da Transformação Joiner para abrir a transformação.

5. Clique na guia Portas.

6. Clique em qualquer caixa da coluna M para alternar a relação mestra/detalhe das origens.

Sugestão: Para aumentar o desempenho de uma Transformação Joiner não ordenada, use a origem com menos linhas que a origem mestra. Para aumentar o desempenho de uma Transformação Joiner ordenada, use a origem com menos valores-chave duplicados como o mestre.

7. Adicione valores padrão para portas específicas.

Algumas portas têm a probabilidade de conter valores nulos, uma vez que o campo em uma das origens pode estar vazio. É possível especificar um valor padrão se o banco de dados de destino não tratar de NULLs.

8. Clique na guia Condições e defina a condição de associação.

9. Clique no botão Adicionar para adicionar uma condição. É possível adicionar várias condições.

A porta mestra e a de detalhes devem ter tipos de dados correspondentes. A Transformação Joiner somente é compatível com associações equivalentes (=).

10. Clique na guia Propriedades e configure as propriedades da transformação.

Nota: É possível editar a condição de associação a partir da guia Condição. A palavra-chave AND separa várias condições.

11. Clique em OK.

12. Clique na guia Extensões de Metadados para configurar extensões de metadados.

Dicas para Transformações Joiner

Quando possível, faça associações em um banco de dados.

É mais rápido fazer uma associação em um banco de dados do que na sessão. Em alguns casos, isso não é possível. Por exemplo, não é possível unir tabelas de dois bancos de dados diferentes ou de sistemas de arquivos simples. Se você quiser realizar uma associação em um banco de dados, utilize as seguintes opções:

- Crie um procedimento armazenado de pré-sessão para unir as tabelas em um banco de dados.
- Use a transformação Qualificador de Origem para fazer a associação.

Quando possível, junte dados classificados.

Você pode melhorar o desempenho da sessão configurando a transformação Joiner para usar a entrada classificada. Quando você configura a transformação Joiner para utilizar dados classificados, o Serviço de Integração melhora o desempenho, minimizando a entrada e a saída de disco. Você vê a máxima melhoria no desempenho quando trabalha com grandes conjuntos de dados.

No caso de uma transformação Joiner não classificada, designe a origem com um número menor de linhas que a origem mestre.

Para alcançar melhores níveis de desempenho e de armazenamento em disco, designe a origem com menos linhas que a origem mestre. Durante a sessão, a transformação Joiner compara cada linha da origem mestre com a origem de detalhes. Quanto menor o número de linhas exclusivas na origem mestre, menor o número de iterações da comparação das associações que ocorrerão, o que acelera o processo de associação.

No caso de uma transformação Joiner classificada, designe a origem com um número menor de valores chave duplicados que a origem mestre.

Para obter níveis ideais de desempenho e armazenamento em disco, indique a origem com menos valores de chave duplicados como a origem mestre. Quando o Serviço de Integração processa uma transformação Joiner classificada, ele armazena as linhas no cache, cem chaves por vez. Se a origem mestre contiver muitas linhas com o mesmo valor chave, o Serviço de Integração deverá armazenar mais linhas no cache. E, com isso, o desempenho ficará mais lento.

CAPÍTULO 17

Transformação Pesquisa

Este capítulo inclui os seguintes tópicos:

- [Visão Geral de Transformação Pesquisa, 294](#)
- [Tipos de Origem da Pesquisa, 295](#)
- [Pesquisas Conectadas e Não Conectadas, 298](#)
- [Componentes da Pesquisa, 301](#)
- [Propriedades de pesquisa, 303](#)
- [Consulta de Pesquisa, 310](#)
- [Condição de pesquisa, 314](#)
- [Caches de Pesquisa, 316](#)
- [Retornar Várias Linhas, 317](#)
- [Configurando Transformações de Pesquisa Não Conectadas, 318](#)
- [Resiliência de Deadlock de Banco de Dados, 320](#)
- [Criando uma transformação Pesquisa, 321](#)
- [Dicas para Transformações de Pesquisa, 322](#)

Visão Geral de Transformação Pesquisa

Use uma transformação de Pesquisa em um mapeamento para pesquisar dados em um arquivo simples, tabela relacional, visualização ou sinônimo. É possível importar uma definição de pesquisa a partir de qualquer arquivo simples ou banco de dados relacional ao qual o Cliente do PowerCenter e o Serviço de Integração possam se conectar. Também é possível criar uma definição de pesquisa a partir de um qualificador de origem. Você pode usar várias transformações de Pesquisa em um mapeamento. A transformação de Pesquisa pode ser uma transformação ativa ou passiva. Você pode configurar uma transformação de Pesquisa conectada ou não conectada.

O Serviço de Integração consulta a origem de pesquisa com base nas portas de pesquisa na transformação e uma condição de transformação. A transformação de Pesquisa retorna o resultado da pesquisa ao destino ou a outra transformação. É possível configurar a transformação de pesquisa para que retorne uma única linha ou várias linhas.

Efetue as seguintes tarefas com uma transformação de Pesquisa:

- **Obtenha um valor relacionado.** Recupere um valor a partir da tabela de pesquisa com base em um valor na origem. Por exemplo, a origem apresenta uma ID de funcionário. Recupere o nome do funcionário a partir da tabela de pesquisa.

- **Obtenha valores múltiplos.** Recupere várias linhas a partir de uma tabela de pesquisa. Por exemplo, retorne todos os funcionários de um departamento.
- **Execute um cálculo.** Recupere um valor de uma tabela de pesquisa e use-a em um cálculo. Por exemplo, recupere uma porcentagem de imposto sobre vendas, calcule um imposto e retorne o imposto a um destino.
- **Atualize tabelas de dimensão de alteração lenta.** Determine se as linhas existem em um destino.

Configure a transformação de Pesquisa para que efetue os seguintes tipos de pesquisa:

- **Pesquisa relacional ou de arquivo simples.** Efetue uma pesquisa em um arquivo simples ou em uma tabela relacional. Ao criar uma transformação de Pesquisa usando uma tabela relacional como a origem de pesquisa, você pode conectar-se à origem de pesquisa usando ODBC e importar a definição da tabela como a estrutura para a transformação de Pesquisa. Ao criar uma transformação de Pesquisa com um arquivo simples na qualidade de origem de pesquisa, o Designer invoca o Assistente de Arquivos Simples.
- **Pesquisa no pipeline.** Execute uma pesquisa nas origens de aplicativo, como JMS ou MSMQ. Arraste a origem para o mapeamento e associe a transformação de Pesquisa ao qualificador de origem. Configure partições para aperfeiçoar o desempenho quando o Serviço de Integração recupera dados de origem para o cache de pesquisa.
- **Pesquisa conectada ou não conectada.** Uma transformação de Pesquisa conectada recebe dados de origem, efetua uma pesquisa e retorna dados ao pipeline. Uma transformação de Pesquisa não conectada não está conectada a uma origem ou a um destino. Uma transformação no pipeline chama a transformação de Pesquisa com uma expressão :LKP. A transformação de Pesquisa não conectada retorna uma coluna à transformação de chamada.
- **Pesquisa armazenada ou não em cache.** Armazene a origem de pesquisa em cache para aperfeiçoar o desempenho. Se você armazenar a origem de pesquisa em cache, é possível usar um cache dinâmico ou estático. Por padrão, o cache de pesquisa permanece estático e não muda durante a sessão. Com um cache dinâmico, o Serviço de Integração insere ou atualiza linhas no cache. Ao armazenar em cache a tabela de destino na qualidade de origem de pesquisa, é possível pesquisar valores no cache para determinar se o valor existe no destino. A transformação de Pesquisa marca linhas para inserir ou atualizar o destino.

Tipos de Origem da Pesquisa

Quando você cria uma transformação Pesquisa, pode escolher uma tabela relacional, um arquivo simples ou um qualificador de origem como a origem da pesquisa.

Pesquisas Relacionais

Ao criar uma transformação Pesquisa usando uma tabela relacional como origem de pesquisa, você pode conectar-se à origem de pesquisa usando ODBC e importar a definição da tabela como a estrutura para a transformação Pesquisa.

Use as seguintes opções com pesquisas relacionais:

- Substitua a instrução SQL padrão para adicionar a cláusula WHERE ou para consultar várias tabelas.
- Classifique os dados nulos como altos ou baixos, com base no suporte do banco de dados.
- Execute comparações com diferenciação de maiúsculas e minúsculas com base no suporte do banco de dados.

Pesquisas de arquivos simples

Ao criar uma transformação de Pesquisa usando um arquivo simples na qualidade de origem de pesquisa, selecione uma definição de arquivo simples no repositório ou importe uma origem ao criar a transformação. Ao importar uma origem de pesquisa de arquivo simples, o Designer invoca o Assistente de Arquivos Simples.

Use as seguintes opções com pesquisas de arquivo simples:

- Use arquivos indiretos na qualidade de origens de pesquisa configurando uma lista de arquivos na qualidade de nome de arquivo de pesquisa.
- Use entradas classificadas para a pesquisa.
- Classifique dados nulos como altos ou baixos.
- Use comparação de string que diferencie maiúsculas de minúsculas com pesquisas de arquivos simples.

Usando Entrada Classificada

Quando você configura uma transformação Pesquisa de arquivo simples para uma entrada classificada, as colunas de condição devem estar agrupadas. Se as colunas de condição não estiverem agrupadas, a transformação Pesquisa retornará resultados incorretos. Para o desempenho ideal de armazenamento em cache, classifique as colunas de condição.

Por exemplo, uma transformação Pesquisa contém a seguinte condição:

```
OrderID = OrderID1  
CustID = CustID1
```

Na origem de pesquisa de arquivo simples a seguir, as chaves estão agrupadas, mas não estão classificadas. Embora o Serviço de Integração possa armazenar os dados no cache, talvez o desempenho não seja o ideal.

OrderID	CustID	ItemNo.	ItemDesc	Comments
1001	CA502	F895S	Flashlight	Key data is grouped, but not sorted. CustID is out of order within OrderID.
1001	CA501	C530S	Compass	-
1001	CA501	T552T	Tent	-
1005	OK503	S104E	Safety Knife	Key data is grouped, but not sorted. OrderID is out of order.
1003	CA500	F304T	First Aid Kit	-
1003	TN601	R938M	Regulator System	-

As chaves não estão agrupadas na origem de pesquisa de arquivo simples a seguir: A transformação Pesquisa retorna resultados incorretos.

OrderID	CustID	ItemNo.	ItemDesc	Comments
1001	CA501	T552T	Tent	-

OrderID	CustID	ItemNo.	ItemDesc	Comments
1001	CA501	C530S	Compass	-
1005	OK503	S104E	Safety Knife	-
1003	TN601	R938M	Regulator System	-
1003	CA500	F304T	First Aid Kit	-
1001	CA502	F895S	Flashlight	

Key data for CustID is not grouped.

Se você escolher entrada classificada para arquivos indiretos, o intervalo de dados não deverá sobrepor os arquivos.

Pesquisas no Pipeline

Crie uma transformação Pesquisa no pipeline para realizar pesquisas na origem de um aplicativo que não seja uma tabela relacional nem um arquivo simples. Uma transformação Pesquisa no pipeline tem um qualificador de origem como origem da pesquisa. Você pode executar pesquisas no pipeline em todas as fontes de dados, exceto em transformações de Qualificador de origem de vários grupos de aplicativos.

Quando você configura uma transformação de Pesquisa no pipeline, a origem de pesquisa e o qualificador de origem ficam em outro pipeline. A origem e o qualificador de origem ficam em um pipeline parcial que não contém destinos. O Serviço de Integração lê os dados de origem nesse pipeline e os passa à transformação Pesquisa para criar o cache. Você pode criar diversas partições no pipeline parcial para aumentar o desempenho.

Para aumentar o desempenho ao processar origens de pesquisa de arquivos simples ou relacionais, crie uma transformação Pesquisa no pipeline em vez de uma transformação Pesquisa de arquivo simples ou relacional. Você pode criar partições para processar a origem de pesquisa e passá-la à transformação Pesquisa.

Crie uma transformação Pesquisa no pipeline conectada ou desconectada.

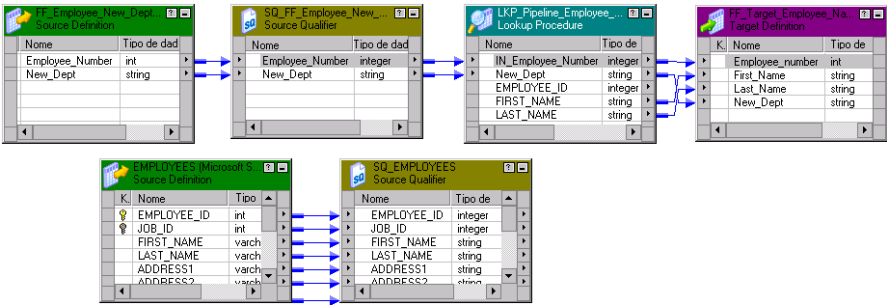
Nota: Não ative a recuperação HA de sessões com origens em tempo real para pesquisas no pipeline. Você pode obter resultados inesperados.

Configurando uma Transformação Pesquisa em Pipeline em um Mapeamento

Um mapeamento que contém a Transformação Pesquisa de pipeline inclui um pipeline parcial que contém a origem da pesquisa e o qualificador da origem. O pipeline parcial não inclui um destino. O Serviço de Integração recupera os dados da origem de pesquisa neste pipeline e os transfere para o cache de pesquisa.

O pipeline parcial está em um grupo de classificação para carregamento de destino separado, nas propriedades da sessão. Você pode criar diversas partições no pipeline para aumentar o desempenho. Você não pode configurar uma classificação para carregamento de destino com o pipeline parcial.

O mapeamento a seguir contém uma Transformação Pesquisa de pipeline e o pipeline parcial que processa a origem da pesquisa:



O mapeamento contém os seguintes objetos:

- A definição da origem da pesquisa e o qualificador da origem estão em um pipeline separado. O Serviço de Integração cria o cache de uma pesquisa depois de processar os dados da origem da pesquisa no pipeline.
- Uma origem de arquivo simples contém novos nomes de departamento por número de funcionários.
- A transformação da Pesquisa de pipeline recebe Employee_Number e New_Dept do arquivo de origem. A Pesquisa de pipeline executa uma pesquisa em Employee_ID no cache de pesquisa. Ela recupera o nome e sobrenome do funcionário do cache de pesquisa.
- Um destino de arquivo simples recebe Employee_ID, First_Name, Last_Name, e New_Dept da transformação da Pesquisa.

Pesquisas Conectadas e Não Conectadas

Você pode configurar uma transformação de Pesquisa conectada ou não conectada. Uma transformação de Pesquisa conectada é uma transformação que tem portas de entrada e saída que você conecta a outras transformações em um mapeamento. Uma transformação de Pesquisa não conectada aparece no mapeamento, mas não está conectada a outras transformações.

Uma transformação de Pesquisa não conectada recebe a entrada do resultado de uma expressão: LKP em uma transformação, como as transformações de Expressão ou de Agregador. A expressão :LKP transmite argumentos para a transformação de Pesquisa, que retorna um resultado. A expressão :LKP pode transmitir os resultados da pesquisa para outra expressão na transformação de Expressão ou de Agregador para filtrar os resultados.

A tabela a seguir lista as diferenças entre pesquisas conectadas e desconectadas:

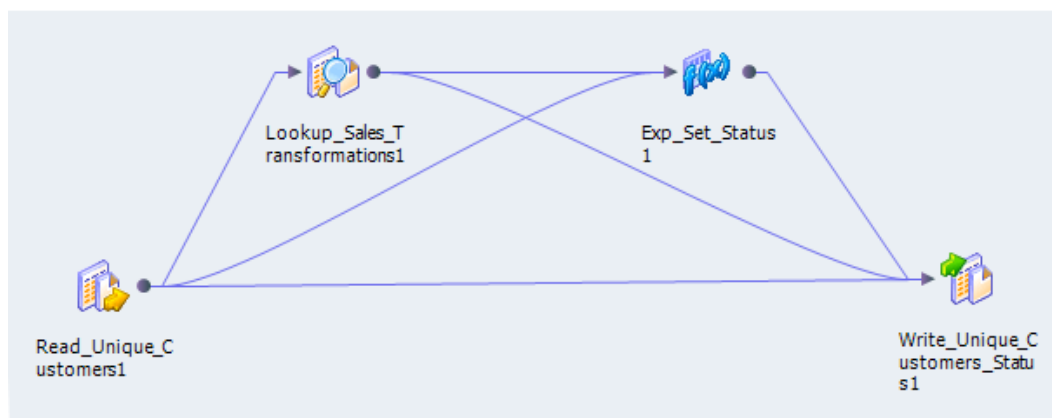
Pesquisa Conectada	Pesquisa Desconectada
Recebe os valores de entrada diretamente do pipeline.	Recebe os valores de entrada do resultado de uma expressão :LKP em outra transformação.
Usar um cache dinâmico ou estático.	Usar um cache estático.
O cache inclui as colunas de origem da pesquisa na condição de pesquisa e as colunas de origem da pesquisa que são portas de saída.	O cache inclui todas as portas de pesquisa e de saída na condição de pesquisa e a porta de pesquisa/retorno.

Pesquisa Conectada	Pesquisa Desconectada
Retorna várias colunas da mesma linha ou insere linhas no cache de pesquisa dinâmica.	Retorna uma coluna de cada linha para uma porta de retorno.
Se não houver correspondência para a condição de pesquisa, o Serviço de Integração retornará o valor padrão para todas as portas de saída. Se você configurar o cache dinâmico, o Serviço de Integração vai inserir linhas no cache ou deixá-lo inalterado.	Se não houver correspondência para a condição de pesquisa, o Serviço de Integração retornará NULL.
Se houver correspondência para a condição de pesquisa, o Serviço de Integração retornará o resultado da condição de pesquisa para todas as portas de pesquisa/saída. Se você configurar o cache dinâmico, o Serviço de Integração atualizará a linha no cache ou a deixará inalterada.	Se uma correspondência ocorrer para a condição de pesquisa, o Serviço de Integração retornará o resultado da condição de pesquisa para a porta de retorno.
Transmite vários valores de saída para outra transformação. Vincular as portas de pesquisa/saída a outra transformação.	Retorna um valor de saída para outra transformação. A porta de retorno da transformação de Pesquisa transmite o valor para a porta que contém a expressão :LKP na outra transformação.
Suporta valores padrão definidos pelo usuário.	Não suporta valores padrão definidos pelo usuário.

Pesquisas Conectadas

Uma transformação de Pesquisa conectada é uma transformação de Pesquisa que está conectada a uma origem ou destino em um mapeamento.

A seguinte figura mostra um mapeamento com uma transformação de Pesquisa conectada:



Quando você executa um mapeamento que contém uma transformação de Pesquisa conectada, o Serviço de Integração realiza as seguintes etapas:

1. O Serviço de Integração transmite valores de outra transformação para as portas de entrada na transformação de Pesquisa.
2. Para cada linha de entrada, o Serviço de Integração consulta o cache ou a origem da pesquisa com base nas portas de pesquisa e na condição de pesquisa na transformação.

3. Se a transformação não estiver armazenada no cache, ou se utilizar um cache estático, o Serviço de Integração retorna valores da consulta de pesquisa.

Se a transformação utilizar um cache dinâmico, o Serviço de Integração insere a linha no cache quando não encontrar a linha no cache. Quando o Serviço de Integração encontrar a linha no cache, ele a atualiza no cache ou a mantém sem alterações. Ele sinaliza a linha como inserção, atualização ou sem alterações.
4. O Serviço de Integração retorna dados da consulta e os transmite para a próxima transformação no mapeamento.

Se a transformação utilizar um cache dinâmico, é possível transferir linhas a uma transformação de filtro ou de roteador para filtrar novas linhas para o destino.

Nota: Este capítulo discorre sobre Transformações Pesquisa Conectada, salvo indicação em contrário.

Pesquisas Não Conectadas

Uma transformação de Pesquisa não conectada é uma transformação de Pesquisa que não está conectada a uma origem ou destino no mapeamento. Chame a pesquisa com uma expressão: LKP em uma transformação que permita expressões.

Um uso comum das transformações Pesquisa não conectada é para a atualização de tabelas de dimensão de alteração lenta. Para obter mais informações sobre tabelas de dimensão de alteração lenta, acesse a Base de Dados de Conhecimento da Informatica em <http://mysupport.informatica.com>.

A sintaxe para a expressão de pesquisa é :LKP lookup_transformation_name(argumento, argumento, ...)

A ordem em que você lista cada argumento deve corresponder à ordem das condições de pesquisa na transformação Pesquisa. A transformação de Pesquisa retorna o resultado da consulta por meio da porta de retorno da transformação de pesquisa. A transformação que chama a pesquisa recebe o valor do resultado da pesquisa na porta que contém a expressão: LKP. Se a consulta de pesquisa não conseguir retornar um valor, a porta receberá um valor nulo.

Ao executar uma pesquisa não conectada, você pode executar a mesma pesquisa várias vezes em um mapeamento. É possível testar os resultados da pesquisa em outra expressão e filtrar linhas com base nos resultados.

Quando você executa um mapeamento que contém uma transformação de Pesquisa não conectada, o Serviço de Integração executa as seguintes etapas:

1. Uma transformação de Pesquisa não conectada recebe valores de entrada do resultado de uma expressão :LKP em outra transformação, como uma transformação de Agregador, uma transformação de Expressão ou uma transformação de Estratégia de Atualização.
2. O Serviço de Integração consulta o cache ou a origem da pesquisa com base nas portas de pesquisa e na condição da transformação de Pesquisa.
3. O Serviço de Integração retorna um valor pela porta de retorno da transformação de Pesquisa.
4. O Serviço de Integração transmite esse valor de retorno à porta que contém a expressão :LKP.

Componentes da Pesquisa

Defina os seguintes componentes quando configurar uma transformação Pesquisa em um mapeamento:

- Origem da pesquisa
- Portas
- Propriedades
- Condição

Origem da Pesquisa

Use um arquivo simples, uma tabela relacional ou um qualificador de origem para uma origem de pesquisa. Ao criar uma transformação de Pesquisa, é possível criar a origem de pesquisa a partir dos locais a seguir:

- Origem relacional ou definição de destino no repositório
- Arquivo simples ou definição de destino no repositório
- Tabela ou arquivo com os quais o mecanismo do Serviço de Integração e do Cliente do PowerCenter conseguem se conectar
- Definição de qualificador de origem em um mapeamento

A tabela de pesquisa pode ser uma tabela única, ou é possível associar várias tabelas no mesmo banco de dados com uma substituição SQL de pesquisa. O Serviço de Integração consulta a tabela de pesquisa ou um cache integrado na memória da tabela em relação a todas as linhas de entrada na transformação de Pesquisa.

O Serviço de Integração consegue se conectar a uma tabela de pesquisa com ODBC ou drivers nativos. Configure os drivers nativos para obter desempenho ideal.

Índices e uma Tabela de Pesquisa

Se tiver privilégios para modificar o banco de dados que contém uma tabela de pesquisa, você poderá melhorar o tempo de inicialização da pesquisa adicionando um índice à tabela de pesquisa. É possível aperfeiçoar o desempenho no caso de tabelas de pesquisa bem grandes. Já que o Serviço de Integração consulta, classifica e compara valores nas colunas de pesquisa, o índice precisa adicionar cada coluna que esteja em condição de pesquisa.

É possível aperfeiçoar o desempenho por meio da indexação dos tipos a seguir:

- **Pesquisas em cache.** É possível aperfeiçoar o desempenho por meio da indexação das colunas na ORDER BY de pesquisa. O log da sessão contém a cláusula ORDER BY.
- **Pesquisas não armazenadas em cache.** Pelo fato de o Serviço de Integração fornecer uma instrução SELECT a cada linha que é transferida para a transformação de Pesquisa, é possível aperfeiçoar o desempenho pela indexação de colunas na condição de pesquisa.

Portas de Pesquisa

A guia Portas contém portas de entrada e saída. A guia Portas também inclui portas de pesquisa que representam as colunas de dados para retornar a partir da origem da pesquisa. Uma transformação Pesquisa desconectada retorna uma coluna de dados para a transformação de chamada nesta porta. Uma transformação Pesquisa desconectada tem uma porta de retorno.

A tabela a seguir descreve os tipos de porta em uma transformação Pesquisa:

Portas	Tipo de pesquisa	Descrição
I	Conectada Não Conectada	Porta de entrada. Crie uma porta de entrada para cada porta de pesquisa que deseja usar na condição de pesquisa. Você deve ter pelo menos uma porta de entrada ou entrada/saída em cada transformação Pesquisa.
O	Conectada Não Conectada	Porta de saída. Crie uma porta de saída para cada porta de pesquisa que você deseja vincular à outra transformação. Você pode designar portas de entrada e de pesquisa como portas de saída. Para as pesquisas conectadas, você deve ter pelo menos uma porta de saída. Para as pesquisas desconectadas, selecione uma porta de pesquisa como uma porta de retorno (R) a fim de transferir um valor de retorno.
L	Conectada Não Conectada	Porta de pesquisa. O Designer designa cada coluna da origem de pesquisa como uma porta de pesquisa (L) e saída (O).
R	Não Conectada	Porta de retorno. Use somente em transformações de Pesquisa não conectadas. Designa a coluna de dados que você deseja retornar com base na condição da pesquisa. Você pode designar uma porta de pesquisa como a porta de retorno.

A transformação Pesquisa também ativa uma propriedade de expressão associada que você configura quando usar um cache dinâmico. A propriedade de expressão associada contém os dados para atualizar o cache de pesquisa. Ela pode conter uma expressão para atualizar o cache dinâmico ou pode conter um nome de porta de entrada.

Use as seguintes diretrizes para configurar portas de pesquisa:

- Se você excluir as portas de pesquisa de uma pesquisa de arquivo simples, a sessão falha.
- Você pode excluir as portas de pesquisa de uma pesquisa relacional se o mapeamento não usá-las. Isto reduz a quantidade de memória de que o Serviço de Integração precisa para executar a sessão.

Propriedades de pesquisa

Na guia Propriedades, configure propriedades, tais como substituição SQL, para fins de pesquisas relacionais, e as propriedades de armazenamento em cache do nome da origem de pesquisa.

TÓPICOS RELACIONADOS:

- [“Propriedades de pesquisa” na página 303](#)

Condição de pesquisa

Na guia Condição, insira as condições que você deseja que o Serviço de Integração use para encontrar os dados na origem da pesquisa.

TÓPICOS RELACIONADOS:

- [“Condição de pesquisa” na página 314](#)

Propriedades de pesquisa

Configure as propriedades de pesquisa, como cache e múltiplas correspondências, na guia Propriedades da Pesquisa. Configure a condição de pesquisa ou as instruções de SQL para consultar a tabela de pesquisa. Você também pode alterar o nome da tabela Pesquisa.

Quando você cria um mapeamento, configura as propriedades para cada transformação Pesquisa. Ao criar uma sessão, você pode substituir propriedades, como o tamanho do índice e do cache de dados em cada transformação.

A tabela a seguir descreve as propriedades da transformação Pesquisa:

Opção	Tipo de Pesquisa	Descrição
Substituição SQL de Pesquisa	Relacional	Substitui a instrução SQL padrão para consultar a tabela de pesquisa. Especifica a instrução SQL que você deseja que o Serviço de Integração utilize para consultar os valores da pesquisa. Use com o cache de pesquisa ativado.
Nome da Tabela de Pesquisa	Pipeline Relacional	O nome da tabela ou do qualificador de origem a partir do qual a transformação efetua a pesquisa e o cache dos valores. Quando você criar a transformação Pesquisa, escolha uma origem, destino ou qualificador de origem como a origem da pesquisa. Você também pode importar uma tabela, exibição ou sinônimo de outro banco de dados quando criar a transformação Pesquisa. Se você inserir uma Substituição SQL de Pesquisa, não precisa inserir o Nome da Tabela de Pesquisa.
Filtro da Origem de Pesquisa	Relacional	Restringe as pesquisas que o Serviço de Integração executa com base no valor dos dados em qualquer porta na transformação Pesquisa. Use com o cache de pesquisa ativado.
Armazenamento em Cache Ativado	Arquivo Simples Pipeline Relacional	Indica se o Serviço de Integração efetua o cache dos valores da pesquisa durante a sessão. Quando você ativa o cache de pesquisa, o Serviço de Integração consulta a origem da pesquisa uma vez, efetua o cache dos valores e procura os valores no cache durante a sessão. Efetuar o cache dos valores de pesquisa pode melhorar o desempenho da sessão. Quando você desabilita o cache, sempre que uma linha é transferida para a transformação, o Serviço de Integração emite uma instrução de seleção para a origem da pesquisa, para os valores de pesquisa. Nota: O Serviço de Integração sempre efetua o cache das pesquisas de arquivos simples e de pipeline.

Opção	Tipo de Pesquisa	Descrição
Diretiva de pesquisa em várias correspondências	Arquivo Simple Pipeline Relacional	<p>Determina quais linhas retornar quando a transformação Pesquisa encontra múltiplas linhas que correspondem à condição de pesquisa. Selecione um dos seguintes valores:</p> <ul style="list-style-type: none"> - Usar Primeiro Valor. Retorna a primeira linha que corresponde à condição de pesquisa. - Usar Último Valor. Retorne a última linha que corresponde à condição de pesquisa. - Usar Todos os Valores. Retorne todas as linhas correspondentes. - Use Qualquer Valor. O Serviço de Integração retorna o primeiro valor que corresponde à condição de pesquisa. Ela cria um índice baseado nas portas-chave ao invés de todas as portas da transformação de Pesquisa. - Relatar Erro. O Serviço de Integração relata um erro e não retorna uma linha. Se você não ativar a opção Valor Antigo da Saída na Atualização, a opção Política de Pesquisa na Correspondência Múltipla é configurada como Relatar Erro para as consultas dinâmicas.
Condição de Pesquisa	Arquivo Simple Pipeline Relacional	Exibe a condição de pesquisa que você configurou na guia Condição.
Informações de Conexão	Relacional	<p>Especifica o banco de dados que contém a tabela de pesquisa. Você pode definir o banco de dados no arquivo de parâmetro, mapeamento ou sessão:</p> <ul style="list-style-type: none"> - Mapeamento. Selecione o objeto de conexão. Você também pode especificar o tipo de conexão do banco de dados. Digite <code>Relational</code>: antes do nome da conexão, caso seja uma conexão relacional. Digite <code>Application</code>: antes do nome da conexão, caso seja uma conexão de aplicativo. - Sessão Use as variáveis de conexão <code>\$Source</code> ou <code>\$Target</code>. Se você usar uma destas variáveis, a tabela de pesquisa deve residir no banco de dados de origem ou de destino. Especifique a conexão do banco de dados para cada variável nas propriedades da sessão. - Arquivo de parâmetros. Use o parâmetro de sessão <code>\$DBConnectionName</code> ou <code>\$AppConnectionName</code>, e o defina no arquivo de parâmetros. <p>Por padrão, o Designer especifica <code>\$Source</code> se você escolher uma tabela de origem e <code>\$Target</code> se escolher uma tabela de destino quando criar a transformação Pesquisa. Você pode substituir esses valores nas propriedades da sessão.</p> <p>O Serviço de Integração provoca a falha da sessão se não puder determinar o tipo de conexão do banco de dados.</p>
Tipo de Origem	Arquivo Simple Pipeline Relacional	Indica que a transformação Pesquisa lê valores de uma tabela relacional, arquivo simples ou qualificador de origem.

Opção	Tipo de Pesquisa	Descrição
Nível de Rastreamento	Arquivo Simple Pipeline Relacional	Configura a quantidade de detalhes incluídos no log da sessão.
Nome do Diretório do Cache de Pesquisa	Arquivo Simple Pipeline Relacional	<p>Especifica o diretório usado para criar os arquivos de cache de pesquisa quando você configura a transformação Pesquisa para efetuar o cache de origem de pesquisa. Também salva os arquivos persistentes do cache de pesquisa quando você seleciona a opção Pesquisa Persistente.</p> <p>Por padrão, o Serviço de Integração usa o diretório \$PMCCacheDir configurado para o Serviço de Integração.</p>
Cache de Pesquisa Persistente	Arquivo Simple Pipeline Relacional	Indica se o Serviço de Integração usa um cache de pesquisa persistente, que consiste em pelo menos dois arquivos de cache. Se uma transformação Pesquisa for configurada para um cache de pesquisa persistente, e os arquivos desse tipo de cache não existirem, o Serviço de Integração cria os arquivos durante a sessão. Use com o cache de pesquisa ativado.
Tamanho do Cache dos Dados de Pesquisa, Tamanho do Cache de Índice da Pesquisa	Arquivo Simple Pipeline Relacional	<p>O padrão é Automático. Indica o tamanho máximo que o Serviço de Integração aloca ao cache de dados e ao índice na memória. É possível usar um valor numérico para o cache, usar um valor de cache proveniente de um arquivo de parâmetros ou configurar o Serviço de Integração para definir o tamanho do cache usando a configuração Automático. Se configurar o Serviço de Integração para determinar o tamanho do cache, você também poderá especificar a área máxima de memória que o Serviço alocará ao cache.</p> <p>Se o Serviço de Integração não puder alocar a quantidade configurada de memória ao inicializar a sessão, ele provocará a sua falha. Quando o Serviço de Integração não pode armazenar todos os dados do cache na memória, ele faz a paginação no disco.</p> <p>Use com o cache de pesquisa ativado.</p>
Cache de Pesquisa Dinâmica	Arquivo Simple Pipeline Relacional	<p>Indica o uso de um cache de pesquisa dinâmica. Insere ou atualiza as linhas no cache de pesquisa à medida que as transfere para a tabela de destino.</p> <p>Use com o cache de pesquisa ativado.</p>
Valor Antigo de Saída na Atualização	Arquivo Simple Pipeline Relacional	<p>Use com o cache dinâmico ativado. Quando você ativa esta propriedade, o Serviço de Integração gera a saída dos valores antigos pelas portas de pesquisa/saída. Quando o Serviço de Integração atualiza uma linha no cache, ele gera a saída do valor que existia no cache de pesquisa antes de atualizar a linha, com base nos dados de entrada. Quando o Serviço de Integração insere uma linha no cache, ele gera a saída de valores nulos.</p> <p>Quando você desativa esta propriedade, o Serviço de Integração gera a saída dos mesmos valores pelas portas de pesquisa/saída e entrada/saída.</p> <p>Essa propriedade é ativada por padrão.</p>

Opção	Tipo de Pesquisa	Descrição
Atualizar Condição de Cache Dinâmico	Arquivo Simples Pipeline Relacional	Uma expressão que indica se o cache dinâmico deve ou não ser atualizado. Crie uma expressão usando as portas de pesquisa ou de entrada. A expressão pode conter valores de entrada ou valores no cache de pesquisa. O Serviço de Integração atualiza o cache quando a condição é verdadeira e o dados existem no cache. Use com o cache dinâmico ativado. O padrão é verdadeiro.
Prefixo do Nome do Arquivo de Cache	Arquivo Simples Pipeline Relacional	Use com o cache de pesquisa persistente. Especifica o prefixo do nome de arquivo para usar nos arquivos de cache de pesquisa persistente. O Serviço de Integração usa o prefixo como nome para os arquivos de cache persistentes que ele salva no disco. Insira o prefixo. Não insira .idx ou .dat. Você pode inserir um parâmetro ou variável para o prefixo do nome de arquivo. Use qualquer tipo de parâmetro ou variável que possa ser definido no arquivo de parâmetro. Se os arquivos de cache persistentes nomeados existirem, o Serviço de Integração cria o cache de memória a partir dos arquivos. Se os arquivos de cache persistentes nomeados não existirem, o Serviço de Integração recria os arquivos de cache persistentes.
Reenviar para cache da Origem de Pesquisa	Arquivo Simples Pipeline Relacional	Use com o cache de pesquisa ativado. Quando selecionado, o Serviço de Integração recria o cache de pesquisa a partir da origem da pesquisa, quando chama pela primeira vez a instância da transformação Pesquisa. Se você usar um cache de pesquisa persistente, ele recria os arquivos de cache persistentes antes de usar o cache. Se você não usar um cache de pesquisa persistente, ele recria o cache de pesquisa na memória antes de usar o cache.
Inserir e Atualizar	Arquivo Simples Pipeline Relacional	Use com o cache dinâmico ativado. Aplica-se a linhas inseridas na transformação Pesquisa com o tipo de linha inserir. Quando ativado, o Serviço de Integração insere linhas no cache e atualiza as linhas existentes; quando desativado, ele não atualiza as linhas existentes.
Atualizar e Inserir	Arquivo Simples Pipeline Relacional	Use com o cache dinâmico ativado. Aplica-se a linhas inseridas na transformação Pesquisa com o tipo de linha atualizar. Quando ativado, o Serviço de Integração atualiza as linhas existentes e insere uma linha se for nova. Quando desativado, o Serviço de Integração não insere linhas novas.
Formato de Data e Hora	Arquivo Simples	Clique no botão Abrir para selecionar um formato de data e hora. Defina o formato e a largura do campo. Os formatos de milissegundos, microssegundos ou nanossegundos têm a largura de campo de 29. Se você não selecionar um formato de data e hora para uma porta, pode inserir qualquer formato de data e hora. O padrão é MM/DD/YYYY HH24:MI:SS. O formato de Data e Hora não altera o tamanho da porta.
Separador de Milhar	Arquivo Simples	Se você não define um separador de milhares para uma porta, o Serviço de Integração usa as propriedades definidas aqui. Você pode escolher nenhum separador, uma vírgula ou um ponto final. O padrão é nenhum separador.

Opção	Tipo de Pesquisa	Descrição
Separador Decimal	Arquivo Simples	Se você não definir um separador decimal para um campo específico na definição de pesquisa ou na guia Portas, o Serviço de Integração usa as propriedades definidas aqui. Você pode escolher um separador decimal de vírgula ou ponto final. O padrão é o ponto final.
Comparação de String com Diferenciação de Maiúscula/Minúscula	Arquivo Simples Pipeline	O Serviço de Integração utiliza comparações de string com diferenciação entre maiúscula e minúscula ao realizar as pesquisas nas colunas de strings. Para pesquisas relacionais, a diferenciação entre maiúscula e minúscula depende do suporte do banco de dados.
Ordenamento Nulo	Arquivo Simples Pipeline	Determina como o Serviço de Integração ordena valores nulos. Você pode optar por classificar valores nulos como altos ou baixos. Por padrão, o Serviço de Integração classifica valores nulos como altos. Isto substitui a configuração do Serviço de Integração para tratar nulos, nos operadores de comparação, como alto, baixo ou nulo. Para pesquisas relacionais, a classificação dos nulos depende do valor padrão do banco de dados.
Entrada Classificada	Arquivo Simples Pipeline	Indica se os dados do arquivo de pesquisa estão ou não na ordem classificada. Isso aumenta o desempenho de pesquisas de arquivo. Se você ativar a entrada classificada, e as colunas de condição não estiverem agrupadas, o Serviço de Integração provoca a falha da sessão. Se as colunas de condição estiverem agrupadas, mas não classificadas, o Serviço de Integração processa a pesquisa como se você não tivesse configurado a entrada classificada.
A Origem da Pesquisa é Estática	Arquivo Simples Pipeline Relacional	A origem da pesquisa não muda em uma sessão.

Opção	Tipo de Pesquisa	Descrição
Cache de Pesquisa Pré-Criado	Arquivo Simples Pipeline Relacional	<p>Permite que o Serviço de Integração crie o cache de pesquisa antes que a transformação Pesquisa receba os dados. O Serviço de Integração pode criar vários arquivos de cache de pesquisa ao mesmo tempo para melhorar o desempenho.</p> <p>Você pode configurar esta opção no mapeamento ou na sessão. O Serviço de Integração usará a configuração de nível de sessão se você configurar a opção de transformação Pesquisa como Automático.</p> <p>Configure uma das seguintes opções:</p> <ul style="list-style-type: none"> - Automático. O Serviço de Integração usa o valor configurado na sessão. - Sempre permitido. O Serviço de Integração pode criar o cache de pesquisa antes que a transformação Pesquisa receba a primeira linha de origem. O Serviço de Integração cria um pipeline adicional para criar o cache. - Nunca permitido. O Serviço de Integração não pode criar o cache de pesquisa antes que a transformação Pesquisa receba a primeira linha. <p>Configure o número de pipelines que o Serviço de Integração pode criar simultaneamente. Configure os Pipelines Simultâneos Adicionais para a propriedade de sessão Criação do Cache de Pesquisa. O Serviço de Integração poderá pré-criar o cache de pesquisa se esta propriedade for maior que zero.</p>
Precisão de Subsegundos	Relacional	<p>Especifica a precisão de subsegundo das portas de data e hora.</p> <p>Em pesquisas relacionais, você pode alterar a precisão de bancos de dados que tenham uma escala editável para dados de data e hora. Você pode alterar a precisão de subsegundos para tipos de dados Oracle Timestamp, Informix Datetime e Teradata Timestamp.</p> <p>Selecione um valor inteiro positivo de 0 a 9. O padrão é 6 microssegundos. Se você ativar a otimização de empilhamento, o banco de dados retorna o valor de data e hora completo, independente da configuração da precisão do subsegundo.</p>

Configurando Propriedades de Pesquisa em uma Sessão

Ao configurar uma sessão, é possível configurar propriedades de pesquisa exclusivas às sessões:

- **Pesquisas de arquivos simples.** Configure informações de localização de pesquisa, como diretório do arquivo de origem, nome do arquivo e tipo de arquivo.
- **Pesquisas relacionais.** É possível definir variáveis \$Origem e \$Destino nas propriedades da sessão. Também é possível substituir informações de conexão para usar o parâmetro de sessão \$DBConnectionName ou \$AppConnectionName.
- **Pesquisas em pipeline.** Configure as propriedades do arquivo de origem de pesquisa, como diretório do arquivo de origem, nome de arquivo e tipo de arquivo. Se a origem é uma tabela relacional ou origem de aplicativo, configure as informações de conexão.

Configurando Pesquisas de arquivos simples em uma sessão

Quando você configura uma pesquisa de arquivo simples em uma sessão, configure as propriedades do arquivo de origem da pesquisa na Exibição de Transformação da guia Mapeamento. Escolha a transformação Pesquisa e configure as propriedades do arquivo simples nas propriedades da sessão para a transformação.

A tabela a seguir descreve as propriedades de sessão que você configura para pesquisas de arquivo simples:

Propriedade	Descrição
Diretório do Arquivo da Origem de Pesquisa	Insira o nome do diretório. Por padrão, o Serviço de Integração procura no diretório da variável do processo, \$PMLookupFileDir, os arquivos da pesquisa. Você pode inserir o caminho completo e o nome do arquivo. Se você especificar o diretório e o nome do arquivo no campo Nome do Arquivo da Origem de Pesquisa, desmarque este campo. O Serviço de Integração concatena este com o campo Nome do Arquivo da Origem de Pesquisa quando executa a sessão. Você também pode inserir o parâmetro de sessão \$InputFileName para configurar o nome do arquivo.
Nome do Arquivo da Origem de Pesquisa	Nome do arquivo de pesquisa. Se você usar um arquivo indireto, insira o nome do arquivo indireto que o Serviço de Integração deverá ler. Você também pode inserir o parâmetro do arquivo de pesquisa, \$LookupFileName, para alterar o nome do arquivo de pesquisa para a sessão. Se você configurar o diretório e o nome do arquivo no campo Diretório do Arquivo de Pesquisa, desmarque Nome do Arquivo da Origem de Pesquisa. O Serviço de Integração concatena o Nome do Arquivo da Origem de Pesquisa com o campo Diretório do Arquivo da Origem de Pesquisa para a sessão. Por exemplo, o campo Diretório do Arquivo da Origem de Pesquisa contém "C:\lookup_data\" e o campo Nome do Arquivo da Origem de Pesquisa contém "filename.txt." Quando o Serviço de Integração começa a sessão, ele procura pelo "C:\lookup_data\filename.txt."
Tipo de Filtro da Origem de Pesquisa	Indica se o arquivo de origem da pesquisa contém os dados de origem ou uma lista de arquivos com as mesmas propriedades. Escolha Direto se o arquivo de origem da pesquisa contém os dados de origem. Escolha Indireto se o arquivo de origem da pesquisa contém uma lista de arquivos. Quando você seleciona Indireto, o Serviço de Integração cria um cache para todos os arquivos. Se você usar a entrada classificada com os arquivos indiretos, verifique se o intervalo de dados nos arquivos não se sobrepõe. Se isso ocorrer, o Serviço de Integração processa a pesquisa como uma origem de pesquisa não classificada.

Configurando Pesquisas Relacionais em uma Sessão

Quando você configurar uma pesquisa relacional em uma sessão, configure a conexão do banco de dados de pesquisa na Exibição de Transformação da guia Mapeamento. Escolha a transformação Pesquisa e configure a conexão nas propriedades da sessão para a transformação.

Escolha entre as seguintes opções para configurar uma conexão para uma transformação Pesquisa relacional:

- Escolha uma conexão relacional ou de aplicativo.
- Configure uma conexão de banco de dados usando a variável de conexão \$Source ou \$Target.
- Configure o parâmetro de sessão \$DBConnectionName ou \$AppConnectionName e o defina em um arquivo de parâmetro.

Configurando Pesquisas de Pipeline em uma Sessão

Quando você configura uma Pesquisa de pipeline em uma sessão, configure a localização do arquivo de origem da pesquisa ou a conexão da tabela de pesquisa no nó Origens da guia Mapeamento. Escolha o Qualificador da origem que represente a origem da pesquisa.

Consulta de Pesquisa

O Serviço de Integração consulta a pesquisa com base nas portas e nas propriedades que você configura na transformação de Pesquisa. O Serviço de Integração executa uma consulta de pesquisa padrão quando a primeira linha é inserida na transformação de Pesquisa.

Se você usar uma pesquisa relacional ou uma pesquisa de pipeline em relação a uma tabela relacional, poderá substituir a consulta de pesquisa. Se você usar uma pesquisa em relação a uma tabela relacional, poderá substituir a consulta de pesquisa. Você pode usar a substituição para alterar a cláusula ORDER BY, adicionar uma cláusula WHERE ou transformar os dados de pesquisa antes que sejam armazenados em cache. É possível usar a substituição para adicionar uma cláusula WHERE ou transformar os dados de pesquisa antes que eles sejam armazenados em cache.

Se você configurar uma substituição SQL e um filtro na consulta de pesquisa, o Serviço de Integração ignorará esse filtro.

Consulta de Pesquisa Padrão

A consulta de pesquisa padrão contém as seguintes instruções:

SELECT

A instrução SELECT inclui todas as portas de pesquisa no mapeamento. Para exibir a instrução SELECT da consulta de pesquisa, selecione a propriedade Substituição SQL de Pesquisa.

SELECT

A instrução SELECT inclui todas as portas de pesquisa no mapeamento. Para exibir a instrução SELECT da consulta de pesquisa, selecione a propriedade Usar Consulta Personalizada.

ORDER BY

A cláusula ORDER BY organiza as colunas na mesma ordem em que elas aparecem na transformação de Pesquisa. O Serviço de Integração gera a cláusula ORDER BY. Não é possível ver isso quando você gera o SQL padrão.

Substituindo a Consulta de Pesquisa

A Substituição SQL de Pesquisa é semelhante a inserir uma consulta personalizada em uma transformação de Consulta de Origem. É possível substituir a consulta de pesquisa para uma pesquisa relacional. É possível inserir toda a substituição, ou gerar e editar a instrução SQL padrão. Quando o Designer gera a instrução SQL padrão para a Substituição SQL de Pesquisa, ele inclui as portas de pesquisa/saída na condição de pesquisa e na porta de pesquisa/retorno.

Você pode inserir uma consulta de pesquisa diferente ou editar a consulta de pesquisa existente, que inclui as portas de pesquisa, as portas de saída e a porta de retorno.

Substituindo uma Cláusula ORDER BY

Por padrão, o Serviço de Integração gera uma cláusula ORDER BY para uma pesquisa em cache. A cláusula ORDER BY contém todas as portas de condição de pesquisa. Para aumentar o desempenho, você pode suprimir a cláusula padrão ORDER BY e inserir uma ORDER BY de substituição com menos colunas.

Nota: O SQL de substituição deve retornar dados classificados nas chaves de pesquisa. Quando a transformação recuperar todas as linhas em uma pesquisa, o Serviço de Integração criará o cache de dados com as chaves em ordem classificada. O Serviço de Integração não poderá recuperar todas as linhas do cache se essas linhas não estiverem classificadas. Se os dados não forem classificados nas chaves, você poderá obter resultados inesperados.

Se você usar a otimização de empilhamento, não é possível substituir a cláusula ORDER BY ou suprimir a cláusula gerada ORDER BY por uma notação de comentário.

O Serviço de Integração sempre gera uma cláusula ORDER BY, mesmo se você inserir outra na substituição. Coloque dois hífenes '--' depois substituição ORDER BY para suprimir a cláusula gerada ORDER BY. Por exemplo, uma transformação de Pesquisa usa a seguinte condição de pesquisa:

```
ITEM_ID = IN_ITEM_ID  
  
PRICE <= IN_PRICE
```

A transformação de Pesquisa inclui duas portas de condição de pesquisa usadas no mapeamento, ITEM_ID e PRICE. Quando você inserir a cláusula ORDER BY com uma ou mais colunas, digite as colunas na mesma ordem que as portas na condição de pesquisa. Você também deve incluir todas as palavras reservadas do banco de dados entre aspas. Insira a seguinte consulta de pesquisa na substituição SQL da pesquisa:

```
SELECT ITEMS_DIM.ITEM_NAME AS ITEM_NAME, ITEMS_DIM.PRICE AS PRICE, ITEMS_DIM.ITEM_ID AS  
ITEM_ID FROM ITEMS_DIM ORDER BY ITEMS_DIM.ITEM_ID --
```

Para substituir a cláusula padrão ORDER BY para uma pesquisa relacional, conclua as etapas a seguir:

1. Gere a consulta de pesquisa na transformação Pesquisa.
2. Insira uma cláusula ORDER BY que contenha as portas de condição na mesma ordem em que aparecem na condição da Pesquisa.
3. Coloque dois hífenes '--' como uma anotação de comentário depois da cláusula ORDER BY, para suprimir a cláusula ORDER BY que o Serviço de Integração gera.

Se você substituir a consulta de pesquisa por uma cláusula ORDER BY sem adicionar a anotação de comentário, a pesquisa falha.

Nota: O Sybase tem uma limitação de ORDER BY de 16 colunas. Se a transformação Pesquisa tiver mais de 16 portas de pesquisa/saída, incluindo inclusive as portas na condição de pesquisa, substitua a cláusula ORDER BY ou use múltiplas transformações Pesquisa para consultar a tabela de pesquisa.

Palavras reservadas

Se algum nome de pesquisa ou de coluna contiver uma palavra reservada por banco de dados, como MONTH ou YEAR, a sessão será encerrada com erros do banco de dados quando o Serviço de Integração executar o SQL no banco de dados. Você pode criar e manter um arquivo de palavras reservadas, reswords.txt, no diretório de instalação do Serviço de Integração.

Você pode criar e manter um arquivo de palavras reservadas, o reswords.txt, no diretório de instalação do Serviço de Integração. Quando o Serviço de Integração inicializa uma sessão, ele pesquisa o arquivo reswords.txt, coloca as palavras reservadas entre aspas e executa o SQL nos bancos de dados de pesquisa, origem e destino.

Você pode criar e manter um arquivo de palavras reservadas, o reswords.txt, no diretório de instalação do Serviço de Integração. Quando o Serviço de Integração inicializa um mapeamento, ele pesquisa o arquivo reswords.txt, coloca as palavras reservadas entre aspas e executa o SQL nos bancos de dados de pesquisa, origem e destino.

Talvez seja necessário ativar alguns bancos de dados, como o Microsoft SQL Server e o Sybase, para usar padrões SQL-92 no que diz respeito a identificadores entre aspas. Use SQL do ambiente de conexão para emitir o comando. Por exemplo, com o Microsoft SQL Server, use o seguinte comando:

Talvez seja necessário ativar alguns bancos de dados, como o Microsoft SQL Server e o Sybase, para usar padrões SQL-92 no que diz respeito a identificadores entre aspas. Use o SQL do ambiente para emitir o comando. Por exemplo, com o Microsoft SQL Server, use o seguinte comando:

```
SET QUOTED_IDENTIFIER ON
```

Orientações para Substituir a Consulta de Pesquisa

Certas regras e diretrizes são aplicáveis quando você substitui uma consulta de pesquisa.

Considere as seguintes diretrizes ao substituir a consulta SQL de pesquisa:

- É possível substituir a consulta SQL de pesquisa para pesquisas relacionais.
- Gere a consulta padrão e, em seguida, configure a substituição. Isso garante que todas as portas de pesquisa/saída estão incluídas na consulta. Caso você adicione ou subtraia portas da instrução SELECT, haverá falha na sessão.
- Adicione um filtro de pesquisa de origem para filtrar as linhas que são adicionadas ao cache de pesquisa. Isso garante que o Serviço de Integração insira ao cache dinâmico e à tabela de destino linhas que coincidam com a cláusula WHERE.
- Se várias transformações de Pesquisa compartilharem um cache de pesquisa, use a mesma substituição SQL de pesquisa para cada uma dessas transformações.
- Quando você configura uma transformação de Pesquisa que retorna todas as linhas, o Serviço de Integração cria o cache de pesquisa com chaves classificadas. Quando a transformação recupera todas as linhas em uma pesquisa, o Serviço de Integração cria o cache de dados com as chaves na ordem de classificação. O Serviço de Integração não poderá recuperar todas as linhas do cache se elas não estiverem classificadas. Se os dados não estiverem classificados nas chaves, você poderá obter resultados inesperados.
- A cláusula ORDER BY deve conter as portas de condição na mesma ordem em que elas aparecem na condição de Pesquisa.
- Se você substituir a cláusula ORDER BY, use a notação de comentário para suprimir a cláusula ORDER BY que a transformação de Pesquisa gera.
- Se usar a otimização de empilhamento, você não poderá substituir a cláusula ORDER BY ou suprimir a cláusula ORDER BY gerada com notação de comentário.
- Se um nome de tabela ou coluna na consulta de pesquisa contiver uma palavra reservada, coloque essa palavra entre aspas.
- Para substituir a consulta de pesquisa para uma pesquisa sem cache, opte por retornar qualquer valor quando o Serviço de Integração encontrar várias correspondências.
- Não é possível adicionar ou excluir nenhuma das colunas da instrução SQL padrão.
- A Developer tool não valida a sintaxe da consulta SQL. Se a substituição SQL em uma consulta de pesquisa desconectada não for válida, o mapeamento falhará.

Etapas para Substituir a Consulta da Pesquisa

Use as seguintes etapas para substituir a consulta SQL de pesquisa padrão:

1. Na guia Propriedades, abra o Editor SQL de dentro do campo Substituição SQL de Pesquisa.
2. Clique em Gerar SQL para gerar a instrução SELECT padrão. Insira a Substituição SQL de Pesquisa.
3. Conecte a um banco de dados e clique em Validar para testar a Substituição SQL de Pesquisa.
4. Clique em OK para retornar à guia Propriedades.

Substituição SQL para Pesquisas sem Cache

Você pode definir uma substituição SQL para pesquisas sem cache. O Serviço de Integração não cria um cache da instrução de substituição para uma pesquisa sem cache. Você pode usar funções SQL na instrução SELECT de substituição. Você pode substituir todas as consultas SQL, incluindo as cláusulas WHERE e ORDER BY.

Quando você gera a instrução SELECT padrão, o Designer gera uma instrução SELECT que inclui a pesquisa e portas de saída e a cláusula WHERE, com base na condição de pesquisa. Se a transformação Pesquisa for uma pesquisa desconectada, a instrução SELECT incluirá as portas de pesquisa e a porta de retorno. O Serviço de Integração não gera a cláusula WHERE a partir da condição configurada na guia Condição da transformação Pesquisa.

Cada coluna na consulta SELECT usa um alias para definir a coluna de saída. Não altere a sintaxe na instrução SQL, senão a consulta falhará. Para fazer referência a portas de entrada na cláusula WHERE, configure a associação de parâmetros. O exemplo a seguir inclui uma instrução WHERE que faz referência à porta Nome:

```
SELECT EMPLOYEE.NAME as NAME, max(EMPLOYEE.ID) as ID from EMPLOYEE WHERE EMPLOYEE.NAME=?  
NAME1?
```

O Editor SQL para pesquisas sem cache mostra as portas de entrada e de pesquisa na guia Portas.

Se você adicionar uma função à instrução SQL, o tipo de dados de retorno devem coincidir com os tipos de dados da coluna ALIAS. Por exemplo, o tipo de dados de ID coincide com o tipo de retorno da função MAX:

```
SELECT EMPLOYEE.NAME as NAME, MAX(EMPLOYEE.ID) as ID FROM EMPLOYEE
```

Nota: Você não pode usar subconsultas na substituição SQL para pesquisas não em cache.

Filtro da Origem de Pesquisa

Você pode configurar um Filtro da Origem de Pesquisa para uma transformação Pesquisa relacional que tenha o cache ativado. Adicione o filtro de origem de pesquisa para limitar o número de pesquisas que o Serviço de Integração realiza em uma tabela de origem da pesquisa.

Quando você configurar um Filtro da Origem de Pesquisa, o Serviço de Integração executa pesquisas com base nos resultados da instrução do filtro. Por exemplo, talvez você precise recuperar o sobrenome de cada funcionário com um ID maior que 510.

Você configura o seguinte filtro de origem de pesquisa na coluna EmployeeID:

```
EmployeeID >= 510
```

EmployeeID é uma porta de entrada na transformação Pesquisa. Quando o Serviço de Integração lê a linha de origem, ele executa uma pesquisa no cache quando o valor de EmployeeID é maior que 510. Quando EmployeeID for menor ou igual a 510, a transformação Pesquisa não recupera o sobrenome.

Quando você adiciona um filtro de origem de pesquisa à consulta de Pesquisa para um mapeamento configurado para a otimização de empilhamento, o Serviço de Integração cria uma exibição para representar a substituição SQL. O Serviço de Integração executa uma consulta de SQL contra esta exibição, para forçar a lógica de transformação para o banco de dados.

Quando você adiciona um Filtro da Origem de Pesquisa à consulta da Pesquisa para uma sessão configurada para a otimização de empilhamento, o Serviço de Integração criará uma exibição para representar a substituição SQL. O Serviço de Integração executa uma consulta de SQL contra esta exibição, para forçar a lógica de transformação para o banco de dados.

Filtrando Linhas de Origem de Pesquisa

Adicione o filtro de origem de pesquisa para limitar o número de pesquisas que o Serviço de Integração realiza em uma origem de pesquisa relacional.

1. No Mapping Designer ou Transformation Developer, abra a transformação de Pesquisa.
2. Selecione a guia **Propriedades**.
3. Verifique se o cache está ativado.

4. Clique no botão **Abrir** no campo **Filtro de Origem de Pesquisa**.
5. No SQL Editor, selecione as portas de entrada ou insira qualquer porta da transformação de Pesquisa que você deseja filtrar.
6. Insira uma condição de filtro.
Não inclua a palavra-chave WHERE na condição do filtro. Inclua os parâmetros e as variáveis de mapeamento de string nos identificadores de string.
7. Para validar a condição, selecione a fonte de dados ODBC que tem a origem incluída na consulta.
8. Insira o nome de usuário e a senha para conectar a este banco de dados.
9. Clique em **Validar**.
O Designer executa a consulta e relata se a sua sintaxe está correta.

Condição de pesquisa

O Serviço de Integração encontra dados na origem da pesquisa com uma condição de pesquisa. A condição de pesquisa é semelhante à cláusula WHERE em uma consulta SQL. Quando você configura uma condição de pesquisa em uma transformação Pesquisa, compara o valor de uma ou mais colunas nos dados de origem com os valores na origem da pesquisa ou cache.

Por exemplo, o dados de origem contêm um employee_number. A tabela de origem da pesquisa contém employee_ID, first_name e last_name. Configure a seguinte condição de pesquisa:

```
employee_ID = employee_number
```

Para cada employee_number, o Serviço de Integração retorna a coluna employee_ID, last_name e first_name a partir da origem da pesquisa.

O Serviço de Integração pode retornar mais de uma linha da origem da pesquisa. Configure a seguinte condição de pesquisa:

```
employee_ID > employee_number
```

O Serviço de Integração retorna linhas para todos os números employee_ID maiores que o número do funcionário de origem.

Use as seguintes orientações quando inserir uma condição para uma transformação Pesquisa:

- Os tipos de dados para as colunas em uma condição de pesquisa devem corresponder.
- Você deve inserir uma condição de pesquisa em todas as transformações Pesquisa.
- Use uma porta de entrada para cada porta de pesquisa na condição de pesquisa. Use a mesma porta de entrada em mais de uma condição em uma transformação.
- Quando você insere múltiplas condições, o Serviço de Integração avalia cada condição como um AND, e não como um OR. O Serviço de Integração retorna linhas que correspondem a todas as condições que você configurar.
- Se você incluir múltiplas condições, insira-as na seguinte ordem para otimizar o desempenho da pesquisa:
 - Igual a (=)
 - Menor que (<), maior que (>), menor ou igual a (<=), maior ou igual a (>=)
 - Diferente de (!=)

- O Serviço de Integração corresponde valores nulos. Por exemplo, se uma coluna da condição da pesquisa de entrada é NULL, o Serviço de Integração avalia o NULL igual a um NULL na pesquisa.
- Se você configurar uma pesquisa de arquivo simples para a entrada classificada, o Serviço de Integração provoca a falha da sessão se as colunas de condição não estiverem agrupadas. Se as colunas estiverem agrupadas, mas não classificadas, o Serviço de Integração processa a pesquisa como se você não houvesse configurado a entrada classificada.

O Serviço de Integração processa as correspondências da pesquisa de maneira diferente, dependendo do fato de você ter configurado a transformação para um cache dinâmico, estático ou sem cache.

Cache Estático ou Sem Cache

Use as seguintes diretrizes ao configurar transformações Pesquisa que tenham cache de pesquisa estático ou uma origem de pesquisa sem cache:

- Use os seguintes operadores ao criar a condição de pesquisa:

=, >, <, >=, <=, !=

Se você incluir mais de uma condição de pesquisa, coloque-as na seguinte ordem para otimizar o desempenho de pesquisa:

- Igual a (=)
- Menor que (<), maior que (>), menor ou igual a (<=), maior ou igual a (>=)
- Diferente de (!=)

Por exemplo, crie a seguinte condição de pesquisa:

```
ITEM_ID = IN_ITEM_ID
PRICE <= IN_PRICE
```

- O valor de entrada deve corresponder a todas as condições para que a pesquisa retorne um valor.

A condição pode corresponder a valores equivalentes ou suprir uma condição limite. Por exemplo, você poderá procurar clientes que não moram na Califórnia ou funcionários com salário maior que US\$ 30 mil. Dependendo da natureza da origem e da condição, a pesquisa poderá retornar diversos valores.

Cache Dinâmico

Se você configurar uma transformação Pesquisa para usar um cache dinâmico, pode usar somente o operador de igualdade (=) na condição da pesquisa.

Manuseando Várias Correspondências

A Transformação Pesquisa encontra valores baseados na condição configurada na transformação. Se a condição de pesquisa não for baseada em uma chave exclusiva, ou se a origem de pesquisa for desordenada, o Serviço de Integração pode encontrar várias correspondências na origem de pesquisa ou no cache de pesquisa.

É possível configurar uma Transformação Pesquisa para tratar de várias correspondências da seguinte forma:

- **Use o primeiro valor de correspondência, ou use o último valor de correspondência.** É possível configurar a transformação para que retorne o primeiro valor de correspondência ou o último valor de correspondência. O primeiro e o último valores são o primeiro e o último valores encontrados no cache de pesquisa que combinam com as condições de pesquisa. Ao armazenar em cache a origem de pesquisa, o Serviço de Integração gera uma cláusula ORDER BY para cada coluna no cache de pesquisa para determinar a primeira e a última linhas no cache. Em seguida, o Serviço de Integração classifica cada coluna de origem de pesquisa em ordem crescente.

O Serviço de Integração classifica as colunas numéricas em ordem numérica ascendente, como de 0 a 10. Classifica as colunas de data/hora de janeiro a dezembro e desde o primeiro dia do mês até o último. O Serviço de Integração classifica as colunas de string com base na ordem de classificação configurada para a sessão.
- **Use qualquer valor de correspondência.** É possível configurar a Transformação Pesquisa para retornar qualquer valor que combine com a condição de pesquisa. Ao configurar a Transformação Pesquisa para retornar qualquer valor de correspondência, a transformação retorna o primeiro valor que combine com a condição de pesquisa. A transformação cria um índice com base nas portas-chave ao invés de todas as Portas de Pesquisa. Ao usar qualquer valor de correspondência, pode haver aumento de desempenho, uma vez que o processo de indexação de filas é mais simples.
- **Usar todos os valores.** A Transformação Pesquisa retorna todas as linhas que combinam. Para usar esta opção, é necessário configurar a Transformação Pesquisa para retornar todas as correspondências ao se criar a transformação. A transformação torna-se uma transformação ativa. Você não pode alterar o modo entre passiva e ativa depois de criar a transformação.
- **Retorno de um erro.** Quando a Transformação Pesquisa usa um cache estático ou nenhum cache, o Serviço de Integração marca a linha como erro. Por padrão, a Transformação Pesquisa grava a linha no log de sessão e aumenta a contagem de erros em um. Quando a Transformação Pesquisa tem um cache dinâmico, o Serviço de Integração falha com a sessão quando encontra várias correspondências. A sessão falha enquanto o Serviço de Integração está armazenando em cache a tabela de pesquisa ou pesquisando os valores-chave duplicados. Além disso, se você configurar a Transformação Pesquisa para fornecer valores antigos sobre atualizações, a Transformação Pesquisa retornará um erro ao encontrar várias correspondências. A transformação cria um índice com base nas portas-chave em vez de todas as Portas da transformação de Pesquisa.

TÓPICOS RELACIONADOS:

- [“Retornar Várias Linhas” na página 317](#)

Caches de Pesquisa

É possível configurar uma transformação de Pesquisa para armazenar o arquivo ou tabela de pesquisa em cache. O Serviço de Integração cria um cache na memória ao processar a primeira linha dos dados em uma transformação de Pesquisa em cache. Ele aloca memória para o cache com base na quantidade configurada nas propriedades de transformação ou de sessão. O Serviço de Integração armazena valores de condição no cache de índice e valores de saída no cache de dados. O Serviço de Integração consulta o cache a cada linha que entra na transformação.

O Serviço de Integração também cria arquivos de cache por padrão no \$PMCacheDir. Se os dados não se encaixarem no cache de memória, o Serviço de Integração armazena os valores excedentes nos arquivos de cache. Quando a sessão for concluída, o Serviço de Integração libera a memória de cache e exclui os

arquivos do cache, a menos que você configure a transformação de Pesquisa para que use um cache persistente.

Ao configurar um cache de pesquisa, é possível configurar as opções a seguir:

- Cache persistente
- Reenviar para cache a partir de origem de pesquisa
- Cache estático
- Cache dinâmico
- Cache compartilhado
- Cache de pesquisa pré-criado

Nota: É possível usar um cache dinâmico para pesquisas de arquivo relacional ou simples.

Retornar Várias Linhas

Quando você configurar uma transformação Pesquisa para retornar todas as linhas correspondentes, essa transformação retornará todas as linhas que corresponderem à condição de pesquisa. Ao criar a transformação, você deverá configurá-la para retornar todas as linhas correspondentes. A transformação de Pesquisa se torna ativa. A Diretiva de Pesquisa na propriedade Várias Correspondências é Usar todos os Valores A propriedade torna-se somente leitura. Depois de criar a transformação, você não poderá alterar a propriedade.

Você precisa ter dados de pedidos de clientes em uma tabela relacional. Cada cliente tem vários pedidos na tabela. Você pode configurar a transformação Pesquisa para retornar todos os pedidos de um cliente em uma pesquisa. Para melhorar o desempenho, você pode armazenar a tabela de pesquisa em cache.

Se você configurar a transformação Pesquisa para armazenamento em cache, o Serviço de Integração armazenará em cache todas as linhas que ele ler na origem da pesquisa. O Serviço de Integração armazena em cache todas as linhas de uma chave de pesquisa pelo índice chave.

Regras e Diretrizes para Retornar Várias Linhas

Use as regras e diretrizes a seguir quando configurar a transformação Pesquisa para retornar várias linhas:

- O Serviço de Integração guarda todas as linhas da origem da pesquisa para pesquisas em cache.
- Você pode configurar uma substituição SQL para uma pesquisa com ou sem cache que retorna várias linhas. O SQL de substituição deve retornar dados classificados nas chaves de pesquisa. Se os dados não forem classificados nas chaves, você poderá obter resultados inesperados.
- Não é possível ativar o cache dinâmico para uma transformação Pesquisa que retorne várias linhas.
- Não é possível retornar várias linhas de uma transformação Pesquisa desconectada.
- Você poderá configurar várias transformações de Pesquisa para compartilhar um cache nomeado se as transformações de Pesquisa tiverem pesquisas de cache correspondentes em várias diretivas de correspondência.
- Uma transformação de Pesquisa que retorna várias linhas não pode compartilhar um cache com outra que retorna uma linha correspondente para cada linha de entrada.

TÓPICOS RELACIONADOS:

- [“Criando uma transformação Pesquisa” na página 321](#)

Configurando Transformações de Pesquisa Não Conectadas

Uma Transformação Pesquisa não conectada é uma Transformação Pesquisa que não está conectada a uma origem ou destino. Chame a pesquisa a partir de outra transformação com uma expressão :LKP.

Você pode executar as seguintes tarefas quando chamar uma pesquisa a partir de uma expressão:

- Teste os resultados de uma pesquisa em uma expressão.
- Filtre as linhas com base nos resultados da pesquisa.
- Marque as linhas para a atualização com base no resultado de uma pesquisa e atualize as tabelas de dimensão de alteração lenta.
- Chame a mesma pesquisa múltiplas vezes em um mapeamento.

Para configurar uma transformação de Pesquisa não conectada, adicione portas de entrada, configure a condição de pesquisa, designe um valor de retorno e configure uma expressão de pesquisa em outra transformação.

Etapa 1. Adicionar Portas de Entrada

Crie uma porta de entrada na transformação Pesquisa de cada argumento na expressão :LKP. Para cada condição de pesquisa que planeja criar, você precisará adicionar uma porta de entrada à transformação Pesquisa. Você pode criar outra porta para cada condição ou usar a mesma porta de entrada em mais de uma condição.

Por exemplo, uma loja de varejo aumentou os preços em todos os departamentos no último mês. O departamento de contabilidade só quer carregar no destino as linhas referentes aos itens que tiveram aumento de preço. Para fazer isso, execute as seguintes tarefas:

- Crie uma condição de pesquisa que compare ITEM_ID na origem com ITEM_ID no destino.
- Compare o preço de cada item na origem com o preço na tabela de destino.
 - Se o item existir na tabela de destino e o preço dele na origem for menor ou igual ao da tabela de destino, você desejará excluir a linha.
 - Se o preço do item na origem for maior que na tabela de destino, você desejará atualizar a linha.
- Crie uma porta de entrada (IN_ITEM_ID) com o tipo de dados Decimal (37,0) para comparar com o ITEM_ID e uma porta de entrada IN_PRICE com Decimal (10,2) para comparar com a porta de pesquisa PRICE.

Etapa 2. Adicionar a Condição de Pesquisa

Depois de definir as portas, defina uma condição de pesquisa para comparar valores de entrada da transformação no cache ou origem da pesquisa. Para aumentar o desempenho, adicione condições com um sinal de igual na frente.

Nesse caso, adicione esta condição de pesquisa:

```
ITEM_ID = IN_ITEM_ID
```

```
PRICE <= IN_PRICE
```

Se o item existir na origem do mapeamento e origem da pesquisa e o preço da origem do mapeamento for menor ou igual ao preço da pesquisa, a condição será verdadeira e a pesquisa retornará valores designados pela porta Retorno. Se a condição de pesquisa for falsa, a pesquisa retornará NULL. Quando você gravar a expressão da estratégia de atualização, use ISNULL aninhado em uma função IIF para testar valores nulos.

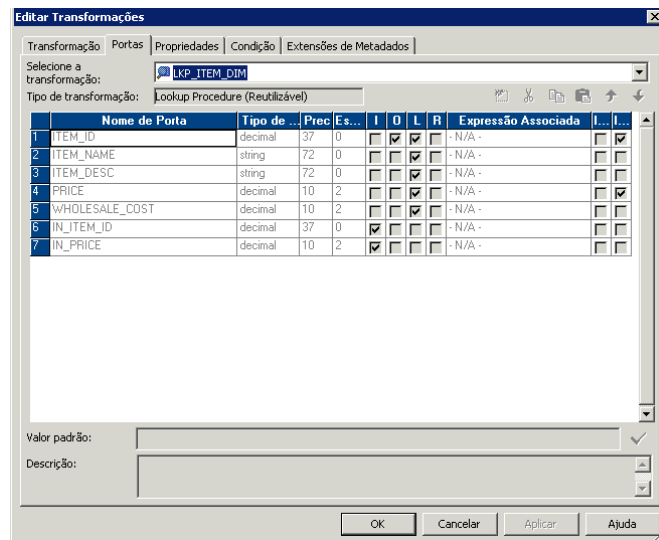
Etapa 3. Designar um valor de retorno

Você pode passar diversos valores de entrada para uma transformação Pesquisa e retornar uma coluna de dados. Não configure a diretiva de pesquisa para usar todos os valores em várias correspondências para uma transformação Pesquisa não conectada. Designe uma porta de saída e pesquisa como porta de retorno. Você pode optar por usar o primeiro, o último ou qualquer valor que corresponda à condição de pesquisa.

Se você chamar a pesquisa desconectada de uma expressão de filtro ou estratégia de atualização, estará geralmente verificando valores nulos. Nesse caso, a porta de retorno pode ser qualquer coisa. Se você chamar a pesquisa de uma expressão que realiza um cálculo, o valor de retorno deverá ser o valor que você quer incluir no cálculo.

Para continuar o exemplo da estratégia de atualização, defina a porta ID_ITEM como porta de retorno. A expressão de estratégia de atualização verificará valores nulos retornados. Se a condição de pesquisa for verdadeira, o Serviço de Integração retornará ID_ITEM. Se a condição for falsa, o Serviço de Integração retornará NULL.

A figura a seguir mostra uma porta de retorno em uma transformação Pesquisa:



Etapa 4. Chamar a Pesquisa Através de uma Expressão

Forneça valores de entrada para uma transformação Pesquisa não conectada a partir de uma expressão :LKP em outra transformação. Os argumentos são portas de entrada local que correspondem às portas de entrada da transformação Pesquisa usadas na condição de pesquisa. Utilize a sintaxe a seguir para uma expressão :LKP:

```
:LKP.lookup_transformation_name(argument, argument, ...)
```

Para continuar o exemplo acima sobre a loja de varejo, quando você gravar a expressão da estratégia de atualização, a ordem das portas na expressão deverá corresponder à ordem na condição de pesquisa. Nesse caso, a condição ITEM_ID é a primeira condição de pesquisa e, portanto, é o primeiro argumento na expressão da estratégia de atualização.

```
IIF(ISNULL(:LKP.lkpITEMS_DIM(ITEM_ID, PRICE)), DD_UPDATE, DD_REJECT)
```

Utilize as diretrizes a seguir para gravar uma expressão que chama uma transformação Pesquisa não conectada:

- A ordem em que você lista cada argumento deve corresponder à ordem das condições de pesquisa na transformação Pesquisa.
- Os tipos de dados das portas na expressão devem corresponder aos tipos de dados das portas de entrada na transformação Pesquisa. O Designer não validará a expressão se os tipos de dados não corresponderem.
- Se uma porta na condição de pesquisa não for uma porta de pesquisa/saída, o Designer não validará a expressão.
- As portas de argumento na expressão devem estar na mesma ordem que as portas de entrada na condição de pesquisa.
- Se você usar uma sintaxe :LKP incorreta, o Designer marcará o mapeamento como inválido.
- Se você chamar uma transformação Pesquisa conectada em uma expressão :LKP, o Designer marcará o mapeamento como inválido.

Sugestão: Para evitar erros de sintaxe ao inserir expressões, utilize o método de apontar e clicar para selecionar funções e portas.

Resiliência de Deadlock de Banco de Dados

A transformação Pesquisa é resiliente a um deadlock do banco de dados para pesquisas sem cache. Quando ocorre um erro de deadlock do banco de dados, a sessão não falha. O Serviço de Integração tenta reexecutar a última instrução para um período especificado de novas tentativas.

Você pode configurar o número de novas tentativas do deadlock e o intervalo de hibernação de deadlock para um Serviço de Integração. Estes valores também afetam os deadlocks do banco de dados para o escritor relacional. Você pode substituir esses valores no nível da sessão como propriedades personalizadas.

Configure as seguintes Propriedades do Serviço de Integração:

- **NumOfDeadlockRetries.** O número de vezes que o Serviço de Integração PowerCenter tenta executar novamente uma gravação no destino, em um deadlock do banco de dados. O mínimo é 0. O padrão é 10. Se você deseja que a sessão falhe no deadlock, configure NumOfDeadlockRetries como zero.
- **DeadlockSleep.** Número de segundos antes que o Serviço de Integração do PowerCenter tente executar novamente uma gravação no destino em um deadlock do banco de dados.

Se um deadlock ocorrer, o Serviço de Integração tenta executar a instrução. O Serviço de Integração espera por um período de atraso antes de cada nova tentativa. Se todas as tentativas falharem devido ao deadlock, a sessão falha. O Serviço de Integração registra uma mensagem no log da sessão sempre que tentar novamente uma instrução.

Criando uma transformação Pesquisa

Crie uma Transformação Pesquisa reutilizável no Transformation Developer. Crie uma transformação Pesquisa não reutilizável no Mapping Designer.

Para criar uma transformação Pesquisa:

1. Para criar uma transformação Pesquisa reutilizável, abra o Transformation Developer.
Para criar uma transformação Pesquisa não reutilizável, abra um mapeamento no Mapping Designer. Se você estiver criando a transformação Pesquisa de pipeline, arraste uma definição de origem para usar como a origem da pesquisa.
2. Clique em Transformação > Criar. Selecione a transformação Pesquisa.
3. Insira um nome para a transformação. Clique em Criar.
A convenção de nomenclatura das transformações Pesquisa é `LKP_TransformationName`.
4. Escolha se a transformação é ativa ou passiva. Clique em OK. Não é possível alterar essa opção.
5. Na caixa de diálogo Selecionar Tabela de Pesquisa, escolha uma das seguintes opções para importar uma definição de pesquisa:
 - Definição da origem a partir do repositório.
 - Definição do destino no repositório.
 - Qualificador de origem a partir do mapeamento.
 - Importe uma tabela relacional ou arquivo do repositório.

Nota: Você pode adicionar manualmente as portas de pesquisa, em vez de importar uma definição. Você pode escolher quais portas de pesquisa também são portas de saída.

Quando você escolhe a origem da pesquisa, o Designer cria portas na transformação com base nas portas do objeto que você escolher. O Designer configura cada porta como uma porta de pesquisa e uma porta de saída. As portas de pesquisa representam as colunas na origem da pesquisa. A transformação Pesquisa recebe os dados da origem em cada porta de pesquisa, e os transfere para o destino.

6. Se você deseja que a transformação Pesquisa retorne todas as linhas correspondentes, ative Retornar Todas as Linhas em Múltiplas Correspondências. Você não pode alterar essa opção depois de criar a transformação. A Transformação Pesquisa se torna ativa.
7. Clique em OK ou em Ignorar se você deseja adicionar as portas de pesquisa manualmente, em vez de importar uma definição. Você pode escolher quais portas de pesquisa também são portas de saída.
8. Para uma transformação Pesquisa conectada, adicione portas de entrada e saída.
Você pode transferir os dados através da transformação e retornar os dados da tabela de pesquisa para o destino.
9. Para uma transformação Pesquisa desconectada, crie uma porta de retorno para o valor que você deseja retornar da pesquisa.
Você pode retornar uma coluna para a transformação que chamou a pesquisa.
10. Clique na guia Propriedades para configurar as propriedades da Transformação Pesquisa. Configure a execução do cache de pesquisa.
O cache de pesquisa é ativado por padrão para transformações de Pesquisa de pipeline e arquivo simples.
11. Para uma transformação Pesquisa que tenha um cache de pesquisa dinâmica, associe uma porta de entrada, uma porta de saída ou uma ID de sequência a cada porta de pesquisa.

O Serviço de Integração insere ou atualiza as linhas no cache de pesquisa com os dados de cada expressão associada. Se você associar um ID de sequência, o Serviço de Integração gera uma chave primária para linhas inseridas no cache de pesquisa.

12. Adicione a condição de pesquisa na guia Condição.

A condição de pesquisa compara os valores da coluna de origem com os valores na origem da pesquisa. A guia Condição em Porta de Transformação representa os valores da coluna de origem. A Tabela de Pesquisa representa a origem da pesquisa.

TÓPICOS RELACIONADOS:

- [“Criando uma Transformação Pesquisa de Pipeline Reutilizável” na página 322](#)
- [“Criando uma Transformação Pesquisa de Pipeline Não Reutilizável” na página 322](#)

Criando uma Transformação Pesquisa de Pipeline Reutilizável

Crie uma Transformação Pesquisa de pipeline reutilizável no Transformation Developer. Quando você cria a transformação, escolha Origem como o local da tabela de pesquisa. O Transformation Developer exibe uma lista de definições da origem a partir do repositório.

Quando você escolhe um qualificador de origem que representa uma tabela relacional ou definição de origem de arquivo simples, o Designer criará uma transformação Pesquisa relacional ou de arquivo simples. Quando o qualificador de origem representa uma origem de aplicativo, o Designer criará uma transformação Pesquisa de pipeline. Para criar uma transformação Pesquisa de pipeline para uma origem de pesquisa relacional ou de arquivo simples, altere o tipo de origem para Qualificador de Origem depois de criar a transformação. Insira o nome da definição de origem na propriedade Tabela de Pesquisa.

Quando você arrastar uma transformação Pesquisa de pipeline reutilizável em um mapeamento, o Mapping Designer adiciona a definição de origem da propriedade Tabela de Pesquisa ao mapeamento. O Designer adiciona a Transformação Qualificador de Origem.

Para alterar o Nome da Tabela de Pesquisa para outro qualificador de origem no mapeamento, clique no botão Abrir na propriedade Nome da Tabela de Pesquisa. Escolha um Qualificador de Origem na lista.

Criando uma Transformação Pesquisa de Pipeline Não Reutilizável

Crie uma transformação Pesquisa de pipeline não reutilizável no Mapping Designer. Arraste uma definição de origem para um mapeamento. Quando você cria a transformação no Mapping Designer, selecione o Qualificador da origem como o local da tabela de pesquisa. O Mapping Designer exibe uma lista dos qualificadores de origem no mapeamento. Quando você seleciona um qualificador de origem, o Mapping Designer preenche a transformação da Pesquisa com os nomes de porta e os atributos do qualificador a partir da origem que você escolher.

Dicas para Transformações de Pesquisa

[Adicione um índice às colunas usadas em uma condição de pesquisa.](#)

Se você tiver privilégios de modificar o banco de dados que contém a tabela de pesquisa, poderá aumentar o desempenho de pesquisas com e sem cache. Isso é importante para tabelas de pesquisa muito grandes. Como o Serviço de Integração precisa consultar, classificar e comparar valores nessas colunas, o índice precisa incluir todas as colunas usadas em uma condição de pesquisa.

Posicione as condições com um operador de igualdade (=) na frente.

Se você incluir mais de uma condição de pesquisa, coloque-as na seguinte ordem para otimizar o desempenho de pesquisa:

- Igual a (=)
- Menor que (<), maior que (>), menor ou igual a (<=), maior ou igual a (>=)
- Diferente de (!=)

Armazene pequenas tabelas de pesquisa no cache.

Melhore o desempenho da sessão armazenando pequenas tabelas de pesquisa no cache. O resultado do processamento e da consulta da pesquisa é o mesmo, independentemente de você armazenar a tabela de pesquisa no cache ou não.

Una tabelas no banco de dados.

Se a tabela de consulta estiver no mesmo banco de dados que a tabela de origem no mapeamento e o cache não for viável, una as tabelas no banco de dados de origem em vez de usar uma transformação Pesquisa.

Use um cache de pesquisa persistente para pesquisas estáticas.

