

## **Oracle® Business Activity Monitoring**

Enterprise Link Transforms リファレンス・ガイド

10g (10.1.3.1.0)

**部品番号 : B31884-01**

2007 年 1 月

Oracle Business Activity Monitoring Enterprise Link Transforms リファレンス・ガイド, 10g (10.1.3.1.0)

部品番号 : B31884-01

原本名 : Oracle Business Activity Monitoring Enterprise Link Transforms Reference 10g (10.1.3.1.0)

Copyright © 2002, 2006 Oracle. All rights reserved.

#### 制限付権利の説明

このプログラム（ソフトウェアおよびドキュメントを含む）には、オラクル社およびその関連会社に所有権のある情報が含まれています。このプログラムの使用または開示は、オラクル社およびその関連会社との契約に記された制約条件に従うものとします。著作権、特許権およびその他の知的財産権と工業所有権に関する法律により保護されています。

独立して作成された他のソフトウェアとの互換性を得るために必要な場合、もしくは法律によって規定される場合を除き、このプログラムのリバース・エンジニアリング、逆アセンブル、逆コンパイル等は禁止されています。

このドキュメントの情報は、予告なしに変更される場合があります。オラクル社およびその関連会社は、このドキュメントに誤りが無いことの保証は致し兼ねます。これらのプログラムのライセンス契約で許諾されている場合を除き、プログラムを形式、手段（電子的または機械的）、目的に関係なく、複製または転用することはできません。

このプログラムが米国政府機関、もしくは米国政府機関に代わってこのプログラムをライセンスまたは使用する者に提供される場合は、次の注意が適用されます。

#### U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

このプログラムは、核、航空産業、大量輸送、医療あるいはその他の危険が伴うアプリケーションへの用途を目的としておりません。このプログラムをかかるとの目的で使用する場合、上述のアプリケーションを安全に使用するために、適切な安全装置、バックアップ、冗長性（**redundancy**）、その他の対策を講じることは使用者の責任となります。万一かかるとのプログラムの使用に起因して損害が発生いたしましても、オラクル社およびその関連会社は一切責任を負いかねます。

Oracle、JD Edwards、PeopleSoft、Siebel は米国 Oracle Corporation およびその子会社、関連会社の登録商標です。その他の名称は、他社の商標の可能性があり得ます。

このプログラムは、第三者の Web サイトへリンクし、第三者のコンテンツ、製品、サービスへアクセスすることがあります。オラクル社およびその関連会社は第三者の Web サイトで提供されるコンテンツについては、一切の責任を負いかねます。当該コンテンツの利用は、お客様の責任になります。第三者の製品またはサービスを購入する場合は、第三者と直接の取引となります。オラクル社およびその関連会社は、第三者の製品およびサービスの品質、契約の履行（製品またはサービスの提供、保証義務を含む）に関しては責任を負いかねます。また、第三者との取引により損失や損害が発生いたしましても、オラクル社およびその関連会社は一切の責任を負いかねます。

---

# 目次

	はじめに	vii
	対象読者	viii
	表記規則	viii
	このマニュアルの構成	viii
	Copyright Information from Group 1 Software	ix
	サポートおよびサービス	ix
<b>第 1 章 :</b>	<b>概要</b>	<b>1</b>
	Transform について	2
	Transform の処理の概要	5
	マルチパスおよびシングルパスの処理	6
	コピー処理およびインプレース処理	6
	ソート	7
	パフォーマンスを向上させるための一般的なルール	8
	Transform および不明スキーマ	9
	Transform の特性のサマリー	9
	その他の推奨事項	12
<b>第 2 章 :</b>	<b>データ・ソース Transform</b>	<b>15</b>
	Delimited Text File Source	16
	Delimited Text File Source Transform の使用方法	17
	「Delimited Text File Source」ダイアログ・ボックスの設定	18
	Delimited Text File Source Transform に対するヒント	20
	Delimited Text File Source の特性	20
	Flat File Source	21
	「Flat File Source」ダイアログ・ボックスの設定	21
	Flat File Source Transform に対するヒント	31
	サポートされている入力データ型	31
	サポートされている出力データ型	32
	Flat File Source の特性	32
	OLEDB Source	32
	「OLEDB Source」ダイアログ・ボックスの設定	33
	OLEDB Source に対するヒント	34

OLEDB Source の特性 . . . . .	34
Oracle BAM Enterprise Message Receiver . . . . .	34
「Oracle BAM Message Receiver」 ダイアログ・ボックスの設定 . . . . .	35
Oracle BAM Message Receiver の特性 . . . . .	36
Oracle BAM Reader . . . . .	36
「Oracle BAM Reader」 ダイアログ・ボックスの設定 . . . . .	37
Oracle BAM Reader の特性 . . . . .	37
SQL Query . . . . .	37
SQL Query の設定 . . . . .	38
SQL エディタの使用 . . . . .	39
SQL Query Transform に対するヒント . . . . .	40
SQL Query の特性 . . . . .	40
Time Generation . . . . .	41
「Time Generation」 ダイアログ・ボックスの設定 . . . . .	41
Time Generation Transform に対するヒント . . . . .	42
Time Generation の特性 . . . . .	43
VBScriptSource . . . . .	43
XML Source . . . . .	44
XML Source Transform の設定 . . . . .	46
XML Source の特性 . . . . .	51
<b>第 3 章 : 表示シンク Transform . . . . .</b>	<b>53</b>
Grid . . . . .	54
<b>第 4 章 : 非表示シンク Transform . . . . .</b>	<b>55</b>
Command Line Sink . . . . .	56
「Command Line Sink」 ダイアログ・ボックスの設定 . . . . .	56
Command Line Sink Transform に対するヒント . . . . .	57
Command Line Sink の特性 . . . . .	57
Delimited Text File Sink . . . . .	57
「Delimited Text File Sink」 ダイアログ・ボックスの設定 . . . . .	58
Delimited Text File Sink Transform に対するヒント . . . . .	58
Delimited Text File Sink の特性 . . . . .	59
Oracle BAM Alert . . . . .	59
「Oracle BAM Alert」 ダイアログ・ボックスの設定 . . . . .	60
Oracle BAM Alert の特性 . . . . .	60
Oracle BAM Delete . . . . .	60
Oracle BAM Delete Transform の設定 . . . . .	61
Oracle BAM Delete の特性 . . . . .	62
Oracle BAM Insert . . . . .	63
「Oracle BAM Insert」 ダイアログ・ボックスの設定 . . . . .	63
Oracle BAM Insert の特性 . . . . .	64
Oracle BAM Message Sender . . . . .	64
「Oracle BAM Message Sender」 ダイアログ・ボックスの設定 . . . . .	64
Oracle BAM Message Sender の特性 . . . . .	65
Oracle BAM Message Tracker . . . . .	65
「Oracle BAM Message Tracker」 ダイアログ・ボックスの設定 . . . . .	66
Oracle BAM Message Tracker の特性 . . . . .	66

	Oracle BAM Update	66
	「Oracle BAM Update」ダイアログ・ボックスの設定	67
	Oracle BAM Update の特性	69
	SQL Command Sink	70
	「SQL Command Sink」ダイアログ・ボックスの設定	70
	SQL Command Sink の特性	71
	Terminal Sink	72
	VBScriptSink	72
<b>第 5 章 :</b>	<b>データ・フロー制御 Transform</b>	<b>73</b>
	Comparison	74
	「Comparison」ダイアログ・ボックスの設定	74
	Comparison Transform に対するヒント	76
	Comparison の特性	77
	Conditional Splitter	77
	「Conditional Splitter」ダイアログ・ボックスの設定	78
	Conditional Splitter Transform に対するヒント	78
	正規表現	79
	Conditional Splitter の特性	79
	Join	80
	「Join」ダイアログ・ボックスの設定	80
	Join Transform に対するヒント	82
	Join の特性	83
	Round Robin Splitter	83
	Splitter	84
	Subplan	84
	「SubPlan Iteration」ダイアログ・ボックスの設定	85
	Subplan Transform に対するヒント	88
	サブプランの特性	88
	Union	88
<b>第 6 章 :</b>	<b>データ操作 Transform</b>	<b>89</b>
	Add Columns	91
	「Add Columns」ダイアログ・ボックスの設定	91
	Add Columns Transform に対するヒント	92
	Add Columns の特性	92
	Column Select	93
	「Column Select」ダイアログ・ボックスの設定	93
	Column Select Transform に対するヒント	94
	Column Select の特性	94
	Concatenation	95
	「Concatenation」ダイアログ・ボックスの設定	95
	Concatenation の特性	96
	Disk Sort	97
	「Disk Sort」ダイアログ・ボックスの設定	98
	Disk Sort Transform に対するヒント	99
	Disk Sort の特性	99

Expression Calculator .....	100
「Expression Calculator」 ダイアログ・ボックスの設定 .....	100
式の作成 .....	102
Expression Calculator に対するヒント .....	104
Expression Calculator の特性 .....	106
Filter .....	106
「Filter」 ダイアログ・ボックスの設定 .....	107
Filter Transform に対するヒント .....	107
Filter の特性 .....	107
Key Generation .....	108
「Key Generation」 ダイアログ・ボックスの設定 .....	108
Key Generation Transform に対するヒント .....	109
Key Generation の特性 .....	109
Key Lookup .....	109
「Key Lookup」 ダイアログ・ボックスの設定 .....	110
Key Lookup Transform に対するヒント .....	111
Key Lookup の特性 .....	112
Memory Sort .....	112
「Memory Sort」 ダイアログ・ボックスの設定 .....	113
Memory Sort Transform に対するヒント .....	113
Memory Sort の特性 .....	114
Message Digest .....	114
キーの構成 .....	115
Message Digest Transform の設定 .....	115
Message Digest Transform の使用方法 .....	117
Oracle BAM Lookup .....	117
「Oracle BAM Lookup」 ダイアログ・ボックスの設定 .....	118
Oracle BAM Lookup の特性 .....	119
Pivot .....	119
「Pivot」 ダイアログ・ボックスの設定 .....	120
Pivot Transform に対するヒント .....	122
Pivot の特性 .....	122
Rank .....	122
「Rank」 ダイアログ・ボックスの設定 .....	122
Rank の特性 .....	124
Record Number .....	125
「Record Number」 ダイアログ・ボックスの設定 .....	125
Record Number の特性 .....	125
Remove Duplicate Rows .....	126
「Remove Duplicate Rows」 ダイアログ・ボックスの設定 .....	127
Remove Duplicate Rows Transform に対するヒント .....	127
Remove Duplicate Rows の特性 .....	128
Rename Columns .....	128
「Rename Columns」 ダイアログ・ボックスの設定 .....	129
Rename Columns Transform に対するヒント .....	129
Rename Columns の特性 .....	130
Search & Replace .....	130

	「Search & Replace」ダイアログ・ボックスの設定 . . . . .	131
	Search & Replace Transform に対するヒント . . . . .	131
	Search & Replace の特性 . . . . .	131
	SQL Command . . . . .	132
	「SQL Command」ダイアログ・ボックスの設定 . . . . .	132
	SQL Command の特性 . . . . .	134
	Substring . . . . .	134
	「Substring」ダイアログ・ボックスの設定 . . . . .	135
	Substring Transform に対するヒント . . . . .	135
	Substring の特性 . . . . .	135
	Time Lookup . . . . .	136
	Time Lookup Transform の設定 . . . . .	136
	Time Lookup Transform に対するヒント . . . . .	137
	Time Lookup の特性 . . . . .	137
	Transpose . . . . .	137
	「Transpose」ダイアログ・ボックスの設定 . . . . .	138
	Transpose Transform に対するヒント . . . . .	139
	Transpose の特性 . . . . .	139
	VBScriptCopy . . . . .	140
	VBScriptInplace . . . . .	140
<b>付録 A:</b>	<b>Expression Calculator の関数 . . . . .</b>	<b>141</b>
	日時関数 . . . . .	142
	文字列関数 . . . . .	144
	数学関数 . . . . .	145
	論理演算子 . . . . .	146
	バイナリ演算子 . . . . .	147
	その他の演算子 . . . . .	147
	カルキュレータの演算子 . . . . .	148
	関係演算子および等価演算子 . . . . .	148
	単項演算子 . . . . .	148
<b>付録 B:</b>	<b>データ型 . . . . .</b>	<b>149</b>
	10 進演算計算 . . . . .	150
	Precision および Scale の自動設定と手動設定 . . . . .	151
	リテラルの処理 . . . . .	152
	10 進演算の有効化と無効化の切替え . . . . .	152
	decimal フィールドを使用する Expression Calculator 関数 . . . . .	153
	サポートされているバイナリ入力データ型 . . . . .	154
	<b>索引 . . . . .</b>	<b>157</b>





---

# はじめに

ここで説明する内容は次のとおりです。

- 対象読者
- 表記規則
- このマニュアルの構成
- サポートおよびサービス

## 対象読者

このマニュアルは、Oracle BAM Enterprise Link 環境で作業を行うデータ設計者およびアプリケーション・プログラマを対象としています。Transform を使用して複雑なプランを作成する Oracle BAM Design Studio のパワー・ユーザーが想定されています。特定のデータベース環境を理解し、データベース・スキーマの設計、データの移動およびディメンション・モデリングの概念に精通していることが前提となっています。

Oracle BAM Design Studio をよく理解していない場合は、このマニュアルを参照する前に『Oracle Business Activity Monitoring Enterprise Link Design Studio ユーザーズ・ガイド』を参照してください。

## 表記規則

このマニュアルでは、次の表記規則を使用しています。

- キーボード上のキーは、[Esc]、[Enter] などの名前で示されます。
- 入力するコマンドまたはマウスを使用してクリックする項目は、「**更新**」をクリック、「**yes** と入力」などのように太字で示されます。
- ユーザーが指定する必要があるマシン名やその他の変数は、**<machine name>** などのように示されます。山カッコ (<>) は入力しないでください。
- パス、ファイル名およびコードは、C:\¥Oracle BAM Enterprise Link などのように固定幅フォントで示されます。

## このマニュアルの構成

このマニュアルは、Transform のリファレンス・ガイドです。このマニュアルでは、Transform の適用時に必要な設定および追加の情報について説明します。このマニュアルは、Transform のタイプに基づいて章に分けられています。次の要約を参照して、参照する必要がある章を確認してください。

- **第1章「概要」**では、Transform の概要および Transform とデータ・フローの管理に関する追加の概念について説明します。
- **第2章「データ・ソース Transform」**では、ソース・データベースからデータを選択する Transform について説明します。
- **第3章「表示シンク Transform」**では、プランの最後にデータを表示するために使用する Transform について説明します。
- **第4章「非表示シンク Transform」**では、結果を表示せずにプランを終了する Transform について説明します。
- **第5章「データ・フロー制御 Transform」**では、データ・フローの構造を変更する Transform について説明します。
- **第6章「データ操作 Transform」**では、データ・フロー内でデータを操作および処理する Transform について説明します。

- 付録 A 「Expression Calculator の関数」では、Expression Calculator で使用可能な関数を示します。
- 付録 B 「データ型」では、10 進演算について説明し、Flat File Source Transform でサポートされているバイナリ入力データ型を示します。

## Copyright Information from Group 1 Software

Information in this document is subject to change without notice. Companies, names and data used in examples herein are fictitious unless otherwise noted. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Group 1 Software, Inc.

Copyright 1996-2004 Group 1 Software, Inc. All rights reserved.

Trademark Notices:

Information Studio and Sagent are registered trademarks of Group 1 Software, Inc. The Sagent Logo, Sagent Design Studio, Flashcube, and StarMart are trademarks of Group 1 Software, Inc. All other trademarks are property of their respective companies.