Se a origem de pesquisa não mudar entre as sessões, configure a transformação de Pesquisa para usar um cache de pesquisa persistente. O Serviço de Integração salvará e reutilizará os arquivos do cache de uma sessão para outra, eliminando o tempo necessário para ler a origem de pesquisa.

Chame transformações Pesquisa desconectadas com o qualificador de referência :LKP.

Ao gravar uma expressão que use o qualificador de referência :LKP, chame apenas transformações Pesquisa desconectadas. Se você tentar chamar uma transformação Pesquisa conectada, o Designer exibirá um erro e marcará o mapeamento como inválido.

Configure uma transformação Pesquisa no pipeline para aumentar o desempenho ao processar origens de pesquisa de arquivos simples e relacionais.

Você pode criar partições para processar origens de pesquisa de arquivos simples e relacionais ao definir a origem da pesquisa como qualificador da origem. Configure uma transformação Pesquisa não reutilizável no pipeline e crie partições no pipeline parcial que processa a origem da pesquisa.

CAPÍTULO 18

Caches de Pesquisa

Este capítulo inclui os seguintes tópicos:

- [Visão geral de Caches de Pesquisa, 324](#)
- [Construindo Caches de Pesquisas Conectadas, 326](#)
- [Usando um Cache de Pesquisa Persistente, 328](#)
- [Trabalhando com pesquisa sem cache ou com cache estático, 330](#)
- [Compartilhando o Cache de Pesquisa, 330](#)
- [Dicas para Caches de Pesquisa, 337](#)

Visão geral de Caches de Pesquisa

Você pode configurar uma transformação de Pesquisa para armazenar a origem da pesquisa em cache e, dessa forma, melhorar o desempenho de pesquisa. Ative o cache de pesquisa quando o arquivo ou a tabela de pesquisa for grande.

O Serviço de Integração cria um cache na memória ao processar a primeira linha de dados em uma transformação de Pesquisa em cache. O Serviço de Integração cria caches à medida que as linhas de origem entram na transformação de Pesquisa. Ele aloca memória para o cache com base na quantidade configurada nas propriedades de transformação ou de sessão. O Serviço de Integração armazena valores de condição no cache de índice e valores de saída no cache de dados. O Serviço de Integração consulta o cache a cada linha que entra na transformação.

Se os dados não se encaixarem no cache de memória, o Serviço de Integração armazena os valores excedentes nos arquivos de cache. O Serviço de Integração também cria arquivos de cache no diretório de cache especificado. Após a conclusão do mapeamento da sessão, o Serviço de Integração libera a memória de cache e exclui os arquivos do cache, a menos que você configure a transformação de Pesquisa para usar um cache persistente.

Se você usar uma pesquisa de arquivo simples ou de pipeline, o Serviço de Integração sempre armazenará a origem da pesquisa em cache. Se você configurar uma pesquisa de arquivo simples na entrada classificada, o Serviço de Integração não poderá armazenar a pesquisa se as colunas de condição não estiverem agrupadas. Se as colunas estiverem agrupadas, mas não classificadas, o Serviço de Integração processa a pesquisa como se você não houvesse configurado a entrada classificada.

Quando você configura um cache de pesquisa, pode definir as seguintes configurações:

Caches sequenciais e simultâneos

Você pode configurar a sessão para construir caches de maneira sequencial ou simultânea. Quando você cria caches sequenciais, o Serviço de Integração cria os caches à medida que as linhas da origem

entram na transformação de Pesquisa. Quando você configura a sessão para criar caches simultâneos, o Serviço de Integração não espera até que a primeira linha entre na transformação de Pesquisa antes de criar os caches. Em vez disso, ele constrói caches múltiplos simultaneamente.

Cache persistente

Você pode salvar os arquivos de cache de pesquisa e reutilizá-los na próxima vez que o Serviço de Integração processar uma transformação de Pesquisa configurada para usar o cache.

Reenviar para cache a partir da origem

Se o cache persistente não estiver sincronizado com a origem da pesquisa, você poderá configurar a transformação de Pesquisa para reconstruir o cache de pesquisa.

É possível configurar a transformação de Pesquisa para reconstruir um cache de pesquisa persistente.

Cache estático

É possível configurar um cache estático para qualquer origem da pesquisa. Por padrão, o Serviço de Integração cria um cache estático. Ele armazena o arquivo de pesquisa ou a tabela em cache e procura valores no cache para cada linha que entrar na transformação. Quando a condição de pesquisa é verdadeira, o Serviço de Integração retorna um valor do cache de pesquisa. O Serviço de Integração não atualiza o cache enquanto processa a transformação de Pesquisa.

Cache dinâmico

Para armazenar uma origem de pesquisa em cache e atualizar o cache, configure uma transformação de Pesquisa com cache dinâmico. O Serviço de Integração insere ou atualiza dinamicamente os dados no cache de pesquisa e transmite os dados para o destino. O cache dinâmico é sincronizado com o destino.

Cache compartilhado

Você pode compartilhar o cache de pesquisa entre várias transformações. Você pode compartilhar um cache não nomeado entre transformações no mesmo mapeamento. Você pode compartilhar um cache nomeado entre transformações no mesmo mapeamento ou em mapeamentos diferentes.

Transformações de Pesquisa podem compartilhar caches estáticos sem nome dentro do mesmo grupo de ordem de carregamento de destino se as regras de compartilhamento de cache corresponderem. Transformações de Pesquisa não podem compartilhar o cache dinâmico dentro do mesmo grupo de ordem de carregamento de destino.

Você pode compartilhar o cache de pesquisa entre várias transformações de Pesquisa no mesmo mapeamento.

Quando você não configura a transformação de Pesquisa para armazenamento em cache, o Serviço de Integração consulta a origem de cada linha de entrada da pesquisa. Não importa se você armazena ou não a origem da pesquisa: o resultado da consulta e do processamento da Pesquisa é o mesmo. No entanto, você pode melhorar o desempenho da pesquisa em uma grande origem de pesquisa se ativar o armazenamento da pesquisa em cache.

Comparação de Cache

O Serviço de Integração funciona de forma diferente com base no tipo de cache de pesquisa que você configura.

A seguinte tabela compara as transformações Pesquisa a uma pesquisa sem cache, um cache estático e um cache dinâmico:

Sem Cache	Cache Estático	Cache Dinâmico
O Serviço de Integração não insere ou atualiza o cache.	O Serviço de Integração não insere ou atualiza o cache.	O Serviço de Integração pode inserir ou atualizar linhas no cache, à medida que ele passa linhas para o destino.
Você pode usar uma pesquisa relacional.	Você não pode usar uma pesquisa relacional, de arquivo simples ou de pipeline. Você pode usar uma pesquisa relacional ou de arquivo simples.	Você pode usar uma pesquisa relacional, de arquivo simples ou de Qualificador de Origem. Você pode usar uma pesquisa relacional ou de arquivo simples.
Quando a condição é verdadeira, o Serviço de Integração retorna um valor a partir da tabela ou cache de pesquisa. Quando a condição não é verdadeira, o Serviço de Integração retorna o valor padrão para as transformações conectadas e NULL para as desconectadas.	Quando a condição é verdadeira, o Serviço de Integração retorna um valor a partir da tabela ou cache de pesquisa. Quando a condição não é verdadeira, o Serviço de Integração retorna o valor padrão para as transformações conectadas e NULL para as desconectadas.	Quando a condição é verdadeira, o Serviço de Integração atualiza as linhas no cache ou deixa o cache inalterado, com base no tipo de linha. Isto indica que a linha está no cache e na tabela de destino. Você pode transferir linhas atualizadas para um destino. Quando a condição não é verdadeira, o Serviço de Integração insere linhas no cache ou deixa o cache inalterado, com base no tipo de linha. Isto indica que a linha não está no cache ou destino. Você pode transferir linhas inseridas para uma tabela de destino.

TÓPICOS RELACIONADOS:

- [“Trabalhando com pesquisa sem cache ou com cache estático” na página 330](#)
- [“Atualizações do Cache de Pesquisa Dinâmica” na página 346](#)

Construindo Caches de Pesquisas Conectadas

O Serviço de Integração pode gerar caches de pesquisa para transformações de pesquisa conectadas das seguintes maneiras:

- **Caches sequenciais.** O Serviço de Integração cria caches de pesquisa sequencialmente. O Serviço de Integração cria um cache na memória ao processar a primeira linha dos dados em uma Transformação Pesquisa em cache.
- **Caches simultâneos.** O Serviço de Integração cria caches de pesquisa simultaneamente. Não é necessário esperar que dados alcancem a Transformação Pesquisa.

Nota: O Serviço de Integração gera caches para transformações de Pesquisa não conectadas de forma sequencial, independente de como você configura a geração de cache. Se você configurar a sessão para que gere caches simultâneos para uma Transformação Pesquisa não conectada, o Serviço de Integração ignora essa configuração e gera caches de transformação de pesquisa não conectadas de forma sequencial.

Caches Sequenciais

Por padrão, o Serviço de Integração compila um cache na memória ao processar a primeira linha de dados em uma transformação Pesquisa com cache. O Serviço de Integração cria cada cache de pesquisa no pipeline de forma sequencial. O Serviço de Integração aguarda alguma transformação upstream ativa para concluir o processamento, antes de processar as linhas na transformação Pesquisa. O Serviço de Integração não compila caches para transformações Pesquisa downstream até que uma transformação Pesquisa upstream termine de compilar um cache.

Por exemplo, o mapeamento a seguir contém uma transformação Agregador não classificada, seguida de duas transformações Pesquisa:

Figura 2. Compilando caches de pesquisa de forma sequencial



O Serviço de Integração processa todas as linhas na transformação Agregador não classificada e, assim que termina, começa a processar a primeira transformação Pesquisa. Ao processar a primeira linha de entrada, o Serviço de Integração começa a compilar o primeiro cache de pesquisa. Após concluir a compilação do primeiro cache de pesquisa, o Serviço de Integração pode começar a processar os dados da pesquisa. O Serviço de Integração começa a compilar o cache de pesquisa seguinte quando a primeira linha de dados alcança a transformação Pesquisa.

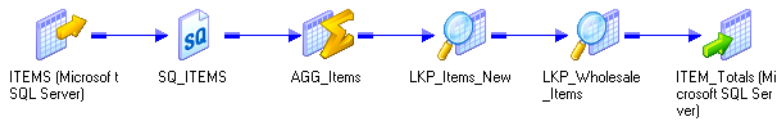
Talvez você queira processar caches de pesquisa de forma sequencial se a transformação Pesquisa não processar dados de linha. A transformação Pesquisa poderá não processar dados de linha, caso a lógica da transformação for configurada para rotear dados para pipelines diferentes com base em uma condição. A configuração do cache sequencial talvez permita evitar a compilação desnecessária de caches de pesquisa. Por exemplo, uma transformação Roteador poderá rotear dados para um pipeline se uma condição for resolvida como verdadeira e para outro pipeline se a condição for resolvida como falsa. Nesse caso, uma transformação Pesquisa pode não receber nenhum dado.

Caches Simultâneos

Você pode configurar o Serviço de Integração para criar caches de pesquisa simultaneamente. Você pode aprimorar o desempenho da sessão usando caches simultâneos. O desempenho pode melhorar especificamente quando o pipeline contém um upstream de transformações ativas na transformação Pesquisa. Você pode desejar configurar a sessão para criar caches simultâneos, se tem a certeza de que precisará criar caches para cada uma das transformações Pesquisa na sessão.

Quando você configura a transformação Pesquisa para criar caches simultâneos, ela não espera o término das transformações upstream antes de criar caches de pesquisa, e não precisa terminar a criação de um cache de pesquisa antes que possa começar a criar outros caches.

A figura a seguir mostra caches de transformação Pesquisa criados simultaneamente:



Ao executar a sessão, o Serviço de Integração cria caches de Pesquisa simultaneamente. Ele não espera o término das transformações upstream e nem que outras transformações Pesquisa terminem a criação do cache.

Nota: Você não pode processar caches para transformações Pesquisa desconectadas simultaneamente.

Para configurar a sessão a fim de criar caches simultâneos, configure um valor para o atributo de configuração da sessão, Pipelines Simultâneos Adicionais para a Criação do Cache de Pesquisa.

Usando um Cache de Pesquisa Persistente

Você pode configurar uma transformação Pesquisa para usar cache persistente ou não persistente. O Serviço de Integração salva ou exclui arquivos de cache de pesquisa após uma sessão bem-sucedida, com base na propriedade cache de pesquisa persistente.

Se a tabela de pesquisa não mudar entre as sessões, você poderá configurar a transformação de Pesquisa para usar um cache de pesquisa persistente. O Serviço de Integração salvará e reutilizará arquivos do cache de uma sessão para outra, eliminando o tempo necessário para ler a tabela de pesquisa.

Usando um Cache Não Persistente

Por padrão, o Serviço de Integração usa cache não persistente quando você ativa o cache em uma transformação Pesquisa. O Serviço de Integração exclui os arquivos de cache ao fim da sessão. Na próxima vez em que você executar a sessão, o Serviço de Integração compila o cache de memória do banco de dados.

Usando um Cache Persistente

Se quiser salvar e reutilizar os arquivos de cache, configure a transformação para usar um cache persistente. Use um cache persistente quando souber que a tabela de pesquisa não se altera entre as sessões.

A primeira vez em que o Serviço de Integração executa uma sessão usando um cache de pesquisa persistente, ele salva os arquivos de cache no disco, em vez de excluí-los. Na vez seguinte em que executar a sessão, o Serviço de Integração compila o cache de memória dos arquivos de cache. Se a tabela de pesquisa for alterada, você pode substituir propriedades da sessão para executar o cache do banco de dados novamente.

Ao usar um cache de pesquisa persistente, você pode especificar um nome para os arquivos de cache. Ao especificar um cache com nome, você pode compartilhar o cache de pesquisa entre sessões.

Recriando o Cache de Pesquisa

Você poderá instruir o Serviço de Integração a recompilar o cache de pesquisa se achar que a origem da pesquisa foi alterada desde a última vez que o Serviço de Integração compilou o cache persistente.

Quando você recompilar o cache, o Serviço de Integração cria novos arquivos de cache, substituindo outros arquivos de cache persistente. O Serviço de Integração grava uma mensagem no log da sessão quando recompila o cache.

Você pode recompilar o cache quando o mapeamento contiver uma transformação Pesquisa ou contiver transformações Pesquisa em vários grupos de ordem de carregamento de destino que compartilhem um cache. Você não precisa recompilar o cache quando uma pesquisa dinâmica compartilha o cache com uma pesquisa estática no mesmo mapeamento.

Se o Serviço de Integração não puder reutilizar o cache, ele efetuará novamente o cache de pesquisa do banco de dados ou encerrará a sessão, dependendo das propriedades da sessão e do mapeamento.

A tabela a seguir resume como o Serviço de Integração lida com cache persistente para caches com e sem nome:

Alterações ao mapeamento ou à sessão entre sessões	Cache nomeado	Cache sem Nome
O Serviço de Integração não pode localizar arquivos de cache. Por exemplo, o arquivo não existe mais ou o Serviço de Integração é executado em uma grade e o arquivo de cache não está disponível para todos os nós.	Recompila o cache.	Recompila o cache.
Ative ou desative a opção Habilitar Alta Precisão nas propriedades da seção.	Encerra a sessão.	Recompila o cache.
Edite a transformação no Mapping Designer, Mapplet Designer ou Transformation Developer Reutilizáveis. ¹	Encerra a sessão.	Recompila o cache.
Edite o mapeamento (excluindo a transformação Pesquisa).	Reutiliza cache.	Recompila o cache.
Altera o número de partições no pipeline que contém a transformação Pesquisa.	Encerra a sessão.	Recompila o cache.
Altera a conexão de banco de dados ou a localização do arquivo para acessar a tabela de pesquisa.	Encerra a sessão.	Recompila o cache.
Altera o modo de movimento dos dados do Serviço de Integração.	Encerra a sessão.	Recompila o cache.
Alterar a ordem de classificação no modo Unicode.	Encerra a sessão.	Recompila o cache.
Altera a página de código do Serviço de Integração para uma página de código compatível.	Reutiliza cache.	Reutiliza cache.
Alterar a página de código do Serviço de Integração para uma página de código incompatível.	Encerra a sessão.	Recompila o cache.

¹ A edição de propriedades, como descrições de transformações ou portas, não afeta a manipulação do cache persistente.

Trabalhando com pesquisa sem cache ou com cache estático

Por padrão, o Serviço de Integração cria um cache de pesquisa estática quando você configura uma transformação Pesquisa para cache. O Serviço de Integração cria o cache quando processa a primeira solicitação de pesquisa. Ele consulta o cache com base na condição de pesquisa para cada linha que passa pela transformação. O Serviço de Integração não atualiza o cache enquanto processa a transformação. O Serviço de Integração processa uma pesquisa sem cache do mesmo modo que processa outra com cache, exceto por que consulta a origem de pesquisa em vez de compilar e consultar o cache.

Se a condição de pesquisa for verdadeira, o Serviço de Integração retorna os valores do cache ou da origem de pesquisa. Para transformações Pesquisa conectadas, o Serviço de Integração retorna os valores representados pelas portas de pesquisa e saída. Para transformações Pesquisa desconectadas, o Serviço de Integração retorna o valor representado pela porta de retorno.

Se a condição não for verdadeira, o Serviço de Integração retorna valores NULL ou padrão. Para transformações Pesquisa conectadas, o Serviço de Integração retorna o valor padrão da porta de saída quando a condição não for atendida. Para transformações Pesquisa desconectadas, o Serviço de Integração retorna NULL quando a condição não for atendida.

Quando você cria diversas partições em um pipeline que usa cache estático, o Serviço de Integração cria um cache de memória para cada partição e um cache de disco para cada transformação.

Compartilhando o Cache de Pesquisa

Você pode configurar várias transformações Pesquisa em um mapeamento para compartilhar um único cache de pesquisa. O Serviço de Integração constrói o cache quando processa a primeira transformação Pesquisa. Ele usa o mesmo cache para executar pesquisas para transformações Pesquisa subsequentes que compartilham o cache.

Você pode compartilhar caches com e sem nome:

- **Cache sem nome.** Quando transformações Pesquisa em um mapeamento têm estruturas de cache compatíveis, por padrão o Serviço de Integração compartilha o cache. Você só pode compartilhar caches estáticos sem nome.
- **Cache com nome.** Utilize um cache persistente com nome quando quiser compartilhar um arquivo de cache em mapeamentos ou compartilhar um cache dinâmico e um estático. As estruturas de cache devem corresponder ou ser compatíveis com um cache com nome. Você pode compartilhar caches estáticos e dinâmicos com nome.

Quando o Serviço de Integração compartilha um cache de pesquisa, ele grava uma mensagem no log da sessão.

Compartilhando um Cache de Pesquisa Sem Nome

Por padrão, o Serviço de Integração compartilha o cache para transformações Pesquisa em um mapeamento que tem estruturas de memória cache compatíveis. Por exemplo, se você tiver duas instâncias da mesma transformação Pesquisa reutilizável em um mapeamento e utilizar as mesmas portas de saída para as duas, por padrão as transformações Pesquisa compartilharão o cache de pesquisa.

Quando duas transformações Pesquisa compartilham um cache sem nome, o Serviço de Integração salva o cache para uma transformação Pesquisa e utiliza-a para transformações Pesquisa subsequentes que têm a mesma estrutura de cache de pesquisa.

Se as propriedades da transformação ou a estrutura do cache não permitir compartilhamento, o Serviço de Integração criará um novo cache.

Diretrizes para Compartilhar um Cache de Pesquisa Sem Nome

Use as orientações a seguir ao configurar transformações de Pesquisa para que compartilhem um cache não nomeado:

- Você pode compartilhar caches estáticos sem nome.
- As transformações compartilhadas devem usar as mesmas portas na condição de pesquisa. As condições podem usar diferentes operadores, mas as portas precisam ser as mesmas.
- É necessário configurar algumas das propriedades de transformação para ativar o compartilhamento de cache não nomeado.
- A estrutura do cache da transformação compartilhada deve ser compatível.
 - Caso você utilize partição de chaves automáticas de hash, as portas de pesquisa/saída para cada transformação devem corresponder.
 - Se você não utilizar partição de chaves automáticas de hash, as portas de pesquisa/saída da primeira transformação compartilhada devem corresponder ou ser um superconjunto das portas de pesquisa/saída para as transformações subsequentes.
- Se as transformações de Pesquisa com partição de chave automática de hash estiverem em diferentes grupos de ordem de carregamento de destino, é necessário configurar o mesmo número de partições para cada grupo. Se você não usar partição de chave automática de hash, é possível configurar um número diferente de partições para cada grupo de ordem para carregamento de destino.

A tabela a seguir mostra quando se pode compartilhar um cache estático não nomeado e dinâmico:

Cache Compartilhado	Localização das Transformações
Estático com Estático	Em qualquer lugar do mapeamento.
Dinâmico com Dinâmico	Não é possível compartilhar.
Dinâmico com Estático	Não é possível compartilhar.

A tabela a seguir descreve as orientações a serem seguidas ao configurar transformações de Pesquisa para compartilhar um cache não nomeado:

Propriedades	Configuração do Cache Compartilhado Não Nomeado
Substituição SQL de Pesquisa	Se você usar a propriedade Substituição SQL de Pesquisa, precisará usar a mesma substituição em todas as transformações compartilhadas.
Nome da Tabela de Pesquisa	Deve corresponder.
Armazenamento em Cache de Pesquisa Ativado	Deve ser ativado.
Diretiva de Pesquisa em Várias Correspondências	-

Propriedades	Configuração do Cache Compartilhado Não Nomeado
Condição de Pesquisa	As transformações compartilhadas devem usar as mesmas portas na condição de pesquisa. As condições podem usar diferentes operadores, mas as portas precisam ser as mesmas.
Informações de Conexão	A conexão deve ser a mesma. Quando você configura as sessões, a conexão de banco de dados deve ser correspondente.
Tipo de Origem	Deve corresponder.
Nível de Rastreamento	-
Nome do Diretório de Cache de Pesquisa	Não precisa corresponder.
Cache de Pesquisa Persistente	Opcional. Você pode compartilhar os persistentes e os não persistentes.
Tamanho do Cache de Dados de Pesquisa	O Serviço de Integração aloca memória para a primeira transformação compartilhada em cada fase de pipeline. Ele não aloca mais memória para transformações compartilhadas subsequentes na mesma fase de pipeline.
Tamanho do Cache de Índice de Pesquisa	O Serviço de Integração aloca memória para a primeira transformação compartilhada em cada fase de pipeline. Ele não aloca mais memória para transformações compartilhadas subsequentes na mesma fase de pipeline.
Cache de Pesquisa Dinâmica	Não é possível compartilhar um cache dinâmico não nomeado.
Valor Antigo de Saída na Atualização	Não precisa corresponder.
Prefixo do Nome do Arquivo de Cache	Não usar. Não é possível compartilhar um cache nomeado com um cache não nomeado.
Reenviar para Cache da Origem de Pesquisa	<p>Se você configurar uma transformação de Pesquisa para que armazene novamente em cache da origem, as transformações de Pesquisa subsequentes no grupo de ordem para carregamento de destino poderão compartilhar o cache atual independentemente de você as configurar para armazenar novamente em cache da origem. Se você configurar transformações de Pesquisa subsequentes para que armazenem novamente em cache da origem, o Serviço de Integração compartilhará o cache em vez de recriá-lo ao processar a transformação de Pesquisa subsequente.</p> <p>Se você não configurar a primeira transformação de Pesquisa em um grupo de ordem para carregamento de destino para que armazene novamente em cache do destino e configurar a transformação de Pesquisa subsequente para armazenar novamente em cache da origem, a transformação não poderá compartilhar o cache. O Serviço de Integração cria o cache ao processar cada transformação de Pesquisa.</p>
Portas de Pesquisa/Saída	As portas de pesquisa/saída da segunda transformação de Pesquisa devem corresponder ou ser um subconjunto das portas da transformação que o Serviço de Integração utiliza para criar o cache. A ordem das portas não precisa corresponder.
Insert Else Update	-
Atualizar e Inserir	-

Propriedades	Configuração do Cache Compartilhado Não Nomeado
Formato Data e Hora	-
Separador de Milhares	-
Separador Decimal	-
Comparação de String com Distinção Entre Maiúsculas e Minúsculas	Deve corresponder.
Ordenamento Nulo	Deve corresponder.
Entrada Classificada	-

Compartilhando um Cache de Pesquisa Nomeado

Você também pode compartilhar o cache entre várias transformações Pesquisa usando um cache de pesquisa persistente e nomeando os arquivos do cache. Você pode compartilhar um cache entre transformações Pesquisa no mesmo mapeamento ou vários mapeamentos.

O Serviço de Integração usa o processo a seguir para compartilhar um cache de pesquisa nomeado:

1. Quando o Serviço de Integração processa a primeira transformação Pesquisa, ele procura arquivos de cache com o mesmo prefixo do nome de arquivo no diretório de cache.
2. Se o Serviço de Integração encontrar os arquivos de cache e não especificar para efetuar novamente o cache da origem, o Serviço de Integração usa arquivos de cache salvos.
3. Se o Serviço de Integração não encontrar os arquivos de cache ou você não especificar para efetuar novamente o cache da origem, o Serviço de Integração criará o cache de pesquisa usando a tabela do banco de dados.
4. O Serviço de Integração salva os arquivos de cache no disco depois de processar cada grupo de classificação de carga da origem.
5. O Serviço de Integração usa as seguintes regras para processar a segunda transformação Pesquisa com o mesmo prefixo do nome de arquivo de cache.
 - O Serviço de Integração usará o cache da memória se as transformações estiverem no mesmo grupo de classificação de carga da origem.
 - O Serviço de Integração recriará o cache da memória de arquivos persistentes se as transformações estiverem em grupos de ordem de carregamento de destino diferentes.
 - O Serviço de Integração recriará o cache do banco de dados se você configurar a transformação para efetuar novamente o cache da origem e a primeira transformação estiver em outro grupo de classificação de carga da origem.
 - Se você não configurar a primeira transformação de Pesquisa em um grupo de ordem para carregamento de destino para que armazene novamente em cache a partir da origem e configurar a transformação de Pesquisa subsequente para armazenar novamente em cache a partir da origem, o Serviço de Integração não reconstruirá o cache.
 - Se as estruturas de cache não coincidirem, o Serviço de Integração fechará a sessão.

Se você executar simultaneamente duas sessões que compartilhem um cache de pesquisa, o Serviço de Integração usa estas regras para compartilhar os arquivos de cache:

- O Serviço de Integração processa diversas sessões simultâneas quando as transformações Pesquisa só precisarem ler os arquivos de cache.
- O Serviço de Integração fechará a sessão se uma sessão atualizar um arquivo de cache enquanto outra sessão tentar ler ou atualizar o arquivo de cache. Por exemplo, transformações Pesquisa atualizam o arquivo cache se estiverem configuradas para usar cache dinâmico ou executar novamente o cache da origem.

Diretrizes para compartilhar um cache de pesquisa nomeado

Use as orientações a seguir ao configurar transformações de Pesquisa para que compartilhem um cache nomeado:

- Não compartilhe um cache de pesquisa entre sessões se ele for um cache de pesquisa dinâmica ou se a transformação estiver configurada para refazer o cache da origem.
- Você pode compartilhar um cache entre transformações Pesquisa dinâmicas e estáticas, mas é necessário seguir as orientações para o local do cache.
- É necessário configurar algumas das propriedades de transformação para ativar o compartilhamento de cache nomeado.
- Uma pesquisa dinâmica não pode compartilhar o cache se ele for nomeado e tiver linhas duplicadas.
- Um cache nomeado criado por uma transformação Pesquisa dinâmica, com uma política de pesquisa de erro em múltiplas correspondências, pode ser compartilhado por uma transformação Pesquisa estática ou dinâmica com qualquer política de pesquisa.
- Um cache nomeado criado por uma transformação Pesquisa dinâmica, com uma política de pesquisa para usar o primeiro, o último ou todos os valores, pode ser compartilhado por uma transformação Pesquisa com a mesma política de pesquisa.
- As transformações compartilhadas devem usar as mesmas portas de saída no mapeamento. Os critérios e as colunas de resultado para o cache devem corresponder aos arquivos de cache.
- Uma transformação de pesquisa dinâmica não pode compartilhar o cache com outra transformação de pesquisa dentro do mesmo grupo de ordem de carregamento de destino. Em um grupo de ordem de carregamento de destino, o processamento de caches ocorre em paralelo e conclui após o carregamento do destino. O Serviço de Integração do PowerCenter não permite compartilhamento de cache, pois o cache completo não está disponível para transformações de pesquisa subsequentes.

O Serviço de Integração pode usar o cache de memória ou criá-lo a partir do arquivo, dependendo do tipo e local das transformações Pesquisa.

A tabela a seguir mostra quando você pode compartilhar um cache nomeado entre uma transformação Pesquisa estática e outra dinâmica:

Tabela 3. Local para compartilhamento do cache nomeado

Cache Compartilhado	Local da Transformações	Cache compartilhado
Estático com Estático	<ul style="list-style-type: none"> - Mesmo grupo de ordem de carregamento no destino. - Separar grupos de ordem de carregamento de destino. - Mapeamentos separados. 	<ul style="list-style-type: none"> - O Serviço de Integração usa o cache da memória. - O Serviço de Integração usa o cache da memória. - O Serviço de Integração cria o cache da memória a partir do arquivo.
Dinâmico com Dinâmico	<ul style="list-style-type: none"> - Separar grupos de ordem de carregamento de destino. - Mapeamentos separados. 	<ul style="list-style-type: none"> - O Serviço de Integração usa o cache da memória. - O Serviço de Integração cria o cache da memória a partir do arquivo.
Dinâmico com Estático	<ul style="list-style-type: none"> - Separar grupos de ordem de carregamento de destino. - Mapeamentos separados. 	<ul style="list-style-type: none"> - O Serviço de Integração cria o cache da memória a partir do arquivo. - O Serviço de Integração cria o cache da memória a partir do arquivo.

A tabela a seguir descreve as orientações a serem seguidas ao configurar transformações de Pesquisa para compartilhar um cache nomeado:

Tabela 4. Propriedades para compartilhamento do cache nomeado

Propriedades	Configuração do Cache Compartilhado Nomeado
Substituição SQL de Pesquisa	Se você usar a propriedade Substituição SQL de Pesquisa, é necessário usar a mesma substituição em todas as transformações compartilhadas.
Nome da Tabela de Pesquisa	Deve corresponder.
Armazenamento em Cache Ativado	Deve ser ativado.
Diretiva de Pesquisa em Várias Correspondências	<ul style="list-style-type: none"> - Um cache nomeado criado por uma transformação Pesquisa dinâmica, com uma política de pesquisa de erro em múltiplas correspondências, pode ser compartilhado por uma transformação Pesquisa estática ou dinâmica com qualquer política de pesquisa. - Um cache nomeado criado por uma transformação Pesquisa dinâmica com uma política de pesquisa de primeiro ou último uso pode ser compartilhado por uma transformação de Pesquisa com a mesma política de pesquisa. - Um cache nomeado pode ser compartilhado por uma transformação Pesquisa dinâmica, com uma política de pesquisa de uso de todos os valores, se ela compartilhar o cache com outra transformação Pesquisa ativa com a mesma política de pesquisa.
Condição de pesquisa	As transformações compartilhadas devem usar as mesmas portas na condição de pesquisa. As condições podem usar diferentes operadores, mas as portas precisam ser as mesmas.
Informações de Conexão	A conexão deve ser a mesma. Ao configurar as sessões, a conexão de bancos de dados deve corresponder.

Propriedades	Configuração do Cache Compartilhado Nomeado
Tipo de Origem	Deve corresponder.
Nível de rastreamento	-
Nome do diretório do cache de pesquisa	Deve corresponder.
Cache de pesquisa persistente	Deve ser ativado.
Tamanho do Cache de Dados de Pesquisa	Quando transformações dentro do mesmo mapeamento compartilham um cache, o Serviço de Integração aloca a memória à primeira transformação compartilhada em cada fase de pipeline. Ele não aloca mais memória para transformações compartilhadas subsequentes na mesma fase de pipeline.
Tamanho do Cache do Índice de Pesquisa	Quando transformações dentro do mesmo mapeamento compartilham um cache, o Serviço de Integração aloca a memória à primeira transformação compartilhada em cada fase de pipeline. Ele não aloca mais memória para transformações compartilhadas subsequentes na mesma fase de pipeline.
Cache de Pesquisa Dinâmica	Para obter mais informações sobre o compartilhamento de caches estáticos e dinâmicos, consulte "Diretrizes para compartilhar um cache de pesquisa nomeado" na página 334 .
Valor Antigo de Saída na Atualização	Não precisa corresponder.
Atualizar Cache Dinâmico	Não precisa corresponder.
Prefixo do Nome do Arquivo de Cache	Deve corresponder. Insira somente o prefixo. Não insira .idx ou .dat. Não é possível compartilhar um cache nomeado com um cache não nomeado.
Reenviar para cache a partir de origem	<p>Se você configurar uma transformação de Pesquisa para que armazene novamente em cache a partir da origem, as transformações de Pesquisa subsequentes no grupo de ordem para carregamento de destino podem compartilhar o cache atual independente de você as configurar para armazenar novamente em cache a partir da origem. Se você configurar transformações de Pesquisa subsequentes para que armazenem novamente em cache a partir da origem, o Serviço de Integração compartilha o cache ao invés de recriá-lo quando processa a transformação de Pesquisa subsequente.</p> <p>Se você não configurar a primeira transformação de Pesquisa em um grupo de ordem para carregamento de destino para que armazene novamente em cache a partir da origem e configurar a transformação de Pesquisa subsequente para armazenar novamente em cache a partir da origem, o Serviço de Integração não reconstruirá o cache.</p>
Portas de Pesquisa/Saída	As portas de pesquisa/saída devem ser idênticas, mas não precisam estar na mesma ordem.
Insert Else Update	-
Update Else Insert	-
Separador de Milhar	-
Separador Decimal	-

Propriedades	Configuração do Cache Compartilhado Nomeado
Comparação de String com Diferenciação de Maiúscula/Minúscula	-
Ordenamento Nulo	-
Entrada Classificada	Deve corresponder.

Nota: Você não pode compartilhar um cache de pesquisa criado em um sistema operacional diferente. Por exemplo, somente um Serviço de Integração no UNIX pode ler um cache de pesquisa criado em um Serviço de Integração no UNIX, e o Serviço de Integração no Windows só lê os criados no Windows.

Dicas para Caches de Pesquisa

Armazene pequenas tabelas de pesquisa no cache.

Aumente o desempenho armazenando em cache pequenas tabelas de pesquisa.

Utilize um cache de pesquisa persistente para tabelas de pesquisa estáticas.

Se a tabela de pesquisa não mudar entre as sessões, configure a transformação de Pesquisa para usar um cache de pesquisa persistente. O Serviço de Integração salva e reutiliza arquivos de cache, eliminando o tempo necessário para ler a tabela de pesquisa.

Se a tabela de pesquisa não mudar entre as sessões de mapeamento, configure a transformação de Pesquisa para usar um cache de pesquisa persistente. O Serviço de Integração salva e reutiliza arquivos de cache, eliminando o tempo necessário para ler a tabela de pesquisa.

CAPÍTULO 19

Cache de Pesquisa Dinâmica

Este capítulo inclui os seguintes tópicos:

- [Visão geral do Cache de Pesquisa Dinâmica, 338](#)
- [Propriedades da Pesquisa Dinâmica, 339](#)
- [Valores de Transformação Pesquisa, 343](#)
- [Atualizações do Cache de Pesquisa Dinâmica, 346](#)
- [Mapeamentos com pesquisas dinâmicas, 348](#)
- [Atualizações de Cache Dinâmico Condicional, 351](#)
- [Atualização de cache dinâmico com resultados de expressões, 352](#)
- [Sincronizando o Cache com a Origem da Pesquisa, 353](#)
- [Exemplo de Cache de Pesquisa Dinâmica, 354](#)
- [Regras e Diretrizes para Cache de Pesquisa Dinâmica, 355](#)

Visão geral do Cache de Pesquisa Dinâmica

Use um cache de pesquisa dinâmica para manter o cache sincronizado com o destino. Você pode usar um cache dinâmico com uma pesquisa relacional, de arquivo simples ou de pipeline. Você pode usar um cache dinâmico com uma pesquisa relacional ou de arquivo simples.

O Serviço de Integração cria o cache de pesquisa dinâmica quando processa a primeira solicitação de pesquisa. Ele consulta o cache com base na condição de pesquisa para cada linha que passa pela transformação. O Serviço de Integração atualiza o cache de pesquisa quando processa cada linha.

Com base nos resultados da consulta de pesquisa, o tipo de linha e as propriedades da transformação de Pesquisa, o Serviço de Integração realiza uma das ações a seguir no cache de pesquisa dinâmica ao ler uma linha da origem:

Insere a linha no cache

O Serviço de Integração insere a linha quando ela não está no cache e você configurou a transformação de Pesquisa para inserir linhas no cache. Você pode configurar a transformação para inserir linhas no cache com base nas portas de entrada ou nas IDs de sequência geradas. O Serviço de Integração sinaliza a linha como inserção.

Atualiza a linha no cache

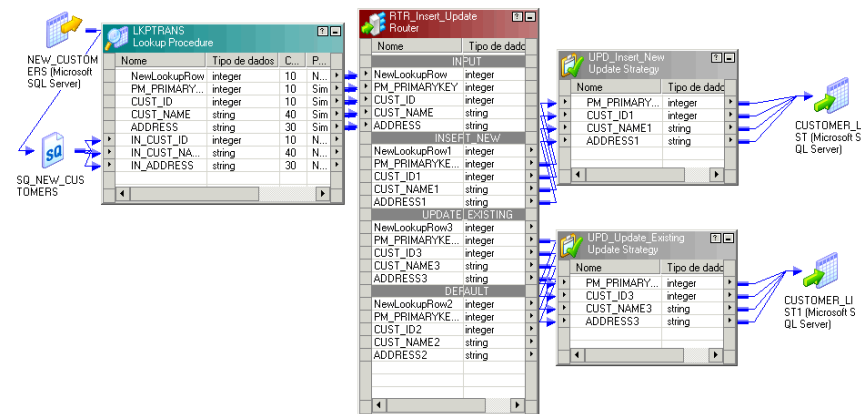
O Serviço de Integração atualiza a linha quando ela existe no cache e você configurou a transformação de Pesquisa para atualizar linhas no cache. O Serviço de Integração atualiza a linha no cache com base nas portas de entrada. O Serviço de Integração sinaliza a linha como uma linha de atualização.

Não faz alterações no cache

O Serviço de Integração não faz alterações quando a linha existe no cache e você configurou a transformação de Pesquisa para inserir somente linhas novas. Ou, a linha não está no cache e você especificou para somente atualizar as linhas existentes. Ou, a linha está no cache, mas com base na condição da pesquisa, nada é alterado. O Serviço de Integração sinaliza a linha como inalterada.

Com base no valor de NewLookupRow, você também pode configurar uma transformação de Roteador ou de Filtro com a transformação de Pesquisa dinâmica para encaminhar as linhas de inserção ou atualização à tabela de destino. Você pode encaminhar linhas inalteradas para outra tabela de destino ou arquivo simples, ou pode descartá-las.

A figura a seguir mostra um mapeamento com uma transformação de Pesquisa que usa um cache de pesquisa dinâmica:



Propriedades da Pesquisa Dinâmica

Uma transformação Pesquisa com um cache dinâmico possui as propriedades a seguir:

- **NewLookupRow.** O Designer adiciona esta porta a uma transformação Pesquisa configurada para usar um cache dinâmico. Ele indica, com um valor numérico, se o Serviço de Integração insere ou atualiza a linha no cache ou se não efetua alterações no cache. Para manter o cache de pesquisa e a tabela de destino sincronizados, transfira as linhas para o destino quando o valor NewLookupRow for igual a 1 ou 2.
- **Expressão Associada.** Associe as portas de pesquisa ou as portas associadas a uma expressão, uma porta de entrada/saída ou um ID de sequência. O Serviço de Integração usa os dados na expressão associada para inserir ou atualizar as linhas no cache de pesquisa. Se você associar um ID de sequência, o Serviço de Integração gera uma chave primária para linhas inseridas no cache de pesquisa.
- **Ignorar Entradas Nulas para as Atualizações.** O Designer ativa esta propriedade para as portas de pesquisa/saída quando você configura a transformação Pesquisa para usar um cache dinâmico. Selecione esta propriedade quando você não deseja que o Serviço de Integração atualize a coluna no cache, quando os dados desta coluna contiverem um valor nulo.
- **Ignorar na Comparação.** O Designer ativa esta propriedade para as portas de pesquisa/saída que não são usadas na condição de pesquisa quando você configura a transformação Pesquisa para usar um cache dinâmico. O Serviço de Integração compara os valores em todas as portas de pesquisa com os valores nas portas de entrada associadas, por padrão. Selecione esta propriedade se desejar que o Serviço de Integração ignore a porta quando for comparar os valores antes de atualizar uma linha.

- **Atualizar Condição de Cache Dinâmico.** Permita que o Serviço de Integração atualize o cache dinâmico condicionalmente. Você pode criar uma expressão booleana que determina se irá ou não atualizar o cache para uma linha de entrada. Ou então, pode permitir que o Serviço de Integração atualize o cache com um resultado de expressão para uma linha de entrada. A expressão pode conter valores a partir da linha de entrada ou do cache de pesquisa.

NewLookupRows

Quando você configura uma transformação Pesquisa para usar um cache dinâmico, o Designer adiciona a porta NewLookupRow à transformação. O Serviço de Integração atribui um valor à porta, dependendo da ação que executa no cache de pesquisa.

A tabela a seguir lista os valores NewLookupRow possíveis:

Valor de NewLookupRow	Descrição
0	O Serviço de Integração não atualiza nem insere a linha no cache.
1	O Serviço de Integração insere a linha no cache.
2	O Serviço de Integração atualiza a linha no cache.

Quando o Serviço de Integração lê uma linha, ele altera o cache de pesquisa, dependendo dos resultados da consulta da pesquisa e das propriedades da transformação Pesquisa que você definiu. Ele atribui o valor 0, 1 ou 2 à porta NewLookupRow para indicar se irá inserir ou atualizar a linha no cache, ou não faz nenhuma alteração.

O valor NewLookupRow indica como o Serviço de Integração altera o cache de pesquisa. Ele não altera o tipo de linha. Portanto, use uma transformação Filtro ou Roteador e uma transformação Estratégia de Atualização para manter a tabela de destino e o cache de pesquisa sincronizados.

Configure a transformação Filtro para transferir linhas novas e atualizadas para a transformação Estratégia de Atualização, antes de transferi-las para o destino com cache. Use a transformação Estratégia de Atualização para alterar o tipo de cada linha para inserir ou atualizar, dependendo do valor NewLookupRow.

Você pode descartar as linhas que não alteram o cache, ou transferi-las para outro destino.

Defina a condição de filtro na transformação Filtro, com base no valor NewLookupRow. Por exemplo, use a seguinte condição para transferir linhas inseridas e atualizadas para o destino com cache:

```
NewLookupRow != 0
```

TÓPICOS RELACIONADOS:

- [“Configurando a Transformação de Estratégia de Atualização Upstream” na página 348](#)

Expressão associada

Quando você configura um cache de pesquisa dinâmica, deve associar cada porta de pesquisa a uma porta de entrada, porta de entrada/saída, uma ID de sequência ou uma expressão. Quando você escolhe uma porta de entrada/saída, o Designer a associa à porta de pesquisa/saída usada na condição de pesquisa. Quando você configura uma expressão, o Serviço de Integração atualiza o cache com o resultado da expressão associada. A expressão pode conter valores de entrada ou valores do cache de pesquisa.

Para associar uma porta de pesquisa a uma porta de entrada/saída, clique na coluna Expressão associada para uma porta de pesquisa. Selecione uma porta na lista.

Para criar uma expressão, clique na coluna Associado referente à porta de pesquisa. Escolha a Expressão associada na lista. O Editor de Expressão será exibido.

Para criar uma chave gerada para uma coluna na tabela de destino, selecione *ID-Sequência* na coluna Expressão associada. Você pode associar uma chave gerada em vez de uma porta de entrada, para as portas de pesquisa com o tipo de dados, Bigint, Inteiro ou Inteiro curto. Para as portas de pesquisa Bigint, o valor máximo da chave gerada é 9.223.372.036.854.775.807. Para as portas de pesquisa Inteiro ou Inteiro curto, o valor máximo da chave gerada é 2.147.483.647.

Quando você seleciona ID-Sequência na coluna Expressão associada, o Serviço de Integração gera uma chave quando ele insere uma linha no cache de pesquisa.

O Serviço de Integração usa o seguinte processo para gerar IDs de sequência:

1. Quando o Serviço de Integração cria o cache de pesquisa dinâmica, rastreia o intervalo de valores para cada porta que tem uma ID de sequência no cache de pesquisa dinâmica.
2. Quando o Serviço de Integração insere uma linha de dados no cache, gera uma chave para uma porta, incrementando o valor da ID da maior sequência em um.
3. Quando o Serviço de Integração atinge o número máximo para uma ID de sequência gerada, recomeça em um. O Serviço de Integração incrementa cada ID da sequência em um, até atingir o menor valor existente menos um. Se o Serviço de Integração esgotar os números de ID de sequência, a sessão falha.

O Serviço de Integração gera uma ID de sequência para cada linha que inserir no cache.

TÓPICOS RELACIONADOS:

- [“Atualização de cache dinâmico com resultados de expressões” na página 352](#)

Valores nulos

Quando você atualiza um cache de pesquisa dinâmica e uma tabela de destino, os dados de origem podem conter alguns valores nulos. O Serviço de Integração pode manipular os valores nulos das seguintes maneiras:

- **Inserir valores nulos.** O Serviço de Integração usa valores nulos a partir da origem e atualiza o cache de pesquisa e a tabela de destino usando todos os valores da origem.
- **Ignorar valores nulos.** O Serviço de Integração ignora os valores nulos na origem e atualiza o cache de pesquisa e a tabela de destino usando apenas os valores da origem que não forem nulos.

Se você souber que os dados da origem contêm valores nulos, e não deseja que o Serviço de Integração atualize o cache de pesquisa ou o destino com valores nulos, selecione a propriedade Ignorar nulo para a porta de pesquisa/saída correspondente.

Por exemplo, você deseja atualizar a tabela mestre de clientes. A origem contém os clientes novos e atuais, cujos sobrenomes foram alterados. A origem contém as IDs e nomes dos clientes cujos nomes mudaram, mas contém valores nulos para as colunas de endereço. Você deseja inserir os novos clientes e atualizar os nomes dos atuais, enquanto mantém as informações atuais de endereço em uma tabela mestre de clientes.

Por exemplo, a tabela de cliente mestre contém os seguintes dados:

Primary Key	CUST_ID	CUST_NAME	ADDRESS	CITY	STATE	ZIP
100001	80001	Marion James	100 Main St.	Mt. View	CA	94040
100002	80002	Laura Jones	510 Broadway Ave.	Raleigh	NC	27601
100003	80003	Shelley Lau	220 Burnside Ave.	Portland	OR	97210

A origem contém os seguintes dados:

CUST_ID	CUST_NAME	ADDRESS	CITY	STATE	ZIP
80001	Marion Atkins	NULL	NULL	NULL	NULL
80002	Laura Gomez	NULL	NULL	NULL	NULL
99001	Jon Freeman	555 6th Ave.	San Jose	CA	95051

Selecione Insert Else Update na transformação de Pesquisa no mapeamento. Selecione a opção Ignorar nulo para todas as portas de pesquisa/saída na transformação de Pesquisa. Quando você executa uma sessão, o Serviço de Integração ignora os valores nulos nos dados da origem e atualiza o cache de pesquisa e a tabela de destino com valores que não sejam nulos:

PRIMARYKEY	CUST_ID	CUST_NAME	ADDRESS	CITY	STATE	ZIP
100001	80001	Marion Atkins	100 Main St.	Mt. View	CA	94040
100002	80002	Laura Gomez	510 Broadway Ave.	Raleigh	NC	27601
100003	80003	Shelley Lau	220 Burnside Ave.	Portland	OR	97210
100004	99001	Jon Freeman	555 6th Ave.	San Jose	CA	95051

Nota: Quando você opta por ignorar NULLs, deve verificar se produz a saída dos mesmos valores para o destino que o Serviço de Integração escreve no cache de pesquisa. Quando você opta por ignorar NULLs, o cache de pesquisa e a tabela de destino podem perder a sincronia se você transferir os valores nulos de entrada para o destino. Configure o mapeamento com base no valor que você deseja que o Serviço de Integração produza a saída a partir das portas de pesquisa/saída, quando ele atualiza uma linha no cache:

- **Valores novos.** Conecte somente as portas de pesquisa/saída a partir da transformação de Pesquisa para o destino.
- **Valores antigos.** Adicione uma transformação de Expressão depois da transformação de Pesquisa e antes da transformação de Filtro ou Roteador. Adicione portas de saída à transformação Expressão para cada porta na tabela de destino e crie expressões para garantir que valores nulos não sejam transferidos ao destino.

Ignorar Portas em Comparação

Ao executar uma sessão que utilize cache de pesquisa dinâmica, o Serviço de Integração compara, por padrão, os valores de todas as portas de pesquisa com os valores das portas de entrada associadas. Ele compara os valores para determinar se vai ou não atualizar a linha no cache de pesquisa. Quando um valor de uma porta de entrada é diferente do valor da porta de pesquisa, o Serviço de Integração atualiza a linha no cache.

Se não quiser comparar todas as portas, é possível escolher as portas que você quer que o Serviço de Integração ignore ao comparar portas. O Designer somente ativa esta propriedade para portas de pesquisa/saída quando a porta não é utilizada na condição de pesquisa. É possível aperfeiçoar o desempenho, ignorando algumas portas durante a comparação.

Talvez você queira fazer isso quando os dados de origem incluírem uma coluna que indique se a linha contém ou não dados que precisem de atualização. Selecione a propriedade Ignorar em Comparação para todas as portas de pesquisa, exceto a porta que indique se a linha do cache e a tabela de destino devem ser atualizadas.

É necessário configurar a transformação de Pesquisa para comparar pelo menos uma porta. O Serviço de Integração não obtém êxito na sessão quando você ignora todas as portas.

Para desmarcar a propriedade Ignorar em Comparação, associe uma porta com a porta de pesquisa. Quando você desmarca a propriedade Ignorar em Comparação, o Serviço de Integração do PowerCenter atualiza a linha no cache.

Substituição SQL

Ao adicionar uma cláusula WHERE a uma transformação de Pesquisa usando cache dinâmico, conecte uma transformação Filtro antes da transformação de Pesquisa para filtrar linhas que você não queira inserir na tabela de destino ou do cache. Se você não incluir a transformação Filtro, poderá obter resultados inconsistentes entre o cache e a tabela de destino.

Por exemplo, você configura uma transformação de Pesquisa para realizar uma pesquisa dinâmica na tabela de funcionários EMP, correspondendo linhas por EMP_ID. Você define a seguinte substituição SQL de pesquisa:

```
SELECT EMP_ID, EMP_STATUS FROM EMP ORDER BY EMP_ID, EMP_STATUS WHERE EMP_STATUS = 4
```

Quando você executa pela primeira vez a sessão, o Serviço de Integração cria o cache de pesquisa da tabela de destino com base na substituição SQL de pesquisa. Todas as linhas no cache corresponderão à condição na cláusula WHERE, EMP_STATUS = 4.

Por exemplo, o Serviço de Integração lê uma linha de origem que atende à condição de pesquisa especificada, mas o valor de EMP_STATUS é 2. Embora o destino possa ter a linha onde EMP_STATUS é 2, o Serviço de Integração não encontra a linha no cache devido à substituição SQL. O Serviço de Integração insere a linha no cache e passa a linha para a tabela de destino. Quando o Serviço de Integração inserir essa linha na tabela de destino, você deverá ter resultados consistentes se a linha já existir. Além disso, nem todas as linhas do cache correspondem à condição da cláusula WHERE na substituição SQL.

Para verificar se você inseriu no cache apenas linhas que correspondam à cláusula WHERE, adicione uma transformação Filtro antes da transformação de Pesquisa e defina a condição de filtro como a condição na cláusula WHERE na Substituição SQL de Pesquisa.

Para o exemplo acima, digite a seguinte condição de filtro na transformação Filtro e a cláusula WHERE na substituição SQL:

```
EMP_STATUS = 4
```

Valores de Transformação Pesquisa

A transformação de Pesquisa contém valores para as portas de entrada, pesquisa e saída. Se você habilitar um cache de pesquisa dinâmica, os valores da porta de saída variarão dependendo de como você o configura.

A transformação de Pesquisa contém os seguintes tipos de valores:

Valor de entrada

Valor que o Serviço de Integração transfere à transformação de Pesquisa.

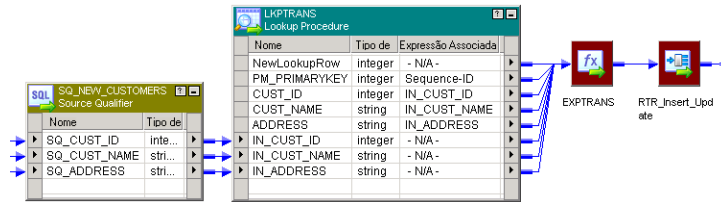
Valor de pesquisa

Valor que o Serviço de Integração insere no cache.

Valor de saída

Valor que o Serviço de Integração transmite da porta de saída da transformação de Pesquisa. O valor de saída da porta de pesquisa/saída depende do fato de haver uma escolha de efetuar a saída de um valor antigo ou novo quando o Serviço de Integração atualiza uma linha.

Por exemplo, a transformação de Pesquisa a seguir usa um cache de pesquisa dinâmica:



Por exemplo, você cria um mapeamento com os seguintes objetos:

```
Source > Lookup transformation > Expression transformation > Router transformation > Target
```

Você ativa o cache de pesquisa dinâmica para a transformação de Pesquisa.

Defina a seguinte condição de pesquisa:

```
IN_CUST_ID = CUST_ID
```

Por padrão, o tipo de todas as linhas inseridas na transformação de Pesquisa é inserir. Para efetuar inserções e atualizações no cache e na tabela de destino, selecione a propriedade **Inserir ou Atualizar** na transformação de Pesquisa.

As seções a seguir descrevem os valores das linhas no cache, as linhas de entrada, as linhas de pesquisa e as linhas de saída ao executar a sessão.

As seções a seguir descrevem os valores das linhas do cache, as linhas de entrada, linhas de pesquisa e linhas de saída ao executar a sessão.

Valores de Cache Iniciais

Quando você executa a sessão, o Serviço de Integração cria o cache de pesquisa da tabela de destino com os dados a seguir:

PK_PRIMARYKEY	CUST_ID	CUST_NAME	ADDRESS
100001	80001	Marion James	100 Main St.
100002	80002	Laura Jones	510 Broadway Ave.
100003	80003	Shelley Lau	220 Burnside Ave.

Valores de Entrada

A origem contém linhas que existem e outras que não existem na tabela de destino. As seguintes linhas são transferidas à transformação de Pesquisa a partir da transformação Qualificador de Origem:

SQ_CUST_ID	SQ_CUST_NAME	SQ_ADDRESS
80001	Marion Atkins	100 Main St.
80002	Laura Gomez	510 Broadway Ave.

SQ_CUST_ID	SQ_CUST_NAME	SQ_ADDRESS
99001	Jon Freeman	555 6th Ave.

Valores de Pesquisa

O Serviço de Integração procura valores no cache com base na condição de pesquisa. Ele atualiza linhas no cache para as IDs de clientes atuais 80001 e 80002. Ele insere uma linha no cache para a ID de cliente 99001. O Serviço de Integração gera uma nova chave (PK_PRIMARYKEY) para a nova linha.

A tabela a seguir mostra as linhas e os valores retornados da pesquisa:

PK_PRIMARYKEY	CUST_ID	CUST_NAME	ADDRESS
100001	80001	Marion Atkins	100 Main St.
100002	80002	Laura Gomez	510 Broadway Ave.
100004	99001	Jon Freeman	555 6th Ave.

Valores de Saída

O Serviço de Integração sinaliza as linhas na transformação de Pesquisa com base nas inserções e atualizações que executa no cache dinâmico. Estas linhas passam por uma transformação Expressão até uma transformação Roteador, que filtra e transfere as linhas inseridas e atualizadas a uma transformação Estratégia de Atualização. A transformação Estratégia de Atualização sinaliza as linhas com base no valor da porta NewLookupRow.

Os valores de saída das portas de pesquisa/saída e entrada/saída dependem do fato de você escolher valores novos ou antigos de saída quando o Serviço de Integração atualizar uma linha. No entanto, os valores de saída da porta NewLookupRow e de qualquer porta de pesquisa/saída que usa a ID da sequência são os mesmos para as linhas novas e atualizadas.

Quando você escolhe gerar a saída de novos valores, as portas de pesquisa/saída geram a saída dos valores a seguir:

NewLookupRow	PK_PRIMARYKEY	CUST_ID	CUST_NAME	ADDRESS
2	100001	80001	Marion Atkins	100 Main St.
2	100002	80002	Laura Gomez	510 Broadway Ave.
1	100004	99001	Jon Freeman	555 6th Ave.

Quando você escolhe gerar a saída de valores antigos, as portas de pesquisa/saída geram a saída dos seguintes valores:

NewLookupRow	PK_PRIMARYKEY	CUST_ID	CUST_NAME	ADDRESS
2	100001	80001	Marion James	100 Main St.
2	100002	80002	Laura Jones	510 Broadway Ave.
1	100004	99001	Jon Freeman	555 6th Ave.

Quando o Serviço de Integração atualiza as linhas no cache de pesquisa, ele usa valores da chave primária (PK_PRIMARYKEY) para as linhas no cache e a tabela de destino.

O Serviço de Integração usa a ID de sequência para gerar uma chave primária para o cliente que não encontra no cache. O Serviço de Integração insere o valor da chave primária no cache de pesquisa e retorna o valor para a porta de pesquisa/saída.

O Serviço de Integração gera como saída os valores das portas de entrada/saída que correspondem aos valores de entrada.

Nota: Se o valor de entrada for NULL e você selecionar a propriedade Ignorar Nulo para a porta de entrada associada, o valor de entrada não é igual ao valor de pesquisa ou ao valor gerado como saída da porta de entrada/saída. Quando você seleciona a propriedade Ignorar Nulo, o cache de pesquisa e a tabela de destino podem perder a sincronia se você transferir os valores nulos para o destino. Você deve certificar-se de não transferir valores nulos para o destino.

Atualizações do Cache de Pesquisa Dinâmica

O Serviço de Integração atualiza o cache dinâmico com base no tipo de linha, no resultado da consulta e nas propriedades da transformação de Pesquisa.

Ao usar um cache de pesquisa dinâmica, defina o tipo das linhas recebidas pela transformação Pesquisa, como inserir ou atualizar. Você pode definir algumas linhas como atualizar ou todas como inserir ou atualizar. Por padrão, o tipo de todas as linhas recebidas por uma transformação Pesquisa é inserir. Você pode adicionar uma transformação Estratégia de Atualização antes da transformação Pesquisa para definir o tipo de linha como atualizar.

O Serviço de Integração inserirá ou atualizará linhas no cache ou não alterará o cache. Por padrão, a transformação de Pesquisa insere uma linha no cache quando o tipo de linha é de inserção e atualiza uma linha quando o tipo de linha é de atualização.

No entanto, você pode configurar as seguintes propriedades de Pesquisa para alterar o modo como o Serviço de Integração lida com inserções e atualizações no cache:

- **Inserir ou Atualizar.** Aplica-se às linhas inseridas na transformação de Pesquisa com o tipo de linha inserir. Use essa opção para atualizar a linha no cache se os dados já existem nele. Se você não habilitar essa opção, o Serviço de Integração inserirá as linhas no cache, independentemente de elas existirem.
- **Atualizar ou Inserir.** Aplica-se às linhas inseridas na transformação de Pesquisa com o tipo de linha atualizar. Use essa opção para inserir uma linha de atualização se os dados não existirem no cache. Se você não habilitar essa opção, o Serviço de Integração ignorará a linha se ele não existir no cache.

Nota: Você pode selecionar a propriedade Inserir ou Atualizar ou Atualizar ou Inserir, ou pode selecionar ambas ou nenhuma.

É possível configurar a Transformação Pesquisa para sincronizar a origem de Pesquisa e o cache dinâmico. Ao sincronizar a origem de Pesquisa e o cache, a Transformação Pesquisa insere linhas diretamente na origem de pesquisa. Não se transfere linhas de inserção ao destino. Use esta configuração para executar várias sessões com transformações de Pesquisa que inserem linhas no mesmo destino.

Inserir ou Atualizar

Use a propriedade Inserir ou Atualizar para atualizar linhas existentes no cache de pesquisa dinâmica quando o tipo de linha for inserir.

Essa propriedade aplica-se somente a linhas que inserem a transformação de Pesquisa com o tipo de linha inserir. Quando uma linha de qualquer outro tipo, como atualizar, acessa a transformação de Pesquisa, a propriedade **Inserir ou Atualizar** não tem efeito sobre como o Serviço de Integração manipula a linha.

Quando você selecionar **Inserir ou Atualizar** e o tipo de linha que insere a transformação de Pesquisa for de inserção, o Serviço de Integração inserirá a linha no cache, caso seja nova. Se a linha existir no cache de índice, mas o cache de dados for diferente da linha atual, o Serviço de Integração atualiza a linha no cache de dados.

Caso você não selecione **Inserir ou Atualizar** e o tipo de linha que entra na transformação de Pesquisa for de inserção, o Serviço de Integração insere a linha no cache se ela for nova e não efetua alterações no cache se a linha já existir.

A seguinte tabela descreve como o Serviço de Integração altera o cache de pesquisa quando o tipo das linhas recebidas na transformação de Pesquisa for inserir:

Opção para Inserir ou Atualizar	Linha encontrada no cache	Cache de dados diferente	Resultado de cache de pesquisa	Valor de NewLookupRow
Eliminado - apenas inserir	Sim	-	Sem alteração	0
Eliminado - apenas inserir	Não	-	Inserir	1
Selecionado	Sim	Sim	Atualizar	2 ¹
Selecionado	Sim	Não	Sem alteração	0
Selecionado	Não	-	Inserir	1

¹ Se você selecionar **Ignorar Nulo** para todas as portas de pesquisa que não estejam na condição de pesquisa e se todas as portas contiverem valores nulos, o Serviço de Integração não alterará o cache e o valor de NewLookupRow será igual a 0.

Atualizar ou Inserir

Use a propriedade Atualizar ou Inserir para inserir novas linhas no cache de pesquisa dinâmica quando o tipo de linha é atualizar.

Você pode selecionar a propriedade **Atualizar ou Inserir** na transformação de Pesquisa. Essa propriedade se aplica apenas a linhas recebidas pela transformação de Pesquisa com o tipo de linha atualizar. Quando uma linha de qualquer outro tipo (como inserir) chega à transformação de Pesquisa, essa propriedade não tem efeito sobre como o Serviço de Integração manipula a linha.

Quando você seleciona essa propriedade e o tipo de linha recebido pela transformação de Pesquisa é atualizar, o Serviço de Integração atualizará a linha no cache se ela existir no cache de índice e os dados do cache forem diferentes dos da linha. O Serviço de Integração inserirá a linha no cache se ela for nova.

Se você não selecionar essa propriedade e o tipo de linha recebida pela transformação de Pesquisa for atualizar, o Serviço de Integração atualizará a linha no cache se ela existir e não alterará o cache se a linha for nova.

Se você selecionar **Ignorar Nulo** para todas as portas de pesquisa que não estejam na condição de pesquisa, e se todas as portas contiverem valores nulos, o Serviço de Integração não alterará o cache, e o valor de NewLookupRow será igual a 0.

A tabela a seguir descreve como o Serviço de Integração altera o cache de pesquisa quando o tipo das linhas recebidas na transformação de Pesquisa for atualizar:

Opção Atualizar ou Inserir	Linha encontrada no cache	Cache de dados diferente	Resultado de cache de pesquisa	Valor de NewLookupRow
Eliminado (apenas atualizar)	Sim	Sim	Atualizar	2
Eliminado (apenas atualizar)	Sim	Não	Sem alteração	0
Eliminado (apenas atualizar)	Não	-	Sem alteração	0
Selecionado	Sim	Sim	Atualizar	2
Selecionado	Sim	Não	Sem alteração	0
Selecionado	Não	-	Inserir	1

Mapeamentos com pesquisas dinâmicas

Quando você configura um mapeamento que contém uma transformação Pesquisa dinâmica, configure as transformações upstream a partir da transformação Pesquisa para afetar como a transformação Pesquisa atualiza o cache dinâmico. Configure as transformações downstream a partir da transformação Pesquisa para garantir que o Serviço de Integração atualize as linhas marcadas para a atualização e insira as que estão marcadas para a inserção no destino.

Configurando a Transformação de Estratégia de Atualização Upstream

Você pode configurar uma transformação de Estratégia de Atualização upstream para alterar o tipo das linhas que a transformação de Pesquisa recebe.

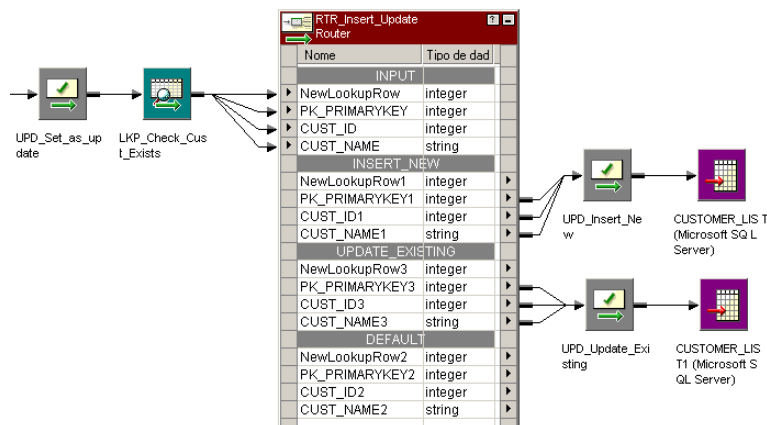
Quando você usar um cache de pesquisa dinâmica, use as transformações Estratégia de Atualização para definir o tipo das seguintes linhas:

- **Linhas que entram na transformação Pesquisa.** Por padrão, o tipo de todas as linhas recebidas por uma Transformação Pesquisa é inserir. No entanto, use uma transformação Estratégia de Atualização antes de uma transformação Pesquisa, para definir as linhas como atualização, ou algumas como atualização e outras como inserção.
- **Linhas que saem da Transformação Pesquisa.** O valor NewLookupRow indica como o Serviço de Integração alterou o cache de pesquisa, mas não altera o tipo de linha. Use uma transformação Filtro ou Roteador depois da transformação Pesquisa, para direcionar as linhas que saem da transformação Pesquisa com base no valor NewLookupRow. Use as transformações Estratégia de Atualização depois da transformação Filtro ou Roteador, para sinalizar as linhas para inserção ou atualização antes de definir o destino no mapeamento.

Nota: Se quiser descartar as linhas inalteradas, não conecte linhas da transformação de Filtro ou Roteador ao destino em que NewLookupRow é igual a 0.

Quando você define o tipo de linha como inserção, para as linhas que entram em uma transformação Pesquisa, use a propriedade Insert Else Update na transformação Pesquisa. Quando você define o tipo de linha como atualização, para as linhas inseridas em uma Transformação Pesquisa, use a propriedade Update Else Insert na Transformação Pesquisa. Se você definir algumas linhas que entram em uma transformação Pesquisa como atualização e outras como inserção, use a propriedade Update Else Insert, Insert Else Update, ou ambas.

A figura a seguir mostra um mapeamento com múltiplas transformações Estratégia de Atualização e uma transformação Pesquisa, usando um cache dinâmico:



Neste exemplo, a transformação Estratégia de Atualização antes da transformação Pesquisa sinaliza todas as linhas como atualização. Selecione a propriedade Update Else Insert na Transformação Pesquisa. A transformação Roteador envia as linhas inseridas à transformação Estratégia de Atualização Insert_New e envia as linhas atualizadas para a transformação Estratégia de Atualização Update_Existing. As linhas de saída que não sejam conectadas ao destino serão descartadas. As duas transformações Estratégia de Atualização à direita da transformação Pesquisa sinalizam as linhas como inserção ou atualização para o destino.

Configurando transformações Downstream

Configure transformações downstream para garantir que o cache de pesquisa dinâmica e o destino sejam sincronizados.

Ao usar um cache de pesquisa dinâmica, o Serviço de Integração registra no cache de pesquisa antes de registrar na tabela de destino. O cache de pesquisa e a tabela de destino podem se tornar não sincronizados se o Serviço de Integração não registrar os dados no destino. Por exemplo, o banco de dados de destino pode rejeitar os dados.

Considere as seguintes orientações para manter o cache de pesquisa sincronizado com a tabela de pesquisa:

- Use uma transformação Roteador para passar linhas ao destino armazenado quando o valor NewLookupRow for igual a um ou dois.
- Use a transformação Roteador para descartar linhas quando o valor NewLookupRow for igual a zero. Ou, promova a saída das linhas para outro destino.
- Use transformações Estratégia de Atualização depois da transformação de Pesquisa para sinalizar linhas para inserção ou atualização no destino.

- Defina o limite de erro para um ao executar uma sessão. Ao definir o limite de erro para um, haverá falha na sessão quando o primeiro erro for encontrado. O Serviço de Integração não registra os novos arquivos de cache no disco. Ao invés disso, recupera os arquivos de cache originais, caso existam. Também é necessário restabelecer a tabela de destino pré-sessão ao banco de dados de destino.
- Verifique se a transformação de Pesquisa gera para o destino os mesmos valores que o Serviço de Integração grava no cache de pesquisa. Se você quiser transferir valores novos na atualização, conecte apenas portas de pesquisa/saída à tabela de destino, em vez de portas de entrada/saída. Quando quiser transferir valores antigos na atualização, adicione uma transformação Expressão depois da transformação de Pesquisa e antes da transformação Roteador. Adicione portas de saída à transformação Expressão para cada porta na tabela de destino e crie expressões para garantir que valores nulos não sejam transferidos ao destino.
- Defina a propriedade Tratar Linhas de Origem Como para Controlado por Dados nas propriedades da sessão.
- Selecione Inserir e Atualizar como Atualização quando você define as opções de tabela de destino da estratégia de atualização. Isso garante que o Serviço de Integração irá atualizar as linhas marcadas para atualização e inserir linhas marcadas para inserção. Selecione essas opções na Exibição de Transformação da guia Mapeamento nas propriedades da sessão.

Valor Nulo em Colunas de Condição de Pesquisa

O Serviço de Integração manipula linhas com valores nulos em colunas de condição de pesquisa de maneira diferente, dependendo de a linha existir ou não no cache.

Se uma linha não existir no cache de pesquisa, o Serviço de Integração inserirá essa linha no cache e a transferirá para a tabela de destino. Se a linha existir no cache de pesquisa, o Serviço de Integração não a atualizará no cache nem na tabela de destino.

Nota: Se os dados de origem contiverem valores nulos nas colunas de condição de pesquisa, defina o limite de erro para um. Isso garante que o cache e a tabela de pesquisa permaneçam sincronizados se o Serviço de Integração inserir uma linha no cache, mas o banco de dados rejeita a linha devido a uma restrição de Não Nulo.

Configurando sessões com um cache de pesquisa dinâmica

Quando você configura uma sessão com uma transformação Estratégia de Atualização e um cache de pesquisa dinâmica, deve configurar uma das seguintes opções da tabela de estratégia de atualização:

- Selecione Inserir
- Selecionar Atualização como Atualização
- Não selecione Excluir

A opção de tabela de origem de estratégia de atualização garante que o Serviço de Integração atualizará linhas marcadas para atualização e inserirá linhas marcadas para inserção.

Nas configurações de Opções Gerais na guia Propriedades nas propriedades da sessão, defina a opção Tratar Linhas de Origem Como como Controlada por Dados. Se você não escolher Controlado por Dados, o Serviço de Integração sinalizará todas as linhas para o tipo de linha que você especificar na opção Tratar Linhas de Origem Como e não usará as transformações de Estratégia de Atualização no mapeamento para sinalizar as linhas. O Serviço de Integração não insere e atualiza as linhas corretas. Se você não escolher Atualização como Atualização, o Serviço de Integração não atualiza corretamente as linhas sinalizadas para serem atualizadas na tabela de destino. Consequentemente, o cache de pesquisa e a tabela de destino podem se tornar não sincronizados.

Atualizações de Cache Dinâmico Condicional

É possível atualizar cache de pesquisa dinâmica com base nos resultados de uma expressão booleana. O Serviço de Integração atualiza o cache quando a expressão é verdadeira.

Por exemplo, é possível que se tenha um número de produto, quantidade em estoque e uma coluna de carimbo de data/hora em uma tabela de destino. É necessário atualizar a quantidade em estoque com os valores de origem mais atuais. É possível atualizar a quantidade em estoque quando os dados de origem têm um carimbo de data/hora maior que o carimbo de data/hora do cache dinâmico. Crie uma expressão na Transformação Pesquisa que seja semelhante à expressão a seguir:

```
lookup_timestamp < input_timestamp
```

A expressão pode ter as portas de pesquisa e de entrada. É possível acessar variáveis internas, de mapeamento e de parâmetros. É possível incluir funções definidas pelo usuário e consultar transformações desconectadas.

A expressão retornará true, false ou NULL. Se o resultado da expressão for NULL, a expressão é falsa. O Serviço de Integração não atualiza o cache. É possível acrescentar uma marca para valores NULL na expressão se precisar alterar o resultado da expressão para true. O valor padrão da expressão é true.

Crie a expressão com o Transformation Developer. Não é possível substituir a **Condição de Atualização de Cache Dinâmico** no nível de sessão.

Processamento de Sessão

Quando você configurar uma atualização de cache dinâmico condicional, o Serviço de Integração não considerará a expressão condicional se os dados não existirem. Quando você habilitar Insert Else Update, o Serviço de Integração inserirá uma linha no cache se os dados não existirem. Se os dados existirem, ele atualizará o cache quando a expressão condicional for verdadeira. Quando você habilitar Update Else Insert, o Serviço de Integração atualizará o cache se os dados existirem no cache e a expressão condicional for verdadeira. Quando os dados não existem no cache, o Serviço de Integração insere a linha no cache.

Quando a expressão for verdadeira, o valor NewLookupRow será um e o Serviço de Integração atualizará a linha. Quando a expressão for falsa ou NULL, o valor de NewLookupRow será zero. O Serviço de Integração não atualiza a linha.

Se você configurar a transformação de Pesquisa para sincronizar o cache dinâmico, o Serviço de Integração inserirá uma linha na origem de pesquisa quando inserir uma linha no cache.

Configuração de Pesquisa de Cache Dinâmico Condicional

Você deve ativar a Transformação Pesquisa para executar uma pesquisa condicional de cache dinâmico para poder criar uma expressão condicional.

Siga as seguintes etapas para configurar uma pesquisa de cache dinâmico condicional:

1. Crie uma Transformação Pesquisa dinâmica no Transformation Developer.
2. Clique na guia Propriedades.
3. Habilite as propriedades Armazenamento em Cache de Pesquisa e Cache de Pesquisa Dinâmica.
4. Para acessar o Editor de Expressão, clique no botão Ir para abrir a propriedade de Condição de Atualização de Cache Dinâmico.
5. Defina uma condição de expressão. É possível selecionar portas de entrada, portas de pesquisa e funções para a expressão.

6. Clique em Validar para verificar se a expressão é válida.

Se você não validar uma expressão, o Designer o fará quando você fechar o Editor de Expressão. Se a expressão não for válida, o Designer exibirá um aviso.

Atualização de cache dinâmico com resultados de expressões

Você pode atualizar os valores do cache de pesquisa dinâmica com os resultados de uma expressão.

Por exemplo, um destino da tabela de produto tem uma coluna numérica que contém uma contagem dos pedidos. Cada vez que a transformação de Pesquisa recebe um pedido do produto, ela atualiza o cache dinâmico `order_count` com os resultados da seguinte expressão:

```
order_count = order_count + 1
```

A transformação de Pesquisa retorna `order_count`.

Você pode configurar como o Serviço de Integração trata o caso quando a expressão é avaliada como nula.

Valores nulos da expressão

A expressão retorna NULL se um dos valores da expressão for nulo. No entanto, você pode configurar uma expressão para retornar um valor não nulo.

Se a expressão se refere a uma porta de pesquisa, mas os dados de origem são novos, a porta de pesquisa contém um valor padrão. O padrão pode ser NULL. Você pode configurar uma expressão `IsNull` para verificar valores nulos.

Por exemplo, a seguinte expressão verifica se `lookup_column` é NULL:

```
iif (isnull(lookup_column), input_port, user_expression)
```

Se a coluna for nula, então é retornado o valor `input_port`. Caso contrário, retorne o valor da expressão.

Processamento de Sessão

O Serviço de Integração pode inserir e atualizar linhas no cache de pesquisa dinâmica com base nos resultados de expressões. Os resultados da expressão podem variar se o valor de porta de pesquisa é NULL e está incluído na expressão.

Quando você habilitar uma instrução Inserir ou Atualizar, o Serviço de Integração inserirá uma linha com o resultado da expressão se os dados não estiverem no cache. O valor da porta de pesquisa será NULO quando não houver dados no cache. Caso a expressão se refira ao valor da porta de pesquisa, o Serviço de Integração substituirá o valor da porta padrão na expressão. Quando você habilitar Inserir ou Atualizar e houver dados no cache, o Serviço de Integração atualizará o cache com o resultado da expressão.

Quando você habilitar Atualizar ou Inserir, o Serviço de Integração atualizará o cache com o resultado da expressão se houver dados no cache. Quando os dados não estiverem no cache, o Serviço de Integração inserirá uma linha que contenha o resultado da expressão. Se a expressão se referir a um valor de porta de pesquisa, o Serviço de Integração substituirá o valor de porta padrão na expressão.

Se você configurar a transformação de Pesquisa para sincronizar o cache dinâmico, o Serviço de Integração inserirá uma linha no cache de pesquisa com o resultado da expressão. Ele inserirá uma linha na origem da pesquisa com o resultado da expressão.

Configurando uma Expressão para Atualizações de Cache Dinâmico

É necessário ativar a Transformação Pesquisa para executar pesquisas dinâmicas antes de poder criar uma expressão condicional.

Siga estas etapas para configurar uma expressão para pesquisa de cache dinâmico:

1. Crie uma transformação de Pesquisa dinâmica no Transformation Developer.
2. Abra a guia Propriedades.
3. Habilite as propriedades Armazenamento em Cache de Pesquisa e Cache de Pesquisa Dinâmica.
4. Para criar uma expressão, clique na lista de Expressão Associada referente à porta de pesquisa que você deseja atualizar.
5. Selecione a Expressão Associada.
O Editor de Expressão será exibido.
6. Defina uma expressão selecionando portas de entrada, portas de pesquisa e funções. O valor de retorno da expressão deve corresponder ao tipo de porta de pesquisa.
7. Clique em Validar para verificar se a expressão é válida.

Se você não validar uma expressão, o Designer o fará quando você fechar o Editor de Expressão. Se a expressão não for válida, o Designer exibirá um aviso.

Sincronizando o Cache com a Origem da Pesquisa

Uma transformação de pesquisa mantém um cache de pesquisa dinâmica para rastrear as linhas que ela passa para o destino. Quando várias sessões atualizam o mesmo destino, você pode configurar a transformação de pesquisa em cada sessão, para sincronizar o cache de pesquisa dinâmica com a mesma origem da pesquisa, e não com um destino.

Quando você configura uma transformação de pesquisa para sincronizar o cache com a origem da pesquisa, a transformação de pesquisa executa uma pesquisa na origem da pesquisa. Se os dados não existirem na origem da pesquisa, a transformação de pesquisa inserirá a linha na origem da pesquisa antes de atualizar o cache de pesquisa dinâmica.

Se outra sessão tiver inserido a linha, pode ser que os dados existam na origem da pesquisa. Para sincronizar o cache de pesquisa com a origem da pesquisa, o Serviço de Integração recupera os valores mais recentes da origem da pesquisa. A transformação de pesquisa insere os valores da origem da pesquisa no cache de pesquisa dinâmica. É preciso que a origem da pesquisa seja uma tabela relacional.

Por exemplo, você tem várias sessões sendo executadas ao mesmo tempo. Cada uma delas gera números de produto para novos nomes de produto. Quando uma sessão gera um número de produto, as outras devem usar o mesmo número de produto para identificar o produto. O número do produto é gerado uma vez e inserido na origem da pesquisa. Se outra sessão processar uma linha que contém o produto, ela deverá usar o número de produto que está na origem da pesquisa. Cada sessão executa uma pesquisa na origem da pesquisa para determinar quais números de produto já foram gerados.

O Serviço de Integração executa as seguintes tarefas para inserir linhas:

- O Serviço de Integração executa uma pesquisa no cache de pesquisa dinâmica. Se os dados não existirem no cache de pesquisa dinâmica, o Serviço de Integração executará uma pesquisa na origem da pesquisa.

- Se eles existirem na origem da pesquisa, o Serviço de Integração recuperará os dados na origem da pesquisa. Ele insere uma linha no cache de pesquisa dinâmica com as colunas da origem da pesquisa. Ele não atualiza o cache com a linha de pesquisa.
- Se os dados não estiverem na origem da pesquisa, o Serviço de Integração inserirá os dados na origem da pesquisa e inserirá a linha no cache.

A origem da pesquisa contém as mesmas colunas que o cache de pesquisa. O Serviço de Integração só inserirá uma coluna no cache de pesquisa se a coluna for projetada na transformação de pesquisa ou fizer parte de uma condição de pesquisa.

NewLookupRow

Quando você sincroniza o cache dinâmico com a origem da pesquisa, o comportamento da transformação de Pesquisa não muda para as linhas de atualização. Quando o Serviço de Integração recebe as linhas de atualização, ele atualiza o cache de pesquisa dinâmica e configura o valor do NewLookupRow como 2. Ele não atualiza a origem da pesquisa. Você pode encaminhar as linhas de atualização através da transformação Estratégia de atualização e transferir as linhas para o destino.

Quando o Serviço de Integração insere linhas na origem da pesquisa, ele configura o NewLookupRow como 1. O Serviço de Integração atualiza o cache de pesquisa dinâmica. Quando a transformação está configurada para sincronizar o cache dinâmico e NewLookupRow é 1, você não precisa transferir as linhas de inserção para o destino.

Quando você configura a propriedade Insert Else Update na transformação da Pesquisa, e a linha da origem é uma linha de inserção, o Serviço de Integração insere a linha no cache e na origem da pesquisa, ou atualiza o cache.

Quando você configura a propriedade Update Else Insert na transformação da Pesquisa, e a linha da origem é uma linha de atualização, o Serviço de Integração atualiza o cache e ou insere a linha no cache e na origem da pesquisa.

Nota: Quando sessões diferentes processam as linhas de atualização para a mesma tabela, a ordem do processamento varia. Você pode obter resultados inesperados.

Configuração da sincronização de cache dinâmico

Quando você cria a transformação da Pesquisa, pode configurá-la para sincronizar o cache dinâmico com a origem da pesquisa para as linhas da inserção.

Para configurar sincronização de cache dinâmico:

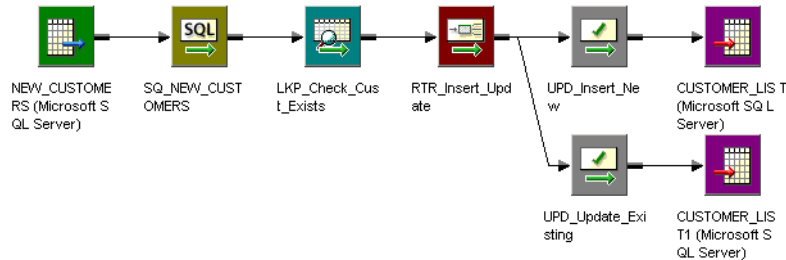
1. Abra a guia Propriedades da transformação da Pesquisa.
2. Ativar cache de pesquisa dinâmica.
3. Ative Sincronizar cache dinâmico.

Exemplo de Cache de Pesquisa Dinâmica

Use um cache de pesquisa dinâmica quando precisar inserir e atualizar linhas no destino. Quando você usa um cache de pesquisa dinâmica, pode inserir e atualizar o cache com os mesmos dados que você transfere ao destino para inserir e atualizar.

Por exemplo, você precisa atualizar uma tabela que contém dados do cliente. Os dados de origem contêm linhas de dados do cliente para inserir ou atualizar no destino. Crie um cache dinâmico que represente o destino. Configure uma transformação Pesquisa para os clientes de pesquisa no cache.

A figura a seguir mostra um mapeamento que possui um cache dinâmico:



A transformação de Pesquisa usa um cache de pesquisa dinâmico. Quando a sessão inicia, o Serviço de Integração cria o cache de pesquisa a partir de uma tabela Customer_List. A tabela Customer_List também é o destino no mapeamento. Quando o Serviço de Integração lê uma linha que não está no cache de pesquisa, ele insere a linha no cache.

A transformação Pesquisa retorna a linha para uma transformação Roteador. A transformação Roteador direciona a linha para a transformação UPD_Insert_New ou UPD_Update_Existing. As transformações Estratégia de Atualização marcam a linha como inserção ou atualização, antes de transferi-la para o destino.

A tabela Customer_List muda conforme a execução da sessão. O Serviço de Integração insere novas linhas e atualiza as existentes no cache de pesquisa. O Serviço de Integração mantém o cache de pesquisa e a tabela Customer_List sincronizados.

Para gerar chaves para o destino, use Sequence-ID na porta associada. A ID de sequência gera chaves primárias para as novas linhas que o Serviço de Integração insere na tabela de destino.

Você aumenta o desempenho de sessão quando usa um cache de pesquisa dinâmico, porque cria o cache uma vez a partir do banco de dados. Você pode continuar usando o cache de pesquisa, embora os dados da tabela de destino mudem.

Regras e Diretrizes para Cache de Pesquisa Dinâmica

Certas regras e diretrizes são aplicáveis quando você usa um cache de pesquisa dinâmico.

Use as seguintes diretrizes ao usar o cache de pesquisa dinâmico:

- Você deverá definir a propriedade **Em Várias Correspondências** como Relatar Erro quando usar um cache de pesquisa dinâmico. Para redefinir a propriedade, altere a pesquisa dinâmico para uma pesquisa estática, altere a propriedade e, em seguida, altere a pesquisa estática para uma pesquisa dinâmico.
- Não se pode compartilhar o cache entre uma transformação de Pesquisa dinâmico e outra estática no mesmo grupo de ordem de carregamento de destino.
- Você pode ativar um cache de pesquisa dinâmico para uma pesquisa relacional, de arquivo simples ou de transformação de Qualificador de Origem.
- Você pode ativar um cache de pesquisa dinâmico para uma pesquisa relacional ou de arquivo simples.
- A transformação de Pesquisa precisa ser uma transformação conectada.

- Você pode usar um cache persistente ou não persistente.
- Se o cache não for persistente, o Serviço de Integração sempre recompilará o cache a partir do banco de dados, mesmo se você não ativar **Efetuar Novamente o Cache de Origem da Pesquisa**.
- Se o cache não for persistente, o Serviço de Integração recompilará sempre o cache a partir do banco de dados, mesmo se você não ativar **Efetuar Novamente o Cache de Origem da Pesquisa**.
- Quando você sincroniza arquivos de cache dinâmico com uma tabela de origem de pesquisa, a transformação de Pesquisa insere linhas na tabela de origem de pesquisa e no cache de pesquisa dinâmica. Se a linha de origem for de atualização, a transformação de Pesquisa atualizará apenas o cache de pesquisa dinâmica.
- Só é possível criar uma condição de pesquisa de igualdade. Você não pode procurar um intervalo de dados em um cache dinâmico.
- Você deve associar cada porta de pesquisa que não esteja na condição de pesquisa a uma porta de entrada, uma ID de sequência ou uma expressão associada.
- Use uma transformação Roteador para passar linhas ao destino armazenado quando o valor NewLookupRow for igual a um ou dois.
- Use a transformação Roteador para descartar linhas quando o valor NewLookupRow for igual a zero. Ou você pode gerar a saída de linhas para outro destino.
- Verifique se o Serviço de Integração gera os mesmos valores que grava no cache de pesquisa para o destino. Se você quiser transferir valores novos na atualização, conecte apenas portas de pesquisa e saída à tabela de destino, em vez de as portas de entrada e saída. Quando quiser transferir valores antigos na atualização, adicione uma transformação Expressão depois da transformação de Pesquisa e antes da transformação Roteador. Adicione portas de saída à transformação Expressão para cada porta na tabela de destino e crie expressões para garantir que valores nulos não sejam transferidos ao destino.
- Ao usar uma substituição SQL de pesquisa, mapeie as colunas certas aos destinos adequados para pesquisa.
- Ao adicionar uma cláusula WHERE à substituição SQL de pesquisa, use uma transformação Filtro antes da transformação de Pesquisa. Isso garante que o Serviço de Integração insira ao cache dinâmico e à tabela de destino linhas que coincidam com a cláusula WHERE.
- Quando você configura uma transformação de Pesquisa reutilizável para usar cache dinâmico, não é possível editar a condição nem desativar a propriedade **Cache de Pesquisa Dinâmica** em um mapeamento.
- Use transformações Estratégia de Atualização depois da transformação de Pesquisa para sinalizar as linhas a inserir ou atualizar para o destino.
- Use uma transformação Estratégia de Atualização antes da transformação de Pesquisa para definir algumas ou todas as linhas como atualização, se quiser usar a propriedade Atualizar ou Inserir na transformação de Pesquisa.
- Defina o tipo de linha para Controlado por Dados nas propriedades da sessão.
- Selecione Inserir e Atualizar como Atualizar para as opções da tabela de destino nas propriedades da sessão.

CAPÍTULO 20

Transformação Normalizador

Este capítulo inclui os seguintes tópicos:

- [Visão Geral de Transformação Normalizador, 357](#)
- [Componentes da Transformação Normalizador, 358](#)
- [Chaves geradas pela transformação Normalizador, 361](#)
- [Transformação de Normalizador VSAM, 362](#)
- [Transformação de Normalizador de Pipeline, 366](#)
- [Usando uma Transformação de Normalizador em um Mapeamento, 370](#)
- [Solução de Problemas de Transformações Normalizador, 374](#)

Visão Geral de Transformação Normalizador

A transformação do Normalizador recebe uma linha que contém colunas de ocorrências múltiplas e retorna uma linha para cada instância dos dados de ocorrência múltipla. A transformação processa colunas ou grupo de colunas de várias ocorrências em cada linha de origem. A transformação do Normalizador é uma transformação ativa.

A transformação do Normalizador analisa as colunas de ocorrência múltipla de origens COBOL, tabelas relacionais ou outras origens. Ela pode processar tipos múltiplos de registro de uma origem COBOL que contenha uma cláusula REDEFINES.

Por exemplo, é possível que se tenha uma tabela relacional que armazene quatro trimestre de vendas por loja. É necessário criar uma linha para cada ocorrência de venda. É possível configurar um transformação de Normalizador para retornar uma única linha para cada trimestre.

As linhas de origem a seguir contêm quatro trimestres de vendas por loja:

```
Store1 100 300 500 700
Store2 250 450 650 850
```

O Normalizador retorna uma linha para cada combinação de loja e vendas. Também retorna um índice que identifica o número do trimestre:

```
Store1 100 1
Store1 300 2
Store1 500 3
Store1 700 4
Store2 250 1
Store2 450 2
Store2 650 3
Store2 850 4
```

A transformação do Normalizador gera uma chave para cada linha de origem. O Serviço de Integração aumenta o número de sequência de chave gerada cada vez que processa uma linha de origem. Quando a linha de origem contém uma coluna de ocorrência múltipla ou um grupo de colunas de ocorrência múltipla, a transformação do Normalizador retorna uma linha para cada ocorrência. Cada linha contém o mesmo valor-chave gerado.

Quando o Normalizador retorna diversas filas a partir de uma fila de origem, ele retorna dados duplicados para colunas de origem que ocorrem uma única vez. Por exemplo, Store1 e Store2 se repetem para cada instância de vendas.

É possível criar uma transformação de Normalizador de VSAM ou uma transformação de Normalizador de pipeline:

- **Transformação Normalizador de VSAM.** Uma transformação não reutilizável que é uma transformação de Qualificador de origem para uma origem COBOL. O Mapping Designer cria colunas de Normalizador de VSAM a partir de uma origem COBOL em um mapeamento. Os atributos da coluna são somente leitura. O Normalizador de VSAM recebe uma coluna de origem de ocorrência múltipla por meio de uma porta de entrada.
- **Transformação Normalizador de Pipeline.** Uma transformação que processa dados de ocorrência múltipla de tabelas relacionais ou arquivos simples. Você cria as colunas manualmente e as edita no Transformation Developer ou no Mapping Designer. A transformação do Normalizador de pipeline representa colunas de ocorrência múltipla com uma porta de entrada para cada ocorrência de coluna de origem.

Componentes da Transformação Normalizador

Uma transformação de Normalizador contém as seguintes guias:

- **Transformação.** Digite o nome e a descrição da transformação. A convenção de nomenclatura de uma transformação de Normalizador é `NRM_TransformationName`. Você também pode tornar a transformação de Normalizador reutilizável.
- **Portas.** Visualize as portas e os atributos da transformação.
- **Propriedades** Configure o nível de rastreamento para determinar a quantidade de detalhes de transação relatados no arquivo de registro da sessão. Opte por redefinir ou reiniciar o valor da sequência de chave gerada na próxima sessão.
- **Normalizador.** Defina a estrutura dos dados de origem. A guia Normalizador define os dados de origem como colunas e grupos de colunas.
- **Extensões de Metadados.** Configure o nome, o tipo de dados, a precisão e o valor da extensão. Também é possível criar extensões de metadados reutilizáveis.

Guia Portas

Quando define uma transformação Normalizador, você configura as colunas na guia Normalizador. O Designer cria as portas. Você pode ver os atributos e as portas do Normalizador na guia Portas.

As transformações Pipeline e Normalizador de VSAM representam colunas de origem recorrentes de modo diferente. A transformação Normalizador de VSAM tem uma porta de entrada para uma coluna recorrente. A transformação Normalizador do pipeline tem várias portas de entrada para uma coluna recorrente.

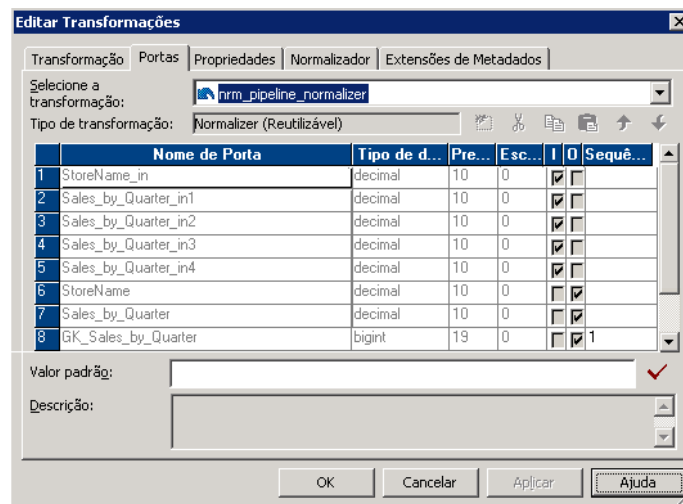
A transformação Normalizador tem uma porta de saída para cada porta de entrada recorrente. Quando uma coluna de origem é recorrente, as transformações Normalizador de VSAM e pipeline têm uma porta de saída para a coluna. A transformação retorna uma linha para cada ocorrência de coluna de origem.

A transformação Normalizador tem uma porta de ID de coluna gerada (GCID) para cada coluna recorrente. A ID de coluna gerada tem um índice para a instância dos dados recorrentes. Por exemplo, se uma coluna ocorrer quatro vezes em um registro de origem, o Normalizador retornará o valor 1, 2, 3 ou 4 na ID da coluna gerada com base na instância em que os dados recorrentes ocorrem na linha.

A convenção de nomenclatura da ID da coluna gerada do Normalizador é GCID_<nome_do_campo>.

A transformação Normalizador contém pelo menos uma porta de chave gerada. O Serviço de Integração aumenta o número de sequência de chave gerada cada vez que processa uma linha de origem.

A figura a seguir mostra a guia Portas da transformação Normalizador:



Neste exemplo, Sales_By_Quarter é recorrente na origem. A transformação Normalizador tem uma porta de saída para Sales_By_Quarter. Ele retorna quatro linhas para cada linha de origem. O valor inicial da chave gerada é 1.

Você pode alterar as portas em uma transformação Normalizador do pipeline editando as colunas na guia Normalizador. Para alterar a transformação Normalizar VSAM, você precisa alterar a origem COBOL e recriar a transformação.

Guia Propriedades

Configure as propriedades gerais da transformação de Normalizador na guia Propriedades.

Você pode configurar as seguintes propriedades da transformação Normalizador:

Propriedade	Obrigatório Opcional	Descrição
Redefinir	Obrigatório	Ao final de uma sessão, redefine a sequência de valores para cada valor-chave gerado para o valor anterior à sessão.
Reiniciar	Obrigatório	Inicia a sequência de chaves gerada em 1. Sempre que você executa uma sessão, o valor-chave da sequência inicia em 1 e substitui o valor da sequência na guia Portas.
Nível de rastreamento	Obrigatório	Define o volume de detalhes incluído no log da sessão quando você executa uma sessão que contém essa transformação.

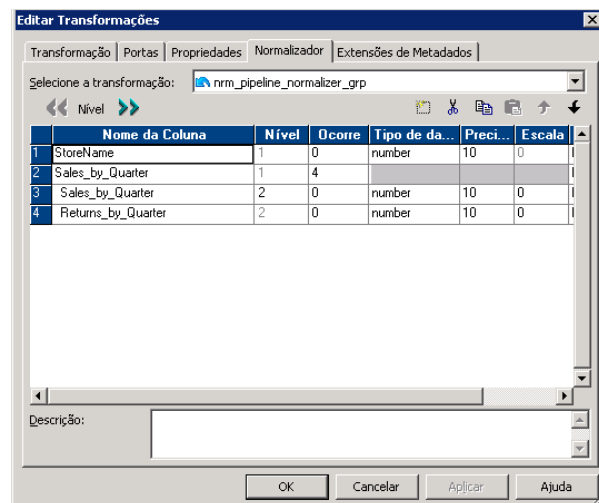
Guia Normalizador

A guia Normalizador define a estrutura dos dados de origem. A guia Normalizador define os dados de origem como colunas e grupos de colunas. Um grupo de colunas pode definir um registro em uma origem COBOL ou pode definir um grupo de campos de múltiplas ocorrências na origem.

O número do nível de colunas identifica grupos de colunas nos dados. Os números de níveis definem uma hierarquia de dados. As colunas em um grupo têm o mesmo número de nível e são exibidas de forma sequencial abaixo de uma coluna de nível de grupo. Uma coluna de nível de grupo tem um número de nível mais baixo e não contém nenhum dado.

A figura a seguir mostra a guia Normalizador de uma transformação de Normalizador de pipeline:

Figura 3. Guia Normalizador



Quarterly_Data é uma coluna de nível de grupo. É de Nível 1. O grupo Quarterly_Data ocorre quatro vezes em cada linha. Sales_by_Quarter e Returns_by_Quarter são colunas de Nível 2 e pertencem ao grupo.

Cada coluna tem um atributo Occurs. O atributo Occurs identifica as colunas ou grupos de colunas que ocorrem mais de uma vez em uma linha de origem.

Ao criar uma transformação de Normalizador de pipeline, é possível editar as colunas. Ao criar uma transformação de Normalizador de VSAM, a guia Normalizador é somente leitura.

A tabela a seguir descreve os atributos da guia Normalizador que são comuns às transformações de Normalizador de VSAM e pipeline:

Atributo	Descrição
Nome da Coluna	Nome da coluna de origem.
Nível	Colunas do grupo. As colunas no mesmo grupo ocorrem abaixo de uma coluna com um número de nível inferior. Quando cada coluna está no mesmo nível, a transformação não contém grupos de coluna.
Occurs	O número de instâncias de uma coluna ou grupo de colunas na linha de origem.
Tipo de dados	O tipo de dados da coluna de transformação pode ser String, String Numérica ou Número.
Prec	Precisão. Comprimento da coluna.
Escala	Número de posições decimais para uma coluna numérica.

A guia Normalizador de uma transformação de Normalizador de VSAM contém os mesmos atributos que a transformação de Normalizador de pipeline, mas também apresenta atributos que são exclusivos a uma definição de origem COBOL.

Chaves geradas pela transformação Normalizador

A transformação Normalizador retorna pelo menos uma coluna de chave gerada na linha de saída. O Serviço de Integração aumenta o número de sequência de chave gerada cada vez que processa uma linha de origem. O Serviço de Integração determina o valor da chave inicial a partir do valor da chave gerada na guia Portas da transformação Normalizador. Quando você cria uma transformação Normalizador, o valor da chave gerada é 1 por padrão. A convenção de nomenclatura para a chave gerada do Normalizador é GK_<redefined_field_name>.

TÓPICOS RELACIONADOS:

- [“Gerando valores-chave” na página 372](#)

Armazenando Valores-chave Gerados

Você pode exibir os valores-chave gerados atuais na guia Portas da transformação Normalizador. Ao fim de cada sessão, o Serviço de Integração atualiza o valor-chave gerado na transformação Normalizador para o último valor gerado para a sessão mais um. O valor-chave gerado máximo é 9.223.372.036.854.775.807. Se houver diversas instâncias da transformação Normalizador no repositório, o Serviço de Integração atualiza o valor-chave gerado em todas as versões quando executa uma sessão.

Nota: Você não pode alterar os valores de chave gerados atualmente na guia Portas da transformação de Normalizador.

Alterando os Valores-chave Gerados

É possível alterar o valor de chave gerado das seguintes maneiras:

- **Redefina a sequência-chave gerada.** Redefina a sequência de chaves gerada na guia Propriedades da transformação do Normalizador. Quando você redefine a sequência de chaves geradas, o Serviço de Integração redefine o valor inicial da chave gerada para o valor que ela tinha antes da sessão. Redefina a sequência de chaves geradas quando desejar criar os mesmos valores de chaves geradas sempre que executar a sessão.
- **Reinicie a sequência-chave gerada.** Reinicie a sequência-chave gerada na guia Propriedades da transformação do Normalizador. Ao reiniciar a sequência de chave gerada, o Serviço de Integração inicia a sequência de chaves geradas em 1 da próxima vez que executar uma sessão. Ao reiniciar a sequência de chaves geradas, o valor inicial das chaves geradas não é alterado na transformação do Normalizador até que você execute uma sessão. Ao executar uma sessão, o Serviço de Integração substitui o valor do número de sequência na guia Portas.

Ao redefinir ou reiniciar a sequência de chaves geradas, a redefinição ou reinicialização afeta os valores da sequência de chaves geradas na próxima vez que você executar uma sessão. Não se alteram os valores atuais de sequência de chaves geradas na Transformação Normalizador. Ao redefinir ou reiniciar a sequência de chaves geradas, a opção é ativada para cada sessão até que você desative a opção.

Transformação de Normalizador VSAM

A transformação Normalizador de VSAM é o qualificador de origem de uma definição de origem COBOL. Uma origem COBOL é um arquivo simples que pode conter dados recorrentes e vários tipos de registros no mesmo arquivo.

VSAM (Virtual Storage Access Method) é um método de acesso a arquivos para um sistema operacional mainframe IBM. Os arquivos VSAM organizam registros em arquivos simples indexados ou sequenciais. No entanto, é possível usar a transformação Normalizador de VSAM para qualquer origem de arquivo simples que você defina com uma definição de origem COBOL.

Uma definição de origem COBOL pode ter uma instrução OCCURS que define uma coluna recorrente. A definição de origem COBOL também pode conter uma instrução REDEFINES para definir mais de um tipo de registro no arquivo.

O copybook COBOL a seguir define um registro de vendas:

```
01 SALES_RECORD.  
  03 HDR_DATA.  
    05 HDR_REC_TYPE          PIC X.  
    05 HDR_STORE             PIC X(02).  
  03 STORE_DATA.  
    05 STORE_NAME            PIC X(30).  
    05 STORE_ADDR1           PIC X(30).  
    05 STORE_CITY            PIC X(30).  
  03 DETAIL_DATA REDEFINES STORE_DATA.  
    05 DETAIL_ITEM           PIC 9(9).  
    05 DETAIL_DESC           PIC X(30).  
    05 DETAIL_PRICE          PIC 9(4)V99.  
    05 DETAIL_QTY            PIC 9(5).  
    05 SUPPLIER_INFO OCCURS 4 TIMES.  
      10 SUPPLIER_CODE       PIC XX.  
      10 SUPPLIER_NAME       PIC X(8).
```

O arquivo de vendas pode conter dois tipos de registros de vendas. Store_Data define uma loja e Detail_Data define merchandise vendido na loja. A cláusula REDEFINES indica que os campos Detail_Data podem ocorrer em um registro, em vez de campos Store_Data.

Os três primeiros caracteres de cada registro de vendas são o cabeçalho. O cabeçalho inclui um tipo de registro e uma ID de loja. O valor Hdr_Rec_Type define se o restante do registro contém informações da loja ou de merchandise. Por exemplo, quando Hdr_Rec_Type é "S", o registro contém dados da loja. Quando Hdr_Rec_Type é "D", o registro contém dados de detalhe.

Quando o registro contém dados de detalhe, ele inclui os campos Supplier_Info. A cláusula OCCURS define quatro fornecedores em cada registro Detail_Data.

A figura a seguir mostra a definição de origem COBOL Sales_File que você poderia criar a partir do copybook COBOL:

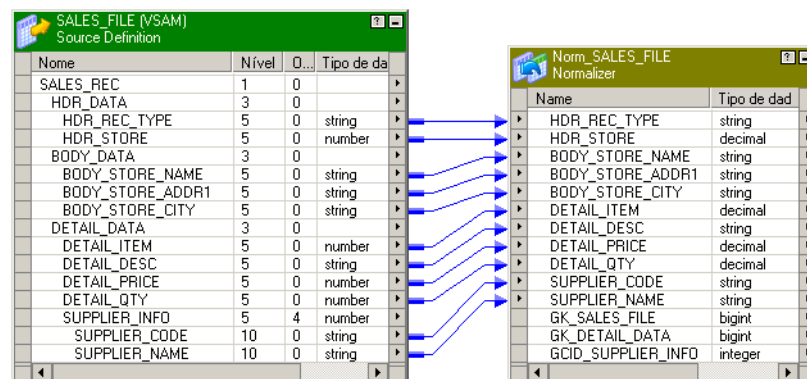
T	Nome	Nível	Oc...	Tipo de dados	C...
	SALES_REC	1	0		0
	HDR_DATA	3	0		0
	HDR_REC...	5	0	string	1
	HDR_STORE	5	0	number	2
	BODY_DATA	3	0		0
	BODY_STO...	5	0	string	30
	BODY_STO...	5	0	string	30
	BODY_STO...	5	0	string	30
	DETAIL_DATA	3	0		0
	DETAIL_ITEM	5	0	number	9
	DETAIL_DE...	5	0	string	30
	DETAIL_PR...	5	0	number	6
	DETAIL_QTY	5	0	number	5
	SUPPLIER...	5	4	number	0
	SUPPLIE...	10	0	string	2
	SUPPLIE...	10	0	string	8

As colunas Sales_Rec, Hdr_Data, Store_Data, Detail_Data e Supplier_Info são colunas no nível do grupo que identificam grupos de dados de nível inferior. As colunas no nível do grupo têm um comprimento zero porque não contêm dados. Nenhuma dessas colunas é porta de saída na definição de origem.

O grupo Supplier_Info contém as colunas Supplier_Code e Supplier_Name. O grupo Supplier_Info ocorre quatro vezes em cada registro Detail_Data.

Quando você cria uma transformação Normalizador de VSAM a partir da definição de origem COBOL, o Mapping Designer cria as portas de entrada/saída na transformação Normalizador com base nessa definição. A transformação Normalizador contém pelo menos uma porta de saída de chave gerada. Quando a origem COBOL contém colunas recorrentes, a transformação Normalizador tem uma porta de saída de ID de coluna gerada.

A figura a seguir mostra as portas da transformação Normalizador que o Mapping Designer cria a partir da definição de origem:



O grupo de colunas Supplier_Info ocorre quatro vezes em cada linha da origem COBOL.

Pode ser que a linha da origem COBOL contenha os seguintes dados:

```
Item1 ItemDesc 100 25 A Supplier1 B Supplier2 C Supplier3 D Supplier4
```

A transformação Normalizador retorna uma linha para cada ocorrência das colunas Supplier_Code e Supplier_Name. Cada linha de saída contém os mesmos valores para item, descrição, preço e quantidade.

O Normalizador retorna as seguintes linhas de dados de detalhe a partir da linha de origem COBOL:

```
Item1 ItemDesc 100 25 A Supplier1 1 1
Item1 ItemDesc 100 25 B Supplier2 1 2
Item1 ItemDesc 100 25 C Supplier3 1 3
Item1 ItemDesc 100 25 D Supplier4 1 4
```

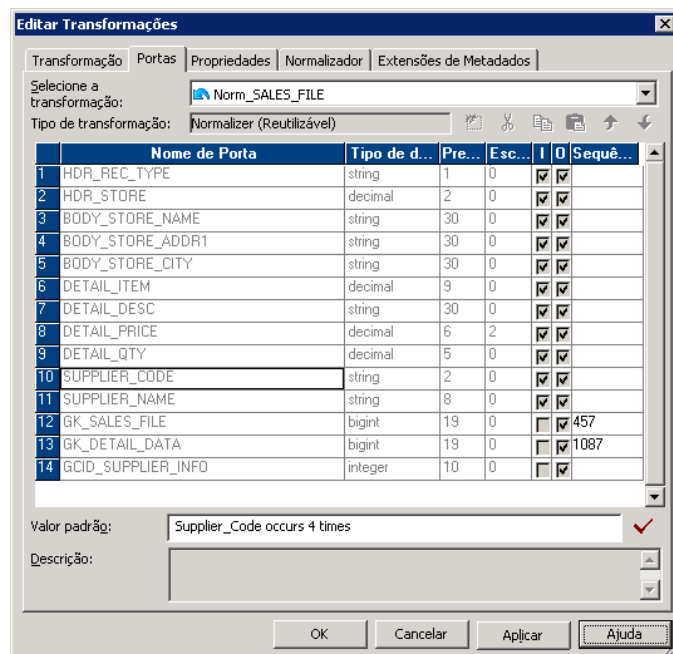
Cada linha de saída contém uma chave gerada e uma ID de coluna. O Serviço de Integração atualiza o valor da chave gerada quando processa uma nova linha da origem. Nas linhas de dados de detalhe, o valor da chave gerada é 1.

A ID da coluna define o número de vezes em que a coluna Supplier_Info ocorrerá. O Serviço de Integração atualiza a ID de coluna de cada ocorrência da coluna Supplier_Info. Os valores da ID de coluna são 1, 2, 3, 4 nas linhas de dados de detalhes.

Guia Portas Normalizadoras VSAM

A guia Portas Normalizadoras VSAM mostra as portas de entrada e saída da transformação. Tem uma porta de entrada e saída para cada coluna de origem COBOL. Ela tem uma porta de entrada/saída para uma coluna com várias ocorrências. A transformação não tem portas de entrada nem de saída para colunas no nível do grupo.

A figura a seguir mostra a guia Portas Normalizadoras VSAM:



Neste exemplo, Supplier_Code e Supplier_Name ocorrem quatro vezes na origem COBOL. A guia Portas mostra uma porta Supplier_Code port e outra Supplier_Name. Os valores iniciais da chave gerada são 457 e 1087.

Guia Normalizador VSAM

Quando você cria uma transformação Normalizador de VSAM, o Mapping Designer cria as colunas a partir de uma origem COBOL. A guia Normalizador mostra as mesmas informações do que a definição de origem COBOL. Não é possível editar as colunas em uma guia Normalizador de VSAM.

A tabela a seguir descreve os atributos na guia Normalizador VSAM:

Atributo	Descrição
POffs	Deslocamento físico. Localização do campo no arquivo. O primeiro byte no arquivo é zero.
Plen	Extensão física. Número de bytes no campo.
Nome da Coluna	Nome do campo de origem.
Nível	Fornece hierarquia do grupo de colunas. Quanto maior o número do nível, mais baixo os dados estão na hierarquia. As colunas no mesmo grupo ocorrem abaixo de uma coluna com um número de nível inferior. Quando cada coluna está no mesmo nível, a transformação não contém grupos de coluna.
Ocorrências	O número de instâncias de uma coluna ou grupo de colunas na linha de origem.
Tipo de Dados	O tipo de dados da transformação pode ser String, String Numérica ou Número.
Prec	Precisão. Comprimento da coluna.
Escala	Número de posições decimais para uma coluna numérica.
Imagem	Como os dados são armazenados ou exibidos na origem. Imagem 99V99 define um campo numérico com dois decimais implícitos. Imagem X(10) indica dez caracteres.
Uso	Formato de armazenamento de dados COBOL, como COMP, BINARY e COMP-3. Quando a Utilização é DISPLAY, a cláusula Picture define como os dados de origem estão formatados quando você os vê.
Tipo de Chave	Tipo de restrição de chave a ser aplicada ao campo para um arquivo VSAM. Escolha um dos seguintes tipos de chave: <ul style="list-style-type: none">- Não é uma Chave. O campo não é um índice em um arquivo VSAM.- Chave Primária. O campo é o índice primário no arquivo VSAM. O campo contém valores exclusivos.- Chave Alternativa. O campo é um índice secundário no arquivo VSAM. O campo contém valores exclusivos.- Chave Duplicada Primária. O campo é o índice primário no arquivo VSAM. O campo pode conter valores duplicados.- Chave Duplicada Alternativa. O campo é um índice secundário no arquivo VSAM. O campo pode conter valores duplicados.
Assinado (S)	Indica se os valores numéricos têm um sinal.
Sinal Direito (T)	Indica que o sinal (+ ou -) existe no último dígito do campo. Se essa opção não estiver ativada, o sinal aparecerá como o primeiro caractere do campo.
Sinal Incluído (I)	Indica se o sinal está incluído em qualquer valor que aparece no campo.
Vírgula Decimal Real (R)	Indica se os decimais são separados por uma vírgula (,) ou se esta é representada pelo caractere V em um campo numérico.

Atributo	Descrição
Redefinições	Indica que a coluna REDEFINE outra coluna.
Nome Comercial	Nome descritivo dado a uma coluna.

Etapas para Criar uma Transformação Normalizador de VSAM

Ao criar uma transformação Normalizador de VSAM, você arrasta uma origem COBOL para um mapeamento e o Mapping Designer cria as colunas de transformação a partir da origem. A transformação Normalizador é o qualificador de origem COBOL no mapeamento.

Quando você adiciona uma origem COBOL a um mapeamento, o Mapping Designer cria e configura uma transformação Normalizador. O Mapping Designer identifica registros aninhados e campos recorrentes na origem COBOL. Ele cria as colunas e portas na transformação Normalizador a partir das colunas da origem.

Para criar uma transformação de Normalizador VSAM:

1. No Mapping Designer, crie um novo mapeamento ou abra um existente.
2. Arraste uma definição de origem COBOL para o mapeamento.
O Designer adiciona uma transformação Normalizador e conecta-a à definição de origem COBOL. Se você não tiver habilitado a opção para criar um qualificador de origem por padrão, a caixa de diálogo Criar Transformação Normalizador aparecerá.
3. Nela, você pode escolher entre as seguintes opções:
 - **Origem VSAM.** Crie uma transformação a partir da definição de origem COBOL no mapeamento.
 - **Pipeline.** Crie uma transformação, mas não defina colunas a partir de uma origem COBOL. Defina as colunas manualmente na guia Normalizador. Você pode escolher esta opção quando quiser processar dados recorrentes de outra transformação no mapeamento.

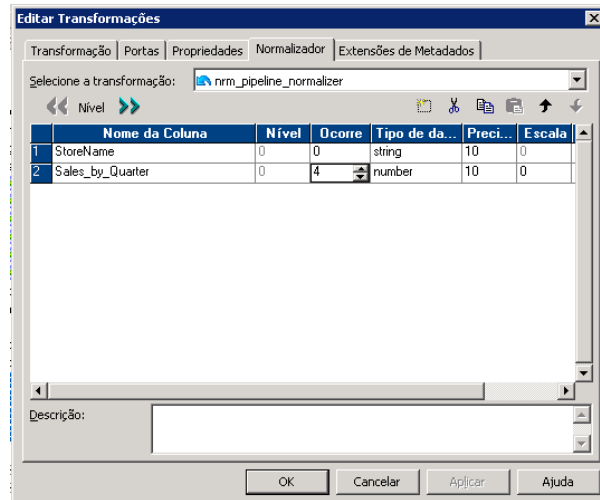
Para criar uma transformação Normalizador de VSAM, selecione a opção Transformação Normalizador de VSAM. A caixa de diálogo exibe o nome da definição da origem COBOL no mapeamento. Selecione a definição de origem COBOL e clique em OK.
4. Abra a transformação de Normalizador.
5. Selecione a guia Portas para ver as portas na transformação Normalizador.
Por padrão, o Designer cria as portas na definição da origem COBOL.
6. Clique na guia Normalizador para ver a organização da coluna de origem.
A guia Normalizador contém as mesmas informações que a guia Colunas da origem COBOL. No entanto, você não pode modificar atributos de coluna na transformação Normalizador. Para alterar atributos de coluna, altere o copybook COBOL, importe a origem COBOL e recrie a transformação Normalizador.
7. Selecione a guia Propriedades para definir o nível de rastreamento.
Você também pode configurar a transformação para redefinir os números da sequência chave gerados no início da próxima sessão.

Transformação de Normalizador de Pipeline

Ao criar uma transformação Normalizador no Transformation Developer, você cria uma transformação pipeline Normalizador por padrão. Ao criar uma transformação pipeline Normalizador, você define as

colunas com base nos dados recebidos pela transformação de outro tipo de transformação, como uma transformação Qualificador de origem. O Designer cria as portas de entrada e saída da transformação Normalizador a partir das colunas definidas.

A figura a seguir mostra as colunas da transformação Normalizador para uma transformação que receber quatro colunas de vendas em cada linha de origem relacional:



Cada linha de origem tem uma coluna StoreName e quatro instâncias de Sales_By_Quarter.

As linhas de origem podem conter os seguintes dados:

```
Dellmark 100 450 650 780
Tonys    666 333 444 555
```

A transformação pipeline Normalizador tem uma porta de entrada para cada instância de colunas recorrentes.

A figura a seguir mostra as portas criadas pelo Designer a partir das colunas na transformação Normalizador:

Nome	Tipo de dados	L...	Seq
StoreName_in	decimal	10	
Sales_by_Quarter_in1	decimal	10	
Sales_by_Quarter_in2	decimal	10	
Sales_by_Quarter_in3	decimal	10	
Sales_by_Quarter_in4	decimal	10	
StoreName	decimal	10	
Sales_by_Quarter	decimal	10	
GK_Sales_by_Quarter	bigint	19	1
GCID_Sales by Quarter	integer	10	

A transformação Normalizador retorna uma linha para cada instância de colunas recorrentes:

```
Dellmark 100 1 1
Dellmark 450 1 2
Dellmark 650 1 3
Dellmark 780 1 4
Tonys    666 2 1
Tonys    333 2 2
Tonys    444 2 3
Tonys    555 2 4
```

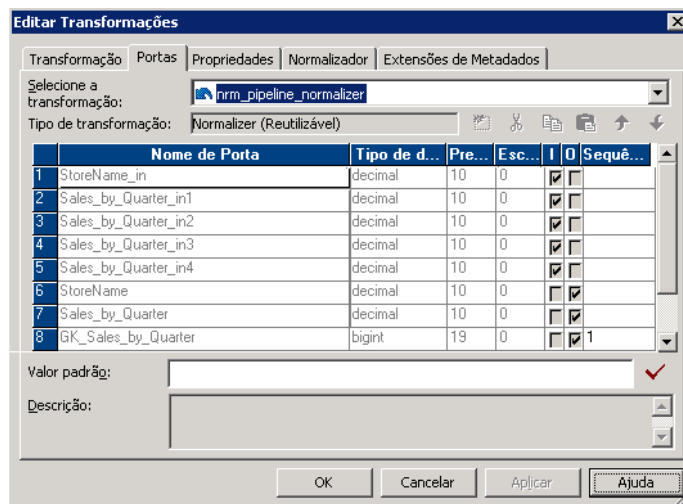
O Serviço de Integração aumenta o número de sequência de chave gerada cada vez que processa uma linha de origem. A chave gerada vincula vendas de cada trimestre à mesma loja. Neste exemplo, a chave gerada para a linha Dellmark é 1. A chave gerada para a loja Tonys é 2.

A transformação pipeline Normalizador tem uma porta de entrada para cada instância de colunas recorrentes. O valor GCID_Sales_by_Quarter é sempre 1, 2, 3 ou 4 neste exemplo.

Guia Portas do Normalizador do Pipeline

A guia Portas do Normalizador do pipeline exibe as portas de entrada e saída da transformação. Ela contém uma porta de entrada/saída para cada coluna de ocorrência única que você define na transformação. E uma porta para cada ocorrência de uma coluna recorrente. A transformação não tem portas de entrada nem de saída para colunas no nível do grupo.

A figura a seguir mostra a guia Portas da transformação Normalizador do pipeline:



O Designer cria uma porta de entrada para cada ocorrência de uma coluna recorrente.

Para alterar as portas em uma transformação Normalizador do pipeline, modifique as colunas na guia Normalizador. Quando você adiciona uma ocorrência de uma coluna, o Designer adiciona uma porta de entrada. O Designer cria portas para as colunas de nível mais baixo. Ele não cria portas para colunas no nível do grupo.

Guia Normalizador de Pipeline

Quando criar uma transformação Normalizador no pipeline, você define as colunas na guia Normalizador. O Designer cria portas de entrada e de saída com base nas colunas inseridas na guia Normalizador.

A tabela a seguir descreve os atributos da guia Normalizador do pipeline:

Atributo	Descrição
Nome da Coluna	Nome da coluna.
Nível	Identifica grupos de colunas. Colunas no mesmo grupo têm o mesmo número de nível. O padrão é zero. Quando cada coluna tem o mesmo nível, a transformação não contém grupos de colunas.
Occurs	O número de instâncias em uma coluna ou grupo de colunas na linha de origem.
Tipo de dados	O tipo de dados da coluna pode ser String, Nstring ou Number.

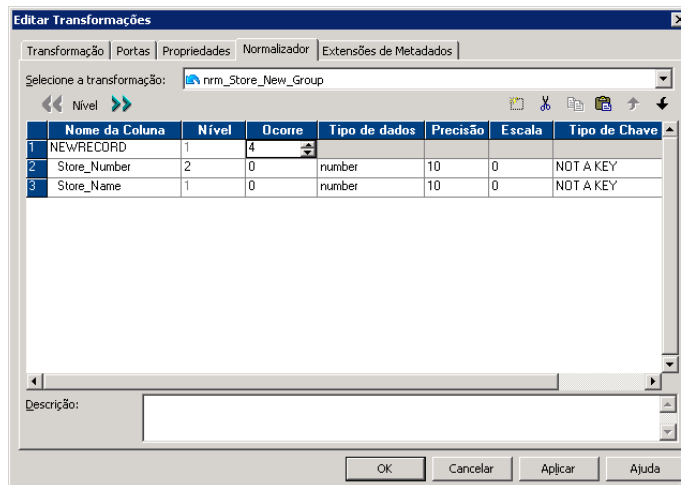
Atributo	Descrição
Prec	Precisão. Comprimento da coluna.
Escala	Número de dígitos decimais em um valor numérico.

Grupos de Colunas da Guia Normalizador

Quando uma linha de origem contém grupos de colunas repetitivas, você pode definir grupos de colunas na guia Normalizador. A transformação Normalizador retorna uma linha para cada ocorrência de grupo de colunas, não para cada ocorrência de coluna.

O número do nível na guia Normalizador identifica uma hierarquia de colunas. As colunas de nível de grupo identificam grupos de colunas. A coluna de nível de grupo tem um número de nível inferior ao das colunas do grupo. Colunas no mesmo grupo têm o mesmo número de nível e são exibidas sequencialmente abaixo da coluna de nível de grupo, na guia Normalizador.

A figura a seguir mostra um grupo de colunas de múltipla ocorrência na guia Normalizador:



Neste exemplo, a coluna NEWRECORD não contém nenhum dado. É uma coluna de grupo do Nível 1. O grupo ocorre quatro vezes em cada linha de origem. Store_Number e Store_Name são colunas de Nível 2. Eles pertencem ao grupo NEWRECORD.

Etapas para Criar uma Transformação de Normalizador de Pipeline

Quando criará uma transformação Normalizador no pipeline, você define as colunas na guia Normalizador.

Você pode criar uma transformação de Normalizador no Transformation Developer ou no Mapping Designer.

Para criar uma transformação Normalizador:

1. No Transformation Developer ou no Mapping Designer, clique em Transformação > Criar. Selecione a transformação Normalizador. Insira um nome para a transformação Normalizador.

A convenção de nomenclatura das transformações Normalizador é *NRM_Nome da Transformação*.

2. Clique em Criar e, em seguida, em Concluído.

3. Abra a transformação Normalizador e clique na guia Normalizador.
4. Clique em Adicionar para adicionar uma nova coluna.
O Designer criará uma nova coluna com atributos padrão. Você pode alterar o nome, o tipo de dados, a precisão e a escala.
5. Para criar uma coluna recorrente, insira o número de ocorrências na coluna Occurs.
6. Para criar um grupo de colunas recorrentes, insira pelo menos uma das colunas na guia Normalizador. Selecione a coluna. Clique em Nível.
O Designer adiciona uma coluna de nível de grupo NEWRECORD acima da coluna selecionada. NEWRECORD torna-se Nível 1. A coluna selecionada torna-se Nível 2. Você pode renomear a coluna NEWRECORD.
Por padrão, todas as colunas são do mesmo nível. O nível define colunas que são agrupadas juntas.
7. Você pode alterar o nível de outras colunas a fim de adicioná-las ao mesmo grupo. Selecione uma coluna e clique em Nível a fim de alterar o nível dela para o mesmo nível da coluna acima.
As colunas do mesmo grupo devem aparecer em sequência na guia Normalizador.
8. Mude a ocorrência no nível do grupo para fazer com que o grupo de colunas seja recorrente.
9. Clique em Aplicar para salvar as colunas e criar portas de entrada e saída.
O Designer cria portas de entrada e saída na transformação Normalizador. Além disso, ele cria as colunas chave geradas e uma ID de coluna para cada coluna ou grupo de colunas recorrente.
10. Selecione a guia Propriedades para alterar o nível de rastreamento ou redefinir os números da sequência chave gerados após a próxima sessão.

Usando uma Transformação de Normalizador em um Mapeamento

Quando uma transformação Normalizador receber mais de um tipo de dados de uma origem COBOL, você deve conectar as portas de saída do Normalizador a destinos diferentes, com base no tipo de dados em cada linha. O exemplo a seguir mostra como mapear a definição de origem COBOL Sales_File através de uma transformação Normalizador para vários destinos.

O registro de origem Sales_File contém informações sobre a loja ou informações sobre itens que a loja vende. O arquivo de vendas contém os dois tipos de registro.

O exemplo a seguir inclui dois arquivos de registro de vendas:

Record Type	Data
Store Record	H01Software Suppliers Incorporated 1111 Battery Street San Francisco
Item Record	D01123456789USB Line - 10 Feet 001495000020 01Supp1 02Supp2 03Supp3 04Supp4

A definição de origem COBOL e a transformação Normalizador têm colunas que representam campos nos dois tipos de registro. Você deve filtrar as linhas da loja das linhas de itens e passá-las a destinos diferentes.

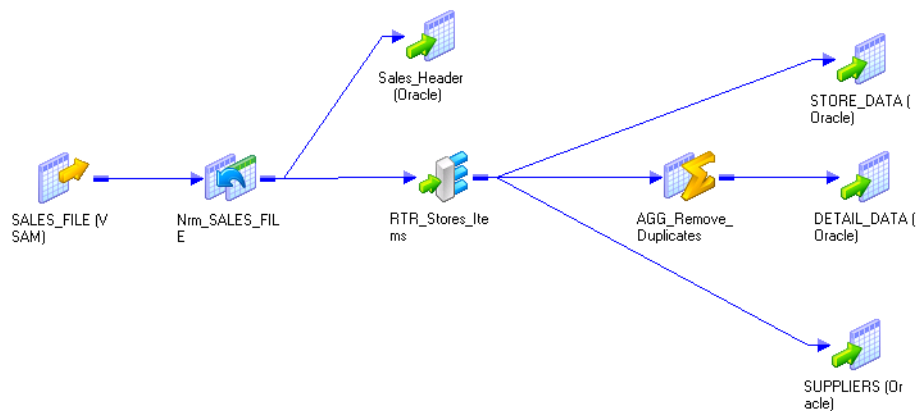
A figura a seguir mostra a origem COBOL Sales_File com seu correspondente Store_Data (que tem um valor de "S") e Detail_Data (que tem um valor de "D"):

T	Nome	Nível	Oc...	Tipo de dados	C...
SALES_REC		1	0		0
HDR_DATA		3	0		0
HDR_REC...		5	0	string	1
HDR_STORE		5	0	number	2
BODY_DATA		3	0		0
BODY_STO...		5	0	string	30
BODY_STO...		5	0	string	30
BODY_STO...		5	0	string	30
DETAIL_DATA		3	0		0
DETAIL_ITEM		5	0	number	9
DETAIL_DE...		5	0	string	30
DETAIL_PR...		5	0	number	6
DETAIL_QTY		5	0	number	5
SUPPLIER...		5	4	number	0
SUPPLIE...		10	0	string	2
SUPPLIE...		10	0	string	8

Hdr_Rec_Type define se o registro contém dados sobre a loja ou sobre a mercadoria. Quando o valor de Hdr_Rec_Type é "S", o registro contém Store_Data. Quando o Hdr_Rec_Type for "D," o registro conterá Detail_Data. Detail_Data sempre inclui quatro ocorrências de campos Supplier_Info.

Para filtrar dados, conecte as linhas de saída do Normalizador a uma transformação Roteador para rotear a loja e os dados do fornecedor para destinos diferentes. Você pode filtrar linhas na transformação Roteador com base no valor de Hdr_Rec_Type.

A figura a seguir mostra o mapeamento que roteia registros Sales_File para destinos diferentes:



O mapeamento filtra vários tipos de tipos de registro da origem COBOL para destinos relacionais. As colunas de ocorrência múltipla na origem são mapeadas para outra tabela relacional. Cada linha é indexada por ocorrência na linha de origem.

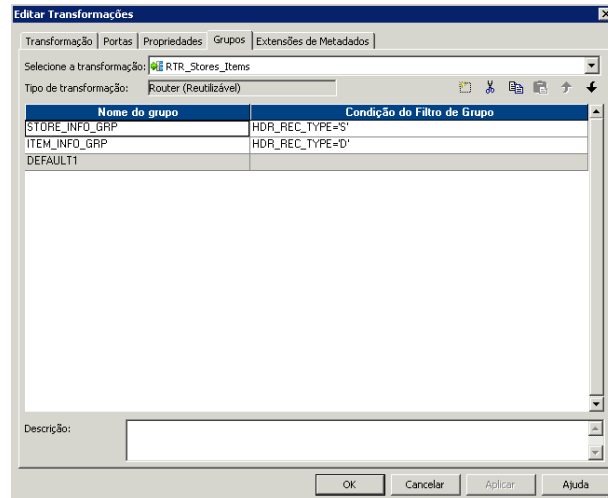
O mapeamento contém as seguintes transformações:

- **Transformação de Normalizador.** A transformação Normalizador retorna várias linhas quando a origem contém Detail_Data de ocorrência múltipla. Ela também processa outros tipos de registro da mesma origem.
- **Transformação de Roteador.** A transformação Roteador roteia dados para destinos com base no valor de Hdr_Rec_Type.
- **Transformação de Agregador.** A transformação de agregador remove linhas Detail_Data duplicadas a cada ocorrência do Supplier_Info.

O mapeamento tem a seguinte funcionalidade:

1. A transformação Normalizador passa o registro de cabeçalho e as colunas de número de cabeçalho da loja para o destino Sales_Header. Cada registro Sales_Header tem uma chave gerada que vincula a linha Sales_Header a uma linha de destino Store_Data ou Detail_Data. O Normalizador retorna Hdr_Data e Store_Data uma vez por linha.
2. A transformação Normalizador passa todas as colunas à transformação Roteador. Ela passa dados de Detail_Data quatro vezes por linha, uma vez para cada ocorrência das colunas Supplier_Info. As colunas Detail_Data contêm dados duplicados, exceto as colunas Supplier_Info.
3. A transformação Roteador passa a chave gerada e o nome, endereço e cidade da loja para Store_Data quando Hdr_Rec_Type é "S". A chave gerada vincula linhas Store_Data a linhas Sales_Header.
A transformação roteador contém um grupo definido pelo usuário para os dados da loja e outro para os itens da mercadoria.
4. A transformação Roteador passa o item, a quantidade, o preço e a descrição do item e as chaves geradas por Detail_Data para uma transformação Agregador quando Hdr_Rec_Type é "D".
5. A transformação Roteador passa o ID da coluna e o código e nome do fornecedor ao destino Fornecedores quando Hdr_Rec_Type é "D". Ela passa a chave gerada que vincula a linha Fornecedores à linha Detail_Data.
6. A transformação Agregador remove as colunas Detail_Data duplicadas. O Agregador passa uma instância do item, descrição, preço, quantidade e chave gerada para Detail_Data. A chave gerada por Detail_Data vincula linhas Detail_Data a linhas de Fornecedores. Detail_Data também tem uma chave que vincula as linhas Detail_Data às linhas Sales_Header.

A figura a seguir mostra os grupos definidos pelo usuário e as condições de filtro na transformação Roteador:



Observe que a transformação Roteador passa dados da loja ou de itens com base no tipo de registro.

Gerando valores-chave

A transformação Normalizador criará uma chave gerada quando a origem COBOL contém um grupo de colunas de múltipla ocorrência. Você pode transferir um grupo de colunas de múltipla ocorrência para um destino diferente das outras colunas na linha. Você pode criar uma relação de chave primária externa entre os destinos, com a chave gerada.

A figura a seguir mostra uma definição da origem COBOL que contém um grupo de colunas de múltipla ocorrência:

Nome	Nível	Oc...	Tipo de dados	Comprim
DETAIL_RECORD	1	0		0
DETAIL_ITEM	3	0	number	9
DETAIL_DESC	3	0	string	30
DETAIL_PRICE	3	0	number	6
DETAIL_QTY	3	0	number	5
DETAIL_SUPPLIERS	3	4	number	0
SUPPLIER_CODE	10	0	string	2
SUPPLIER_NAME	10	0	string	8

Neste exemplo, o grupo de colunas Detail_Suppliers ocorre quatro vezes no Detail_Record.

A transformação do Normalizador gera uma chave GK_Detail_Sales para cada linha de origem. A chave GK_Detail_Sales representa uma linha da origem Detail_Record.

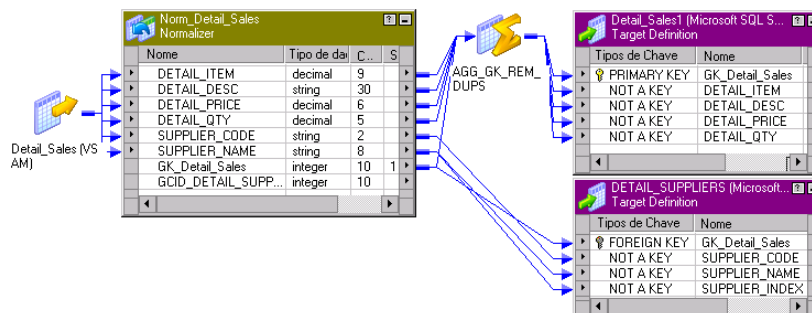
A seguinte figura mostra os relacionamentos de chave externa primária entre os destinos:

DETAIL_SUPPLIERS (Microsoft SQL Server)			
T	Nome	Tipo de dados	C...
	GK_Detail_Sales	int	10
	SUPPLIER_CO...	varchar	2
	SUPPLIER_NA...	varchar	8

Details_Sales (Microsoft SQL Server)			
T	Nome	Tipo de dados	C...
	GK_Detail_Sales	int	10
	DETAIL_ITEM	numeric	9
	DETAIL_DESC	varchar	30
	DETAIL_PRICE	numeric	6
	DETAIL_QTY	numeric	5

Linhas de múltipla ocorrência Detail_Supplier têm uma chave externa que as vincula à mesma linha Detail_Sales. O destino Detail_Sales tem uma relação um-para-vários com o destino Detail_Suppliers.

A figura a seguir mostra a chave gerada GK_Detail_Sales conectada com chaves primárias e externas no destino:



Transfira GK_Detail_Sales para a chave primária Detail_Sales e a chave externa Detail_Suppliers.

Vincule as colunas de saída do Normalizador com os objetos a seguir:

- **Detail_Sales_Target.** Transfira as colunas Detail_Item, Detail_Desc, Detail_Price e Detail_Qty para um destino Detail_Sales. Transfira a chave GK_Detail_Sales à chave primária Detail_Sales.
- **Transformação Agregador.** Transfira cada linha Detail_Sales através de uma transformação Agregador para remover linhas duplicadas. O Normalizador retorna colunas Detail_Sales duplicadas para cada ocorrência de Detail_Suppliers.

- **Detail_Suppliers.** Transfira cada instância das colunas Detail_Suppliers para o destino Detail_Suppliers. Transfira a chave GK_Detail_Sales à chave estrangeira Detail_Suppliers. Cada instância das colunas Detail_Suppliers tem uma chave externa que relaciona a linha Detail_Suppliers com a linha Detail_Sales.

Solução de Problemas de Transformações Normalizador

Não consigo editar as portas na minha transformação Normalizador quando uso uma origem relacional.

Ao criar portas manualmente, adicione-as à guia Normalizador na transformação, e não à guia Portas.

A importação de um arquivo COBOL falhou com numberrrors. O que devo fazer?

Verifique se o programa COBOL segue o padrão COBOL, incluindo espaços, guias e caracteres de fim de linha. Os cabeçalhos de arquivos COBOL devem ser semelhantes ao texto a seguir:

```
identification division.
                    program-id. mead.
environment division.
                    select file-one assign to "fname".
data division.
file section.
fd FILE-ONE.
```

O Designer não lê caracteres ocultos no programa COBOL. Utilize um editor somente de texto para fazer alterações no arquivo COBOL. Não use o Word nem o Wordpad. Remova espaços extras.

Foi concluída uma sessão de leitura de dados binários, mas as informações na tabela de destino estão erradas.

Edite a sessão no Workflow Manager e verifique se o formato do arquivo de origem está definido corretamente. O formato do arquivo poderia ser EBCDIC ou ASCII. O número de bytes a serem ignorados entre os registros deve estar configurado como 0.

Tenho uma descrição de campo COBOL que usa um tipo não IBM COMP. Como devo importar a origem?

Na definição da origem, apague a opção IBM COMP.

No meu mapeamento, uso uma transformação Expressão e uma transformação Pesquisa para modificar duas portas de saída da transformação Normalizador. O mapeamento concatena-as em uma única transformação. Todas as portas estão no mesmo nível. Quando verifico os dados carregados no destino, eles estão incorretos. Por que isso acontece?

Você só pode concatenar portas de nível um. Remova a concatenação.

CAPÍTULO 21

Transformação Classificação

Este capítulo inclui os seguintes tópicos:

- [Visão Geral da Transformação Classificação, 375](#)
- [Portas em uma Transformação Classificação, 377](#)
- [Definição de Grupos, 378](#)
- [Criando uma transformação Classificação, 378](#)

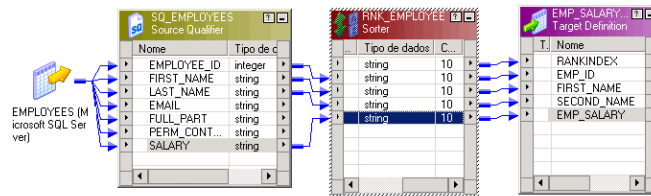
Visão Geral da Transformação Classificação

Você pode selecionar apenas as classificações de dados mais altas ou mais baixas com uma transformação de Classificação. A transformação de Classificação é uma transformação ativa. Use uma transformação Classificação para retornar o maior ou menor valor numérico em uma porta ou grupo. Você pode ainda usar uma transformação Classificação para retornar strings no início ou fim da ordem de classificação de uma sessão. Durante a sessão, o Serviço de Integração armazena dados de entrada até que possa calcular a classificação.

A transformação Classificação difere das funções de transformação MAX e MIN, pois permite selecionar um grupo de valores superiores e inferiores, não apenas um valor. Por exemplo, use Classificação para selecionar os dez maiores vendedores em determinado território. Ou, para gerar um relatório financeiro, você deve usar também uma transformação Classificação para identificar os três departamentos com gastos mais baixos com salários e despesas gerais. Enquanto a linguagem SQL fornece muitas funções para controlar grupos de dados, não é possível identificar segmentos superiores ou inferiores de um conjunto de linhas usando funções SQL padrão.

Você conecta à transformação todas as portas que representam o mesmo conjunto de linhas. Apenas as linhas dentro dessa classificação, com base em alguma medida definida durante a configuração, passarão através da transformação Classificação. Você também pode criar expressões para transformar dados ou executar cálculos.

A figura a seguir mostra um mapeamento que passa dados de funcionários de uma tabela de recursos humanos através de uma transformação Classificação. A transformação Classificação passa apenas as linhas dos 10 funcionários com salário mais alto para a transformação seguinte.



Como transformação ativa, a transformação Classificação pode alterar o número de linhas que passam por ela. Você pode passar 100 linhas à transformação Classificação, mas só é possível selecionar as 10 linhas superiores, passadas da transformação Classificação a outra transformação.

Você pode conectar portas de apenas uma transformação. Você também pode criar variáveis locais e gravar expressões não agregadas.

Classificando Valores de String

Quando o Serviço de Integração é executado no modo de movimentação de dados ASCII, classifica dados da sessão usando uma ordem de classificação binária.

Quando o Serviço de Integração é executado no modo de movimentação de dados Unicode, usa a ordem de classificação configurada para a sessão. Você seleciona a ordem de classificação da sessão nas propriedades da sessão. As propriedades da sessão listam todas as ordens de classificação disponíveis na página de código usada pelo Serviço de Integração.

Por exemplo, você tem uma transformação Classificação configurada para retornar os três valores mais altos de uma porta de string. Quando configura o fluxo de trabalho, você seleciona o Serviço de Integração no qual o fluxo de trabalho deve ser executado. As propriedades da sessão mostram todas as ordens de classificação associadas à página de código do Serviço de Integração selecionado, como Francês, Alemão e Binário. Se você configurar a sessão para usar uma ordem de classificação binária, o Serviço de Integração calculará o valor binário de cada string e retornará as três linhas com maior valor binário na string.

Caches de Classificação

Durante uma sessão, o Serviço de Integração compara uma linha de entrada com linhas do cache de dados. Se a linha de entrada for de um nível superior que uma linha em cache, o Serviço de Integração substituirá a linha armazenada em cache pela linha de entrada. Se você configurar a transformação Classificação para classificar vários grupos, o Serviço de Integração ordenará incrementalmente cada grupo que encontrar.

O Serviço de Integração armazena informações de grupo em um cache de índice e armazena dados em linha no cache de dados. Se você criar várias partições em um pipeline, o Serviço de Integração criará caches separados para cada partição.

Propriedades da Transformação Classificação

Quando cria uma transformação Classificação, você pode configurar as propriedades a seguir:

- Insira um diretório de cache.
- Selecione a ordem superior ou inferior.

- Selecione a porta de entrada/saída que contém valores usados para determinar a classificação. Você pode selecionar somente uma porta para definir uma classificação.
- Selecione o número de linhas que se encaixam em uma classificação.
- Defina grupos para ordens, como os 10 produtos mais baratos de cada fabricante.

Portas em uma Transformação Classificação

A transformação Classificação inclui portas de entrada ou de entrada e saída conectadas a outra transformação no mapeamento. Inclui também portas variáveis e uma porta de classificação. Use a porta de classificação para especificar a coluna que você quer classificar.

A tabela a seguir descreve as portas em uma transformação Classificação:

Portas	Número obrigatório	Descrição
I	Mínimo de um	Porta de entrada. Crie uma porta de entrada para receber dados de outra transformação.
O	Mínimo de um	Porta de saída. Crie uma porta de saída para cada porta que você queira vincular a outra transformação. Você pode designar portas de entrada como portas de saída.
V	Não obrigatório	Porta variável. Pode ser usado para armazenar valores ou cálculos para usar em expressões. Portas variáveis não podem ser portas de entrada nem de saída. Elas passam dados apenas no interior da transformação.
R	Apenas um	Porta de classificação. Use para designar a coluna para a qual você quer classificar valores. Você pode designar apenas uma porta de classificação em uma transformação Classificação. A porta de classificação é uma porta de entrada e saída. Você deve vincular a porta de classificação a outra transformação.

Índice de Classificação

O Designer criará uma porta RANKINDEX para cada transformação Classificação. O Serviço de Integração utiliza a porta Índice de Classificação para armazenar a posição na ordem de cada linha em um grupo.

Por exemplo, se você criar uma transformação de Classificação que classifique os cinco principais vendedores de cada trimestre, o índice de classificação numerará os vendedores de 1 a 5:

RANKINDEX	SALES_PERSON	SALES
1	Sam	10,000
2	Mary	9,000
3	Alice	8,000
4	Ron	7,000
5	Alex	6,000

RANKINDEX é uma porta somente de saída. Você pode passar o índice de classificação para outra transformação no mapeamento ou diretamente para um destino.

Definição de Grupos

Assim como a transformação Agregador, a transformação Classificação permite agrupar as informações. Por exemplo, se você deseja selecionar os 10 itens mais caros por fabricante, deve definir primeiro um grupo para cada fabricante. Quando você configura a transformação Classificação, pode configurar uma de suas portas de entrada/saída como um grupo por porta. Para cada valor exclusivo na porta do grupo, a transformação cria um grupo de linhas que se encaixem na definição da classificação (primeira ou última e um número específico em cada classificação).

Portanto, a transformação Classificação altera o número de linhas de duas maneiras diferentes. Filtrando todas, menos as linhas que se encaixam na primeira ou na última classificação, você reduz o número de linhas transferidas pela transformação. Definindo grupos, você cria um conjunto de linhas classificadas para cada grupo.

Por exemplo, você pode criar uma transformação Classificação para identificar os 50 funcionários com os salários mais altos na empresa. Neste caso, você identificaria a coluna SALARY como a porta de entrada/saída usada para medir as classificações, e configuraria a transformação para filtrar todas as linhas exceto as primeiras 50.

Depois que a transformação Classificação identifica todas as linhas que pertencem à primeira ou à última classificação, ela atribui os valores de índice. No caso dos 50 primeiros funcionários, medidos por salário, o funcionário com o salário mais alto recebe um índice de classificação de 1. O próximo funcionário de salário mais alto recebe um índice de 2 e assim por diante. Ao medir uma classificação inferior, como os 10 produtos de preço mais baixo no inventário, a transformação Classificação atribui um índice de classificação desde o mais baixo até o mais alto. Portanto, o item menos caro receberia um índice de classificação de 1.

Se dois valores de classificação corresponderem, eles recebem o mesmo valor no índice de classificação e a transformação ignora o próximo valor. Por exemplo, se você deseja ver as cinco principais lojas de varejo no país e duas lojas têm as mesmas vendas, os dados retornados podem ser semelhantes aos mostrados abaixo:

RANKINDEX	SALES	STORE
1	10000	Orange
1	10000	Brea
3	90000	Los Angeles
4	80000	Ventura

Criando uma transformação Classificação

Você pode adicionar uma transformação Classificação em qualquer lugar do mapeamento depois do qualificador de origem.

Para criar uma transformação Classificação:

1. No Mapping Designer, clique em Transformação > Criar. Selecione a transformação de Classificação. Insira um nome para a Classificação. A convenção de nomenclatura das transformações de Classificação é `RNK_TransformationName`.
Insira uma descrição para a transformação. Esta descrição aparece no Repository Manager.
2. Clique em Criar e em Concluído.
O Designer cria a transformação Classificação.
3. Vincule as colunas de uma transformação de entrada à transformação de Classificação.
4. Clique na guia Portas e selecione a opção Classificação (R) para a porta da classificação.
Se deseja criar grupos para as linhas classificadas, selecione Agrupar Por para a porta que define o grupo.
5. Clique na guia Propriedades e selecione se deseja a primeira ou a última classificação.
6. Para a opção Número de Classificações, insira o número de linhas que deseja selecionar para a classificação.
7. Altere as outras propriedades da transformação de Classificação, se necessário.

A tabela a seguir descreve as propriedades da transformação Classificação:

Configuração	Descrição
Diretório de Cache	Diretório local em que o Serviço de Integração cria o índice e os arquivos do cache de dados. Por padrão, o Serviço de Integração usa o diretório inserido no Workflow Manager para a variável de processo <code>\$PMCCacheDir</code> . Se você inserir um novo diretório, certifique-se de que ele exista e contenha espaço em disco suficiente para os arquivos de cache.
Início/Fim	Especifica se você deseja a primeira ou a última classificação para uma coluna.
Número de Classificações	Número de linhas que você deseja classificar.
Comparação de String com Diferenciação de Maiúscula/Minúscula	Ao executar no modo Unicode, o Serviço de Integração ordena as strings com base na ordem de classificação selecionada para a sessão. Se essa ordem fizer diferenciação entre maiúsculas e minúsculas, selecione essa opção para permitir comparação de string com diferenciação de maiúscula/minúscula e desmarque esta opção para que o Serviço de Integração ignore essa diferenciação. Se a ordem de classificação não fizer a diferenciação, o Serviço de Integração ignorará essa configuração. Por padrão, essa opção é selecionada.
Nível de Rastreamento	Determina a quantidade de informações que o Serviço de Integração grava no log de sessão sobre os dados transferidos por essa transformação em uma sessão.
Classificar Tamanho do Cache de Dados	Tamanho do cache de dados para a transformação. O padrão é 2.000.000 bytes. Se o tamanho total do cache de sessão configurado for 2 GB (2.147.483.648 bytes) ou mais, você deverá executar a sessão em um Serviço de Integração de 64 bits. É possível usar um valor numérico para o cache, usar um valor de cache proveniente de um arquivo de parâmetros ou configurar o Serviço de Integração para definir o tamanho do cache usando a configuração Automático. Se você configurar o Serviço de Integração para determinar o tamanho do cache, também poderá especificar a quantidade máxima de memória alocada ao cache pelo Serviço de Integração.

Configuração	Descrição
Classificar Tamanho do Cache de Índice	Tamanho do cache de índice para a transformação. O padrão é 1.000.000 bytes. Se o tamanho total do cache de sessão configurado for 2 GB (2.147.483.648 bytes) ou mais, você deverá executar a sessão em um Serviço de Integração de 64 bits. É possível usar um valor numérico para o cache, usar um valor de cache proveniente de um arquivo de parâmetros ou configurar o Serviço de Integração para definir o tamanho do cache usando a configuração Automático. Se você configurar o Serviço de Integração para determinar o tamanho do cache, também poderá especificar a quantidade máxima de memória alocada ao cache pelo Serviço de Integração.
Escopo da Transformação	Especifica como o Serviço de Integração aplica a lógica de transformação aos dados de entrada: <ul style="list-style-type: none"> - Transação. Aplica a lógica de transformação a todas as linhas de uma transação. Escolha Transação quando uma linha de dados depender de todas as linhas na mesma transação, mas não de linhas em outras transações. - Todas as Entradas. Aplica a lógica de transformação em todos os dados de entrada. Quando você escolhe Todas as Entradas, o PowerCenter descarta os limites de transação de entrada. Escolha Todas as entradas quando uma linha de dados depender de todas as linhas na origem.

8. Clique em OK.

CAPÍTULO 22

Transformação Roteador

Este capítulo inclui os seguintes tópicos:

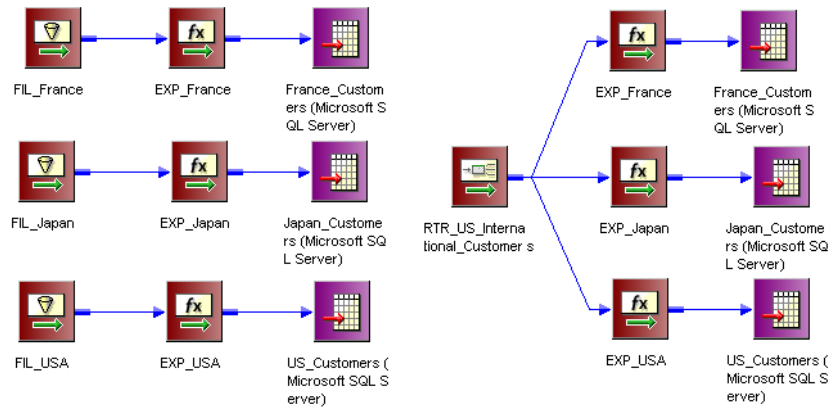
- [Visão Geral da Transformação Roteador, 381](#)
- [Trabalhando com Grupos, 383](#)
- [Trabalhando com Portas, 385](#)
- [Conexão das Transformações Roteador em um Mapeamento, 386](#)
- [Criando uma Transformação Roteador, 386](#)

Visão Geral da Transformação Roteador

A transformação Roteador é semelhante à transformação Filtro, porque as duas permitem que você utilize uma condição para testar dados. A transformação Filtro testa dados para uma condição e descarta as linhas de dados que não atenderem à condição. No entanto, uma transformação de Roteador testa dados para uma ou mais condições e permite rotear para um grupo de saída padrão as linhas de dados que não atenderem a nenhuma das condições. A transformação de Roteador é uma transformação ativa.

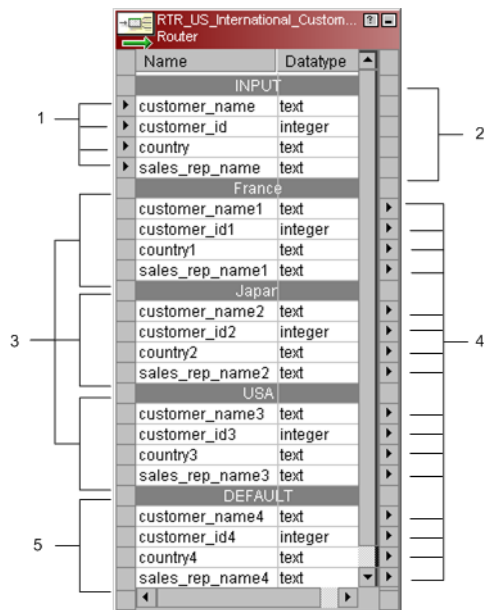
Se você precisar testar os mesmos dados de entrada com base em várias condições, utilize uma transformação Roteador em um mapeamento, em vez de criar várias transformações Filtro para executar a mesma tarefa. A transformação Roteador é mais eficiente. Por exemplo, para testar dados com base em três condições, você precisa somente de uma transformação Roteador, em vez de três transformações Filtro para realizar essa tarefa. Da mesma forma, quando você usa uma transformação Roteador em um mapeamento, o Serviço de Integração processa os dados de entrada uma única vez. Quando você usa várias transformações Filtro em um mapeamento, o Serviço de Integração processa os dados de entrada para cada transformação.

A figura a seguir mostra dois mapeamentos para executar a mesma tarefa. O primeiro utiliza três transformações Filtro; o segundo produz o mesmo resultado com uma única transformação Roteador.



A transformação Roteador é formada por grupos de entrada e saída, portas de entrada e saída, condições de filtro de grupo e propriedades que você configura no Designer.

A figura a seguir mostra um exemplo da transformação Roteador:



1. Portas de entrada.
2. Grupo de entrada.
3. Grupos de saída definidos pelo usuário.
4. Portas de saída
5. Grupo de saída padrão

Trabalhando com Grupos

Uma transformação Roteador tem os seguintes tipos de grupos:

- Entrada
- Saída

Grupo de Entrada

O Designer copia as informações de propriedade das portas de entrada do grupo de entrada, a fim de criar um conjunto de portas de saída para cada grupo de saída.

Grupos de Saída

Há dois tipos de grupos de saída:

- Grupos definidos pelo usuário
- Grupo padrão

Você não pode modificar ou excluir as portas de saída ou suas propriedades.

Grupos Definidos pelo Usuário

Você cria um grupo definido pelo usuário para testar uma condição com base em dados de entrada. Um grupo definido pelo usuário é formado por portas de saída e por uma condição de filtro de grupo. Você pode criar e editar grupos definidos pelo usuário na guia Grupos com o Designer. Crie um grupo definido pelo usuário para cada condição que deseje especificar.

O Serviço de Integração utiliza a condição para avaliar cada linha de dados de entrada. Antes de processar o grupo padrão, ele testa as condições de cada grupo definido pelo usuário. O Serviço de Integração determina a ordem de avaliação de cada condição, com base na ordem dos grupos de saída conectados. O Serviço de Integração processa grupos definidos pelo usuário que estão conectados a uma transformação ou a um destino em um mapeamento. O Serviço de Integração só processará grupos definidos pelo usuário que *não* estejam conectados em um mapeamento, se o grupo padrão estiver conectado a uma transformação ou a um destino.

Se uma linha atender a mais de uma condição de filtro de grupo, o Serviço de Integração passará essa linha várias vezes.

O Grupo Padrão

O Designer criará o grupo padrão depois que você criar um novo grupo definido pelo usuário. O Designer não permite editar nem excluir o grupo padrão. Esse grupo não tem uma condição de filtro de grupo associada. Se *todas* as condições equivalerem a FALSE, o Serviço de Integração passará a linha ao grupo padrão. Se você quiser que o Serviço de Integração descarte todas as linhas no grupo padrão, não o conecte a uma transformação nem a um destino no mapeamento.

O Designer excluirá o grupo padrão quando você excluir da lista o último grupo definido pelo usuário.

Usando Condições do Filtro de Grupo

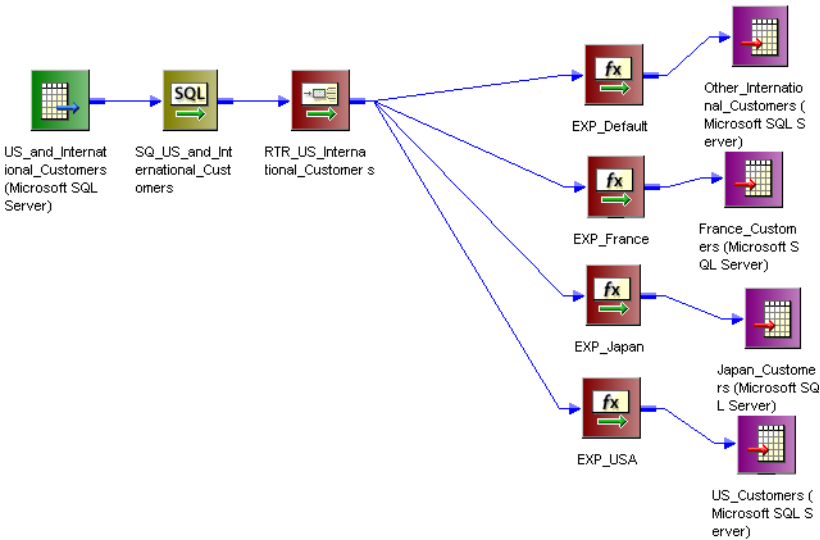
Você pode testar dados com base em uma ou mais condições do filtro de grupo. Você cria condições do filtro de grupo na guia Grupos usando o Editor de Expressão. Você pode inserir qualquer expressão que retorne um único valor. Você pode ainda especificar uma constante para a condição. Uma condição do filtro

de grupo retornará TRUE ou FALSE para cada linha que passa pela transformação, se a linha satisfizer a condição especificada. Zero (0) equivale a FALSE e todos os valores diferentes de zero equivalem a TRUE. O Serviço de Integração passa as linhas de dados equivalentes a TRUE para cada transformação ou destino associado a cada grupo definido pelo usuário.

Por exemplo, você tem clientes de nove países e quer realizar diversos cálculos sobre os dados de somente três países. Talvez você prefira usar uma transformação Roteador no mapeamento para filtrar os dados para três transformações Expressão diferentes.

Não há condição do filtro de grupo associada ao grupo padrão. Mas você pode criar uma transformação Expressão para realizar cálculos com base nos dados dos outros seis países.

A figura a seguir mostra um mapeamento com a transformação Roteador, que filtra dados com base em várias condições:



Já que você quer realizar vários cálculos com base nos dados de três países diferentes, crie três grupos definidos pelo usuário e especifique três condições do filtro de grupo na guia Grupos.

A tabela a seguir mostra as condições do filtro que filtram dados de clientes:

Nome do grupo	Condição do Filtro de Grupo
França	customer_name='France'
Japão	customer_name='Japan'
EUA	customer_name='USA'

Na sessão, o Serviço de Integração passa as linhas de dados equivalentes a TRUE para cada transformação ou destino associado a cada grupo definido pelo usuário, como Japão, França e EUA. O Serviço de Integração passará a linha ao grupo padrão se todas as condições equivalerem a FALSE. Se isso acontecer, o Serviço de Integração passará os dados dos outros seis países à transformação ou destino associado ao grupo padrão. Se você quiser que o Serviço de Integração descarte todas as linhas no grupo padrão, não o conecte a uma transformação nem a um destino no mapeamento.

A transformação Roteador passa dados através de cada grupo que atenda à condição. Assim, se os dados atenderem a três condições do grupo de saída, a transformação Roteador passará dados através de três grupos de saída.

Por exemplo, você configura as seguintes condições de grupo em uma transformação Roteador:

Nome do grupo	Condição do Filtro de Grupo
Grupo de Saída 1	employee_salary > 1000
Grupo de Saída 2	employee_salary > 2000

Quando a transformação Roteador processa dados de uma linha de entrada com employee_salary=3000, roteia os dados através dos grupos de saída 1 e 2.

Adição de Grupos

A adição de um grupo é semelhante à adição de uma porta em outras transformações. O Designer copia as informações de propriedade das portas de entrada para as portas de saída.

Para adicionar um grupo a uma transformação Roteador:

1. Clique na guia Grupos.
2. Clique no botão Adicionar.
3. Insira um nome para o novo grupo na seção Nome do Grupo.
4. Clique no campo Condições de Filtro de Grupo e abra o Editor de Expressão.
5. Insira a condição do filtro de grupo.
6. Clique em Validar para verificar a sintaxe da condição.
7. Clique em OK.

Trabalhando com Portas

Uma transformação Roteador tem portas de entrada e de saída. As portas de entrada estão no grupo de entrada; as de saída estão nos grupos de saída. Você pode criar portas de entrada copiando-as de outra transformação ou criando-as manualmente na guia Portas.

O Designer cria portas de entrada copiando as seguintes propriedades das portas de entrada:

- Nome da porta
- Tipo de dados
- Precisão
- Escala
- Valor padrão

Quando você faz alterações nas portas de entrada, o Designer atualiza as portas de entrada para refletir essas alterações. Não é possível editar ou excluir portas de saída. As portas de entrada são exibidas no modo de exibição Normal da transformação Roteador.

O Designer cria nomes de porta de saída com base nos nomes de portas de entrada. Para cada porta de entrada, o Designer criará uma porta de saída correspondente em cada grupo de saída.

Conexão das Transformações Roteador em um Mapeamento

Quando você conectar as transformações a uma transformação Roteador em um mapeamento, considere as regras a seguir:

- Você pode conectar um grupo a uma transformação ou destino.
- Você pode conectar uma porta de saída em um grupo a múltiplas transformações ou destinos.
- Você pode conectar múltiplas portas de saída em um grupo a múltiplas transformações ou destinos.
- Você não pode conectar mais de um grupo a um destino ou a uma única transformação de grupo de entrada.
- Você pode conectar mais de um grupo a uma transformação com vários grupos de entrada, exceto nas transformações Separador, quando conectar cada grupo de saída a um grupo de entrada diferente.

Criando uma Transformação Roteador

Para adicionar uma transformação de Roteador a um mapeamento, conclua as seguintes etapas:

1. No Mapping Designer, abra uma transformação Qualificador de origem.
2. Clique em Transformação > Criar.
Selecione a Transformação Roteador e insira o nome da nova transformação. A convenção de nomenclatura da Transformação Roteador é `RTR_TransformationName`. Clique em Criar e, em seguida, em Concluído.
3. Selecione e arraste todas as portas de uma transformação para adicioná-las à Transformação Roteador; também é possível criar portas de entrada manualmente na guia Portas.
4. Clique duas vezes na barra de títulos da Transformação Roteador para editar as propriedades de transformação.
5. Clique na guia Transformação e configure as propriedades de transformação.
6. Clique na guia Propriedades e configure os níveis de rastreamento.
7. Clique na guia Grupos e, em seguida, clique no botão Adicionar para criar um grupo definido pelo usuário.
O Designer cria o grupo padrão quando você cria o primeiro grupo definido pelo usuário.
8. Clique no campo Condições de Filtro de Grupo para abrir o Editor de Expressão.
9. Insira uma condição do filtro de grupo.
10. Clique em Validar para verificar a sintaxe das condições inseridas.
11. Clique em OK.
12. Conecte as porta de saída de grupo às transformações ou destinos.

CAPÍTULO 23

Transformação de Gerador de Sequência

Este capítulo inclui os seguintes tópicos:

- [Visão Geral da Transformação de Gerador de Sequência, 387](#)
- [Portas do Gerador de Sequência, 388](#)
- [Propriedades da Transformação Gerador de Sequência, 394](#)
- [Objeto de Dados de Sequência, 400](#)
- [Criando uma Transformação de Gerador de Sequência, 402](#)
- [Criando uma Transformação de Gerador de Sequência, 403](#)
- [Perguntas frequentes, 404](#)
- [Transformação de Gerador de Sequência em um ambiente não nativo, 405](#)

Visão Geral da Transformação de Gerador de Sequência

A transformação de Gerador de Sequência é uma transformação passiva que gera valores numéricos. Use a transformação Gerador de Sequência para criar valores de chave primária exclusivos, substituir chaves primárias ausentes ou percorrer uma faixa sequencial de números.

A transformação de Gerador de Sequência é uma transformação conectada. Ela contém duas portas de saída que você pode conectar a uma ou mais transformações. O Serviço de Integração gera um bloco de números em sequência, toda vez que um bloco de linhas insere uma transformação conectada. Se você conectar CURRVAL, o Serviço de Integração processará uma linha em cada bloco. Quando NEXTVAL está conectado à porta de entrada da transformação, o Serviço de Integração gera uma sequência numérica. Quando CURRVAL está conectado à porta de entrada de outra transformação, o Serviço de Integração gera o valor NEXTVAL, além do valor Aumentar em.

O Gerador de Sequência contém portas de passagem e uma porta de saída. Você conecta a porta NEXTVAL às portas de entrada de outras transformações. O Serviço de Integração incrementa a sequência quando o mapeamento é executado.

Você pode criar uma transformação de Gerador de Sequência para ser usada em um único mapeamento ou uma transformação de Gerador de Sequência reutilizável para ser usada em vários mapeamentos. Uma transformação Gerador de Sequência reutilizável mantém a integridade da sequência em cada mapeamento que usa uma instância da transformação Gerador de Sequência.

Você pode criar uma transformação de Gerador de Sequência com base em uma nova sequência ou em um objeto de dados de sequência. Um objeto de dados de sequência é um objeto que cria e mantém uma sequência de valores.

Você pode fazer com que uma transformação Gerador de Sequência torne-se reutilizável e usá-la em vários mapeamentos. Você pode reutilizar uma transformação Gerador de Sequência ao fazer vários carregamentos em um único destino.

Por exemplo, se você tiver um arquivo de entrada grande e separá-lo em três sessões que são executadas em paralelo, utilize a transformação Gerador de Sequência para gerar valores de chave primária. Se você utilizar diferentes transformações Gerador de Sequência, o Serviço de Integração poderá gerar valores de chave diferentes. Em vez disso, utilize a transformação Gerador de Sequência reutilizável para todas as três sessões, a fim de fornecer um valor exclusivo para cada linha de destino.

Portas do Gerador de Sequência

A transformação Gerador de Sequência tem duas portas de saída: NEXTVAL e CURRVAL. Não é possível editar ou excluir essas portas. Do mesmo modo, você não pode adicionar portas à transformação.

A transformação de Gerador de Sequência tem portas de passagem e uma porta de saída, NEXTVAL. Não é possível editar ou excluir a porta de saída.