Any provisions of the Sagent license related to ICU that differ from the IBM Public License 1.0 are offered by Sagent alone and not by any other party. The Source Code of the ICU program is available from Sagent upon written request. Further information regarding the ICU Program may be found at: <http://oss.software.ibm.com/icu/>.

You may make up to three copies of this publication, without the express written permission of Group 1 Software, for use at the authorized installation site. Reproduction and distribution outside of the licensed site is strictly prohibited. The contents of this Guide are subject to the confidentiality and non-disclosure provisions in your license agreement with Group 1 Software.

Group 1 products are warranted to perform as documented on hardware platforms running versions/releases of operating systems, subsystems, compilers, and facilities that are currently supported by their manufacturers. Please see your license agreement for additional details.

## サポートおよびサービス

次の各項に、各サービスに接続するための URL を記載します。

### Oracle サポート・サービス

オラクル製品サポートの購入方法、および Oracle サポート・サービスへの連絡方法の詳細は、次の URL を参照してください。

<http://www.oracle.co.jp/support/>

## 製品マニュアル

製品のマニュアルは、次の URL にあります。

<http://otn.oracle.co.jp/document/>

## 研修およびトレーニング

研修に関する情報とスケジュールは、次の URL で入手できます。

<http://www.oracle.co.jp/education/>

## その他の情報

オラクル製品やサービスに関するその他の情報については、次の URL から参照してください。

<http://www.oracle.co.jp>

<http://otn.oracle.co.jp>

**注意：** ドキュメント内に記載されている URL や参照ドキュメントには、Oracle Corporation が提供する英語の情報も含まれています。日本語版の情報については、前述の URL を参照してください。

# 第 1 章

---

## 概要

ここで説明する内容は次のとおりです。

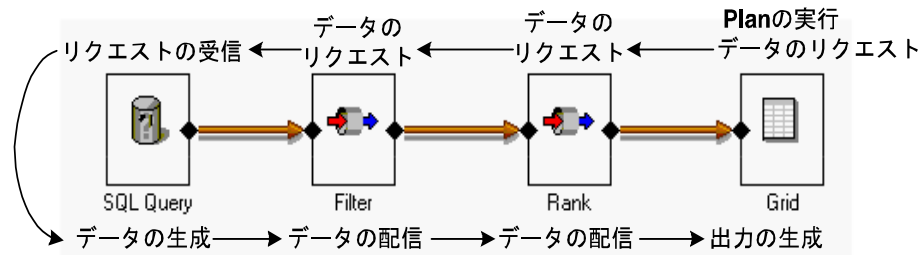
- [Transform](#) について
- [Transform](#) の処理の概要

## Transform について

Transform によって、Oracle BAM Enterprise Link でのデータの抽出、変換およびロード機能が提供されます。特定の状況に対応して Transform を選択すると、様々なソースからのデータの選択、データに対する計算の実行、データ・スキーマの変更、他のターゲット・データベースへのデータのロードを行うことができます。リアルタイム・データおよび履歴データを解析し、グラフまたはレポートにデータを表示することができます。

Transform はプランの基本的な要素です。それぞれの Transform は、専門の操作を実行し、ソース、ステップまたはシンク Transform として機能します。ソース Transform によってデータ・ソースからデータが生成され、ステップによって入力レコードおよび出力レコードが処理され、シンクによって結果がロード、保存または表示されてデータ・フローが終了されます。データ・フローに Transform を追加してプランを作成します。

プランは、プル・モデルで動作します。概念上、プランは、シンクから逆方向に実行されます。Transform では、ダウンストリーム Transform でデータがリクエストされるまでデータは処理されません。実行時、次のアップストリーム Transform からデータがリクエストされると、シンクによってプランの実行が開始されます。データへのこのリクエストは、ソース Transform に到達します。このソース Transform によって、処理用のデータ・ダウンストリームが配信されます。



Transform は、Tool Bin 内でタイプごとに分けられています。次の項に、各 Transform およびその説明を示すクイック・リファレンス表を示します。

- [データ・ソース Transform](#)
- [表示シンク Transform](#)
- [非表示シンク Transform](#)
- [データ・フロー制御 Transform](#)
- [データ操作 Transform](#)

**注意：** 使用しているインストールに、このマニュアルで説明する Transform がすべて含まれているわけではありません。Tool Bin 内の Transform は、ライセンスを取得した Oracle BAM Enterprise Link 製品によって異なります。

## データ・ソース Transform

次の Transform では、データ・ソースが指定されます。

Transform 名	説明
<a href="#">Delimited Text File Source</a>	ソース・ファイルからデリミタ付きフィールドをインポートします。
<a href="#">Flat File Source</a>	ソース・ファイルからデリミタ付きフィールドまたは固定長フィールドをインポートします。
<a href="#">OLEDB Source</a>	データ・ソースとして OLE DB プロバイダに接続します。
<a href="#">Oracle BAM Enterprise Message Receiver</a>	エンタープライズ・メッセージ・ソースからのリアルタイム情報のフローを継続的に処理します。
<a href="#">Oracle BAM Reader</a>	データ・オブジェクトからのデータの読み取りおよび次のステップへのレコードの送信を実行します。
<a href="#">SQL Query</a>	SQL を介してアクセス可能な他の場所からデータベース・レコードまたはレコードを取得します。
<a href="#">Time Generation</a>	データ・フローで日時値を生成します。
<a href="#">VBScriptSource</a>	VBScript を使用して外部アプリケーションから入力データを取得します。
<a href="#">XML Source</a>	ソース・ファイルから XML ファイルをインポートします。

## 表示シンク Transform

次の Transform (クライアント・シンクとも呼ばれる) では、データが表示されます。

Transform 名	説明
<a href="#">Grid</a>	結果を列形式で表示します。

## 非表示シンク Transform

次の Transform では、表示なしでデータの宛先が指定されます。

Transform 名	説明
<a href="#">Command Line Sink</a>	結果をコマンドライン・ユーティリティに送信します。
<a href="#">Delimited Text File Sink</a>	結果をデリミタ付きの値としてテキスト・ファイルにエクスポートします。
<a href="#">Oracle BAM Alert</a>	指定したアラートをプランの条件に基づいて起動します。
<a href="#">Oracle BAM Delete</a>	ADC 内のデータ・オブジェクトからレコードを削除します。
<a href="#">Oracle BAM Insert</a>	受信したレコードを ADC 内のデータ・オブジェクトに挿入します。
<a href="#">Oracle BAM Message Sender</a>	受信したレコードを ADC 内のデータ・オブジェクトに挿入します。
<a href="#">Oracle BAM Message Tracker</a>	グローバル・トランザクション用のメッセージを追跡します。

Transform 名	説明
Oracle BAM Update	ADC 内のデータ・オブジェクトのレコードを更新します。
SQL Command Sink	各行に対してカスタマイズされた SQL コマンドを実行します。
Terminal Sink	表示なしまたは後続のアクションなしでデータ・フローを終了します。
VBScriptSink	VBScript を使用して外部アプリケーションにレコードを送信します。

## データ・フロー制御 Transform

次の Transform では、プランのデータ・フローの構造が定義されます。

Transform 名	説明
Comparison	2 セットの入力レコードを比較し、一致するレコードを識別します。
Conditional Splitter	フィルタ条件を使用してデータ・フローを分割します。
Join	2 つのデータ・フローの列を 1 つに結合します。
Round Robin Splitter	指定した数の出力にデータを分散します。
Splitter	データ・フロー出力の 2 つの同一コピーを作成します。
Subplan	説明図を簡略化し、プランのステップを階層的にグループ化してプランを編成します。また、データ・フローの反復処理も提供します。
Union	2 つのデータ・フローを 1 つに結合します。

## データ操作 Transform

次の Transform では、データが処理されます。

Transform 名	説明
Add Columns	1 つ以上の空の列を追加します。
Column Select	列の削除および並置を実行します。
Concatenation	列データと入力テキストを結合します。
Disk Sort	ディスク・ファイルを使用して、列をソートします。
Expression Calculator	式を使用して、計算された列を追加します。
Filter	フィルタ条件を使用して、行を削除します。
Key Generation	一意の数値キー値を生成します。
Key Lookup	表内の値を検索して、別の表に挿入します。
Memory Sort	メモリーを使用して、列内のレコードをソートします。
Message Digest	任意の型の入力データに基づいてキーを作成します。



Transform 名	説明
Oracle BAM Lookup	データ・フロー内のデータおよびデータ・オブジェクト内のデータのキー・フィールドに対して検索を実行します。
Pivot	列をクロス集計形式で集計および配置します。
Rank	列の上位の行または下位の行を出力します。
Record Number	順序番号を付けて列を追加します。
Remove Duplicate Rows	データ内の重複行を削除します。
Rename Columns	列名を変更します。
Search & Replace	文字列または部分文字列を別の文字列に置換します。
SQL Command	各行に対してカスタマイズされた SQL コマンドを実行します。
Substring	文字列値から部分文字列を抽出します。
Time Lookup	時間ディメンションのファクト表でキー値を生成します。
Transpose	列を行形式で配置します。
VBScriptCopy	VBScript を使用してレコード属性およびデータを変更できるように、レコードをコピーします。
VBScriptInplace	VBScript を使用してデータ値のみを変更して、レコードを変更します。

## Transform の処理の概要

この項では、Transform でのデータの処理方法およびプランのパフォーマンスへのその方法の影響について説明します。プランのパフォーマンスとは、レコードに対してリクエストされた処理を特定のプランで完了する場合の効率性のことです。

プランのパフォーマンスは、使用可能なリソースで処理する必要があるプランの複雑度およびデータの量によって影響を受けます。また、プランでの Transform の組合せおよび順序もパフォーマンスに影響します。Transform によるデータの処理方法について理解すると、効率的なプランの設計方法に関してよりの確な判断を行うことができます。

Transform で他のレコードを参照することなく 1 つのレコードを処理できる場合、レコード・ストレージには最小のリソースが使用されます。1 つのレコードまたは 1 ブロックのレコードが一度に処理され、次の Transform に配信されます。一部の Transform では、処理を続行する前にすべてのレコードが必要です。たとえば、ソート Transform では、レコードのソートおよび処理の前にすべての入力レコードが累積されます。

## マルチパスおよびシングルパスの処理

Transform では、同じレコードに対して 1 つのパスまたは複数のパスでレコードが処理されます。

マルチパスの Transform では、データに対して複数のパスが実行されます。また、レコードが再フォーマットされる場合もあります。マルチパスの Transform では、入力データがバッファに格納されます。これは、入力データの読取り時に入力データをメモリーまたはディスクに保持することによって行われます。これを行うには、シングルパスの Transform より多くのリソースが必要です。

マルチパスの Transform の例としては、Percent of Total Transform があります。Percent of Total Transform では、合計を計算するためにデータの 1 つ目のパスが、各値の合計の割合（パーセント）を計算するために 2 つ目のパスが実行されます。

## コピー処理およびインプレース処理

Transform では、レコードは、新しい出力を生成するためにコピーされるか、またはインプレース処理されます。インプレース処理では、レコードの値は変更されますが、レコード・サイズの変更またはフィールドの追加や削除は行われません。

コピー Transform では、出力用にレコードを再フォーマットできます。レコードはブロックに格納され、処理されます。コピー Transform では、1 つ目のセットのブロックが入力として読み取られ、2 つ目のセットのブロックが出力として生成されます。出力として生成されたブロックは、入力ブロックとは別の形式にすることができます。フィールドではサイズおよび値、レコードではサイズを変更することができます。また、フィールドは追加または削除することができます。文字列値を操作するほとんどの Transform はコピー Transform です。コピー Transform では、入力ブロックを出力ブロックにコピーする必要があるため、インプレース Transform より多くのメモリー・リソースが必要です。コピー Transform は、シングルパスまたはマルチパスの Transform として使用できます。

いくつかの Transform では、既存のフィールドでの出力の生成を選択した場合はインプレース処理が、新しいフィールドへの出力の格納を選択した場合はコピー処理が実行されます。

Expression Calculator 内に出力用の新しいフィールドを追加すると、シングルパスのコピー Transform として動作します。Expression Calculator 内に作成した新しいフィールドによって、レコードが再フォーマットされ、1 セットの出力ブロックが実行時に作成されます。これは、シングルパスの Transform です。この Transform は、各レコードに対して 1 回のみ動作して、単一レコードのコンテキスト内ですべての計算を実行します。

## ソート

一部の Transform では、入力のソートが必要です。ソートは、プランの様々なステージで実行できます。ソートは、ディスク・ソートとして SQL Query で実行するか、またはメモリー内ソートとして内部で実行できます。

ソートが行われるプランのステージによって、必要なリソースの量が影響を受けます。SQL Query のステージで実行されるソートでは、必要なリソースが少なくなります。内部で実行されるソートでは、すべてのデータをメモリーまたはディスクにコピーする必要があるため、より多くのリソースが使用されます。

プランは、ソートの実行場所を決定する場合、特定のルールに従います。ソートは、次のいずれかの条件にあてはまらないかぎり、SQL Query のステージで自動的に実行されます。

- プランに SQL Query が存在しない。
- SQL Query は、ユーザーが入力したものであるか、またはユーザーが作成した ORDER BY 句を含んでいる。

中間 Transform (SQL Query とソートを必要とする Transform の間に位置する) の動作もソートが実行される場所に影響します。ソートは、中間 Transform で次のいずれかが実行されると、SQL Query のステージでは実行されません。

- いずれかのソート・フィールドの変更
- ソートされたデータのリクエスト
- レコードの追加または削除
- ソート・フィールドの生成
- 複数の入力または出力の格納

これらのいずれかの条件にあてはまる場合、ソートはディスク・ソートまたは内部ソートのいずれかで実行されます。プランに 1 つ以上の非表示シンク (表示出力が含まれていないシンク) が含まれている場合は、ディスク・ソートが実行されます。プランが実行され、そのデータ・フローに表示されない場合、ディスク・ソートは自動的に実行されます。

プラン内のシンクが表示シンクのみの場合、ソートは、ソートが必要な Transform によって内部で実行されます。内部でのソートには、ディスク・ソートまたは SQL Query を使用したソートより多くのメモリーが必要です。

次の Transform では、ソートが必要です。

- Join
- Comparison
- Rank
- Pivot
- Memory Sort
- Disk Sort
- Remove Duplicate Rows

## パフォーマンスを向上させるための一般的なルール

プランのパフォーマンスを向上させるには、プランに次の変更を行うことを検討してください。

- すでにソートされているデータの場合、Join Transform、Comparison Transform、Rank Transform、Remove Duplicate Rows Transform および Pivot Transform で「Data is pre-sorted」チェック・ボックスを選択します。不要なソートを実行しないことで、プランのパフォーマンスを向上させることができます。
- データ・フローの先頭近辺に Add Columns Transform を追加して、1 ステップでプランのすべての新しい列を指定します。Transform をこのように配置すると、他の Transform でデータをインプレース処理できるため、使用するリソースが少なくなります。たとえば、Expression Calculator は、計算された出力を Add Columns Transform によって作成された列内に格納できるため、インプレース Transform として機能できます。
- 可能な場合、シングルパスの Transform を使用します。マルチパスの Transform の処理は、個別のシングルパスの Transform に分けることができます。たとえば、Percent of Total で多くのメモリーが必要となり、プランの速度が低下した場合は、2つの個別のプランを実行します。最初のプランでは、非ピボット・モードで Pivot Transform を使用して合計が計算されます。2つ目のプランでは、Expression Calculator Transform を使用して割合（パーセント）が計算されます。Percent of Total Transform を使用するプランではなく、これらの2つの個別のプランを使用すると、パフォーマンスを向上できます。

別の例として、Key Lookup Transform を Join Transform と置き換える方法があります。プランの検索表が入力量と比較して大きい場合、または検索している値があまり繰り返して入力されていない場合、Key Lookup は、リソース集中型となり、プランの速度が低下します。Key Lookup Transform を使用してランダム・アクセス検索を実行するのではなく、Join Transform を使用して一致検索 / マージ操作を実行します。

- 複数の Expression Calculator Transform を追加するのではなく、1つの Expression Calculator で複数の式を作成します。
- ロード・プランにロード・ステップの段階でパフォーマンスの問題があり、データベース・サーバーでパラレル・ロードを実行できる場合は、同じ表にロードする複数のバッチ・ローダーに接続されている Round Robin Splitter の使用を試行します。ロード・ステップは、バッチ・ローダーを Terminal Sink に置き換えて実行時間が大幅に向上するかどうかを確認することによってテストできます。

## Transform および不明スキーマ

スキーマには、レコードのデータ構造に関する情報が含まれています。各列には、サイズ、データ型などの特定の属性が含まれています。データ・スキーマはこれらの情報によって構築されています。これらの情報は、データの処理時およびロード時にそのまま保存する必要があります。

データ・フローに不明スキーマが含まれている場合、Transform ダイアログ・ボックスでは、列をプラン内の Part 名ごとに表示し、列の選択および操作のためのユーザー・インタフェース機能を提供できます。一部の Transform では、スキーマが不明な場合、別のダイアログ・ボックスが表示されます。これらのダイアログ・ボックスでは、Part 名または列名を入力して Transform に指定する必要があります。不明スキーマの原因には、次のものがあります。

- 接続されていない Transform が使用されているデータ・フロー
- ユーザー入力の SQL
- Pivot Transform では、出力スキーマは提供されません。

Pivot Transform の後または他のデータ・フロー内に Add Columns Transform を追加して、ダウンストリーム Transform で列名の認識および表示を実行するために列を追加できます。プランの実行時に名前ベースの一致検索が実行されます。

次に、不明スキーマに対しては異なって機能する Transform の例を示します。

- Column Select
- Disk Sort
- Memory Sort
- Rename Columns
- Transpose

## Transform の特性のサマリー

このマニュアルで説明する各 Transform に関して、基本的な Transform の特性をリファレンス表に示します。これらの特性は、Transform で実行されるデータ処理のタイプを示しています。

特性の定義については、表 1 を参照してください。Transform でのデータの処理方法を理解すると、プランのパフォーマンスを向上させるための決定を行う場合に役立てることができます。

すべての Transform およびそれらの特性の完全な表については、表 2 を参照してください。

表 1: Transform の特性の定義

Transform の特性	説明
パッケージ・タイプ	Transform は、Oracle BAM Enterprise Link サーバーに付属しています。追加の Transform は、Power Products Transforms として個別に販売されています。
スキーマの変更	Transform では、データ・フローを介して渡されるデータ・スキーマの構造を変更できます。
マルチパスおよびシングルパスの処理	Transform では、同じレコードに対して 1 つのパスまたは複数のパスでレコードを処理できます。プランによっては、評価の必要がある追加のリソースがマルチパスの Transform で必要な場合があります。
コピーまたはインプレース	Transform では、レコードをインプレース変更するか、または処理するレコードの新しいコピーを作成してレコードを処理できます。一部の Transform は、指定した設定に基づいていずれかの方法で機能できます。
入力のソート	Transform では、入力がソートされてから処理されるようにリクエストできます。
プロパティの公開	Transform では、プランの実行時に値を設定できるプロパティがプランのプロンプト、フィルタなどに公開されます。

次の表に、すべての Transform の特性を示します。

表 2: Transform およびそれらの特性

Transform	パッケージ・タイプ	スキーマの変更	マルチパスまたはシングルパス	コピーまたはインプレース	入力のソート	プロパティの公開
Add Columns	Basic	実行する	シングルパス	コピー	実行しない	一部
Analytical Calculator	Power Product	実行する	いずれか (関数による)	コピー	実行する	なし
Column Select	Basic	実行する	シングルパス	コピー	実行しない	一部
Command Line Sink	Power Product	実行しない	シングルパス	N/A	実行しない	一部
Comparison	Basic	実行する	マルチパス (処理時にメモリーを解放)	インプレース	実行する	一部
Concatenation	Basic	実行する	シングルパス	コピー	実行しない	一部
Conditional Splitter	Basic	実行する	シングルパス	コピー	実行しない	すべて
Delimited Text File Sink	Power Product	実行する	シングルパス	コピー	実行しない	一部
Disk Sort	Basic	実行しない	マルチパス	コピー	実行する	一部
Expression Calculator	Basic	実行する	シングルパス	既存のフィールドを更新する場合はインプレース、それ以外の場合はコピー	実行しない	なし
Filter	Basic	実行しない	シングルパス	コピー	実行しない	すべて
Flat File Source	Basic	実行する	シングルパス	コピー	実行しない	一部
Grid	Basic	実行しない	N/A	N/A	実行する	なし

表 2: Transform およびそれらの特性 (続き)

Transform	パッケージ・タイプ	スキーマの変更	マルチパスまたはシングルパス	コピーまたはインプレース	入力のソート	プロパティの公開
Join	Basic	実行しない	マルチパス (処理時にメモリーを解放)	コピー	実行する	一部
Key Generation	Basic	実行する	シングルパス	コピー	実行しない	一部
Key Lookup	Basic	実行する	シングルパス	既存のフィールドを更新する場合はインプレース、それ以外の場合はコピー	実行しない	一部
Memory Sort	Basic	実行しない	マルチパス	コピー	実行する	一部
Message Digest	Basic	実行しない	シングルパス	コピー	実行しない	一部
OLED Source	Basic	実行しない	シングルパス	コピー	実行しない	一部
Pivot	Basic	実行する	マルチパス	コピー	実行する	一部
Rank	Basic	実行する	マルチパス	コピー	実行する	すべて
Record Number	Basic	実行する	シングルパス	コピー	実行しない	すべて
Remove Duplicate Rows	Power Product	実行する	マルチパス	コピー	実行する	一部
Rename Columns	Basic	実行する	シングルパス	インプレース	実行しない	一部
Round Robin Splitter	Basic	実行する	N/A	N/A	実行しない	N/A
Search & Replace	Basic	実行する	シングルパス	コピー	実行しない	すべて
Splitter	Basic	実行する	N/A	N/A	実行しない	N/A
SQL Command	Power Product	実行しない	シングルパス	インプレース	実行しない	すべて
SQL Command Sink	Power Product	実行しない	シングルパス	N/A	実行しない	すべて
SQL Query	Basic	実行する	シングルパス	コピー	N/A	すべて
Subplan	Basic	N/A	N/A	N/A	状況によって異なる	N/A
Substring	Basic	実行する	シングルパス	コピー	実行しない	すべて
Terminal Sink	Basic	実行しない	シングルパス	N/A	実行しない	N/A
Time Generation	Power Product	実行する	シングルパス	コピー	実行しない	一部
Time Lookup	Power Product	実行する	シングルパス	コピー	実行しない	なし
Transpose	Basic	実行する	シングルパス	コピー	実行しない	すべて
Union	Basic	実行しない	N/A	N/A	実行しない	N/A
VBScriptCopy	Basic	実行する	シングルパス	コピー	実行しない	なし
VBScriptInplace	Basic	実行しない	シングルパス	インプレース	実行しない	なし
VBScriptSink	Power Product	実行しない	シングルパス	N/A	実行しない	なし
VBScriptSource	Basic	実行する	シングルパス	コピー	実行しない	なし
XML Source	Basic	実行する	シングルパス	コピー	N/A	なし

## その他の推奨事項

この項では、Transform および Oracle BAM Design Studio での作業についての一般的なヒントおよび追加情報を示します。

### 再利用のための Transform の保存

Transform は、Tool Bin 内にカスタム Transform として保存できます。カスタム Transform には Transform の設定が保存されるため、設定を再度指定する必要がなくなります。たとえば、SQL 構文を編集した場合は、SQL Query Transform をカスタム Transform として保存できます。Tool Bin にアクセスできる他のユーザーも、この保存された Transform を使用できます。Transform および設定のコピーを保存するには、その Transform のステップを右クリックし、「Create Transform」を選択します。

**注意：** すべての Transform は、保存して再利用できます。ただし、次の Transform は再利用できません。

表 3: 保存して再利用することができない Transform

Grid	Splitter
Union	

### プランへのコメントの追加

プランの各ステップにコメントを追加して、その目的に関する追加情報を提供できます。

コメントを表示および編集するには、次の手順を実行します。

1. データ・フローのステップを右クリックし、「About Step "<step name>"」を選択します。  
「About Step」ダイアログ・ボックスが表示されます。
2. 「comments」フィールドにステップの説明を入力します。
3. 「OK」をクリックして情報を保存します。

### Transform のダイアログ・ボックスでのファイル・パスの指定

一部の Transform のダイアログ・ボックスでは、プランで使用できるようにファイルへのパスを指定する必要があります。Data Flow Service がシステム・アカウント（デフォルトのアカウント）によって実行されている場合は、Data Flow Service マシンに相対してパスを指定する必要があります。個別のマシン上に存在するファイルは、ネットワークマシン上の共有ディレクトリに存在している必要があります。マップされたドライブは検出できません。たとえば、Mars というマシンへのファイル・パスは、¥¥Mars¥CDrive¥Program Files¥Oracle BAM Enterprise Link¥Save.csv のようになります。



## 標準形式および Transform

Transform は、内部的には、プロパティ値を標準形式で格納します。標準形式では、小数点が小数点セパレータとして、カンマがリスト・セパレータおよび千の位のセパレータとして使用されます。

Transform のダイアログ・ボックス以外でプロパティ値を設定する場合は、標準形式を使用する必要があります。



# データ・ソース Transform

ここで説明する内容は次のとおりです。

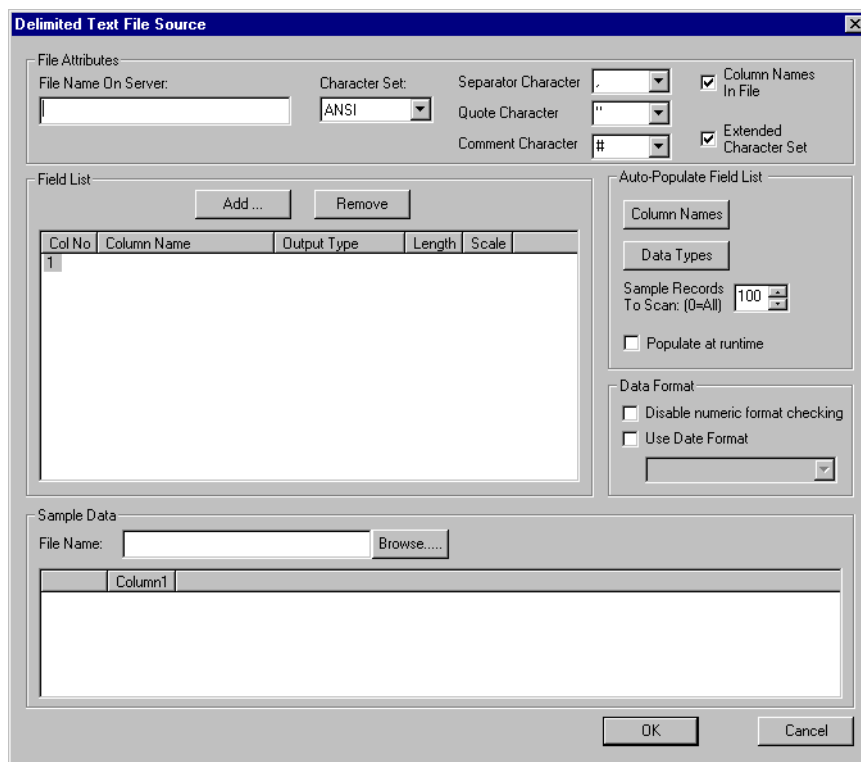
Transform 名	説明
<a href="#">Delimited Text File Source</a>	ソース・ファイルからデリミタ付きフィールドをインポートします。
<a href="#">Flat File Source</a>	ソース・ファイルからデリミタ付きフィールドまたは固定長フィールドをインポートします。
<a href="#">OLEDB Source</a>	データ・ソースとして OLE DB プロバイダに接続します。
<a href="#">Oracle BAM Enterprise Message Receiver</a>	エンタープライズ・メッセージ・ソースからのリアルタイム情報のフローを継続的に処理します。
<a href="#">Oracle BAM Reader</a>	データ・オブジェクトからのデータの読取りおよび次のステップへのレコードの送信を実行します。
<a href="#">SQL Query</a>	SQL を介してアクセス可能な他の場所からデータベース・レコードまたはレコードを取得します。
<a href="#">Time Generation</a>	データ・フローで日時値を生成します。
<a href="#">VBScriptSource</a>	VBScript を使用して外部アプリケーションから入力データを取得します。
<a href="#">XML Source</a>	XML 文書を Oracle BAM Enterprise Link にインポートします。

## Delimited Text File Source

Delimited Text File Source は、テキスト・ファイルからデリミタ付き値を抽出する場合に使用します。パスおよびファイル名を選択し、ファイルの最初の行から列名を読み取るかどうかを指定します。また、サンプル・データが含まれているファイルを使用して、プラン実行前にフィールド指定を適用する方法のテストも行います。

拡張機能には、使用される可能性が高いデータ型および長さに関する各列のスキーマまたはこれらの属性の独自の設定などがあります。また、テキスト・ファイルで使用するデリミタ、引用符およびコメント文字を指定することもできます。

図 1: 「Delimited Text File Source」ダイアログ・ボックス



ここで説明する内容は次のとおりです。

- [Delimited Text File Source Transform の使用方法](#)
- 「Delimited Text File Source」ダイアログ・ボックスの設定
- [Delimited Text File Source Transform に対するヒント](#)
- [Delimited Text File Source の特性](#)

## Delimited Text File Source Transform の使用方法

Delimited Text File Source Transform のパラメータを設定するには、次の手順を実行します。

1. Delimited Text File Source Transform をデータ・フローに配置し、そのダイアログ・ボックスを表示します。
2. 「File Name on Server」フィールドにテキスト・ファイルのフルパスおよびファイル名を入力します。
3. テキスト・ファイルの入力キャラクタ・セットを選択します。
4. 必要に応じて、デフォルトの特殊文字設定を変更します。
  - 「Separator Character」ドロップダウン・リストからデリミタ文字を選択します。
  - 「Quote Character」フィールドから識別文字列に使用する文字を選択します。
  - 「Comment Character」ドロップダウン・リストから識別コメント行に使用する文字を選択します。
5. テキスト・ファイルの最初の行に列名が含まれている場合は、「**Column Names in File**」チェック・ボックスを選択します。
6. 英数字以外の特殊文字を使用している場合は、「**Extended Character Set**」チェック・ボックスを選択します。
7. 入力ファイルから列名またはデータ型を「Field List」に自動で読み込む場合は、「Auto-Populate Field List」領域内の「**Column Names**」または「**Data Types**」（あるいはその両方）を選択します。
8. プランの実行時、ファイル内の列名およびデータ型でこの Transform の選択内容を上書きするには、「**Populate at Runtime**」チェック・ボックスを選択します。
9. データの数値書式がすでに正しく設定されていることを指定するには、「**Disable numeric format checking**」チェックボックスを選択します。
10. 日付書式を指定する場合は、「**Use Date Format**」チェック・ボックスを選択し、ドロップダウン・リストから日付書式を選択します。
11. 「Field List」に入力フィールドを追加するには、「**Add**」をクリックするか、または Col No 値をダブルクリックします。  
「Field Specification」ダイアログ・ボックスが表示されます。
12. 「Column Name」フィールドに、ターゲット・データベースで作成する列の名前を入力します。
13. 「Output Data Type」ドロップダウン・リストからこのフィールドの正しい出力タイプを選択します。