Portas de Passagem

Você pode adicionar uma porta à transformação de Gerador de Sequência como uma porta de passagem. As portas de passagem são portas de entrada e de saída que recebem dados de entrada e retornam os mesmos dados a um mapeamento sem alterá-los.

Você deve adicionar pelo menos uma porta de entrada à transformação e conectá-la a uma transformação ou origem upstream antes de vincular as portas de saída NEXTVAL e CURRVAL a destinos. Para adicionar uma porta de passagem à transformação, arraste uma porta de uma transformação ou origem upstream no mapeamento até a transformação de Gerador de Sequência.

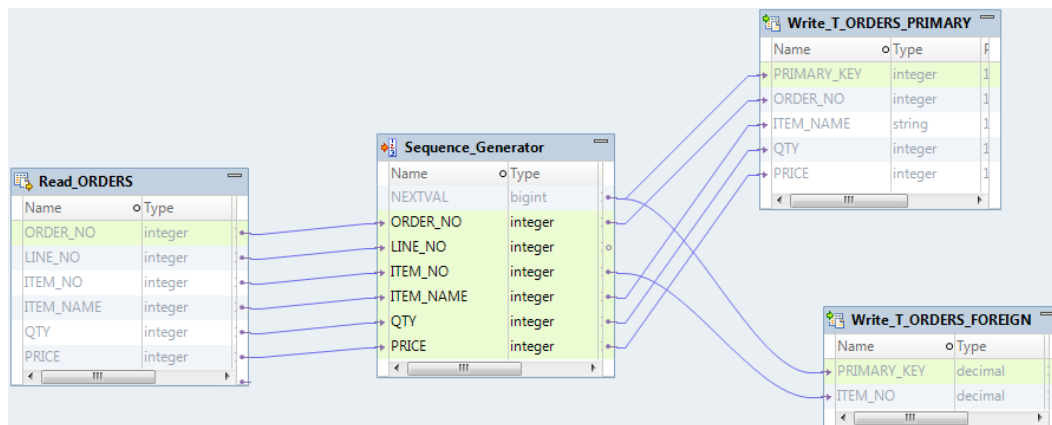
Você deve adicionar pelo menos uma porta de entrada à transformação e conectá-la a uma transformação ou origem upstream antes de vincular a porta de saída NEXTVAL aos destinos. Para adicionar uma porta de passagem à transformação, arraste uma porta de uma transformação ou origem upstream no mapeamento até a transformação de Gerador de Sequência.

Porta NEXTVAL

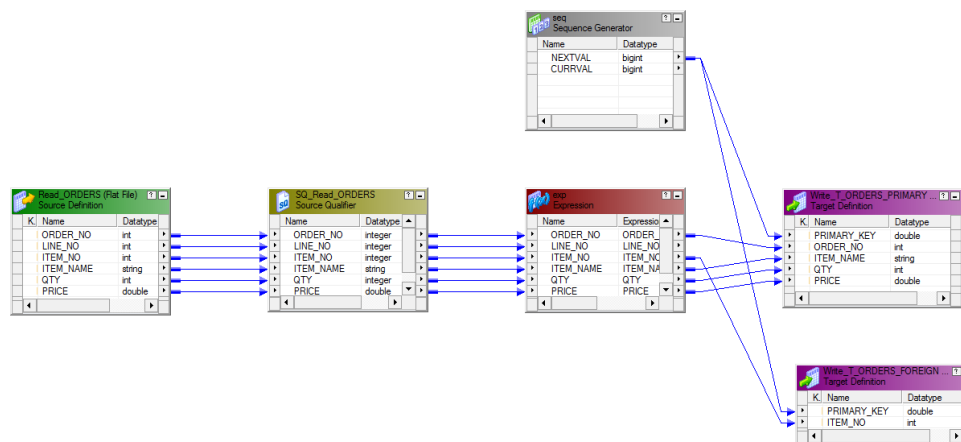
Você pode conectar NEXTVAL a uma transformação para gerar valores exclusivos para cada linha na transformação. Conecte a porta NEXTVAL a um destino ou transformação downstream para gerar uma sequência de números. Se você conectar NEXTVAL a várias transformações, o Serviço de Integração gerará a mesma sequência de números para cada transformação.

Você conecta a porta NEXTVAL para gerar a sequência com base nas propriedades Valor Inicial e Valor de Incremento. Se o Gerador de Sequência não estiver configurado para percorrer a sequência, a porta NEXTVAL gerará números de sequência até o valor final configurado.

A seguinte imagem mostra um mapeamento com a porta NEXTVAL da transformação de Gerador de Sequência conectada a uma origem e a dois destinos para gerar valores de chave primária e externa:



A seguinte imagem mostra um mapeamento com a porta NEXTVAL da transformação de Gerador de Sequência conectada a dois destinos para gerar valores de chave primária e externa:



Quando você configura a transformação de Gerador de Sequência com um Valor Inicial = 1 e um Valor de Incremento = 1, o Serviço de Integração gera os mesmos valores de chave primária para as tabelas de destino T_ORDERS_PRIMARY e T_ORDERS_FOREIGN.

Conecte NEXTVAL a várias transformações para gerar valores exclusivos para cada linha de cada transformação. Use a porta NEXTVAL para gerar números de sequência, conectando-a a um destino ou transformação downstream. Você conecta a porta NEXTVAL para gerar a sequência com base nas propriedades Valor Atual e Incrementar em. Se o Gerador de Sequência não estiver configurado para percorrer a sequência, a porta NEXTVAL gerará números de sequência até o Valor Final configurado.

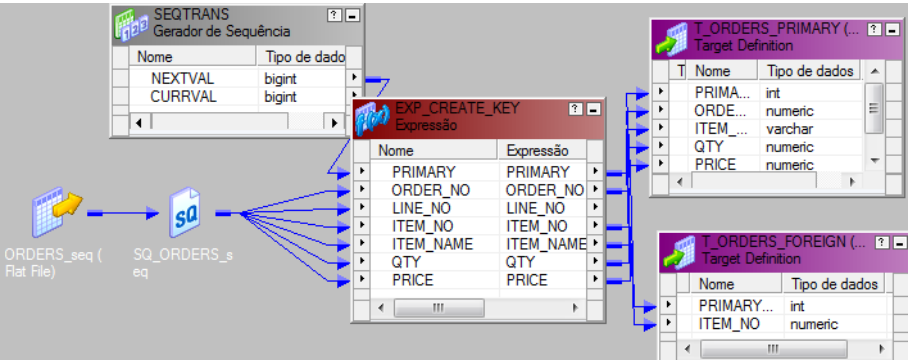
Por exemplo, você pode conectar o NEXTVAL a dois destinos em um mapeamento, a fim de gerar valores chaves primários exclusivos. O Serviço de Integração criará uma coluna de valores de chave primária exclusivos para cada tabela de destino. A coluna de valores de chave primária exclusivos é enviada a uma tabela de destino como um bloco de números de sequência. O outro destino recebe um bloco de números de sequência da transformação Gerador de Sequência, depois que o primeiro destino recebeu o bloco de números de sequência.

Por exemplo, configure a transformação de Gerador de Sequência da seguinte forma: Valor atual = 1, Incrementar em = 1. O Serviço de Integração gera os seguintes valores de chave primária para as tabelas de destino T_ORDERS_PRIMARY e T_ORDERS_FOREIGN:

T_ORDERS_PRIMARY TABLE: PRIMARY KEY	T_ORDERS_FOREIGN TABLE: PRIMARY KEY
1	6
2	7
3	8
4	9
5	10

Se você deseja os *mesmos* valores em mais de um destino que recebe os dados de uma única transformação, pode conectar uma transformação Gerador de Sequência para essa transformação precedente. O Serviço de Integração processa os valores em um bloco de números de sequência. Isto permite o Serviço de Integração transfira valores exclusivos para a transformação, e então encaminhe as linhas da transformação para os destinos.

A figura a seguir mostra um mapeamento com um Gerador de Sequência que transfere valores exclusivos para a transformação de Expressão:



A transformação de Expressão preenche os dois destinos com valores de chave primária idênticos.

Por exemplo, configure a transformação de Gerador de Sequência da seguinte forma: Valor atual = 1, Incrementar em = 1. O Serviço de Integração gera os seguintes valores de chave primária para as tabelas de destino T_ORDERS_PRIMARY e T_ORDERS_FOREIGN:

T_ORDERS_PRIMARY TABLE: PRIMARY KEY	T_ORDERS_FOREIGN TABLE: PRIMARY KEY
1	1
2	2
3	3
4	4
5	5

Nota: Ao executar uma sessão particionada em uma grade, a transformação de Gerador de Sequência ignora valores, dependendo do número de linhas em cada partição.

Criar Chaves

É necessário criar valores de chave primários ou estrangeiros com a transformação Gerador de Sequência conectando a porta NEXTVAL a uma transformação de destino ou de fluxo para baixo. É possível usar um intervalo de valores de 1 a 9.223.372.036.854.775,807 com o menor intervalo de 1.

É necessário criar valores de chave primários ou estrangeiros com a transformação Gerador de Sequência conectando a porta NEXTVAL a uma transformação de destino ou de fluxo para baixo. É possível usar um intervalo de valores de 1 a 9.223.372.036.854.775,807 com o menor intervalo de 1. Você também pode criar chaves de composição para identificar cada linha na tabela.

Ao criar chaves primárias ou externas, use a opção Ciclo para evitar que o Serviço de Integração crie chaves primárias duplicadas. É possível fazer isso selecionando a opção Truncar Tabela de Destino nas propriedades da sessão ou criando chaves de composição.

Para criar uma chave composta, é possível configurar o Serviço de Integração para circular por um conjunto menor de valores. Por exemplo, se três lojas gerarem números de pedido, configure uma transformação de Gerador de Sequência para percorrer valores de 1 a 3, com incrementos de 1. Quando você conecta a porta ORDER_NO à transformação de Gerador de Sequência, os valores gerados criam chaves de composição exclusivas.

O seguinte exemplo mostra chaves de composição e números de pedido:

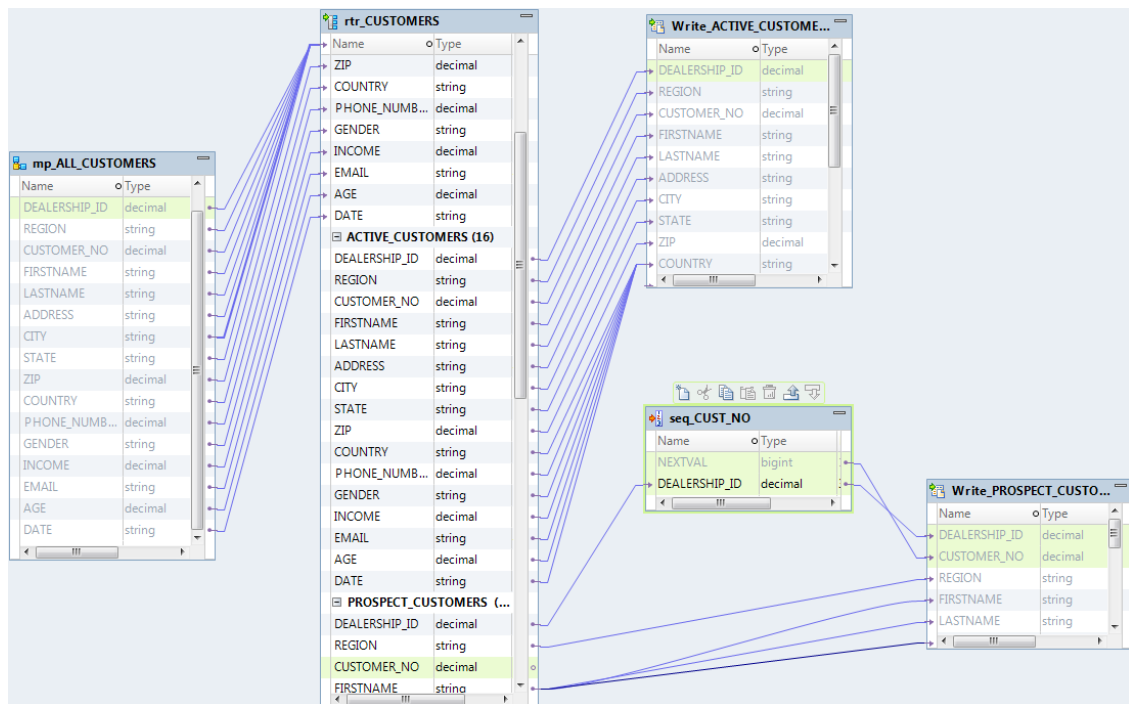
COMPOSITE_KEY	ORDER_NO
1	12345
2	12345
3	12345
1	12346
2	12346
3	12346

Substituir Valores não Encontrados

Ao usar uma transformação de Gerador de Sequência para substituir chaves ausentes, você também pode usar uma transformação de Roteador para filtrar os valores nulos de colunas com valores atribuídos. Você conecta a transformação de Roteador à transformação de Gerador de Sequência e usa NEXTVAL para gerar uma sequência de valores numéricos para preencher os valores nulos.

Por exemplo, para substituir valores nulos em uma coluna CUSTOMER_NO, crie um mapeamento com uma origem que contenha dados de clientes. Você adiciona uma transformação de Roteador para filtrar os clientes com números de cliente atribuídos daqueles com valores nulos. Adicione uma transformação de Gerador de Sequência para gerar valores CUSTOMER_NO exclusivos. Você adiciona destinos de cliente nos quais gravar dados.

A figura abaixo mostra um mapeamento que substitui valores nulos na coluna CUSTOMER_NO:



Use a transformação de Gerador de Sequência para substituir chaves ausentes usando NEXTVAL pelas funções IIF e ISNULL.

Por exemplo, para substituir valores nulos na coluna ORDER_NO, você cria uma transformação de Gerador de Sequência com as propriedades e arrasta a porta NEXTVAL até uma transformação de Expressão. Na transformação Expressão, arraste a porta ORDER_NO até a transformação junto com todas as outras portas necessárias. Em seguida, crie a porta de saída ALL_ORDERS.

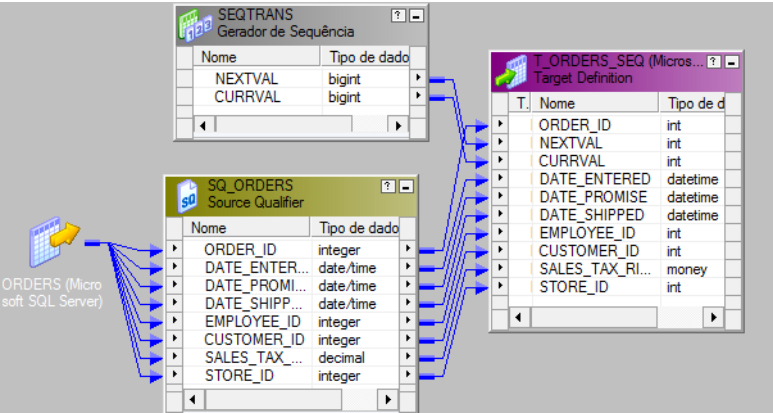
Em ALL_ORDERS, você pode inserir a expressão a seguir para substituir pedidos nulos:

```
IIF( ISNULL( ORDER_NO ), NEXTVAL, ORDER_NO )
```

CURRVAL

CURRVAL é NEXTVAL com a adição do Valor de Incremento. Em geral, você somente conecta a porta CURRVAL quando a porta NEXTVAL já está conectada a uma transformação downstream. Quando uma linha entra em uma transformação conectada à porta CURRVAL, o Serviço de Integração transfere o último valor criado NEXTVAL mais 1.

A figura a seguir demonstra a conexão das portas CURRVAL e NEXTVAL a um destino:



Por exemplo, configure a transformação de Gerador de Sequência da seguinte forma: Valor atual = 1, Incrementar em = 1. O Serviço de Integração gera os seguintes valores para NEXTVAL e CURRVAL:

NEXTVAL	CURRVAL
1	2
2	3
3	4
4	5
5	6

Caso você conecte a porta CURRVAL sem conectar a porta NEXTVAL, o Serviço de Integração transfere um valor constante para cada linha. Quando você conectar a porta CURRVAL em uma transformação de Gerador de Sequência, o Serviço de Integração processa uma linha em cada bloco. É possível otimizar o desempenho conectando apenas a porta NEXTVAL em um mapeamento.

Nota: Quando você executa uma sessão particionada em uma grade, a transformação do Gerador de Sequência pode ignorar valores, dependendo do número de linhas em cada partição.

Propriedades da Transformação Gerador de Sequência

Configure propriedades de transformação que o Serviço de Integração usa para gerar valores sequenciais.

A seguinte tabela lista as propriedades que você configura para um objeto de dados de sequência e uma nova sequência:

Propriedade	Descrição
Valor de Início	Valor inicial da sequência gerada que você deseja que o Serviço de Integração use, se estiver usando a opção Ciclo. Se você selecionar Ciclo, o Serviço de Integração voltará a esse valor quando alcançar o valor final. O padrão é 0. O valor máximo é 9.223.372.036.854.775.806.
Valor Final	Valor máximo gerado pelo Serviço de Integração. Se o Serviço de Integração atingir esse valor durante a sessão e a sequência não estiver configurada como ciclo, a sessão será encerrada. O valor máximo é 9,223,372,036,854,775,807.
Valor de Incremento	Diferença entre dois valores consecutivos da porta NEXTVAL. O padrão é 1. Deve ser um inteiro positivo. O valor máximo é 2,147,483,647.
Ciclo	Se ativado, o Serviço de Integração percorrerá o intervalo da sequência e será iniciado com o valor inicial. Se desativado, o Serviço de Integração interromperá a sequência no valor final configurado. O Serviço de Integração encerrará a sessão com erros de overflow se atingir o valor final e ainda houver linhas a serem processadas.

A seguinte lista descreve as propriedades da transformação de Gerador de Sequência que você pode configurar:

Valor Inicial

Valor inicial da sequência gerada que você deseja que o Serviço de Integração use, se estiver usando a opção Ciclo. Se você selecionar Ciclo, o Serviço de Integração voltará a esse valor quando alcançar o valor final.

O padrão é 0.

O valor máximo é 9.223.372.036.854.775.806.

Incrementar Em

Diferença entre dois valores consecutivos da porta NEXTVAL.

O padrão é 1.

Deve ser um inteiro positivo.

O valor máximo é 2,147,483,647.

Valor Final

Valor máximo gerado pelo Serviço de Integração. Se o Serviço de Integração atingir esse valor durante a sessão e a sequência não estiver configurada como ciclo, a sessão será encerrada.

O valor máximo é 9,223,372,036,854,775,807.

Se você conectar a porta NEXTVAL a uma porta de inteiro downstream, defina o valor final como um valor menor que ou igual ao valor máximo de inteiro. Se a NEXTVAL exceder o valor máximo do tipo de dados para a porta downstream, a sessão falhará.

Valor Atual

Valor atual da sequência. Insira o valor que você deseja que o Serviço de Integração use como valor inicial na sequência. Para percorrer vários valores, o valor deve ser maior ou igual ao valor inicial e menor do que o valor final.

Se o Número de Valores Guardados em Cache for definido como 0, o Serviço de Integração atualizará o valor atual para refletir o último valor gerado para a sessão mais um e o usará como base na próxima vez que você executar a sessão. Contudo, se você usar a opção Redefinir, o Serviço de Integração redefinirá esse valor como o valor original após cada sessão.

Nota: Se você editar essa configuração, a sequência será redefinida com a nova configuração. Se você redefinir o Valor Atual como 10 e o incremento for 1, na próxima vez que usar a sessão, o Serviço de Integração gerará o valor inicial 10.

O valor máximo é 9.223.372.036.854.775.806. Se o valor atual exceder o valor máximo, o Serviço de Integração definirá o valor como NULL.

Ciclo

Se ativado, o Serviço de Integração percorrerá o intervalo da sequência e será iniciado com o valor inicial.

Se desativado, o Serviço de Integração interromperá a sequência no valor final configurado. O Serviço de Integração encerrará a sessão com erros de overflow se atingir o valor final e ainda houver linhas a serem processadas.

Número de Valores Guardados em Cache

Número de valores sequenciais que o Serviço de Integração armazena em cache de cada vez. Use essa opção quando várias sessões usarem o mesmo Gerador de Sequência reutilizável ao mesmo tempo para garantir que cada sessão receba valores exclusivos. O Serviço de Integração atualiza o repositório enquanto armazena cada valor em cache. Quando for definido como 0, o Serviço de Integração não armazenará valores em cache.

O valor padrão é 0.

O valor padrão de um Gerador de Sequência reutilizável é 1.000.

O valor máximo é 9,223,372,036,854,775,807.

Esta propriedade é aplicável apenas às transformações reutilizáveis do Gerador de Sequência.

Redefinir

Se ativado, o Serviço de Integração gerará valores com base no valor atual original para cada sessão. Se desativado, o Serviço de Integração atualizará o valor atual para refletir o último valor gerado para a sessão mais um e o usará como base para a próxima vez que a sessão for executada.

Esta propriedade é aplicável apenas às transformações não reutilizáveis do Gerador de Sequência.

Nível de Rastreamento

Nível de detalhes sobre a transformação que o Serviço de Integração grava no log da sessão.

Valor Inicial

Utilize Ciclo para gerar uma sequência de repetição, como números de 1 a 12 para corresponder aos meses do ano.

1. Insira o menor valor na sequência que você deseja que o Serviço de Integração use para o valor inicial.
2. Insira o maior valor a ser usado para o Valor Final.
3. Selecione Ciclo.

À medida que percorrer a sequência, o Serviço de Integração atingirá o valor final configurado. Ele continuará até o fim da sequência e iniciará o ciclo novamente, começando pelo valor inicial configurado.

Incrementar em

O Serviço de Integração gera uma sequência na porta NEXTVAL com base nas propriedades Valor Atual e Incrementar em na transformação de Gerador de Sequência.

A propriedade Valor atual é o valor em que o Serviço de Integração começa a criar a sequência para cada sessão. Incrementar em é o número inteiro que o Serviço de Integração adiciona ao valor existente, para criar o novo valor da sequência. Por padrão, o Valor Atual está definido como 1, enquanto Incrementar em está definido como 1.

Por exemplo, você pode criar uma transformação de Gerador de Sequência com a um valor atual de 1000 e um incremento de 10. Se você passar três linhas através do mapeamento, o Serviço de Integração gera o seguinte conjunto de valores:

```
1000  
1010  
1020
```

Valor final

Valor Final é o valor máximo que você deseja que o Serviço de Integração gere. Se o Serviço de Integração atingir o valor final e o Gerador de Sequência não estiver configurado para percorrer a sequência, a sessão falhará com um erro de overflow.

O valor final é o valor máximo que você deseja que o Serviço de Integração gere. Se o Serviço de Integração atingir o valor final e o gerador de sequência não estiver configurado para percorrer a sequência, a execução de mapeamento falhará com um erro de overflow.

Defina o valor final como qualquer inteiro entre 1 e 9.233.372.036.854.775.807. Se você conectar a porta NEXTVAL a uma porta de inteiro downstream, defina o valor final como um valor menor que ou igual ao valor máximo de inteiro. Por exemplo, se você conectar a porta NEXTVAL a uma porta de Inteiro Curto, configure o valor final como no máximo 32.767. Se a NEXTVAL exceder o valor máximo do tipo de dados para a porta downstream, a sessão falhará.

Defina o valor final como qualquer inteiro entre 1 e 9.233.372.036.854.775.807. Se você conectar a porta NEXTVAL a uma porta de inteiro downstream, defina o valor final como um valor menor que ou igual ao valor máximo de inteiro. Por exemplo, se você conectar a porta NEXTVAL a uma porta de Inteiro Curto, configure o valor final como no máximo 32.767. Se NEXTVAL exceder o valor máximo do tipo de dados para a porta downstream, o mapeamento falhará.

Valor de Incremento

O Serviço de Integração gera uma sequência na porta NEXTVAL com base nas propriedades Valor Atual e Incrementar em na transformação de Gerador de Sequência.

A propriedade Valor atual é o valor em que o Serviço de Integração começa a criar a sequência para cada sessão. Incrementar em é o número inteiro que o Serviço de Integração adiciona ao valor existente, para criar o novo valor da sequência. Por padrão, o Valor Atual está definido como 1, enquanto Incrementar em está definido como 1.

Por exemplo, você pode criar uma transformação de Gerador de Sequência com a um valor atual de 1000 e um incremento de 10. Se você passar três linhas através do mapeamento, o Serviço de Integração gera o seguinte conjunto de valores:

```
1000
1010
1020
```

Percorrer um Intervalo de Valores

Você pode estabelecer um intervalo de valores para a transformação Gerador de Sequência. Se você usar a opção ciclo, a transformação Gerador de Sequência repete o intervalo quando atingir o valor final.

Por exemplo, se você definir o intervalo da sequência para iniciar em 10 e terminar em 50 e definir um valor de incremento de 10, a transformação de Gerador de Sequência criará os valores 10, 20, 30, 40, 50. A sequência é iniciada novamente em 10.

Valor Atual

O Serviço de Integração usa o valor atual como base para os valores gerados para cada sessão. Para indicar o valor que você deseja que o Serviço de Integração use na primeira vez que utilizar a transformação de Gerador de sequência, insira-o como o valor atual. Se você deseja usar a transformação de Gerador de Sequência para percorrer uma série de valores, o valor atual deve ser maior ou igual ao Valor Inicial e menor que o valor final.

O Serviço de Integração usa o valor atual como base para os valores gerados para cada execução de mapeamento. O Serviço de Integração sempre usa o valor inicial como o valor atual na transformação de Gerador de Sequência.

No final de cada sessão, o Serviço de Integração atualizará o valor atual para o último valor gerado para sessão mais 1 se a propriedade Número de Valores Guardados em Cache do Gerador de Sequência for 0. Por exemplo, se o Serviço de Integração terminar uma sessão com um valor gerado igual a 101, ele atualizará o valor atual do Gerador de Sequência para 102 no repositório. Da próxima vez em que o Gerador de Sequência for usado, o Serviço de Integração usará 102 como a base para o próximo valor gerado. Se o valor de Incrementar em do Gerador de Sequência for 1, quando o Serviço de Integração iniciar outra sessão usando o Gerador de Sequência, o primeiro valor gerado será 102.

No final de cada execução de mapeamento, o Serviço de Integração atualizará o valor atual para o último valor gerado para a sessão mais 1 se a propriedade Número de Valores Guardados em Cache do Gerador de Sequência for 0. Por exemplo, se o Serviço de Integração terminar uma sessão com um valor gerado igual a 101, ele atualizará o valor atual do Gerador de Sequência para 102 no repositório. Da próxima vez em que o Gerador de Sequência for usado, o Serviço de Integração usará 102 como a base para o próximo valor gerado. Se o valor de Incrementar em do Gerador de Sequência for 1, quando o Serviço de Integração iniciar outra sessão usando o Gerador de Sequência, o primeiro valor gerado será 102.

Se você tem múltiplas versões de uma transformação do Gerador de sequência, o Serviço de Integração atualiza o valor atual para todas as versões quando executar uma sessão. O Serviço de integração atualiza o valor atual em todas as versões, independentemente de você ter feito check-out da transformação do Gerador de Sequência ou do mapeamento pai. O valor atual atualizado substitui um valor atual editado para uma transformação do Gerador de sequência, se os dois valores forem diferentes.

Por exemplo, o Usuário 1 cria a transformação do Gerador de sequência e faz o seu check-in, salvando um valor atual de 10 para o Gerador de sequência da versão 1. Então, o Usuário 1 confirma a transformação do

Gerador de sequência e insere um novo valor atual de 100 para o Gerador de sequência da versão 2. O Usuário 1 mantém o check-out da transformação do Gerador de sequência. Enquanto isso, o Usuário 2 executa uma sessão que usa a transformação do Gerador de Sequência da versão 1. O Serviço de Integração usa o valor de 10 do check-in como o valor atual quando o Usuário 2 executa a sessão. Quando a sessão termina, o valor atual é 150. O Serviço de Integração atualiza o valor atual como 150 para a versão 1 e a versão 2 da transformação do Gerador de sequência, embora o Usuário 1 tenha feito o check-out da transformação do Gerador de sequência.

Se você abrir o mapeamento depois que executar a sessão, o valor atual exibe o último valor gerado para a sessão mais um. Como o Serviço de Integração usa o valor atual para determinar o primeiro valor de cada sessão, você deverá editar o valor atual apenas quando desejar redefinir a sequência.

Se você tiver múltiplas versões da transformação do Gerador de Sequência e quiser redefinir a sequência, faça check-in do mapeamento ou da transformação do Gerador de Sequência reutilizável depois de modificar o valor atual.

Nota: Se você configurar o Gerador de sequência para Redefinir, o Serviço de Integração usa o valor atual como a base para o primeiro valor gerado para cada sessão.

Número de Valores Guardados em Cache

Número de Valores Guardados em Cache determina o número de valores cujo cache o Serviço de Integração efetua uma vez. Quando Número de Valores em Cache é maior que zero, o Serviço de Integração efetua o cache do número configurado de valores e atualiza o valor atual cada vez que efetua o cache dos valores.

Quando múltiplas sessões usam a mesma transformação reutilizável Gerador de Sequência ao mesmo tempo, pode haver múltiplas instâncias da transformação Gerador de Sequência. Para evitar gerar os mesmos valores para cada sessão, reserve um intervalo de valores de sequência para cada sessão, configurando o Número de Valores em Cache.

Sugestão: Para melhorar o desempenho ao executar uma sessão em uma grade, aumente o número de valores em cache para a transformação Gerador de Sequência. Isto reduz a comunicação exigida entre os processos DTM mestre e operador, e o repositório.

Geradores de Sequência Não-Reutilizáveis

No caso de transformações de Gerador de Sequência não reutilizáveis, o Número de Valores em Cache é definido para zero por padrão, e o Serviço de Integração não armazena valores em cache durante a sessão. Quando o Serviço de Integração não armazena os valores em cache, ele acessa o repositório do valor atual no início da sessão. Em seguida, o Serviço de Integração gera valores para a sequência. No final da sessão, o Serviço de Integração atualiza o valor atual no repositório.

Quando você define Número de Valores em Cache maiores que zero, o Serviço de Integração armazena os valores em cache durante a sessão. No início da sessão, o Serviço de Integração acessa o repositório do valor atual, armazena em cache o número configurado de valores e atualiza o valor atual de maneira apropriada. Se o Serviço de Integração usar todos os valores no cache, ele acessa o repositório do próximo conjunto de valores e atualiza o valor atual. No final da sessão, o Serviço de Integração elimina quaisquer valores restantes no cache.

Para transformações de Gerador de Sequência não reutilizáveis, definir um Número de Valores em Cache maior que zero pode aumentar o número de vezes que o Serviço de Integração acessa o repositório durante a sessão. Isso também gera sessões de valores ignorados, uma vez que valores em cache não utilizados são descartados no final de cada sessão.

Por exemplo, configure uma transformação Gerador de Sequência da seguinte forma: Número de Valores Guardados em Cache = 50, Valor Atual = 1, Incremento De = 1. Ao iniciar a primeira sessão, o Serviço de Integração armazena 50 valores para a sessão e atualiza o valor atual para 50 no repositório. O Serviço de

Integração usa valores de 1 a 39 para a sessão e descarta os valores não utilizados, de 40 a 49. Quando o Serviço de Integração executa a sessão novamente, ele verifica o valor atual do repositório, que é 50. Ele armazena então os 50 valores seguintes e atualiza o valor atual para 100. Durante a sessão, ele usa valores de 50 a 98. Os valores gerados para as duas sessões são 1 a 39 e 50 a 98.

Geradores de Sequência Reutilizáveis

Quando há uma transformação Gerador de Sequência reutilizável em várias sessões e as sessões forem executadas ao mesmo tempo, use Número de Valores Guardados em Cache para garantir que cada sessão receba valores exclusivos na sequência. Por padrão, o Número de Valores Guardados em Cache é definido como 1.000 para Geradores de Sequência reutilizáveis.

Quando várias sessões usam a mesma transformação Gerador de Sequência ao mesmo tempo, há o risco de serem gerados os mesmos valores para cada sessão. Para evitar isso, configure o Número de Valores Guardados em Cache, para que o Serviço de Integração armazene um número de valores definido para cada sessão.

Por exemplo, você configura uma transformação Gerador de Sequência como a seguir: Número de Valores Guardados em Cache = 50, Valor Atual = 1, Incremento De = 1. Duas sessões usam o Gerador de Sequência e foram agendadas para ser executadas aproximadamente ao mesmo tempo. Ao iniciar a primeira sessão, o Serviço de Integração armazena 50 valores para a sessão e atualiza o valor atual para 50 no repositório. O Serviço de Integração inicia a sessão usando valores de 1 a 50. Ao iniciar a segunda sessão, o Serviço de Integração verifica o valor atual do repositório, que é 50. Ele armazena então os 50 valores seguintes e atualiza o valor atual para 100. Na segunda sessão, ele usa valores de 51 a 100. Quando ambas as sessões usam todos os valores guardados em cache, o Serviço de Integração armazena outro conjunto de valores e atualiza o valor atual, para garantir que esses valores permaneçam exclusivos para o Gerador de Sequência.

Para transformações Gerador de Sequência reutilizáveis, você pode reduzir o Número de Valores Guardados em Cache para minimizar o descarte de valores, mas ele deve ser maior do que um. Ao reduzir o Número de Valores Guardados em Cache, você pode aumentar o número vezes o Serviço de Integração acessa o repositório, para guardar valores no cache durante a sessão.

Redefinir

Se você selecionar Redefinir para uma transformação de Gerador de Sequência não reutilizável, o Serviço de Integração gerará valores com base no valor inicial original sempre que iniciar a sessão. Caso contrário, o Serviço de Integração atualizará o valor atual para refletir o último valor gerado mais o valor do incremento e usará o valor atualizado na próxima vez que usar a transformação de Gerador de Sequência.

Se você configurar uma transformação de Gerador de Sequência não reutilizável para usar a propriedade reset, o Serviço de Integração usará o valor inicial original para cada execução de mapeamento. Caso contrário, o Serviço de Integração incrementará o valor atual e o usará na próxima execução de mapeamento.

Por exemplo, você configura uma transformação de Gerador de Sequência para criar valores de 1 a 1.000 com um incremento de 1. Você escolhe redefinir para redefinir o valor inicial como 1. Durante a execução da primeira sessão, o Serviço de Integração gera números de 1 a 234. Em cada execução de mapeamento subsequente, o Serviço de Integração gerará novamente números começando com o valor inicial de 1.

Por exemplo, você configura uma transformação de Gerador de Sequência para criar valores de 1 a 1.000 com um incremento de 1, um valor inicial de 1 e escolhe redefinir. Durante a primeira execução de mapeamento, o Serviço de Integração gera números de 1 a 234. Em cada execução de mapeamento subsequente, o Serviço de Integração gerará novamente números começando com o valor inicial de 1.

Se você não redefinir, o Serviço de Integração atualizará o valor atual para 235 ao final da primeira execução. Da próxima vez em que a transformação de Gerador de Sequência for usada, o primeiro valor gerado será 235.

Nota: Redefinir é desativado para transformações Gerador de Sequência reutilizáveis.

Manter Ordem das Linhas

Mantenha a ordem de linhas dos dados de entrada na transformação. Selecione essa opção para o Serviço de Integração não realizar nenhuma otimização que possa alterar a ordem das linhas.

Quando o Serviço de Integração realiza otimizações, ele pode perder uma ordem estabelecida anteriormente no mapeamento. Você pode estabelecer a ordem em um mapeamento com uma origem de arquivo simples classificada, uma origem relacional classificada ou uma transformação do Classificador. Quando você configura uma transformação para manter a ordem das linhas, o Serviço de Integração considera essa configuração quando ele executa as otimizações para o mapeamento. O Serviço de Integração realizará otimizações para a transformação se ele puder manter a ordem. O Serviço de Integração não realizará otimizações para a transformação se a otimização alterar a ordem das linhas.

Objeto de Dados de Sequência

Um objeto de dados de sequência cria e mantém uma sequência de valores numéricos. A transformação de Gerador de Sequência usa o objeto de dados de sequência para gerar valores para a transformação.

Você pode usar um objeto de dados de sequência reutilizável em várias transformações de Gerador de Sequência. Todas as transformações de Gerador de Sequência que usam o mesmo objeto de dados de sequência usarão a mesma sequência de valores se forem executados no mesmo Serviço de Integração. Você também pode usar um objeto de dados de sequência reutilizável em uma transformação de Gerador de Sequência não reutilizável. Você pode usar um objeto de dados de sequência não reutilizável de uma transformação de Gerador de Sequência não reutilizável.

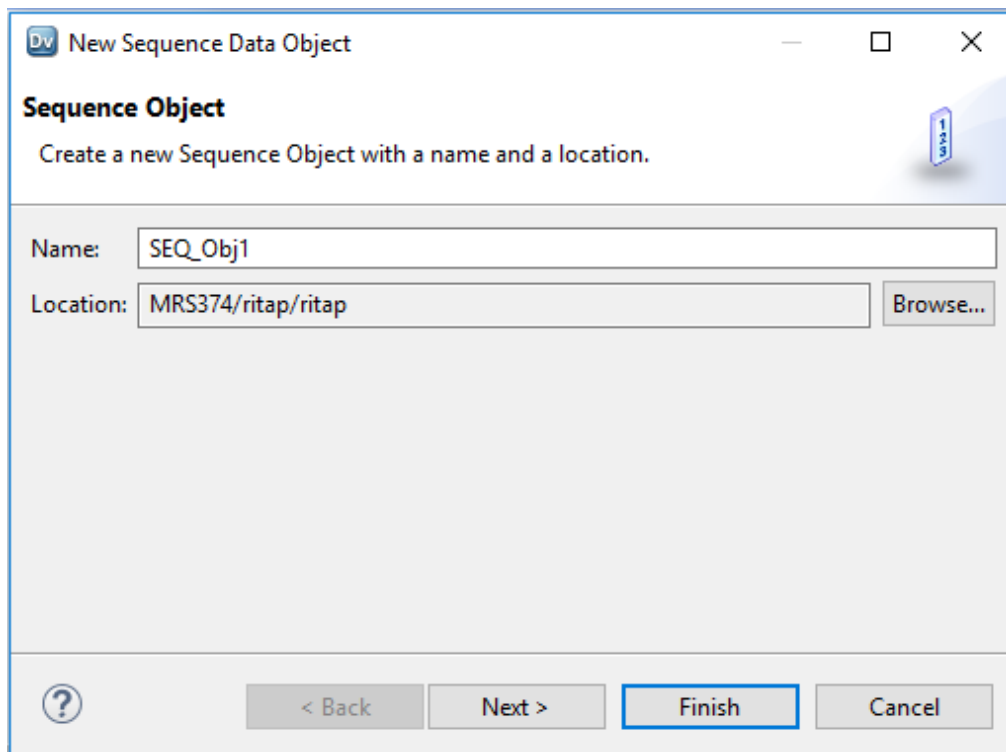
Por exemplo, crie vários mapeamentos que gravam no mesmo campo de chave primária em uma tabela relacional. Cada mapeamento usa a mesma transformação de Gerador de Sequência reutilizável, que usa o mesmo objeto de dados de sequência reutilizável e é executado no mesmo Serviço de Integração. Cada mapeamento grava valores exclusivos no campo de chave primária.

Criando um objeto de dados de sequência

Para usar um objeto de dados de Sequência para criar uma transformação de Gerador de Sequência, crie o objeto de dados de Sequência, configure as propriedades do objeto e selecione o objeto na caixa de diálogo Transformação de Gerador de Sequência.

1. No editor de mapeamento, role para baixo na paleta de mapeamento para localizar a transformação de Gerador de Sequência e arraste-a até o mapeamento.
O assistente de **Novas Transformações** é aberto.
2. Clique em **Novo Objeto de Dados de Sequência**.

O assistente para **Novo Objeto de Dados** é aberto.




New Sequence Data Object

Sequence Object
Create a new Sequence Object with a name and a location.

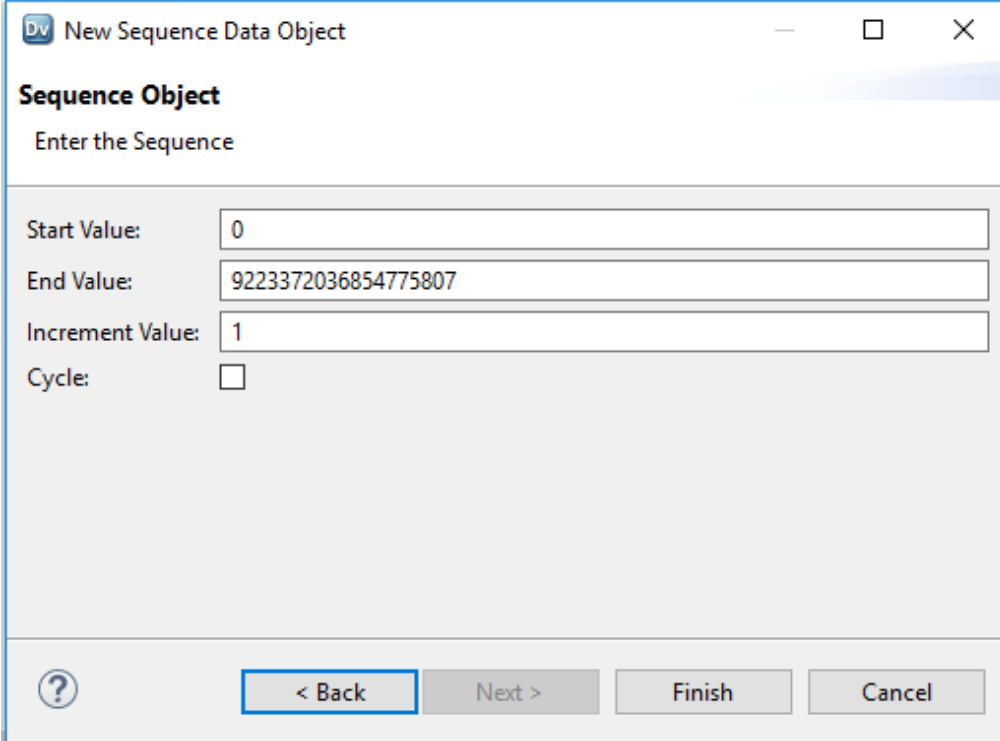
Name: SEQ_Obj1

Location: MRS374/ritap/ritap



3. Insira um nome para o objeto de dados de Sequência.
A convenção de nomenclatura para objetos de dados de Sequência é SEQ_<nome do objeto de dados>.
4. Clique em **Avançar** para configurar as propriedades do objeto de dados de Sequência.

Se você criar uma transformação de Gerador de Sequência a partir do objeto, essa transformação usará as propriedades que você inserir para o objeto de dados. A seguinte imagem mostra as propriedades que você pode configurar:



5. Depois de configurar as propriedades do objeto de dados, você pode criar uma transformação de Gerador de Sequência usando o objeto de dados de Sequência. Ao criar a transformação, nomeie a transformação de Gerador de Sequência e selecione **Escolher um objeto de Sequência existente**. Navegue até o objeto de dados e clique em **OK**.

A transformação de Gerador de Sequência aparece no editor de mapeamento com um porta somente de saída NEXTVAL. É possível conectar a porta NEXTVAL a um destino ou transformação em downstream para gerar uma sequência de números.

Criando uma Transformação de Gerador de Sequência

Para usar uma transformação Gerador de Sequência em um mapeamento, adicione-a ao mapeamento, configure as propriedades de transformação e conecte NEXTVAL ou CURRVAL a uma ou mais transformações.

Para criar uma Transformação Gerador de Sequência:

1. No Mapping Designer, clique em Transformação > Criar. Selecione a Transformação de Gerador de Sequência.

A convenção de nomenclatura das transformações Gerador de Sequência é *SEQ_Nome da Transformação*.

2. Insira um nome para o Gerador de Sequência e clique em Criar. Clique em Concluído.
O Designer cria a transformação Gerador de Sequência.
3. Clique duas vezes na barra de títulos da transformação.
4. Insira uma descrição para a transformação.
5. Selecione a guia Propriedades. Insira as configurações.
Nota: Não é possível substituir as propriedades da transformação Gerador de Sequência no nível de sessão. Isso protege a integridade dos valores de sequência gerados.
6. Clique em OK.
7. Para gerar novas sequências durante uma sessão, conecte a porta NEXTVAL a pelo menos uma transformação no mapeamento.
Use as portas NEXTVAL ou CURRVAL em uma expressão em outras transformações.

Criando uma Transformação de Gerador de Sequência

Para usar uma transformação de Gerador de Sequência em um mapeamento, adicione-a ao mapeamento, configure as propriedades de transformação e conecte NEXTVAL a uma ou mais transformações.

1. No editor de mapeamento, role para baixo na paleta de mapeamento para localizar a transformação de Gerador de Sequência e arraste-a até o mapeamento.
O assistente de **Novas Transformações** é aberto.

New Sequence Generator Transformation

Sequence Generator
Create a Sequence Generator Transformation

Name: SEQ_Gen1

Location: MRS374/ritap/ritap/seq_gen_test

☒ Create a new sequence
☐ Choose an existing Sequence object

Browse

[New Sequence Data Object](#)

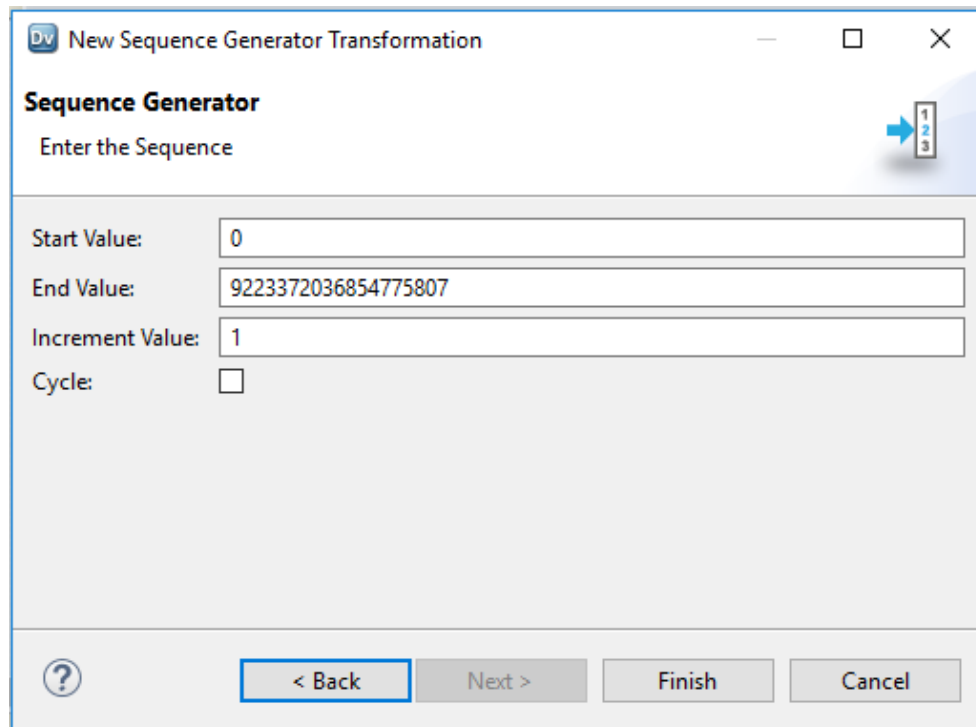
< Back Next > Finish Cancel

2. Forneça um nome para a transformação Gerador de Sequência.

A convenção de nomenclatura para transformações de Gerador de Sequência é SEQ_<nome da transformação>.

3. Escolha para criar uma nova sequência ou usar um objeto de Sequência existente.

- Para criar uma nova sequência, selecione **Criar uma nova sequência**. Clique em **Avançar** para configurar as propriedades da sequência. A seguinte imagem mostra as propriedades que você pode configurar:



- Para usar um objeto de Sequência existente, selecione **Escolher um objeto de Sequência existente**. Navegue até o objeto de Sequência e clique em **OK**.

A transformação de Gerador de Sequência aparece no editor de mapeamento com um porta somente de saída NEXTVAL. É possível conectar a porta NEXTVAL a um destino ou transformação em downstream para gerar uma sequência de números.

Perguntas frequentes

Posso alterar uma transformação de Gerador de Sequência não reutilizável para torná-la reutilizável?

Você não pode tornar a transformação reutilizável, mas pode alterar a transformação para usar um objeto de dados de sequência. O objeto de dados de sequência mantém a integridade da sequência, independentemente do número de transformações que o utilizam.

Posso colocar uma transformação de Gerador de Sequência não reutilizável em um mapplet?

Não, não é possível. Mapplets são objetos reutilizáveis, por isso todos os objetos do mapplet também devem ser reutilizáveis. Use uma transformação Gerador de Sequência reutilizável.

Transformação de Gerador de Sequência em um ambiente não nativo

O processamento da transformação de Gerador de Sequência em um ambiente não nativo depende do mecanismo que executa a transformação.

Considere o suporte para os seguintes mecanismos de tempo de execução não nativos:

- Mecanismo Blaze. Suportado com restrições.
- Mecanismo Spark. Com suporte com restrições em mapeamentos em lote. Sem suporte em mapeamentos de fluxo.
- Mecanismo Databricks Spark. Sem suporte.

Transformação de Gerador de Sequência no mecanismo Blaze

Um mapeamento com uma transformação de Gerador de Sequência consome recursos significativos quando as seguintes condições são verdadeiras:

- Você define a propriedade **Manter Ordem da Linha** na transformação como *true*.
- O mapeamento é executado em uma única partição.

Transformação de Gerador de Sequência no mecanismo Spark

A transformação do Gerador de Sequência não mantém a ordem da linha nos dados de saída. Se você ativar a propriedade **Manter Ordem da Linha** na transformação, o Serviço de Integração de Dados irá ignorar a propriedade.

CAPÍTULO 24

Transformação Classificador

Este capítulo inclui os seguintes tópicos:

- [Visão Geral da Transformação Classificador, 406](#)
- [Classificando Dados, 406](#)
- [Propriedades da Transformação de Classificador, 408](#)
- [Criando uma Transformação Classificador, 410](#)

Visão Geral da Transformação Classificador

Você pode classificar dados com a transformação Classificador. Você pode classificar dados em ordem crescente ou decrescente, de acordo com uma chave de classificação especificada. Você também pode configurar a transformação Classificador para uma classificação que não faça distinção entre letras maiúsculas e minúsculas e especificar se as linhas de saída devem ser diferentes. A transformação Classificador é uma transformação ativa. Ela precisa estar conectada ao fluxo de dados.

Você pode classificar dados de origens de arquivos simples ou relacionais. Você também pode utilizar a transformação Classificador para classificar dados que passam através de uma transformação Agregador configurada para utilizar uma entrada classificada.

Quando cria uma transformação Classificador em um mapeamento, você especifica uma ou mais portas como uma chave de classificação e configura cada porta de chave de classificação para classificar em ordem crescente ou decrescente. Você também configura critérios de classificação que o Serviço de Integração aplica a todas as portas de chave de classificação e os recursos de sistema que ele aloca para executar a operação de classificação.

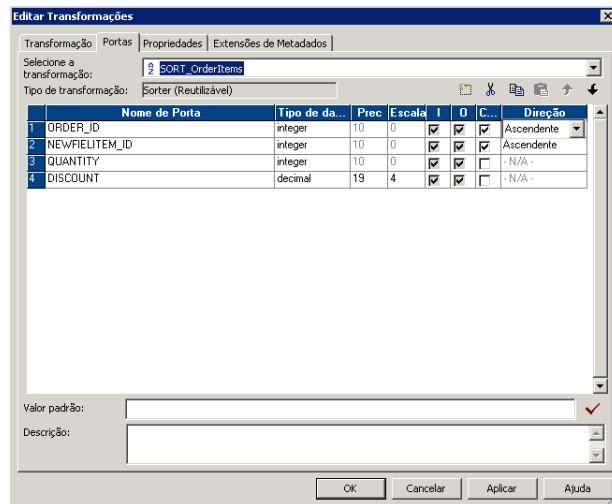
Classificando Dados

A transformação Classificador contém somente portas de entrada/saída. Todos os dados que passam pela transformação Classificador são classificados de acordo com uma chave de classificação. A chave de classificação é uma ou mais portas que você quer usar como critérios de classificação.

Você pode especificar mais de uma porta como parte da chave de classificação. Quando você especifica várias portas para a chave de classificação, o Serviço de Integração classifica cada uma delas sequencialmente. A ordem em que as portas aparecem na guia Portas determina a sucessão de operações de classificação. A transformação Classificador trata os dados que passam por cada porta de chave de classificação subsequente como uma classificação secundária da porta anterior.

No tempo de execução da sessão, o Serviço de Integração classifica dados de acordo com a ordem de classificação especificada nas propriedades da sessão. A ordem de classificação determina os critérios de classificação dos caracteres e símbolos especiais.

A figura a seguir mostra a configuração da guia Portas para a transformação Classificador, classificando os dados em ordem crescente por ID do pedido e ID do item:



No tempo de execução da sessão, o Serviço de Integração passa as seguintes linhas para a transformação Classificador:

ORDER_ID	ITEM_ID	QUANTITY	DISCOUNT
45	123456	3	3.04
45	456789	2	12.02
43	000246	6	34.55
41	000468	5	.56

Depois de classificar os dados, o Serviço de Integração retira as seguintes linhas da transformação Classificador:

ORDER_ID	ITEM_ID	QUANTITY	DISCOUNT
41	000468	5	.56
43	000246	6	34.55
45	123456	3	3.04
45	456789	2	12.02

Propriedades da Transformação de Classificador

A transformação de Classificador contém várias propriedades que especificam mais critérios de classificação. O Serviço de Integração aplica esses critérios a todas as principais portas de classificação. As propriedades da transformação de Classificador também determinam os recursos do sistema que o Serviço de Integração aloca ao classificar dados.

Tamanho do Cache do Classificador

O Serviço de Integração utiliza a propriedade Tamanho do Cache do Classificador para determinar a área máxima de memória que ele pode alocar para realizar a operação de classificação. Antes de realizar a operação de classificação, o Serviço de Integração passa todos os dados recebidos para a transformação Classificador. É possível usar um valor numérico para o cache, usar um valor de cache proveniente de um arquivo de parâmetros ou configurar o Serviço de Integração para definir o tamanho do cache usando a configuração Automático. Se você configurar o Serviço de Integração para determinar o tamanho do cache, também poderá especificar a quantidade máxima de memória alocada ao cache pelo Serviço de Integração. Se o tamanho total do cache de sessão configurado for 2 GB (2.147.483.648 bytes) ou mais, você deverá executar a sessão em um Serviço de Integração de 64 bits.

Antes de iniciar a operação de classificação, o Serviço de Integração aloca a área de memória configurada para o tamanho do cache do Classificador. Se o Serviço de Integração executar uma sessão particionada, ele alocará a área especificada da memória do cache do Classificador a cada partição.

Se não conseguir alocar uma área suficiente de memória, o Serviço de Integração interromperá a sessão. Para um melhor desempenho, configure o tamanho do cache do Classificador com um valor menor ou igual à área da RAM física disponível na máquina do Serviço de Integração. Aloque, pelo menos, 16 MB (16.777.216 bytes) de memória física para classificar dados usando a transformação Classificador. Por padrão, o tamanho do cache do Classificador está definido como 16.777.216 bytes.

Se o volume de dados de entrada for maior que o tamanho do cache do Classificador, o Serviço de Integração armazenará temporariamente os dados no diretório de trabalho da transformação de Classificador. Ao armazenar dados no diretório de trabalho, o Serviço de Integração requer que o espaço em disco seja pelo menos duas vezes maior que o volume dos dados de entrada. Se o volume de dados de entrada for muito maior que o tamanho do cache do Classificador, pode ser que o Serviço de Integração precise de bem mais que o dobro de espaço em disco disponível para o diretório de trabalho.

Quando você configura o nível de rastreamento da transformação Classificador como Normal, o Serviço de Integração também grava no log da sessão o volume de memória que essa transformação utiliza.

Distinção Entre Maiúsculas e Minúsculas

A propriedade Distinção Entre Maiúsculas e Minúsculas determina se o Serviço de Integração considera essa distinção ao classificar os dados. Quando você ativa a propriedade Distinção Entre Maiúsculas e Minúsculas, o Serviço de Integração classifica as maiúsculas como sendo superiores às minúsculas.

Diretório de Trabalho

Você deve especificar um diretório de trabalho usado pelo Serviço de Integração para criar arquivos temporários ao classificar dados. Depois de o Serviço de Integração classificar os dados, ele excluirá os arquivos temporários. Você pode especificar qualquer diretório na máquina do Serviço de Integração como diretório de trabalho. Por padrão, o Serviço de Integração usa o valor especificado para a variável de processo \$PMTempDir.

Ao particionar uma sessão com a transformação Classificador, você pode especificar um diretório de trabalho diferente para cada partição no pipeline. Para aumentar o desempenho da sessão, especifique diretórios de trabalho em discos fisicamente separados no sistema do Serviço de Integração.

Linhas de Saída Distintas

Você pode configurar a transformação Classificador para tratar as linhas de saída como distintas. Se você configurar a transformação Classificador para linhas de saída distintas, o Mapping Designer configura todas as portas como parte da tecla de classificação. O Serviço de Integração descarta as linhas duplicadas comparadas durante a operação de ordenação.

Nível de rastreamento

Configure o nível de rastreamento da transformação Classificador para controlar o número e o tipo de mensagens de status e erro do Classificador que o Serviço de Integração gravará no log de sessão. No nível de rastreamento Normal, o Serviço de Integração grava o tamanho da linha passada à transformação Classificador e a quantidade de memória alocada pela transformação Classificador para a operação de classificação. O Serviço de Integração também grava a hora e a data em que passou a primeira e a última linha de entrada à transformação Classificador.

Se você configurar o nível de rastreamento da transformação Classificador para Dados Detalhados, o Serviço de Integração gravará a hora em que a transformação Classificador terminar de passar todos os dados à transformação seguinte no pipeline. O Serviço de Integração também gravará a hora do log de sessão quando a transformação Classificador liberar memória e remover arquivos temporários do diretório de trabalho.

Nulo Tratado como Baixo

É possível configurar a maneira como a transformação Classificação trata os valores nulos. Ative esta propriedade se desejar que o Serviço de Integração trate valores nulos como mais baixos que qualquer outro valor ao efetuar a operação de classificação. Desative esta opção se desejar que o Serviço de Integração trate dos valores nulos como superiores a qualquer outro valor.

Escopo de Transformação

O escopo de transformação especifica como o Serviço de Integração aplica a lógica de transformação aos dados de entrada:

- **Transação.** Aplica a lógica de transformação a todas as linhas de uma transação. Escolha Transação quando uma linha de dados depender de todas as linhas na mesma transação, mas não de linhas em outras transações.
- **Todas as Entradas.** Aplica a lógica de transformação a todos os dados de entrada. Quando você escolhe Todas as Entradas, o PowerCenter descarta os limites de transação de entrada. Escolha Todos de Entrada quando uma linha de dados depender de todas as linhas na origem.

Criando uma Transformação Classificador

Para adicionar uma transformação de Classificador a um mapeamento, conclua as seguintes etapas:

1. No Mapping Designer, clique em Transformação > Criar. Selecione a Transformação Classificador.
A convenção de nomenclatura para as transformações de Classificador é *SRT_TransformationName*. Insira uma descrição para a transformação. Esta descrição aparece no Repository Manager, tornando mais fácil entender o que a transformação faz.
2. Insira um nome para o Classificador e clique em Criar.
O Designer cria a transformação Classificador.
3. Clique em Feito.
4. Arraste as portas que deseja classificar para a transformação Classificador.
O Designer cria as portas de entrada/saída para cada porta que você incluir.
5. Clique duas vezes na barra de título da transformação para abrir a caixa de diálogo Editar transformações.
6. Selecione a guia Portas.
7. Selecione as portas que você deseja usar como chave da classificação.
8. Para cada porta selecionada como parte da chave de classificação, especifique se deseja que o Serviço de Integração classifique os dados em ordem crescente ou decrescente.
9. Selecione a guia Propriedades. Modifique as propriedades da transformação Classificador.
10. Selecione a guia Extensões de metadados. Crie ou edite as extensões de metadados para a transformação Classificador.
11. Clique em OK.

CAPÍTULO 25

Transformação de Qualificador de Origem

Este capítulo inclui os seguintes tópicos:

- [Visão Geral da Transformação de Qualificador de Origem, 411](#)
- [Propriedades da Transformação de Qualificador de Origem, 414](#)
- [Consulta Padrão, 415](#)
- [Unindo Dados de Origem, 416](#)
- [Adicionando uma Consulta SQL, 418](#)
- [Inserindo uma Associação Definida pelo Usuário, 420](#)
- [Suporte à Associação Externa, 420](#)
- [Inserindo um Filtro de Origem, 428](#)
- [Usando Portas Classificadas, 429](#)
- [Selecionar Distintos, 430](#)
- [Adicionando Comandos SQL Pré e Pós-Sessão, 431](#)
- [Criando uma Transformação Qualificador de Origem, 431](#)
- [Solucionando Problemas de Transformações de Qualificador de Origem, 432](#)

Visão Geral da Transformação de Qualificador de Origem

Ao adicionar uma definição de origem de arquivos simples ou relacional a um mapeamento, você deve conectá-la a uma transformação Qualificador de origem. A transformação de Qualificador de Origem representa as linhas lidas pelo Serviço de Integração quando ele executa uma sessão. A transformação de Qualificador de Origem é uma transformação ativa.

Use a transformação Qualificador de origem para concluir estas tarefas:

- **Juntar dados originários do mesmo banco de dados de origem.** Você pode vincular as origens a uma transformação Qualificador de origem para juntar duas tabelas ou mais com relações de chaves primária e estrangeira.
- **Filtrar linhas quando o Serviço de Integração ler os dados de origem.** Se você incluir uma condição de filtro, o Serviço de Integração adicionará uma cláusula WHERE à consulta padrão.

- **Especifique uma associação externa no lugar da associação interna padrão.** Se você incluir uma associação definida pelo usuário, o Serviço de Integração substituirá as informações da associação especificada pelos metadados na consulta SQL.
- **Especificar portas classificadas.** Se você especificar um número de portas classificadas, o Serviço de Integração adicionará uma cláusula ORDER BY à consulta SQL padrão.
- **Selecionar apenas valores distintos da origem.** Se você escolher Selecionar Distintos, o Serviço de Integração adicionará uma instrução SELECT DISTINCT à consulta SQL padrão.
- **Criar uma consulta personalizada para emitir uma instrução SELECT especial para o Serviço de Integração ler os dados de origem.** Por exemplo, você deve usar uma consulta personalizada para realizar cálculos agregados.

Tipos de Dados da Transformação

A transformação Qualificador de origem mostra os tipos de dados da transformação. Os tipos de dados da transformação determinam como o banco de dados de origem liga dados quando o Serviço de Integração os lê. Não altere os tipos de dados na transformação Qualificador de origem. Se os tipos de dados na definição de origem não coincidirem com os da transformação Qualificador de origem, o Designer marcará o mapeamento como inválido quando você o salvar.

Ordem de Carregamento de Destino

Você especifica uma ordem de carregamento de destino nas transformações Qualificador de origem em um mapeamento. Se houver várias transformações Qualificador de origem conectadas a vários destinos, você poderá designar a ordem na qual o Serviço de Integração carregará dados nos destinos.

Se uma transformação Qualificador de origem fornecer dados para vários destinos, você poderá ativar o carregamento com base em restrições em uma sessão para que o Serviço de Integração carregue dados com base em relações de chave estrangeira e primária da tabela de destino.

Por exemplo, um mapeamento contém três pipelines. Cada pipeline contém uma fonte, um qualificador de origem, uma transformação e um destino de arquivo simples separados, que são vinculados em um mapeamento. Você pode configurar a ordem de carregamento de destino para o mapeamento de modo que o Serviço de Integração processe cada grupo de ordem de carregamento de destino sequencialmente.

Valores de data e hora

Quando você utilizar um valor de data e hora ou um parâmetro ou uma variável de data e hora na consulta SQL, altere o formato de data para o formato usado na origem. O Serviço de Integração passa os valores de data e hora aos sistemas de origem na qualidade de strings na consulta ao SQL. O Serviço de Integração converte um valor de data e hora em uma string com base no banco de dados de origem.

A tabela a seguir descreve os formatos de data e hora para cada tipo de banco de dados:

Origem	Formato da data
DB2	YYYY-MM-DD-HH24:MI:SS
Informix	YYYY-MM-DD-HH24:MI:SS
Microsoft SQL Server	MM/DD/YYYY HH24:MI:SS
ODBC	YYYY-MM-DD-HH24:MI:SS

Origem	Formato da data
Oracle	MM/DD/YYYY HH24:MI:SS
Sybase	MM/DD/YYYY HH24:MI:SS
Teradata	YYYY-MM-DD-HH24:MI:SS

Alguns bancos de dados solicitam que você identifique os valores de data e hora com uma pontuação complementar, como aspas simples ou funções específicas de bancos de dados. Por exemplo, para converter o valor \$\$\$SessStartTime para uma origem do Oracle, use a seguinte função do Oracle na substituição SQL:

```
to_date ('$$$SessStartTime', 'mm/dd/yyyy hh24:mi:ss')
```

Para Informix, use a seguinte função do Informix na substituição SQL para converter o valor \$\$\$SessStartTime:

```
DATETIME ($$$SessStartTime) YEAR TO SECOND
```

Para obter informações sobre as funções específicas de bancos de dados, consulte a documentação do banco de dados.

Parâmetros e Variáveis

Você pode usar parâmetros e variáveis na consulta SQL, na associação definida pelo usuário, no filtro de origem e nos comandos de SQL pré- e pós-sessão de uma transformação Qualificador de Origem. Use qualquer tipo de parâmetro ou variável que você possa definir no arquivo de parâmetros. É possível inserir um parâmetro ou variável na instrução SQL, ou é possível usar um parâmetro ou variável na qualidade de consulta SQL. Por exemplo, você pode usar um parâmetro de sessão, \$ParamMyQuery, como consulta SQL, e definir \$ParamMyQuery como a instrução SQL em um arquivo de parâmetro.

Primeiro, o Serviço de Integração gera uma consulta de SQL e expande cada parâmetro ou variável. Ele substitui cada parâmetro e variável de mapeamento, e também a variável de fluxo de trabalho pelo seu valor inicial. Em seguida, executa a consulta no banco de dados de origem.

Ao usar um parâmetro ou uma variável de mapeamento de string na transformação Qualificador de Origem, use um identificador de cadeia de caracteres que seja apropriado ao sistema de origem. A maioria dos bancos de dados usa uma única aspa como identificador de string. Por exemplo, para usar o parâmetro de string \$\$IPAddress em um filtro de origem para uma tabela de banco de dados do Microsoft SQL Server, inclua o parâmetro em aspas simples como segue: '\$\$IPAddress.'

Quando você usa um parâmetro ou variável de mapeamento de datetime, ou a variável incorporada \$\$\$SessStartTime, altere o formato da data para o formato usado na origem. O Serviço de Integração passa os valores de data e hora aos sistema de origem na qualidade de strings na consulta ao SQL.

Sugestão: Para garantir que o formato de um parâmetro ou variável de datetime corresponda ao que é usado pela origem, valide a consulta SQL.

Propriedades da Transformação de Qualificador de Origem

Configure as propriedades gerais da transformação Normalizador na guia Propriedades:

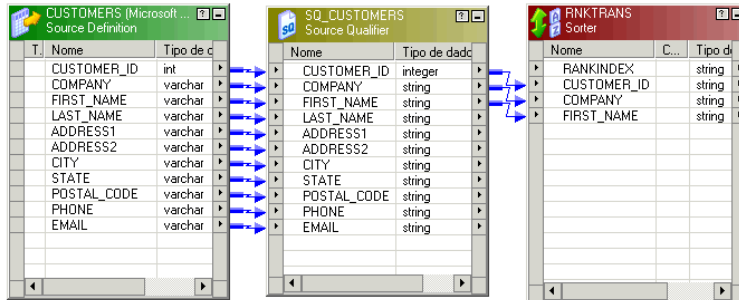
Opção	Descrição
Consulta SQL	Define uma consulta personalizada que substitui a consulta padrão usada pelo Serviço de Integração para ler dados de origens representadas nessa transformação Qualificador de origem. Uma consulta personalizada substitui uma associação personalizada ou um filtro de origem.
Associação definida pelo usuário	Especifica a condição usada para juntar dados de várias origens representadas na mesma transformação Qualificador de origem.
Filtro de origem	Especifica a condição de filtro que o Serviço de Integração aplica ao consultar linhas.
Número de Portas Classificadas	Indica o número de colunas usadas ao classificar linhas consultadas de origens relacionais. Se você selecionar essa opção, o Serviço de Integração adicionará ORDER BY à consulta padrão ao ler linhas da origem. ORDER BY inclui o número de portas especificadas, começando no início da transformação. Quando selecionada, a ordem de classificação do banco de dados coincidirá com a da sessão.
Nível de rastreamento	Define o volume de detalhes incluído no log da sessão quando você executa uma sessão que contém essa transformação.
Selecionar Distintos	Especifica se você quer selecionar apenas linhas exclusivas. O Serviço de Integração incluirá uma instrução SELECT DISTINCT se você escolher essa opção.
Pré-SQL	Comandos SQL pré-sessão a serem executados no banco de dados de origem antes de o Serviço de Integração ler a origem.
Pós-SQL	Comandos SQL pós-sessão a serem executados no banco de dados de origem após o Serviço de Integração gravar no destino.
Saída é determinística	Saída de transformação ou origem relacional que não se altera entre a execução de sessões quando os dados de entrada são consistentes entre elas. Quando você configurar essa propriedade, o Serviço de Integração não preparará a recuperação de dados da origem se as transformações no pipeline sempre produzirem dados repetíveis.
Saída é repetível	Saída de transformação ou origem relacional na mesma ordem entre a execução de sessões quando a ordem dos dados de entrada for consistente. Se a saída for determinística e repetível, o Serviço de Integração não preparará a recuperação de dados.

Aviso: Se você configurar uma transformação como repetível e determinística, será sua responsabilidade garantir que os dados sejam repetíveis e determinísticos. Se você tentar recuperar uma sessão com transformações que não gerem os mesmos dados entre a sessão e a recuperação, o processo de recuperação poderá resultar em dados corrompidos.

Consulta Padrão

Para origens relacionais, o Serviço de Integração gera uma consulta para cada transformação de Qualificador de Origem ao executar uma sessão. A consulta padrão é uma instrução SELECT para cada coluna de origem usada no mapeamento. Em outras palavras, o Serviço de Integração lê apenas as colunas que estão conectadas a outra transformação.

A figura a seguir mostra uma definição de origem única conectada a uma transformação de Qualificador de origem, com muitas colunas de definição de origem, mas somente três conectadas à transformação final:



Embora haja muitas colunas na definição de origem, apenas três colunas são conectadas a outra transformação. Nesse caso, o Serviço de Integração gera uma consulta padrão que seleciona apenas essas três colunas:

```
SELECT CUSTOMERS.CUSTOMER_ID, CUSTOMERS.COMPANY, CUSTOMERS.FIRST_NAME  
  
FROM CUSTOMERS
```

Se algum nome de tabela ou nome de coluna contiver uma palavra reservada para banco de dados, é possível criar e manter um arquivo, o `reswords.txt`, que contenha palavras reservadas. Quando o Serviço de Integração inicializa uma sessão, ele procura `reswords.txt` no seu diretório de instalação. Se o arquivo existir, o Serviço de Integração colocará palavras reservadas coincidentes entre aspas ao executar a SQL no banco de dados. Se você substituir o SQL, é necessário colocar quaisquer palavras reservadas entre aspas.

Ao gerar a consulta padrão, o Designer delimita os nomes de tabela e de campo que contenham os caracteres a seguir com aspas duplas:

```
/ + - = ~ ` ! % ^ & * ( ) [ ] { } ' ; ? , < > \ | <space>
```

Exibindo a Consulta Padrão

Você pode exibir a consulta padrão na transformação Qualificador de origem.

Para exibir a consulta padrão:

1. Na guia Propriedades, selecione Consulta SQL.
O Editor SQL exibe a consulta padrão que o Serviço de Integração utiliza para selecionar dados de origem.
2. Clique em Gerar SQL.
3. Clique em Cancelar para sair.

Se você não cancelar a consulta SQL, o Serviço de Integração substituirá a consulta padrão pela consulta SQL personalizada.

Não se conecte ao banco de dados de origem. Você só pode se conectar ao banco de dados de origem quando inserir uma consulta SQL que substitua a consulta padrão.

Você deve conectar as colunas na transformação Qualificador de origem a outra transformação ou destino antes de poder gerar a consulta padrão.

Substituindo a Consulta Padrão

É possível alterar ou substituir a consulta padrão na transformação do Qualificador de Origem por meio da alteração das configurações padrão das propriedades de transformação. Não altere a lista de portas selecionadas ou a ordem na qual aparecem na consulta. Esta lista deve combinar com as portas de saída de transformação conectadas.

Ao editar as propriedades de transformação, a transformação do Qualificador de origem inclui essas configurações na consulta padrão. Entretanto, se você inserir uma consulta SQL, o Serviço de Integração usará apenas a instrução SQL definida. A Consulta SQL substitui as configurações de Associação Definida pelo Usuário, o Filtro de Origem, o Número de Portas Classificadas e Selecionar Distintos na transformação do Qualificador de Origem.

Nota: Ao substituir a consulta SQL padrão, você deve deixar todas as palavras reservadas do banco de dados entre aspas.

Unindo Dados de Origem

Use uma transformação Qualificador de Origem para juntar os dados de múltiplas tabelas relacionais. Estas tabelas devem ser acessíveis a partir da mesma instância ou do servidor de banco de dados.

Quando um mapeamento usa origens relacionais relacionadas, você pode juntar as duas origens em uma transformação Qualificador de Origem. Durante a sessão, o banco de dados da origem executa a associação antes de transferir os dados para o Serviço de Integração. Isto pode aumentar o desempenho quando as tabelas de origem são indexadas.

Sugestão: Use a transformação Joiner para as origens heterogêneas e para juntar arquivos simples.

Associação Padrão

Quando você associa tabelas relacionadas em uma transformação Qualificador de Origem, o Serviço de Integração associa as tabelas com base nas chaves relacionadas em cada tabela.

Esta associação padrão é um equijoin interno e usa a seguinte sintaxe na cláusula WHERE:

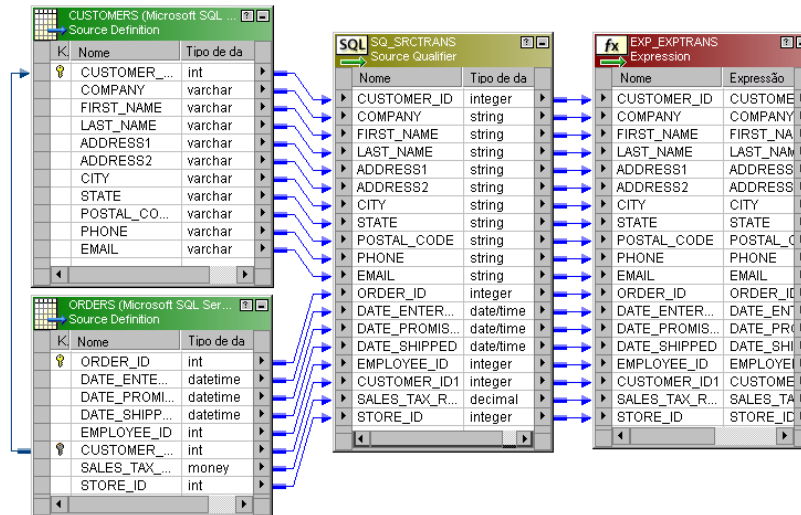
```
Source1.column_name = Source2.column_name
```

As colunas na associação padrão devem ter:

- Um relacionamento de chave primária-chave estrangeira
- Tipos de dados correspondentes

Por exemplo, você pode ver todos os pedidos do mês, inclusive o número e a quantidade do pedido e o nome do cliente. A tabela ORDERS inclui o número do pedido e a quantidade de cada pedido, mas não o nome do cliente. Para incluir o nome do cliente, você precisa associar as tabelas ORDERS e CUSTOMERS. As duas tabelas incluem uma ID de cliente, e assim você pode associa-las em uma transformação Qualificador de Origem.

A figura a seguir mostra a associação de duas tabelas com uma transformação Qualificador de Origem:



Quando você inclui múltiplas tabelas, o Serviço de Integração gera uma instrução SELECT para todas as colunas usadas no mapeamento. Neste caso, a instrução SELECT parece semelhante à instrução a seguir:

```
SELECT CUSTOMERS.CUSTOMER_ID, CUSTOMERS.COMPANY, CUSTOMERS.FIRST NAME,
CUSTOMERS.LAST_NAME, CUSTOMERS.ADDRESS1, CUSTOMERS.ADDRESS2, CUSTOMERS.CITY,
CUSTOMERS.STATE, CUSTOMERS.POSTAL_CODE, CUSTOMERS.PHONE, CUSTOMERS.EMAIL,
ORDERS.ORDER_ID, ORDERS.DATE_ENTERED, ORDERS.DATE PROMISED, ORDERS.DATE SHIPPED,
ORDERS.EMPLOYEE_ID, ORDERS.CUSTOMER_ID, ORDERS.SALES_TAX_RATE, ORDERS.STORE_ID
FROM CUSTOMERS, ORDERS
WHERE CUSTOMERS.CUSTOMER_ID=ORDERS.CUSTOMER_ID
```

A cláusula WHERE é um equijoin que inclui o CUSTOMER_ID das tabelas ORDERS e CUSTOMER.

Associações Personalizadas

Se você precisar substituir a associação padrão, pode inserir conteúdos da cláusula WHERE que especifiquem a associação na consulta personalizada. Se a consulta executar uma associação externa, o Serviço de Integração pode inserir a sintaxe da associação na cláusula WHERE ou FROM, dependendo da sintaxe do banco de dados.

Você pode precisar substituir a associação padrão nas circunstâncias a seguir:

- As colunas não têm um relacionamento de chave primária-chave estrangeira.
- Os tipos de dados das colunas usadas para a associação não correspondem.
- Você deseja especificar um tipo diferente de associação, como uma associação externa.

Associações Heterogêneas

Para executar uma associação heterogênea, use a transformação Joiner. Use a transformação Joiner quando precisar associar os seguintes tipos de origens:

- Associar dados de diferentes bancos de dados de origem
- Associar dados de diferentes sistemas de arquivos simples
- Associar origens relacionais e arquivos simples

Criando Relações de Chaves

É possível associar tabelas na transformação Qualificador de Origem se as tabelas tiverem relações de chave primária/chave estrangeira. Entretanto, é possível criar relações de chave primária/chave estrangeira no Source Analyzer ligando as colunas que combinam em diferentes tabelas. Essas colunas não precisam ser chaves, mas devem ser incluídas no índice para cada tabela.

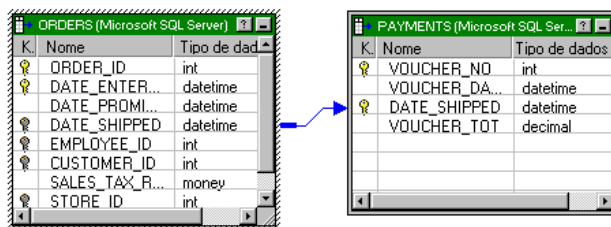
Sugestão: Se a tabela de origem tiver mais de 1.000 linhas, você poderá aumentar o desempenho pela indexação de chave primária/chave estrangeira. Se a tabela de origem tiver menos de 1.000 linhas, é possível diminuir o desempenho pela indexação de chave primária/chave estrangeira.

Por exemplo, o escritório corporativo de uma cadeia de varejo deseja extrair pagamentos recebidos com base em pedidos. As tabelas ORDERS e PAYMENTS não compartilham chaves primárias e estrangeiras. Entretanto, as duas tabelas apresentam uma coluna DATE_SHIPPED. É possível criar uma relação chave primária/chave estrangeira nos metadados do Source Analyzer.

Observe que as duas tabelas não estão conectadas. Logo, o Designer não reconhece a relação nas colunas DATE_SHIPPED.

Cria-se uma relação entre as tabelas ORDERS e PAYMENTS vinculando as colunas DATE_SHIPPED. O Designer adiciona as chaves primárias e estrangeiras às colunas DATE_SHIPPED nas definições de tabela ORDERS e PAYMENTS.

A figura a seguir mostra o relacionamento DATE_SHIPPED entre duas tabelas:



Se você não conectar as colunas, o Designer não reconhece as relações.

As relações de chave primária/chave estrangeira existem apenas nos metadados. Não é necessário gerar SQL ou alterar as tabelas de origem.

Assim que as relações das chaves existirem, use uma transformação Qualificador de Origem para associar as duas tabelas. A associação padrão se baseia em DATE_SHIPPED.

Adicionando uma Consulta SQL

A transformação Qualificador de Origem fornece a opção Consulta de SQL para substituir a consulta padrão. Você pode inserir uma instrução SQL suportada pelo banco de dados de origem. Antes de inserir a consulta, conecte todas as portas de entrada e saída que você deseja usar no mapeamento.

Quando você edita a Consulta SQL, pode gerar e editar a consulta padrão. Quando o Designer gera a consulta padrão, ele incorpora todas as outras opções configuradas, como um filtro ou número de portas classificadas. A consulta resultante substitui todas as outras opções que você pode configurar subsequentemente na transformação.

É possível utilizar um parâmetro ou uma variável como consulta de SQL ou incluir parâmetros e variáveis na consulta. Quando você inclui um parâmetro ou uma variável de mapeamento de string, use um identificador

de string que seja apropriado ao sistema de origem. No caso da maioria dos bancos de dados, é necessário colocar o nome de um parâmetro ou uma variável de string entre aspas simples.

Quando você inclui um valor de data e hora ou um parâmetro/variável de mapeamento de data e hora na consulta de SQL, altere o formato de data para corresponder ao formato usado na origem. O Serviço de Integração converte um valor de data e hora em uma string baseada no sistema de origem.

Use as seguintes regras e diretrizes ao inserir uma consulta SQL personalizada:

- A instrução SELECT deve listar os nomes de porta na ordem em que eles aparecem na transformação.
- Se a origem for o Microsoft SQL Server, o número de colunas na instrução SELECT da consulta deverá corresponder ao número de portas na transformação de Qualificador de Origem. Caso contrário, a sessão pode falhar com o seguinte erro: Erro de SQL [FnName: Buscar Otimizar -- [Informatica][driver de Protocolo ODBC SQL Server Wire] O número de Colunas vinculadas excede o número de colunas de resultados.].

Quando você substitui a consulta SQL padrão para uma sessão configurada para a otimização de empilhamento, o Serviço de Integração criará uma exibição para representar a substituição SQL. Ele executa uma consulta de SQL nesta exibição para forçar a lógica de transformação no banco de dados.

Se você editar a consulta de SQL, é necessário deixar todas as palavras reservadas ao banco de dados entre aspas.

1. Abra a transformação de Qualificador de Origem e clique na guia Propriedades.

2. Clique no botão Abrir no campo Consulta SQL.

A caixa de diálogo Editor de SQL é exibida.

3. Clique em Gerar SQL.

O Designer exibe a consulta padrão que ele gera quando consulta as linhas de todas as origens incluídas na transformação Qualificador de Origem.

4. Insira uma consulta no espaço em que a consulta padrão aparece.

Cada nome de coluna deve ser qualificado conforme o nome da tabela, exibição ou sinônimo no qual aparece. Por exemplo, se você deseja incluir a coluna ORDER_ID da tabela ORDERS, insira ORDERS.ORDER_ID. Você pode clicar duas vezes nos nomes de coluna que aparecem na janela Portas, para evitar digitar o nome de cada coluna.

É possível utilizar um parâmetro ou uma variável como consulta, ou incluir parâmetros e variáveis na consulta.

Inclua os parâmetros e as variáveis de mapeamento de string nos identificadores de string. Quando for necessário, altere o formato de data para os parâmetros e as variáveis de mapeamento de data e hora.

5. Selecione a origem de dados ODBC que contém as origens incluídas na consulta.

6. Insira o nome de usuário e a senha para se conectar a este banco de dados.

A opção Usar Autenticação Kerberos indica que o banco de dados na conexão é executado em uma rede que usa a autenticação Kerberos. Se essa opção for selecionada, você não poderá inserir o nome de usuário e a senha. A conexão usará as credenciais da conta de usuário conectada à máquina em que o Designer é executado.