14. 出力列の特性を指定します。
15. 「OK」をクリックしてフィールドの指定を保存します。  
「Field List」に新しいエントリが表示されます。
16. 入力ファイルの各フィールドに対して、手順 11 ～ 15 を繰り返します。
17. (オプション) このダイアログ・ボックスの設定が、入力ファイルのデータにどのように影響するかを確認するには、クライアント・マシンでサンプル入力ファイルを作成し、「File Name」フィールドにそのファイルを指定します。このファイルは、Enterprise Link Server 上の入力ファイルと同じ特性である必要がありますが、必要なデータ量は一般的な量のみです。  
フィールド指定への変更は、同じサンプル入力ファイルに動的に適用されます。
18. 「OK」をクリックしてダイアログ・ボックスを閉じます。

## 「Delimited Text File Source」ダイアログ・ボックスの設定

Delimited Text File Source を定義するには、次の表を参照してください。

表 4: Delimited Text File Source: 「File Attribute」の設定

設定	説明
File Name on Server	デリミタ付きテキスト・ファイルの名前を入力します。このパスを Enterprise Link Server のローカル・パスとして指定します。
Character Set	文字のタイプをドロップダウン・リストから選択します。テキスト・ファイルが Windows のテキスト・エディタまたはキャラクタ・セットで作成されている場合は「ANSI」を選択します。そうでない場合は「OEM」を選択してキャラクタ・セットを ANSI 形式に変換します。
Separator Character	セパレータ文字のタイプをドロップダウン・リストから選択します。
Quote Character	引用符文字のタイプをドロップダウン・リストから選択します。Transform では、文字列値を表示する際に引用符文字を除去し、引用符文字の間にあるすべての文字（カンマを含む）が文字列値の要素として使用されます。
Comment Character	コメント文字のタイプをドロップダウン・リストから選択します。Transform では、行を表示する際にコメント行を除去します。
Column Names in File	ファイルに列名が含まれている場合はこのチェック・ボックスを選択します。最初の行に列名以外の値が含まれている場合は、このチェック・ボックスを選択しないでください。Transform によって列名が割り当てられます。ファイルに列名が存在しない場合は、Transform によって、Column1、Column2 などの名前が割り当てられます。
Extended Character Set	英数字以外の特殊文字を使用している場合は、このチェック・ボックスを選択します。

表 5: Delimited Text File Source: 「Field List」 の設定

設定	説明
Add	列の追加および仕様の設定を行う場合にクリックします。「Field Specification」ダイアログ・ボックスが表示されます。
Remove	列番号リストからアイテムを選択した後、このボタンをクリックします。列が削除されます。

図 2: 「Field Specification」ダイアログ・ボックス

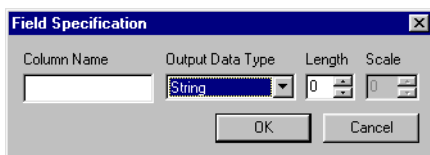


表 6: 「Field Specification」 の設定

設定	説明
Column Name	列名を入力します。
Output Data Type	出力データ型を選択します。
Length	文字列出力データの長さを指定します。
Precision	10 進数出力の桁数（整数および小数部分を含む）を指定します。桁数を多くすると、精度が高くなります。「Precision」の値は、38 以下にする必要があります。
Scale	固定小数点数または浮動小数点数内の小数点の位置を指定します。「Scale」の値は、「Precision」の値以下にする必要があります。

表 7: Delimited Text File Source: 「Sample Data」 の設定

設定	説明
File Name	ダイアログ・ボックスの下部に表示されているファイル名およびサンプル・データ表示を選択します。
Browse	これは、ファイルを検索するための参照に使用します。
Sample Data Display	この出力形式でデータが表示されます。

## Delimited Text File Source Transform に対するヒント

- この Transform は、デリミタ付きのクリーン・テキスト・ファイルからデータを選択する場合に使用することをお勧めします。選択したデータが複雑な場合は、Flat File Source Transform のほうがデータ管理のための設定がより多く提供されるため、Flat File Source Transform を使用します。
- Delimited Text File Source を使用して列の名前を変更または削除すると、プラン後半での余分なステップが不要になります。
- データ型を操作できます。たとえば、日付を **string** データ型として保持し、出力には **datetime** 型を選択できます。文字として入力し、整数として出力する数値を保持することができます。
- 「sample data」セクションから開始します。ファイルを検索し、列の設定を確認します。これらの設定が正しい場合は、「sample data」セクションのファイル名をコピーして上部に貼り付けます。列指定への変更は、同じサンプル表示に動的に適用されます。
- Delimited Text File Source では、空の **string** フィールドに NULL フィールド・フラグは設定されません。

## Delimited Text File Source の特性

ロード・サーバーまたはアクセス・サーバー	両方
スキーマの変更	実行する
マルチパスまたはシングルパス	シングルパス
コピーまたはインプレース	コピー
入力のソート	実行しない
プロパティの公開	一部



## Flat File Source

Flat File Source は、デリミタ付きの固定長バイナリ・フラット・ファイルから値を抽出する場合に使用します。File Transfer Protocol (FTP) ストリームによって、ファイル転送の完了を待機せずにリモート・システムでファイルを処理できます。Flat File Source Transform では、複数のレコード・タイプが処理されます。データ・フローを中断せずに、想定するデータ型と一致しない値を格納するための列を追加できます。追加された列で、エラーの確認および修正を行うことができます。

ここで説明する内容は次のとおりです。

- 「Flat File Source」ダイアログ・ボックスの設定
- Flat File Source Transform に対するヒント
- サポートされている入力データ型
- サポートされている出力データ型
- Flat File Source の特性

### 「Flat File Source」ダイアログ・ボックスの設定

Flat File Source Transform を定義するには、ダイアログ・ボックスを表示し、次のタブを設定します。

- 「File Select」タブ
- 「Attributes」タブ
- 「Processing」タブ
- 「Tags」タブ
- 「Columns」タブ
- 「Errors」タブ

**注意：** Flat File Sources が含まれているほとんどのプランでは、「File Select」、「Columns」および「Errors」タブにのみ設定を指定する必要があります。

## 「File Select」 タブ

ソース・ファイルを定義するには、「File Select」タブを選択し、表 8 に従って設定項目を指定します。

図 3: 「File Select」タブのダイアログ・ボックス

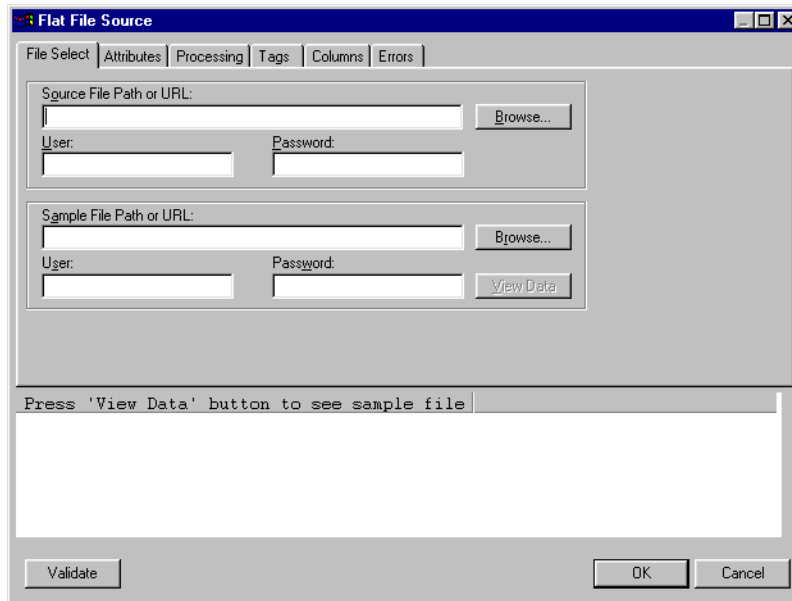


表 8: 「File Select」タブの設定

設定	説明
Source File Path or URL	Data Flow Service マシン上のソース・ファイルの場所を参照します。 このファイルが Data Flow Service マシンに存在しない場合は、このマシンに相対したフルパスを指定します。 ソース・ファイルが FTP または HTTP を介してアクセス可能なマシン上に存在する場合は、URL を使用できます。
User	ソース・ファイルが別のマシンに存在するか、またはアクセスが制限されている場合は、有効なユーザー名を入力してアクセスします。
Password	ソース・ファイルが別のマシンに存在するか、またはアクセスが制限されている場合は、有効なパスワードを入力してアクセスします。
Sample File Path or URL	Data Flow Service マシン上のサンプル・ファイルの場所を参照します。 このファイルが Data Flow Service マシンに存在しない場合は、このマシンに相対したフルパスを指定します。 ソース・ファイルが FTP または HTTP を介してアクセス可能なマシン上に存在する場合は、URL を使用できます。
User	サンプル・ファイルが別のマシンに存在するか、またはアクセスが制限されている場合は、有効なユーザー名を入力してアクセスします。

表 8: 「File Select」 タブの設定 (続き)

設定	説明
Password	サンプル・ファイルが別のマシンに存在するか、またはアクセスが制限されている場合は、有効なパスワードを入力してアクセスします。
View Data	このボタンは、サンプル・データ表示に読み込む場合にクリックします。フィールド指定への変更は、同じサンプル入力ファイルに動的に適用されます。
Validate	これは、ファイル形式の設定を検証する場合にクリックします。

## 「Attributes」タブ

使用しているソース・ファイルのタイプを定義するには、「Attributes」タブを選択し、表 9 に従って設定項目を指定します。

図 4: 「Attributes」タブのダイアログ・ボックス

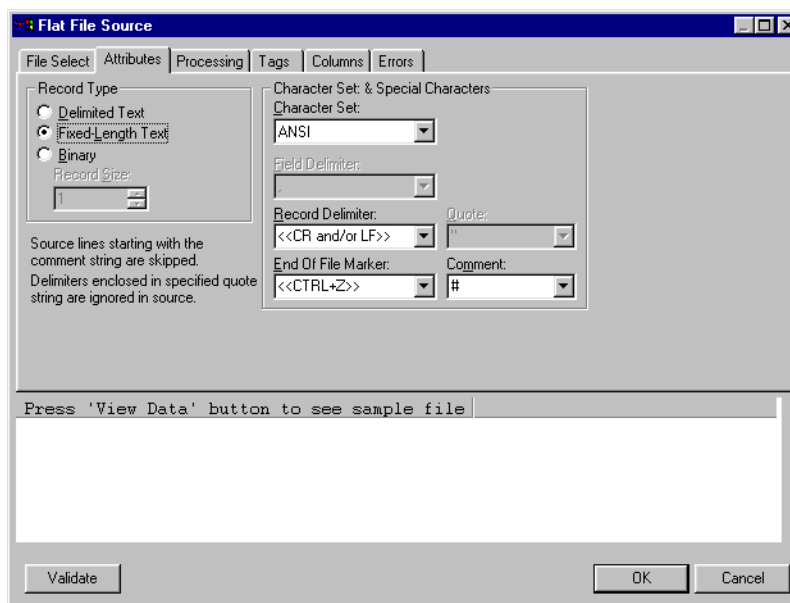


表 9: 「Attributes」タブの設定

設定	説明
Delimited Text	このオプションは、ソース・ファイルがデリミタ付きテキスト・ファイルの場合に選択します。
Fixed-Length Text	このオプションは、ソース・ファイルが固定長ファイルの場合に選択します。デフォルトでは、これが選択されています。
Binary	このオプションは、ソース・ファイルがバイナリ・ファイルの場合に選択します。
Record Size	バイナリをレコード・タイプとして選択した場合は、レコード・サイズを指定する必要があります。

表 9: 「Attributes」 タブの設定 (続き)

設定	説明
Character Set	ファイルで使用するキャラクタ・セットをドロップダウン・リストから選択します。デフォルト値は ANSI です。
Field Delimiter	デリミタ付きファイルを使用している場合は、フィールド・デリミタをドロップダウン・リストから選択します。デフォルト値はカンマです。
Record Delimiter	ファイルで使用するレコード・デリミタをドロップダウン・リストから選択します。デフォルト値は <<CR and/or LF>> です。
Quote	デリミタ付きファイルを使用している場合は、引用符マーカをドロップダウン・リストから選択します。デフォルト値は二重引用符です。
End of File Marker	ファイルで使用する、ファイルの末尾マーカをドロップダウン・リストから選択します。デフォルト値は <<CTRL+Z>> です。
Comment	ファイルで使用するコメント・マーカをドロップダウン・リストから選択します。デフォルト値は # です。コメント・マーカで囲まれている値は、プランの実行時に削除されます。

## 「Processing」 タブ

ソース・ファイルの読取り方法を指定するには、「Processing」タブを選択し、表 10 に従って設定項目を指定します。これらの設定は、IBM DB2 ユーザーにとって重要です。

図 5: 「Processing」 タブのダイアログ・ボックス

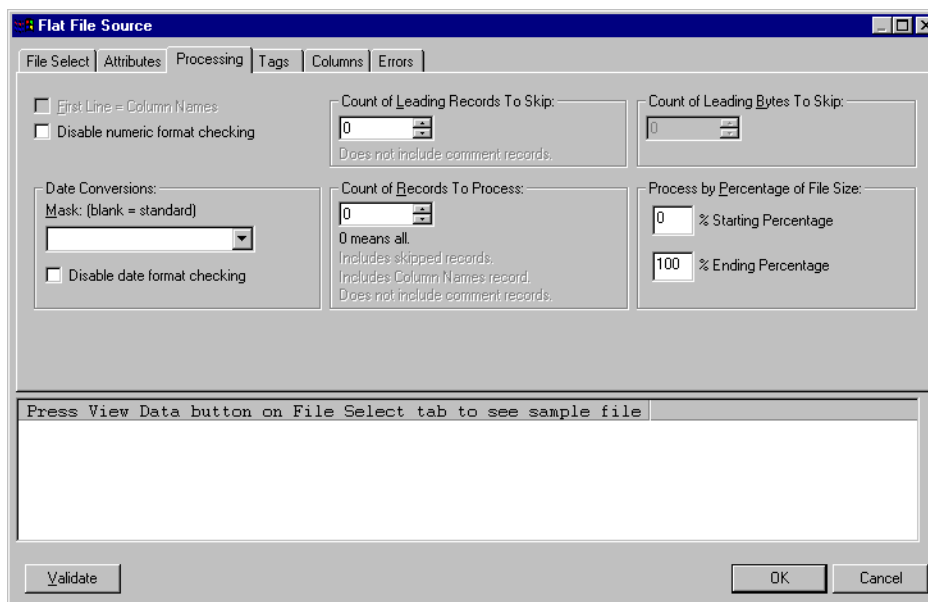


表 10: 「Processing」 タブの設定

設定	説明
First Line = Column Names	このチェック・ボックスは、デリミタ付きファイルを使用し、最初の行に列ヘッダーが含まれている場合に選択します。
Disable numeric format checking	データがすでに正しく書式設定されていることを指定する場合に選択します。重複した処理を行わないことによって、パフォーマンスを向上させることができます。
Mask	日付変換で使用する日付書式をドロップダウン・リストから選択するか、または入力します。
Disable date format checking	日付がすでに正しく書式設定されていることを指定する場合に選択します。余分な処理を行わないことによって、パフォーマンスを向上させることができます。
Count of Leading Records to Skip	このボックスは、スキップするレコードの数をソース・ファイルの先頭に指定する場合に使用します。数値を入力するか、または小さい矢印ボタンを使用して数値を指定することができます。 デフォルトでは、すべてのレコードが処理されます。 たとえば、プランで以前に 400 件のレコードを処理した後障害が発生した場合は、このフィールドに値 400 を指定して、エラーが発生した時点で実行されていたプランを開始することによってトラブルシューティングを実行できます。
Count of Records to Process	このフィールドは、処理するレコードの数を指定する場合に使用します。これによって、データ・フロー内のソース・ファイル・レコードのサブセットのみを処理することができます。数値を入力するか、または小さい矢印ボタンを使用して数値を指定することができます。 デフォルトの 0 (ゼロ) を使用すると、すべてのレコードが処理されます。
Count of Leading Bytes to Skip	バイナリをレコード・タイプとして選択した場合は、別の位置をファイルの開始位置として指定できます。
Starting Percentage	ファイルの一部を処理する場合、開始時の割合 (パーセント) を示します。
Ending Percentage	ファイルの一部を処理する場合、終了時の割合 (パーセント) を示します。

## 「Tags」 タブ

ソース・ファイルで使用するタグを定義するには、「Tags」タブを選択し、表 11 に従って設定項目を指定します。

**注意：** タグは、特定のレコード・タイプを識別する名前付きラベルです。データ・フローでタグを使用するには、まず、ソース・ファイルで入力レコードをタグ付けする必要があります。複数のレコード・タイプが含まれているファイルでは、各レコード・タイプにタグを定義する必要があります。

図 6: 「Tags」タブのダイアログ・ボックス

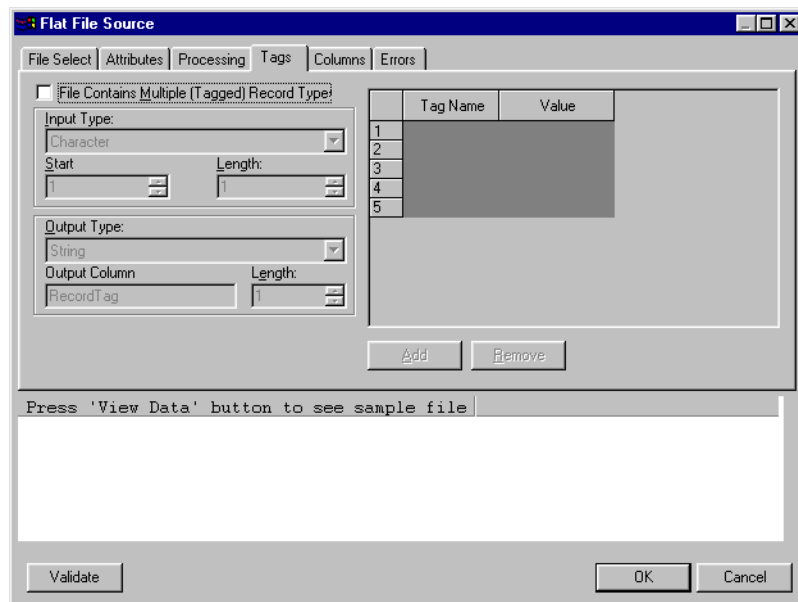


表 11: 「Tags」タブの設定

設定	説明
File Contains Multiple (Tagged) Record Type	このチェック・ボックスは、ファイルに複数のレコード・タイプが含まれている場合に選択します。
Input Type	ソース・ファイル内のタグ・フィールドの入力データ型をドロップダウン・リストから選択します。
Start	ソース・ファイル内のタグ・フィールドの開始位置を指定します。
Length	ソース・ファイル内のタグ・フィールドの長さを指定します。
Output Type	互換性のある出力データ型をドロップダウン・リストから選択します。
Output Column	出力列の名前を指定します。デフォルト値は Record Tag です。プランの実行後、列にタグ値が格納されます。
Length	出力データ型として string を選択した場合は、出力列の長さを指定する必要があります。

表 11: 「Tags」 タブの設定 (続き)

設定	説明
Tag Name	タグのラベルを入力します。このラベルは、説明のためにのみ使用されます。このラベルには、任意の文字列を使用できません。
Value	ソース・レコードのタグの値を入力します。たとえば、顧客名レコードに対してコード 01 を使用します。
Add	このボタンは、別のタグ行を追加する場合にクリックします。
Remove	このボタンは、タグ行を削除する場合にクリックします。

## 「Columns」 タブ

列を定義するには、「Columns」タブを選択し、表 12 に従って設定項目を指定します。

**注意：** プランで最後にバッチ・ローダーが実行され、位置に基づいてデータをロードしている場合は、ターゲット表と一致するようにフィールドを並べ替えます。列および行を並べ替えるには、それらをクリックして新しい位置にドラッグします。

図 7: 「Columns」 タブのダイアログ・ボックス

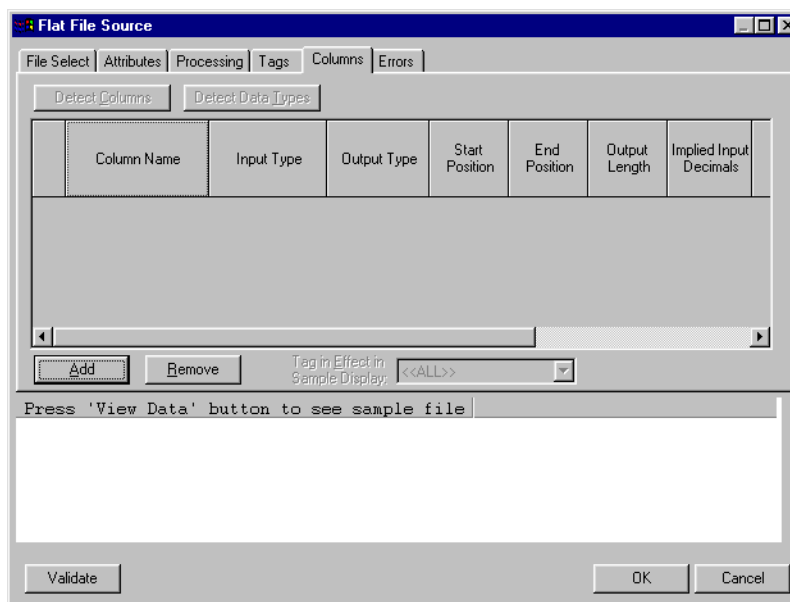


表 12: 「Columns」 タブの設定

設定	説明
Detect Columns	入力ファイルがデリミタ付きファイルで、「Processing」タブの「First Line=Column Names」チェック・ボックスが選択されている場合は、このボタンをクリックしてその入力ファイルから列名を検出します。
Detect Data Types	入力ファイルがデリミタ付きファイルで、「Processing」タブの「First Line=Column Names」チェック・ボックスが選択されている場合は、このボタンをクリックしてその入力ファイルからデータ型を検出します。
Add	このボタンは、別の列行を追加する場合にクリックします。
Remove	列を削除するには、その列を選択し、このボタンをクリックします。
Column Name	ソース・フィールドの列ラベルを入力します。データ列の検出を選択すると、これらの名前が自動的に読み込まれます。
Input Type	入力列のデータ型を選択します。
Output Type	出力列のデータ型を選択します。入力タイプと互換性がある必要があります。
Start Position	フィールドの開始位置を入力します。
End Position	フィールドの終了位置を入力します。
Output Length	出力列のバイト数 (string フィールドの場合) または合計桁数 (decimal フィールドの場合) を入力します。
Implied Input Decimals	入力列の小数点より右側の 10 進数の桁数 (decimal 型の列の場合) を入力します。
Output Decimal Digits	出力列の小数点より右側の 10 進数の桁数 (decimal 型の列の場合) を入力します。
Error Override Value	無効な値が検出され、「Errors」タブの「Use Override Values」が選択されている場合は、置換値を入力または選択します。また、データ・フロー内の元の値の保存も指定することができます。これは、エラーを処理する場合に有効です。
Blanks as Null	プランの実行時に空白文字を NULL 値として解釈するためのフラグです。これは、エラーを処理する場合に有効です。
Required Field	データ・フロー内の該当するフィールドに値を指定する必要があることを示すフラグです。この列のフィールドに NULL 値を指定すると、プランの実行時にレコードがエラーとして処理されます。 このフィールドが確認された場合、NULL 値およびエラーが検出されます。
Import	データ・フロー内の列を処理するかどうかを決定するためのフラグです。 パフォーマンスを向上させるには、インポートする必要がある列のみを選択します。
Tag in effect in Sample Display	固定長ファイルおよびバイナリ・ファイルでは、複数のレコード・タイプが「Tags」タブに定義されている場合、「Record Types Column」内のドロップダウン・リストからタグを選択できます。レコードは、各タグ値に対して個別の列を持つタグで区切られています。



## 「Errors」タブ

拒否されたレコードおよびエラーが発生したレコードを処理する方法を指定するには、「Errors」タブを選択し、表 13 に従って設定項目を指定します。

図 8: 「Errors」タブのダイアログ・ボックス

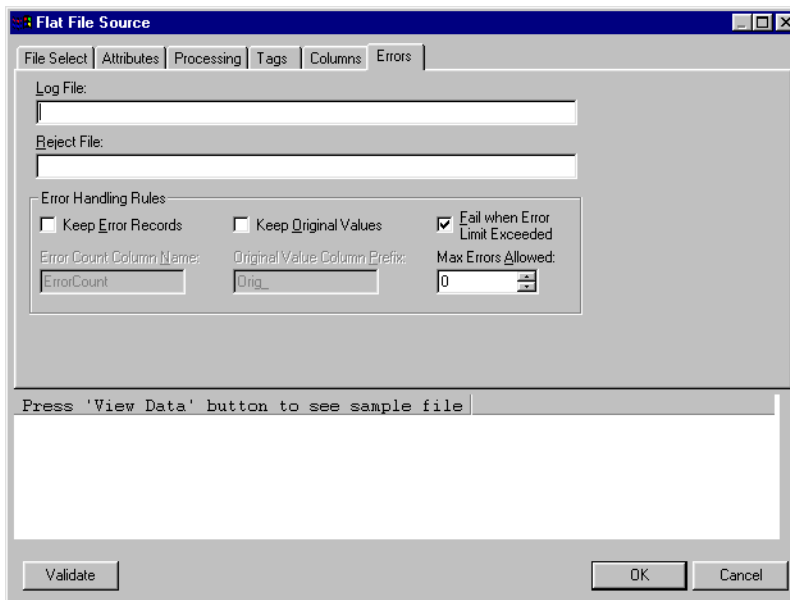


表 13: 「Errors」タブの設定

設定	説明
Log File	Data Flow Service マシン上のエラー・ログ・ファイルのパスおよび名前を入力します。このファイルが Data Flow Service マシンに存在しない場合は、このマシンに相対したフルパスを指定します。 このファイルには、プランの実行時に生成されたエラー・メッセージおよび説明が含まれています。指定した場所にこの名前を持つファイルがすでに存在する場合は、上書きされます。
Reject File	拒否された入力レコードを格納するためのファイルのパスおよび名前を入力します。このファイルが Data Flow Service マシンに存在しない場合は、このマシンに相対したフルパスを指定します。 このファイルには、エラーが含まれているため拒否されたレコードが含まれています。フィールドに1つのみの空白または NULL 値が含まれている場合は、NULL に設定されます。

表 13: 「Errors」 タブの設定 (続き)

設定	説明
Keep Error Record	<p>このチェック・ボックスは、データ・フローにエラー・レコードを含める場合に選択します。</p> <p>このオプションを選択すると、エラーが含まれているレコードが拒否ファイルに格納され、データ・フロー内に保存されません。無効なフィールド値のかわりに、上書き値が使用されます。各レコードの末尾に、エラー・カウントを示す number フィールドが追加されます。</p>
Error Count Column Name	<p>エラー・カウントのための列名を指定します。</p> <p>プランの実行後、この列には、各レコードの無効なフィールドの合計数が含まれます。デフォルト名は ErrorCount です。エラー・カウント列フィールドに値が存在しない場合、エラー・カウント列は作成されません。</p>
Keep Original Values	<p>このチェック・ボックスは、元の入力形式でデータ・フローにフィールド値を保存する場合に選択します。</p> <p>このオプションを選択すると、元の値が個別の列に格納され、データ・フロー内に保存されます。また、処理済の値もデータ・フローに存在するため、出力列の数は 2 倍になります。</p> <p>このオプションは、プラン内の別の Transform (Expression Calculator など) で拡張ロジックを使用してエラーを処理する場合に有効です。</p> <p>元の値は、文字列フィールドとしてデータ・フロー内に格納され、出力データ型には変換されません。16 進文字の 0 (ゼロ) が含まれているエラー値は、データ・フローから破棄されません。</p>
Original Value Column Prefix	<p>「Keep Original Values」チェック・ボックスが選択されている場合は、元の入力値を使用して新しい列の名前を指定するための接頭辞を指定できます。</p> <p>デフォルトの接頭辞は Orig_ です。</p> <p>たとえば、プランの実行時に「Address」列でエラーが検出されると、アドレスが Orig_Address という名前の新しい列に格納され、「Address」の値が上書き値に置換されます。</p>
Fail when Error Limit Exceeded	<p>このチェック・ボックスは、「Max Errors Allowed」フィールドに設定されているエラー制限に達した場合にプランを強制終了するために選択します。</p>
Max Errors Allowed	<p>プランを強制終了する前にプランで許容可能なエラーの最大数を指定します。</p> <p>各エラーのメッセージは、Data Flow Service マシン上のエラー・ログ・ファイルに格納されています。</p> <p>たとえば、プランを停止するまでに 5 つのエラーを許容する場合は、値 6 を指定します。</p> <p>デフォルト値の 0 (ゼロ) は、プランがエラーに関係なく実行されることを意味します。</p>

## Flat File Source Transform に対するヒント

- ファイルの場所にアクセスするための読取りおよび書込み権限を持っているユーザー名を指定する必要があります。
- Flat File Source Transform の「Columns」タブ内のグリッドの並替えは、出力に悪影響を及ぼすため、行わないことをお勧めします。
- Flat File Source Transform では、Solaris float または Solaris double データ型のバイナリ・データはサポートされていません。
- Flat File Source Transform を固定長ファイルとともに使用すると、「Reject File」に追加される出力レコードは、元の入力ファイル内のレコードの形式と同じではなくなります。
- ソース・ファイルに日付が含まれている場合、最適なパフォーマンスを実現するために日付マスクを使用します。
- Flat File Source では、空の文字列フィールドに NULL フィールド・フラグが設定されます。
- Process by Percent of File 機能を使用すると、同じファイルに対する作業を 2 つ以上の Flat File Source Transform に割り当てることによって大規模ファイルの平行処理を実行できます。受信データを単一ストリームのデータにマージするには、Union Transform を使用します。ファイルを二等分して平行処理するには、2 つの Flat File Source Transform をデータ・フローに追加します。1 つ目の Transform では、0% を開始時の割合（パーセント）、50% を終了時の割合（パーセント）として指定します。2 つ目では、50% および 100% を指定します。1 つ目の Transform の終了値は、レコードがスキップされるように、2 つ目の Transform の開始値と重なっている必要があります。