7. Clique em Validar.

O Designer executa a consulta e informa se a sua sintaxe está correta.

8. Clique em OK para retornar à caixa de diálogo Editar Transformações. Clique em OK novamente para retornar ao Designer.

Dica: Você pode redimensionar o Editor de Expressão. Amplie a caixa de diálogo arrastando a partir das bordas. O Designer salva o novo tamanho da caixa de diálogo na qualidade de configuração de cliente.

Inserindo uma Associação Definida pelo Usuário

Inserir uma associação definida pelo usuário é semelhante a inserir uma consulta personalizada de SQL. Entretanto, só são inseridos os conteúdos da cláusula WHERE, e não toda a consulta. Ao executar uma associação externa, o Serviço de Integração pode inserir a sintaxe de associação na cláusula WHERE ou na cláusula FROM da consulta, dependendo da sintaxe do banco de dados.

Ao adicionar uma associação definida pelo usuário, a transformação de Qualificador de Origem acrescenta a configuração na consulta de SQL padrão. Todavia, se você modificar a consulta padrão após adicionar uma associação definida pelo usuário, o Serviço de Integração utilizará apenas a consulta definida nas propriedades de Consulta de SQL da transformação de Qualificador de Origem.

É possível utilizar um parâmetro ou uma variável como associação definida pelo usuário ou incluir parâmetros e variáveis na associação. Quando você inclui um parâmetro ou uma variável de mapeamento de string, use um identificador de string que seja apropriado ao sistema de origem. No caso da maioria dos bancos de dados, é necessário colocar o nome de um parâmetro ou uma variável de string entre aspas simples.

Quando você inclui um parâmetro ou variável de data e hora, talvez seja necessário alterar o formato da data para corresponder ao formato usado pela origem. O Serviço de Integração converte um valor e uma variável de data e hora em uma string com base no sistema de origem.

Para criar uma associação definida pelo usuário:

1. Crie uma transformação de Qualificador de Origem que contenha dados de várias origens ou origens associadas.
2. Abra a transformação de Qualificador de Origem e clique na guia Propriedades.
3. Clique no botão Abrir no campo Associação Definida pelo Usuário.
A caixa de diálogo Editor de SQL é exibida.
4. Insira a sintaxe para a associação.

Não insira a palavra-chave WHERE no início da associação. O Serviço de Integração acrescenta essa palavra-chave ao consultar as linhas.

Inclua os parâmetros e as variáveis de mapeamento de string nos identificadores de string. Quando for necessário, altere o formato de data para os parâmetros e as variáveis de mapeamento de data e hora.

5. Clique em OK para retornar a Editar Transformações, depois clique em OK para voltar ao Designer.

Suporte à Associação Externa

Use as transformações Qualificador de Origem e Qualificador de Origem do Aplicativo Origem para executar uma associação externa de duas origens no mesmo banco de dados. Quando o Serviço de Integração executa uma associação externa, ele retorna todas as linhas de uma tabela de origem e as linhas da segunda tabela de origem que correspondem à condição de associação.

Use uma associação externa quando deseja unir duas tabelas e retornar todas as linhas a partir de uma das tabelas. Por exemplo, você pode executar uma associação externa quando deseja unir uma tabela de clientes registrados com uma tabela de compras mensais, a fim de determinar a atividade do cliente registrado. Usando uma associação externa, você pode unir a tabela do cliente registrado com a tabela de compras mensais, e retornar todas as linhas na tabela do cliente registrado, inclusive os clientes que não fizeram compras no último mês. Se você executar uma associação normal, o Serviço de Integração retorna

apenas os clientes registrados que fizeram compras durante o mês e somente as compras efetuadas por clientes registrados.

Com uma associação externa, você pode gerar os mesmos resultados que uma associação externa mestre ou detalhada na transformação Joiner. No entanto, quando você usa uma associação externa, reduz o número de linhas no fluxo de dados. Isso pode melhorar o desempenho.

O Serviço de Integração suporta dois tipos de associações externas:

- **À Esquerda.** O Serviço de Integração retorna todas as linhas para a tabela à esquerda da sintaxe da associação e as linhas de ambas as tabelas que cumprem a condição de associação.
- **À Direita.** O Serviço de Integração retorna todas as linhas para a tabela à direita da sintaxe da associação e as linhas de ambas as tabelas que cumprem a condição de associação.

Nota: Use associações externas nas instruções de consulta aninhadas, quando substituir a consulta padrão.

Sintaxe de Associação Informativa

Quando você insere a sintaxe de associação, use a sintaxe específica do Informativa ou do banco de dados. Quando você usa a sintaxe de associação do Informativa, o Serviço de Integração traduz a sintaxe e a transfere para o banco de dados de origem durante a sessão.

Nota: Sempre use uma sintaxe específica do banco de dados para as condições de associação.

Quando você usa a sintaxe de associação do Informativa, inclua a instrução de associação inteira entre chaves ({sintaxe do Informativa}). Quando você usa a sintaxe do banco de dados, insira a sintaxe suportada pelo banco de dados de origem, sem chaves.

Quando usar a sintaxe de associação do Informativa, use os nomes de tabela como prefixos dos nomes de coluna. Por exemplo, se você tem uma coluna nomeada FIRST_NAME na tabela REG_CUSTOMER, insira "REG_CUSTOMER.FIRST_NAME" na sintaxe de associação. Além disso, quando usar um alias para um nome de tabela, use o alias dentro da sintaxe de associação do Informativa para garantir que o Serviço de Integração o reconheça.

A tabela a seguir lista a sintaxe de associação que você pode inserir, em diferentes locais para diferentes transformações Qualificador de Origem, ao criar uma associação externa:

Transformação	Configuração de transformação	Descrição
Transformação Qualificador de Origem	Associação definida pelo usuário	Crie uma substituição de associação. O Serviço de Integração anexa a substituição de associação à cláusula WHERE ou FROM da consulta padrão.
Transformação Qualificador de Origem	Consulta SQL	Insira a sintaxe de associação imediatamente de WHERE na consulta padrão.
Transformação Qualificador de Origem de Aplicativo	Substituição de associação	Crie uma substituição de associação. O Serviço de Integração anexa a substituição de associação à cláusula WHERE da consulta padrão.
Transformação Qualificador de Origem de Aplicativo	Substituição de extração	Insira a sintaxe de associação imediatamente de WHERE na consulta padrão.

Você pode combinar associações externas direitas e esquerdas com associações normais em um único qualificador de origem. Use múltiplas associações normais e externas esquerdas.

Quando for combinar associações, insira-as na ordem a seguir:

1. Normal
2. Exterior à esquerda
3. Exterior à direita

Nota: Alguns bancos de dados o limitam a usar uma associação externa direita.

Sintaxe de Associação Normal

É possível criar uma associação normal usando a condição de associação em um qualificador de origem. Entretanto, se você estiver criando uma associação externa, será necessário substituir a associação padrão para efetuar uma associação externa. Consequentemente, será necessário colocar a associação normal na substituição de associação. Ao incorporar uma associação normal na substituição de associação, liste a associação normal antes das associações externas. É possível inserir várias associações normais na substituição de associação.

Para criar uma associação normal, use a seguinte sintaxe:

```
{ source1 INNER JOIN source2 on join_condition }
```

A tabela a seguir exibe a sintaxe para Associações Normais em Substituição de Associação:

Sintaxe	Descrição
<i>source1</i>	Nome da tabela de origem. O Serviço de Integração retorna linhas dessa tabela que correspondam à condição de associação.
<i>source2</i>	Nome da tabela de origem. O Serviço de Integração retorna linhas dessa tabela que correspondam à condição de associação.
<i>join_condition</i>	Condição para a associação. Use sintaxe compatível com o banco de dados de origem. Você pode combinar diversas condições de associação com um operador AND.

Por exemplo, você tem uma tabela REG_CUSTOMER com dados de clientes registrados:

CUST_ID	FIRST_NAME	LAST_NAME
00001	Marvin	Chi
00002	Dinah	Jones
00003	John	Bowden
00004	J.	Marks

A tabela PURCHASES, atualizada mensalmente, contém os seguintes dados:

TRANSACTION_NO	CUST_ID	DATE	AMOUNT
06-2000-0001	00002	6/3/2000	55.79
06-2000-0002	00002	6/10/2000	104.45
06-2000-0003	00001	6/10/2000	255.56
06-2000-0004	00004	6/15/2000	534.95

TRANSACTION_NO	CUST_ID	DATE	AMOUNT
06-2000-0005	00002	6/21/2000	98.65
06-2000-0006	NULL	6/23/2000	155.65
06-2000-0007	NULL	6/24/2000	325.45

Para retorna linhas que estão exibindo nomes de clientes para cada transação no mês de junho, use a sintaxe a seguir:

```
{ REG_CUSTOMER INNER JOIN PURCHASES on REG_CUSTOMER.CUST_ID = PURCHASES.CUST_ID }
```

O Serviço de Integração retornará os seguintes dados:

CUST_ID	DATE	AMOUNT	FIRST_NAME	LAST_NAME
00002	6/3/2000	55.79	Dinah	Jones
00002	6/10/2000	104.45	Dinah	Jones
00001	6/10/2000	255.56	Marvin	Chi
00004	6/15/2000	534.95	J.	Marks
00002	6/21/2000	98.65	Dinah	Jones

O Serviço de Integração retorna linhas com IDs de clientes correspondentes. Não estão incluídos clientes que não efetuaram compras em junho. Também não são incluídas compras efetuadas por clientes não registrados.

Sintaxe de Associação Externa Esquerda

Você pode criar uma associação externa esquerda com uma substituição de associação. Você pode inserir múltiplas associações externas esquerdas em uma única substituição de associação. Quando usar associações externas esquerdas com outras associações, liste todas as associações externas esquerdas em conjunto, depois de qualquer associação normal na instrução.

Para criar uma associação externa esquerda, use a seguinte sintaxe:

```
{ source1 LEFT OUTER JOIN source2 on join_condition }
```

As tabelas a seguir mostram a sintaxe de associações externas esquerdas em uma substituição de associação:

Sintaxe	Descrição
<i>source1</i>	Nome da tabela de origem. Com uma associação externa esquerda, o Serviço de Integração retorna todas as linhas da tabela.
<i>source2</i>	Nome da tabela de origem. O Serviço de Integração retorna linhas dessa tabela que correspondam à condição de associação.
<i>join_condition</i>	Condição para a associação. Use sintaxe compatível com o banco de dados de origem. Você pode combinar diversas condições de associação com um operador AND.

Por exemplo, usando as mesmas tabelas REG_CUSTOMER e PURCHASES descritas em [“Sintaxe de Associação Normal” na página 422](#), você pode determinar quantos clientes compraram algo em junho com a seguinte substituição de associação:

```
{ REG_CUSTOMER LEFT OUTER JOIN PURCHASES on REG_CUSTOMER.CUST_ID = PURCHASES.CUST_ID }
```

O Serviço de Integração retornará os seguintes dados:

CUST_ID	FIRST_NAME	LAST_NAME	DATE	AMOUNT
00001	Marvin	Chi	6/10/2000	255.56
00002	Dinah	Jones	6/3/2000	55.79
00003	John	Bowden	NULL	NULL
00004	J.	Marks	6/15/2000	534.95
00002	Dinah	Jones	6/10/2000	104.45
00002	Dinah	Jones	6/21/2000	98.65

O Serviço de Integração retorna todos os clientes registrados na tabela REG_CUSTOMERS, usando valores nulos para o cliente que não fez nenhuma compra em junho. Não são incluídas compras efetuadas por clientes não registrados.

Use múltiplas condições de associação para determinar quantos clientes registrados gastaram mais que US \$100.00 em uma única compra em junho:

```
{REG_CUSTOMER LEFT OUTER JOIN PURCHASES on (REG_CUSTOMER.CUST_ID = PURCHASES.CUST_ID AND PURCHASES.AMOUNT > 100.00) }
```

O Serviço de Integração retornará os seguintes dados:

CUST_ID	FIRST_NAME	LAST_NAME	DATE	AMOUNT
00001	Marvin	Chi	6/10/2000	255.56
00002	Dinah	Jones	6/10/2000	104.45
00003	John	Bowden	NULL	NULL
00004	J.	Marks	6/15/2000	534.95

Você pode usar múltiplas associações externas esquerdas se deseja incorporar informações sobre as devoluções durante o mesmo período. Por exemplo, a tabela RETURNS contém os seguintes dados:

CUST_ID	CUST_ID	RETURN
00002	6/10/2000	55.79
00002	6/21/2000	104.45

Para determinar quantos clientes fizeram compras e devoluções durante o mês de junho, use duas associações externas esquerdas:

```
{ REG_CUSTOMER LEFT OUTER JOIN PURCHASES on REG_CUSTOMER.CUST_ID = PURCHASES.CUST_ID LEFT OUTER JOIN RETURNS on REG_CUSTOMER.CUST_ID = RETURNS.CUST_ID }
```


O Serviço de Integração retornará os seguintes dados:

CUST_ID	FIRST_NAME	LAST_NAME	DATE	AMOUNT	RET_DATE	RETURN
00001	Marvin	Chi	6/10/2000	255.56	NULL	NULL
00002	Dinah	Jones	6/3/2000	55.79	NULL	NULL
00003	John	Bowden	NULL	NULL	NULL	NULL
00004	J.	Marks	6/15/2000	534.95	NULL	NULL
00002	Dinah	Jones	6/10/2000	104.45	NULL	NULL
00002	Dinah	Jones	6/21/2000	98.65	NULL	NULL
00002	Dinah	Jones	NULL	NULL	6/10/2000	55.79
00002	Dinah	Jones	NULL	NULL	6/21/2000	104.45

O Serviço de Integração usa NULLs para valores ausentes.

Sintaxe de Associação Externa Direita

Você pode criar uma associação externa direita com uma substituição de associação. A associação externa direita retornará os mesmos resultados de uma associação externa esquerda se você inverter a ordem das tabelas na sintaxe de associação. Use apenas uma associação externa direita em uma substituição de associação. Se você quiser criar mais de uma associação externa direita, experimente inverter a ordem das tabelas de origem e alterar os tipos de associação para associações externas esquerdas.

Ao usar uma associação externa direita com outras associações, insira-a no final da substituição de associação.

Para criar uma associação externa direita, use a seguinte sintaxe:

```
{ source1 RIGHT OUTER JOIN source2 on join_condition }
```

A tabela a seguir mostra a sintaxe de uma associação externa direita em uma substituição de associação:

Sintaxe	Descrição
<i>source1</i>	Nome da tabela de origem. O Serviço de Integração retorna linhas dessa tabela que correspondam à condição de associação.
<i>source2</i>	Nome da tabela de origem. Com uma associação externa direita, o Serviço de Integração retorna todas as linhas da tabela.
<i>join_condition</i>	Condição para a associação. Use sintaxe compatível com o banco de dados de origem. Você pode combinar diversas condições de associação com um operador AND.

Você pode usar uma associação externa direita com uma associação externa esquerda e retornar todos os dados de ambas as tabelas, simulando uma associação externa completa. Por exemplo, você pode extrair todos os clientes registrados e todas as compras do mês de junho com a seguinte substituição de associação:

```
{REG_CUSTOMER LEFT OUTER JOIN PURCHASES on REG_CUSTOMER.CUST_ID = PURCHASES.CUST_ID  
RIGHT OUTER JOIN PURCHASES on REG_CUSTOMER.CUST_ID = PURCHASES.CUST_ID }
```

O Serviço de Integração retornará os seguintes dados:

CUST_ID	FIRST_NAME	LAST_NAME	TRANSACTION_NO	DATE	AMOUNT
00001	Marvin	Chi	06-2000-0003	6/10/2000	255.56
00002	Dinah	Jones	06-2000-0001	6/3/2000	55.79
00003	John	Bowden	NULL	NULL	NULL
00004	J.	Marks	06-2000-0004	6/15/2000	534.95
00002	Dinah	Jones	06-2000-0002	6/10/2000	104.45
00002	Dinah	Jones	06-2000-0005	6/21/2000	98.65
NULL	NULL	NULL	06-2000-0006	6/23/2000	155.65
NULL	NULL	NULL	06-2000-0007	6/24/2000	325.45

Criando uma Associação Externa

É possível inserir uma associação externa na qualidade de substituição de associação ou como parte de uma substituição da consulta padrão.

Ao criar uma substituição de associação, o Designer anexa a substituição de associação à cláusula WHERE da consulta padrão. Durante a sessão, o Serviço de Integração converte a sintaxe de associação da Informatica e a inclui na consulta padrão usada para extrair dados de origem. Quando for possível, insira uma substituição de associação ao invés de substituir a consulta padrão.

Ao substituir a consulta padrão, insira a sintaxe de associação na cláusula WHERE da consulta padrão. Durante a sessão, o Serviço de Integração converte a sintaxe de associação da Informatica e usa a consulta para extrair dados de origem. Se efetuar alterações na transformação após criar a substituição, o Serviço de Integração ignora as alterações. Assim, quando for possível, insira a sintaxe de associação externa na qualidade de substituição de associação.

Criando uma Associação Externa como uma Substituição de Associação

Para uma associação externa na qualidade de substituição de associação:

1. Abra a transformação Qualificador de Origem e clique na guia Propriedades.
2. Em uma transformação Qualificador de Origem, clique no botão do campo Associação Definida pelo Usuário.

Em uma transformação Qualificador de Origem de Aplicativo, clique no botão no campo Substituição de Associação.
3. Insira a sintaxe para a associação.

Não insira WHERE no início da associação. O Serviço de Integração acrescenta essa expressão ao consultar as linhas.

Coloque a sintaxe de associação da Informatica entre chaves ({ }).

Ao usar um alias para uma tabela e para a sintaxe de associação da Informatica, use o alias dentro da sintaxe de associação da Informatica.

Use nomes de tabelas na qualidade de prefixos a nomes de colunas seguindo o exemplo "tabela.coluna".

Use as condições de união compatíveis com o banco de dados de origem.

Ao inserir várias associações, agrupe as associações segundo o tipo e, em seguida, liste-as de acordo com a ordem a seguir: normal, externa à esquerda, externa à direita. Acrescente apenas uma associação externa à direita por consulta aninhada.

Selecione os nomes de porta a partir da guia Portas, para garantir precisão.

4. Clique em OK.

Criando uma Associação Externa como uma Substituição de Extração

Para criar uma associação externa como uma substituição de extração:

1. Depois de conectar as portas de entrada e saída para a transformação Qualificador da Origem do Aplicativo, clique duas vezes na barra de título da transformação e selecione a guia Propriedades.
2. Em uma transformação Qualificador de Origem de Aplicativo, clique no botão no campo Substituição de Extração.
3. Clique em Gerar SQL.
4. Insira a sintaxe para a associação na cláusula WHERE, imediatamente depois do WHERE.

Coloque a sintaxe de associação da Informatica entre chaves ({}).

Ao usar um alias para uma tabela e para a sintaxe de associação da Informatica, use o alias dentro da sintaxe de associação da Informatica.

Use nomes de tabelas na qualidade de prefixos a nomes de colunas seguindo o exemplo "tabela.coluna".

Use as condições de união compatíveis com o banco de dados de origem.

Ao inserir várias associações, agrupe as associações segundo o tipo e, em seguida, liste-as de acordo com a ordem a seguir: normal, externa à esquerda, externa à direita. Acrescente apenas uma associação externa à direita por consulta aninhada.

Selecione os nomes de porta a partir da guia Portas, para garantir precisão.

5. Clique em OK.

Restrições Comuns de Sintaxe de Banco de Dados

Bancos de dados diferentes possuem restrições diferentes na sintaxe da associação externa. Considere as seguintes restrições quando for criar associações externas:

- Não combine condições de associação com o operador OR na cláusula IN da sintaxe da associação externa.
- Não use o operador IN para comparar colunas na cláusula IN da sintaxe da associação externa.
- Não compare uma coluna a uma sub-consulta na cláusula IN da sintaxe da associação externa.
- Ao combinar duas ou mais associações externas, não use a mesma tabela como a tabela interna de mais de uma associação externa. Por exemplo, não use uma das seguintes associações externas:

```
{ TABLE1 LEFT OUTER JOIN TABLE2 ON TABLE1.COLUMN1 = TABLE2.COLUMN1 TABLE3 LEFT OUTER  
JOIN TABLE2 ON TABLE3.COLUMN2 = TABLE2.COLUMN2 }  
{ TABLE1 LEFT OUTER JOIN TABLE2 ON TABLE1.COLUMN1 = TABLE2.COLUMN1 TABLE2 RIGHT OUTER  
JOIN TABLE3 ON TABLE2.COLUMN2 = TABLE3.COLUMN2 }
```

- Não use as duas tabelas de uma associação externa em uma condição de associação regular. Por exemplo, não use a seguinte condição de associação:

```
{ TABLE1 LEFT OUTER JOIN TABLE2 ON TABLE1.COLUMN1 = TABLE2.COLUMN1 WHERE  
TABLE1.COLUMN2 = TABLE2.COLUMN2 }
```

No entanto, use as duas tabelas em uma condição de filtro, como a seguinte:

```
{ TABLE1 LEFT OUTER JOIN TABLE2 ON TABLE1.COLUMNA = TABLE2.COLUMNB WHERE  
TABLE1.COLUMNB = 32 AND TABLE2.COLUMNC > 0 }
```

Nota: Inserir uma condição na cláusula IN pode retornar resultados diferentes da inserção da mesma condição na cláusula WHERE.

- Ao usar um alias para uma tabela, use-o como prefixo das colunas da tabela. Por exemplo, se você chamar REG_CUSTOMER de tabela C, ao referenciar a coluna FIRST_NAME, use "C.FIRST_NAME."

Inserindo um Filtro de Origem

É possível inserir um filtro de origem para reduzir o número de linhas consultadas pelo Serviço de Integração. Se a string 'WHERE' ou objetos grandes forem incluídos no filtro de origem, o Serviço de Integração não obtém êxito na sessão.

Se você adicionar um filtro de origem à transformação de Qualificador de Origem no mapeamento, a consulta SQL padrão conterá a condição de filtro. Todavia, se você modificar a consulta padrão após adicionar um filtro de origem, o Serviço de Integração usa apenas a consulta definida na porção de consulta SQL da transformação do Qualificador de Origem.

É possível utilizar um parâmetro ou uma variável como filtro de origem ou incluir parâmetros e variáveis no filtro de origem. Quando você inclui um parâmetro ou uma variável de mapeamento de string, use um identificador de string que seja apropriado ao sistema de origem. No caso da maioria dos bancos de dados, é necessário colocar o nome de um parâmetro ou uma variável de string entre aspas simples.

Quando você inclui um parâmetro ou variável de data e hora, talvez seja necessário alterar o formato da data para corresponder ao formato usado pela origem. O Serviço de Integração converte um valor e uma variável de data e hora em uma string com base no sistema de origem.

Nota: Quando você insere uma consulta SQL nas propriedades da sessão, você substitui a condição de filtro e a consulta SQL no nível de mapeamento.

Para inserir um filtro de origem:

1. No Mapping Designer, abra a transformação de Qualificador de Origem.

A caixa de diálogo Editar Transformação aparecerá.

2. Selecione a guia Propriedades.
3. Clique no botão Abrir no campo de Filtro de Origem.
4. Na caixa de diálogo Editor de SQL, insira o filtro.

Acrescente os nomes da tabela e da porta. Não acrescente a palavra-chave WHERE no filtro.

Inclua os parâmetros e as variáveis de mapeamento de string nos identificadores de string. Quando for necessário, altere o formato de data para os parâmetros e as variáveis de mapeamento de data e hora.

5. Clique em OK.

Usando Portas Classificadas

Quando você usa portas classificadas, o Serviço de Integração adiciona as portas à cláusula ORDER BY na consulta padrão. O Serviço de Integração adiciona o número de portas configurado, começando pelo início da transformação Qualificador de origem. Se um subconjunto de portas estiver conectado à porta downstream, a consulta padrão incluirá somente o subconjunto de portas. As portas classificadas são aplicadas nas portas conectadas em vez das portas que iniciam na parte superior da transformação do Qualificador de Fonte.

Você pode usar portas classificadas para melhorar o desempenho ao incluir qualquer uma das transformações a seguir em um mapeamento:

- **Agregador.** Quando configura uma transformação Agregador para uma entrada classificada, você pode enviar dados classificados usando portas classificadas. O agrupamento por portas na transformação Agregador deve corresponder à ordem das portas classificadas na transformação Qualificador de origem.
- **Joiner.** Ao configurar uma transformação Joiner para entrada classificada, você pode enviar dados classificados usando portas classificadas. Configure a ordem das portas classificadas da mesma maneira em toda transformação Qualificador de origem.

Nota: Você também pode usar a transformação Classificador para classificar dados de arquivos simples e relacionais antes das transformações Agregador e Joiner.

Utilize portas classificadas somente para origens relacionais. Ao usar portas classificadas, a ordem de classificação do banco de dados de origem deve corresponder à ordem configurada para a sessão. O Serviço de Integração cria a consulta SQL usada para extrair dados de origem, inclusive a cláusula ORDER BY para portas classificadas. O servidor de banco de dados executa a consulta e passa os dados resultantes para o Serviço de Integração. Para assegurar que os dados sejam classificados conforme o Serviço de Integração exige, a ordem de classificação do banco de dados deve ser a mesma que a ordem de classificação da sessão definida pelo usuário.

Quando você configura o Serviço de Integração para validação da página de código dos dados e executa um fluxo de trabalho no modo de movimento de dados Unicode, o Serviço de Integração utiliza a ordem de classificação selecionada para classificar dados de caractere.

Quando você configura o Serviço de Integração para validação da página de código de dados reduzida, o Serviço de Integração utiliza a ordem de classificação selecionada para classificar todos os dados de caractere que se enquadram no intervalo de idiomas da ordem de classificação selecionada. O Serviço de Integração classifica todos os dados de caractere fora do intervalo de idiomas da ordem de classificação selecionada, de acordo com a ordem de classificação Unicode padrão.

Quando o Serviço de Integração é executado no modo ASCII, ele ignora essa configuração e classifica todos os dados de caractere usando uma ordem de classificação binária. A ordem de classificação padrão varia na página de código do Serviço de Integração.

A transformação Qualificador de origem inclui o número de portas classificadas na consulta SQL padrão. No entanto, se você modificar a consulta padrão depois de escolher o número de portas classificadas, o Serviço de Integração usará somente a consulta definida na propriedade Consulta SQL.

Para usar portas classificadas:

1. No Mapping Designer, abra uma transformação Qualificador de origem e clique na guia Propriedades.
2. Clique em Número de Portas Classificadas e insira o número de portas que deseja classificar.

O Serviço de Integração adiciona o número de colunas configurado para uma cláusula ORDER BY, começando pelo início da transformação Qualificador de origem.

A ordem de classificação do banco de dados de origem deve corresponder à ordem de classificação da sessão.

Sugestão: O Sybase suporta, no máximo, 16 colunas em uma cláusula ORDER BY. Se a origem for Sybase, não classifique mais de 16 colunas.

3. Clique em OK.

Selecionar Distintos

Se você quiser que o Serviço de Integração selecione valores exclusivos de uma origem, use a opção Selecionar Distintos. Você pode usar esse recurso para extrair IDs exclusivas de clientes de uma tabela de vendas totais. O uso de Selecionar Distintos filtra dados desnecessários mais cedo no fluxo de trabalho, o que pode aumentar o desempenho.

Por padrão, o Designer gera uma instrução SELECT. Se você escolher Selecionar Distintos, a transformação Qualificador de origem incluirá a configuração à consulta SQL padrão.

Por exemplo, na transformação Qualificador de origem em [“Unindo Dados de Origem” na página 416](#), você ativa a opção Selecionar Distintos. O Designer adiciona SELECT DISTINCT à consulta padrão, como a seguir:

```
SELECT DISTINCT CUSTOMERS.CUSTOMER_ID, CUSTOMERS.COMPANY, CUSTOMERS.FIRST_NAME,
CUSTOMERS.LAST_NAME, CUSTOMERS.ADDRESS1, CUSTOMERS.ADDRESS2, CUSTOMERS.CITY,
CUSTOMERS.STATE, CUSTOMERS.POSTAL_CODE, CUSTOMERS.EMAIL, ORDERS.ORDER_ID,
ORDERS.DATE_ENTERED, ORDERS.DATE_PROMISED, ORDERS.DATE_SHIPPED, ORDERS.EMPLOYEE_ID,
ORDERS.CUSTOMER_ID, ORDERS.SALES_TAX_RATE, ORDERS.STORE_ID
FROM
CUSTOMERS, ORDERS
WHERE
CUSTOMERS.CUSTOMER_ID=ORDERS.CUSTOMER_ID
```

No entanto, se você modificar a consulta padrão depois de escolher a opção Selecionar Distintos, o Serviço de Integração usará somente a consulta definida na propriedade Consulta SQL. Ou seja, a consulta SQL substitui a configuração Selecionar Distintos.

Para usar a opção Selecionar Distintos:

1. Abra a transformação Qualificador de origem no mapeamento e clique na guia Propriedades.
2. Marque Selecionar Distintos e clique em OK.

Substituindo Selecionar Distintos na Sessão

É possível substituir a opção de nível de transformação por Selecionar Distintos ao configurar a sessão no Workflow Manager.

Para substituir a opção Selecionar Distintos:

1. No Workflow Manager, abra a tarefa Sessão e clique na guia Mapeamento.
2. Clique na visualização Transformações e clique em Qualificador de origem no nó Origens.
3. Nas configurações de Propriedades, ative Selecionar Distintos, depois clique em OK.

Adicionando Comandos SQL Pré e Pós-Sessão

Você pode adicionar comandos de SQL pré e pós-sessão na guia Propriedades da transformação Qualificador de Origem. Você pode usar um SQL pré-sessão para escrever uma linha de carimbo de data e hora na tabela de origem, quando uma sessão inicia.

O Serviço de Integração executa comandos de SQL pré-sessão contra o banco de dados de origem, antes de ler a origem. Ele executa comandos de SQL pós-sessão contra o banco de dados de origem, depois de escrever no destino.

Você pode anular os comandos de SQL na exibição Transformações na guia Mapeamento, nas propriedades da sessão. Você também pode configurar o Serviço de Integração para interromper ou continuar quando encontra erros ao executar os comandos de SQL pré ou pós-sessão.

Use as seguintes diretrizes quando inserir comandos de SQL pré e pós-sessão na transformação Qualificador de Origem:

- Use qualquer comando que seja válido para o tipo de banco de dados. No entanto, o Serviço de Integração não permite comentários aninhados, embora o banco de dados possa permitir.
- Você pode usar parâmetros e variáveis nos comandos de SQL pré e pós-sessão na origem, ou pode usar um parâmetro ou variável como comando. Use qualquer tipo de parâmetro ou variável que você possa definir no arquivo de parâmetros.
- Use um ponto-e-vírgula (;) para separar múltiplas instruções. O Serviço de Integração emite uma confirmação depois de cada instrução.
- O Serviço de Integração ignora o ponto-e-vírgula dentro de /*...*/.
- Se você precisa usar um ponto-e-vírgula fora dos comentários, pode usar o escape da barra invertida (\). Quando você usa o escape do ponto-e-vírgula, o Serviço de Integração ignora a barra invertida e não usa o ponto-e-vírgula como um separador de instrução.
- O Designer não valida o SQL.

Nota: Você também pode inserir comandos de SQL pré e pós-sessão na guia Propriedades da instância de destino em um mapeamento.

Criando uma Transformação Qualificador de Origem

Você pode configurar o Designer para criar uma Transformação Qualificador de Origem por padrão, ao arrastar uma origem para um mapeamento, ou pode criar uma Transformação Qualificador de Origem manualmente.

Criando uma Transformação Qualificador de Origem Manualmente

Você pode criar manualmente uma transformação Qualificador de Origem no Mapping Designer.

Para criar uma transformação do Qualificador de Origem manualmente:

1. No Mapping Designer, clique em Transformação > Criar.
2. Insira um nome para a transformação e clique em Criar.
3. Selecione a origem e clique em OK.
4. Clique em Concluído.

Configurando Opções da Transformação Qualificador de Origem

Depois de criar a transformação Qualificador de Origem, você pode configurar várias opções.

Para configurar uma transformação do Qualificador de Origem:

1. No Designer, abra um mapeamento.
2. Clique duas vezes na barra de títulos da transformação do Qualificador de Origem.
3. Na caixa de diálogo Editar Transformações, clique em Renomear, insira um nome descritivo para a transformação e clique em OK.
A convenção de nomenclatura para as transformações Qualificador de Origem é `SQ_TransformationName`, como `SQ_AllSources`.
4. Clique na guia Propriedades.
5. Insira as propriedades da transformação do Qualificador de Origem.
6. Clique na guia Origens e indique qualquer definição da origem associada, que você deseja definir para esta transformação.
Identifique as origens associadas apenas quando precisar juntar dados de múltiplos bancos de dados ou sistemas de arquivo simples.
7. Clique em OK.

Solucionando Problemas de Transformações de Qualificador de Origem

Não consigo realizar uma operação de arrastar e soltar, como a conexão de portas.

Revise a mensagem de erro na barra de status para obter detalhes.

Não consigo conectar uma definição de origem a uma definição de destino.

Não é possível conectar origens a destinos diretamente. Em vez disso, conecte-os através de uma transformação Qualificador de Origem para origens de arquivos simples e relacionais ou através de uma transformação Normalizador para origens COBOL.

Não consigo conectar várias origens a um destino.

O Designer não permite conectar várias transformações Qualificador de origem a um único destino. Existem duas soluções:

- **Reutilizar destinos.** Como definições de destino são reutilizáveis, você pode adicionar o mesmo destino várias vezes ao mapeamento. Conecte cada transformação Qualificador de origem a cada destino.
- **Una as origens em uma transformação de Qualificador de Origem.** Remova a cláusula WHERE da consulta SQL.

A origem tem valores QNAN (não é um número) em algumas colunas, mas o destino mostra 1.#QNAN.

Sistemas operacionais têm diferentes representações de string de NaN. O Serviço de Integração converte valores QNAN para 1.#QNAN em plataformas Win64EMT. 1.#QNAN é uma representação válida de QNAN.

Eu inseri uma consulta personalizada, mas ela não funciona quando executo o fluxo de trabalho que contém a sessão.

Não deixe de testar essa configuração para a transformação Qualificador de origem antes de executar o fluxo de trabalho. Retorne à transformação Qualificador de origem e abra novamente a caixa de diálogo em que você inseriu a consulta personalizada. Você pode se conectar a um banco de dados e clicar no botão Validar para testar o SQL. O Designer mostrará se há erros. Revise o arquivo de log da sessão se precisar de mais informações.

O motivo mais comum para encerramento de sessões é que o logon do banco de dados na sessão e na transformação Qualificador de origem não é o proprietário da tabela. Você deve especificar o proprietário da tabela na sessão e quando gerar a consulta SQL na transformação Qualificador de origem.

Você pode cortar e colar a consulta SQL na ferramenta do cliente do banco de dados (como o Oracle Net) para ver se ela retorna erros.

Eu usei uma variável de mapeamento em um filtro de origem e agora a sessão é encerrada.

Experimente gerar e validar o SQL na transformação Qualificador de origem para testar a consulta. Se o parâmetro ou variável for uma string, provavelmente você deva colocá-la entre aspas simples. Se for um parâmetro ou variável de data e hora, talvez você deva alterar o formato para o sistema da origem.

CAPÍTULO 26

Transformação SQL

Este capítulo inclui os seguintes tópicos:

- [Visão Geral da Transformação SQL, 434](#)
- [Modo de Script, 435](#)
- [Modo de Consulta, 436](#)
- [Conectando com Bancos de Dados, 442](#)
- [Processamento de Sessão, 446](#)
- [Cardinalidade de linha de entrada para linha de saída, 449](#)
- [Propriedades da Transformação SQL, 453](#)
- [Instruções SQL, 457](#)
- [Criação de uma Transformação SQL, 458](#)

Visão Geral da Transformação SQL

A transformação SQL processa consultas SQL midstream em um pipeline. A transformação SQL pode ser uma transformação ativa ou passiva. Você pode inserir, excluir, atualizar e recuperar linhas de um banco de dados. Você pode passar as informações sobre conexão a banco de dados para a transformação SQL como dados de entrada em tempo de execução. A transformação processa scripts SQL externos ou consultas SQL que você cria em um editor SQL. A transformação SQL processa a consulta e retorna erros de banco de dados e linhas.

Por exemplo, talvez você precise criar tabelas do banco de dados antes de adicionar novas transações. Você pode criar uma transformação SQL para criar as tabelas em um fluxo de trabalho. A transformação SQL retorna erros do banco de dados em uma porta de saída. Você pode configurar outro fluxo de trabalho para ser executado, caso a transformação SQL não retorne erros.

Ao criar uma transformação SQL, você configura as seguintes opções:

- **Modo.** A transformação SQL é executada em um destes modos:
 - **Modo de script.** A transformação SQL executa scripts ANSI SQL com localização externa. Você passa à transformação o nome de um script a cada linha de entrada. A transformação SQL retorna uma linha para cada linha de entrada.
 - **Modo de consulta.** A transformação SQL executa uma consulta definida em um editor de consultas. Você pode passar strings ou parâmetros à consulta, para definir consultas dinâmicas ou alterar os parâmetros de seleção. Você poderá retornar várias linhas se a consulta tiver uma instrução SELECT.

- **Transformação passiva ou ativa.** A transformação SQL é uma transformação ativa por padrão. Ao criá-la, você pode configurá-la como uma transformação passiva.
- **Tipo de banco de dados:** O tipo de banco de dados ao qual a transformação SQL se conecta.
- **Tipo de conexão.** Passe informações de conexão do banco de dados à transformação SQL ou use um objeto de conexão.

Modo de Script

Uma transformação SQL executada em modo de script executa scripts SQL de arquivos de texto. Você passa o nome de cada nome de arquivo da origem para a porta ScriptName da transformação SQL. O nome de arquivo do script contém o caminho completo para o arquivo de script.

Ao configurar a transformação SQL para ser executada no modo de script, você criará uma transformação passiva. A transformação retorna uma linha de cada linha de entrada. A linha de saída contém resultados da consulta e de qualquer erro do banco de dados.

Quando a transformação SQL é executada em modo de script, os dados e a instrução da consulta não se alteram. Se precisar executar diversas consultas em modo de script, passe os scripts nos dados de origem. Use o modo de script para executar consultas de definição de dados, como criar ou descartar tabelas.

Quando você configurar uma transformação SQL para ser executada em modo de script, o Designer adiciona a porta de entrada ScriptName à transformação. Ao criar um mapeamento, você conecta a porta ScriptName a outra que contenha o nome de um script a ser executado para cada linha. Você pode executar um script SQL diferente para cada linha de entrada. O Designer cria portas padrão que retornam informações sobre resultados de consultas.

Uma transformação SQL configurada para modo de script tem as seguintes portas padrão:

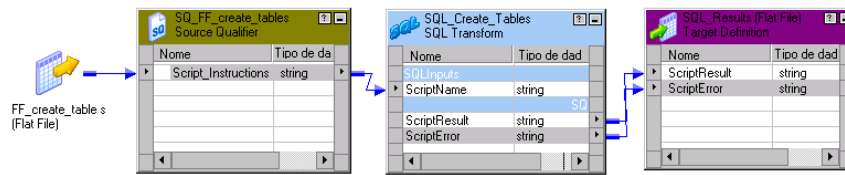
Porta	Tipo	Descrição
ScriptName	Entrada	Recebe o nome do script a ser executado para a linha atual.
ScriptResult	Saída	Retornará PASSED se a execução do script para a linha tiver êxito. Caso contrário, conterà FAILED.
ScriptError	Saída	Retorna erros que ocorrem quando um script para uma linha falha.

Exemplo

Você precisa criar tabelas de pedido e inventário antes de adicionar novos dados às tabelas. Você pode criar um script SQL para criar as tabelas e configurar uma transformação SQL para executar o script.

Você cria um arquivo nomeado create_order_inventory.txt, que contém as instruções SQL para criar as tabelas.

O seguinte mapeamento mostra como passar o nome de script para a transformação de SQL:



O Serviço de Integração lê uma linha da origem. A linha de origem contém o nome e o caminho do arquivo de script SQL:

```
C:\81\server\shared\SrcFiles\create_order_inventory.txt
```

A transformação recebe o nome do arquivo na porta `ScriptName`. O Serviço de Integração localiza o arquivo de script e analisa o script. Ele cria um procedimento SQL e o envia para o banco de dados processá-lo. O banco de dados valida o SQL e executa a consulta.

A transformação SQL retorna `ScriptResults` e `ScriptError`. Se o script executar com sucesso, a porta de saída `ScriptResult` retorna `PASSED`. Caso contrário, a porta `ScriptResult` retorna `FAILED`. Quando `ScriptResult` é `FAILED`, a transformação SQL retorna mensagens de erro na porta `ScriptError`. A transformação SQL retorna uma linha para cada linha de entrada que recebe.

Regras e Diretrizes do Modo de Script

Utilize as regras e diretrizes a seguir para uma transformação SQL que seja executada no modo de script:

- Com o modo de script, você pode usar uma conexão de banco de dados estática ou dinâmica.
- Para incluir várias instruções de consulta em um script, separe-as com ponto-e-vírgula.
- Você pode usar parâmetros ou variáveis de mapeamento no nome do arquivo de script.
- A página de código do script assume como padrão a localização do sistema operacional. Você pode alterar a localização do script.
- É necessário que o Serviço de Integração tenha acesso ao arquivo de script. O Serviço de Integração precisa ter permissões de leitura no diretório que contém o script. Se o Serviço de Integração utilizar perfis do sistema operacional, é necessário que o usuário do sistema operacional do perfil do sistema operacional tenha permissões de leitura no diretório que contém o script.
- O Serviço de Integração ignora a saída de qualquer instrução `SELECT` que você inclua no script SQL. A transformação SQL no modo de script não produz mais de uma linha de dados para cada linha de entrada.
- Você não pode usar linguagens de scripts como Oracle PL/SQL ou Microsoft/Sybase T-SQL no script.
- Você não pode usar scripts aninhados nos casos em que o script SQL chama outro script SQL.
- Um script não pode aceitar argumentos de tempo de execução.

Modo de Consulta

Quando uma transformação SQL é executada no modo de consulta, executa uma consulta SQL que você define na transformação. Você passa strings ou parâmetros das portas de entrada da transformação para alterar a instrução ou os dados da consulta.

Ao configurar a transformação SQL para ser executada no modo de consulta, você criará uma transformação ativa. A transformação pode retornar diversas linhas para cada linha de entrada.

Criar consultas no Editor SQL de transformações SQL. Para criar uma consulta, digite a instrução da consulta na janela principal do Editor SQL. O Editor SQL fornece uma lista das portas da transformação a que você pode fazer referência na consulta. Você pode clicar duas vezes no nome de uma porta para adicioná-la como parâmetro da consulta.

Quando você cria uma consulta, o Editor SQL valida os nomes das portas na consulta. Verifica também se as portas usadas para substituição de strings são tipos de dados de string. O Editor SQL não valida a sintaxe da consulta SQL.

Você pode criar os seguintes tipos de consultas SQL na transformação SQL:

- **Consulta SQL estática.** A instrução da consulta não é alterada, mas você pode usar parâmetros da consulta para alterar dados. O Serviço de Integração prepara a consulta uma vez e a executa para todas as linhas de entrada.
- **Consulta SQL dinâmica.** Você pode alterar instruções e dados da consulta. O Serviço de Integração prepara uma consulta para cada linha de entrada.

Quando você cria uma consulta estática, o Serviço de Integração prepara o procedimento SQL uma vez e o executa para cada linha. Quando você cria uma consulta dinâmica, o Serviço de Integração prepara o SQL para cada linha de entrada. Você pode criar consultas estáticas para otimizar o desempenho.

Usando Consultas SQL Estáticas

Crie uma consulta SQL estática quando precisar executar as mesmas instruções de consulta para cada linha de entrada, mas quiser alterar os dados na consulta para cada uma dessas linhas. Ao criar uma consulta SQL estática, você usa a associação de parâmetros no Editor SQL para definir parâmetros para dados da consulta.

Para alterar os dados na consulta, configure parâmetros de consulta e associe-os a portas de entrada na transformação. Quando vincula um parâmetro a uma porta de entrada, você identifica a porta pelo nome na consulta. O Editor SQL coloca o nome entre pontos de interrogação (?). Os dados da consulta mudam de acordo com o valor dos dados na porta de entrada.

As portas de entrada da transformação SQL recebem os dados para os valores de dados na consulta, ou os valores na cláusula WHERE da consulta.

Estas consultas estáticas utilizam a associação de parâmetros:

```
DELETE FROM Employee WHERE Dept = ?Dept?
INSERT INTO Employee(Employee_ID, Dept) VALUES (?Employee_ID?, ?Dept?)
UPDATE Employee SET Dept = ?Dept? WHERE Employee_ID > 100
```

A consulta SQL estática a seguir contém parâmetros de consulta que vinculam às portas de entrada Employee_ID e Dept de uma transformação SQL:

```
SELECT Name, Address FROM Employees WHERE Employee_Num =?Employee_ID? and Dept = ?Dept?
```

A origem poderia ter as seguintes linhas:

Employee_ID	Dept
100	Products
123	HR
130	Accounting

O Serviço de Integração gera as seguintes instruções de consulta a partir das linhas:

```
SELECT Name, Address FROM Employees WHERE Employee_ID = '100' and DEPT = 'Products'
SELECT Name, Address FROM Employees WHERE Employee_ID = '123' and DEPT = 'HR'
SELECT Name, Address FROM Employees WHERE Employee_ID = '130' and DEPT = 'Accounting'
```

Selecionando Diversas Linhas do Banco de Dados

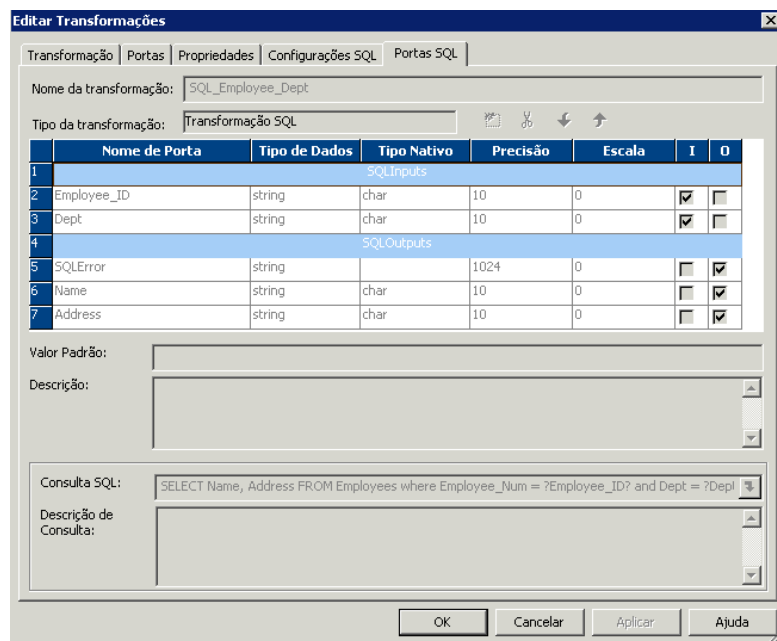
Quando a consulta SQL contém uma instrução SELECT, a transformação retorna uma linha para cada linha recuperada do banco de dados. Você deve configurar uma porta de saída para cada coluna na instrução SELECT. As portas de saída devem estar na mesma ordem das colunas na instrução SELECT.

Ao configurar portas de saída para colunas do banco de dados, você deve configurar o tipo de dados de cada coluna selecionada do banco de dados. Selecione um tipo de dados nativo na lista. Quando você seleciona o tipo de dados nativo, o Designer configura o tipo de dados da transformação para você.

O tipo de dados nativo na transformação deve coincidir com o tipo de dados da coluna do banco de dados. O Serviço de Integração corresponde o tipo de dados da coluna no banco de dados ao tipo do banco de dados nativo na transformação em tempo de execução. Se os tipos de dados não corresponderem, o Serviço de Integração gerará um erro de linha.

Nota: Embora o banco de dados Teradata permita colunas Bigint, o Transformation Developer não inclui o tipo de dados como um dos tipos de dados nativos que você pode usar na transformação SQL.

A figura a seguir mostra as portas na transformação configurada para ser executada no modo de consulta:



As portas de entrada recebem os dados na cláusula WHERE. As portas de saída retornam as colunas da instrução SELECT. A consulta SQL seleciona o nome e o endereço da tabela de funcionários. A transformação SQL grava uma linha no destino para cada linha recuperada do banco de dados.

Utilizando Consultas SQL Dinâmicas

Uma consulta SQL dinâmica pode executar diferentes instruções de consulta para cada linha de entrada. Ao criar uma consulta SQL dinâmica, você usa substituição de strings para definir parâmetros de sequência na consulta e as vincula a portas de entrada na transformação.

Para alterar uma instrução de consulta, configure uma variável de string na consulta para a parte da consulta que deseja alterar. Para configurar a variável de string, identifique uma porta de entrada com o nome na consulta e coloque um til (~) no final do nome. A consulta muda de acordo com o valor dos dados na porta. A porta de entrada da transformação que contém o parâmetro de consulta deve ser um tipo de dados de string. Você pode usar substituição de strings para alterar a instrução e os dados da consulta.

Quando você cria uma consulta SQL dinâmica, o Serviço de Integração prepara uma consulta para cada linha de entrada. Você pode passar a consulta inteira ou parte dela em uma porta de entrada:

- **Consulta inteira.** Você pode substituir a consulta SQL toda por instruções de consulta dos dados de origem.
- **Consulta parcial** Você pode substituir uma parte da instrução de consulta, como o nome da tabela.

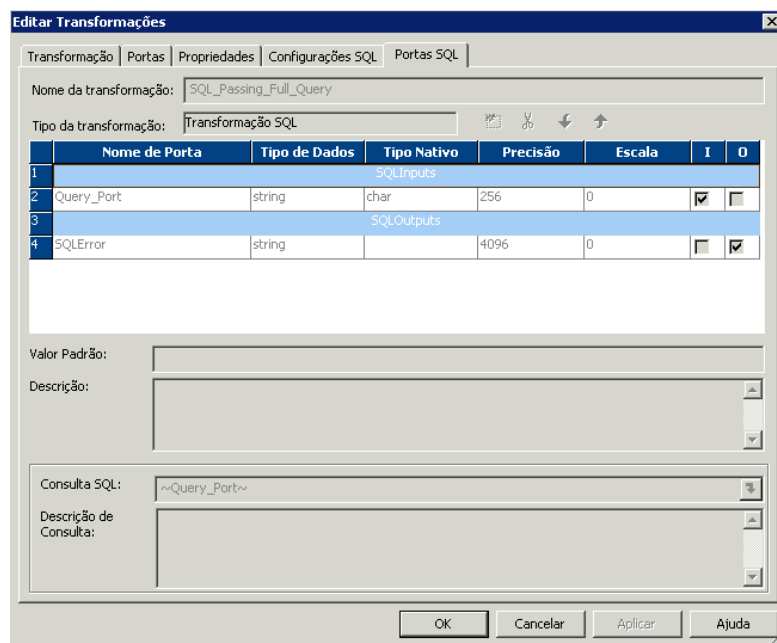
Passando a Consulta Completa

Você pode passar a consulta SQL completa através de uma porta de entrada na transformação. Para passar a consulta completa, crie uma consulta no Editor SQL formada por uma variável de string que represente a consulta completa:

```
~Query_Port~
```

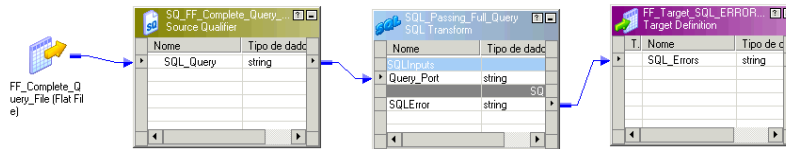
A transformação recebe a consulta na porta de entrada Query_Port.

A figura a seguir mostra portas na transformação de SQL:



O Serviço de Integração substitui a variável ~Query_Port~ na consulta dinâmica pelas instruções SQL da origem. Ele prepara a consulta e envia-a para o banco de dados processá-la. O banco de dados executa a consulta. A transformação de SQL retorna erros do banco de dados para a porta SQL_Error.

O mapeamento a seguir mostra como passar a consulta para a transformação de SQL:



Ao passar a consulta completa, você pode passar mais de uma instrução de consulta para cada linha de entrada. Por exemplo, a origem pode conter as seguintes linhas:

```
DELETE FROM Person WHERE LastName = 'Jones'; INSERT INTO Person (LastName, Address)
VALUES ('Smith', '38 Summit Drive')
DELETE FROM Person WHERE LastName = 'Jones'; INSERT INTO Person (LastName, Address)
VALUES ('Smith', '38 Summit Drive')
DELETE FROM Person WHERE LastName = 'Russell';
```

Você pode passar qualquer tipo de consulta nos dados de origem. Ao configurar instruções SELECT na consulta, você precisa configurar portas de saída para as colunas que recupera do banco de dados. Quando você combina instruções SELECT e outros tipos de consulta, as portas de saída que representam colunas do banco de dados contêm valores nulos caso nenhuma coluna de banco de dados seja recuperada.

Substituindo o Nome da Tabela em uma String

Você pode substituir o nome da tabela em uma consulta. Para substituir o nome da tabela, configure uma porta de entrada para receber o nome da tabela de cada linha de entrada. Identifique a porta de entrada com o nome na consulta e coloque um til (~) no final do nome.

A consulta dinâmica a seguir contém uma variável de string, ~Table_Port~:

```
SELECT Emp_ID, Address from ~Table_Port~ where Dept = 'HR'
```

A origem poderá passar os seguintes valores à coluna Table_Port:

Table_Port

Employees_USA

Employees_England

Employees_Australia

O Serviço de Integração substitui a variável ~Table_Port~ pelo nome da tabela na porta de entrada:

```
SELECT Emp_ID, Address from Employees_USA where Dept = 'HR'
SELECT Emp_ID, Address from Employees_England where Dept = 'HR'
SELECT Emp_ID, Address from Employees_Australia where Dept = 'HR'
```

TÓPICOS RELACIONADOS:

- [“Exemplo de atualização dinâmica” na página 460](#)

Configuração de Portas de Passagem

Você pode adicionar portas de passagem à transformação SQL. As portas de passagem são portas de entrada/saída que passam dados pela transformação. A transformação SQL retorna dados das portas de passagem, independentemente de uma consulta SQL retornar linhas ou não.

Quando uma linha de origem contiver uma instrução de consulta SELECT, a transformação SQL retornará os dados na porta de passagem em cada linha que retornar do banco de dados. Se o resultado da consulta contiver várias linhas, a transformação SQL repetirá os dados de passagem em cada linha.

Quando uma consulta não retornar linhas, a transformação SQL retornará os dados da coluna de passagem com valores nulos nas colunas de saída. Por exemplo, as consultas que contêm instruções INSERT, UPDATE e DELETE não retornam linhas. Quando a consulta contiver erros, a transformação SQL retornará os dados da coluna de passagem, a mensagem `SQLError` e valores nulos nas portas de saída.

Para criar uma porta de passagem:

- Crie uma porta de entrada e habilite-a para saída. O Designer criará uma porta de saída e adiciona o sufixo “_output” ao nome da porta.
- Arraste uma porta de uma transformação Qualificador de origem até a transformação SQL. O Designer criará uma porta de passagem.

Configuração do Modo Passivo

Quando criar uma transformação SQL, você pode configurar a transformação SQL para ser executada no modo passivo, em vez de no modo ativo. Uma transformação passiva não altera o número de linhas que passam por ela. Ela mantém limites da transação e tipos de linha.

Se configurar a transformação para ser executada no modo Consulta, você poderá configurar o modo passivo ao criar a transformação. Depois de criar a transformação, você não poderá alterar o modo.

Regras e Diretrizes do Modo Passivo

Ao configurar a transformação SQL para ser executada no modo passivo, utilize as regras e diretrizes a seguir:

- Se uma consulta SELECT retornar mais de uma linha, o Serviço de Integração retornará a primeira linha e um erro para a porta `SQLError`. O erro indica que a transformação SQL gerou várias linhas.
- Se a consulta SQL tiver várias instruções SQL, o Serviço de Integração executará todas elas. O Serviço de Integração retorna dados somente para a primeira instrução SQL. A transformação SQL retorna uma linha. A porta `SQLError` contém os erros de todas as instruções SQL. Quando ocorrem vários erros, eles são separados por pontos-e-vírgulas na porta `SQLError`.
- Se a consulta SQL contiver várias instruções SQL e houver uma porta de estatísticas habilitada, o Serviço de Integração retornará os dados e as estatísticas da primeira instrução SQL. A porta `SQLError` contém os erros de todas as instruções SQL.

Regras e Diretrizes do Modo de Consulta

Utilize as seguintes regras e diretrizes ao configurar a transformação SQL para ser executada no modo de consulta:

- O número e a ordem das portas de saída deve corresponder ao número e à ordem dos campos na cláusula SELECT da consulta.
- O tipo de dados nativo de uma porta de saída na transformação deve corresponder ao tipo de dados da coluna coincidente no banco de dados. O Serviço de Integração gera um erro de linha quando os tipos de dados não correspondem.
- Se a consulta SQL contiver uma cláusula INSERT, UPDATE ou DELETE, a transformação retornará dados à porta `SQLError`, às portas pass-through e à porta `NumRowsAffected`, quando estiver ativada. Se você adicionar portas de saída, elas receberão valores de dados NULL.

- Se a consulta SQL contiver uma instrução SELECT e a transformação tiver uma porta pass-through, a transformação retornará dados à porta pass-through, independente de a consulta retornar dados do banco de dados. A transformação SQL retorna uma linha com dados NULL nas portas de saída.
- Não é possível adicionar o sufixo "_output" aos nomes de portas de saída que você criar.
- Não é possível usar a porta pass-through para retornar dados de uma consulta SELECT.
- Quando o número de portas de saída é maior do que o número de colunas na cláusula SELECT, as portas extras recebem um valor NULL.
- Quando o número de portas de saída é menor do que o número de colunas na cláusula SELECT, o Serviço de Integração gera um erro de linha.
- Você pode usar substituição de string, em vez de associação de parâmetros em uma consulta. Entretanto, as portas de entrada devem ser tipos de dados de string.

Conectando com Bancos de Dados

É possível usar uma conexão de bancos de dados estática ou transferir informações de conexão de bancos de dados para a transformação SQL em tempo de execução.

É possível usar uma instrução SQL de ambiente de conexão ou instrução SQL transacional com a transformação SQL. Configure as instruções SQL em um objeto de conexão relacional. O Serviço de Integração executa o SQL de ambiente de conexão ao se conectar ao banco de dados. Ele executa a instrução SQL transacional antes do início de cada transação.

Use um dos seguintes tipos de conexão para conectar a transformação SQL a um banco de dados:

- **Conexão estática.** Configure o objeto de conexão na sessão. Primeiramente, é necessário criar o objeto de conexão no Workflow Manager.
- **Conexão lógica.** Transfira o nome da conexão para a transformação SQL na qualidade de dados de entrada em tempo de execução. Primeiramente, é necessário criar o objeto de conexão no Workflow Manager.
- **Conexão total com o banco de dados.** Transfira a cadeia de conexão, o nome de usuário, a senha e demais informações de conexão às portas de entrada da transformação SQL em tempo de execução.

Nota: Se uma sessão apresentar várias partições, a transformação SQL criará uma conexão de banco de dados distinta para cada partição.

Usando uma Conexão de Banco de Dados Estática

Você pode configurar a transformação SQL para se conectar a um banco de dados com uma conexão estática. Uma conexão de banco de dados estática é definida no Workflow Manager.

Para usar uma conexão estática, selecione um objeto de conexão relacional ao configurar a sessão. Para evitar erros de conversão de tipos de dados, use uma conexão relacional para o mesmo tipo de dados do banco de dados configurada na transformação.

Passando uma Conexão Lógica de Banco de Dados

Você pode configurar a transformação SQL para se conectar a um banco de dados com uma conexão de banco de dados lógica. Uma conexão lógica de banco de dados é o nome de um objeto de conexão que você passa à transformação na execução. Defina o objeto de conexão relacional no Workflow Manager. Quando

you configure the transformation to use a logical database connection, the Designer creates an input LogicalConnectionObject.

You can pass a logical connection for each input line. Configure the mapping to pass the name of the connection object to the LogicalConnectionObject input. To avoid type conversion errors, use a relational connection for the same data type configured in the transformation.

Passando Informações de Conexão Completa

You can pass all the connection information of the database to a SQL transformation as data from the input port. When you configure a SQL transformation to connect to a database with a complete connection, the Designer creates input ports for the connection components. The type of database adopted as the standard is the type of database configured for the transformation.

The following table describes the ports created by the Designer when you configure a SQL transformation to connect to a database with a complete connection:

Porta	Obrigatório Opcional	Descrição
ConnectionString	Obrigatório	Contém o nome do banco de dados e o nome do servidor de banco de dados.
DBUser	Obrigatório	Nome do usuário com permissões para ler e gravar no banco de dados.
DBPasswd	Obrigatório	Senha do DBUser.
CodePage	Opcional	Página de código usada pelo Serviço de Integração para ler ou gravar no banco de dados. Use o nome da página de código ISO, como ISO-8859-6. O nome da página de código não diferencia maiúsculas de minúsculas.
AdvancedOptions	Opcional	Atributos da conexão. Passe os atributos como pares de nome-valor. Delimite cada atributo com ponto-e-vírgula. Os nomes de atributos não diferenciam maiúsculas de minúsculas.

Passando a String de Conexão

A native connection string contains the database name and the server name. A connection string allows the PowerCenter and the database client to direct calls directly to the correct database.

The following table lists the syntax of the native connection string for each database:

Banco de Dados	Sintaxe de String de Conexão	Exemplo
IBM DB2	<i>dbname</i>	mydatabase
Microsoft SQL Server	<i>servername@dbname</i>	sqlserver@mydatabase
Oracle	<i>dbname.mundo</i> (mesmo da entrada TNSNAMES)	oracle.world
Sybase ASE ¹	<i>servername@dbname</i>	sambrown@mydatabase

Banco de Dados	Sintaxe de String de Conexão	Exemplo
Teradata ²	<i>ODBC_data_source_name</i> ou <i>ODBC_data_source_name@db_name</i> ou <i>ODBC_data_source_name@db_user_name</i>	TeradataODBC TeradataODBC@mydatabase TeradataODBC@sambrown
Vertica	<i>ODBC_data_source_name</i>	VerticaDSN

¹. O *servername* Sybase ASE é o nome do Servidor Adaptável do arquivo de interfaces.

². Use drivers ODBC Teradata para conectar-se com bancos de dados de origem e destino.

Passando Opções Avançadas

Você pode configurar atributos opcionais de conexão. Para configurá-los, passe os atributos como pares nome-valor. Delimite os atributos com ponto-e-vírgula. Os nomes de atributos não diferenciam maiúsculas de minúsculas.

Por exemplo, você poderia passar a seguinte string para configurar opções de conexão:

```
Use Trusted Connection = 1; Connection Retry Period = 5
```

A seguinte tabela descreve as opções avançadas que você pode configurar:

Atributo	Tipo de Banco de Dados	Descrição
Período de Repetição de Conexão	Tudo	Número inteiro. O número de segundos que o Serviço de Integração tentará se reconectar com o banco de dados, caso a conexão falhe.
Nome da Fonte de Dados	Teradata	String. O nome da origem de dados ODBC do Teradata.
Nome do Banco de Dados	Sybase ASE Microsoft SQL Server Teradata	String. Substitua o nome do banco de dados padrão no ODBC. Se você não inserir um nome de banco de dados, as mensagens relacionadas à conexão não mostrarão o nome de um banco de dados.
Nome do Domínio	Microsoft SQL Server	String. O nome do domínio em que o Microsoft SQL Server está sendo executado.
Habilitar Modo Paralelo	Oracle	Número inteiro. Ative o processamento paralelo quando você carregar dados no modo em massa. 0 não está ativado. 1 está ativado. Por padrão, essa opção fica ativada.
Nome do Proprietário	Tudo	String. Nome do proprietário da tabela.

Atributo	Tipo de Banco de Dados	Descrição
Tamanho do Pacote	Sybase ASE Microsoft SQL Server	Número inteiro. Otimize a conexão ODBC para Sybase ASE e Microsoft SQL Server.
Nome do Servidor	Sybase ASE Microsoft SQL Server	String. O nome do servidor de banco de dados.
Usar Conexão Confiável	Microsoft SQL Server	Número inteiro. O Serviço de Integração utiliza a autenticação do Windows, quando ela está ativada, para acessar o banco de dados do Microsoft SQL Server. O nome de usuário que inicia o Serviço de Integração deve corresponder a um usuário válido do Windows com acesso ao banco de dados do Microsoft SQL Server. 0 não está ativado. 1 está ativado.

Regras e Diretrizes para Conexões com Bancos de Dados

Use as seguintes regras e diretrizes ao configurar conexões com bancos de dados para a transformação SQL:

- Você precisa da chave de licença do PowerCenter para conectar diferentes tipos de bancos de dados. A sessão será encerrada se o PowerCenter não tiver licença para se conectar com o banco de dados.
- Para aumentar o desempenho, use uma conexão estática com o banco de dados. Quando você configura uma consulta dinâmica, o Serviço de Integração estabelece uma nova conexão para cada linha de entrada.
- Quando houver um número limitado de conexões para usar na sessão, você pode configurar diversas transformações SQL. Configure cada transformação SQL para usar uma conexão estática diferente. Use transformação Roteador para rotear linhas para uma transformação SQL com base em informações de conectividade na linha.
- Quando você conectar a transformação SQL para usar dados de conexão completos, a senha do banco de dados será texto simples. Você pode passar conexões lógicas quando tiver um número limitado de conexões que deva usar em uma sessão. Uma conexão lógica proporciona a mesma funcionalidade que a conexão completa e a senha do banco de dados é protegida.
- Quando você passa conexões de banco de dados lógicas para a transformação SQL, o Serviço de Integração acessa o repositório para recuperar as informações de conexão de cada linha de entrada. Se houver muitas linhas para processar, passar conexões de banco de dados lógicas poderá ter impacto sobre o desempenho.
- A transformação SQL usa um tipo de banco de dados nativo por padrão. Se desejar executar a sessão usando ODBC, será necessário configurar a transformação com um tipo de banco de dados ODBC. Se a transformação contiver uma porta datetime ou datetime2, defina o tipo nativo correspondente como carimbo de data/hora.

Processamento de Sessão

Quando o Serviço de Integração processa uma transformação SQL, ele executa consultas SQL midstream no pipeline. Quando uma consulta SELECT recupera linhas de banco de dados, a transformação SQL retorna as colunas do banco de dados nas portas de saída. No caso de outros tipos de consulta, a transformação SQL retorna resultados da consulta, dados de passagem ou erros de banco de dados em portas de saída.

A transformação SQL configurada para ser executada no modo de script sempre retorna uma linha para cada linha de entrada. Uma transformação SQL que seja executada no modo de consulta pode retornar um número diferente de linhas para cada linha de entrada. O número de linhas que a transformação SQL retorna baseia-se no tipo da consulta que ela executa e no sucesso dessa consulta.

Você pode exibir um log da consulta SQL que o Serviço de Integração passa para processamento do banco de dados. Quando você define o log como detalhado, o Serviço de Integração grava cada consulta SQL no log da sessão. Defina o log como detalhado quando precisar depurar uma sessão com a transformação SQL.

Você pode usar o controle de transação com a transformação SQL quando configurá-la para usar uma conexão com banco de dados estática. Você também pode emitir instruções de confirmação e reversão na consulta.

A transformação SQL fornece alguma resiliência de conexão de banco de dados. Ela fornece resiliência para deadlocks de banco de dados.

TÓPICOS RELACIONADOS:

- [“Cardinalidade de linha de entrada para linha de saída” na página 449](#)
- [“Alta disponibilidade” na página 447](#)

Controle de transação

Uma transformação SQL executada no modo script descarta todos os limites de transações recebidas de um gerador de transação ou origem de upstream. O Serviço de Integração emite uma confirmação depois de executar o script para cada linha de entrada na transformação SQL. A transação contém o conjunto de linhas afetadas pelo script.

Uma transformação SQL executada no modo de consulta confirma transações em pontos diferentes, com base no tipo de conexão do banco de dados:

- **Conexão de banco de dados dinâmica.** O Serviço de Integração emite uma confirmação depois de executar o SQL para cada linha de entrada. A transação é o conjunto de linhas afetadas pelo script. Não se pode usar uma transformação Controle de Transação com conexões dinâmicas no modo de consulta.
- **Conexão estática.** O Serviço de Integração emite uma confirmação depois de processar todas as linhas de entrada. A transação inclui todas as linhas a atualizar do banco de dados. Você pode substituir o comportamento padrão usando uma transformação Controle de Transação para controlar a transação ou usando instruções de confirmação e reversão na consulta SQL.

Ao configurar uma instrução SQL para confirmar ou reverter linhas, configure a transformação SQL para gerar transações com a propriedade da transformação Gerar Transação. Configure a sessão para confirmação definida pelo usuário.

As instruções SQL de controle de transação não são válidas com a transformação SQL:

- **SAVEPOINT.** Identifica um ponto de reversão na transação.
- **SET TRANSACTION.** Altera opções da transação.

TÓPICOS RELACIONADOS:

- [“Exemplo de conexão dinâmica” na página 465](#)

Confirmação Automática

Você pode configurar as conexões do banco de dados de transformação SQL para operar no modo de confirmação automática. Quando você configura o modo de confirmação automática, ocorre uma confirmação quando uma instrução SQL termina.

Para ativar a confirmação automática, selecione confirmação automática na guia Configurações SQL da Transformação SQL.

Se a recuperação do HA for ativada com a confirmação automática, o log da sessão contém uma mensagem de aviso de que a integridade dos dados não pode ser garantida se a consulta contiver uma instrução de inserção, atualização ou exclusão.

Alta disponibilidade

Quando você tem alta disponibilidade, a transformação SQL fornece a resiliência da conexão ao banco de dados para conexões estáticas e dinâmicas. Quando o Serviço de Integração falha ao conectar-se ao banco de dados, ele tenta refazer a conexão. Você pode configurar o período de novas tentativas de conexão. Quando o Serviço de Integração não pode conectar-se ao banco de dados no período que você configurou, ele gera um erro de linha para uma conexão dinâmica ou provoca a falha da sessão para uma conexão estática.

Nota: A conexão ao banco de dados da transformação SQL não é resiliente para conexões Informix ou ODBC.

Processamento exatamente um para sessões em tempo real

O Serviço de Integração fornece a entrega “exatamente um” de mensagens de origens em tempo real, com a transformação SQL. Uma mensagem em tempo real deve ser entregue uma vez para a transformação SQL. Se o processamento for interrompido, o Serviço de Integração pode se recuperar sem exigir que a mensagem seja enviada novamente. Se a mensagem for reenviada, o Serviço de Integração não executa instruções DML a partir da mensagem duas vezes. O Serviço de Integração pode executar outras instruções de SQL como SELECT ou SET novamente.

Para executar o processamento exatamente um, o Serviço de Integração armazena um estado de operações para um ponto de verificação na tabela PM_REC_STATE. Cada transformação SQL possui um estado separado de operações. Cada transformação SQL mantém um estado consistente e não compartilha as conexões. Você deve ter uma alta disponibilidade para o processamento exatamente um.

Use as seguintes regras e diretrizes para recuperar as sessões em tempo real com as transformações SQL:

- Você deve configurar o escopo da transformação para Transação, a fim de ativar a recuperação para uma sessão que contenha uma transformação SQL.
- Você não pode incluir a auto-confirmação, as instruções de confirmação ou as instruções de DDL nas consultas SQL.
- Você não pode ativar a recuperação HA para uma transformação SQL se for uma transformação passiva, ou se você a configurou para conexões dinâmicas ou para o Modo de script.
- Uma sessão pode falhar quando você habilita a recuperação para a transformação da SQL e o fluxo de trabalho é habilitado para execução simultânea ou a sessão é executada em várias partições. Pode ocorrer um deadlock do banco de dados na tabela PM_REC_STATE.

Resiliência do Modo de Consulta

Quando ocorre um erro de conexão com o banco de dados enquanto o Serviço de Integração executa uma consulta SQL para uma linha de entrada, ele tenta se reconectar com o banco de dados.

Use as seguintes regras e diretrizes para resiliência no modo de consulta:

- Quando o Serviço de Integração executa a primeira consulta de uma lista e ocorre uma falha na conexão com o banco de dados, o Serviço de Integração a executa novamente para a linha. Executa o restante das consultas para a linha.
- Se a instrução da primeira consulta for SELECT e ocorrer um erro de comunicação depois que algumas linhas forem liberadas no pipeline, a sessão será encerrada.
- Se ocorrer uma falha de conexão enquanto o Serviço de Integração executa uma consulta que não seja a primeira de uma lista, ele se conectará novamente ao banco de dados e ignorará o restante das consultas para a linha. Podem ser perdidos resultados de consultas das linhas anteriores.
- Se o Serviço de Integração não puder se conectar novamente ao banco de dados, a sessão será encerrada caso a transformação SQL use uma conexão estática. Se a transformação SQL estiver usando uma conexão dinâmica, a sessão continuará.
- Se a consulta SQL na lista tiver instruções explícitas de reversão ou confirmação, todos os resultados criados após o ponto de confirmação serão perdidos quando a sessão continuar.

Resiliência do Modo de Script

Quando ocorre uma falha na comunicação enquanto o Serviço de Integração está executando um script para uma linha de entrada, ele tenta reconectar-se ao banco de dados.

Use as seguintes regras e diretrizes para resiliência no modo de script:

- Se ele não conseguir se conectar ao banco de dados e a conexão permanecer estática, a sessão falhará. Se a transformação SQL estiver usando uma conexão dinâmica, a sessão continuará.
- Se o Serviço de Integração conseguir reconectar-se ao banco de dados, ele ignorará o processamento da linha atual e continuará na seguinte.

Resiliência de Deadlock de Banco de Dados

A transformação SQL é resiliente a erros de deadlock de bancos de dados ao ativar a propriedade de sessão Repetir a Sessão em caso de Deadlock. A transformação SQL é resiliente a erros de deadlock de bancos de dados em modo de Consulta, mas não é resiliente a erros de deadlock em modo Script. Caso ocorra um deadlock em modo de Consulta, o Serviço de Integração tenta se reconectar com o banco de dados, com base no número de novas tentativas de deadlock que você configurou. Configure o Serviço de Integração para que defina o número de novas tentativas de deadlock e o período do tempo de inatividade do deadlock.

Ao ocorrer um deadlock, o Serviço de Integração tenta novamente seguir as instruções SQL na linha atual, caso essa não apresente instruções DML. Se a linha contiver uma instrução DML, como INSERT, UPDATE ou DELETE, o Serviço de Integração não processará a linha atual novamente.

Em uma conexão dinâmica, o Serviço de Integração retorna um erro na porta SQLError no caso de falha na nova tentativa. O Serviço de Integração processa a próxima instrução com base na propriedade Continuar no Erro de SQL na Linha. Se a propriedade estiver desativada, o Serviço de Integração ignora a linha atual. Se a linha atual contiver uma instrução DML, como INSERT, UPDATE ou DELETE, o Serviço de Integração incrementará a contagem de erros.

Em uma conexão estática, se as novas tentativas falharem, o Serviço de Integração retornará um erro na porta SQLError. Se a linha atual contiver uma instrução DML, o Serviço de Integração falha com a sessão. O Serviço de Integração processa a próxima instrução com base na propriedade Continuar no Erro de SQL na Linha. Se a propriedade estiver desativada, o Serviço de Integração ignorará a linha atual.

Log de Consultas SQL

Você pode exibir um log da consulta SQL que o Serviço de Integração passa para processamento do banco de dados. Quando você define o log como detalhado, o Serviço de Integração grava cada consulta SQL no log da sessão. Defina o log como detalhado quando precisar depurar uma sessão com a transformação SQL.

Se a consulta contiver dados CLOB ou BLOB, o log da sessão não conterá a consulta. O log da sessão contém uma mensagem que descreve os dados, como dados CLOB.

Cardinalidade de linha de entrada para linha de saída

Quando o Serviço de Integração executa uma consulta SELECT, a transformação SQL retorna uma linha para cada linha que ela recupera. Quando a consulta não recupera dados, a transformação SQL retorna zero ou uma linha para cada linha de entrada.

O número de linhas de saída que a transformação SQL retorna depende dos seguintes fatores:

- **Processamento de instruções de consulta.** Quando a consulta contém uma instrução SELECT, o Serviço de Integração pode recuperar múltiplas linhas de saída. Quando uma consulta SELECT tem sucesso, a transformação SQL pode recuperar múltiplas linhas. Quando a consulta contém outras instruções, o Serviço de Integração pode gerar uma linha que contém erros de SQL ou o número de linhas afetadas.
- **Configuração da porta.** A porta de saída NumRowsAffected contém o número total de linhas afetadas pelas atualizações, inserções ou exclusões de uma linha de entrada. Quando a transformação SQL contém portas de passagem, ela retorna pelo menos uma vez os dados da coluna para cada linha de origem.
- **A configuração da contagem máxima de linhas.** A opção Contagem Máx das Linhas de Saída limita o número de linhas que a transformação SQL retorna das consultas SELECT.
- **Linhas de erro.** Quando encontra um erro de conexão ou de sintaxe, o Serviço de Integração retorna linhas de erro. Quando a transformação SQL executa em modo de consulta, retorna erros para a porta SQL_Error. Quando a transformação SQL é executada em modo de script, ela retorna erros para a porta ScriptError.
- **Continuar no Erro de SQL** Você pode configurar a transformação SQL para continuar processando quando houver um erro em uma instrução SQL. A transformação de SQL não gera um erro de linha.

Processamento de Instruções de Consulta

O tipo de consulta determina quantas linhas a transformação SQL retornará. A transformação SQL executada em modo de consulta pode retornar nenhuma, uma ou várias linhas. Quando a consulta contém uma instrução SELECT, a transformação SQL retorna cada coluna do banco de dados para uma porta de saída. A transformação retorna todas as linhas qualificadas.

A tabela a seguir lista as linhas de saída geradas pela transformação SQL para diversos tipos de instruções de consulta quando não ocorrerem erros no modo de consulta:

Instrução SQL	Linhas de saída
Apenas UPDATE, INSERT, DELETE	Nenhuma linha.
Uma ou mais instruções SELECT	Número total de linhas recuperadas do banco de dados.
Consultas DDL, como CREATE, DROP, TRUNCATE	Nenhuma linha.

Número de Linhas Afetadas

Você pode ativar a porta de saída NumRowsAffected para retornar o número de linhas afetadas pelas instruções de consulta INSERT, UPDATE ou DELETE em cada linha de entrada. O Serviço de Integração retorna o NumRowsAffected para cada instrução da consulta. O NumRowsAffected está desabilitado por padrão.

Quando você ativa NumRowsAffected no modo de consulta, e a consulta SQL não contém uma instrução INSERT, UPDATE ou DELETE, NumRowsAffected é zero em cada linha de saída.

Nota: Quando você ativa NumRowsAffected e a transformação está configurada para executar no modo de script, NumRowsAffected é sempre NULL.

A tabela a seguir lista as linhas de saída geradas pela transformação SQL quando você ativa o NumRowsAffected no modo de consulta:

Instrução SQL	Linhas de saída
Apenas UPDATE, INSERT, DELETE	Uma linha para cada instrução com o NumRowsAffected referente à instrução.
Uma ou mais instruções SELECT	Número total de linhas recuperadas do banco de dados. NumRowsAffected é zero em cada linha.
Consultas DDL, como CREATE, DROP, TRUNCATE	Uma linha com NumRowsAffected zero.

Quando a transformação SQL executa no modo de consulta e uma consulta contém múltiplas instruções, o Serviço de Integração retorna o NumRowsAffected para cada instrução. NumRowsAffected contém a soma das linhas afetadas por cada instrução INSERT, UPDATE e DELETE em uma linha de entrada.

Por exemplo, uma consulta contém as seguintes instruções:

```
DELETE from Employees WHERE Employee_ID = '101';  
SELECT Employee_ID, LastName from Employees WHERE Employee_ID = '103';  
INSERT into Employees (Employee_ID, LastName, Address)VALUES ('102', 'Gein', '38 Beach Rd')
```

A instrução DELETE afeta uma linha. A instrução SELECT não afeta nenhuma linha. A instrução INSERT afeta uma linha.

O Serviço de Integração retorna uma linha a partir da instrução DELETE. NumRowsAffected é igual a um. Retorna uma linha a partir da instrução SELECT e NumRowsAffected é zero. Retorna uma linha a partir da instrução INSERT, com NumRowsAffected igual a um.

A porta NumRowsAffected retorna zero quando todas as seguintes condições forem verdadeiras:

- O banco de dados é Informix.

- A transformação está executando no modo de consulta.
- A consulta não contém parâmetros.

Contagem Máxima das Linhas de Saída

Você pode limitar o número de linhas que transformação SQL retorna para as consultas SELECT. Configure a propriedade Contagem Máx das Linhas de Saída para limitar o número de linhas. Quando uma consulta contém múltiplas instruções SELECT, a transformação SQL limita os totais de linha de todas as instruções SELECT.

Por exemplo, você configurou a Contagem Máx das Linhas de Saída como 100. A consulta contém duas instruções SELECT:

```
SELECT * FROM table1; SELECT * FROM table2;
```

Se a primeira instrução SELECT retorna 200 linhas, e a segunda instrução SELECT retorna 50 linhas, a transformação SQL retorna 100 linhas a partir da primeira instrução SELECT. Ela não retorna nenhuma linha a partir da segunda declaração.

Para configurar linhas de saída ilimitadas, defina a propriedade Contagem Máx. de Linhas de Saída como zero.

Compreendendo Linhas de Erro

Quando encontra um erro de conexão ou de sintaxe, o Serviço de Integração retorna linhas de erro. A transformação SQL contém as seguintes portas padrão para emitir o texto de erro:

- **SQLException.** Retorna erros de banco de dados quando a transformação SQL é executada no modo de consulta.
- **ScriptError.** Retorna erros de banco de dados quando a transformação SQL é executada no modo de script.

Quando a consulta SQL contém erros de sintaxe, a porta de erro contém o texto de erro do banco de dados. Por exemplo, a seguinte consulta SQL gera uma linha de erro em um banco de dados Oracle:

```
SELECT Product_ID FROM Employees
```

A tabela Funcionários não contém Product_ID. O Serviço de Integração gera uma linha. A porta SQLException contém o texto do erro em uma linha:

```
ORA-0094: "Product_ID": invalid identifier Database driver error... Function Name:
Execute SQL Stmt: SELECT Product_ID from Employees Oracle Fatal Error
```

Quando uma consulta contém várias instruções e você configura a transformação SQL para prosseguir, mesmo havendo um erro SQL, a transformação SQL pode retornar linhas do banco de dados para uma instrução de consulta, mas retorna erros de banco de dados para outra instrução de consulta. A transformação SQL retorna qualquer erro de banco de dados em uma linha separada.

Quando você configura uma porta de passagem ou a porta NumRowsAffected, a transformação SQL retorna, pelo menos, uma linha para cada linha de origem. Quando uma consulta não retorna dados, a transformação SQL retorna os dados de passagem e os valores NumRowsAffected, mas retorna valores nulos nas portas de saída. Você pode remover linhas com valores nulos passando as linhas de saída por uma transformação Filtro.

As tabelas a seguir descrevem as linhas de saída que a transformação SQL retorna, dependendo do tipo de instruções de consulta.

A tabela a seguir descreve as linhas que a transformação SQL gera para as instruções de consulta UPDATE, INSERT ou DELETE:

Porta NumRowsAffected ou porta de passagem configurada	SQLError	Saída das linhas
Nenhuma porta configurada.	Não	Nenhuma linha.
Nenhuma porta configurada.	Sim	Uma linha com o erro na porta SQLError.
Qualquer uma das duas portas configuradas.	Não	Uma linha para cada instrução de consulta com os dados da coluna de passagem ou NumRowsAffected.
Qualquer uma das duas portas configuradas.	Sim	Uma linha com o erro na porta SQLError, a porta NumRowsAffected ou os dados da porta de passagem.