## サポートされている入力データ型

入力データでは、次のデータ型がサポートされています。

表 14: 入力データでサポートされているデータ型

データ型	説明
Character	可変長シーケンスの文字。
Character-Trim	可変長シーケンスの文字。すべての後続空白が削除されます。
Zoned Decimal	可変長シーケンスの 10 進文字 (0-9)。最後の文字の上位 4 ビットは、数値の記号をエンコードするために次のように変更されます。 EBCDIC ファイルの場合、16 進数の C または F は正の値に使用され、16 進数の D は負の値に使用されます。 ASCII ファイルの場合、16 進数の 3 または F は正の値に使用され、16 進数の D は負の値に使用されます。

バイナリ・ファイルの入力データ型の詳細は、[154 ページ](#)の「サポートされているバイナリ入力データ型」を参照してください。

## サポートされている出力データ型

出力データでは、次の列型がサポートされています。

表 15: 出力データでサポートされている列型

データ型	説明	制限
DateTime	日時値	なし
Decimal	記号および小数点付きの 10 進数文字列	decimal フィールドの出力の長さは、38 を超えて指定することはできません。 出力 10 進数の値は、出力の長さの値を超えて指定することはできません。
Double	実数値（小数を含む）	最大 $1.8 \times 10^{23}$
Integer	整数値	~ +/- 2,000,000,000
Long Integer	整数値	~ +/- 2,000,000,000
String	英数値	なし
Unsigned Integer	正の整数値	4,000,000,000 未満

## Flat File Source の特性

パッケージ・タイプ	Basic
スキーマの変更	実行する
マルチパスまたはシングルパス	シングルパス
コピーまたはインプレース	コピー
入力のソート	実行しない
プロパティの公開	一部

## OLEDB Source

OLEDB Source Transform は、OLE DB プロバイダとの接続に使用します。OLEDB Source Transform では、OLE DB プロバイダがデータ・フロー内のデータ・ソースとして使用されます。OLEDB Source Transform は、OLE DB または OLAP インタフェース用の OLE DB が提供されているソースにアクセスします。この Transform では、他のソースからデータを取得するための MDX および SQL コマンドの両方がサポートされています。MDX は、OLAP 用の OLE DB の多次元問合せ構文で、OLAP データ・ソースを問い合わせるための SQL 言語の上に定義されているマクロで構成されています。OLEDB Source Transform では、いくつかのタイプの単純なプロバイダまたは複雑なプロバイダを使用できます。

図 9: 「OLEDB Source」 ダイアログ・ボックス

The screenshot shows a dialog box titled 'OLEDB Source'. It has the following fields:

- Provider:** A dropdown menu.
- Server:** A text input field.
- Database/Catalog:** A text input field.
- User Name:** A text input field.
- Password:** A text input field.
- Command:** A large text area for entering commands.
- Buttons:** 'OK' and 'Cancel' buttons at the bottom right.

ここで説明する内容は次のとおりです。

- 「OLEDB Source」 ダイアログ・ボックスの設定
- OLEDB Source に対するヒント
- OLEDB Source の特性

## 「OLEDB Source」 ダイアログ・ボックスの設定

OLEDB Source を定義するには、「OLEDB Source」 ダイアログ・ボックスを表示し、次の表を使用します。

表 16: 「OLEDB Source」 の設定

設定	説明
Provider	プロバイダをドロップダウン・リストから選択します。プロバイダが Oracle BAM Enterprise Link Server にのみインストールされている場合は、このフィールドにそのプロバイダの名前を入力します。
Server	アクセスするサーバーの名前を入力します。
Database/Catalog	アクセスするデータベースの名前を指定します。
User	ユーザー名を入力します。
Password	パスワードを入力します。
Command	実行する MDX または SQL コマンドを入力します。

## OLEDB Source に対するヒント

- プロバイダ名をダイアログ・ボックスのドロップダウン・リストに表示するには、OLE DB プロバイダをクライアント・マシンおよび Oracle BAM Enterprise Link Server の両方にインストールする必要があります。プロバイダが Oracle BAM Enterprise Link Server にのみインストールされている場合は、ダイアログ・ボックスでそのプロバイダの名前を入力する必要があります。メタデータが含まれているサーバーをプロバイダとして使用するには、そのサーバーを実行している必要があります。他のプロバイダの例としては、SQL Server 用の SQLOLEDB、ODBC 接続用の MSDASQL などがあります。
- ユーザー入力の SQL と同様に文が処理されるため、データ・フローで OLEDB Source Transform の後に続く Transform 内の列名は表示されません。データ・フローで追加の処理を行うために他の Transform 内の列名を入力できます。
- 次に、単純な MDX の例を示します。
 

```
select{[Measures].[Unit Sales]} on columns,
order(except([Promotion Media].[Media Type].members,{[Promotion Media].[Media Type].[No Media]}),[Measures].[Unit Sales],DESC) on rows from Sales
```

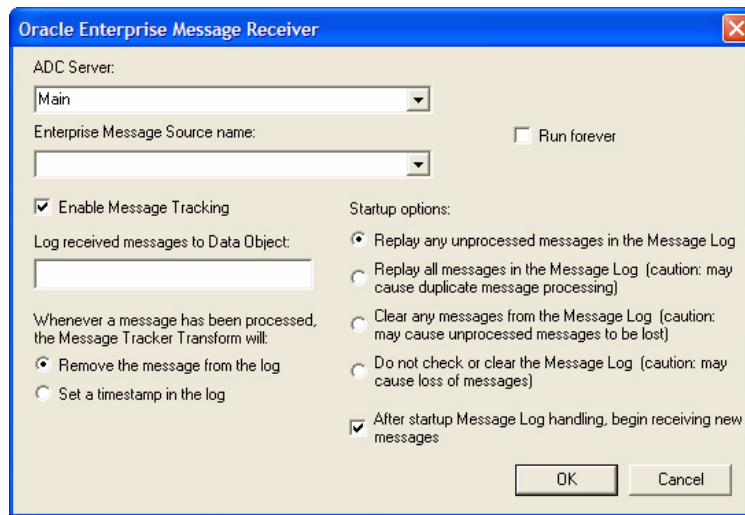
## OLEDB Source の特性

パッケージ・タイプ	Basic
スキーマの変更	実行しない
マルチパスまたはシングルパス	シングルパス
コピーまたはインプレース	コピー
入力のソート	実行しない
プロパティの公開	一部

## Oracle BAM Enterprise Message Receiver

Oracle BAM Transform では、エンタープライズ・メッセージ・ソースへの接続および ADC へのデータのロードを行う場合に特定の機能が実行されます。Oracle BAM Message Receiver Transform が含まれているプランは、エンタープライズ・メッセージ・ソースからのリアルタイム情報のフローの処理が継続して実行されるように設定できます。これらのタイプのプランのステータスを監視するには、プランの監視を使用します。

図 10: 「Oracle BAM Enterprise Message Receiver」 ダイアログ・ボックス



ここで説明する内容は次のとおりです。

- 「Oracle BAM Message Receiver」 ダイアログ・ボックスの設定
- Oracle BAM Message Receiver の特性

## 「Oracle BAM Message Receiver」 ダイアログ・ボックスの設定

Oracle BAM Message Receiver を定義するには、「Oracle BAM Message Receiver」 ダイアログ・ボックスを表示し、次の表を使用します。

表 17: Oracle BAM Message Receiver の設定

設定	説明
ADC Server Name	ADC サーバー名を入力します。デフォルト値は Main です。
Enterprise Message Source	使用するエンタープライズ・メッセージ・ソースの名前をクリックします。エンタープライズ・メッセージ・ソースは、Oracle BAM Architect によって定義されています。
Run Forever	「Run Forever」チェック・ボックスは、このプランを継続して実行する場合に選択します。ライブ・データをデータ・オブジェクトに継続してロードするには、このオプションを選択する必要があります。
Enable Message Logging	「Enable Message Logging」チェック・ボックスは、メッセージ・キューから ADC への保証されたメッセージ配信に対するメッセージ・ロギングを有効にする場合に選択します。
Data Object	メッセージ・ロギングに使用するデータ・オブジェクトの名前を入力します。

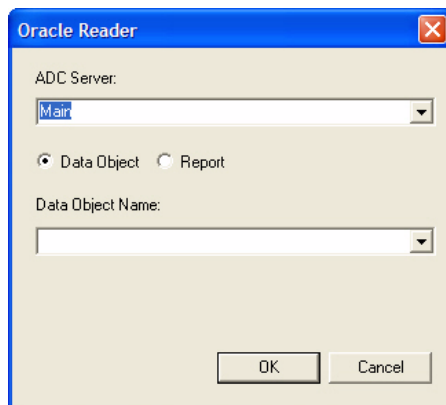
## Oracle BAM Message Receiver の特性

パッケージ・タイプ	Basic
スキーマの変更	実行しない
マルチパスまたはシングルパス	シングルパス
コピーまたはインプレース	コピー
入力のソート	実行しない
プロパティの公開	一部

## Oracle BAM Reader

Oracle BAM Reader Transform は、データ・オブジェクトからデータを読み取り、データ・フロー内の次のステップにレコードを送信するソース・ステップです。

図 11: 「Oracle BAM Reader」 ダイアログ・ボックス



ここで説明する内容は次のとおりです。

- 「Oracle BAM Reader」 ダイアログ・ボックスの設定
- Oracle BAM Reader の特性



## 「Oracle BAM Reader」ダイアログ・ボックスの設定

Oracle BAM Reader を定義するには、「Oracle BAM Reader」ダイアログ・ボックスを表示し、次の表を使用します。

表 18: 「Oracle BAM Reader」の設定

設定	説明
ACD Server Name	ADC サーバー名を入力します。デフォルト値は Main です。
Data Object	使用するデータ・オブジェクトの名前を選択します。

## Oracle BAM Reader の特性

パッケージ・タイプ	Basic
スキーマの変更	実行しない
マルチパスまたはシングルパス	シングルパス
コピーまたはインプレース	コピー
入力のソート	実行しない
プロパティの公開	一部

## SQL Query

SQL Query は、Part をワークスペースまたはデータ・フロー・エディタにドラッグして SQL 文を作成する場合に使用します。SQL の SELECT 文を直接入力または編集するには、「SQL Editor」ダイアログ・ボックスを使用します。

プランには複数のデータ・フローを含めることができ、それぞれが SQL Query ソースを持ちます。Part を Parts Bin からワークスペースにドラッグすると、SQL Query および Grid で構成されているデータ・フローが自動的に作成されます。

ここで説明する内容は次のとおりです。

- [SQL Query の設定](#)
- [SQL エディタの使用](#)
- [SQL Query Transform に対するヒント](#)
- [SQL Query の特性](#)

## SQL Query の設定

SQL Query の設定を定義するには、表 19 に従って設定項目を指定します。

図 12: 「SQL Query」 ダイアログ・ボックス

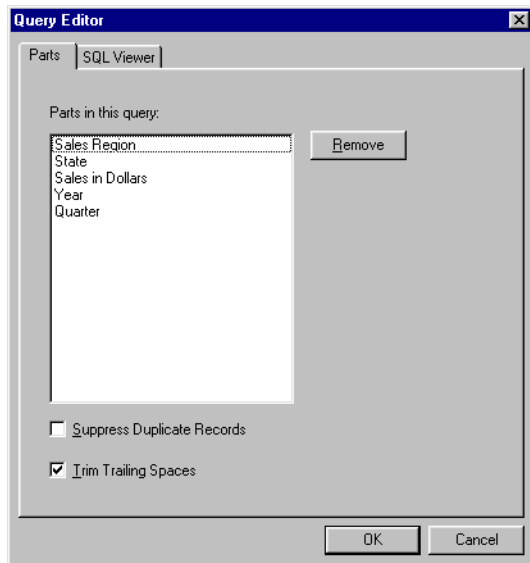


表 19: 「Parts」 タブの設定

設定	説明
Parts in this query	SQL 文に存在する、プランに追加した Part を示します。
Remove	SQL 文から Part を削除する場合に、その Part を選択してこのボタンをクリックします。
Suppress Duplicate Records	このチェック・ボックスは、結果にレコードが重複して表示されないようにする場合に選択します。結果には、選択したすべての Part の値がそれぞれ 1 回のみ表示されます。
Trim Trailing Spaces	このチェック・ボックスは、後続の空白を結果から削除する場合に選択します。

「SQL Viewer」タブでは、SQL 文を表示できます。SQL を手動で変更するには、SQL エディタを使用する必要があります。

## SQL エディタの使用

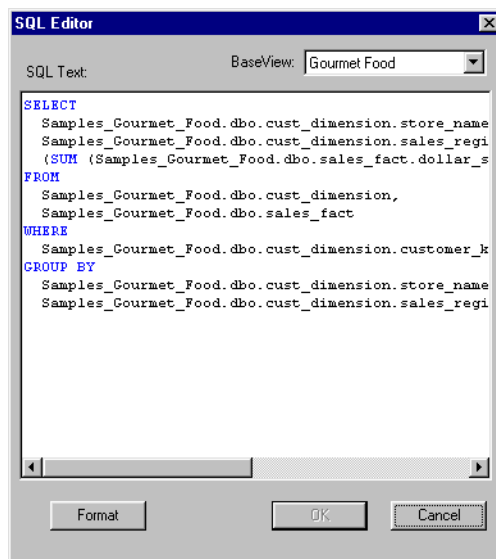
SQL エディタは、SQL 文の作成および編集に使用します。クエリー・エディタは、SQL 文の表示に使用します。使用している RDBMS サーバー・ソフトウェアでサポートされている「SQL Editor」ダイアログ・ボックスで SELECT 文を入力します。

**注意：** 編集した SQL は、Part をワークスペースにドラッグして作成した SQL 文と組み合わせることはできません。SQL 文を手動で編集した場合、Part をワークスペースにドラッグすると、新しい SQL Query が作成されます。

SQL 文を入力または編集するには、次の手順を実行します。

1. 「SQL Query」を右クリックし、「SQL Editor」を選択します。  
「SQL Editor」ダイアログ・ボックスが表示されます。

図 13: 「SQL Editor」ダイアログ・ボックス



2. 問い合わせる BaseView を「BaseView」ドロップダウン・リストから選択します。

SQL 文を編集している場合は、BaseView がすでに選択されています。

3. SQL テキストを入力または編集します。

アイテムを Parts Bin からワークスペースにドラッグしてプランを作成すると、このプランによって生成された SQL 文がダイアログ・ボックスに表示されます。

4. 「Format」をクリックして、SQL 文の表示を構造化します。  
SQL の各キーワードが単一行に青字で配置され、読みやすくなるように SQL テキストが書式設定されます。
5. 「OK」をクリックしてダイアログ・ボックスを閉じ、変更を保存します。  
アイテムを Parts Bin からドラッグして問合せを作成した場合、これらの列の表示はワークスペースから消去されます。

## SQL Query Transform に対するヒント

- 2つの SQL Query を結合する場合は、両方の「Trim Trailing Spaces」および「Suppress Duplicate Records」に同じ設定を選択する必要があります。
- 各 SQL Query は、単一の BaseView を対象としています。データを2つの BaseView から選択する場合は、別の SQL Query をプランに追加します。
- 複数の SQL Query が含まれているプランに Part を追加すると、ダイアログ・ボックスで、それらの Part が追加される SQL Query の選択を求められます。SQL Query の各ステップを区別するには、Part をワークスペースにドラッグする前にステップの名前を変更します。
- 列は、Grid から削除しても、SQL Query の SELECT 文からは削除されません。SQL Query から Part を削除するには、「Parts」タブでその Part を選択し、「Remove」をクリックします。これによって、SQL の SELECT 文から列が削除されます。

## SQL Query の特性

パッケージ・タイプ	Basic
スキーマの変更	実行する
マルチパスまたはシングルパス	シングルパス
コピーまたはインプレース	コピー
入力のソート	N/A
プロパティの公開	すべて

# Time Generation

Time Generation は、時間ディメンション表に移入する場合に使用します。時間ディメンション表によって、複雑な SQL 計算を使用せずに時間データを分析することが可能になります。時間ディメンション表は、生成されたデータのみが移入されるため、他のディメンションとは異なります。

図 14: 「Time Generation」ダイアログ・ボックス

ここで説明する内容は次のとおりです。

- 「Time Generation」ダイアログ・ボックスの設定
- Time Generation Transform に対するヒント
- Time Generation の特性

## 「Time Generation」ダイアログ・ボックスの設定

「Time Generation」ダイアログ・ボックスを定義するには、次の表を参照してください。

表 20: 「Time Generation」の「Input」の設定

設定	説明
Start Date	これを選択した場合は、開始日を入力する必要があります。
Use Current Date	これを選択した場合は、現在の日付が開始日として使用されません。
Duration	生成する時間ディメンションの存続期間（日数）を指定します。

表 21: 「Time Generation」の「Output」の設定

出力設定	説明	データ型
Day of the Month	月単位での日付を表す数値	int
Weekday Number	週単位での日付を表す数値	int
Weekday Name	曜日	char
Day of the Year	通年での日付を表す数値	int
Julian Day	ユリウス暦に基づいた日付を表す数値	int
Weekday	平日フラグ	char
Weekend	週末フラグ	char
Week of the Year	通年での週を表す数値	int
Julian Week	ユリウス暦に基づいた週を表す数値	int
Month Number	通年での月を表す数値	int
Quarter	通年での四半期を表す数値	int
Month Name	月名	char
Year	4文字の年数	int
Julian Year	ユリウス暦に基づいた年数	int
Date	MMDDYYYY 形式の日付	datetime

## Time Generation Transform に対するヒント

- 「Julian day」の出力列名を day\_key に設定します。
- 予約語との問題が発生しないように「Date」の出力列名を変更します。
- 入力設定によって、開始日および生成するレコードの数が指定されます。出力設定によって、既存の時間ディメンション表にロードするデータの型が指定されます。
- 時間ディメンション表に生成するレコードの開始日を指定します。Data Flow Service マシンのシステム日付を開始日として使用するには、「Use Current Date」をクリックします。
- 「Duration」に、生成するレコードの数を入力します。存続期間の値として入力したそれぞれの日にレコードが生成されます。「Duration」に大きい値を指定すると、時間ディメンション表がリフレッシュされるまでの間隔が長くなります。
- 「Output」では、「Date」および「Julian Day」出力タイプを選択し、ターゲットにこれらの列の名前を入力します。時間ディメンション表には、これらの2つの列が含まれている必要があります。時間ディメンション表の主キーは「Julian Day」列に対して作成され、一意の日付値は「Date」列の各レコードに対して生成されます。「Julian Day」列は integer 型で、「Date」は datetime 型です。ユリウス日付値は、西暦 1,4713 年 B.C.1 月から始まります。

- ユーザーがターゲット・データベースのファクトのサマリーを作成する場合に使用する出力列タイプを選択する必要があります。たとえば、年ごとの合計売上を取得するためにプランを実行する場合は「Year」を選択します。
- 列名は、空白を使用せずに入力します。
- このプランを実行すると、指定したすべての出力列の値が時間ディメンション表に移入されます。「Julian Day」列内の値に基づいて、連続整数キーも生成されます。

## Time Generation の特性

パッケージ・タイプ	Basic
スキーマの変更	実行する
マルチパスまたはシングルパス	シングルパス
コピーまたはインプレース	コピー
入力のソート	実行しない
プロパティの公開	一部

## VBScriptSource

VBScript Source は、外部アプリケーションから入力データを読み取る場合に使用します。ソース・アプリケーションは、OLE 準拠で、Data Flow Service が実行されているクライアント・コンピュータとシステムの両方にインストールされている必要があります。

VBScriptSource Transform に設定項目はありません。ダイアログ・ボックスには、テキスト・フィールドがあります。ここに VBScript を入力して「OK」ボタンをクリックし、ダイアログ・ボックスを閉じて変更を保存します。

詳細は、『Oracle BAM Enterprise Link Design Studio ユーザーズ・ガイド』または Oracle BAM Enterprise Link のオンライン・ヘルプを参照してください。

VBScript Transform を使用してバイナリ・データ型の列は作成できません。

パッケージ・タイプ	Basic
スキーマの変更	実行する
マルチパスまたはシングルパス	シングルパス
コピーまたはインプレース	コピー
入力のソート	N/A
プロパティの公開	なし

## XML Source

XML Source Transform を使用すると、XML 文書を Oracle BAM Enterprise Link にインポートできます。

ここで説明する内容は次のとおりです。

- [XML Source Transform の設定](#)
- [XML Source の特性](#)

### XML Source Transform の使用例

図 15 内のコードでは、GroceryList ごとに任意の数の Food 要素、および Food ごとに任意の数の Item および Type を指定できます。また、GroceryList には Drink 要素も含まれています。各 Drink 要素に Type および Item が含まれています。

図 15: サンプル XML コード

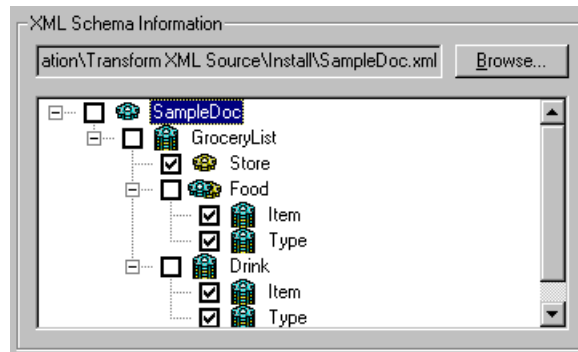
```
<?xml version="1.0" encoding="UTF-8" ?>
<SampleDoc>
<GroceryList Store="A">
<Food>
  <Item>Pepper</Item>
  <Item>Apple</Item>
  <Type>Red</Type>
  <Type>Green</Type>
  <Type>Yellow</Type>
</Food>
<Food>
  <Type>Mint Chip</Type>
  <Item>Ice Cream</Item>
</Food>
<Drink>
  <Item>Juice</Item>
  <Type>Grape</Type>
</Drink>
</GroceryList>
</SampleDoc>
```

次の使用例では、GroceryList または Food 要素の列は生成されません。これらの列に格納されるデータが含まれていないためです。また、Drink を無視することもできます。Food の両方の子 (Item および Type) は出力されます。



図 16 では、Food および Item の両方の要素が最大出現数 2 でフラット化されています。また、Type 要素に最大出現数 1 が指定されていることに注意してください。

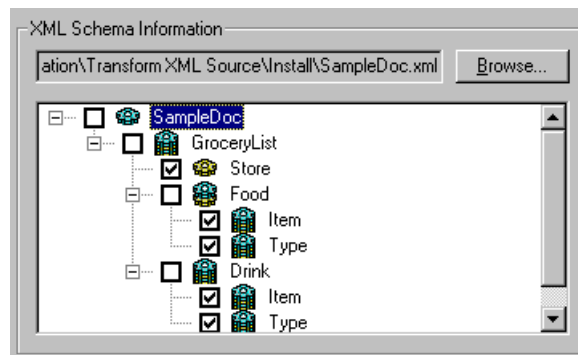
図 16: Food および Item 要素がフラット化されている場合 (ダイアログ・ボックスおよび出力)



	Store	Food1_Item1	Food1_Item2	Food1_Type	Food2_Item1	Food2_Item2	Food2_Type	Item	Type
1	A	Pepper	Apple	Red	Ice Cream		Mint Chip	Juice	Grape

要素がフラット化されていない場合、その親要素内にその要素が出現するたびに新しいレコードが生成されます。要素はフラット化されず、最大出現数の設定はそのままになるように前述の使用例を変更すると、図 17 に示す出力が生成されます。

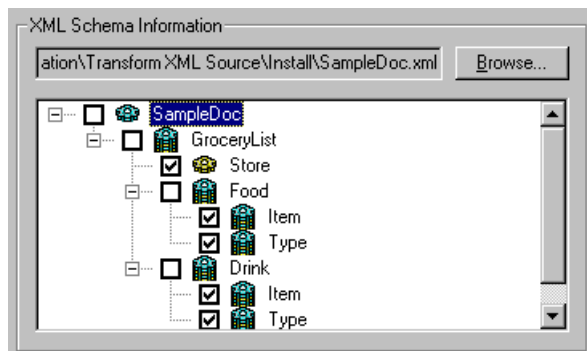
図 17: 要素がフラット化されていない場合 (ダイアログ・ボックスおよび出力)



	Store	Item	Type	Item	Type
1	A	Pepper	Red	Juice	Grape
2	A	Apple	Red	Juice	Grape
3	A	Ice Cream	Mint Chip	Juice	Grape

前述の使用例を編集してすべての要素を無制限に出現可能にすると、[図 18](#)に示す出力が生成されます。出力レコードは、各要素のすべての可能な組合せに対して生成されることに注意してください。

図 18: すべての要素が無制限に出現する場合（ダイアログ・ボックスおよび出力）



	Store	Item	Type	Item	Type
1	A	Pepper	Red	Juice	Grape
2	A	Apple	Red	Juice	Grape
3	A	Pepper	Green	Juice	Grape
4	A	Apple	Green	Juice	Grape
5	A	Pepper	Yellow	Juice	Grape
6	A	Apple	Yellow	Juice	Grape
7	A	Ice Cream	Mint Chip	Juice	Grape

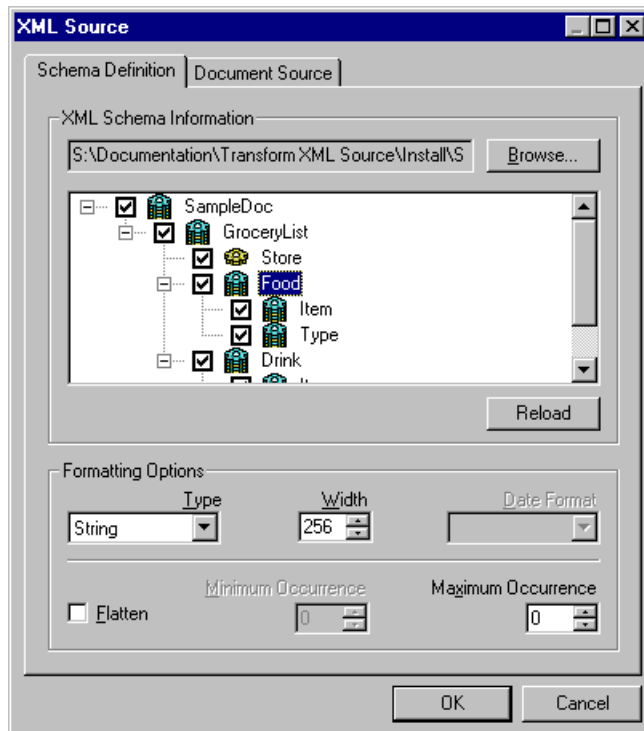
## XML Source Transform の設定

「XML Source」ダイアログ・ボックスには、次のタブが含まれています。

- Schema Definition
- Document Source

## 「Schema Definition」 タブ

図 19: 「Schema Definition」 タブ








「Schema Definition」 タブには、次のセクションが含まれています。

- XML Schema Information
- Formatting Options

### 「XML Schema Information」 セクション

このセクションでは、要素およびそれら相互の階層関係を表す次のアイコンが含まれているツリーが使用されます。

-  最大出現数 0 (無制限の出現数)
-  最大出現数 2 以上
-  最大出現数 1
-  フラット化されている要素 (最大出現数 2 以上が指定されている)
-  実際は XML 属性

要素チェック・ボックスが選択されている場合、その要素は1つ以上の列（フラット化設定によって異なる）として出力に表示されます。要素チェック・ボックスが選択されていない場合、要素は出力されません。

要素の可視性設定は、その要素の子の可視性、文書の解析方法または生成される行の数には影響しません。ただし、要素のすべての子のチェック・ボックスが未選択の場合、そのノードのブランチはデフォルトでは折りたたまれています。

要素のチェック・ボックスが選択されている場合、「Type」、「Width」、「Date Format」、「Flatten」および「Maximum Occurrence」オプションが有効になっています。

**注意：** 前述のルールには例外があります。「Flatten」および「Maximum Occurrence」オプションは、チェック・ボックスが選択されている場合でも、ルート要素に対しては有効になりません。

このセクションのオプションの詳細は、次の表を参照してください。

表 22: 「XML Schema Information」の設定

設定	説明
XML Schema Information	このテキスト・ボックスで、ユーザーは、クライアント・マシンで表示可能な XML 文書を検索します。
Reload	XML Schema Information ファイルは、明示的に参照する場合は読み取り専用になります。このファイルを再度読み取る場合は、「Reload」をクリックします。内部文書のツリーに変更が行われても、このファイルは変更されません。 XML Schema information ファイルに変更が行われても、内部文書のツリーは「Reload」をクリックするまで変更されません。

## 「Formatting Options」セクション

このセクションのオプションの詳細は、次の表を参照してください。

表 23: 「Formatting Options」の設定

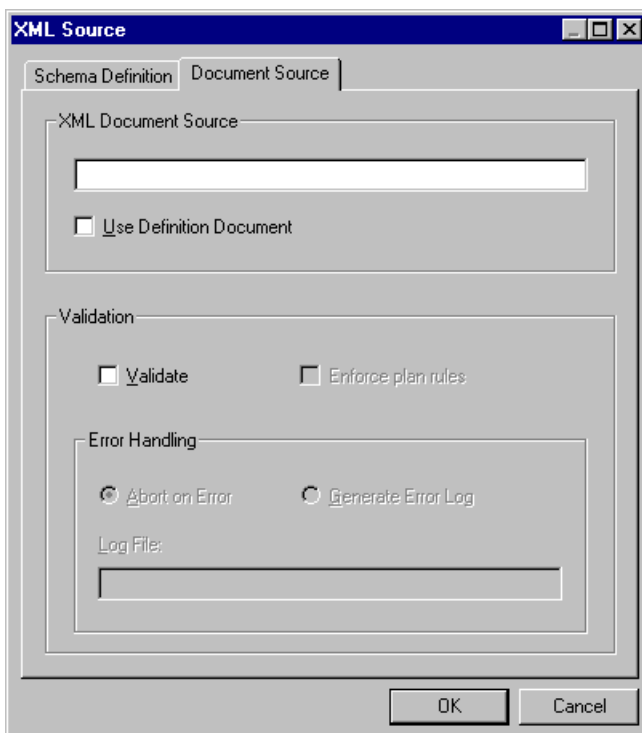
設定	説明
Type	要素の出力タイプの設定に使用されます。このオプションは、出力が選択されていない要素に対しては無効になっています。
Width	要素の出力幅の設定に使用されます。このオプションは、出力が選択されていない要素に対しては無効になっています。
Date Format	内部プラットフォームの表現に変換できるように Datetime 要素の入力形式を指定する場合に使用されます。 このオプションは、出力が選択されていない要素または Datetime 以外のタイプの要素に対しては無効になっています。