A tabela a seguir descreve o número de linhas de saída que a transformação SQL gera para instruções SELECT:

Porta NumRowsAffected ou porta de passagem configurada	SQLError	Saída das linhas
Nenhuma porta configurada.	Não	Nenhuma ou mais linhas, dependendo das linhas retornadas para cada instrução SELECT.
Nenhuma porta configurada.	Sim	Uma linha maior que a soma das linhas de saída para as instruções bem-sucedidas. A última linha contém o erro na porta SQLError.
Qualquer uma das duas portas configuradas.	Não	Uma ou mais linhas, dependendo das linhas retornadas para cada instrução SELECT. <ul style="list-style-type: none"> - Se NumRowsAffected estiver habilitada, cada linha conterá uma coluna NumRowsAffected com um valor zero. - Se uma porta de passagem estiver configurada, cada linha conterá os dados da coluna de passagem. Quando a consulta retorna várias linhas, os dados da coluna de passagem são duplicados em cada linha.
Qualquer uma das duas portas configuradas.	Sim	Uma ou mais linhas, dependendo das linhas retornadas para cada instrução SELECT. A última linha contém o erro na porta SQLError: <ul style="list-style-type: none"> - Quando a porta NumRowsAffected está ativada, cada linha contém uma coluna NumRowsAffected com o valor zero. - Se uma porta de passagem estiver configurada, cada linha conterá os dados da coluna de passagem. Quando a consulta retorna várias linhas, os dados da coluna de passagem são duplicados em cada linha.

A tabela a seguir descreve o número de linhas de saída que a transformação SQL gera para consultas DDL como CREATE, DROP ou TRUNCATE:

Porta NumRowsAffected ou porta de passagem configurada	SQLError	Saída das linhas
Nenhuma porta configurada.	Não	- Nenhuma linha.
Nenhuma porta configurada.	Sim	- Uma linha que contém o erro da porta SQLError.
Qualquer uma das duas portas configuradas.	Não	- Uma linha que inclui a coluna NumRowsAffected com valor zero e os dados da coluna de passagem.
Qualquer uma das duas portas configuradas.	Sim	- Uma linha com o erro na porta SQLError, a porta NumRowsAffected com valor zero e os dados da coluna de passagem.

Continuando no Erro de SQL

É possível escolher ignorar um erro de SQL em uma instrução ativando a opção Continuar no Erro de SQL na Linha. O Serviço de Integração continua executando o restante da instrução SQL da linha. O Serviço de Integração não gera um erro de linha. Entretanto, a porta SQLError contém a instrução SQL com falha e mensagens de erro. Há falha em uma sessão quando a contagem de erros da fila for maior que o limite de erros da sessão.

Por exemplo, uma consulta pode ter as instruções a seguir:

```
DELETE FROM Persons WHERE FirstName = 'Ed';  
  
INSERT INTO Persons (LastName, Address)VALUES ('Gein', '38 Beach Rd')
```

Se houver falha na instrução DELETE, a transformação SQL retorna uma mensagem de erro do banco de dados. O Serviço de Integração continua processando a instrução INSERT.

Sugestão: Desative a opção Continuar no Erro de SQL para depurar os erros do banco de dados. Caso contrário, talvez não seja possível associar erros às instruções de consulta que os ocasionaram.

Propriedades da Transformação SQL

Depois de criar a transformação SQL, você pode definir portas e configurar atributos nestas guias da transformação:

- **Portas.** Mostra atributos e portas da transformação criados por você na guia Portas SQL.
- **Propriedades.** Propriedades gerais da transformação SQL.
- **Configurações SQL.** Atributos exclusivos da transformação SQL.
- **Portas SQL.** Atributos e portas da transformação SQL.

Nota: Não é possível atualizar as colunas na guia Portas. Quando você define portas na guia Portas SQL, elas aparecem na guia Portas.

Guia Propriedades

Configure as propriedades gerais da transformação SQL na guia Propriedades. Algumas propriedades de transformação não se aplicam à transformação SQL ou não são configuráveis.

A tabela a seguir descreve as propriedades da transformação SQL:

Propriedade	Descrição
Localização do Tempo de Execução	<p>Local que contém a DLL ou a biblioteca compartilhada.</p> <p>Insira um caminho relativo ao nó do Serviço de Integração que executa a sessão de transformação SQL.</p> <p>Se esta propriedade estiver em branco, o Serviço de Integração usará a variável de ambiente definida no nó do Serviço de Integração para localizar a DLL ou a biblioteca compartilhada.</p> <p>Copie todas as DLLs ou bibliotecas compartilhadas para o local do tempo de execução ou para a variável de ambiente definida no nó do Serviço de Integração. O Serviço de Integração não carrega o procedimento quando não consegue localizar a DLL, a biblioteca compartilhada ou um arquivo especificado.</p>
Nível de rastreamento	<p>Define o volume de detalhes incluído no log da sessão quando você executa uma sessão que contém essa transformação. Quando você configura o nível de rastreamento da transformação SQL para Dados Detalhados, o Serviço de Integração grava cada consulta SQL que prepara no log da sessão.</p>
IsPartitionable	<p>Várias partições em um pipeline podem usar essa transformação. Use estas opções:</p> <ul style="list-style-type: none">- Não. A transformação não pode ser particionada. A transformação e outras transformações no mesmo pipeline estão limitadas a uma partição. Você deve escolher Não se a transformação processa todos os dados de entrada em conjunto, como limpeza de dados.- Localmente. A transformação pode ser particionada, mas o Serviço de Integração deve executar todas as partições da pipeline no mesmo nó. Escolha Localmente quando diferentes partições da transformação precisarem compartilhar objetos na memória.- Na Grade. A transformação pode ser particionada e o Serviço de Integração pode distribuir cada partição para nós diferentes. <p>O padrão é Não.</p>
Transformação Estratégia de Atualização	<p>A transformação define a estratégia de atualização das linhas de saída. Você pode ativar essa propriedade para transformações SQL no modo de consulta.</p> <p>O padrão é desativado.</p>
Escopo de Transformação	<p>O método pelo qual o Serviço de Integração aplica a lógica de transformação aos dados de entrada. Use estas opções:</p> <ul style="list-style-type: none">- Linha- Transação- Todas as Entradas <p>Defina o escopo de transação para transação quando usar o controle de transação no modo de consulta estática.</p> <p>O padrão é Linha para transformações no modo script.</p> <p>O padrão é Todos de Entrada para transformações no modo consulta.</p>

Propriedade	Descrição
Saída é repetível	Indica se a ordem dos dados de saída é consistente entre as execuções de sessão. <ul style="list-style-type: none"> - Nunca. A ordem dos dados de saída é inconsistente entre as execuções de sessão. - Baseada na Ordem de Entrada. A ordem de saída é consistente entre as execuções de sessão quando a ordem dos dados de entrada é consistente entre elas. - Sempre. A ordem dos dados de saída será consistente entre as execuções de sessão mesmo se a ordem de entrada for inconsistente entre elas. O padrão é Nunca.
Gerar Transação	A transformação gera linhas de transação. Ative essa propriedade para transformações SQL no modo de consulta que confirme dados em uma consulta SQL. O padrão é desativado.
Requer Thread Único por Partição	Indica se o Serviço de Integração processa cada partição de um procedimento com um thread.
Saída é determinística	A transformação gera dados de saída consistentes entre as execuções de sessão. Ative esta propriedade para executar a recuperação em sessões que usam essa transformação. O padrão é ativado.

Aviso: Se você configurar uma transformação como repetível e determinística, será sua responsabilidade garantir que os dados sejam repetíveis e determinísticos. Se você tentar recuperar uma sessão com transformações que não gerem os mesmos dados entre a sessão e a recuperação, o processo de recuperação poderá resultar em dados corrompidos.

Guia Configurações SQL

Configure atributos da transformação SQL na guia Configurações SQL. Os atributos SQL são exclusivos da transformação SQL.

A tabela a seguir lista os atributos que você pode configurar na guia Configuração SQL.

Opção	Descrição
Continuar no Erro de SQL na linha	Continua processando as instruções SQL restantes em uma consulta mesmo depois de ocorrer um erro de SQL.
Adicionar Porta de Saída Estática	Adiciona uma porta de saída NumRowsAffected. A porta retorna o número total de linhas de banco de dados afetadas pelas instruções de consulta INSERT, DELETE e UPDATE para uma linha de entrada.
AutoCommit	Habilita a confirmação automática para cada conexão com banco de dados. Cada instrução SQL em uma consulta define uma transação. Uma confirmação ocorre quando a instrução SQL é concluída ou quando a instrução seguinte é executada, o que acontecer primeiro
Contagem Máx. de Linhas de Saída	Define o número máximo de linhas que a transformação SQL pode emitir a partir de uma consulta SELECT. Para configurar linhas ilimitadas, defina a propriedade Contagem Máx. de Linhas de Saída como zero.
Localidade de Scripts	Identifica a página de código de um script SQL. Escolha a página de código na lista. O padrão é a localidade do sistema operacional.

Opção	Descrição
Usar Pool de Conexão	Mantém um pool de conexão para conexões com banco de dados. Você pode habilitar um pool para conexões dinâmicas somente.
Número Máximo de Conexões no Pool	O número máximo de conexões ativas disponíveis no pool de conexões. O mínimo é um. O máximo é 20. O padrão é 10.

Guia Portas SQL

Quando você cria uma transformação SQL, o Designer adiciona portas padrão, dependendo de como você configurou a transformação. Depois de criar a transformação, você pode adicionar portas na guia Portas SQL.

A tabela a seguir lista as portas da transformação SQL:

Porta	Tipo	Modo	Descrição
ScriptName	Entrada	Script	O nome do script a ser executado para cada linha.
ScriptError	Saída	Script	Erros de SQL que o banco de dados passa de volta à transformação quando o script falha.
ScriptResult	Saída	Script	Resultados da execução da consulta. Conterá PASSED quando a consulta for executada com êxito. Conterá FAILED quando a consulta não tiver êxito.
SQLException	Saída	Consulta	Erros de SQL que o banco de dados retorna para a transformação.
LogicalConnectionObject	Entrada	Consulta	Conexão dinâmica. O nome de uma conexão definida nas conexões do Workflow Manager.
String de Conexão	Entrada	Consulta Script	Apenas objeto de conexão. O nome do banco de dados e o nome do servidor de banco de dados.
DBUser	Entrada	Consulta Script	Apenas conexão completa. O nome do usuário com permissões para ler e gravar no banco de dados.
DBPasswd	Entrada	Consulta Script	Apenas conexão completa. A senha do banco de dados.
CodePage	Entrada	Consulta Script	Apenas conexão completa. A página de código usada pelo Serviço de Integração para ler ou gravar no banco de dados.

Porta	Tipo	Modo	Descrição
Opções avançadas	Entrada	Consulta Script	Apenas conexão completa. Atributos opcionais de conexão, como tamanho do pacote, período entre tentativas de conexão e ativar modo paralelo.
NumRowsAffected	Saída	Consulta Script	O número total de linhas de banco de dados afetadas pelas instruções de consulta INSERT, DELETE e UPDATE para uma linha de entrada.

Instruções SQL

A tabela a seguir lista as instruções que podem ser usadas com a transformação SQL:

Tipo de Instrução	Instrução	Descrição
Definição de Dados	ALTER	Modifica a estrutura do banco de dados.
Definição de Dados	COMMENT	Adiciona comentários ao dicionário de dados.
Definição de Dados	CREATE	Cria um banco de dados, tabela ou índice.
Definição de Dados	DROP	Exclui um banco de dados, tabela ou índice.
Definição de Dados	RENAME	Renomeia um objeto de banco de dados.
Definição de Dados	TRUNCATE	Remove todas as linhas de uma tabela.
Manipulação de Dados	CALL	Chama um subprograma Java ou PL/SQL.
Manipulação de Dados	DELETE	Exclui linhas de uma tabela.
Manipulação de Dados	EXPLAIN PLAN	Grava o plano de acesso de uma instrução às tabelas. Explicar do banco de dados.
Manipulação de Dados	INSERT	Insere linhas em uma tabela.
Manipulação de Dados	LOCK TABLE	Evita que processos de aplicativos concorrente usem ou alterem uma tabela.
Manipulação de Dados	MERGE	Atualiza uma tabela com dados de origem.
Manipulação de Dados	SELECT	Recupera dados do banco de dados.
Manipulação de Dados	UPDATE	Atualiza os valores de linhas de uma tabela.

Tipo de Instrução	Instrução	Descrição
Linguagem de Controle de Dados	GRANT	Concede privilégios a um usuário do banco de dados.
Linguagem de Controle de Dados	REVOKE	Retira privilégios de acesso de um usuário ao banco de dados.
Controle de Transações	COMMIT	Salva uma unidade de trabalho e realiza alterações no banco de dados para essa unidade de trabalho.
Controle de Transações	ROLLBACK	Reverte alterações ao banco de dados desde o COMMIT mais recente.

Criação de uma Transformação SQL

Você pode criar uma transformação SQL no Transformation Developer ou Mapping Designer.

Para criar uma transformação SQL:

1. Clique em Transformação > Criar.
2. Selecione a transformação SQL.
3. Insira um nome para a transformação.
A convenção de nomenclatura para uma transformação SQL é SQL_TransformationName.
4. Insira uma descrição para a transformação e clique em Criar.
5. Configure o modo da transformação SQL:
 - **Modo de consulta.** Configure uma transformação ativa que execute consultas de SQL dinâmicas.
 - **Modo de script.** Configure uma transformação passiva que execute scripts de SQL externos.
6. Configure o tipo de banco de dados ao qual a transformação SQL se conecta. Escolha o tipo de banco de dados na lista.
7. Configure as opções de conexão da transformação SQL:

Opção	Descrição
Conexão estática.	Use um objeto de conexão que você configura para a sessão. A transformação SQL conecta-se uma vez ao banco de dados durante uma sessão.
Conexão Dinâmica	Conecte-se aos bancos de dados, dependendo das informações de conexão que você transfere à transformação em um mapeamento. Quando você configura uma conexão dinâmica, escolha se a transformação exige um nome de objeto de conexão ou se exige todas as informações da conexão. O padrão é o objeto de conexão.

Opção	Descrição
Objeto de Conexão	Conexão dinâmica somente. Use uma porta LogicalConnectionObject para receber o nome do objeto de conexão. Defina o objeto de conexão nas conexões do Workflow Manager.
Informações Completas da Conexão	Conexão dinâmica somente. Use portas de entrada para receber todos os componentes da conexão.

8. Para configurar a transformação SQL para executar no modo passivo, selecione Transformação SQL Executará no Modo Passivo.
9. Clique em OK para configurar a transformação.
O Designer cria portas padrão na transformação, com base nas opções que você escolhe. Você não pode alterar a configuração, exceto o tipo de banco de dados.
10. Clique na guia Portas para adicionar portas à transformação. Adicione portas de passagem depois das portas de banco de dados.

CAPÍTULO 27

Usando a Transformação SQL em um Mapeamento

Este capítulo inclui os seguintes tópicos:

- [Visão Geral do Exemplo da Transformação SQL, 460](#)
- [Exemplo de atualização dinâmica, 460](#)
- [Exemplo de conexão dinâmica, 465](#)

Visão Geral do Exemplo da Transformação SQL

A transformação SQL processa consultas SQL midstream em um pipeline. A transformação processa scripts SQL externos ou consultas SQL que você cria em um editor SQL. Você pode passar as informações sobre conexão a banco de dados para a transformação SQL como dados de entrada em tempo de execução.

Este capítulo fornece dois exemplos que ilustram a funcionalidade da transformação SQL. Você utiliza os exemplos deste capítulo para criar e executar consultas SQL dinâmicas e para se conectar dinamicamente a bancos de dados. O capítulo fornece dados de amostra e descrições das transformações que você pode incluir em mapeamentos.

O capítulo fornece os exemplos a seguir:

- **Criação de uma consulta SQL dinâmica para atualizar um banco de dados.** O exemplo de atualização da consulta dinâmica mostra como atualizar preços de produtos em uma tabela, com base em um código de preço recebido de um arquivo de origem.
- **Configuração de uma conexão dinâmica com banco de dados.** O exemplo de conexão dinâmica mostra como se conectar a diferentes bancos de dados, com base no valor de uma localização de cliente em uma linha de origem.

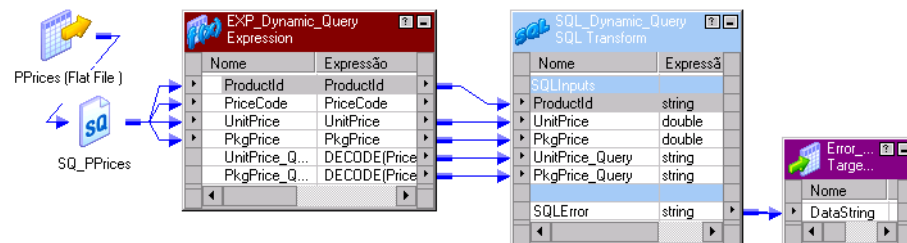
Exemplo de atualização dinâmica

Este exemplo demonstra como configurar uma transformação de Expressão e uma transformação de SQL para gerar consultas SQL com base no valor de uma coluna em um arquivo de origem.

Neste exemplo, temos uma tabela de banco de dados que contém preços de produtos. É necessário atualizar os preços a partir de um arquivo de transações. Cada linha de transação atualiza os preços de atacado, varejo ou de manufatura no banco de dados com base em uma coluna de códigos de preços.

O arquivo de origem é um arquivo simples. É possível configurar uma transformação de Expressão para que retorne os nomes de coluna a serem atualizados, com base no valor de uma coluna de código de preço em cada linha de origem. A transformação de Expressão transfere os nomes de coluna à transformação SQL. A transformação SQL executa uma consulta SQL dinâmica que atualiza colunas na tabela Prod_Cost com base nos nomes de coluna que recebe. A transformação de SQL retorna erros do banco de dados para o destino Error_File.

A figura a seguir demonstra como a transformação de Expressão transfere nomes de coluna à transformação SQL:



O mapeamento contém os seguintes componentes:

- **Definição da origem de preços.** O arquivo simples PPrices contém uma ID de produto, preço de pacote, preço unitário e código de preço. O código de preço define se o preço de pacote e o preço unitário são preços de atacado, varejo ou de manufatura.
- **Definição do arquivo simples Error_File.** O destino contém o campo Datastring que recebe erros de banco de dados da transformação SQL.
- **Transformação Exp_Dynamic_Expression.** A transformação de Expressão define quais nomes da coluna Prod_Cost devem ser atualizados com base no valor da coluna PriceCode. Ela retorna os nomes de coluna nas portas UnitPrice_Query e PkgPrice_Query.
- **Transformação SQL_Dynamic_Query.** A transformação SQL tem uma consulta SQL dinâmica para atualizar uma coluna UnitPrice e uma coluna PkgPrice na tabela Prod_Cost. Ela atualiza as colunas nomeada nas colunas UnitPrice_Query e PkgPrice_Query.

Nota: O mapeamento não contém uma definição de tabela relacional para a tabela Prod_Cost. A transformação SQL tem uma conexão estática com o banco de dados que contém a tabela Prod_Cost. A transformação gera as instruções de SQL para atualizar os preços unitários e preços de pacote na tabela.

Definição do Arquivo de Origem

O arquivo de transação é uma origem de arquivo simples que contém os preços para atualização no banco de dados. Cada linha tem um código que define se os preços são de atacado, varejo ou fabricante. O nome do arquivo de origem é PPrices.dat.

Você pode criar um arquivo PPrices.dat no Srcfiles que contém as seguintes linhas:

```
100,M,100,110
100,W,120,200
100,R,130,300
200,M,210,400
200,W,220,500
200,R,230,600
300,M,310,666
300,W,320,680
300,R,330,700
```

Você pode importar o arquivo PPrices.dat para criar a definição da origem do PPrices.dat no repositório.

O arquivo PPrices contém as seguintes colunas:

Coluna	Tipo de dados	Precisão	Descrição
ProductID	String	10	Um número exclusivo que identifica o produto para atualizar.
PriceCode	String	2	M, W ou R. Define se os preços são Fabricado, Atacado ou Varejo.
UnitPrice	Número	10	O preço de cada unidade do produto.
PkgPrice	Número	10	O preço de um pacote do produto.

Criação de uma Definição de Destino

Error_File é uma definição de destino de arquivo simples que recebe mensagens de erro do banco de dados da transformação SQL.

A definição Error_File tem a seguinte porta de entrada:

Nome da Porta	Tipo de Dados	Precisão	Escala
DataString	String	4000	0

Crie a definição de destino no Target Designer.

Criação da Tabela de Banco de Dados

A transformação SQL escreve os preços de produto na tabela relacional Prod_Cost. A tabela Prod_Cost contém preços de unidade e pacote para cada produto. Ela armazena o preço de atacado, varejo e de fabricação por unidade e pacote.

Você não criará uma definição de destino relacional para a tabela Prod_Cost no repositório. A transformação SQL gera as instruções de SQL para atualizar a tabela no banco de dados.

A tabela Prod_Cost contém as seguintes colunas:

Tipo de custos	Tipo de dados	Descrição
ProductID	varchar	Um número exclusivo que identifica o produto para atualizar.
WUnitPrice	número	Preço da unidade para o atacado.
WPkgPrice	número	Preço do pacote para o atacado.
RUnitPrice	número	Preço da unidade para o varejo.
RPkgPrice	número	Preço do pacote de varejo.
MUnitPrice	número	Preço da unidade para fabricantes.
MPkgPrice	número	Preço do pacote de fabricantes.

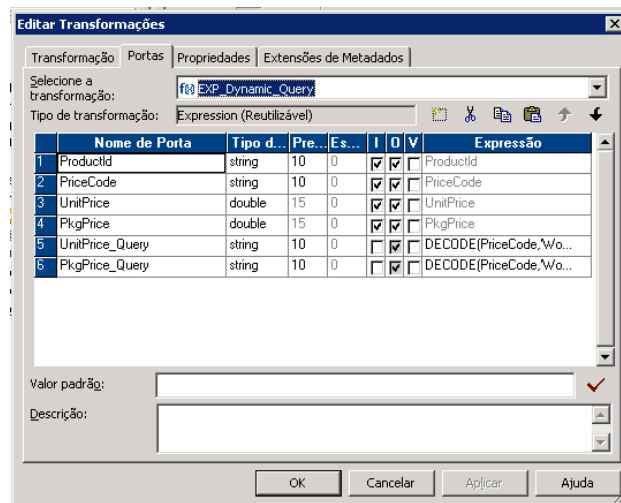
As seguintes instruções de SQL criam a tabela Prod_Cost e três linhas de produto em um banco de dados do Oracle:

```
Create table Prod_Cost (ProductId varchar (10), WUnitPrice number, WPkgPrice number,
RUnitPrice number, RPkgPrice number, MUnitPrice number, MPkgPrice number );
insert into Prod_Cost values ('100',0,0,0,0,0,0);
insert into Prod_Cost values ('200',0,0,0,0,0,0);
insert into Prod_Cost values ('300',0,0,0,0,0,0);
commit;
```

Configuração da Transformação Expressão

A transformação Expressão tem uma porta de entrada/saída para cada coluna de origem. Ela transfere nomes de coluna para a transformação SQL com base no valor da coluna PriceCode.

A figura a seguir mostra as portas da transformação Expressão que retornam nomes de coluna:



A transformação SQL tem as seguintes colunas que contêm os resultados de expressões:

- **UnitPrice_Query.** Retorna o nome de coluna "MUnitprice," "RUnitPrice" ou "WUnitprice" com base no código de preço ser "M," "R" ou "W."

```
DECODE(PriceCode,'M', 'MUnitPrice','R', 'RUnitPrice','W', 'WUnitPrice')
```

- **PkgPrice_Query.** Retorna o nome de coluna "MPkgPrice", "RPkgPrice" ou "WPkgPrice", com base no código de preço ser "M," "R" ou "W."

```
DECODE(PriceCode,'M', 'MPkgPrice','R', 'RPkgPrice','W', 'WPkgPrice')
```

Definindo a Transformação SQL

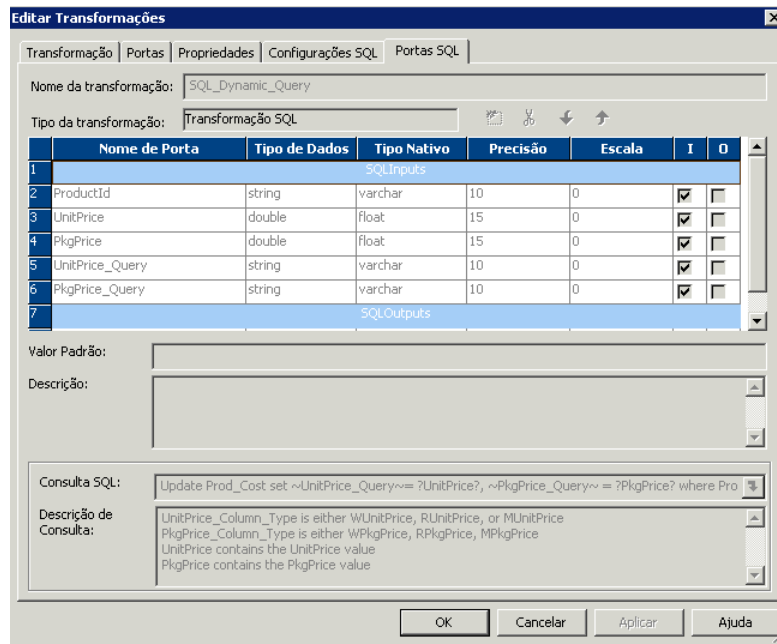
A transformação SQL executa uma consulta SQL dinâmica que insere os dados de preço unitário e de preço de pacote na tabela Prod_Cost. A transformação SQL recebe os nomes de coluna para atualizar nas portas UnitPrice_Query e PkgPrice_Query.

Ao criar uma transformação SQL, você define o modo de transformação, o tipo de banco de dados e o tipo de conexão. Você não pode alterar o modo ou tipo de conexão depois de criar a transformação.

Crie uma transformação SQL com as seguintes propriedades:

- **Modo de Consulta.** A transformação de SQL executa consultas SQL dinâmicas.
- **Conexão estática.** A transformação SQL se conecta uma vez ao banco de dados com o objeto de conexão que você definir no Workflow Manager.

A figura a seguir mostra a guia Portas da transformação SQL com a consulta SQL e descrição da consulta:



A transformação SQL tem uma consulta SQL dinâmica que atualiza uma das colunas UnitPrice e uma das colunas PkgPrice da tabela Prod_Cost com base nos nomes de coluna que recebe nas portas UnitPrice_Query e PkgPrice_Query.

A transformação SQL tem a seguinte consulta:

```
Update Prod_Cost set ~UnitPrice_Query~ = ?UnitPrice?, ~PkgPrice_Query~ = ?PkgPrice?
where ProductId = ?ProductId?;
```

A transformação de SQL substitui as variáveis de string UnitPrice_Query e PkgPrice_Query pelos nomes de coluna a serem atualizados.

A transformação SQL une os parâmetros ProductId, UnitPrice e PkgPrice na consulta com dados que recebe nas portas correspondentes.

Por exemplo, a seguinte linha de origem contém um preço unitário e um preço de pacote do produto 100:

```
100,M,100,110
```

Quando o PriceCode for "M," os preços serão preços de manufatura. A transformação de Expressão transfere os nomes de coluna MUnitprice e MPkgPrice à transformação SQL para que os atualize.

A transformação SQL executa a seguinte consulta:

```
Update Prod_Cost set MUnitprice = 100, MPkgPrice = 110 where ProductId = '100';
```

A seguinte linha de origem contém preços de atacado do produto 100:

```
100,W,120,200
```

A transformação de Expressão transfere os nomes de coluna WUnitprice e WPkgPrice para a transformação SQL. A transformação SQL executa a seguinte consulta:

```
Update Prod_Cost set WUnitprice = 120, WPkgPrice = 200 where ProductId = '100';
```


Configurando os Atributos da Sessão

Quando você configura a sessão, configure o arquivo de origem, o de destino e a conexão do banco de dados:

Atributo	Valor
Nome do Arquivo de Origem	PPrices.dat
Nome do Arquivo de Saída	ErrorFile.dat
Conexão relacional SQL_Dynamic_Query	Conexão com o banco de dados de teste que contém a tabela Prod_Cost.

Resultados de Dados de Destino

Quando você executa uma sessão com os dados de amostra, o Serviço de Integração preenche a tabela de destino Prod_Cost com estes dados:

ProductID	WUnitPrice	WPkgPrice	RUnitPrice	RPkgPrice	MUnitPrice	MPkgPrice
100	120	200	130	300	100	110
200	220	500	230	600	210	400
300	320	680	330	700	310	666

Se o banco de dados retornar algum erro para a transformação SQL, o destino Error_File conterá o texto do erro.

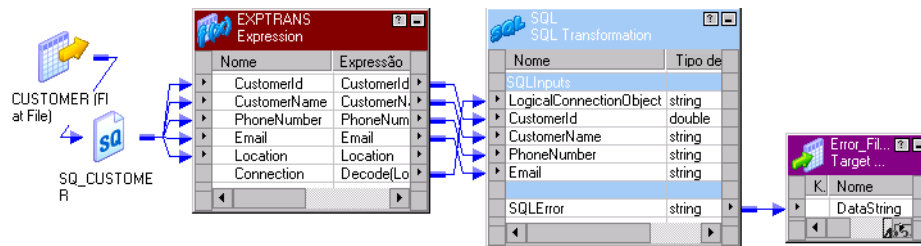
Exemplo de conexão dinâmica

O exemplo da transformação SQL de conexão dinâmica mostra como conectar-se dinamicamente a um banco de dados, com base nos dados do arquivo de origem.

Neste exemplo, você tem um banco de dados do cliente para os Estados Unidos, Reino Unido e Canadá. Você precisa inserir os dados do cliente a partir de um arquivo de transação em um banco de dados, com base na localização do cliente.

A transformação Expressão retorna um nome de objeto de conexão do banco de dados com base no valor da coluna do local. A transformação de Expressão transfere um nome de objeto de conexão para a porta LogicalConnectionObject da transformação SQL. A transformação SQL conecta-se ao banco de dados com base no valor da coluna LogicalConnectionObject.

A figura a seguir mostra a transformação Expressão e a transformação SQL em um mapeamento:



O mapeamento contém os seguintes componentes:

- **Definição da origem do cliente.** Uma definição da origem de arquivo simples que inclui informações do cliente. O local do cliente determina a qual banco de dados a transformação SQL se conecta quando insere os dados do cliente.
- **Definição de destino Error_File.** O destino contém um campo Datastring que recebe erros de banco de dados da transformação SQL.
- **Transformação Exp_Dynamic_Connection.** A transformação Expressão define a qual banco de dados irá se conectar, com base no valor da coluna Local. A transformação Expressão retorna o nome do objeto de conexão na porta Conexão. Um objeto de conexão é uma conexão de banco de dados definida no Workflow Manager.
- **Transformação SQL_Dynamic_Connection.** A transformação SQL recebe um nome do objeto de conexão na LogicalConnectionPort. Ela se conecta ao banco de dados e insere os dados do cliente no banco de dados.

Definição do Arquivo de Origem

O arquivo de transação é uma origem de arquivo simples que contém o cliente para adicionar ao banco de dados. Cada linha tem um local que define o país onde o cliente mora. O nome do arquivo de origem é Customer.dat.

Um registro de arquivo do Cliente contém os seguintes campos:

Nome da porta	Tipo de dados	Precisão	Escala	Descrição
CustomerId	Número	10	0	Número de identificação exclusiva do cliente.
CustomerName	String	25	0	Nome do cliente.
PhoneNumber	String	15	0	O número de telefone inclui o código de área e sete dígitos sem espaços.
E-mail	String	25	0	Endereço de e-mail do cliente.
Location	String	10	0	Local do cliente. Os valores são US, UK, CAN.

Você pode criar um arquivo Customer.dat no Srcfiles que contém as seguintes linhas:

```
1,John Smith,6502345677,jsmith@catgary.com,US
2,Nigel Whitman,5123456754,nwhitman@nwhitman.com,UK
3,Girish Goyal,5674325321,ggoyal@netcompany.com,CAN
4,Robert Gregson,5423123453,rgregson@networkmail.com,US
```

Você pode importar o arquivo Customer.dat para criar a definição da origem do Cliente no repositório.

Criação de uma Definição de Destino

Error_File é uma definição de destino de arquivo simples que recebe mensagens de erro do banco de dados da transformação SQL.

A definição Error_File inclui a seguinte porta de entrada:

Nome da Porta	Tipo de Dados	Precisão	Escala
DataStream	String	4000	0

Crie a definição de destino no Target Designer.

Criação de tabelas de banco de dados

Para este exemplo, você precisa criar três bancos de dados do Oracle, um para os EUA, um para o Reino Unido e um canadense. Cada banco de dados possui uma tabela Cust que contém os dados do cliente dessa região.

As seguintes instruções de SQL criam uma tabela Cust no banco de dados do Oracle:

```
Create table Cust (CustomerId number, CustomerName varchar2(25), PhoneNumber  
varchar2(15), Email varchar2(25));
```

Nota: Este exemplo inclui três bancos de dados para ilustrar as conexões de bancos de dados dinâmicos. Se as tabelas estiverem no mesmo banco de dados, use uma conexão estática para melhorar o desempenho.

Criação de Conexões ao Banco de Dados

Use o Workflow Manager para criar conexões com os bancos de dados US, UK e CAN.

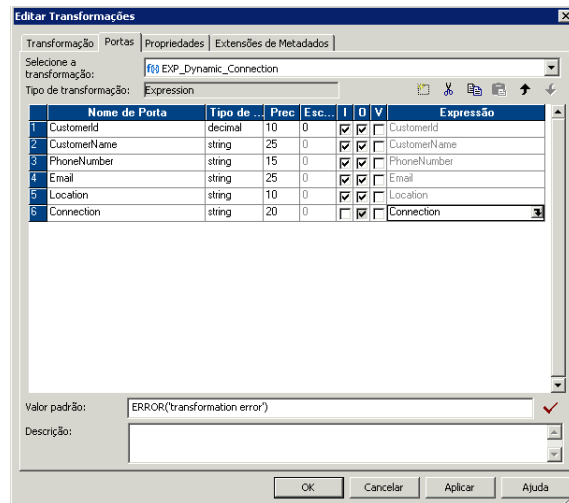
Neste exemplo, a transformação SQL conecta-se aos bancos de dados usando os seguintes nomes de conexão:

Banco de Dados	Nome do Objeto de Conexão
US	DBORA_US
UK	DBORA_UK
CAN	DBORA_CAN

Configuração da Transformação Expressão

A transformação Expressão tem uma porta de entrada/saída para cada coluna de origem. A transformação retorna um nome de objeto de conexão na porta Conexão, com base nos resultados de uma expressão.

A figura a seguir mostra a guia Portas na transformação de Expressão e as colunas de porta de entrada/saída para cada coluna de origem:



A porta Conexão tem a expressão a seguir:

```
Decode(Location, 'US', 'DBORA_US', 'UK', 'DBORA_UK', 'CAN', 'DBORA_CAN', 'DBORA_US')
```

A expressão retorna um nome de objeto de conexão dependendo do fato de o local ser US, UK ou CAN. Quando o local não corresponde a um local na expressão, o nome do objeto de conexão usa o padrão "DBORA_US".

Por exemplo, as seguintes informações do cliente da linha de origem para um cliente nos Estados Unidos:

```
1,John Smith,6502345677,jsmith@catgary.com,US
```

Quando o local do cliente é "US", a transformação Expressão retorna o nome de objeto de conexão "DBORA_US". A transformação Expressão transfere "DBORA_US" para a porta LogicalConnectionObject da transformação SQL.

Definindo a Transformação SQL

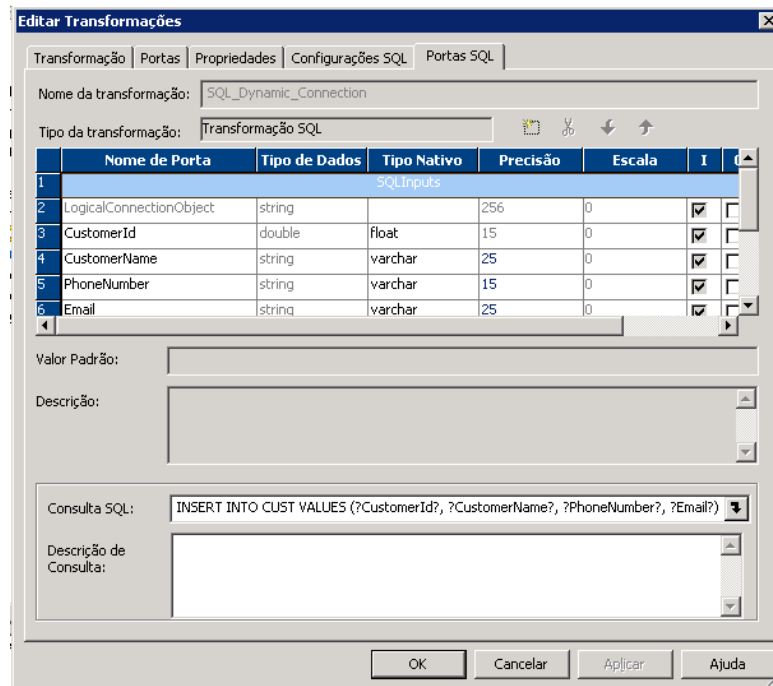
A transformação SQL conecta-se ao banco de dados e executa uma consulta de SQL dinâmica que insere os dados do cliente na tabela CUST.

Ao criar uma transformação SQL, você define o modo de transformação, o tipo de banco de dados e o tipo de conexão. Você não pode alterar o modo ou tipo de conexão depois de criar a transformação.

Crie uma transformação SQL com as seguintes propriedades:

- **Modo de Consulta.** A transformação de SQL executa consultas SQL dinâmicas.
- **Conexão Dinâmica.** A transformação SQL conecta-se ao banco de dados dependendo das informações de conexão que você passa para a transformação em um mapeamento.
- **Objeto de Conexão.** A transformação de SQL tem uma porta LogicalConnectionObject que recebe o nome do objeto de conexão. O objeto de conexão deve ser definido nas conexões do Workflow Manager.

A seguinte figura mostra as portas na transformação SQL:



A transformação SQL recebe um nome do objeto de conexão na porta LogicalConnectionObject. Ela conecta-se ao banco de dados com o nome do objeto de conexão a cada vez que processa uma linha.

A transformação tem a seguinte consulta de SQL dinâmica para inserir os dados do cliente em uma tabela CUST:

```
INSERT INTO CUST VALUES (?CustomerId?, ?CustomerName?, ?PhoneNumber?, ?Email?);
```

A transformação de SQL substitui os parâmetros da consulta pelos dados do cliente, a partir das portas de entrada da transformação. Por exemplo, a seguinte linha de origem contém informações para o cliente número 1:

```
1, John Smith, 6502345677, jsmith@catgary.com, US
```

A transformação de SQL conecta-se ao banco de dados com o objeto de conexão DBORA_US. Ele executa a seguinte consulta SQL:

```
INSERT INTO CUST VALUES (1, 'John Smith', '6502345677', 'jsmith@catgary.com');
```

Configurando os Atributos da Sessão

Configure a sessão para acessar a origem e o destino a seguir:

Atributo	Valor
Nome do Arquivo de Origem	Customer.dat
Nome do Arquivo de Saída	Error_File.dat

Nota: Não configure uma conexão de banco de dados para a sessão. A transformação SQL pode se conectar a um banco de dados diferente a cada vez que processar uma linha de entrada.

Resultados de Dados de Destino

Quando você executa uma sessão com os dados de amostra, o Serviço de Integração preenche a tabela Cust no banco de dados Estados Unidos, Reino Unido ou Canadá.

O banco de dados Estados Unidos contém as seguintes linhas:

CustomerID	CustomerName	PhoneNumber	E-mail
1	John Smith	6502345677	jsmith@catagary.com
4	Robert Gregson	5423123453	rgregson@networkmail.com

O banco de dados Reino Unido contém as seguintes linhas:

CustomerID	CustomerName	PhoneNumber	E-mail
2	Nigel Whitman	5123456754	nwhitman@nwhitman.com

O banco de dados canadense contém a seguinte linha:

CustomerID	CustomerName	PhoneNumber	E-mail
3	Girish Goyal	5423123453	ggoyal@netcompany.com

Se o banco de dados retornar algum erro para a transformação SQL, o destino Error_File conterá o texto do erro.

CAPÍTULO 28

Transformação Procedimento Armazenado

Este capítulo inclui os seguintes tópicos:

- [Visão Geral da Transformação Procedimento Armazenado, 471](#)
- [Usando um Procedimento Armazenado em um Mapeamento, 475](#)
- [Gravando um Procedimento Armazenado, 476](#)
- [Criação de uma Transformação Procedimento Armazenado, 478](#)
- [Configurando uma transformação conectada, 482](#)
- [Configurando uma transformação desconectada, 483](#)
- [Tratamento de erros, 487](#)
- [Bancos de Dados Suportados, 488](#)
- [Regras de Expressão, 489](#)
- [Dicas para Transformações Procedimento Armazenado, 490](#)
- [Solução de Problemas de Transformações Procedimento Armazenado, 490](#)

Visão Geral da Transformação Procedimento Armazenado

Uma transformação Procedimento Armazenado é uma ferramenta importante para preencher e manter bancos de dados. Administradores de bancos de dados criam procedimentos armazenados para automatizar tarefas complicadas demais para instruções SQL padrão. A transformação de Procedimento Armazenado é uma transformação passiva. Você pode configurar uma transformação de Procedimento Armazenado conectada ou desconectada.

Um procedimento armazenado é uma coleção pré-compilada de Transact-SQL, PL-SQL ou outras instruções de procedimento de banco de dados e instruções de fluxo opcional, semelhante a um script executável. Procedimentos armazenados são guardados e executados no banco de dados. Você pode executar um procedimento armazenado com a instrução EXECUTE SQL na ferramenta do cliente do banco de dados como instruções SQL. Diferente do SQL padrão, contudo, procedimentos externos permitem variáveis definidas pelo usuário, instruções condicionais e outros recursos de programação poderosos.

Nem todos os bancos de dados dão suporte a procedimentos armazenados e a sintaxe destes varia de acordo com o banco de dados. Você pode usar procedimentos armazenados para realizar as seguintes tarefas:

- Verifique o status do banco de dados de destino antes de carregar dados para ele.
- Determine se há espaço suficiente no banco de dados.
- Execute um cálculo especializado.
- Descarte e recrie índices.

Programadores e desenvolvedores de bancos de dados usam procedimentos armazenados em várias tarefas no banco de dados, pois eles permitem flexibilidade maior que instruções SQL. Procedimentos armazenados proporcionam ainda o tratamento e log necessário a tarefas cruciais. O desenvolvedor cria procedimentos armazenados no banco de dados usando ferramentas fornecidas no banco de dados.

O procedimento armazenado deve estar no banco de dados antes de criar a transformação Procedimento Armazenado e pode estar em uma origem, destino ou qualquer banco de dados com conexão válida com o Serviço de Integração.

Você pode usar um procedimento armazenado para realizar consultas ou cálculos que de outra forma fariam parte do mapeamento. Por exemplo, se você já tiver um procedimento armazenado testado para calcular impostos sobre vendas, poderá realizar cálculos através dele, em vez de recriar o mesmo cálculo em uma transformação Expressão.

Dados de Entrada e Saída

Um dos recursos mais úteis dos procedimentos armazenados é a capacidade de enviar e receber dados do procedimento armazenado. Existem três tipos de dados que são transferidos entre o Serviço de Integração e o procedimento armazenado:

- Parâmetros de entrada/saída
- Valores retornados
- Códigos de status

Existem algumas limitações ao transferir os dados, dependendo da implementação do banco de dados, e elas são discutidas ao longo deste capítulo. Além disso, nem todos os procedimentos armazenados enviam e recebem dados. Por exemplo, se você escreve um procedimento armazenado para reconstruir um índice de banco de dados ao término de uma sessão, não poderá receber dados, uma vez que a sessão já terminou.

Parâmetros de Entrada/Saída

No caso de vários procedimentos armazenados, é fornecido um valor e recebido um valor de volta. Esses valores são conhecidos como parâmetros de entrada e de saída. Por exemplo, um procedimento armazenado de cálculo de imposto sobre vendas pode usar um único parâmetro de entrada, como o preço de um item. Após efetuar o cálculo, o procedimento armazenado retorna dois parâmetros de entrada: a quantidade de impostos e o custo total do item com os impostos.

A transformação do Procedimento Armazenado envia e recebe parâmetros de entrada e de saída usando portas, variáveis ou inserindo um valor em uma expressão, como 10 ou SALES.

Valores de Retorno

A maioria dos bancos de dados fornece um valor de retorno após executar um procedimento armazenado. Dependendo da implementação do banco de dados, este valor pode ser definível pelo usuário, o que significa que ele pode funcionar da mesma maneira que um parâmetro de saída individual, ou ele pode retornar somente um valor inteiro.

A transformação Procedimento Armazenado captura valores de retorno de uma maneira semelhante aos parâmetros de entrada/saída, dependendo de como os parâmetros de entrada/saída são capturados. Em algumas instâncias, pode ser capturado somente um parâmetro ou valor de retorno.

Se um procedimento armazenado retornar um conjunto de resultados, em vez de um único valor de retorno, a transformação Procedimento Armazenado pegará somente o primeiro valor retornado no procedimento.

Nota: Uma função armazenada Oracle é muito parecida com um procedimento armazenado Oracle, mas ela oferece suporte a parâmetros de saída e a valores de retorno. Neste capítulo, quaisquer instruções sobre procedimentos armazenados também se aplicam a funções armazenadas, a menos que seja especificado o contrário.

Códigos de Status

Códigos de status proporcionam ao Serviço de Integração tratamento de erros durante um fluxo de trabalho. O procedimento armazenado edita um código de status que notifica se o procedimento armazenado foi concluído com êxito. Não é possível ver esse valor. O Serviço de Integração o usa para determinar se deve ou não continuar a executar a sessão. Você configura as opções no Workflow Manager para continuar ou interromper a sessão no caso de erro em um procedimento armazenado.

Conectada ou desconectada

Os procedimentos armazenados são executados em modo conectado ou não. O modo que você usa depende daquilo que o procedimento armazenado faz e de como você planeja utilizá-lo em uma sessão. É possível configurar transformações de Procedimento Armazenado conectadas ou não em um mapeamento.

- **Conectada.** O fluxo de dados por meio de um mapeamento em modo conectado também passa pela Transformação Procedimento Armazenado. Todos os dados que entram na transformação por meio das portas de entrada afetam o procedimento armazenado. É necessário usar uma transformação conectada de Procedimento Armazenado quando for necessário que dados de uma porta de entrada sejam enviados na qualidade de parâmetro de entrada ao procedimento armazenado, ou o resultado de um procedimento armazenado enviado na qualidade de um parâmetro de saída a outra transformação.
- **Desconectada.** A Transformação Procedimento Armazenado não está diretamente conectada ao fluxo do mapeamento. Ela é executada antes ou depois da sessão, ou então é chamada por uma expressão em outra transformação no mapeamento.

A tabela a seguir compara transformações que estão ou não conectadas:

Se você desejar	Usar este modo
Executar um procedimento armazenado antes ou depois de uma sessão.	Não Conectada
Executar um procedimento armazenado uma vez durante um mapeamento, como pré ou pós-sessão.	Não Conectada
Executar um procedimento armazenado a cada vez que uma linha passar por uma Transformação Procedimento Armazenado.	Conectada ou Não Conectada
Executar um procedimento armazenado com base em dados que passam pelo mapeamento, como quando uma porta específica não contém um valor nulo.	Não Conectada
Transferir parâmetros para o procedimento armazenado e receber um único parâmetro de saída.	Conectada ou Não Conectada

Se você deseja	Usar este modo
Transfira parâmetros para o procedimento armazenado e receba parâmetros de saída. Nota: Para obter vários parâmetros de saída a partir de uma Transformação Procedimento Armazenado não conectada, é necessário criar variáveis para cada parâmetro de saída.	Conectada ou Não Conectada
Executar procedimentos armazenados aninhados.	Não Conectada
Chamar várias vezes dentro de um mapeamento.	Não Conectada

TÓPICOS RELACIONADOS:

- [“Configurando uma transformação desconectada” na página 483](#)
- [“Configurando uma transformação conectada” na página 482](#)

Especificando Quando o Procedimento Armazenado é Executado

Além de especificar o modo da transformação Procedimento Armazenado, você também especifica quando ela é executada. No caso do procedimento armazenado não conectado descrito acima, a transformação Expressão faz referência ao procedimento armazenado, o que significa que ele é executado toda vez que uma linha passa por essa transformação. No entanto, se nenhuma transformação fizer referência à transformação Procedimento Armazenado, você terá a opção de executar o procedimento armazenado uma vez antes ou depois da sessão.

A lista a seguir descreve as opções para executar uma transformação Procedimento Armazenado:

- **Normal.** O procedimento armazenado é executado onde a transformação existe no mapeamento, em uma base linha por linha. Isso é útil para chamar o procedimento armazenado para cada linha de dados que passa pelo mapeamento, como executar um cálculo em uma porta de entrada. Os procedimentos armazenados conectados são executados somente no modo normal.
- **Pré-carregamento da origem.** Antes de a sessão recuperar dados na origem, o procedimento armazenado é executado. É útil para verificar a existência de tabelas ou fazer associações de dados em uma tabela temporária.
- **Pós-carregamento da origem.** Depois que a sessão recupera os dados da origem, o procedimento armazenado é executado. É útil para remover tabelas temporárias.
- **Pré-carregamento do destino.** Antes de a sessão enviar dados ao destino, o procedimento armazenado é executado. É útil para verificar tabelas de destino ou o espaço em disco no sistema de destino.
- **Pós-carregamento do destino.** Após a sessão enviar dados ao destino, o procedimento armazenado é executado. É útil para recriar índices do banco de dados.

Você pode executar mais de uma transformação Procedimento Armazenado em diferentes modos no mesmo mapeamento. Por exemplo, um procedimento armazenado de origem pré-carregada pode verificar a integridade da tabela; um procedimento armazenado normal pode preencher a tabela; e um procedimento armazenado pós-carregado pode reconstruir índices no banco de dados. No entanto, você não pode executar a mesma instância de uma transformação Procedimento Armazenado no modo conectado e não conectado em um mapeamento. Para isso, você precisa criar instâncias diferentes da transformação.

Se o mapeamento chamar mais de um procedimento armazenado pré ou pós-carregado em um mapeamento, o Serviço de Integração irá executá-los na ordem de execução que você especificar no mapeamento.

O Serviço de Integração executa cada procedimento armazenado usando a conexão com banco de dados que você especifica nas propriedades de transformação. Ele abre a conexão com o banco de dados quando

encontra o primeiro procedimento armazenado. Ela permanece aberta até o Serviço de Integração terminar de processar todos os procedimentos armazenados dessa conexão. O Serviço de Integração fecha as conexões com banco de dados e abre uma nova quando encontra um procedimento armazenado que use outra conexão com banco de dados.

Para executar vários procedimentos armazenados que utilizem a mesma conexão com banco de dados, configure-os para serem executados um após o outro. Se você não configurá-los para serem executados dessa forma, poderá obter resultados inesperados no destino. Por exemplo, você tem dois procedimentos armazenados: Procedimento Armazenado A e Procedimento Armazenado B: o A começa uma transação e o B confirma-a. Se você executar o Procedimento Armazenado C antes do B, usando outra conexão com o banco de dados, o Procedimento Armazenado B não poderá confirmar a transação, porque o Serviço de Integração fechará a conexão com o banco de dados quando executar o Procedimento Armazenado C.

Utilize as diretrizes a seguir para executar vários procedimentos armazenados dentro de uma conexão com banco de dados:

- Os procedimentos armazenados utilizam a mesma cadeia de conexão com banco de dados definida em suas propriedades.
- Você define os procedimentos armazenados para serem executados um após o outro.
- Os procedimentos armazenados têm o mesmo tipo de procedimento armazenado:
 - Pré-carregamento da origem
 - Pós-carregamento da origem
 - Pré-carregamento do destino
 - Pós-carregamento do destino

Usando um Procedimento Armazenado em um Mapeamento

Você precisa executar várias etapas para usar uma transformação Procedimento Armazenado em um mapeamento. Como o procedimento armazenado já existe no banco de dados, além de configurar o mapeamento e a sessão, você também deve configurar o procedimento armazenado no banco de dados.

Para usar uma transformação Procedimento Armazenado, execute as etapas a seguir:

1. Crie o procedimento armazenado no banco de dados.

Antes de usar o Designer para criar a transformação, você precisa criar o procedimento armazenado no banco de dados. Você também deve testar o procedimento armazenado por meio das ferramentas de cliente de banco de dados fornecidas.
2. Importe ou crie a transformação de Procedimento Armazenado.

Utilize o Designer para importar ou criar a transformação Procedimento Armazenado, fornecendo portas para quaisquer valores de retorno e de entrada/saída necessários.
3. Determine se a transformação deverá ser usada como conectada ou não conectada.

Antes de configurar a transformação, determine como o procedimento armazenado está relacionado ao mapeamento.
4. Se estiver conectado, mapeie as portas de entrada e saída apropriadas.

Utilize as transformações Procedimento Armazenado conectado da mesma forma que faria com a maioria das outras transformações. Arraste as portas de fluxo de entrada apropriadas até a transformação e crie mapeamentos das portas de saída até outras transformações.

5. Se for não conectado, configure o procedimento armazenado para ser executado antes ou depois da sessão, ou configure-o para ser executado a partir de uma expressão em outra transformação.

Como os procedimentos armazenados podem ser executados antes ou depois da sessão, talvez seja necessário especificar quando a transformação não conectada deverá ser executada. Por outro lado, se o procedimento armazenado for chamado a partir de outra transformação, você grava a expressão em outra transformação, que chama o procedimento armazenado. A expressão pode conter variáveis. Além disso, ela pode ou não incluir um valor de retorno.

6. Configure a sessão.

A área de propriedades de sessão no Workflow Manager inclui opções para tratamento de erros durante a execução de procedimentos armazenados e várias opções de substituição SQL.

Gravando um Procedimento Armazenado

Grave instruções SQL para criar um procedimento armazenado no banco de dados e você poderá adicionar outras instruções Transact-SQL e funções específicas de banco de dados. Elas podem incluir tipos de dados definidos pelo usuário e instruções de pedidos de execução.

Exemplo de Procedimento Armazenado

No exemplo a seguir, o banco de dados de origem contém um procedimento armazenado que pega um parâmetro de entrada de um número de ID de um funcionário e retorna um parâmetro de saída do nome do funcionário. Além disso, o valor de retorno 0 é retornado como uma notificação de que o procedimento armazenado foi concluído com sucesso.

A tabela do banco de dados que contém as IDs e os nomes dos funcionários é exibida da seguinte maneira:

ID do Funcionário	Nome do Funcionário
101	Bill Takash
102	Louis Li
103	Sarah Ferguson

O procedimento armazenado recebe a ID 101 do funcionário como um parâmetro de entrada e retorna o nome Bill Takash. Dependendo de como o mapeamento chama esse procedimento armazenado, qualquer ID, ou todas elas, poderão ser passadas para o procedimento armazenado.

Como a sintaxe varia de um banco de dados para outro, as instruções SQL para criar esse procedimento armazenado pode variar. As ferramentas de cliente usadas para passar as instruções SQL para o banco de dados também variam. A maioria dos bancos de dados fornece um conjunto de ferramentas de cliente, inclusive um editor SQL padrão. Alguns bancos de dados, como o Microsoft SQL Server, fornecem ferramentas para criar algumas das instruções SQL iniciais.

Nota: O Serviço de Integração falha as sessões que contêm argumentos de procedimentos armazenados com objetos grandes.

Informix

Na Informix, a sintaxe para declarar um parâmetro de saída é diferente de outros bancos de dados. Com a maioria dos bancos de dados, declaram-se variáveis com o uso de IN ou OUT para especificar se a variável age como um parâmetro de entrada ou de saída. A Informix usa a palavra-chave RETURNING, o que torna difícil distinguir parâmetros de entrada/saída dos valores de retorno. Por exemplo, usa-se o comando RETURN para retornar um ou mais parâmetros de saída:

```
CREATE PROCEDURE GET_NAME_USING_ID (nID integer)
RETURNING varchar(20);
define nID integer;
define outVAR as varchar(20);
SELECT FIRST_NAME INTO outVAR FROM CONTACT WHERE ID = nID
return outVAR;
END PROCEDURE;
```

Observe que, nesse caso, a instrução RETURN transfere o valor de outVAR. Todavia, diferentemente de outros bancos de dados, outVAR não é um valor de retorno, mas um parâmetro de saída. Vários parâmetros de saída seriam retornados da seguinte forma:

```
return outVAR1, outVAR2, outVAR3
```

A Informix realmente transfere um valor de retorno. O valor de retorno não é definido pelo usuário, mas gerado como um valor de verificação de erros. Na transformação, o valor R deve ser verificado.

Oracle

No Oracle, qualquer procedimento armazenado que retorne um valor é denominado uma função armazenada. Em vez de usar a instrução CREATE PROCEDURE para criar um novo procedimento armazenado com base no exemplo, você usa a instrução CREATE FUNCTION. Nesta amostra, as variáveis são declaradas como IN e OUT, mas o Oracle só suporta um tipo de parâmetro INOUT que permite transferir em um parâmetro, modificá-lo e retornar o valor modificado:

```
CREATE OR REPLACE FUNCTION GET_NAME_USING_ID (
nID IN NUMBER,
outVAR OUT VARCHAR2)
RETURN VARCHAR2 IS
RETURN_VAR varchar2(100);
BEGIN
SELECT FIRST_NAME INTO outVAR FROM CONTACT WHERE ID = nID;
RETURN_VAR := 'Success';
RETURN (RETURN_VAR);
END;
/
```

Observe que o valor do retorno é um valor de string (Success) com o tipo de dados VARCHAR2. O Oracle é o único banco de dados que permite valores de retorno com tipos de dados de string.

Sybase ASE e Microsoft SQL Server

A Sybase e a Microsoft implementam procedimentos armazenados de maneira idêntica, como mostra a sintaxe a seguir:

```
CREATE PROCEDURE GET_NAME_USING_ID @nID int = 1, @outVar varchar(20) OUTPUT
AS
SELECT @outVar = FIRST_NAME FROM CONTACT WHERE ID = @nID
return 0
```

Observe que o valor de retorno não precisa ser uma variável. Nesse caso, se a instrução SELECT obtiver sucesso, 0 será retornado como o valor de retorno.

IBM DB2

O texto a seguir é um exemplo de um procedimento SQL armazenado no IBM DB2:

```
CREATE PROCEDURE get_name_using_id ( IN id_in int,
                                     OUT emp_out char(18),
                                     OUT sqlcode_out int)
    LANGUAGE SQL
P1: BEGIN
    -- Declare variables
    DECLARE SQLCODE INT DEFAULT 0;
    DECLARE emp_TMP char(18) DEFAULT ' ';
    -- Declare handler
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
        SET SQLCODE_OUT = SQLCODE;
    select employee into emp_TMP
        from doc_employee
        where id = id_in;
    SET emp_out = EMP_TMP;
    SET sqlcode_out = SQLCODE;
END P1
```

Teradata

O texto a seguir é um exemplo de um procedimento SQL armazenado no Teradata. Ele assume o número de ID de um funcionário como parâmetro de entrada e retorna o nome do funcionário como parâmetro de saída.

```
CREATE PROCEDURE GET_NAME_USING_ID (IN nID integer, OUT outVAR varchar(40))
BEGIN
    SELECT FIRST_NAME INTO :outVAR FROM CONTACT where ID = :nID;
END;
```

Criação de uma Transformação Procedimento Armazenado

Depois de configurar e testar um procedimento armazenado no banco de dados, você deve criar a transformação Procedimento Armazenado no Mapping Designer. Existem duas maneiras para configurar a transformação Procedimento Armazenado:

- Use a caixa de diálogo Importar Procedimento Armazenado para configurar as portas usadas pelo procedimento armazenado.
- Configure a transformação manualmente, criando as portas apropriadas para qualquer parâmetro de entrada ou saída.

As transformações Procedimento Armazenado são criadas como tipo Normal por padrão, o que significa que elas executam durante o mapeamento, não antes ou depois da sessão.

Novas transformações Procedimento Armazenado não são criadas como transformações reutilizáveis. Para criar uma transformação reutilizável, clique em Tornar Reutilizável nas propriedades da Transformação, depois de criá-la.

Nota: Configure as propriedades das transformações reutilizáveis no Transformation Developer, e não no Mapping Designer, para efetuar as alterações globalmente na transformação.

Importando procedimentos armazenados

Quando você importa um procedimento armazenado, o Designer cria portas com base nos parâmetros de entrada e saída do procedimento armazenado. Você deve importar o procedimento armazenado sempre que possível.

Existem três maneiras de importar um procedimento armazenado no Mapping Designer:

- Selecione o ícone do procedimento armazenado e adicione uma transformação Procedimento Armazenado.
- Clique em Transformação > Importar Procedimento Armazenado.
- Clique em Transformação > Criar e, em seguida, selecione Procedimento Armazenado.

Quando você importa um procedimento armazenado que contém um ponto-final (.) no nome do procedimento armazenado, o Designer substitui um sublinhado (_) por um ponto-final no nome da transformação Procedimento Armazenado.

Para importar um procedimento armazenado:

1. No Mapping Designer, Clique em Transformação > Importar Procedimento Armazenado.
2. Selecione o banco de dados que contém o procedimento armazenado, na lista de origens ODBC. Insira o nome do usuário, nome do proprietário e senha para conectar-se ao banco de dados e clique em Conectar.

A pasta na caixa de diálogo exibe FUNCTIONS. Os procedimentos armazenados nesta pasta contêm parâmetros de entrada, parâmetros de saída ou um valor de retorno. Se existirem procedimentos armazenados no banco de dados que não contêm parâmetros ou valores de retorno, eles aparecem em uma pasta chamada PROCEDURES. Isto se aplica principalmente aos procedimentos armazenados do Oracle. Para que um Procedimento Armazenado normal conectado apareça na lista de funções, é necessário pelo menos uma porta de entrada e uma de saída.

Sugestão: Você pode selecionar Ignorar para adicionar uma transformação Procedimento Armazenado sem importar o procedimento armazenado. Neste caso, você precisa adicionar as portas manualmente e conectar as informações dentro da transformação.

3. Como opção, use o campo de pesquisa para limitar o número de procedimentos exibidos.
4. Selecione o procedimento para importar e clique em OK.

A transformação Procedimento Armazenado aparece no mapeamento. O nome da transformação Procedimento Armazenado é igual ao procedimento armazenado que você selecionou. Se o procedimento armazenado contiver parâmetros de entrada, parâmetros de saída ou um valor de retorno, você verá as portas apropriadas que correspondem a cada parâmetro ou valor de retorno na transformação Procedimento Armazenado.

Nesta transformação Procedimento Armazenado, você pode ver que o procedimento armazenado contém os seguintes valores e parâmetros:

- Um valor de retorno inteiro, RETURN_VALUE, com uma porta de saída.
- Um parâmetro de entrada de string, nNAME, com uma porta de entrada.
- Um parâmetro de saída inteiro, outVar, com uma porta de entrada e saída.

Nota: Se você alterar o nome da transformação, precisará configurar o nome do procedimento armazenado nas propriedades da transformação. Se você tiver múltiplas instâncias do mesmo procedimento armazenado em um mapeamento, também deve configurar o nome do procedimento armazenado.

5. Abra a transformação e clique na guia Propriedades.

Selecione o banco de dados em que o procedimento armazenado existe, a partir da linha Informações da Conexão. Se você alterou o nome da transformação Procedimento Armazenado para algo diferente do nome do procedimento armazenado, insira o Nome do Procedimento Armazenado.

6. Clique em OK.

Criando manualmente transformações de Procedimento Armazenado

Para criar uma transformação Procedimento Armazenado manualmente, você precisa saber os parâmetros de entrada, os de saída e os valores de retorno do procedimento armazenado, se houver. Você também deve saber o tipo de dados desses parâmetros, e o nome do procedimento armazenado. Tudo isto é configurado em Importar Procedimento Armazenado.

Para criar uma transformação Procedimento Armazenado:

1. No Mapping Designer, clique em Transformação > Criar, e então selecione Procedimento Armazenado.

A convenção de nomenclatura para uma transformação Procedimento Armazenado é o nome do procedimento armazenado, que acontece automaticamente. Se você alterar o nome da transformação, então precisa configurar o nome do procedimento armazenado em Propriedades da Transformação. Se você tiver múltiplas instâncias do mesmo procedimento armazenado em um mapeamento, deve executar esta etapa.

2. Clique em Ignorar.

A transformação Procedimento Armazenado aparece no Mapping Designer.

3. Abra a transformação e clique na guia Portas.

Você deve criar portas com base nos parâmetros de entrada ou de saída e valores de retorno no procedimento armazenado. Crie uma porta na transformação Procedimento Armazenado para cada parâmetros de procedimento armazenado a seguir:

- Um parâmetro de entrada de número inteiro
- Um parâmetro de saída de string
- Um valor de retorno

Para o parâmetro de entrada de número inteiro, você deve criar uma porta de entrada de número inteiro. O parâmetro e a porta devem ter o mesmo tipo de dados e precisão. Repita isto para o parâmetro de saída e o valor de retorno.

A coluna R deve ser selecionada e a porta de saída para o valor de retorno. Para procedimentos armazenados com parâmetros múltiplos, você deve listar as portas na mesma ordem em que elas aparecem no procedimento armazenado.

4. Clique na guia Propriedades.

Insira o nome do procedimento armazenado na linha Nome do Procedimento Armazenado, e selecione o banco de dados em que o procedimento armazenado existe, na linha Informações da Conexão.

5. Clique em OK.

Embora o repositório valide e salve o mapeamento, o Designer não valida a transformação Procedimento Armazenado manualmente inserida. Nenhuma verificação é concluída para verificar se existem os parâmetros ou valores de retorno adequados no procedimento armazenado. Se a transformação Procedimento Armazenado não estiver adequadamente configurada, haverá falha na sessão.

Definindo Opções Para o Procedimento Armazenado

A tabela a seguir descreve as propriedades da transformação Procedimento Armazenado:

Configuração	Descrição
Nome do procedimento armazenado	Nome do procedimento armazenado no banco de dados. O serviço de Integração usará esse texto para chamar o procedimento armazenado se o nome da transformação for diferente do nome do procedimento armazenado real no banco de dados. Deixe esse campo em branco se o nome da transformação corresponder ao nome do procedimento armazenado. Ao usar o recurso Importar Procedimento Armazenado, esse nome corresponderá ao procedimento armazenado.
Informações de Conexão	<p>Especifica o banco de dados contendo o procedimento armazenado. Você pode definir o banco de dados no arquivo de parâmetro, mapeamento ou sessão:</p> <ul style="list-style-type: none">- Mapeamento. Selecione o objeto de conexão relacional.- Sessão Use as variáveis de conexão <code>\$\$Source</code> ou <code>\$\$Target</code>. Se você usar uma dessas variáveis, o procedimento armazenado deverá residir no banco de dados de origem ou de destino especificado quando você executar a sessão. Especifique a conexão do banco de dados para cada variável nas propriedades da sessão.- Arquivo de parâmetros. Use o parâmetro de sessão <code>\$\$DBConnectionName</code> e o defina no arquivo de parâmetros. <p>Por padrão, o Designer especifica <code>\$\$Target</code> para tipos de procedimento armazenado Normal. Para pré e pós-carregamento da origem, o Designer especifica <code>\$\$Source</code>. Para pré e pós-carregamento do destino, o Designer especifica <code>\$\$Target</code>. Você pode substituir esses valores nas propriedades da sessão.</p>
Chamada de texto	<p>Texto usado para chamar o procedimento armazenado. Usado apenas quando o Tipo de Procedimento Armazenado não for Normal. Você deve incluir todos os parâmetros de entrada passados ao procedimento armazenado na chamada de texto.</p> <p>Você também pode usar uma variável ou um parâmetro do PowerCenter na chamada de texto. Use qualquer tipo de parâmetro ou variável que você possa definir no arquivo de parâmetros.</p>
Tipo de Procedimento Armazenado	Determina quando o Serviço de Integração chamará o procedimento armazenado. As opções incluem Normal (durante o mapeamento) ou pré ou pós-carregamento no banco de dados de origem ou de destino. O padrão é Normal.
Nível de rastreamento	<p>Quantidade de detalhes de transações relatados no arquivo de log da sessão. Use os seguintes níveis de rastreamento:</p> <ul style="list-style-type: none">- Conciso- Normal- Inicialização Detalhada- Dados Detalhados <p>O padrão é Normal.</p>
Ordem de Execução	Ordem em que o Serviço de Integração chama o procedimento armazenado na transformação, relativa a todos os outros procedimentos armazenados no mesmo mapeamento. Usada apenas quando o Tipo de Procedimento Armazenado for diferente de Normal e houver mais de um procedimento armazenado.
Precisão de Subsegundos	<p>Especifica a precisão de subsegundo das portas de data e hora.</p> <p>Você pode alterar a precisão de bancos de dados que tenham uma escala editável para dados de data e hora. Você pode alterar a precisão de subsegundos para tipos de dados Oracle Timestamp, Informix Datetime e Teradata Timestamp.</p> <p>Selecione um valor inteiro positivo de 0 a 9. O padrão é 6 (microsegundos).</p>

Configuração	Descrição
Saída é repetível	<p>Indica se a transformação gerará linhas na mesma ordem entre execuções da sessão. O Serviço de Integração poderá retomar uma sessão do último ponto de verificação quando a saída for repetível e determinista. Use os seguintes valores:</p> <ul style="list-style-type: none"> - Sempre. A ordem dos dados de saída será consistente entre as execuções de sessão mesmo se a ordem de entrada for inconsistente entre elas. - Baseada na Ordem de Entrada. A transformação produzirá dados repetíveis entre execuções da sessão quando a ordem dos dados de saída de todos os grupos de entrada for consistente entre as execuções da sessão. Se os dados de entrada de qualquer grupo de entrada não estiverem ordenados, a saída não será ordenada. - Nunca. A ordem dos dados de saída é inconsistente entre as execuções de sessão. Não será possível configurar a retomada da recuperação do último ponto de verificação se uma transformação não produzir dados repetíveis. <p>O padrão é Baseada na Ordem de Entrada.</p>
Saída é determinística	<p>Indica se a transformação gera dados de saída consistentes entre as execuções de sessão. Você deve ativar essa propriedade para executar a recuperação em sessões que usam essa transformação.</p> <p>O padrão é desativado.</p>

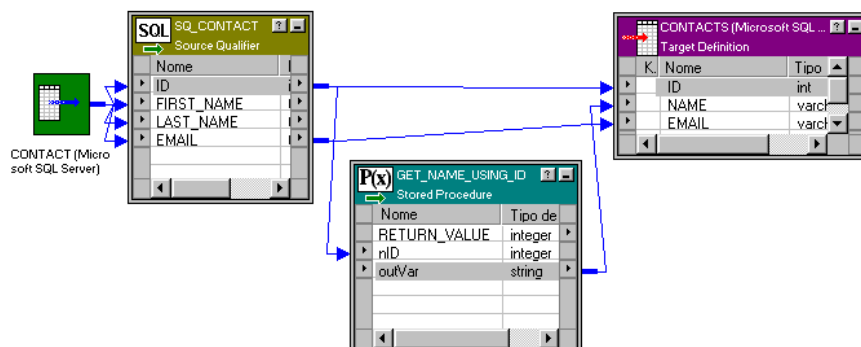
Aviso: Se você configurar uma transformação como repetível e determinística, será sua responsabilidade garantir que os dados sejam repetíveis e determinísticos. Se você tentar recuperar uma sessão com transformações que não gerem os mesmos dados entre a sessão e a recuperação, o processo de recuperação poderá resultar em dados corrompidos.

Alterando o Procedimento Armazenado

Se o número de parâmetros ou o valor de retorno em um procedimento armazenado mudar, é possível importá-lo novamente ou editar a transformação do Procedimento Armazenado manualmente. O Designer não verifica a Transformação Procedimento Armazenado a cada vez que se abre o mapeamento. Após importar ou criar a transformação, o Designer não valida o procedimento armazenado. Haverá falha na sessão se o procedimento armazenado não combinar com a transformação.

Configurando uma transformação conectada

A figura a seguir mostra um mapeamento que envia a ID do Qualificador de Origem para um parâmetro de entrada na transformação de Procedimento Armazenado:



A transformação de Procedimento Armazenado transfere um parâmetro de saída para o destino. Cada linha de dados na transformação de Qualificador de Origem transfere os dados através da transformação de Procedimento Armazenado.

Embora não seja exigido, quase todas as transformações Procedimento Armazenados conectadas contêm parâmetros de entrada e saída. Os parâmetros de entrada exigidos são especificados como as portas de entrada da transformação Procedimento Armazenado. Os parâmetros de saída aparecem como portas de saída na transformação. Um valor de retorno também é uma porta de saída, e tem o valor R selecionado na configuração Portas da transformação. Para que um Procedimento Armazenado normal conectado apareça na lista de funções, é necessário pelo menos uma porta de entrada e uma de saída.

Os parâmetros de saída e valores de retorno do procedimento armazenado são usados como qualquer outra porta de saída em uma transformação. Você pode vincular estas portas a outra transformação ou destino.

Para configurar uma transformação Procedimento Armazenado conectada:

1. Crie a transformação de Procedimento Armazenado no mapeamento.
2. Arraste as portas de saída do Procedimento Armazenado para outras transformações ou destinos.
3. Abra a transformação de Procedimento Armazenado e selecione a guia Propriedades.
4. Selecione o banco de dados apropriado nas Informações de Conexão se você não o tiver selecionado ao criar a transformação.
5. Selecione o nível de Rastreamento da transformação.
Se você estiver testando o mapeamento, selecione a opção Inicialização Detalhada para fornecer o máximo de informações caso a transformação falhe.
6. Clique em OK.

Configurando uma transformação desconectada

Uma Transformação Procedimento armazenado desconectada não é conectada diretamente ao fluxo de dados pelo mapeamento. Em vez disso, o procedimento armazenado executa uma destas opções:

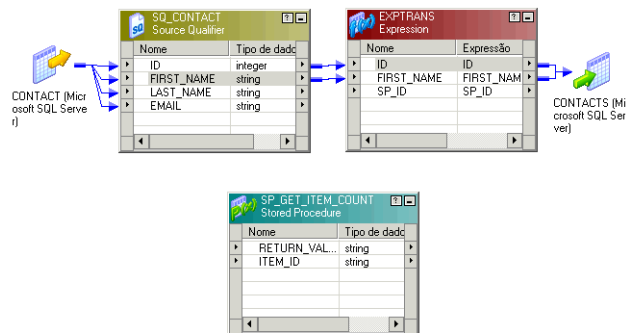
- **Em uma expressão.** Chamado a partir de uma expressão escrita no Editor de Expressão, dentro de outra transformação no mapeamento.
- **Pré ou pós-sessão.** Executa antes ou depois de uma sessão.

As seções abaixo explicam como você pode executar uma Transformação Procedimento armazenado desconectada.

Chamando um Procedimento Armazenado de Uma Expressão

Em um mapeamento não conectado, a transformação do Procedimento Armazenado não se conecta ao pipeline.

A figura a seguir mostra um mapeamento com uma Transformação Expressão que faz referência à Transformação Procedimento Armazenado:



Entretanto, como ocorre com um mapeamento conectado, é possível aplicar o procedimento armazenado ao fluxo de dados por meio do mapeamento. Na verdade, há maior flexibilidade, uma vez que se usa uma expressão para chamar o procedimento armazenado, que significa que é possível selecionar os dados que você transfere ao procedimento armazenado na qualidade de parâmetro de entrada.

Ao utilizar uma Transformação Procedimento Armazenado não conectada em uma expressão, é necessário um método de retornar o valor de parâmetros de saída a uma porta. Use um dos métodos a seguir para capturar os valores de saída:

- Atribua o valor de saída a uma variável local.
- Atribua o valor de saída à variável de sistema PROC_RESULT.

Usando a PROC_RESULT, você atribui o valor do parâmetro de retorno diretamente a uma porta de saída, que pode ser aplicada diretamente a um destino. Também é possível combinar as duas opções atribuindo um parâmetro de saída como PROC_RESULT, e o outro parâmetro como uma variável.

Use PROC_RESULT somente dentro de uma expressão. Caso não utilize o PROC_RESULT ou uma variável, a porta que contiver a expressão obtém um NULL. Não é possível utilizar PROC_RESULT em uma Transformação Pesquisa conectada ou dentro da Chamada de texto de uma Transformação Procedimento Armazenado.

Se forem necessários procedimentos armazenados aninhados, onde os parâmetros de saída de um procedimento armazenado são transferidos para outro procedimento armazenado, use PROC_RESULT para transferir o valor.

O Serviço de Integração chama a transformação não conectada do Procedimento Armazenado a partir da Transformação Expressão. Observe que a transformação do Procedimento Armazenado tem duas portas de entrada e uma porta de saída. As três portas são tipos de dados de string.

Para chamar um procedimento armazenado de dentro de uma expressão:

1. Crie a Transformação Procedimento Armazenado no mapeamento.
2. Em qualquer transformação que ofereça suporte a portas de saída e variáveis, crie uma nova porta de saída na transformação que chame o procedimento armazenado. Atribua um nome para a porta de saída.

A porta de saída que chama o procedimento armazenado deve oferecer suporte a expressões.

Dependendo de como a expressão é configurada, a porta de saída contém o valor do parâmetro de saída ou o valor de retorno.

3. Abra o Editor de Expressão da porta.

O valor da nova porta é definido no Editor de Expressão como uma chamada ao procedimento armazenado usando a palavra-chave :SP na Linguagem de Transformação. A maneira mais fácil de efetuar essa configuração adequadamente é selecionar o nó dos Procedimentos Armazenados no Editor de Expressão e clicar no nome da transformação do Procedimento Armazenado listada. Para que um Procedimento Armazenado normal conectado apareça na lista de funções, é necessário pelo menos uma porta de entrada e uma de saída.

O procedimento armazenado aparece no Editor de Expressão com dos parênteses vazios. Os parâmetros de entrada e/ou saída necessários são exibidos no canto inferior esquerdo do Editor de Expressão.

4. Configure a expressão para enviar parâmetros de entrada e obter parâmetros de saída ou valor de retorno.

É necessário saber se os parâmetros mostrados no Editor de Expressão são de entrada ou de saída. Insira variáveis ou nomes de porta entre os parênteses na ordem em que aparecem no procedimento armazenado. Os tipos de dados das portas e variáveis devem combinar com os dos parâmetros transferidos ao procedimento armazenado.

Por exemplo, ao clicar no procedimento armazenado, aparecerá algo semelhante ao que segue:

```
:SP.GET_NAME_FROM_ID()
```

Esse procedimento armazenado, em especial, exige um valor inteiro na qualidade de parâmetro de entrada e retorna um valor de string na qualidade de parâmetro de saída. A maneira com que o parâmetro de saída ou valor de retorno é obtido depende do número de parâmetros de entrada e do fato de o valor de retorno precisar ser obtido.

Se o procedimento armazenado retornar um único parâmetro de saída ou um valor de retorno (mas não os dois), a variável reservada PROC_RESULT deverá ser usada como variável de saída. No exemplo anterior, a expressão aparecerá como:

```
:SP.GET_NAME_FROM_ID(inID, PROC_RESULT)
```

inID pode ser ou uma porta de entrada para a transformação ou uma variável na transformação. O valor de PROC_RESULT é aplicado à porta de saída para a expressão.

Se o procedimento armazenado retornar diversos parâmetros de saída, será necessário criar variáveis para cada parâmetro de saída. Por exemplo, se você criar uma porta chamada varOUTPUT2 para a expressão do procedimento armazenado e uma variável chamada varOUTPUT1, a expressão aparecerá como:

```
:SP.GET_NAME_FROM_ID(inID, varOUTPUT1, PROC_RESULT)
```

O valor da segunda porta de saída é aplicado à porta de saída para a expressão, e o valor da primeira porta de saída é aplicado a varOUTPUT1. Os parâmetros de saída são retornados na ordem em que são declarados no procedimento armazenado.

Com todas essas expressões, os tipos de dados das portas e variáveis devem combinar com os tipos de dados das variáveis de entrada/saída e valores de retorno.

5. Clique em Validar para verificar a expressão, depois clique em OK para fechar o Editor de Expressão.

A validação da expressão garante que os tipos de dados para os parâmetros do procedimento armazenado combinem com os que são inseridos na expressão.

6. Clique em OK.

Ao salvar o mapeamento, o Designer não valida a expressão do procedimento validado. Se a expressão do procedimento armazenado não estiver adequadamente configurada, haverá falha na sessão. Ao testar um mapeamento com um procedimento armazenado, defina a opção da sessão Substituir Rastreamento para um modo detalhado e configure a opção da sessão No Procedimento Armazenado para que pare de executar em caso de falha do procedimento armazenado. Configure essas opções de sessão nas configurações de Tratamento de Erros da guia Objeto de Configuração nas propriedades da sessão.

O procedimento armazenado na expressão inserida em uma porta não tem que afetar todos os valores que passam pela porta. Com a instrução IIF, por exemplo, é possível transferir apenas determinados valores, como números de ID que comecem com 5, ao procedimento armazenado e ignorar todos os outros valores. Também é possível configurar procedimentos armazenados aninhados para que o valor de retorno de um procedimento armazenado se torne um parâmetro de entrada de um segundo procedimento armazenado.

Chamada de um Procedimento Armazenado Pré ou Pós-sessão

Você pode querer executar um procedimento armazenado uma vez por sessão. Por exemplo, se você precisar verificar se as tabelas existem em um banco de dados de destino antes de executar um mapeamento, um procedimento armazenado do destino pré-carregamento pode verificar as tabelas, e depois continuar executando o fluxo de trabalho ou interrompê-lo. Você pode executar um procedimento armazenado no banco de dados de origem, destino ou qualquer outro conectado.

Para criar um procedimento armazenado pré ou pós-carregamento:

1. Crie a Transformação Procedimento Armazenado no mapeamento.
2. Clique duas vezes na transformação Procedimento Armazenado e selecione a guia Propriedades.
3. Insira o nome do procedimento armazenado.

Se você importou o procedimento armazenado, o nome do mesmo aparece por padrão. Se você configurou o procedimento armazenado manualmente, insira o nome do mesmo.

4. Selecione o banco de dados que contém o procedimento armazenado em Informações de Conexão.
5. Insira o texto de chamada do procedimento armazenado.

O texto de chamada é o nome do procedimento armazenado, seguido por todos os parâmetros de entrada aplicáveis entre parênteses. Se não houver parâmetros de entrada, você deve incluir um par vazio de parênteses, ou a chamada para as falhas de procedimento armazenado. Você não precisa incluir a instrução EXEC SQL, e nem usar a palavra-chave :SP. Por exemplo, para chamar um procedimento armazenado com o nome `check_disk_space`, insira o texto a seguir:

```
check_disk_space()
```

Para transferir um parâmetro de entrada de string, insira sem aspas. Se a string tiver espaços, inclua o parâmetro entre aspas duplas. Por exemplo, se o procedimento armazenado `check_disk_space` exigia um nome de máquina como parâmetro de entrada, insira o texto a seguir:

```
check_disk_space(oracle_db)
```

Ao transferir um valor datetime através de um procedimento armazenado pré ou pós-sessão, o valor deve estar no formato de data padrão da Informatica e entre aspas duplas, conforme segue:

```
SP("12/31/2000 11:45:59")
```

Você pode usar os parâmetros e variáveis do PowerCenter no texto de chamada. Use qualquer tipo de parâmetro ou variável que você possa definir no arquivo de parâmetros. Você pode inserir um parâmetro ou variável dentro do texto de chamada, ou usar um parâmetro ou variável como o texto de chamada. Por exemplo, você pode usar um parâmetro de sessão, `$ParamMyCallText`, como texto de chamada, e configurar o `$ParamMyCallText` para o texto de chamada em um arquivo de parâmetro.

Nota: Digite valores em vez de parâmetros de procedimento para os parâmetros de entrada de procedimentos de pré-sessão e pós-sessão.

6. Selecione o tipo de procedimento armazenado.

As opções de tipo de procedimento armazenado incluem:

- **Pré-carregamento da origem.** Antes de a sessão recuperar dados na origem, o procedimento armazenado é executado. É útil para verificar a existência de tabelas ou fazer associações de dados em uma tabela temporária.

- **Pós-carregamento da origem.** Depois que a sessão recupera os dados da origem, o procedimento armazenado é executado. É útil para remover tabelas temporárias.
 - **Pré-carregamento do destino.** Antes de a sessão enviar dados ao destino, o procedimento armazenado é executado. É útil para verificar tabelas de destino ou o espaço em disco no sistema de destino.
 - **Pós-carregamento do destino.** Após a sessão enviar dados ao destino, o procedimento armazenado é executado. É útil para recriar índices do banco de dados.
7. Selecione Ordem de Execução, e clique na seta Para cima ou Para baixo a fim de alterar a ordem, se necessário.

Se você adicionou vários procedimentos armazenados que executam no mesmo ponto de uma sessão, como dois procedimentos que executam no Pós-carregamento da Origem, pode configurar um plano de execução de procedimento armazenado para determinar a ordem em que o Serviço de Integração chama esses procedimentos armazenados. Você precisa repetir esta etapa para cada procedimento armazenado que deseja alterar.

8. Clique em OK.

Embora o repositório valide e salve o mapeamento, o Designer não valida se a expressão de procedimento armazenado executar sem um erro. Se a expressão do procedimento armazenado não estiver adequadamente configurada, haverá falha na sessão. Ao testar um mapeamento com um procedimento armazenado, defina a opção da sessão Substituir Rastreamento para um modo detalhado e configure a opção da sessão No Procedimento Armazenado para que pare de executar em caso de falha do procedimento armazenado. Configure essas opções de sessão nas configurações de Manipulação de Erros da guia Objeto de Configuração nas propriedades da sessão.

Você perde parâmetros de saída ou os valores de retorno chamados durante os procedimentos armazenados pré ou pós-sessão, uma vez que não existe local para capturar os valores. Se você precisa capturar os valores, pode configurar o procedimento armazenado para salvar o valor em uma tabela no banco de dados.

Tratamento de erros

Às vezes um procedimento armazenado retorna um erro de banco de dados, como "dividir por zero" ou "mais nenhuma linha". O resultado final de um erro de banco de dados, durante um procedimento armazenado, depende de quando o procedimento armazenado ocorre e como a sessão é configurada.

Você pode configurar a sessão para interromper ou continuar executando a sessão, ao encontrar um erro de procedimento armazenado pré- ou pós-sessão. Por padrão, o Serviço de Integração interrompe uma sessão quando ocorre um erro de banco de dados do procedimento armazenado pré- ou pós-sessão.

Erros Pré-sessão

Procedimentos armazenados pré-leitura e pré-carregamento são considerados procedimentos armazenados pré-sessão. Ambos são executados antes de o Serviço de Integração começar a ler dados de origem. Se ocorrer um erro de banco de dados durante um procedimento armazenado pré-sessão, o Serviço de Integração realizará outra ação, dependendo da configuração da sessão.

- Se você configurar a sessão para parar em erros de procedimento armazenado, o Serviço de Integração encerrará a sessão.
- Se você configurar a sessão para continuar em erros de procedimento armazenado, o Serviço de Integração continuará com a sessão.

Erros Pós-sessão

Procedimentos armazenados pós-leitura e pós-carregamento são considerados procedimentos armazenados. Ambos são executados depois que o Serviço de Integração confirma todos os dados ao banco de dados. Se houver erros de banco de dados durante o procedimento armazenado pós-sessão, o Serviço de Integração realizará outra ação, dependendo da configuração da sessão:

- Se você configurar a sessão para parar em erros de procedimento armazenado, o Serviço de Integração encerrará a sessão.

Contudo, o Serviço de Integração já terá confirmado todos os dados aos destinos da sessão.

- Se você configurar a sessão para continuar em erros de procedimento armazenado, o Serviço de Integração continuará com a sessão.

Erros de Sessão

A opção de tratamento de erros de sessão não interfere nos erros de procedimento armazenado conectado ou não conectado que ocorrem durante a sessão. Se o banco de dados retornar um erro para uma linha específica, o Serviço de Integração ignorará essa linha e passará para a seguinte. Como acontece com outros erros de transformação de linha, a linha ignorada aparece no log da sessão.

Bancos de Dados Suportados

As opções suportadas para bancos de dados Oracle e outros, como Informix, Microsoft SQL Server e Sybase, são descritas abaixo.

TÓPICOS RELACIONADOS:

- [“Gravando um Procedimento Armazenado” na página 476](#)

Declaração SQL

No banco de dados, a instrução que cria procedimentos armazenados parece semelhante a este procedimento armazenado do Oracle:

```
create or replace procedure sp_combine_str
(str1_inout IN OUT varchar2,
str2_inout IN OUT varchar2,
str_out OUT varchar2)
is
begin
str1_inout := UPPER(str1_inout);
str2_inout := upper(str2_inout);
str_out := str1_inout || ' ' || str2_inout;
end;
```

Nesse caso, a instrução do Oracle começa com CREATE OR REPLACE PROCEDURE. Como suporta procedimentos armazenados e funções armazenadas, somente o Oracle usa a instrução opcional CREATE FUNCTION.

Tipos de Parâmetro

Existem três tipos possíveis de parâmetros nos procedimentos armazenados:

- **IN.** Define o parâmetro como algo que deve ser transferido para o procedimento armazenado.

- **OUT.** Define o parâmetro como um valor retornado do procedimento armazenado.
- **INOUT.** Define o parâmetro como entrada e saída. Somente o Oracle suporta este tipo de parâmetro.

Porta de entrada/saída no mapeamento

Uma vez que a Oracle oferece suporte ao tipo de parâmetro INOUT, uma porta em uma transformação de Procedimento Armazenado pode agir como porta de entrada e de saída para o mesmo parâmetro de procedimento armazenado. Outros bancos de dados não devem ter as caixas de seleção de entrada e de saída selecionadas em uma porta.

Tipo de Valor de Retorno Suportado

Diferentes bancos de dados suportam diferentes tipos de dados de valor de retorno. E somente o Informix não oferece suporte a valores de retorno definidos pelo usuário.

Regras de Expressão

As transformações de Procedimento Armazenado não conectadas podem ser chamadas a partir de uma expressão em outra transformação. Use as seguintes regras e diretrizes ao configurar a expressão:

- Um único parâmetro de saída é retornado usando a variável PROC_RESULT.
- Ao usar um procedimento armazenado em uma expressão, use o qualificador de referência :SP. Para evitar erros de digitação, selecione o nó de Procedimento Armazenado no Editor de Expressão e clique duas vezes no nome do procedimento armazenado.
- No entanto, você não pode executar a mesma instância de uma transformação Procedimento Armazenado no modo conectado e não conectado em um mapeamento. Para isso, você precisa criar instâncias diferentes da transformação.
- Os parâmetros de entrada/saída na expressão devem combinar com as portas de entrada/saída na transformação de Procedimento Armazenado. Se o procedimento armazenado apresentar um parâmetro de entrada, deve haver também uma porta de entrada na transformação do Procedimento Armazenado.
- Ao registrar uma expressão que tenha um procedimento armazenado, liste os parâmetros na mesma ordem em que aparecem no procedimento armazenado e na transformação do Procedimento Armazenado.
- Os parâmetros da expressão devem apresentar todos os parâmetros da transformação de Procedimento Armazenado. Não se pode omitir um parâmetro de entrada. Caso seja necessário, transfira uma variável fictícia ao procedimento armazenado.
- Os argumentos da expressão devem ser dos mesmos tipos de dados e da mesma precisão que os que estão na transformação do Procedimento Armazenado.
- Use PROC_RESULT para aplicar o parâmetro de saída de uma expressão de procedimento armazenado diretamente no destino. Não é possível usar uma variável para que o parâmetro de saída transfira os resultados diretamente a um destino. Use uma variável local para transferir os resultados a uma porta de saída com a mesma transformação.

- Os procedimentos armazenados aninhados permitem a transferência do valor de retorno de um procedimento armazenado na qualidade de parâmetro de valor de entrada de outro procedimento armazenado. Por exemplo, se você tem os dois procedimentos armazenados a seguir:
 - `get_employee_id (employee_name)`
 - `get_employee_salary (employee_id)`
 E se o valor de retorno para `get_employee_id` for um número de ID de funcionário, a sintaxe de um procedimento armazenado aninhado é:


```
:sp.get_employee_salary (:sp.get_employee_id (employee_name))
```

 É possível ter vários níveis de procedimentos armazenados aninhados.
- Não use aspas simples para delimitar parâmetros de string. Se o parâmetro de entrada não contiver espaços, não use aspas. Se o parâmetro de entrada contiver espaços, use aspas duplas.

Dicas para Transformações Procedimento Armazenado

Não execute instâncias desnecessárias de procedimentos armazenados.

Cada vez que um procedimento armazenado é executado durante um mapeamento, a sessão deve aguardar que o procedimento armazenado seja concluído no banco de dados. Você tem duas opções para evitar isso:

- **Reduzir a contagem de linhas.** Use uma transformação ativa antes que a transformação Procedimento Armazenado reduza o número de linhas que devem ser passadas ao procedimento armazenado. Ou crie uma expressão que teste os valores antes de passá-los ao procedimento armazenado, para se certificar de que o valor deva realmente ser passado.
- **Criar uma expressão.** Muito da lógica usada em procedimentos armazenados pode ser reproduzida facilmente usando expressões no Designer.

Solução de Problemas de Transformações Procedimento Armazenado

Eu vejo o erro "procedimento armazenado não encontrado" no arquivo de log da sessão.

Certifique-se de que o procedimento armazenado foi executado no banco de dados certo. Por padrão, a transformação Procedimento Armazenado usa o banco de dados de destino para executar o procedimento armazenado. Clique duas vezes na transformação no mapeamento, selecione a guia Propriedades e verifique o banco de dados selecionado em Informações de Conexão.

Meu parâmetro de saída não foi retornado usando um procedimento armazenado do Microsoft SQL Server.

Verifique se o parâmetro para reter o valor retornado foi declarado como OUTPUT no procedimento armazenado. Com o Microsoft SQL Server, OUTPUT implica em entrada e saída. No mapeamento, talvez você tenha marcado as caixas I e O para a porta. Limpe a porta de entrada.

A sessão não tinha erros antes, mas agora ela é encerrada no procedimento armazenado.

O motivo mais comum para problemas com uma transformação Procedimento Armazenado resulta de alterações feitas ao procedimento armazenado no banco de dados. Se os valores dos parâmetros de entrada e saída forem alterados em um procedimento armazenado, a transformação Procedimento Armazenado se tornará inválida. Você deve importar o procedimento armazenado novamente ou configurá-lo manualmente para adicionar, remover ou modificar as portas adequadas.

A sessão foi invalidada porque eu editei o mapeamento. Por quê?

Qualquer alteração feita à transformação Procedimento Armazenado pode invalidar a sessão. O motivo mais comum é a alteração do tipo de procedimento armazenado, de Normal para Origem Pós-Carregamento, por exemplo.

CAPÍTULO 29

Transformação Controle de Transações

Este capítulo inclui os seguintes tópicos:

- [Visão Geral da Transformação Controle de Transação, 492](#)
- [Propriedades da Transformação Controle de Transação, 493](#)
- [Usando a Transformação Controle de Transação em Mapeamentos, 495](#)
- [Mapeando Orientações e Validação, 497](#)
- [Criando uma Transformação Controle de Transação, 497](#)

Visão Geral da Transformação Controle de Transação

O PowerCenter permite que você controle transações de confirmação e reversão com base em um conjunto de linhas que passam uma transformação Controle de Transação. Uma transformação de Controle de transação é uma transformação ativa. Uma transação é o conjunto de linhas associadas por linhas de confirmação ou reversão. Você pode definir uma transação com base em um número variável de linhas de entrada. Talvez você queira definir transações com base em um grupo de linhas ordenadas em uma chave comum, como ID do funcionário ou data de entrada do pedido.

No PowerCenter, você define o controle de transação nos seguintes níveis:

- **Dentro de um mapeamento.** Em um mapeamento, você utiliza a transformação Controle de Transação para definir uma transação. Você define transações usando uma expressão em uma transformação Controle de Transação. Com base no valor de retorno da expressão, você pode optar por confirmar, reverter ou continuar sem quaisquer mudanças na transação.
- **Dentro de uma sessão.** Ao configurar uma sessão, você configura-a para uma confirmação definida pelo usuário. Você pode optar por confirmar ou reverter uma transação se o Serviço de Integração não conseguir transformar ou gravar uma linha no destino.

Quando você executa a sessão, o Serviço de Integração avalia a expressão para cada linha que insere a transformação. Quando avalia uma linha de confirmação, ele confirma todas as linhas na transação para o(s) destino(s). Quando o Serviço de Integração avalia uma linha de reversão, ele reverte todas as linhas na transação a partir do(s) destino(s).

Se o mapeamento tiver um destino de arquivo simples, você poderá gerar um arquivo de saída toda vez que o Serviço de Integração iniciar uma nova transação. Você pode nomear dinamicamente cada arquivo simples de destino.

Nota: Você também pode usar o escopo de transformação nas outras propriedades de transformação para definir transações.

Propriedades da Transformação Controle de Transação

Use a transformação controle de transação para definir condições para transações de confirmação e reversão de destinos transacionais. Destinos transacionais incluem destinos relacionais, XML e MQSeries dinâmico. Defina esses parâmetros em uma expressão de controle de transação na guia Propriedades. Uma transação é a linha ou o conjunto de linhas associadas por linhas de confirmação ou reversão. O número de linhas pode variar para cada transação.

Ao configurar uma transformação Controle de Transação, você define os seguintes componentes:

- **Guia Transformação.** Você pode renomear a transformação e adicionar uma descrição na guia Transformação.
- **Guia Portas.** Você pode adicionar portas de entrada e saída a uma transformação Controle de Transação.
- **Guia Propriedades.** Você pode definir a expressão de controle de transação, que sinaliza transações para confirmação, reversão ou nenhuma ação.
- **Guia Extensões de Metadados.** Você pode associar informações à transformação controle de transação para estender os metadados armazenados no repositório.

Guia Propriedades

Na guia Propriedades, você pode configurar as seguintes propriedades:

- Expressão de controle de transação
- Nível de rastreamento

Insira a expressão de controle de transação no campo Condição de controle de transação. A expressão de controle de transação utiliza a função IIF para testar cada linha em relação à condição. Utilize a sintaxe a seguir para a expressão:

```
IIF (condition, value1, value2)
```

A expressão contém valores que representam ações realizadas pelo Serviço de Integração com base no valor de retorno da condição. O Serviço de Integração avalia a condição em uma base linha a linha. O valor de retorno determina se o Serviço de Integração confirma, reverte ou não faz alterações de transação na linha. Quando o Serviço de Integração emite uma confirmação ou uma reversão com base no valor de retorno da expressão, ele inicia uma nova transação. Use as seguintes variáveis internas no Editor de Expressão ao criar uma expressão de controle de transação:

- **TC_CONTINUE_TRANSACTION.** O Serviço de Integração não faz qualquer alteração de transação para esta linha. Esse é o valor padrão para a expressão.
- **TC_COMMIT_BEFORE.** O Serviço de Integração confirma a transação, começa a nova transação e grava a linha atual no destino. A linha atual está na nova transação.

- **TC_COMMIT_AFTER.** O Serviço de Integração grava a linha atual no destino, confirma a transação e começa uma nova transação. A linha atual está na transação comprometida.
- **TC_ROLLBACK_BEFORE.** O Serviço de Integração reverte a transação atual, começa uma nova transação e grava a linha atual no destino. A linha atual está na nova transação.
- **TC_ROLLBACK_AFTER.** O Serviço de Integração grava a linha atual no destino, reverte a transação e inicia uma nova transação. A linha atual está na transação revertida.

A expressão de controle de transação resulta em um valor diferente de confirmar, reverter ou continuar. O Serviço de Integração interrompe a sessão.

Exemplo

Você deseja usar o controle de transação para gravar informações de ordem com base na data de entrada da ordem. Você quer garantir que todas as ordens inseridas em qualquer data estão comprometidas ao destino na mesma transação. Para fazer isto, é possível criar um mapeamento com as seguintes transformações:

- **Transformação Classificador** Classifique os dados da origem por data de entrada da ordem.
- **Transformação Expressão.** Use as variáveis locais para determinar se a data inserida é uma nova data.

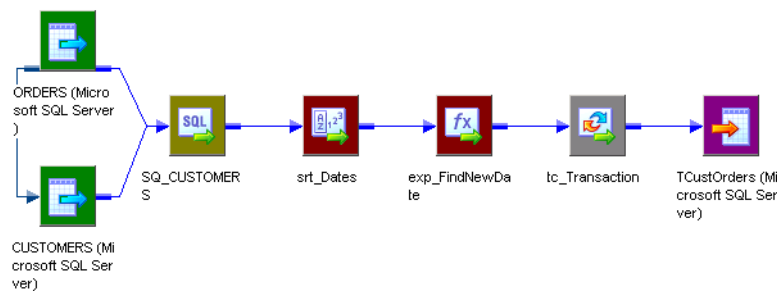
A tabela a seguir descreve as portas na transformação de Expressão:

Nome da porta	Expressão	Descrição
DATE_ENTERED	DATE_ENTERED	Porta de entrada/saída. Recebe e transmite a data inserida.
NEW_DATE	IIF(DATE_ENTERED=PREVDATE, 0,1)	Porta variável. Testa o valor atual de DATE_ENTERED em relação ao valor armazenado para DATE_ENTERED na porta variável, PREV_DATE.
PREV_DATE	DATE_ENTERED	Porta variável. Recebe o valor de DATE_ENTERED após o Serviço de Integração avaliar a porta NEW_DATE.
DATE_OUT	NEW_DATE	Porta de saída. Transmite o sinalizador de NEW_DATE à Transformação Controle de Transação.
Nota: O Serviço de Integração avalia as portas por dependência. A ordem na qual as portas são exibidas em uma transformação deve combinar com a ordem de avaliação: portas de entrada, portas variáveis, portas de saída.		

- **Transformação Controle de Transação.** Crie a seguinte expressão de controle de transação para confirmar os dados quando o Serviço de Integração encontrar uma nova data de entrada de ordem:

```
IIF(NEW_DATE = 1, TC_COMMIT_BEFORE, TC_CONTINUE_TRANSACTION)
```

A figura a seguir mostra um exemplo de mapeamento com o uso de uma Transformação Controle de Transação:



Usando a Transformação Controle de Transação em Mapeamentos

As transformações Controle de Transação são geradores de transações. Elas definem e redefinem limites de transação em um mapeamento. Elas descartam qualquer limite de transação de entrada de um gerador de transação ou de uma origem ativa upstream e geram novos limites de transação downstream.

Você também pode usar transformações personalizadas configuradas para gerar transações, a fim de definir limites de transação.

As transformações Controle de Transação podem ser eficazes ou ineficazes para destinos e transformações downstream no mapeamento. A transformação Controle de Transação se tornará ineficaz para destinos e transformações downstream se você colocar depois dela uma transformação que descarte limites de transação de entrada. Isso inclui qualquer uma das transformações ou origens ativas a seguir:

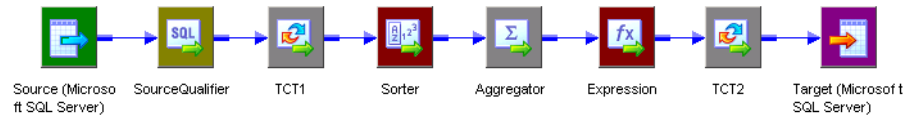
- A transformação Agregador com o escopo de transformação de nível Todas as Entradas.
- Transformação Joiner com o escopo de transformação de Todas as Entradas
- Transformação Classificação com o escopo de transformação de Todas as Entradas
- Transformação de Classificador com o escopo de transformação de Todas as Entradas
- Transformação personalizada com o escopo de transformação de Todas as Entradas
- Uma transformação personalizada configurada para gerar transações
- Transformação Controle de Transação
- Uma transformação com vários grupos de entrada, como uma transformação personalizada conectada a vários pontos de controle de transação upstream

Mapeamentos com transformações Controle de Transação que são ineficazes para destinos podem ser válidos ou inválidos. Quando você salva ou valida o mapeamento, o Designer exibe uma mensagem indicando quais transformações Controle de Transação são ineficazes para os destinos.

Embora uma transformação Controle de Transação possa ser ineficaz para um destino, ela pode ser eficaz para transformações downstream. As transformações downstream com o escopo de transformação de nível Transação podem usar os limites de transação definidos por uma transformação Controle de Transação upstream.

A figura a seguir mostra um mapeamento válido com uma transformação Controle de Transação que é eficaz para uma transformação Classificador, mas ineficaz para o destino.

Neste exemplo, a transformação TCT1 é ineficaz para o destino, mas eficaz para a transformação Classificador. A propriedade Escopo da Transformação da transformação Classificador é Transação. Ela usa os limites da transação definidos por TCT1. A propriedade do Escopo da Transformação Agregador é Todas as Entradas. Ela descarta os limites de transação definidos por TCT1. A transformação TCT2 é uma transformação Controle de Transação efetiva para o destino.

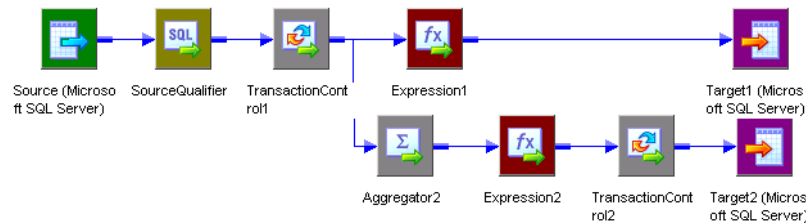


Mapeamentos Controle de Transação de Exemplo com Vários Destinos

Uma transformação Controle de Transação pode ser efetiva para um destino e não para outro.

Se cada destino estiver conectado a uma transformação Controle de Transação efetiva, o mapeamento será válido. Se cada destino estiver conectado a uma transformação Controle de Transação efetiva, o mapeamento será válido.

A figura a seguir mostra um mapeamento válido com uma transformação Controle de Transação efetiva e outra não efetiva:

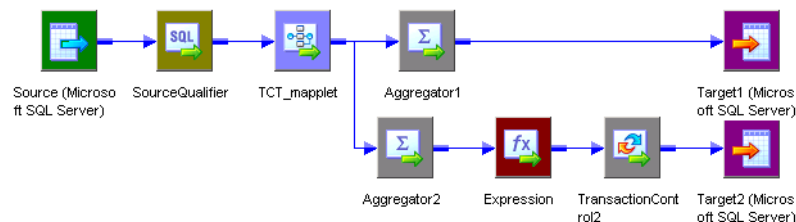


O Serviço de Integração processa TransactionControl1, avalia a expressão de controle de transação e cria limites de transação. O mapeamento não inclui transformações que descartem limites de transação entre TransactionControl1 e Target1, tornando TransactionControl1 efetiva para Target1. O Serviço de Integração usa os limites de transação definidos por TransactionControl1 para Target1.

Entretanto, o mapeamento inclui uma transformação que descarta limites de transação entre TransactionControl1 e Target2, tornando TransactionControl1 sem efeito para Target2. Quando o Serviço de Integração processa Agregator2, com o Escopo da Transação definido para Todos de entrada, descarta os limites de transação definidos por TransactionControl1 e lança todas as linhas em uma transação aberta. Quando o Serviço de Integração avalia TransactionControl2, cria limites de transação e os usa para Target2.

Se ocorrer uma reversão em TransactionControl1, o Serviço de Integração reverterá apenas linhas para Target1. Ele não reverterá nenhuma linha para Target2.

A figura a seguir mostra um mapeamento inválido com uma transformação Controle de Transação sem efeito e efetiva:



O mapplet TCT_mapplet contém uma transformação Controle de Transação. Não tem efeito para Target1 e Target2. A propriedade Escopo de Transformação da transformação de Agregador1 é Todas as Entradas. É uma origem ativa para Target1. A propriedade Escopo de Transformação da transformação de Agregador2 é Todas as Entradas. É uma fonte ativa para Target2. A transformação TransactionControl2 é efetiva para Target2.

O mapeamento é inválido porque Target1 não está conectado a uma transformação Controle de Transação efetiva.

Mapeando Orientações e Validação

Use as seguintes regras e diretrizes ao criar um mapeamento com uma transação de Controle de Transação:

- Se o mapeamento também tiver um destino XML, e você escolher acrescentar ou criar um novo documento na submissão, os grupos de entrada devem receber dados do mesmo ponto de controle de transação.
- As Transformações Controle de Transação que estejam conectadas a qualquer destino que não seja relacional, XML ou MQSeries dinâmicos não têm efeito para esses destinos.
- É necessário conectar cada instância de destino a uma Transformação Controle de Transação.
- É possível conectar vários destinos a uma única Transformação Controle de Transação.
- É possível conectar apenas uma Transformação Controle de Transação efetiva a um destino.
- Não é possível colocar uma Transformação Controle de Transação em uma ramificação de pipeline que inicie com uma transformação de Gerador de Sequência.
- Se você utilizar uma transformação de Pesquisa Dinâmica e uma transformação Controle de Transação no mesmo mapeamento, pode ocorrer uma transação invertida em dados de destino não sincronizados.
- Uma transformação Controle de Transação pode ser efetiva para um destino e não para outro. Se cada destino estiver conectado a uma transformação Controle de Transação efetiva, o mapeamento será válido.
- Deve-se conectar ou todos ou nenhum destino do mapeamento a uma Transformação Controle de Transação efetiva.

Criando uma Transformação Controle de Transação

Use o seguinte procedimento para adicionar uma transformação de Controle de Transação a um mapeamento:

1. No Mapping Designer, clique em Transformação > Criar. Selecione a transformação Controle de Transação.
2. Insira um nome para a transformação.
A convenção de nomenclatura das transformações Controle de Transação é *TC_TransformationName*.
3. Insira uma descrição para a transformação.
Essa descrição aparece ao exibir os detalhes da transformação no Repository Manager, tornando mais fácil entender o que a transformação faz.
4. Clique em Criar.

O Designer cria a Transformação Controle de Transação.

5. Clique em Concluído.

6. Arraste as portas para a transformação.

O Designer cria as portas de entrada/saída para cada porta que você incluir.

7. Abra a caixa de diálogo Editar Transformações e selecione a guia Portas.

Você pode adicionar portas, editar nomes de porta, adicionar descrições de porta e inserir valores padrão.

8. Selecione a guia Propriedades. Insira a expressão de controle de transação que define o comportamento da confirmação e da reversão.

9. Selecione a guia Extensões de metadados. Crie ou edite as extensões de metadados para a Transformação Controle de Transação.

10. Clique em OK.

CAPÍTULO 30

Transformação União

Este capítulo inclui os seguintes tópicos:

- [Visão Geral da Transformação de União, 499](#)
- [Trabalhando com Grupos e Portas, 500](#)
- [Criação de uma Transformação União, 500](#)
- [Usando uma Transformação de União em um Mapeamento, 501](#)

Visão Geral da Transformação de União

A Transformação de União é uma transformação de vários grupos de entrada que você usa para mesclar dados de várias pipelines ou de ramificações de pipeline em uma única ramificação de pipeline. Ela mescla dados de várias origens, de modo semelhante à instrução UNION ALL SQL, para combinar os resultados de duas ou mais instruções SQL. Da mesma forma que a instrução UNION ALL, a transformação de União não remove linhas duplicadas. A transformação de União é uma transformação ativa.

O Data Integration Service processa todos os grupos de entrada em paralelo. Ele lê simultaneamente origens conectadas à transformação de União e envia por push blocos de dados para os grupos de entrada da transformação. A transformação de União processa os blocos de dados com base na ordem em que os recebe do Serviço de Integração.

Você pode conectar origens heterogêneas a uma transformação de União. A transformação mescla origens com portas correspondentes e envia os dados de um grupo de saída com as mesmas portas que os grupos de entrada.

A transformação de União é desenvolvida usando a transformação personalizada.

Regras e Diretrizes para Transformações de União

Use as seguintes regras e diretrizes ao trabalhar com uma transformação de União:

- Você pode criar vários grupos de entrada, mas somente um grupo de saída.
- Todos os grupos de entrada e o grupo de saída devem ter portas correspondentes. A precisão, o tipo de dados e a escala devem ser idênticos em todos os grupos.
- A transformação de União não remove linhas duplicadas. Para remover linhas duplicadas, adicione outra transformação, como uma transformação Roteador ou Filtro.
- A transformação de União não gera transações.

Componentes da Transformação de União

Quando configura uma transformação de União, você define os seguintes componentes:

- **Guia Transformação.** Você pode renomear a transformação e adicionar uma descrição.
- **Guia Propriedades.** Você pode especificar o nível de rastreamento.
- **guia Grupos.** Você pode criar e excluir grupos de entrada. O Designer exibe os grupos que você cria na guia Portas.
- **guia Portas do Grupo.** Você pode criar e excluir portas para os grupos de entrada. O Designer exibe as portas criadas na guia Portas.

Você não pode modificar a guia Portas em uma transformação de União.

Trabalhando com Grupos e Portas

Uma transformação de União contém vários grupos de entrada e um grupo de saída. Crie grupos de entrada na guia Grupos e crie portas na guia Portas do Grupo.

Você pode criar um ou mais grupos de entrada na guia Grupos. Por padrão, o Designer cria um grupo de saída. Você não pode editá-lo nem excluí-lo.

Você pode criar portas, copiando-as de uma transformação, ou pode criá-las manualmente. Quando você cria portas na guia Portas do Grupo, o Designer cria portas de entrada em cada grupo de entrada e portas de saída no grupo de saída. O Designer utiliza os nomes de portas especificados na guia Portas do Grupo para cada porta de entrada e saída e anexa um número para que cada nome de porta na transformação seja exclusivo. Ele também utiliza os mesmos metadados para cada porta, como tipo de dados, precisão e escala.

A guia Portas exibe os grupos e as portas que você cria. Você não pode editar informações dos grupos e das portas na guia Portas. Utilize a guia Grupos e Portas do Grupo para editar grupos e portas.

Criação de uma Transformação União

Use o seguinte procedimento para criar uma transformação de União:

1. No Mapping Designer, clique em Transformações > Criar.
2. Selecione a transformação União e insira o nome da transformação.
A convenção de nomenclatura das transformações União é *UN_TransformationName*.
3. Insira uma descrição para a transformação. Clique em Criar e, em seguida, em Concluído.
4. Clique na guia Grupos.
5. Adicione um grupo de entrada para cada pipeline ou ramificação de pipeline que você deseja mesclar.
O Designer atribuirá um nome padrão para cada grupo, mas eles podem ser renomeados.
6. Clique na guia Portas do Grupo.
7. Adicione uma nova porta para cada linha de dados que você deseja mesclar.
8. Insira as propriedades da porta, como nome e tipo de dados.
9. Clique na guia Propriedades para configurar o nível de rastreamento.

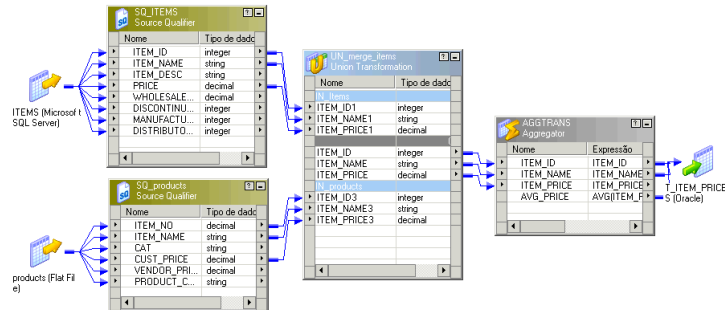
10. Clique em OK.

Usando uma Transformação de União em um Mapeamento

A transformação de União é uma transformação de grupos de entrada múltiplos sem bloqueio. Você pode conectar os grupos de entrada a várias ramificações em um único pipeline ou a vários pipelines de origem.

Quando você adiciona uma transformação de União a um mapeamento, deve verificar se conectou as mesmas portas em todos os grupos de entrada. Se você conectar todas as portas em um grupo de entrada e não conectar uma porta em outro, o Serviço de Integração passará valores NULL à porta desconectada.

A figura a seguir mostra um mapeamento com uma transformação de União:



Quando uma transformação de União em um mapeamento recebe dados de um único gerador de transação, o Serviço de Integração propaga limites de transação. Quando a transformação recebe dados de diversos geradores de transação, o Serviço de Integração descarta todas as linhas das saídas e limites de transação recebidos em uma transação aberta.

CAPÍTULO 31

Transformação Dados Não Estruturados

Este capítulo inclui os seguintes tópicos:

- [Visão Geral do Data Transformation não estruturado, 502](#)
- [Configuração da opção de dados não estruturados, 503](#)
- [Tipos de Serviço de Data Transformation, 504](#)
- [Componentes da Transformação de Dados Não Estruturados, 505](#)
- [Portas da Transformação de Dados Não Estruturados, 507](#)
- [Nomes de Serviço da Transformação de Dados não Estruturados, 510](#)
- [Hierarquias Relacionais, 510](#)
- [Mapeamentos, 511](#)
- [Criar uma transformação de dados não estruturados, 514](#)

Visão Geral do Data Transformation não estruturado

O Data Transformation não estruturado processa formatos de arquivos não estruturados e semiestruturados, como formatos de mensagens, páginas HTML e documentos PDF. Também transforma formatos estruturados, como ACORD, HIPAA, HL7, EDI-X12, EDIFACT, e SWIFT.

O Data Transformation não estruturado chama um serviço de Data Transformation de uma sessão do PowerCenter. O Data Transformation é o aplicativo que transforma formatos de arquivos não estruturados e semiestruturados. Você pode passar dados do Data Transformation não estruturado para um serviço Data Transformation, transformá-los e retornar os dados transformados ao pipeline. O Data Transformation não estruturado pode ser uma transformação ativa ou passiva.

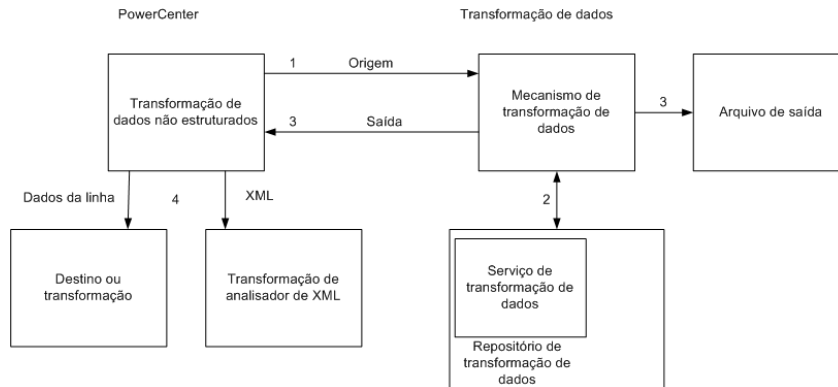
Data Transformation

- Informatica Developer. Uma ferramenta de desenvolvimento que permite criar e configurar projetos de transformação.
- Serviço de Data Transformation. Um projeto do Data Transformation implantado no repositório do Data Transformation e pronto para execução.
- Repositório do Data Transformation Um diretório que armazena serviços executáveis que você cria no Informatica Developer. Você pode implantar projetos em diferentes repositórios, como repositórios para teste e serviços de produção.

- Mecanismo de Data Transformation. Um processador que executa os serviços implantados no repositório.

Ao executar um serviço, o Mecanismo de Data Transformation grava ou retorna os dados de saída ao Serviço de Integração. Ao retornar a saída para o Serviço de Integração, o Mecanismo de Data Transformation retorna dados XML. Você pode configurar o Data Transformation não estruturado para retornar XML em uma porta de saída ou configurar grupos de saída para retornar dados de linhas.

A figura a seguir mostra a interface entre o Data Transformation não estruturado do PowerCenter e o serviço de Data Transformation que transforma dados de uma sessão do PowerCenter:



Os seguintes eventos ocorrem ao chamar um serviço de Data Transformation de uma sessão do PowerCenter:

1. A Data Transformation não estruturado passa dados de origem para o Mecanismo de Data Transformation.
2. O Mecanismo de Data Transformation executa um Serviço de Data Transformation para transformar os dados. O Serviço de Data Transformation fica na pasta do Repositório do Data Transformation.
3. O Mecanismo de Data Transformation grava dados transformados diretamente em arquivos de saída ou os retorna ao Data Transformation não estruturado.
4. O Data Transformation não estruturado retorna dados XML ou de linhas. Se o Data Transformation não estruturado retornar XML, conecte-o a uma transformação Analisador XML no mapeamento.

Configuração da opção de dados não estruturados

A transformação de dados não estruturados é instalada no PowerCenter. A transformação de dados possui um instalador em separado. Instale os componentes Servidor e Cliente da transformação de dados depois de instalar o PowerCenter.

Para instalar a opção Dados não estruturados, conclua as seguintes etapas:

1. Instale o PowerCenter.
2. Instale a transformação de dados.

Quando você instala a transformação de dados, recebe um prompt para selecionar um Java Runtime Environment. No prompt, navegue até o JRE usado pelo PowerCenter.

Por padrão, a instalação configura o Mecanismo de transformação de dados para executar dentro do processo. Para usar a transformação de dados com a transformação de dados não estruturados, não configure a transformação de dados para executar fora do processo.

3. Configure a pasta de repositório transformação de dados

Configuração do diretório do repositório do Data Transformation

O repositório Data Transformation contém serviços executáveis de Data Transformation. Quando você instala o Data Transformation, o instalador cria o repositório na seguinte pasta:

```
<Data_Transformation_install_dir>\DataTransformation\ServiceDB
```

Para configurar um local de pasta de repositório diferente, abra a Configuração do Data Transformation do menu Iniciar. O local do repositório está no seguinte caminho na Configuração da Data Transformation:

Configuração CM > Repositório CM > Sistema de arquivo > Caminho base

Se o Informatica Developer puder acessar o sistema de arquivos remoto, você poderá alterar o repositório do Data Transformation para um local remoto e implantar os serviços diretamente do Informatica Developer no sistema que executa o Serviço de Integração.

Copie os arquivos personalizados dos diretórios <Data_Transformation_install_dir>\DataTransformation\autoInclude\user e <Data_Transformation_install_dir>\DataTransformation\externLibs\user para os mesmos diretórios no nó que executa o Serviço de Integração.

Tipos de Serviço de Data Transformation

Ao criar uma transformação de Processador de Dados no Informatica Developer, escolha um objeto ou componente para definir o tipo de serviço de transformação. Data Transformation oferece os seguintes tipos de serviços que transformam dados:

- **Analizador.** Converte documentos de origem em XML ou JSON. A saída de um analisador sempre é XML. A entrada pode ter qualquer formato, como texto, HTML, Word, PDF ou HL7.
- **Serializador.** Converte um arquivo XML ou JSON em um documento de saída de qualquer formato. A saída de um serializador pode ser em qualquer formato, como um documento de texto, um documento HTML ou um PDF.
- **Mapeador e XMap.** Converte um documento de origem XML ou JSON em outra estrutura XML ou JSON. Um mapeador processa a entrada de maneira semelhante a um serializador. Ele gera a saída de maneira semelhante a um analisador. A entrada e a saída são XML ou JSON completamente estruturado.
- **Transformador.** Modifica os dados em qualquer formato. Adiciona, remove, converte ou altera o texto. Use transformadores com um analisador, mapeador ou serializador. Você também pode executar um transformador como componente independente.
- **Streamer.** Divide grandes documentos de entrada, como fluxos de dados multigigabytes, em segmentos. O streamer processa documentos que contêm múltiplas mensagens ou registros, como arquivos HIPAA ou EDI.

Componentes da Transformação de Dados Não Estruturados

A transformação de dados não estruturados contém as seguintes guias:

- **Transformação.** Insira o nome e a descrição da transformação. A convenção de nomenclatura para uma transformação de dados não estruturados é UD_TransformationName. Você também pode tornar a transformação de dados não estruturados reutilizável.
- **Propriedades.** Configure as propriedades gerais da transformação de dados não estruturados, como IsPartitionable e Output is Repeatable.
- **Configurações UDT.** Modifique configurações da transformação de dados não estruturados, como tipo de entrada, tipo de saída e nome do serviço.
- **Portas UDT.** Configure as portas e os atributos da transformação de dados não estruturados.
- **Hierarquia de entrada.** Defina uma hierarquia de grupos e portas de entrada para permitir que a transformação Dados Não Estruturados leia linhas de dados.
- **Hierarquia de Saída.** Defina uma hierarquia de grupos e portas de saída para permitir que a transformação Dados Não Estruturados grave linhas em destinos relacionais.

Guia Propriedades

Configure as propriedades gerais de transformação de Dados Não Estruturados na guia **Propriedades**.

A tabela a seguir descreve propriedades da guia **Propriedades** que você pode configurar:

Propriedade	Descrição
Nível de Rastreamento	O volume de detalhes incluído no log da sessão quando você executa uma sessão que contém essa transformação. O padrão é Normal.
IsPartitionable	A transformação pode ser executada em mais de uma partição. Selecione uma das seguintes opções: <ul style="list-style-type: none">- N° A transformação não pode ser particionada.- Localmente. A transformação pode ser particionada, mas o Serviço de Integração deve executar todas as partições da pipeline no mesmo nó.- Na Grade. A transformação pode ser particionada e o Serviço de Integração pode distribuir cada partição em nós diferentes. O padrão é Na Grade.
Saída é Repetível	A ordem dos dados de saída é consistente entre as execuções de sessão. <ul style="list-style-type: none">- Nunca. A ordem dos dados de saída é inconsistente entre as execuções de sessão.- Baseada na Ordem de Entrada. A ordem de saída é consistente entre as execuções de sessão quando a ordem dos dados de entrada é consistente entre elas.- Sempre. A ordem dos dados de saída será consistente entre as execuções de sessão mesmo se a ordem de entrada for inconsistente entre elas. O padrão é Nunca para transformações ativas. O padrão é Baseada na Ordem de Entrada para execuções de transformação passiva.
Saída é Determinística	Indica se a transformação gera dados de saída consistentes entre as execuções de sessão. Ative esta propriedade para executar a recuperação em sessões que usam essa transformação.

Aviso: Se você configurar uma transformação como repetível e determinística, será sua responsabilidade garantir que os dados sejam repetíveis e determinísticos. Se você tentar recuperar uma sessão com

transformações que não gerem os mesmos dados entre a sessão e a recuperação, o processo de recuperação poderá resultar em dados corrompidos.

Guia Configurações de UDT

Configure os atributos da transformação de dados não estruturados na guia **Configurações de UDT**.

A tabela a seguir descreve os atributos da guia de **configurações de UDT**:

Atributo	Descrição
InputType	<p>Tipo de dados de entrada transmitidos pela transformação Dados Não Estruturados ao Mecanismo de Data Transformation. Escolha um dos seguintes tipos de entrada:</p> <ul style="list-style-type: none">- Buffer. A transformação Dados Não Estruturados recebe dados de origem na porta InputBuffer e transmite dados da porta para o Mecanismo de Data Transformation.- Arquivo. A transformação Dados Não Estruturados recebe um caminho de arquivo de origem na porta InputBuffer e o transmite ao Mecanismo de Data Transformation. O Mecanismo de Data Transformation abre o arquivo de origem.
OutputType	<p>Tipo de dados de saída retornados pela transformação Dados Não Estruturados ou pelo Mecanismo de Data Transformation. Escolha um dos seguintes tipos de saída:</p> <ul style="list-style-type: none">- Buffer. A transformação Dados Não Estruturados retornará dados XML na porta OutputBuffer, a menos que você configure uma hierarquia relacional de portas de saída. Se você configurar uma hierarquia relacional de portas, a transformação Dados Não Estruturados não gravará na porta OutputBuffer.- Arquivo. O Mecanismo de Data Transformation grava a saída em um arquivo. Ele não retornará os dados para a transformação Dados Não Estruturados, a menos que você configure uma hierarquia relacional de portas na transformação Dados Não Estruturados.- Dividindo. A transformação Dados Não Estruturados divide um arquivo XML de saída grande em arquivos menores que possam ajustar-se à porta OutputBuffer. Transmite os arquivos XML divididos à transformação Analisador XML.
ServiceName	<p>Nome do serviço de Data Transformation. O serviço deve ser apresentado no repositório Data Transformation local.</p>
Tamanho da Parte do Streamer	<p>O tamanho do buffer dos dados que a transformação Dados Não Estruturados transmite ao Mecanismo de Data Transformation quando o serviçoData Transformation executa um streamer. Os valores válidos vão de 1 a 1.000.000 KB. O padrão é 256 KB.</p>
Nome do Serviço Dinâmico	<p>Execute um serviço Data Transformation diferente para cada linha de entrada. Quando o Nome do Serviço Dinâmico está habilitado, a transformação Dados Não Estruturados recebe o nome do serviço na porta de entrada Nome do Serviço.</p> <p>Quando o nome do Serviço Dinâmico está desabilitado, a transformação Dados Não Estruturados executa o mesmo serviço para cada linha de entrada. O atributo Nome do Serviço nas Configurações de UDT deve conter um nome de serviço. O padrão é desabilitado.</p>

Atributo	Descrição
Nível de Rastreamento de Status	<p>Defina o nível de mensagens de status do serviço Data Transformation.</p> <ul style="list-style-type: none"> - Somente Descrição. Retorna um código de status e uma descrição curta para indicar se o serviço Data Transformation teve êxito ou falhou. - Status Completo. Retorna um código de status e uma mensagem de status do serviço Data Transformation em XML. - Nenhum. Não retorna status do serviço Data Transformation. O padrão é nenhum.
Permitir Liberação de Entrada	<p>Habilite Permitir Liberação de Entrada quando você configurar grupos de entrada de portas. A transformação Dados Não Estruturados cria o XML quando ela tem todos os dados de um grupo. Configure o mapeamento para que a transformação receba dados classificados por grupo.</p> <p>Quando a liberação de entrada não está habilitada, a transformação Dados Não Estruturados armazena os dados na memória e cria o XML depois que recebe os dados de todos os grupos.</p>

Exibindo Mensagens de Rastreamento de Status

Você pode visualizar as mensagens de status do serviço de Data Transformation. Defina o nível do rastreamento de status como Apenas Descrição ou Status Completo. O Designer cria a porta UDT_Status_Code e as portas de saída UDT_Status_Message na transformação de dados não estruturados.

Se você escolher Apenas Descrição, o Mecanismo de Data Transformation retornará um código de status e uma destas mensagens de status:

Status Code	Mensagem de Status
1	Sucesso
2	Aviso
3	Falha
4	Erro
5	Erro Fatal

Se você escolher Status Completo, o Mecanismo de Data Transformation retornará um código de status e a mensagem de erro do serviço de Data Transformation. A mensagem estará em formato XML.

Portas da Transformação de Dados Não Estruturados

Quando você cria uma transformação de dados não estruturados, o Designer cria portas padrão. Ele cria outras portas com base em como você configura a transformação. Os tipos de entrada e saída da

transformação de dados não estruturados determinam como a transformação de dados não estruturados passa e recebe dados do Mecanismo de Data Transformation.

A tabela a seguir descreve as portas padrão da transformação de dados não estruturados:

Porta	Entrada/ Saída	Descrição
InputBuffer	Entrada	Recebe dados de origem quando o tipo de entrada é buffer. Recebe o nome e o caminho de um arquivo de origem quando o tipo de entrada é arquivo.
OutputBuffer	Saída	Retorna dados XML quando o tipo de saída é buffer. Retorna o nome do arquivo de saída quando o tipo de saída é arquivo. Não retorna dados quando você configura grupos de portas de saída hierárquicos.

A tabela a seguir descreve outras portas da transformação de dados não estruturados criadas pelo Designer quando você configura a transformação:

Porta	Entrada/ Saída	Descrição
OutputFileName	Entrada	Recebe o nome para um arquivo de saída quando o tipo de saída é arquivo.
ServiceName	Entrada	Recebe o nome de um serviço de Data Transformation quando você habilita Nome do Serviço Dinâmico.
UDT_Status_Code	Saída	Retorna um código de status do Mecanismo de Data Transformation quando o nível de rastreamento de status é Somente Descrição ou Status Completo.
UDT_Status_Message	Saída	Retorna uma mensagem de status do Mecanismo de Data Transformation quando o nível de rastreamento de status é Somente Descrição ou Status Completo.

Nota: Você pode adicionar grupos de portas de saída para destinos relacionais nas guias Hierarquia de **Entrada** ou **Saída**. Quando você configura grupos de portas, uma mensagem aparece na guia **Portas UDT**, dizendo que as portas e grupos hierárquicos são definidos em outra guia.

Tipos de entrada e saída

O tipo de entrada determina o tipo de dados que o Serviço de Integração passa para o Mecanismo de Data Transformation. O tipo da entrada determina se ela é um caminho de arquivos de origem ou dados.

Configure um dos seguintes tipos de entrada:

- **Buffer.** A transformação de dados não estruturados recebe dados de origem na porta InputBuffer. O Serviço de Integração passa linhas de origem da porta InputBuffer para o Mecanismo de Data Transformation.
- **Arquivo.** A transformação de dados não estruturados recebe o caminho do arquivo de origem na porta InputBuffer. O Serviço de Integração passa o caminho do arquivo de origem para o Mecanismo de Data Transformation. O Mecanismo de Data Transformation abre o arquivo de origem. Utilize o tipo de entrada do arquivo para analisar dados binários, como arquivos do Microsoft Excel ou do Microsoft Word.

Se você não quiser definir portas e grupos de saída, a transformação de dados não estruturados retornará dados com base no tipo de saída.

Configure um dos seguintes tipos de saída:

- **Buffer.** A transformação de dados não estruturados retorna XML através da porta Outputbuffer. Você deve conectar uma transformação Analisador XML à porta Outputbuffer.
- **Arquivo.** O Mecanismo de Data Transformation grava o arquivo de saída, em vez de passar dados para o Serviço de Integração. O Mecanismo de Data Transformation nomeia o arquivo de saída com base no nome do arquivo na porta OutputFilename. Escolha o tipo de saída Arquivo para transformar XML em dados binários, como um arquivo PDF ou um arquivo Microsoft Excel.

O Serviço de Integração retorna o nome do arquivo de saída na porta OutputBuffer para cada linha de origem. Se o nome do arquivo de saída estiver em branco, o Serviço de Integração retornará um erro de linha. Quando ocorre um erro, o Serviço de Integração grava um valor nulo na porta OutputBuffer e retorna uma linha de erro.

- **Dividindo.** A transformação de dados não estruturados divide dados XML do Mecanismo de Data Transformation em vários segmentos. Escolha o tipo de saída Divisão quando a transformação de dados não estruturados retornar arquivos XML que são muito grandes para a porta OutputBuffer. Quando você configurar a saída Divisão, passe os dados XML para a transformação Analisador XML. Configure a transformação Analisador XML para processar várias linhas XML como um único arquivo XML.

Portas adicionais da transformação de dados não estruturados

Um serviço de Data Transformation pode precisar de vários arquivos de entrada, nomes de arquivos e parâmetros. Pode retornar vários arquivos de saída. Ao criar uma Transformação de Dados Não Estruturados, o Designer criará uma porta InputBuffer e uma porta OutputBuffer. Caso seja necessário transferir mais arquivos ou nomes de arquivos entre a Data Transformation e o Mecanismo de Transformação de Dados, acrescente as portas de entrada ou de saída. É possível adicionar portas manualmente ou a partir do serviço de Data Transformation.

A tabela a seguir descreve as portas que podem ser criadas na guia **Portas UDT**:

Tipo de Porta	Entrada/ Saída	Descrição
Entrada Adicional (buffer)	Entrada	Recebe dados de entrada para transferir ao Mecanismo de Data Transformation.
Entrada adicional (arquivo)	Entrada	Recebe o nome do arquivo e o caminho para que o Mecanismo de Data Transformation abra.
Parâmetro de Serviço	Entrada	Recebe um parâmetro de entrada para um serviço de Data Transformation.
Saída adicional (buffer)	Saída	Recebe dados em XML do Mecanismo de Data Transformation.
Saída Adicional (arquivo)	Saída	Recebe o nome de um arquivo de saída do Mecanismo de Data Transformation.
Passagem	Entrada/ Saída	Transfere dados por meio da Transformação de Dados Não Estruturados sem efetuar alteração.

Criando Portas a partir de um serviço de Data Transformation

O serviço de Data Transformation pode exigir parâmetros de entrada, arquivos de entrada adicionais ou variáveis definidas pelo usuário. O serviço pode retornar mais de um arquivo de saída para a transformação

de dados não estruturados. Você pode adicionar portas que transferem parâmetros e arquivos de entrada e saída adicionais. O Designer cria portas que correspondem às portas no serviço de Data Transformation.

Nota: Você deve configurar um nome de serviço para preencher as portas a partir de um serviço.

1. Clique na guia **Portas** na transformação de dados não estruturados.
2. Clique em **Preencher a partir do serviço**.

O Designer exibe os requisitos dos parâmetros de serviço e das portas de entrada e de saída adicionais a partir do serviço de Data Transformation. Os parâmetros de serviço incluem as variáveis do sistema de Data Transformation e as variáveis definidas pelo usuário.

3. Selecione as portas para criar e configurar cada uma delas como uma porta de buffer ou de arquivo.
4. Clique em **Preencher** para criar as portas que você selecionar. Você pode selecionar todas as portas exibidas.

Nomes de Serviço da Transformação de Dados não Estruturados

Quando você cria uma transformação de dados não estruturados, o Designer exibe uma lista dos serviços de Data Transformation que estão no repositório Data Transformation. Escolha o nome de um serviço de Data Transformation que você deseja chamar, a partir da transformação de dados não estruturados. Você pode alterar o nome do serviço depois de criar a transformação. O nome do serviço aparece na guia

Configurações de UDT.

Para executar um serviço de Data Transformation diferente para cada linha de origem, ative o atributo Nome do Serviço Dinâmico. Transfira o nome do serviço com cada linha de origem. O Designer cria a porta de entrada ServiceName quando você ativa os nomes de serviço dinâmicos.

Quando você ativa os nomes do serviço dinâmico, não pode criar portas a partir de um serviço de Data Transformation.

Ao definir uma estrutura relacional de portas de entrada, você não pode ativar os nomes do serviço dinâmico. Cada serviço de Data Transformation poderá exigir diferentes dados de entrada.

Hierarquias Relacionais

Você pode definir grupos de portas e uma estrutura relacional para os grupos. Para criar uma hierarquia de portas de entrada, configure as portas na guia **Hierarquia de Entrada**. Para transmitir dados de linha para tabelas relacionais ou para outros destinos, configure as portas de saída na guia **Hierarquia de Saída**.

Quando você define uma estrutura relacional de portas de entrada, a transformação Dados Não Estruturados gera o XML a ser transmitido para um serviço de Data Transformation. Para aumentar o desempenho, habilite o Serviço de Integração do PowerCenter para liberar o XML para a transformação Dados Não Estruturados. Ao habilitar a liberação de entrada, os Serviços de Integração do PowerCenter liberam o XML de cada grupo depois de receber todos os dados para o valor raiz. Por exemplo, você tem um grupo de funcionários e um grupo de endereços de funcionários. O Serviço de Integração do PowerCenter poderá liberar dados dos dois grupos para a transformação Dados Não Estruturados sempre que houver um funcionário diferente nos dados. Classifique os dados em cada grupo por chave primária do grupo raiz. Você poderá unir grupos no pipeline se um grupo não tiver a mesma chave.

Quando você configura grupos de saída, eles representam as tabelas relacionais ou os destinos para os quais você deseja passar os dados de saída. O Mecanismo de Data Transformation retorna linhas para portas de grupo, em vez de gravar um arquivo XML na porta OutputBuffer. A transformação grava linhas com base no tipo de saída.

Crie uma hierarquia de grupos no painel esquerdo da guia **Hierarquia de Saída**. Todos os grupos estão no grupo raiz. Não é possível excluir a raiz. Cada grupo pode conter portas e outros grupos. A estrutura do grupo representa o relacionamento entre as tabelas de destino. Quando define um grupo dentro de outro, você define um relacionamento de pai para filho entre eles. O Designer define um relacionamento de chave primária-chave estrangeira entre os grupos com uma chave gerada.

Selecione um grupo para exibir as portas dele. Você pode adicionar ou excluir portas no grupo. Quando você adiciona uma porta, o Designer criará uma configuração de porta padrão. Mude o nome da porta, o tipo de dados e a precisão. Se a porta precisar conter dados, selecione Não É Nulo. Caso contrário, os dados de saída são opcionais.

Quando você exibir a transformação de dados não estruturados no espaço de trabalho, cada porta de um grupo de transformações terá um prefixo com o nome do grupo.

Ao excluir um grupo, você exclui as portas do grupo e os grupos filho.

Exportando o Esquema de Hierarquia

Ao definir grupos de saída hierárquicos no Data Transformation não estruturado, você precisa definir a mesma estrutura no serviço de Data Transformation criado para transformar os dados. Exporte a estrutura hierárquica como um arquivo de esquema XML a partir do Data Transformation não estruturado. Importe o esquema para a transformação de Processador de Dados. Em seguida, você pode mapear o conteúdo de um documento de origem para os elementos e atributos XML na transformação de Processador de Dados.

Para exportar a hierarquia de grupo na guia **Hierarquia Relacional**, clique em **Exportar para Esquema XML**. Escolha um nome e uma localização para o arquivo XSD. Escolha um local que você possa acessar ao importar o esquema com o Informatica Developer.

O Designer cria um esquema com o seguinte namespace:

```
"www.informatica.com/UDT/XSD/<mappingName_<Transformation_Name>>"
```

O esquema inclui o seguinte comentário:

```
<!-- ===== AUTO-GENERATED FILE - DO NOT EDIT ===== -->
<!-- ===== This file has been generated by Informatica PowerCenter ===== -->
```

Caso você modifique o esquema, o Data Transformation pode retornar dados que não têm o mesmo formato que a porta de saída no Data Transformation não estruturado.

Os elementos de XML no esquema representam as portas de saída na hierarquia. As colunas que podem conter valores nulos têm um atributo de XML minOccurs=0 e maxOccurs=1.

Mapeamentos

Ao criar um mapeamento, desenvolva-o de acordo com o tipo de serviço de Data Transformation que será executado. Por exemplo, o Analisador e o Mapeador do Data Transformation geram dados XML. Você pode configurar o Data Transformation não estruturado para retornar linhas dos dados XML ou pode configurá-lo para retornar um arquivo XML.

O componente Serializador do Data Transformation pode gerar qualquer saída XML. Pode gerar arquivos HTML ou binários como os do Microsoft Word ou Microsoft Excel. Quando a saída são os dados binários, o

Mecanismo de Data Transformation escreve a saída em um arquivo, em vez de transferi-lo para o Data Transformation não estruturado.

Os seguintes exemplos mostram como configurar o mapeamento com um Data Transformation não estruturado.

Analizando Documentos de Word para Tabelas Relacionais

É possível extrair informações de pedidos de um documento do Microsoft Word e registrar as mesmas informações em uma tabela cabeçalhos de pedidos e uma tabela de detalhes de pedidos. Configure uma transformação de dados não estruturados para chamar um serviço de analisador de Transformação de dados e transfira o nome de cada documento do Word a ser analisado. O Mecanismo de Data Transformation abre o documento do Word, analisa-o e retorna as linhas à transformação de dados não estruturados. A transformação de dados não estruturados transfere o cabeçalho de pedidos e os detalhes de pedidos aos destinos relacionais.

O mapeamento contém os seguintes objetos:

- Transformação Qualificador de Origem. Transmite cada nome de arquivo do Microsoft Word para a transformação Dados Não Estruturados. O nome do arquivo de origem contém o caminho completo ao arquivo que contém as informações de pedido.
- Transformação Dados Não Estruturados. O tipo de entrada é arquivo. O tipo de saída é buffer. A transformação contém um grupo de saída de cabeçalho de pedido e um grupo de saída de detalhes de pedido. Os grupos têm uma relação das chaves externas primária.

A transformação de dados não estruturados recebe o nome do arquivo de origem na porta InputBuffer. Transfere o nome ao Mecanismo de Data Transformation. O Mecanismo de Data Transformation executa um serviço de analisador para extrair as linhas de cabeçalho de pedidos e detalhe de pedidos do documento de Word. O Mecanismo de Data Transformation devolve os dados à transformação de dados não estruturados. A transformação de dados não estruturados transfere os dados do grupo de cabeçalho de pedidos e do grupo de detalhes de pedidos aos destinos relacionais.

- Destinos relacionais. Recebe as linhas da transformação de dados não estruturados.

Criando uma Planilha do Excel a Partir de um XML

É possível extrair nomes e endereços de funcionários de um arquivo XML e criar uma planilha do Microsoft Excel com a lista de nomes.

O mapeamento contém os seguintes componentes:

- Arquivo de origem XML. Contém nomes e endereços de funcionários.
- Transformação Qualificador de Origem. Transfere dados em XML e um nome de arquivo de saída à Transformação de Dados Não Estruturados. O arquivo XML contém nomes de funcionários.
- Transformação Dados Não Estruturados. O tipo de entrada é buffer e o tipo de saída é arquivo. A Transformação de Dados Não Estruturados recebe os dados em XML na porta InputBuffer e o nome do arquivo na porta OutputFileName. Ele transfere os dados XML e o nome de arquivo para o Mecanismo de Data Transformation.

O Mecanismo de Data Transformation executa um serviço de serializador para transformar os dados em XML em um arquivo do Microsoft Excel. Ele registra o arquivo do Excel com um nome de arquivo baseado no valor do OutputFilename.

A transformação Dados Não Estruturados recebe somente o nome do arquivo de saída do Mecanismo de Data Transformation. A porta OutputBuffer da Transformação de Dados Não Estruturados retorna o valor do OutputFilename.

- Destino de Arquivo Simples Recebe o nome do arquivo de saída.

Saída de Arquivo XML Dividido

Os componentes do Analisador de Data Transformation e do mapeador têm a capacidade de transformar dados de qualquer formato e gerar dados em XML. Quando os dados em XML são grandes, é possível dividir o XML em segmentos e transferir esses segmentos para uma transformação de Analisador XML. A transformação do Analisador XML recebe os segmentos e processa os dados em XML como um documento.

Ao configurar a Transformação de dados não estruturados para que divida a saída XML, a Transformação de dados não estruturados retorna XMLs baseados no tamanho da porta OutputBuffer. Se o tamanho do arquivo XML for maior do que a precisão da porta de saída, o Serviço de Integração divide o XML em arquivos iguais ou menores que o tamanho da porta. A transformação do Analisador XML analisa o XML e transfere as linhas para tabelas relacionais ou outros destinos.

Por exemplo, é possível extrair o cabeçalho de pedidos e informações sobre detalhes dos documentos do Microsoft Word com um serviço de analisador de Data Transformation.

O mapeamento contém os seguintes componentes:

- Transformação Qualificador de Origem. Passa o nome de arquivo do documento Word para a transformação Dados Não Estruturados. O nome do arquivo de origem contém o caminho completo ao arquivo que contém as informações de pedido.
- Transformação Dados Não Estruturados. O tipo de entrada é arquivo. O tipo de saída é divisão. A transformação de dados não estruturados recebe o nome do arquivo de origem na porta InputBuffer. Ele transfere o nome de arquivo para o Mecanismo de Data Transformation. O Mecanismo de Data Transformation abre o documento de origem, analisa-o e retorna dados em XML à transformação de dados não estruturados.

A transformação de dados não estruturados recebe os dados em XML, divide o arquivo XML em arquivos menores e transfere os segmentos para uma transformação de Analisador XML. A transformação de dados não estruturados retorna dados em segmentos inferiores ao tamanho da porta OutputBuffer. Quando a transformação retorna dados em XML em diversos segmentos, ela gera os mesmos dados de passagem para cada linha. A transformação de dados não estruturados retorna dados em portas de passagem quando uma linha tem ou não êxito.

- Transformação de Analisador de XML. A propriedade da sessão Ativar fluxo de entrada está ativada. A transformação do Analisador XML recebe os dados em XML na porta DataInput. Os dados de entrada são divididos em segmentos. A transformação do Analisador XML separa os dados em XML em cabeçalho de pedido e linhas de entrada. Ela transfere cabeçalhos de ordem e linhas de detalhe a destinos relacionais. Ela retorna os dados de passagem a uma transformação de Filtro.
- Transformação Filtro. Remove os dados duplicados de passagem antes de transferi-los aos destinos relacionais.
- Destinos relacionais. Recebem dados de cada grupo na transformação do Analisador XML e a transformação de Filtro.

Regras e Diretrizes para Mapeamentos de Dados Não Estruturados

Use as seguintes regras e diretrizes ao criar um mapeamento de dados não estruturado:

- Quando você configura grupos hierárquicos de portas de saída, o Serviço de Integração grava nos grupos de portas, em vez de gravar na porta OutputBuffer. O Serviço de Integração grava nos grupos de portas, independentemente do tipo de saída que você definir para a transformação.
- Se uma transformação de dados não estruturados tiver um tipo de saída Arquivo e você não tiver definido portas de saída em grupo, será necessário vincular a porta OutputBuffer a uma transformação downstream. Caso contrário, o mapeamento será inválido. A porta OutputBuffer contém o nome do arquivo de saída quando o serviço de Data Transformation grava no arquivo de saída.

- Habilite o Nome do Serviço Dinâmico para passar um nome de serviço para a transformação de dados não estruturados na porta de entrada do Nome do Serviço. Quando você habilita o Nome do Serviço Dinâmico, o Designer cria a porta de entrada do Nome do Serviço.
- Você precisa configurar um nome de serviço com a transformação de dados não estruturados ou habilitar a opção Nome do Serviço Dinâmico. Caso contrário, o mapeamento será inválido.
- Vincule a saída XML da transformação de dados não estruturados a uma transformação Analisador XML.

Criar uma transformação de dados não estruturados

Use o PowerCenter Transformation Developer ou Mapping Designer para criar uma transformação de dados não estruturados.

1. No Mapping Designer ou no Transformation Developer, clique em **Transformação > Criar**.
2. Selecione a **transformação de dados não estruturados** como o tipo da transformação.
3. Insira um nome para a transformação.
4. Clique em **Criar**.

A caixa de diálogo **Transformação de dados não estruturados** será exibida.

5. Configure as seguintes propriedades:

Propriedade	Descrição
Nome do serviço	Nome do serviço de Data Transformation que você deseja usar. O Designer exibe os Serviços de Data Transformation na pasta de repositório Data Transformation. Não escolha um nome se você estiver planejando habilitar nomes de serviços dinâmicos. Depois de criar a transformação, você poderá adicionar um nome de serviço na guia Configurações UDT.
Tipo de Entrada	Descreve como o Mecanismo de Data Transformation recebe dados de entrada. Buffer é o padrão.
Tipo de Saída	Descreve como o Mecanismo de Data Transformation retorna os dados de saída. Buffer é o padrão.

6. Clique em **OK**.
7. Você pode alterar o nome do serviço, a entrada e o tipo de saída na guia **Configurações UDT**.
8. Configure as propriedades de transformação de Dados Não Estruturados na guia **Propriedades**.
9. Se o serviço de Data Transformation tiver mais de um arquivo de entrada ou saída, ou se ele precisar de parâmetros de entrada, você poderá adicionar portas na guia **Portas UDT**. Também é possível adicionar portas de passagem na guia **Portas**.
10. Se você quiser retornar dados de linha a partir da transformação de dados não estruturados em vez de dados XML, crie grupos de portas de saída na guia **Hierarquia relacional**.
11. Se você criar grupos de portas, exporte o esquema que descreve-os a partir da guia **Hierarquia Relacional**.
12. Importe o esquema para o projeto Data Transformation, a fim de definir a saída do projeto.

CAPÍTULO 32

Transformação Estratégia de Atualização

Este capítulo inclui os seguintes tópicos:

- [Visão Geral da Transformação de Estratégia de Atualização, 515](#)
- [Sinalizando Linhas Dentro de um Mapeamento, 516](#)
- [Definindo a Estratégia de Atualização para uma Sessão, 518](#)
- [Lista de Verificação da Estratégia de Atualização, 520](#)

Visão Geral da Transformação de Estratégia de Atualização

A transformação de Estratégia de Atualização é uma transformação ativa. Ao criar um data warehouse, você deve decidir o tipo de informação a ser armazenado nos destinos. Como parte do projeto da tabela de destino, determine se quer manter todos os dados históricos ou apenas as alterações mais recentes.

Por exemplo, você pode ter uma tabela de destino T_CUSTOMERS que contém dados de clientes. Quando o endereço de um cliente muda, talvez você queira salvar o endereço original na tabela em vez de atualizar essa parte da linha do cliente. Nesse caso, você criaria uma linha nova contendo o endereço atualizado e manteria a linha original com o endereço antigo do cliente. Isso mostra como você pode armazenar informações históricas em uma tabela de destino. Contudo, se você quiser que a tabela T_CUSTOMERS seja um instantâneo dos dados do cliente, atualize a linha do cliente e esqueça o endereço original.

O modelo que você escolher determinará como lidar com alterações a linhas existentes. No PowerCenter, você define a estratégia de atualização em dois níveis:

- **Dentro de uma sessão.** Ao configurar uma sessão, você pode instruir o Serviço de Integração para tratar todas as linhas da mesma forma (por exemplo, tratar todas as linhas como inserções) ou usar instruções codificadas no mapeamento de sessão para sinalizar linhas para operações diferentes de banco de dados.
- **Dentro de um mapeamento.** Em um mapeamento, use a transformação Estratégia de Atualização para sinalizar linhas para inserir, excluir, atualizar ou rejeitar.

Nota: Você também pode usar a transformação personalizada para sinalizar linhas para inserir, excluir, atualizar ou rejeitar.

Definindo a Estratégia de Atualização

Para definir uma estratégia de atualização, conclua as etapas a seguir:

1. Para controlar a forma como as linhas são sinalizadas para inserir, atualizar, excluir ou rejeitar dentro de um mapeamento, adicione uma transformação Estratégia de Atualização ao mapeamento. As transformações Estratégia de Atualização são essenciais, caso você queira sinalizar as linhas destinadas ao mesmo destino em operações de banco de dados diferentes, ou se você quiser rejeitar as linhas.
2. Defina como sinalizar as linhas ao configurar uma sessão. Você pode sinalizar todas as linhas para inserir, excluir ou atualizar, ou selecionar a opção controlada por dados, onde o Serviço de Integração segue as instruções codificadas nas transformações de Estratégia de Atualização dentro de um mapeamento de sessão.
3. Defina as opções inserir, atualizar e excluir para cada destino ao configurar a sessão. Você pode permitir ou não permitir inserções e exclusões de destino a destino, e escolher três formas diferentes de lidar com as atualizações.

TÓPICOS RELACIONADOS:

- [“Definindo a Estratégia de Atualização para uma Sessão” na página 518](#)

Sinalizando Linhas Dentro de um Mapeamento

Para obter o mais alto grau de controle sobre a estratégia de atualização, adicione transformações de Estratégia de Atualização a um mapeamento. A função mais importante dessa transformação é sua expressão de estratégia de atualização, utilizada para sinalizar cada linha para inserção, exclusão, atualização ou rejeição.

A tabela a seguir lista as constantes para cada operação de bancos de dados e seu equivalente numérico:

Operação	Constante	Valor numérico
Inserir	DD_INSERT	0
Atualizar	DD_UPDATE	1
Excluir	DD_DELETE	2
Rejeitar	DD_REJECT	3

O Serviço de Integração trata qualquer outro valor como uma inserção.

Encaminhamento de Linhas Rejeitadas

Você pode configurar a transformação Estratégia de Atualização para transferir linhas rejeitadas para a próxima transformação ou para descartá-las. Por padrão, o Serviço de Integração encaminha as linhas rejeitadas para a próxima transformação. O Serviço de Integração sinaliza as linhas para rejeição e as escreve no arquivo rejeitado da sessão. Se você não selecionar Encaminhar Linhas Rejeitadas, o Serviço de Integração descarta as linhas rejeitadas e as escreve no arquivo de log da sessão.

Se você ativar a manipulação do erro de linha, o Serviço de Integração escreve as linhas descartadas e rejeitadas nos logs de erros de linha. Ele não gera um arquivo rejeitado. Se quiser gravar linhas descartadas no log da sessão, além dos logs de erro de linha, você poderá ativar o rastreamento detalhado de dados.

Expressões de Estratégia de Atualização

Geralmente, a expressão da estratégia de atualização utiliza a função IIF ou DECODE da linguagem de transformação para testar cada linha e ver se ela atende a uma condição específica. Caso atenda, você poderá, então, atribuir a cada linha um código numérico, a fim de marcá-la para uma operação de banco de dados específica. Por exemplo, a instrução IIF a seguir sinaliza uma linha para ser rejeitada caso a data de entrada seja posterior à de aplicação. Caso contrário, a instrução sinaliza a linha para atualização:

```
IIF( ( ENTRY_DATE > APPLY_DATE), DD_REJECT, DD_UPDATE )
```

Para criar uma transformação de Estratégia de Atualização:

1. No Mapping Designer, adicione uma transformação Estratégia de Atualização a um mapeamento.
2. Clique em Layout > Vincular Colunas.
3. Arraste todas as portas de outra transformação que representem os dados os quais você deseja passar pela transformação Estratégia de Atualização.

Na transformação Estratégia de Atualização, o Designer criará uma cópia de cada porta arrastada. O Designer também conecta a nova porta à porta original. Cada porta na transformação Estratégia de Atualização é uma combinação porta de entrada/saída.

Normalmente, você precisaria selecionar todas as colunas orientadas a um destino específico. Depois de passar pela transformação Estratégia de Atualização, essas informações são sinalizadas para atualização, inserção, exclusão ou rejeição.

4. Abra a transformação Estratégia de Atualização e renomeie-a.
A convenção de nomenclatura das transformações Estratégia de Atualização é *UPD_Nome da Transformação*.
5. Clique na guia Propriedades.
6. Clique no botão no campo Expressão da Estratégia de Atualização.
O Editor de Expressão será exibido.
7. Insira uma expressão da estratégia de atualização para sinalizar linhas como inserções, exclusões, atualizações ou rejeições.
8. Valide a expressão e clique em OK.
9. Clique em OK.
10. Conecte as portas na transformação Estratégia de Atualização a outra transformação ou a uma instância de destino.

Transformações de Agregador e de Estratégia de Atualização

Ao conectar as transformações Agregador e Estratégia de Atualização como parte do mesmo pipeline, coloque o Agregador antes da transformação de Estratégia de Atualização. Nesta ordem, o Serviço de Integração realiza o cálculo de agregado e, em seguida, sinaliza as linhas que contêm os resultados deste cálculo para inserir, atualizar, excluir ou rejeitar.

Se colocar a estratégia de atualização antes da transformação de Agregador, você deverá considerar como a transformação Agregador tratará as linhas sinalizadas para diferentes operações. Nesta ordem, o Serviço de Integração sinaliza linhas para inserir, atualizar, excluir ou rejeitar antes de executar o cálculo de agregado. A maneira como você sinaliza uma linha específica determina como a transformação Agregador trata qualquer valor nessa linha usada no cálculo. Por exemplo, se você sinalizar uma linha para exclusão e depois

usá-la para calcular a soma, o Serviço de Integração subtrairá o valor que aparece nesta linha. Se você sinaliza uma linha para rejeitar e, em seguida, usa a linha para calcular a soma, o Serviço de Integração não incluirá o valor desta linha. Se você sinaliza uma linha para inserção ou atualização e, em seguida, usa a linha para calcular a soma, o Serviço de Integração adiciona o valor nessa linha para a soma.

Transformações Pesquisa e Estratégia de Atualização

Ao criar um mapeamento com uma transformação de pesquisa que utilize um cache de pesquisa dinâmica, é necessário usar transformações de Estratégia de Atualização para sinalizar as linhas das tabelas de destino. Ao configurar uma sessão utilizando as transformações de Atualização de Estratégia e um cache de pesquisa dinâmica, é necessário definir algumas propriedades de sessão.

É necessário definir a opção Tratar Linhas de Origem Como Controlado por Dados. Especifique esta opção na guia Propriedades nas propriedades da sessão.

Também é necessário definir as seguintes opções de tabela de origem de estratégia de atualização:

- Selecione Inserir
- Selecionar Atualização como Atualização
- Não selecione Excluir

Essas opções de tabela de origem de estratégia de atualização garantem que o Serviço de Integração atualizará linhas marcadas para atualização e inserirá linhas marcadas para inserção.

Se você não escolher Controlado por Dados, o Serviço de Integração sinalizará todas as linhas para a operação dos bancos de dados que você especificar na opção Tratar Linhas de Origem Como e não usará as transformações de Estratégia de Atualização no mapeamento para sinalizar as linhas. O Serviço de Integração não insere e atualiza as linhas corretas. Se você não escolher Atualização como Atualização, o Serviço de Integração não atualiza corretamente as linhas sinalizadas para serem atualizadas na tabela de destino. Consequentemente, o cache de pesquisa e a tabela de destino podem se tornar não sincronizados.

Definindo a Estratégia de Atualização para uma Sessão

Quando configura uma sessão, você tem várias opções para tratar operações de banco de dados, inclusive atualizações.

Especificando uma Operação para Todas as Linhas

Ao configurar uma sessão, você pode selecionar uma operação de banco de dados para todas as linhas, usando a configuração Tratar Linhas de Origem Como.

A tabela a seguir mostra as opções da configuração Tratar Linhas de Origem Como:

Configuração	Descrição
Inserir	Trate todas as linhas como inserções. Se a inserção da linha violar uma restrição de chave primária ou estrangeira no banco de dados, o Serviço de Integração rejeitará a linha.
Excluir	Trata todas as linhas como exclusões. Para cada linha, se o Serviço de Integração encontrar uma linha correspondente na tabela de destino (com base no valor da chave primária), o Serviço de Integração excluirá essa linha. Observe que a restrição de chave primária deve existir na definição de destino no repositório.
Atualizar	Trata todas as linhas como atualizações. Para cada linha, o Serviço de Integração procura um valor de chave primária correspondente na tabela de destino. Se ele existir, o Serviço de Integração atualizará a linha. É necessário que a restrição de chave primária exista na definição de destino.
Controlado por Dados	<p>O Serviço de Integração segue instruções codificadas nas transformações personalizadas e Estratégia de Atualização dentro do mapeamento da sessão, para determinar como sinalizar linhas para inserção, exclusão, atualização ou rejeição.</p> <p>Se o mapeamento da sessão contiver uma transformação Estratégia de Atualização, por padrão esse campo será marcado como Controlado por Dados.</p> <p>Se você não escolher Controlado por Dados quando um mapeamento contiver uma transformação personalizada ou Estratégia de Atualização, o Workflow Manager exibirá um aviso. Quando você executa a sessão, o Serviço de Integração não segue instruções na transformação personalizada ou na transformação Estratégia de Atualização no mapeamento para determinar como sinalizar linhas.</p>

A tabela a seguir descreve a estratégia de atualização para cada configuração:

Configuração	Use para
Inserir	Preencher as tabelas de destino na primeira vez ou manter um Data Warehouse histórico. No último caso, você deve definir essa estratégia para todo o Data Warehouse, e não apenas para um grupo selecionado de tabelas de destino.
Excluir	Apagar tabelas de destino.
Atualizar	Atualizar tabelas de destino. Você pode escolher essa configuração se o Data Warehouse contiver dados históricos ou um instantâneo. Por último, ao configurar o modo de atualização das tabelas de destino, você pode definir se deseja inserir linhas atualizadas como novas linhas ou usar as informações atualizadas para modificar linhas existentes no destino.
Controlado por Dados	Exerça um controle mais detalhado sobre o modo de sinalização das linhas para inserção, exclusão, atualização ou rejeição. Escolha essa configuração se as linhas destinadas à mesma tabela precisarem ser sinalizadas na ocasião para uma operação (por exemplo, atualização) ou para uma operação diferente (por exemplo, rejeição). Além disso, essa configuração fornece a única maneira de sinalizar linhas para rejeição.

Especificando Operações para Tabelas de Destino Individuais

Após determinar como tratar todas as linhas na sessão, você também precisará definir opções da estratégia de atualização para destinos individuais. Defina as opções da estratégia de atualização no modo de exibição Transformações na guia Mapeamento das propriedades da sessão.

Você pode definir as seguintes opções de estratégia de atualização:

- **Inserir.** Selecione esta opção para inserir uma linha em uma tabela de destino.

- **Excluir.** Selecione esta opção para excluir uma linha de uma tabela.
- **Atualizar.** Nessa situação, você tem as seguintes opções:
 - **Atualizar como Atualizar.** Atualizar cada linha sinalizada para atualização, caso ela exista na tabela de destino.
 - **Atualizar como Inserir.** Inserir cada linha sinalizada para atualização.
 - **Update Else Insert.** Atualizar a linha se ela existir. Caso contrário, insira-a.
- **Truncar tabela.** Selecione esta opção para truncar a tabela de destino antes de carregar os dados.

Lista de Verificação da Estratégia de Atualização

Para selecionar uma estratégia de atualização é preciso configurar as opções certas na sessão e talvez adicionar transformações Estratégia de Atualização ao mapeamento. Esta seção resume o que é necessário para implementar versões diferentes de uma estratégia de atualização.

- Realize inserções apenas em tabelas de destino.
Quando configurar a sessão, selecione Inserir para a propriedade de sessão Tratar Linhas de Origem Como. Não deixe de selecionar a opção Insert para todas as instâncias de destino da sessão.
- Excluir todas as linhas em uma tabela de destino.
Ao configurar a sessão, selecione Excluir para a propriedade de sessão Tratar Linhas de Origem Como. Não deixe de selecionar a opção Excluir para todas as instâncias de destino da sessão.
- Realize atualizações apenas no conteúdo de tabelas de destino.
Ao configurar a sessão, selecione Atualizar para a propriedade de sessão Tratar Linhas de Origem Como. Ao configurar as opções de atualização para cada instância da tabela de destino, não deixe de selecionar a opção Atualizar para cada instância de destino.
- Realize várias operações de banco de dados em várias linhas destinadas à mesma tabela de destino.
Adicione uma transformação de Estratégia de Atualização ao mapeamento. Ao gravar a expressão da estratégia de atualização da transformação, use as funções DECODE e IIF para sinalizar linhas para diferentes operações (inserir, excluir, atualizar ou rejeitar). Ao configurar uma sessão que use esse mapeamento, selecione Controlado por Dados para a propriedade de sessão Tratar Linhas de Origem Como. Não deixe de selecionar Inserir, Excluir ou uma das opções Atualizar para cada instância da tabela de destino.
- Rejeitar dados.
Adicione uma transformação de Estratégia de Atualização ao mapeamento. Ao gravar a expressão da estratégia de atualização da transformação, use DECODE e IIF para especificar os critérios de rejeição da linha. Ao configurar uma sessão que use esse mapeamento, selecione Controlado por Dados para a propriedade de sessão Tratar Linhas de Origem Como.

CAPÍTULO 33

Transformações XML

Este capítulo inclui os seguintes tópicos:

- [Transformação de Qualificador de Origem XML, 521](#)
- [Transformação de Analisador de XML, 521](#)
- [Transformação Gerador de XML, 522](#)

Transformação de Qualificador de Origem XML

Você pode adicionar uma transformação Qualificador de Origem XML a um mapeamento arrastando uma definição de origem XML até o espaço de trabalho do Mapping Designer ou criando uma manualmente. Ao adicionar uma definição de origem XML a um mapeamento, você precisa conectá-la a uma transformação de Qualificador de Origem XML. A transformação de Qualificador de Origem XML define os elementos de dados que o Serviço de Integração lê ao executar uma sessão. Ela determina como o PowerCenter lê os dados de origem. A transformação de Qualificador de Origem XML é uma transformação ativa.

Uma transformação de Qualificador de Origem XML sempre tem uma porta de entrada ou saída para cada coluna na origem XML. Quando você cria uma transformação de Qualificador de Origem XML para uma definição de origem, o Designer vincula cada porta na definição de origem XML a uma porta na transformação de Qualificador de Origem XML. Não é possível remover ou editar nenhum dos links. Se você remover uma definição de origem XML de um mapeamento, o Designer também removerá a transformação de Qualificador de Origem XML correspondente. Você pode vincular uma definição de origem XML a uma transformação de Qualificador de Origem XML.

Você pode vincular portas de um grupo de Qualificador de Origem XML a portas de diferentes transformações para formar fluxos de dados separados. No entanto, não é possível vincular portas de mais de um grupo em uma transformação de Qualificador de Origem XML a portas na mesma transformação de destino.

Você pode editar algumas das propriedades e adicionar extensões de metadados a uma transformação de Qualificador de Origem XML.

Transformação de Analisador de XML

Use uma transformação Analisador XML para extrair XML de um pipeline. A transformação Analisador XML permite extrair dados XML de sistemas de mensagens, como Séries MQ ou TIBCO, e de outras origens, como arquivos ou bancos de dados. A funcionalidade da transformação Analisador XML é semelhante à da origem

XML, exceto que aquela analisa o XML no pipeline. Por exemplo, talvez você queira extrair dados XML de uma origem TIBCO e passá-los a destinos relacionais. A transformação de Analisador de XML é uma transformação ativa.

A transformação Analisador XML lê dados XML de uma única porta de entrada e os grava em uma ou mais portas de saída.

Transformação Gerador de XML

Use uma transformação de Gerador de XML para criar XML em uma pipeline. A transformação Gerador XML permite ler dados de sistemas de mensagens, como Séries MQ e TIBCO, ou de outras origens, como arquivos ou bancos de dados. A funcionalidade da transformação Gerador XML é semelhante à do destino XML, exceto que aquela gera o XML no pipeline. Por exemplo, talvez você queira extrair dados de origens relacionais e passar dados XML a destinos. A transformação de Gerador de XML é uma transformação ativa.

A transformação Gerador XML aceita dados de várias portas e grava XML através de uma única porta de saída.

ÍNDICE

A

- address.dic
 - mascarando dados [157](#)
- adicionando
 - comentários sobre as expressões [33](#)
 - grupos [385](#)
- Adicionar Porta de Saída Estática (propriedade)
 - transformação SQL [455](#)
- alta precisão
 - ativação para a transformação Java [242](#)
- analisador
 - Data Transformation [504](#)
 - Saída de transformação de dados [513](#)
- Armazenamento em Cache Ativado (propriedade)
 - descrição [303](#)
- armazenando em cache
 - Linhas mestres na transformação de Associador [289](#)
- arquivo rejeitado
 - estratégias de atualização [516](#)
- arquivos com prefixo TX
 - procedimentos externos [180](#)
- arquivos de log
 - Consulte logs da sessão [arquivos de log aaa] [47](#)
- arquivos simples
 - associando dados [278](#)
 - pesquisas [296](#)
- arquivos simples classificados
 - usando na transformação Joiner [285](#)
- ASCII
 - transformação personalizada [62](#)
 - Transformação Procedimento Externo [167](#)
- assistentes
 - Assistente de Configuração do Aplicativo Visual Basic [186](#)
 - ATL COM AppWizard [171](#)
 - MFC AppWizard [183](#)
- associação de parâmetros
 - consultas de transformação de SQL [437](#)
- associação externa
 - Consulte também o tipo de associação[outer join aab] [426](#)
 - criando [426](#)
 - criando como uma substituição de associação [426](#)
 - criando como uma substituição de extração [427](#)
 - tipos suportados pelo Serviço de Integração [420](#)
- associação externa completa
 - definição [283](#)
- Associação externa detalhada
 - descrição [283](#)
- associação externa direita
 - criando [425](#)
 - sintaxe [425](#)
- associação externa esquerda
 - criando [423](#)
 - sintaxe [423](#)

- associação externa mestre
 - descrição [283](#)
 - preservando limites da transação [291](#)
- associação normal
 - criando [422](#)
 - definição [282](#)
 - preservando limites da transação [291](#)
 - sintaxe [422](#)
- associação padrão
 - Qualificador de Origem [416](#)
- associações
 - criando relações de chaves para [418](#)
 - definida pelo usuário [420](#)
 - padrão para Qualificador de Origem [416](#)
 - personalizar [417](#)
 - sintaxe do Informatica [421](#)
- associações heterogêneas
 - Consulte transformação Joiner[associações heterogêneas aab] [417](#)
- associações personalizadas
 - Consulte associações definidas pelo usuário [associações personalizadas aab] [420](#)
- Ativar Streaming de Entrada XML
 - propriedade da sessão do Analisador XML [513](#)
- atributo de nível
 - transformação de Normalizador de pipeline [369](#)
 - transformação de Normalizador VSAM [365](#)
- atributo de tipo de chave
 - transformação Normalizador [365](#)
- Atributo Occurs
 - transformação Normalizador [360](#)
- atributos de porta
 - editando [66](#)
 - visão geral [66](#)
- Atualizar como Atualização (propriedade)
 - descrição [519](#)
- Atualizar Como Inserção (propriedade)
 - descrição [519](#)
- Atualizar ou Inserir (propriedade)
 - descrição [347](#)
- AutoCommit
 - guia Configurações SQL [455](#)

B

- bancos de dados
 - associando dados de diferentes [278](#)
 - Consulte também fornecedores de bancos de dados específicos, como a Oracle [databases aab] [488](#)
 - opções suportadas [488](#)
- bancos de dados relacionais
 - associando [278](#)

- bibliotecas
 - para procedimentos externos C++ [173](#)
 - para procedimentos externos do VB [177](#)
 - para procedimentos externos Informatica [183](#)
- bibliotecas vinculadas dinâmicas
 - Consulte DLLs[bibliotecas vinculadas dinâmicas aaa] [183](#)
- bloqueando
 - linhas de detalhes na transformação Joiner [289](#)
- bloqueando dados
 - Funções de Transformação Personalizada [115](#)
 - Transformação Joiner [289](#)
 - transformação personalizada [71](#)
- bloqueando transformações
 - descrição [31](#)
- Botão Validar
 - transformações [46](#)

C

- C/C++
 - Consulte também Visual C++[C/C++ aab] [189](#)
 - vinculando ao Serviço de Integração [189](#)
- cache de pesquisa dinâmica
 - descrição [338](#)
 - limite de erros [349](#)
 - rejeitar o carregamento [349](#)
 - sincronizando com o destino [349](#)
 - substituição SQL de pesquisa [343](#)
 - usando origens de arquivo simples [338](#)
- Cache de Pesquisa Dinâmica (propriedade)
 - descrição [303](#)
- cache de pesquisa estático
 - visão geral [330](#)
- cache de pesquisa persistente
 - arquivos com nome [330](#)
 - com e sem nome [328](#)
 - compartilhando [330](#)
 - executar o cache do banco de dados novamente [326](#)
 - visão geral [328](#)
- Cache de pesquisa persistente (propriedade)
 - descrição [303](#)
- cache de pesquisa persistente nomeado
 - compartilhando [333](#)
 - visão geral [330](#)
- Cache de Pesquisa Pré-Criado
 - propriedade de pesquisa [303](#)
- cache do joiner
 - Transformação de Associador [289](#)
- cache nomeado
 - compartilhando [333](#)
 - executar o cache do banco de dados novamente [328](#)
 - persistente [328](#)
- cache sem nome
 - compartilhando [330](#)
 - executar o cache do banco de dados novamente [328](#)
 - persistente [328](#)
- caches
 - cache de pesquisa dinâmica [338](#)
 - cache de pesquisa estático [330](#)
 - compartilhando pesquisa [330](#)
 - pesquisa persistente nomeada [330](#)
 - sequencial [326](#), [327](#)
 - simultâneo [326](#), [327](#)
 - Transformação de Associador [289](#)
 - transformação de Pesquisa [324](#)

- caches de pesquisa
 - ativação do cache [303](#)
 - cache pré-criação [303](#)
 - caches persistentes nomeados [330](#)
 - compartilhando [330](#)
 - compartilhando pesquisas não nomeadas [331](#)
 - definição [324](#)
 - dinâmico [338](#)
 - dinâmico, limite de erros [349](#)
 - dinâmico, sincronizando com a origem de pesquisa [353](#)
 - dinâmico, sincronizando com o destino [349](#)
 - estático [330](#)
 - executar o cache do banco de dados novamente [326](#)
 - particionando orientações com caches não nomeados [331](#)
 - persistente [328](#)
 - Propriedade Cache de pesquisa persistente [303](#)
 - Propriedade Tamanho do Cache de Dados de Pesquisa [303](#)
 - rejeitar o carregamento [349](#)
 - substituição de ORDER BY [310](#)
 - tratamento de primeiros e últimos erros [315](#)
 - visão geral [324](#)
- caches sequenciais
 - Consulte caches [326](#)
- caches simultâneos
 - Consulte caches [326](#)
- Cadastros de Pessoas Físicas
 - máscara de código de área [150](#)
 - maskamento de dados repetíveis [151](#)
- cálculos
 - agregado [51](#)
 - usando variáveis com [37](#)
 - uso da transformação da Expressão [163](#)
- caracteres da string de origem
 - transformação de Mascaramento de Dados [144](#)
- caracteres de substituição da string de resultados
 - transformação de Mascaramento de Dados [145](#)
- chamada de texto
 - procedimento armazenado, inserção [486](#)
- Chamada de texto (propriedade)
 - Transformação Procedimento Armazenado [481](#)
- chave de criptografia de armazenamento
 - transformação de Mascaramento de dados [155](#)
- chaves
 - criando com a transformação Gerador de Sequência [391](#)
 - criando para associações [418](#)
 - definições de origem [418](#)
- chaves de composição
 - criando com a transformação Gerador de Sequência [391](#)
- chaves externas
 - criando com a transformação Gerador de Sequência [391](#)
- chaves geradas
 - transformação Normalizador [361](#)
- chaves primárias
 - criando com a transformação Gerador de Sequência [391](#)
- ciclo
 - Propriedade da Transformação de Gerador de Sequência [396](#)
- Ciclo (propriedade)
 - Propriedade da Transformação de Gerador de Sequência [394](#)
- Classe EDataType
 - expressões Java [264](#)
- Classe JExpression
 - expressões Java [266](#), [267](#)
- Classe JExprParaMetadata
 - expressões Java [265](#)
- classes wrapper
 - para bibliotecas ou funções pré-existentis [189](#)
- classificação
 - grupos de dados [378](#)

- classificação ()
 - valores de string [376](#)
- CLASSPATH
 - Transformação Java, configurando [240](#)
- Classpath Java
 - propriedade da sessão [240](#)
- COBOL
 - transformação de Normalizador VSAM [362](#)
- código estático
 - Transformações Java [237](#)
- códigos de status
 - Transformação Procedimento Armazenado [473](#)
- coluna de dependentes
 - mascaramento de dados [139](#)
- comentários
 - adicionando a expressões [33](#)
- commit
 - método API da transformação Java [246](#)
- company_names.dic
 - mascarando dados [157](#)
- compartilhando
 - caches de pesquisa nomeados [333](#)
 - caches de pesquisa sem nome [330](#)
- compilando
 - DLLs nos sistemas Windows [183](#)
 - Procedimentos da transformação personalizada [82](#)
 - Transformações Java [242](#)
- componente streamer
 - Data Transformation [504](#)
- componente transformador
 - Data Transformation [504](#)
- condição de junção
 - definição [286](#)
 - usando portas de origem classificadas [284](#)
 - visão geral [281](#)
- condição de pesquisa
 - Configurar [314](#)
 - definição [302](#)
 - transformação de Mascaramento de dados [138](#)
 - visão geral [315](#)
- condição do filtro de grupo
 - transformação Roteador [383](#)
- condições
 - Transformação Filtro [202](#)
 - Transformação Joiner [281](#)
 - Transformação Pesquisa [314](#), [318](#)
 - transformação Roteador [383](#)
- conectando com bancos de dados
 - transformação SQL [442](#)
- conectividade
 - exemplos de string de conexão [443](#)
- conexão de banco de dados completa
 - passando para a transformação SQL [443](#)
- conexão de banco de dados estática
 - considerações sobre desempenho [445](#)
 - descrição [442](#)
- conexão lógica de banco de dados
 - considerações sobre desempenho [445](#)
 - passando para a transformação SQL [442](#)
- conexões
 - transformação SQL [442](#)
- conexões de banco de dados
 - transformação SQL [442](#)
- conexões dinâmicas
 - considerações sobre desempenho [445](#)
 - exemplo de transformação de SQL [465](#)
 - transformação SQL [443](#)
- configurações de conexão
 - transformação SQL [443](#)
- Configurações de UDT
 - Transformação de Dados Não Estruturados [506](#)
- configurando
 - portas [29](#)
- confirmação automática
 - configurando com a transformação SQL [447](#)
 - descrição [455](#)
- constante TC_COMMIT_AFTER
 - descrição [493](#)
- constante TC_COMMIT_BEFORE
 - descrição [493](#)
- constante TC_CONTINUE_TRANSACTION
 - descrição [493](#)
- constante TC_ROLLBACK_AFTER
 - descrição [493](#)
- constante TC_ROLLBACK_BEFORE
 - descrição [493](#)
- constantes
 - substituindo valores nulos por [41](#)
- consulta de pesquisa
 - consulta padrão [310](#)
 - descrição [310](#)
 - Limitação Sybase ORDER BY [310](#)
 - ORDER BY [310](#)
 - palavras reservadas [311](#)
 - substituindo [310](#)
- consulta padrão
 - exibindo [415](#)
 - métodos de substituição [416](#)
 - substituição usando o Qualificador de Origem [418](#)
 - visão geral [415](#)
- consulta parcial
 - substituindo [440](#)
- Consulta SQL
 - adição da consulta personalizada [418](#)
 - exemplo de atualização dinâmica [460](#)
 - exemplo de conexão dinâmica [465](#)
 - exibindo a consulta padrão [415](#)
 - substituindo a consulta padrão [416](#), [418](#)
- consultas
 - substituindo a pesquisa [310](#)
 - Transformação Pesquisa [310](#)
 - Transformação Qualificador de Fonte [415](#), [418](#)
- consultas SQL dinâmicas
 - exemplo de transformação de SQL ☺ [460](#)
 - transformação SQL [438](#)
- consultas SQL estáticas
 - configurando portas [438](#)
 - transformação SQL [437](#)
- contagem de erros
 - incrementando para transformações Java [250](#)
- Contagem Máx. de Linhas de Saída (propriedade)
 - transformação SQL [455](#)
- Continuar com Erro de SQL dentro da Linha (propriedade)
 - transformação SQL [455](#)
- controlado por dados
 - visão geral [518](#)
- Controle de Transação
 - exemplo [494](#)
 - expressão [493](#)
 - transformação [493](#)
 - transformação Java [233](#)
 - transformação personalizada [70](#)
 - transformação SQL [446](#)
 - visão geral [492](#)

criando

- Objeto de Dados de Sequência [400](#)
 - portas [29](#)
 - procedimentos externos COM [170](#)
 - Procedimentos externos Informatica [178](#)
 - transformação Agregador [59](#)
 - transformação Classificação [378](#)
 - transformação de Expressão [165](#)
 - Transformação de filtro [203](#)
 - transformação de Gerador de Sequência [403](#)
 - Transformação de Gerador de Sequência [402](#)
 - Transformação Estratégia de Atualização [516](#)
 - Transformação Joiner [292](#)
 - transformação personalizada [63](#), [74](#)
 - Transformação Procedimento Armazenado [478](#)
 - transformação reutilizável de instância não reutilizável [49](#)
 - transformação Roteador [386](#)
 - Transformação União [500](#)
 - transformações [28](#)
 - transformações reutilizáveis [48](#)
- criptografar o armazenamento
- transformação de Mascaramento de dados [155](#)

D

dados

- armazenamento temporário [37](#)
- associando [278](#)
- pré-classificação [58](#)
- rejeitando através da transformação de Estratégia de Atualização [520](#)
- selecionando diferente [430](#)

dados classificados

- unindo a partir de pipelines particionados [284](#)
- usando na transformação Joiner [284](#)

dados em tempo real

- com a transformação Joiner [290](#)
- processando com a transformação Joiner [291](#)

dados relacionais classificados

- usando na transformação Joiner [285](#)

Data Integration Service)

- modo de reinicialização [253](#)

Data Transformation

- local do repositório [504](#)
- visão geral [502](#)

Data Transformation não estruturado

- exportando para esquema XML [511](#)
- visão geral [502](#)

databases

- Consulte também o Guia de Configuração[bancos de dados aab] [488](#)

DB2

- Consulte IBM DB2 [443](#)

defineJExpression

- Método de API de transformação Java [265](#)

definição

- dependências de porta na transformação Personalizada [65](#)

definições de origem COBOL

- criando uma transformação de Normalizador [366](#)
- instrução OCCURS [362](#)

dependências

- portas em transformações Personalizadas [65](#)

dependências de porta

- transformação personalizada [65](#)

depuração

- procedimentos externos [189](#)
- transformação Java [243](#)

desempenho

- conexões estáticas com bancos de dados [445](#)
- conexões lógicas com bancos de dados [445](#)
- filtro de aprimoramento [204](#)
- transformação Agregador [57](#)
- Transformação Joiner [292](#)
- Transformação Pesquisa [322](#)
- usando variáveis para melhorar [36](#)

desenvolvendo

- procedimentos externos COM [170](#)
- Procedimentos externos Informatica [178](#)

Destino

- Transformações Pesquisa [303](#)
- Transformações Procedimento Armazenado [481](#)

destinos

- atualizando [515](#)

dicionário

- mascaramento de dados de substituição [136](#)
- mascaramento repetível de expressão [148](#)

dígito de início

- CPF [154](#)

Diretiva de Pesquisa em Várias Correspondências (propriedade)

- descrição [303](#)

diretório de cache

- transformação de Mascaramento de dados [155](#)

Diretório de Cache (propriedade)

- Transformação Joiner [279](#)

diretório de trabalho

- Transformação Classificador, especificando [408](#)

diretórios de cache

- configurando o nome do diretório do cache de pesquisa [303](#)

Distinção Entre Maiúsculas e Minúsculas (propriedade)

- Transformação Classificador [408](#)

distribuindo

- Procedimentos da transformação personalizada [63](#)
- procedimentos externos [186](#)

dividindo XML

- Transformação de Dados Não Estruturados [513](#)

DLLs (bibliotecas vinculadas dinâmicas)

- compilando procedimentos externos [183](#)

E

Editor de Expressão

- cores de sintaxe [34](#)
- usando com expressões Java [260](#)
- validando expressões usando [34](#)
- visão geral [33](#)

embaçando

- valores de data [147](#)
- valores numéricos [146](#)

Encaminhamento de Linhas Rejeitadas

- configurando [516](#)

opção [516](#)

entrada classificada

- pesquisas de arquivos simples [296](#)

Entrada Classificada (propriedade)

- transformação Agregador [57](#)
- Transformação Joiner [279](#)

Entradas Devem Bloquear (propriedade)

- transformação Java [232](#)

entrega exatamente um

- transformação SQL [447](#)

erros

- aumentando o limite em transformações Java [250](#)
- com cache de pesquisa dinâmica [349](#)
- manipulando [43](#)

- erros ()
 - validando no Editor de Expressão [34](#)
- erros de código de não usuário
 - em transformações Java [244](#)
- erros de código do usuário
 - Transformações Java [244](#)
- erros de compilação
 - Identificando a origem da transformação Java [243](#)
- escopo da transformação
 - definindo para uma transformação Joiner [290](#)
 - Escopo de transformação de linha com a transformação Joiner [290](#)
 - Escopo de transformação de Todas as Entradas com a transformação Joiner [290](#)
 - Escopo de transformação de transação com a transformação Joiner [290](#)
 - propriedades da transformação Joiner [279](#)
 - Transformação do Classificador [409](#)
 - transformação Java [233](#)
 - transformação personalizada [70](#)
- Escopo da Transformação (propriedade)
 - transformação Java [232](#)
- Escopo da transformação Linha
 - comportamento na transformação Joiner [290](#)
- escopo da transformação Transação
 - comportamento na transformação Joiner [290](#)
- Escopo de transformação de Todas as Entradas
 - comportamento na transformação Joiner [290](#)
- Está Ativo (propriedade)
 - transformação Java [232](#)
- estratégia de atualização
 - configurando com uma transformação Personalizada [69](#)
 - definindo com uma transformação Java [234](#)
 - funções de estratégia de linha (baseadas em linha) [119](#)
- exceções
 - de procedimentos externos [188](#)
- executar o cache do banco de dados novamente
 - cache nomeado [328](#)
 - cache sem nome [328](#)
 - visão geral [326](#)
- exemplo do BankSoft
 - Procedimento externo da Informatica [178](#)
 - visão geral [168](#)
- exportar para esquema XML
 - Data Transformation não estruturado [511](#)
- expressões
 - chamando pesquisas [319](#)
 - chamando procedimento armazenado de [483](#)
 - Condição de filtro [202](#)
 - Consulte também a Referência de Linguagem de Transformação do PowerCenter [54](#)
 - estratégia de atualização [517](#)
 - inserção [31](#), [33](#)
 - não agregadas [57](#)
 - regras de transformações de Procedimentos Armazenados [489](#)
 - simplificando [37](#)
 - transformação Agregador [54](#)
 - Transformações Java [258](#)
 - validando [34](#)
 - valores retornados [31](#)
- expressões Java
 - chamando com interface simples [262](#)
 - Classe EDataType [264](#)
 - Classe JExpression [266](#), [267](#)
 - Classe JExprParaMetadata [265](#)
 - configurando [259](#)
 - configurando funções [260](#)
 - criando [260](#)
 - criando na caixa de diálogo Definir Expressão [260](#)

- expressões Java ()
 - criando na caixa de diálogo Definir Função [260](#)
 - exemplo de interface avançada [266](#)
 - exemplo de interface simples [262](#)
 - gerando [259](#)
 - gerar código Java [260](#)
 - interface avançada [263](#)
 - interface simples [261](#)
 - invocando [251](#)
 - invocando com interface avançada [263](#)
 - método API invokeJExpression [262](#)
 - regras e diretrizes [262](#), [264](#)
 - regras e diretrizes para invocar [251](#)
 - tipos de função de expressão [259](#)
 - Transformações Java [258](#)
 - uso de funções definidas pelo usuário [259](#)
 - uso de funções do idioma da transformação [259](#)
 - uso de funções personalizadas [259](#)
- expressões não agregadas
 - visão geral [57](#)
- extensões de metadados
 - em transformações personalizadas [73](#)

F

- falha em mapeamentos
 - Transformações Java [248](#)
- falha nas sessões
 - Transformações Java [248](#)
- filtragem das linhas de origem
 - transformação de Pesquisa [313](#)
 - Transformação Pesquisa [313](#)
- filtrando linhas
 - Qualificador de origem como filtro [204](#)
 - transformação para [201](#), [406](#)
- filtro da origem de pesquisa
 - descrição [303](#)
 - limitação de pesquisas [313](#)
- filtros de origem
 - adicionando ao Qualificador de Origem [428](#)
- firstnames.dic
 - maskando dados [157](#)
- formato de máscara
 - maskamento de valores de string [143](#)
- função ABORT
 - usando [42](#)
- função de acesso das propriedades
 - descrição [195](#)
- função de modo de acesso aos dados
 - função de transformação personalizada [97](#)
- função de nível de rastreamento
 - descrição [199](#)
- função definir de página de código de dados
 - função de transformação personalizada [118](#)
- função distribuir
 - descrição [194](#)
- função ERROR
 - usando [42](#)
- função IIF
 - substituindo chaves ausentes com a transformação Gerador de Sequência [391](#)
- funções
 - agregação [54](#)
 - API da transformação Personalizada [96](#)
 - não agregadas [55](#)
- funções API
 - transformação personalizada [96](#)

- funções baseadas em linhas
 - estratégia de linha [119](#)
 - manipulação de dados [109](#)
- funções baseadas em matriz
 - definir linha de erro de entrada [127](#)
 - é válido para linhas [123](#)
 - estratégia de linha [126](#)
 - manipulação de dados [123](#)
 - número de linhas [122](#)
 - Número máximo de linhas [121](#)
 - visão geral [121](#)
- funções de acesso a parâmetros
 - 64-bit [196](#)
 - descrição [196](#)
- funções de agregação
 - Consulte também Referência sobre Linguagem de Transformação[aggregate functions
 - aaa] [55](#)
 - lista de [54](#)
 - usando em expressões [54](#)
 - valores nulos [55](#)
 - visão geral [54](#)
- funções de desinicialização
 - transformação personalizada [94](#)
- funções de estratégia de linha
 - baseada em linhas [119](#)
 - baseada em matriz [126](#)
- funções de inicialização
 - transformação personalizada [91](#)
- funções de manipulação de dados
 - baseada em linhas [109](#)
 - baseada em matriz [123](#)
- funções de não agregação
 - exemplo [55](#)
- funções de notificação
 - transformação personalizada [92](#)
- funções de propriedade
 - transformação personalizada [100](#)
- Funções de Transformação Personalizada
 - alterar estratégia da linha padrão [120](#)
 - alterar modo de string [117](#)
 - API [96](#)
 - baseada em matriz [121](#)
 - configurar o modo de acesso aos dados [97](#)
 - configurar porta de passagem [112](#)
 - definir linha de erro de entrada [127](#)
 - definir página de código de dados [118](#)
 - desinicialização [94](#)
 - é válido para linhas [123](#)
 - erro [113](#)
 - está terminado [115](#)
 - estratégia de linha (baseada em linha) [119](#)
 - estratégia de linha (baseada em matriz) [126](#)
 - geradas [90](#)
 - Incremento de contagem do erro [114](#)
 - inicialização [91](#)
 - log da sessão [114](#)
 - lógica de bloqueio [115](#)
 - manipulação de dados (com base em linha) [109](#)
 - manipulação de dados (com base em matriz) [123](#)
 - navegação [97](#)
 - notificação [92](#)
 - notificação de saída [112](#)
 - número de linhas [122](#)
 - Número máximo de linhas [121](#)
 - ponteiro [116](#)
 - propriedade [100](#)
 - religar tipos de dados [107](#)

- Funções de Transformação Personalizada ()
 - saída de limite de dados [113](#)
 - trabalhando com identificadores [85](#), [97](#)
- funções definidas pelo usuário
 - usando com expressões Java [259](#)
- funções geradas
 - transformação personalizada [90](#)
- funções personalizadas
 - usando com expressões Java [259](#)
- funções relacionadas a partições
 - descrição [199](#)

G

- geração de transações
 - transformação Java [234](#), [246](#)
 - transformação personalizada [70](#)
- gerar código Java
 - expressões Java [260](#)
- gerar linha de rollBack
 - transformação Java [254](#)
- gerar linha de saída
 - Transformações Java [248](#)
- gerar transação
 - transformação Java [233](#)
- Gerar Transação (propriedade)
 - transformação Java [232](#)
- gerenciamento de memória
 - para procedimentos externos [189](#)
- grupo definido pelo usuário
 - transformação Roteador [383](#)
- grupo por portas
 - expressão não agregada [57](#)
 - transformação Agregador [55](#)
 - usando valores padrão [57](#)
- grupos
 - adicionando [385](#)
 - definido pelo usuário [383](#)
 - Regras de transformação personalizada [65](#)
 - transformação de União [500](#)
 - transformação HTTP [208](#)
 - transformação Java [230](#)
 - transformação personalizada [64](#)
 - transformação Roteador [383](#)
- grupos padrão
 - transformação Roteador [383](#)
- Guia Auxiliares
 - Transformações Java [237](#)
- Guia Código Auxiliar
 - exemplo [274](#)
- Guia Código Java
 - usando [229](#)
- Guia Configurações SQL
 - transformação SQL [455](#)
- guia Hierarquia de Entrada
 - Transformação de Dados Não Estruturados [510](#)
- guia Hierarquia de Saída
 - Transformação de Dados Não Estruturados [510](#)
- Guia Importações
 - Transformações Java [236](#)
- Guia Importar Pacotes
 - exemplo [273](#)
 - Transformações Java [236](#)
- Guia Na Linha de Entrada
 - exemplo [274](#)
 - transformação Java [238](#)

- Guia No Final dos Dados
 - transformação Java [238](#)
- Guia Passiva
 - Transformações Java [237](#)
- guia Portas SQL
 - transformação SQL [456](#)
- Guia Propriedades da Máscara
 - transformação de Mascaramento de Dados [131](#)
- Guia Sobre a Recepção de Transações
 - transformação Java [239](#)

H

- hierarquias relacionais
 - Transformação de Dados Não Estruturados [510](#)

I

- IBM DB2
 - sintaxe da string de conexão [443](#)
- ID da página de código
 - transformação Personalizada, alterando [118](#)
- ID da sequência
 - Transformação Pesquisa [340](#)
- ID de coluna gerada
 - transformação Normalizador [358](#)
- Identificador Programático (propriedade)
 - Transformação Procedimento Externo [168](#)
- identificadores
 - transformação personalizada [85](#)
- identificadores entre aspas
 - palavras reservadas [415](#)
- IDM_Dictionary
 - conexão de mascaramento de dados [139](#)
- IDM_Storage
 - conexão de mascaramento de dados [139](#)
- IDs de propriedade
 - transformação personalizada [101](#)
- Ignorar em Comparação (propriedade)
 - descrição [342](#)
- Ignorar Nulo (propriedade)
 - seleção [341](#)
- incrementando
 - configuração do intervalo da sequência [396](#)
- Incremento De (propriedade)
 - Propriedade da Transformação de Gerador de Sequência [394](#)
- Informações de Conexão (propriedade)
 - Transformação Pesquisa [303](#)
 - Transformação Procedimento Armazenado [481](#)
- Informações do Dicionário
 - transformação de Mascaramento de dados [138](#)
- Informix
 - notas sobre procedimentos armazenados [477](#)
- inicializando
 - Procedimentos da transformação personalizada [73](#)
 - procedimentos externos [191](#)
 - suporte a variável do Serviço de Integração para [193](#)
 - variáveis [39](#)
- InputType
 - Transformação de Dados Não Estruturados [506](#)
- inserção
 - expressões [31](#), [33](#)
 - filtros de origem [428](#)
 - junções definidas pelo usuário [420](#)
 - substituição da consulta de SQL [418](#)

- Inserir ou Atualizar (propriedade)
 - descrição [347](#)
- instâncias
 - criando transformações reutilizáveis [48](#)
- instrução OCCURS
 - exemplo COBOL [362](#)
- instruções SQL
 - compatível com transformação SQL [457](#)
- interface avançada
 - chamando expressões Java [263](#)
 - Classe EDataType [264](#)
 - Classe JExpression [266](#), [267](#)
 - Classe JExprParaMetadata [265](#)
 - exemplo [266](#)
 - expressões Java [263](#)
- interface do IDispatch
 - definindo uma classe [170](#)
- interface simples
 - exemplo [262](#)
 - expressões Java [261](#)
 - Métodos API de transformação Java [261](#)
- intervalo
 - mascaramento de valores numéricos [146](#)
- intervalo de confirmação de armazenamento
 - transformação de Mascaramento de dados [155](#)
- invoke
 - Método de API de transformação Java [269](#)
- invokeJExpression
 - método API [262](#)

J

- janela código completo
 - erros de compilação Java [243](#)
- JDK
 - transformação Java [226](#)
- JRE
 - transformação Java [226](#)
- junções definidas pelo usuário
 - inserção [420](#)

L

- limite da transação
 - descartando na transformação Joiner [291](#)
 - preservando na transformação Joiner [291](#)
 - transformação personalizada [70](#)
- Linguagem (propriedade)
 - transformação Java [232](#)
- linguagem de transformação
 - funções de agregação [54](#)
 - usando com expressões Java [259](#)
- Linha de Saída
 - definindo o tipo de linha na transformação Java [255](#)
- linhas
 - excluindo [520](#)
 - signalizando para atualização [516](#)
- linhas de detalhe
 - bloqueando na transformação Joiner [289](#)
 - processando em transformação Joiner não classificada [289](#)
 - processando na transformação de Associador classificada [289](#)
- linhas de entrada
 - obtendo o tipo de linha [249](#)
- linhas de erro
 - transformação SQL [451](#)

- linhas de saída distintas
 - transformação Classificador [409](#)
- linhas mestres
 - armazenando em cache [289](#)
 - processando em transformação Joiner não classificada [289](#)
 - processando na transformação de Associador classificada [289](#)
- links da web
 - adicionando a expressões [33](#)
- Local do Tempo de Execução (propriedade)
 - Transformação Procedimento Externo [168](#)
- LogicalConnectionObject
 - descrição [442](#)
 - exemplo de transformação de SQL [468](#)
- logs
 - Transformações Java [252](#), [253](#)
- logs de sessão
 - Transformações Java [253](#)

M

- manter ordem das linhas
 - transformação de Gerador de Sequência [400](#)
- mapeador
 - Data Transformation [504](#)
- mapeamentos
 - adicionando transformações [28](#)
 - adicionando transformações reutilizáveis [50](#)
 - adicionando uma origem COBOL [362](#)
 - afetado por procedimentos armazenados [473](#)
 - componentes da pesquisa [301](#)
 - configurando transformação Procedimento Armazenado [482](#)
 - configurando transformação Procedimento Armazenado não conectada [483](#)
 - modificando transformações reutilizáveis [50](#)
 - sinalizando linhas para atualização [516](#)
 - usando transformações Roteador [386](#)
 - usando uma transformação Procedimento Externo [175](#)
- Mapping Designer
 - adicionando transformação reutilizável [50](#)
- maskamento aleatório
 - maskamento de valores de data [142](#)
 - maskamento de valores de string [142](#)
 - valores numéricos [141](#)
- maskamento de chave
 - descrição [133](#)
 - maskamento de valores de string [134](#)
 - maskamento de valores numéricos [135](#)
 - maskando valores de data e hora [135](#)
 - valores numéricos [133](#)
- maskamento de dados
 - mapeando com a transformação de Expressão [161](#)
 - mapeando com a transformação Pesquisa [157](#)
- maskamento de expressão
 - descrição [147](#)
 - exemplo de maskamento repetido [148](#)
 - maskamento repetível [148](#)
 - regras e diretrizes [149](#)
- maskamento de formato especial
 - Cadastros de Pessoas Físicas [150](#)
 - CPF [154](#)
 - Endereços de e-mail [152](#)
 - Endereços IP [154](#)
 - números de telefone [152](#)
 - números SIN repetíveis [154](#)
 - URLs [155](#)
- maskamento de substituição
 - configurando dicionários relacionais [138](#)
- maskamento de substituição ()
 - descrição [136](#)
 - propriedades de maskamento [137](#)
 - regras e diretrizes de maskamento de dados [139](#)
- maskamento dependente
 - descrição [139](#)
 - maskamento repetível [141](#)
- maskamento dependente repetível
 - descrição [141](#)
- maskando dados
 - dados de pesquisa de amostra [157](#)
 - endereços de amostra [157](#)
 - nomes da amostra [157](#)
 - nomes de empresa de amostra [157](#)
 - sobrenomes da amostra [157](#)
- mensagens de erro
 - para procedimentos externos [189](#)
 - rastreando procedimentos externos [189](#)
- Método API de transformação Java
 - getResultDataType [268](#)
- Método defineJExpression
 - Transformações Java [247](#)
- Método failSession
 - Transformações Java [248](#)
- Método generateRow
 - Transformações Java [248](#)
- Método GetBytes
 - Transformações Java [268](#)
- Método GetDouble
 - Transformações Java [268](#)
- Método getInRowType
 - Transformações Java [249](#)
- Método getInt
 - Transformações Java [268](#)
- Método getLong
 - Transformações Java [268](#)
- Método getMetada
 - transformações de Java [249](#)
- Método getResultDataType
 - Transformações Java [268](#)
- Método getResultMetadata
 - Transformações Java [269](#)
- Método getStringBuffer
 - Transformações Java [269](#)
- Método incrementErrorCount
 - Transformações Java [250](#)
- método invokeJExpression
 - Transformações Java [251](#)
- Método isNull
 - Transformações Java [252](#)
- método isResultNull
 - Transformações Java [270](#)
- Método LogError
 - Transformações Java [252](#)
- Método logInfo
 - Transformações Java [253](#)
- Método resetNotification
 - Transformações Java [253](#)
- Método setNull
 - Transformações Java [255](#)
- método storeMetada
 - transformações de Java [256](#)
- métodos
 - API da transformação de Java [245](#)
 - transformação Java [238](#), [239](#)
- métodos API
 - transformação Java [245](#)
 - Transformações Java [245](#)

Métodos de API de transformação Java

- commit [246](#)
- defineJExpression [265](#)
- getBytes [268](#)
- getDouble [268](#)
- getInt [268](#)
- getLong [268](#)
- getResultMetadata [269](#)
- getStringBuffer [269](#)
- invoke [269](#)
- isResultNull [270](#)
- rollBack [254](#)
- setOutRowType [255](#)

métodos definidos pelo usuário

- transformação Java [238](#), [239](#)
- Transformações Java [237](#)

MFC AppWizard

- visão geral [183](#)

Microsoft SQL Server

- notas sobre procedimentos armazenados [477](#)
- sintaxe da string de conexão [443](#)

Modo ASCII

- configurando a ordem de classificação para a transformação Joiner [284](#)

modo de consulta

- regras e diretrizes [441](#)
- transformação SQL [436](#)

modo de script

- regras e diretrizes [436](#)
- transformação SQL [435](#)

modo passivo

- transformação SQL [442](#)

Modo Unicode

- configurando a ordem de classificação para a transformação Joiner [284](#)

- Consulte também Guia do Administrador [modo Unicode
aaa] [62](#)

- transformação personalizada [62](#)

- Transformação Procedimento Externo [167](#)

Módulo (propriedade)

- Transformação Procedimento Externo [168](#)

N

NaN

- convertendo para 1.#QNAN [432](#)

níveis de rastreamento

- Conciso [47](#)
- Dados Detalhados [47](#)
- Inicialização Detalhada [47](#)
- Normal [47](#)
- Propriedade da transformação Classificador [409](#)
- propriedade da transformação Java [232](#)
- Propriedade da transformação Procedimento Armazenado [481](#)
- Propriedade da Transformação Procedimento Externo [168](#)
- propriedades da sessão [52](#)
- propriedades da transformação Joiner [279](#)
- substituindo [47](#)
- visão geral [47](#)

Níveis de rastreamento concisos

- defined [47](#)

Níveis de rastreamento normais

- visão geral [47](#)

Nível de Rastreamento de Dados Detalhados

- transformação SQL [454](#)
- visão geral [47](#)

Nível de rastreamento de Inicialização Detalhada

- visão geral [47](#)

nível de rastreamento do status

- Transformação de Dados Não Estruturados [507](#)

Nome de Classe (propriedade)

- transformação Java [232](#)

nome do arquivo de cache

- especificação para o cache de pesquisa persistente [303](#)

Nome do diretório do cache de pesquisa (propriedade)

- descrição [303](#)

Nome do Serviço Dinâmico

- Transformação de Dados Não Estruturados [506](#)

número de valores guardados em cache

- Propriedade da Transformação de Gerador de Sequência [398](#)

- Valor da propriedade de Gerador de Sequência [394](#)

Número Máximo de Conexões no Pool (propriedade)

- transformação SQL [455](#)

números SIN

- mascaramento de dados repetíveis [154](#)

- mascarando CPFs [154](#)

O

Objeto de Dados de Sequência

- criando [400](#)

- propriedades [400](#)

objetos de conexão

- configuração nas transformações Pesquisa [303](#)

- configurando em transformações Procedimento Armazenado [481](#)

opção de localidade de scripts

- transformação SQL [455](#)

opção Substituição SQL de Pesquisa

- descrição [303](#)

- mapeando parâmetros e variáveis [310](#)

- reduzindo o tamanho do cache [310](#)

opções avançadas

- transformação SQL [444](#)

operações específicas do thread

- gravando [69](#)

- transformações HTTP [207](#)

- transformações Personalizadas [67](#)

operadores

- condição de pesquisa [315](#)

operators

- Consulte também Referência sobre Linguagem de
Transformação[operadores

- aab] [315](#)

Oracle

- notas sobre procedimentos armazenados [477](#)

- sintaxe da string de conexão [443](#)

ordem de carregamento

- Qualificador de Origem [412](#)

ordem de carregamento de destino

- Qualificador de Origem [412](#)

ordem de classificação

- configuração para a transformação Joiner [285](#)

- transformação Agregador [58](#)

- Transformação Qualificador de Fonte [429](#)

- Ordem de Classificação Mestre (propriedade)

- Transformação Joiner [279](#)

Ordem de Execução (propriedade)

- Transformação Procedimento Armazenado [481](#)

Ordenação dos Nulos em Detalhe (propriedade)

- Transformação Joiner [279](#)

Ordenação dos Nulos no Mestre (propriedade)

- Transformação Joiner [279](#)

- ORDER BY
 - consulta de pesquisa [310](#)
 - substituindo [310](#)
 - substituir [310](#)
- \$Origem
 - Transformações Pesquisa [303](#)
 - Transformações Procedimento Armazenado [481](#)
- origem de classificação
 - configurando a condição de associação para uso [284](#)
 - definição [284](#)
- origens
 - associando [278](#)
 - associando múltiplos [278](#)
 - mesclando [499](#)
 - unindo dados a partir da mesma origem [287](#)
- OutputType
 - Transformação de Dados Não Estruturados [506](#)

P

- Pacotes Java
 - importando [236](#)
- páginas de código
 - Consulte também o Guia do Administrador do PowerCenter[code pages aab] [62](#)
 - funções de acesso [198](#)
 - transformação personalizada [62](#)
 - Transformação Procedimento Externo [167](#)
- palavras reservadas
 - consulta de pesquisa [311](#)
 - gerando SQL com [415](#)
 - resword.txt [415](#)
- Parâmetros de entrada
 - procedimentos armazenados [472](#)
- parâmetros de mapeamento
 - em transformações do Qualificador de Origem [413](#)
 - na substituição SQL da pesquisa [310](#)
- parâmetros de mascaramento
 - mascaramento de dados [132](#)
- parâmetros de saída
 - procedimentos armazenados [472](#)
 - transformação de Procedimento Armazenado não conectada [483](#)
- particionamento de pipeline
 - transformação HTTP [207](#)
 - transformação personalizada [67](#)
- pesquisa no pipeline
 - configuração das propriedades da sessão [309](#)
 - descrição [294](#)
 - exemplo de mapeamento [297](#)
 - exemplo de transformação Pesquisa [297](#)
 - mapeamento [297](#)
 - propriedades da transformação Pesquisa [297](#)
 - transformação não reutilizável [322](#)
 - transformação reutilizável [322](#)
- pesquisas conectadas
 - criando [321](#)
 - descrição [298](#)
 - visão geral [299](#)
- pesquisas de arquivos simples
 - descrição [296](#)
 - entrada classificada [296](#)
- pesquisas não conectadas
 - adicionando condições de pesquisa [318](#)
 - chamando através de expressões [319](#)
 - descrição [298](#)
 - designando valores de retorno [319](#)
 - pesquisas não conectadas ()
 - visão geral [300, 318](#)
- pipelines
 - mesclando com a transformação de União [499](#)
- pipelines particionados
 - unindo dados classificados [284](#)
- porta CURRVAL
 - Transformação de Gerador de Sequência [392](#)
- porta de retorno
 - Transformação Pesquisa [301, 319](#)
- Porta InputBuffer
 - Transformação de Dados Não Estruturados [508](#)
- Porta NEXTVAL
 - Gerador de Sequência [388](#)
- Porta NumRowsAffected
 - transformação SQL [450](#)
- Porta OutputBuffer
 - Transformação de Dados Não Estruturados [508](#)
- Porta OutputFileName
 - Transformação de Dados Não Estruturados [508](#)
- Porta ScriptError
 - descrição [435](#)
- Porta ScriptName
 - transformação SQL [435](#)
- Porta ScriptResults
 - transformação SQL [435](#)
- portas
 - Agrupar por [55](#)
 - classificadas [57, 429](#)
 - configurando [29](#)
 - configurar [30](#)
 - criando [29, 30](#)
 - opção de portas classificadas [429](#)
 - ordem de avaliação [38](#)
 - portas variáveis [37](#)
 - Qualificador de Origem [429](#)
 - transformação Agregador [53](#)
 - transformação Classificação [377](#)
 - Transformação de Dados Não Estruturados [509](#)
 - Transformação de Gerador de Sequência [388](#)
 - transformação de União [500](#)
 - transformação HTTP [208](#)
 - transformação Java [230](#)
 - transformação personalizada [64](#)
 - Transformação Pesquisa [301](#)
 - transformação Roteador [385](#)
 - visão geral de valores padrão [39](#)
- portas associadas
 - ID da sequência [340](#)
 - Transformação Pesquisa [340](#)
- portas classificadas
 - dados de pré-classificação [58](#)
 - motivos para não usar [58](#)
 - ordem de classificação [429](#)
 - Qualificador de Origem [429](#)
 - requisitos de cache [53](#)
 - transformação Agregador [57](#)
- portas de entrada
 - Transformações Java [231](#)
 - usando como variantes [238](#)
 - valores padrão [39](#)
 - visão geral [30](#)
- portas de entrada/saída
 - visão geral [30](#)
- portas de passagem
 - adicionando à transformação SQL [440](#)
 - valores padrão [39](#)

- portas de pesquisa
 - descrição [301](#)
- portas de saída
 - exigido para a transformação Expressão [164](#)
 - Transformações Java [231](#)
 - tratamento de erros [39](#)
 - usando como variantes [238](#)
 - valores padrão [39](#)
 - visão geral [30](#)
- portas variáveis
 - visão geral [37](#)
- pós-sessão
 - erros [488](#)
 - procedimentos armazenados [486](#)
- pré-sessão
 - erros [487](#)
 - procedimentos armazenados [486](#)
- precisão de subsegundos
 - Transformação Pesquisa [303](#)
 - Transformação Procedimento Armazenado [481](#)
- preenchimento de portas
 - Transformação de Dados Não Estruturados [510](#)
- prefixo do nome do arquivo de cache
 - visão geral [330](#)
- Prefixo do Nome do Arquivo de Cache (propriedade)
 - descrição [303](#)
- procedimentos armazenados
 - alterando parâmetros [482](#)
 - bancos de dados suportados [488](#)
 - criação de sessões para a execução pré ou pós-sessão [486](#)
 - definição [471](#)
 - definindo tipo de [481](#)
 - erros de sessão [488](#)
 - erros pós-sessão [488](#)
 - erros pré-sessão [487](#)
 - Exemplo da Informix [477](#)
 - exemplo de Microsoft [477](#)
 - exemplo de Sybase [477](#)
 - exemplo de Teradata [478](#)
 - Exemplo do IBM DB2 [478](#)
 - Exemplo do Oracle [477](#)
 - gravando [476](#)
 - importando [479](#)
 - observações de sintaxe específicas de bancos de dados [477](#)
 - ordem de processamento, especificando [474](#)
 - registrando em variáveis [38](#)
 - tipos de carregamento [486](#)
 - tratamento de erros [487](#)
- Procedimentos da transformação personalizada
 - criando [74](#)
 - específico do thread [67](#)
 - exemplo [77](#)
 - gerando arquivos de código [76](#)
 - trabalhando com linhas [89](#)
- procedimentos externos
 - depuração [189](#)
 - distribuindo [186](#)
 - distribuindo procedimentos externos da Informatica [187](#)
 - funções de interface [194](#)
 - observações de desenvolvimento [187](#)
- procedimentos externos COM
 - adicionando ao repositório [174](#)
 - criando [170](#)
 - criando um destino [174](#)
 - criando uma origem [174](#)
 - depuração [189](#)
 - desenvolvendo em Visual C++ [171](#), [174](#)
 - desenvolvimento no Visual Basic [176](#)

- procedimentos externos COM ()
 - distribuindo [186](#)
 - em comparação aos procedimentos externos da Informatica [168](#)
 - gerenciamento de memória [189](#)
 - inicializando [191](#)
 - manipulação de exceção [188](#)
 - não conectadas [191](#)
 - observações de desenvolvimento [187](#)
 - procedimentos no nível de linha [188](#)
 - registrando com repositórios [174](#)
 - tipo de servidor [170](#)
 - tipos de dados [187](#)
 - valores retornados [188](#)
 - visão geral [170](#)
- Procedimentos externos Informatica
 - depuração [189](#)
 - desenvolvendo [178](#)
 - distribuindo [187](#)
 - em comparação à COM [168](#)
 - gerando código C++ [180](#)
 - gerenciamento de memória [189](#)
 - inicializando [191](#)
 - manipulação de exceção [188](#)
 - não conectadas [191](#)
 - observações de desenvolvimento [187](#)
 - procedimentos no nível de linha [188](#)
 - valores retornados [188](#)
- promovendo
 - transformações não reutilizáveis [49](#)
- propriedades de pesquisa
 - configurando em uma sessão [308](#)

R

- rastreamento de mensagens
 - para procedimentos externos [189](#)
- reassociar funções de tipo de dados
 - Funções de Transformação Personalizada [107](#)
- redefinir
 - Transformação de Gerador de Sequência [399](#)
- registrando
 - Procedimentos COM com repositórios [174](#)
- regras
 - valores padrão [46](#)
- regras de mascaramento
 - aplicação [143](#)
 - caracteres da string de origem [144](#)
 - caracteres de substituição da string de resultados [145](#)
 - embaçando [146](#)
 - formato de máscara [143](#)
 - intervalo [146](#)
- reinicializando o cache de pesquisa
 - Consulte reenviar para cache a partir de banco de dados [326](#)
- repositórios
 - procedimentos externos COM [174](#)
 - registrando procedimentos COM com [174](#)
- Requer Thread Único (propriedade)
 - transformação personalizada [67](#)
- Requer Thread Único por Partição (propriedade)
 - transformação HTTP [207](#)
 - transformação Java [232](#)
- resiliência
 - Bancos de dados de transformação SQL [448](#)
- resiliência do banco de dados
 - transformação SQL [447](#)
- retornar várias linhas
 - Transformação Pesquisa [317](#)

Reutilizável

Transformação de Gerador de Sequência [400](#)

rollback

método API da transformação Java [254](#)

roteando linhas

transformação para [381](#)

S

Saída é Determinística (propriedade)

transformação Java [232](#)

Transformação Procedimento Armazenado [481](#)

Transformação Procedimento Externo [168](#)

Saída é Ordenada (propriedade)

transformação Java [232](#)

Saída é Repetível (propriedade)

Transformação Procedimento Armazenado [481](#)

Transformação Procedimento Externo [168](#)

Saída exclusiva

transformação de Mascaramento de dados [138](#)

Saída Repetível

transformação de Mascaramento de Dados [132](#)

selecionar distintos

opção Qualificador de Origem [430](#)

substituindo em sessões [430](#)

serializador

Data Transformation [504](#)

Serviço de Integração

agregação de dados [55](#)

em execução no modo de depuração [189](#)

erro na manipulação de procedimentos armazenados [487](#)

limite da transação [70](#)

suporte a variável [193](#)

tipos de dados [187](#)

servidores COM

tipo para procedimentos externos COM [170](#)

sessões

\$\$\$SessStartTime [413](#)

agregação incremental [51](#)

configuração para manipular erros de procedimento armazenado [487](#)

definindo a estratégia de atualização [518](#)

pré e pós-procedimentos armazenados, execução [486](#)

substituindo selecionar distintos [430](#)

Transformação Procedimento Armazenado [473](#)

Transformação Procedimento Externo [175](#)

setOutRowType

método API da transformação Java [255](#)

sintaxe

criando associações externas direitas [425](#)

criando associações normais [422](#)

criando junções externas esquerdas [423](#)

restrições de banco de dados comuns [427](#)

sintaxe de associação

associação externa direita [425](#)

associação externa esquerda [423](#)

associação normal [422](#)

Sistemas Windows

compilação de DLLs em [183](#)

solucionando problemas

transformação Agregador [60](#)

transformação Java [243](#)

transformação Normalizador [374](#)

Transformação Procedimento Armazenado [490](#)

Transformação Qualificador de Fonte [432](#)

Source Analyzer

criando relações de chaves [418](#)

SQL

adição da consulta personalizada [418](#)

exibindo a consulta padrão [415](#)

substituindo a consulta padrão [416](#), [418](#)

SQL de pré- e pós-sessão

Transformação Qualificador de Fonte [431](#)

string de conexão

sintaxe [443](#)

strings

classificação [376](#)

subssegundos

processando em transformações Java [242](#)

substituição da string

consultas de transformação de SQL [439](#)

substituição de associação

sintaxe da associação externa esquerda [423](#)

sintaxe de associação externa direita [425](#)

sintaxe de associação normal [422](#)

substituição SQL

consulta padrão [416](#)

substituição SQL de pesquisa

caches dinâmicos [343](#)

substituindo

consulta de SQL padrão do Qualificador de Origem [418](#)

surnames.dic

mascarando dados [157](#)

Sybase ASE

limitação de ORDER BY [310](#)

notas sobre procedimentos armazenados [477](#)

sintaxe da string de conexão [443](#)

T

tabela de armazenamento

mascaramento de dados de substituição [137](#)

mascaramento de expressão [148](#)

tabela de armazenamento compartilhado

transformação de Mascaramento de dados [155](#)

tabela de pesquisa

índices [301](#), [322](#)

nome [303](#)

tabelas

criando relações de chaves [418](#)

tabelas de destino

definindo a estratégia de atualização de [519](#)

excluindo linhas [520](#)

inserções [520](#)

Tamanho da parte do streamer

Transformação de Dados Não Estruturados [506](#)

tamanho do cache

transformação de Mascaramento de dados [155](#)

Tamanho do cache de dados de Joiner

propriedades da transformação Joiner [279](#)

Tamanho do cache de índice de Joiner

propriedades da transformação Joiner [279](#)

Teradata

sintaxe da string de conexão [443](#)

threads

transformação personalizada [69](#)

tipo de associação

associação externa completa [283](#)

Associação externa detalhada [283](#)

associação externa direita [420](#)

associação externa esquerda [420](#)

associação externa mestre [283](#)

Transformação Qualificador de Fonte [420](#)

- Tipo de associação
 - associação normal [282](#)
 - Propriedades do Joiner [281](#)
- Tipo de associação (propriedade)
 - Transformação Joiner [279](#)
- tipo de dados BigDecimal
 - transformação Java [242](#)
- tipo de entrada do arquivo
 - Transformação de Dados Não Estruturados [508](#)
- tipo de parâmetro TINFParam
 - definição [190](#)
- Tipo de Procedimento Armazenado (propriedade)
 - Transformação Procedimento Armazenado [481](#)
- tipo de saída de arquivo
 - Transformação de Dados Não Estruturados [512](#)
- tipos de carregamento
 - procedimentos armazenados [486](#)
- tipos de dados
 - COM [187](#)
 - Qualificador de Origem [412](#)
 - transformação [187](#)
 - Transformações Java [227](#)
- tipos de dados duplos
 - transformação Java [242](#)
- tipos de dados nativos
 - transformação SQL [438](#)
- tipos de dados primitivos Java
 - transformações Java [227](#)
- transação
 - definição [493](#)
 - gerando [70](#), [234](#), [246](#)
 - trabalhando com na transformação Joiner [290](#)
- transformação Agregador
 - agregação aninhada [54](#)
 - Combinação de Estratégia de Atualização [517](#)
 - comparado com a transformação Expressão [51](#)
 - componentes [52](#)
 - criando [59](#)
 - exemplo de função não agregada [55](#)
 - Função STDDEV (desvio padrão) [54](#)
 - função VARIANCE [54](#)
 - grupo por portas [55](#)
 - lista de funções [54](#)
 - níveis de rastreamento [52](#)
 - otimizando o desempenho [59](#)
 - portas [53](#)
 - portas classificadas [57](#)
 - solucionando problemas [60](#)
 - usando com a transformação Joiner [286](#)
 - usando variáveis [36](#)
 - valores nulos [55](#)
 - visão geral [51](#)
- transformação Classificação
 - criando [378](#)
 - definindo grupos para [378](#)
 - opções [376](#)
 - porta RANKINDEX [377](#)
 - portas [377](#)
 - usando variáveis [36](#)
 - visão geral [375](#)
- transformação Classificador
 - criando [410](#)
 - usando com a transformação Joiner [285](#)
- Transformação Controle de Transação
 - criando [497](#)
 - eficaz [495](#)
 - em mapeamentos [495](#)
 - ineficaz [495](#)
- Transformação Controle de Transação ()
 - mapeando a validação [497](#)
 - propriedades [493](#)
 - visão geral [493](#)
- transformação Controle de Transação efetiva
 - definição [495](#)
- transformação Controle de Transação sem efeito
 - definição [495](#)
- transformação de Analisador de XML
 - ativando a entrada dividida [513](#)
 - visão geral [521](#)
- Transformação de Associador
 - caches [289](#)
- Transformação de Classificador
 - configurando o Tamanho do Cache do Classificador [408](#)
- Transformação de Dados Não Estruturados
 - componentes [505](#)
 - configurando [514](#)
 - definindo o nível de rastreamento do status [507](#)
 - dividindo saídas em XML [513](#)
 - exemplo de tipo de saída de arquivo [512](#)
 - gravando em destinos relacionais [510](#), [512](#)
 - gravando em um arquivo XML [513](#)
 - guia Configurações de UDT [506](#)
 - liberando dados de entrada [510](#)
 - Porta InputBuffer [508](#)
 - Porta OutputBuffer [508](#)
 - Porta OutputFileName [508](#)
 - portas [509](#)
 - preenchimento de portas a partir da Transformação de Dados [510](#)
- Transformação de Estratégia de Atualização
 - Combinação do Agregador [517](#)
- transformação de Expressão
 - criando [165](#)
 - roteando dados [165](#)
 - visão geral [163](#)
- Transformação de filtro
 - criando [203](#)
 - dicas de desempenho [204](#)
 - dicas para desenvolvimento [204](#)
- Transformação de Filtro
 - exemplo [201](#)
 - visão geral [201](#)
- transformação de Gerador de Sequência
 - criando [403](#)
 - manter ordem das linhas [400](#)
 - Propriedade Incrementar em [396](#)
- Transformação de Gerador de Sequência
 - Ambiente não nativo [405](#)
 - ciclo [396](#)
 - Ciclo [397](#)
 - criando [402](#)
 - criando chaves compostas [391](#)
 - intervalo de valores [397](#)
 - Mecanismo Blaze [405](#)
 - Mecanismo Spark [405](#)
 - não reutilizável [398](#)
 - número de valores guardados em cache [398](#)
 - porta CURRVAL [392](#)
 - Porta NEXTVAL [388](#)
 - portas [388](#)
 - Propriedade Incrementar em [396](#)
 - propriedades [394](#), [400](#)
 - redefinir [399](#)
 - reutilizável [399](#)
 - usando a função IIF para substituir chaves ausentes [391](#)
 - Valor Atual [397](#)
 - valor inicial [396](#)

Transformação de Gerador de Sequência ()
 visão geral [387](#)
 transformação de HTTP
 códigos de resposta [205](#)
 transformação de mascaramento de dados
 mascarando os endereços IP [154](#)
 transformação de Mascaramento de dados
 diretório de cache [155](#)
 intervalo de confirmação de armazenamento [155](#)
 mascaramento de dados dependente [139](#)
 mascaramento dependente repetível [141](#)
 propriedades de mascaramento de substituição [138](#)
 propriedades de sessão [155](#)
 Saída exclusiva [155](#)
 tabela de armazenamento compartilhado [155](#)
 tamanho do cache [155](#)
 transformação de Mascaramento de dados [155](#)
 transformação de Mascaramento de Dados
 arquivo de valor padrão [155](#)
 caracteres da string de origem [144](#)
 configurando dicionários relacionais [138](#)
 CPF repetível [151](#)
 dicionário para mascaramento da substituição [136](#)
 diretrizes para mascaramento de expressão [149](#)
 embaçando [146](#)
 formato de máscara [143](#)
 intervalo [146](#)
 mascaramento aleatório [141](#)
 mascaramento de expressão [147](#)
 mascaramento de expressão do nome do dicionário [148](#)
 mascaramento de substituição [136](#)
 mascaramento de valores de data [146](#)
 mascaramento repetível de expressão [148](#)
 mascarando CPFs [154](#)
 mascarando números de previdência social [150](#)
 mascarando números de telefone [152](#)
 mascarando o endereço de e-mail [152](#)
 mascarando URLs [155](#)
 números SIN repetíveis [154](#)
 propriedades de mascaramento [131](#)
 propriedades de mascaramento de substituição [137](#)
 regras e diretrizes [156](#)
 tabela de armazenamento [148](#)
 tabelas de armazenamento [137](#)
 usando parâmetros de mapeamento [132](#)
 transformação de Normalizador de pipeline
 criando [369](#)
 descrição [366](#)
 Guia Normalizador [368](#)
 Guia Portas. [368](#)
 transformação de Normalizador VSAM
 criando [366](#)
 descrição [362](#)
 Guia Normalizador [365](#)
 Guia Portas. [364](#)
 transformação de Pesquisa
 cache dinâmico [338](#)
 conectada [299](#)
 descrição de pesquisa em pipeline [294](#)
 não conectada [300](#)
 origens de pesquisa [294](#)
 pesquisas de arquivos simples [294](#)
 sincronizando cache dinâmico com o destino [349](#)
 visão geral [294](#)
 Transformação de pesquisa
 conectada [298](#)
 não conectada [298](#)
 transformação de União
 componentes [500](#)
 diretrizes [499](#)
 grupos [500](#)
 portas [500](#)
 visão geral [499](#)
 Transformação do Classificador
 configurando [408](#)
 diretório de trabalho [408](#)
 propriedades [408](#)
 visão geral [406](#)
 Transformação do Gerador XML
 visão geral [522](#)
 Transformação Estratégia de Atualização
 Combinação de pesquisa [518](#)
 criando [516](#)
 definindo opções para sessões [518](#), [519](#)
 encaminhando linhas rejeitadas [516](#)
 etapas para configurar [516](#)
 inserindo expressões [517](#)
 lista de verificação [520](#)
 visão geral [515](#)
 Transformação Estratégia de Atualização (propriedade)
 transformação Java [232](#)
 transformação Expressão
 usando variáveis [36](#)
 Transformação Filtro
 condição [202](#)
 transformação HTTP
 autenticação [206](#)
 código específico do thread [207](#)
 configuração da guia HTTP [208](#)
 configuração de propriedades [208](#)
 configurando grupos e portas [208](#)
 criando [206](#)
 É Particionável (propriedade) [207](#)
 exemplos [215](#)
 grupos [208](#)
 Requer a propriedade Thread Único por Partição [207](#)
 URL de Base [213](#)
 transformação Java
 analisando um arquivo simples [239](#)
 classpath de nível de sessão [240](#)
 classpath do nível de transformação [240](#)
 configurando CLASSPATH [240](#)
 Controle de Transação [233](#)
 criação do código Java [234](#)
 criando grupos [230](#)
 criando portas [230](#)
 definindo a estratégia de atualização [234](#)
 definindo o tipo de linha de saída [255](#)
 depuração [243](#)
 É Particionável (propriedade) [232](#)
 Entradas Devem Bloquear (propriedade) [232](#)
 erros de compilação [243](#)
 Escopo da Transformação (propriedade) [232](#), [233](#)
 Estratégia de Atualização (propriedade) [234](#)
 exemplo [271](#)
 Gerar Transação (propriedade) [234](#)
 Guia Código Java [229](#)
 Guia Na Linha de Entrada [238](#)
 Guia No Final dos Dados [238](#)
 Guia Sobre a Recepção de Transações [239](#)
 Linguagem (propriedade) [232](#)
 localizando erros [243](#)
 métodos API [245](#)
 Nível de rastreamento (propriedade) [232](#)
 Nome de Classe (propriedade) [232](#)

transformação Java ()

- processando subssegundos [242](#)
- Propriedade Gerar Transação [232](#)
- Propriedade Saída é Determinística [232](#)
- Propriedade Saída é Ordenada [232](#)
- propriedades [232](#)
- Requer a propriedade Thread Único por Partição [232](#)
- visão geral [226](#)

transformação Joiner

- configuração da ordem de classificação [285](#)
- usando com a transformação Classificador [285](#)

Transformação Joiner

- associando vários bancos de dados [278](#)
- bloqueio dos dados de origem [289](#)
- comportamento com o escopo de transformação de Linha [290](#)
- comportamento com o escopo de transformação de Todas as Entradas [290](#)
- comportamento com o escopo de transformação Transação [290](#)
- condições [281](#)
- configurando a condição de associação para usar portas de origem de classificação [284](#)
- configurando a ordem de classificação no modo ASCII [284](#)
- configurando a ordem de classificação no modo Unicode [284](#)
- criando [292](#)
- dados em tempo real [290](#)
- descarte dos limites de transação [291](#)
- dicas de desempenho [292](#)
- escopo da transformação [290](#)
- Escopo da transformação Linha [290](#)
- escopo da transformação Transação [290](#)
- Escopo de transformação de Todas as Entradas [290](#)
- preservando limites de transação [291](#)
- processando dados em tempo real [291](#)
- propriedades [279](#)
- regras de entrada [278](#)
- tipos de associação [281](#)
- transações [290](#)
- unindo dados a partir da mesma origem [287](#)
- visão geral [278](#)

transformação Joiner não classificada

- processando as linhas de detalhe [289](#)
- processando linhas mestre [289](#)

Transformação Mascaramento de Dados

- requisitos de conexão [139](#)

transformação Normalizador

- atributo de nível [365](#), [369](#)
- Atributo de Tipo de Chave [365](#)
- Atributo Occurs [360](#)
- chave gerada [361](#)
- criando uma transformação de Normalizador de pipeline [369](#)
- criando uma transformação de Normalizador VSAM [366](#)
- exemplo [370](#)
- exemplo de mapeamento [370](#)
- Guia Normalizador [360](#)
- guia Portas do Normalizador da pipeline [368](#)
- Guia Portas. [358](#)
- Guia Propriedades [359](#)
- ID de coluna gerada [358](#)
- Normalizador de VSAM [362](#)
- pipeline Normalizador [366](#)
- solucionando problemas [374](#)
- visão geral [357](#)

transformação personalizada

- As entradas podem bloquear a propriedade [71](#)
- atributos de porta [66](#)
- bloqueando dados [71](#)
- código específico do thread [67](#)
- compilando procedimentos [82](#)

transformação personalizada ()

- componentes [64](#)
 - construindo o módulo [82](#)
 - controle de transação [70](#)
 - criação de procedimentos [74](#)
 - criando [63](#), [74](#)
 - criando grupos [64](#)
 - criando portas [64](#)
 - definição de relações de porta [65](#)
 - definindo a estratégia de atualização [69](#)
 - distribuindo [63](#)
 - É Particionável (propriedade) [67](#)
 - extensões de metadados [73](#)
 - funções [85](#)
 - gerando arquivos de código [63](#), [76](#)
 - IDs de propriedade [101](#)
 - limite da transação [70](#)
 - páginas de código [62](#)
 - passando linhas ao procedimento [89](#)
 - Propriedade Escopo da Transformação [70](#)
 - propriedade Estratégia de Atualização [69](#)
 - Propriedade Gerar Transação [70](#)
 - Propriedade Requer Thread Único [67](#)
 - propriedades [67](#)
 - propriedades de inicialização [73](#)
 - propriedades de procedimento [73](#)
 - regras e diretrizes [64](#)
 - threads [69](#)
 - visão geral [61](#)
- #### Transformação Pesquisa
- cache persistente [328](#)
 - cache persistente com nome [330](#)
 - caches [324](#)
 - Combinação de Estratégia de Atualização [518](#)
 - compartilhamento de cache [330](#)
 - componentes de [301](#)
 - condição [314](#), [318](#)
 - configuração das propriedades da sessão de pesquisa do pipeline [309](#)
 - consulta padrão [310](#)
 - Consulte também o Guia de Fluxo de Trabalho Avançado do PowerCenter[transformação de Pesquisa aab] [316](#)
 - criação da pesquisa conectada [321](#)
 - dicas de desempenho [322](#)
 - executar o cache do banco de dados novamente [326](#)
 - exemplo de pesquisa de pipeline [297](#)
 - expressões [319](#)
 - ID da sequência [340](#)
 - ignorar em comparação [342](#)
 - ignorar nulo [341](#)
 - inserindo consultas personalizadas [312](#)
 - limite de erros [349](#)
 - mapeando parâmetros e variáveis [310](#)
 - não conectadas [318](#)
 - pesquisas de arquivos simples [296](#)
 - pipeline não reutilizável [322](#)
 - pipeline reutilizável [322](#)
 - porta de entrada [301](#)
 - porta de entrada associada [340](#)
 - porta de pesquisa [301](#)
 - porta de saída [301](#)
 - portas [301](#)
 - precisão de subsegundos para portas de data e hora [303](#)
 - propriedades [303](#)
 - rejeitar o carregamento [349](#)
 - retornando várias linhas [317](#)
 - sincronizando o cache dinâmico com a origem de pesquisa [353](#)

Transformação Pesquisa ()

- sincronizando o cache dinâmico com uma origem de pesquisa [353](#)
- substituindo a consulta padrão [310](#)
- usando pipeline em mapeamento [297](#)
- valores retornados [319](#)
- várias correspondências [315](#)

transformação Pesquisa Não conectada

- porta de retorno [319](#)
- portas de entrada [318](#)

transformação Procedimento Armazenado

- conectadas [473](#)
- não conectadas [473](#)

Transformação Procedimento Armazenado

- Chamada de texto (propriedade) [481](#)
- códigos de status [473](#)
- configuração do procedimento armazenado conectado [482](#)
- configurando [475](#)
- configurando procedimento armazenado não conectado [483](#)
- criando ao importar [478](#)
- criando manualmente [480](#), [481](#)
- dados de entrada [472](#)
- dados de saída [472](#)
- definindo opções [481](#)
- dicas de desempenho [490](#)
- especificando quando executar [474](#)
- execução da pré ou pós-sessão [486](#)
- importação de procedimentos armazenados [479](#)
- Informações de Conexão (propriedade) [481](#)
- modificando [482](#)
- não conectadas [473](#), [483](#)
- Nível de rastreamento (propriedade) [481](#)
- Ordem de Execução (propriedade) [481](#)
- parâmetros de entrada/saída [472](#)
- pré e pós-sessão [486](#)
- precisão de subsegundos para portas de data e hora [481](#)
- propriedades [481](#)
- regras de expressão [489](#)
- Saída é Determinística (propriedade) [481](#)
- Saída é Repetível (propriedade) [481](#)
- solucionando problemas [490](#)
- tempo de execução da sessão, especificando [474](#)
- Tipo de Procedimento Armazenado (propriedade) [481](#)
- valores retornados [472](#)
- visão geral [471](#)

Transformação Procedimento Externo

- 64 bits[transformação de Procedimento Externo sessenta e quatro] [196](#)
- classes wrapper [189](#)
- código multithread [166](#)
- Consulte também procedimentos externos COM[transformação Procedimento Externo aab] [178](#)
- Consulte também Procedimentos externos Informatica [transformação Procedimento Externo aai] [178](#)
- criando bibliotecas para procedimentos externos C++ [173](#)
- criando bibliotecas para procedimentos externos da Informatica [183](#)
- criando bibliotecas para procedimentos externos do Visual Basic [177](#)
- criando no Designer [178](#)
- depuração [189](#)
- descrição [167](#)
- É Particionável (propriedade) [168](#)
- exemplo do BankSoft [168](#)
- função de acesso a página de códigos [198](#)
- função de acesso das propriedades [195](#)
- função de nível de rastreamento [199](#)

Transformação Procedimento Externo ()

- função de procedimento externo [195](#)
- função distribuir [194](#)
- funções de acesso a parâmetros [196](#)
- funções de interface [194](#)
- funções relacionadas a partições [199](#)
- gerenciamento de memória [189](#)
- Identificador Programático (propriedade) [168](#)
- inicializando [191](#)
- interface do IDispatch [170](#)
- Local do Tempo de Execução (propriedade) [168](#)
- manipulação de exceção [188](#)
- MFC AppWizard [183](#)
- Módulo (propriedade) [168](#)
- não conectadas [191](#)
- Nível de rastreamento (propriedade) [168](#)
- objetos ATL [171](#)
- observações de desenvolvimento [187](#)
- Procedimento externo Informatica usando o exemplo do BankSoft [178](#)
- procedimento no nível de linha [188](#)
- procedimentos externos COM [170](#)
- Procedimentos externos Informatica [178](#)
- propriedades [167](#), [168](#)
- Saída é Determinística (propriedade) [168](#)
- Saída é Repetível (propriedade) [168](#)
- sessão [175](#)
- suporte a variável de processo [193](#)
- Tipos de COM vs. Informatica [168](#)
- tipos de dados COM [187](#)
- Transformação Procedimento externo arquivos necessários [193](#)
- usando em um mapeamento [175](#)
- valores retornados [188](#)
- variáveis de membro [196](#)
- visão geral [166](#)
- Visual Basic [176](#)
- Visual C++ [171](#)

Transformação Qualificador de Fonte

- \$\$\$SessStartTime [413](#)
- associação dos dados de origem [416](#)
- associação padrão [416](#)
- associações [418](#)
- associações personalizadas [417](#)
- como origem de pesquisa [297](#)
- configurando [431](#)
- consulta padrão [415](#)
- criando relações de chaves [418](#)
- exibindo a consulta padrão [415](#)
- inserindo a origem do filtro [428](#)
- inserindo associação definida pelo usuário [420](#)
- mapeando parâmetros e variáveis [413](#)
- Opção Número de Portas Classificadas [429](#)
- opção Selecionar Distintos [430](#)
- ordem de carregamento de destino [412](#)
- ordem de classificação com Agregador [58](#)
- propriedades [432](#)
- Qualificador de Origem XML [521](#)
- solucionando problemas [432](#)
- SQL de pré- e pós-sessão [431](#)
- substituição SQL [418](#)
- substituindo a consulta padrão [416](#), [418](#)
- suporte à associação externa [420](#)
- tipos de dados [412](#)
- visão geral [411](#)
- transformação Roteador
- condição do filtro de grupo [383](#)
- conexão em mapeamentos [386](#)
- criando [386](#)

- transformação Roteador ()
 - exemplo [383](#)
 - filtrando dados do Normalizador [370](#)
 - grupos [383](#)
 - portas [385](#)
 - visão geral [381](#)
- transformação SQL
 - coluna de tipo de dados nativo [438](#)
 - configurando conexões [442](#)
 - Consultas SELECT [438](#)
 - consultas SQL dinâmicas [438](#)
 - consultas SQL estáticas [437](#)
 - Controle de Transação [446](#)
 - definindo atributos SQL [455](#)
 - definindo Dados Detalhados [454](#)
 - descrição [434](#)
 - Descrição de portas SQL [456](#)
 - Diretrizes de recuperação de HA [447](#)
 - entrega de mensagem exatamente um [447](#)
 - exemplo de conexão dinâmica [465](#)
 - exemplo de consulta dinâmica [460](#)
 - instruções SQL compatíveis [457](#)
 - modo de consulta [436](#)
 - modo de script [435](#)
 - modo passivo [441](#), [442](#)
 - NumRowsAffected [450](#)
 - opções avançadas [444](#)
 - passando informações de conexão completas [443](#)
 - Porta ScriptError [435](#)
 - Porta ScriptName [435](#)
 - Porta ScriptResults [435](#)
 - portas de consulta estática [438](#)
 - portas de passagem [440](#)
 - propriedades [453](#)
 - resiliência de banco de dados [448](#)
 - resiliência do banco de dados [447](#)
 - Tabela PM_REC_STATE
 - Recuperação da transformação da SQL [447](#)
 - usando a substituição de string [439](#)
- Transformação União
 - criando [500](#)
- transformações
 - adicionando a mapeamentos [28](#)
 - Agregador [51](#)
 - Analizador XML [521](#)
 - ativa e passiva [25](#)
 - Classificação [375](#)
 - conectadas [25](#)
 - Consulte também transformações ativas[transformations aab] [25](#)
 - criando [28](#)
 - definição [24](#)
 - descrições [26](#)
 - Estratégia de Atualização [515](#)
 - Expressão [163](#)
 - Filtro [201](#)
 - Gerador de Sequência [387](#)
 - Gerador XML [522](#)
 - Java [226](#)
 - Joiner [278](#)
 - manipulação de erros [43](#)
 - não conectadas [25](#)
 - nativas e não nativas [25](#)
 - níveis de rastreamento [47](#)
 - Normalizador [357](#)
 - Personalizar [61](#)
 - Pesquisa [294](#)
 - Procedimento armazenado [471](#)
- transformações ()
 - promovendo para reutilizável [49](#)
 - Qualificador de Origem [411](#)
 - Qualificador de Origem XML [521](#)
 - Roteador [381](#)
 - SQL [434](#)
 - tipos [24](#)
 - tipos que permitem para expressões [31](#)
 - tornando reutilizável [49](#)
 - transformações reutilizáveis [47](#)
 - União [499](#)
 - vários grupos [31](#)
 - visão geral [24](#)
- transformações ativas
 - Agregador [51](#)
 - Analizador XML [521](#)
 - Classificação [375](#)
 - Classificador [406](#)
 - Consulte também transformações[active transformations aab] [25](#)
 - Controle de transação [493](#)
 - Estratégia de Atualização [515](#)
 - Filtro [201](#)
 - Gerador XML [522](#)
 - Java [226](#), [227](#)
 - Joiner [278](#)
 - Normalizador [357](#)
 - Personalizar [61](#)
 - Qualificador de Origem [411](#), [434](#)
 - Qualificador de Origem XML [521](#)
 - Roteador [381](#)
 - União [499](#)
 - visão geral [25](#)
- transformações conectadas
 - Agregador [51](#)
 - Analizador XML [521](#)
 - Classificação [375](#)
 - Estratégia de Atualização [515](#)
 - Expressão [163](#)
 - Filtro [201](#)
 - Gerador de Sequência [387](#)
 - Gerador XML [522](#)
 - Java [226](#)
 - Joiner [278](#)
 - Normalizador [357](#)
 - Personalizar [61](#)
 - Pesquisa [294](#)
 - Procedimento armazenado [471](#)
 - Qualificador de Origem [411](#)
 - Qualificador de Origem XML [521](#)
 - Roteador [381](#)
 - SQL [434](#)
 - transformação de União [499](#)
- Transformações de Agregador
 - exemplo de cláusula condicional [55](#)
- transformações de Java
 - armazenamento de metadados [256](#)
 - Método getMetadata [249](#)
 - método storeMetadata [256](#)
 - recuperação de metadados [249](#)
- transformações desconectadas
 - Pesquisa [294](#)
 - Transformação Pesquisa [318](#)
 - Transformação Procedimento Armazenado [471](#)
 - Transformação Procedimento Externo [191](#)
- transformações Java
 - obtendo o tipo de linha de entrada [249](#)

Transformações Java

- ativa [227](#)
- compilando [242](#)
- configurando valores nulos em [255](#)
- conversão de tipo de dados [227](#)
- criando grupos [230](#)
- criando portas [230](#)
- criando trechos de código Java [235](#)
- erros de código de não usuário [244](#)
- erros de código do usuário [244](#)
- falha em mapeamentos [248](#)
- falha nas sessões em [248](#)
- Guia Auxiliares [237](#)
- Guia Importações [236](#)
- Guia Importar Pacote [236](#)
- Guia Passiva [237](#)
- Identificando a origem de erros de compilação [243](#)
- logs [252](#), [253](#)
- logs de sessão [253](#)
- Método defineJExpression [247](#)
- Método failSession [248](#)
- Método generateRow [248](#)
- Método getInRowType [249](#)
- Método incrementErrorCount [250](#)
- Método invokeJExpression [251](#)
- Método isNull [252](#)
- Método LogError [252](#)
- Método LogInfo [253](#)
- Método resetNotification [253](#)
- Método setNull [255](#)
- métodos API [245](#)
- passiva [227](#)
- portas de entrada [231](#)
- portas de saída [231](#)
- redefinindo variáveis em [253](#)
- tipos de dados primitivos Java [227](#)
- valores padrão para portas [231](#)
- verificando valores nulos em [252](#)
- transformações não conectadas
 - transformação de Pesquisa [300](#)
- transformações não nativas
 - visão geral [25](#)
- transformações nativas
 - visão geral [25](#)
- transformações passivas
 - configurando a transformação SQL [441](#)
 - Expressão [163](#)
 - Gerador de Sequência [387](#)
 - Java [226](#), [227](#)
 - Pesquisa [294](#)
 - Procedimento armazenado [471](#)
 - visão geral [25](#)
- transformações reutilizáveis
 - adicionando a mapeamentos [50](#)
 - alterando [50](#)
 - criando [48](#)
 - criando uma instância não reutilizável [49](#)
 - variáveis de mapeamento [48](#)
 - visão geral [47](#)
- Transformações XML
 - Analizador XML [521](#)
 - Gerador XML [522](#)
 - Qualificador de Origem [521](#)
- Transformation Developer
 - transformações reutilizáveis [47](#)
- tratamento de erros
 - com cache de pesquisa dinâmica [349](#)
 - para procedimentos armazenados [487](#)

Tratar Linhas de Origem Como

- estratégia de atualização [518](#)
- trechos de código
 - criando para transformações Java [235](#)
- trechos de código Java
 - criando para transformações Java [235](#)
 - exemplo [273](#)
- Troca de Transformação (TX)
 - definição [166](#)

U

- unindo dados classificados
 - configuração para otimizar o desempenho da associação [285](#)
 - usando arquivos simples classificados [285](#)
 - usando dados relacionais classificados [285](#)
 - usando transformações do Classificador [285](#)
- Update Else Insert (propriedade)
 - descrição [519](#)
- URL
 - adição por meio de links de documentos de negócios [33](#)
- URL de Base
 - configurando [213](#)
- Usar Pool de Conexão (propriedade)
 - transformação SQL [455](#)

V

- validando
 - expressões [34](#)
 - valores padrão [46](#)
- Valor Antigo de Saída na Atualização (propriedade)
 - descrição [303](#)
- valor atual
 - Transformação de Gerador de Sequência [397](#)
- Valor Atual (propriedade)
 - Transformação de Gerador de Sequência [394](#)
- Valor Final (propriedade)
 - Transformação de Gerador de Sequência [394](#)
- valor inicial
 - Transformação de Gerador de Sequência [396](#)
- Valor Inicial
 - transformação de Mascaramento de Dados [132](#)
- Valor Inicial (propriedade)
 - Transformação de Gerador de Sequência [394](#)
- valores
 - cálculo com a transformação Expressão [164](#)
- valores de data
 - mascaramento de dados aleatórios [142](#)
- Valores de Data e Hora
 - mascaramento de dados [135](#)
 - Transformação Qualificador de Fonte [412](#)
- valores de porta
 - Transformações Java [231](#)
- valores de string
 - mascaramento de dados chave [134](#)
 - mascaramento de dados personalizado [142](#)
- valores não encontrados
 - substituindo com o Gerador de Sequência [391](#)
- valores nulos
 - configurando para transformação Java [255](#)
 - filtro [203](#)
 - funções de agregação [55](#)
 - ignorando [43](#)
 - substituindo o uso de funções de agregação [57](#)
 - substituindo por uma constante [41](#)

- valores nulos ()
 - Transformação Classificador [409](#)
 - verificando transformações Java [252](#)
- valores numéricos
 - maskamento aleatório [141](#)
 - maskamento de chave [135](#)
- valores padrão
 - definido pelo usuário [40](#)
 - Grupo de Agregadores por portas [57](#)
 - inserção [46](#)
 - maskamento de dados [155](#)
 - portas de entrada [39, 40](#)
 - portas de passagem [39, 40](#)
 - portas de saída [39, 40](#)
 - regras para [46](#)
 - validando [46](#)
 - visão geral [39](#)
- valores retornados
 - de procedimentos externos [188](#)
 - transformação de Pesquisa [319](#)
 - Transformação Procedimento Armazenado [472](#)
- várias correspondências
 - propriedade da política de pesquisa [303](#)
 - Transformação Pesquisa [315](#)
- variáveis
 - Inicializações [39](#)
 - ordem de avaliação da porta [38](#)
 - resultados de procedimento de armazenamento, captura [38](#)
 - transformação Java [238, 239](#)
 - Transformações Java [237](#)
 - visão geral [36, 37](#)
- variáveis de ambiente
 - configuração para os pacotes Java [240](#)
- variáveis de conexão
 - usando transformações de Procedimento Armazenado [481](#)
 - uso nas transformações Pesquisa [303](#)
- variáveis de instância
 - Transformações Java [237](#)
- variáveis de mapeamento
 - em transformações do Qualificador de Origem [413](#)

- variáveis de mapeamento ()
 - na substituição SQL da pesquisa [310](#)
 - transformações reutilizáveis [48](#)
- variáveis de processo
 - nas propriedades de inicialização [193](#)
- variáveis estáticas
 - Transformações Java [237](#)
- variáveis locais
 - visão geral [36](#)
- variável de instância
 - transformação Java [238, 239](#)
- vários grupos
 - transformações [31](#)
- Vertica
 - sintaxe da string de conexão [443](#)
- Visual Basic
 - adicionando funções ao Serviço de Integração [189](#)
 - Assistente de Configuração de Aplicativo [186](#)
 - classes wrapper para [189](#)
 - código de procedimentos externos [167](#)
 - desenvolvendo procedimentos externos COM [176](#)
 - distribuindo procedimentos manualmente [186](#)
 - tipos de dados COM [187](#)
- Visual C++
 - adicionando bibliotecas ao Serviço de Integração [189](#)
 - classes wrapper para [189](#)
 - desenvolvendo procedimentos externos COM [171](#)
 - distribuindo procedimentos manualmente [186](#)
 - tipos de dados COM [187](#)

X

- XML
 - dividindo saída grande [513](#)