表 23: 「Formatting Options」 の設定 (続き)

設定	説明
Flatten	<p>「Flatten」を選択すると、現在選択されている要素が「Maximum Occurrence」に指定されている回数ピボットされます。これは、選択した要素の親要素が出現するたびに、その要素がすべての子要素とともに「Maximum Occurrence」に指定されている回数解析される（選択されている場合は表示される）ことを意味します。この回数を超えてこの要素が親内に出現しても、すべて無視されます。</p> <p>「Flatten」オプションは、子要素が含まれている要素のブランチをフラット化する場合に使用されます。そのような要素をフラット化すると、この Transform では、多くのレコード（行）を作成するかわりに、情報をレコードに追加列として追加します。このオプションは、属性、ルート要素、および自身が表示不可能で、表示可能な子も含まれていないすべての要素に対しては（チェック・ボックスが選択されている場合でも）無効になっています。</p> <p>ルート要素および属性はフラット化できません。この設定は、解析時に無視される要素に対しては無効になっています。そのような要素は表示不可能で、表示可能な子も含まれていません。</p>
Scale	要素のスケールの設定に使用されます。このオプションは、decimal データ型に対してのみ有効になっています。
Precision	要素の精度の設定に使用されます。このオプションは、decimal データ型に対してのみ有効になっています。

## 「Document Source」 タブ

図 20: 「Document Source」 タブ



「Document Source」タブには、次のセクションが含まれています。

- XML Document Source
- Validation
- Error Handling

## 「XML Document Source」セクション

このセクションのオプションの詳細は、次の表を参照してください。

表 24: 「XML Document Source」の設定

設定	説明
XML Document Source	解析する XML 文書の URL（またはサーバーに相対するパス）の指定に使用されます。
Use Definition Document	このオプションを選択した場合、「XML Document Source」が「Schema Definition」タブの XML Schema Information ファイルと同じになります。

## 「Validation」セクション

このセクションのオプションの詳細は、次の表を参照してください。

表 25: 「Validation」の設定

設定	説明
Validate	パーサー検証確認を開始します。これは、解析対象の XML 文書によって DTD または XSD 定義が参照された場合、解析されたデータがその定義に対して検証されることを意味します。

## 「Error Handling」セクション

このセクションのオプションの詳細は、次の表を参照してください。

表 26: 「Error Handling」の設定

設定	説明
Abort on Error	このオプションを選択すると、最初の検証エラーによってプランの実行が停止され、エラーに関する情報を示すエラー・ダイアログ・ボックスが表示されます。
Generate Error Log	このオプションを選択すると、「Log File」で定義されているファイルにすべての検証エラーが記録されます。
Log File	解析時に検証エラー・メッセージが含まれるファイルの場所および名前を指定します。このファイルの場所は、DFS マシンに相対して指定する必要があります。このファイルがすでに存在する場合、そのファイルは破棄され、再作成されます（新しいデータは元のファイルには追加されません）。

**注意：** 整形形式ではない文書によって生成されたエラーは、致命的であるとみなされ、「Validate」が選択されているかどうかに関係なく、常にプランの実行を強制終了し、エラー・メッセージを戻します。XML 仕様では、整形形式文書が標準の XML ルールでコンパイルされます（たとえば、すべてのタグには関連閉じタグが含まれます）。

## XML Source の特性

パッケージ・タイプ	Basic
スキーマの変更	実行する
マルチパスまたはシングルパス	シングルパス
コピーまたはインプレース	コピー
入力のソート	N/A
プロパティの公開	一部





---

# 表示シンク Transform

ここで説明する内容は次のとおりです。

Transform 名	説明
<a href="#">Grid</a>	結果を列形式で表示します。

## Grid

Grid Transform を使用すると、書式設定が可能なコラム表示形式で結果セットが表示されます。Grid Transform は、Part をワークスペースにドラッグしてプランを作成すると、データ・フローに自動的に追加されます。

プランには、各データ・フローに独自の Grid を含めて、複数のデータ・フローを含めることができます。複数の Grid が含まれているプランでは、ワークスペース内の異なる表示タブ上に各 Grid が表示されます。また、プランには、同じデータ・フローに複数の Grid 出力を含めることもできます。

Grid 表示の右クリック・メニューのオプションを使用すると、Grid のテキストおよび背景の外観を書式設定できます。詳細は、Oracle BAM Design Studio のオンライン・ヘルプを参照してください。

Grid では、西暦 100 年 1 月 1 日から西暦 9999 年 12 月 31 日までの日付がサポートされています。

Grid Transform では、Decimal Math Provider の設定を有効にしているかどうかに関係なく、すべての decimal 型フィールドの実際の文字列結果が表示されます。「Scale」のデフォルト値は 2 です。結果は、小数点以降の指定した桁数に四捨五入されます。列が分数用に書式設定されている場合、decimal 型の結果は、分数を計算する前に double 型に変換されます。Grid Transform では、double 型および decimal 型のフィールドに同じ書式設定オプションが含まれます。

パッケージ・タイプ	Basic
スキーマの変更	実行しない
マルチパスまたはシングルパス	N/A
コピーまたはインプレース	N/A
入力のソート	実行する
プロパティの公開	なし

---

# 非表示シンク Transform

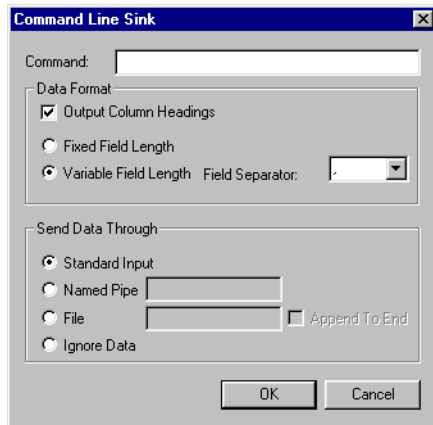
ここで説明する内容は次のとおりです。

Transform 名	説明
<a href="#">Command Line Sink</a>	結果をコマンドライン・ユーティリティに送信します。
<a href="#">Delimited Text File Sink</a>	結果をデリミタ付きの値としてテキスト・ファイルにエクスポートします。
<a href="#">Oracle BAM Alert</a>	指定したアラートをプランの条件に基づいて起動します。
<a href="#">Oracle BAM Delete</a>	ADC 内のデータ・オブジェクトからレコードを削除します。
<a href="#">Oracle BAM Insert</a>	受信したレコードを ADC 内のデータ・オブジェクトに挿入します。
<a href="#">Oracle BAM Message Sender</a>	受信したレコードを ADC 内のデータ・オブジェクトに挿入します。
<a href="#">Oracle BAM Message Tracker</a>	グローバル・トランザクション用のメッセージを追跡します。
<a href="#">Oracle BAM Update</a>	ADC 内のデータ・オブジェクトのレコードを更新します。
<a href="#">SQL Command Sink</a>	各行に対してカスタマイズされた SQL コマンドを実行します。
<a href="#">Terminal Sink</a>	表示なしまたは後続のアクションなしでデータ・フローを終了します。
<a href="#">VBScriptSink</a>	VBScript を使用して外部アプリケーションにレコードを出力します。

## Command Line Sink

Command Line Sink は、コマンドライン・ユーティリティにデータを渡す場合に使用します。これは、データ・マート・データベースにデータをロードするユーティリティをコマンドラインから実行する場合に有効です。

図 21: 「Command Line Sink」ダイアログ・ボックス



ここで説明する内容は次のとおりです。

- 「Command Line Sink」ダイアログ・ボックスの設定
- Command Line Sink Transform に対するヒント
- Command Line Sink の特性

### 「Command Line Sink」ダイアログ・ボックスの設定

Command Line Sink を定義するには、次の表に従って設定項目を指定します。

表 27: Command Line Sink: 「Command」の設定

設定	説明
Command	処理するコマンドラインを入力します。

表 28: Command Line Sink: 「Data Format」の設定

設定	説明
Output Column Headings	このチェック・ボックスは、出力データに列ヘッダーを含める場合に選択します。
Fixed Field Length	これは、データを固定長ファイルに変換する場合に選択します。
Variable Field Length	これは、データをデリミタ付きファイルに変換する場合に選択します。
Field Separator	データを区切るフィールド・セパレータを指定します。

表 29: Command Line Sink: 「Send Data Through」 の設定

設定	説明
Standard Input	これは、データを標準入力方法で送信する場合に選択します。
Named Pipe	これは、データを名前付きパイプで送信する場合に選択します。
File	これは、データをファイルに送信する場合に選択します。ファイルのローカル・パスを指定する必要があります。
Append To End	このチェック・ボックスは、データをファイルの最後に追加する場合に選択します。
Ignore Data	これは、データを送信しない場合に選択します。これは、Terminal Sink として機能します。

## Command Line Sink Transform に対するヒント

Command Line Sink は、デリミタ付きテキスト・ファイルを作成する場合に非常に効率的です。この目的での使用をお勧めします。

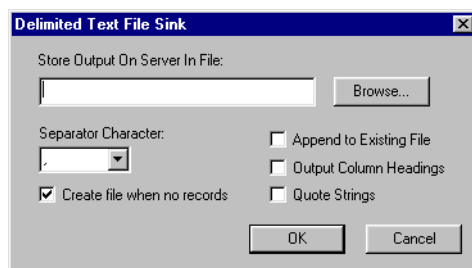
## Command Line Sink の特性

パッケージ・タイプ	Power Product
スキーマの変更	実行しない
マルチパスまたはシングルパス	シングルパス
コピーまたはインプレース	N/A
入力のソート	実行しない
プロパティの公開	一部

## Delimited Text File Sink

Delimited Text File Sink は、プランの結果をデリミタ付き値としてテキスト・ファイルにエクスポートする場合に使用します。ほとんどのリレーショナル・データベースでは、デリミタ付き値をテキスト・ファイルから表にインポートできます。パス、ファイル名および行ヘッダーを含めるかどうかを指定します。

図 22: 「Delimited Text File Sink」 ダイアログ・ボックス



ここで説明する内容は次のとおりです。

- 「[Delimited Text File Sink](#)」ダイアログ・ボックスの設定
- [Delimited Text File Sink Transform](#) に対するヒント
- [Delimited Text File Sink](#) の特性

## 「Delimited Text File Sink」ダイアログ・ボックスの設定

Delimited Text File Sink を定義するには、次の表に従って設定項目を指定します。

表 30: 「Delimited Text File Sink」の設定

設定	説明
Store Output On Server In File	出力の格納先のファイルを入力または検索します。
Separator Character	デリミタを指定します。
Create file when no records	このチェック・ボックスは、処理するレコードが存在しなくてもファイルを作成する場合に選択します。
Append to Existing File	このチェック・ボックスは、処理されたレコードを既存のファイルに上書きしないで追加する場合に選択します。
Output Column Headings	このチェック・ボックスは、出力ファイルに列ヘッダーを出力する場合に選択します。
Quote Strings	このチェック・ボックスは、出力ファイルで引用符文字列を使用できるようにする場合に選択します。

## Delimited Text File Sink Transform に対するヒント

- Command Line Sink を使用すると、パフォーマンスが向上します。
- 作成するファイルのパスおよびファイル名を「Store Output On Server In File」フィールドに入力する場合は、このパスを Oracle BAM Enterprise Link Server のローカル・パスとして指定します。パスを指定しない場合、テキスト・ファイルは、Oracle BAM Enterprise Link Server 上のデフォルトの「Enterprise Link」ディレクトリに保存されます。Enterprise Link Server 以外のマシンにファイルを保存するには、Enterprise Link Server に相対してパスを指定します。たとえば、Mars というマシンへのパスは、`¥¥Mars¥CDrive¥Program Files¥Oracle BAM Enterprise Link¥Save.csv` のようになります。
- 文字列値を引用符で囲むと、string フィールドにフィールド区切り文字を含めることができます。使用する入力ソフトウェアでこの変換がサポートされていることを確認してください。.csv ファイル内の NULL 値は空の文字列として格納されます。
- Delimited Text File Sink では、NULL 値は空白として出力されます。

## Delimited Text File Sink の特性

パッケージ・タイプ	Power Product
スキーマの変更	実行しない
マルチパスまたはシングルパス	シングルパス
コピーまたはインプレース	N/A
入力のソート	実行しない
プロパティの公開	すべて

## Oracle BAM Alert

Oracle BAM Alert Transform では、指定したアラートがこの Transform が含まれているプランの条件に基づいて起動されます。プランの開始時、プランの完了時、プランでのエラー発生時またはプランの停止時にアラートの通知を受信することができます。

Data Flow Service を実行しているユーザーが所有しているアラート・ルールのみを起動できます。プランにアクセスできるエンド・ユーザーが、自身のアラートを起動できない場合があります。この場合は、アラート・ルールを Data Flow Service アカウントに再割当てする必要があります。

Alert Transform は、プランで別のシンク Transform が必要な場合、Splitter を使用していずれの場所でもプランに接続できます。サブプラン内の Alert Transform を接続すると、アラートは、メイン・レベルのプランの終了時ではなく、サブプランが終了するたびに起動されます。

アラート名を入力する場合は、DOMAIN¥username.alertname という構文を使用する必要があります。たとえば、COMPANY1¥mjohanson.Alert45 です。

図 23: 「Oracle BAM Alert」 ダイアログ・ボックス

ここで説明する内容は次のとおりです。

- 「Oracle BAM Alert」 ダイアログ・ボックスの設定
- Oracle BAM Alert の特性

## 「Oracle BAM Alert」 ダイアログ・ボックスの設定

Oracle BAM Alert を定義するには、次の表に従って設定項目を指定します。

表 31: 「Oracle BAM Alert」 の設定

設定	説明
Name of rule to launch on Plan start	プランの開始時に起動されるルールの名前を入力します。
Name of rule to launch if Plan completes successfully	プランが正常に完了した場合に起動されるルールの名前を入力します。
Name of rule to launch if Plan fails	プランで障害が発生した場合に起動されるルールの名前を入力します。このルールは、エラー（正常終了または手動停止以外）が原因でプランが終了した場合に起動されません。
Name of rule to launch if Plan is stopped	プランが停止された場合に起動されるルールの名前を入力します。このルールは、Oracle BAM Administrator を使用してプランの実行を停止することによって、プランを手動で停止した場合に起動されます。
Name of rule to launch for each row	レコードが Transform に送信されるたびに起動されるルールの名前を入力します。継続して実行されているプランで大量のデータがデータ・フローに追加される場合は、このフィールドによって多数のアラートが起動される可能性があります。アラートに頻度制約を追加すると、そのアラートが起動される回数を制限できます。 ルールを起動しない条件に対するフィールドは、空白のままにします。

## Oracle BAM Alert の特性

パッケージ・タイプ	Basic
スキーマの変更	実行しない
マルチパスまたはシングルパス	シングルパス
コピーまたはインプレース	N/A
入力のソート	実行しない
プロパティの公開	一部

## Oracle BAM Delete

Oracle BAM Delete Transform では、データ・フローから取得したレコードに基づいて、ADC 内のデータ・オブジェクトからレコードが削除されます。

ここで説明する内容は次のとおりです。

- [「Oracle BAM Alert」 ダイアログ・ボックスの設定](#)
- [Oracle BAM Alert の特性](#)



## Oracle BAM Delete Transform の設定

Oracle BAM Delete Transform を定義するには、ダイアログ・ボックスを表示し、次のタブを設定します。

- 「ADC Information」 タブ
- 「Record Selection」 タブ

### 「ADC Information」 タブ

設定を定義するには、「ADC Information」タブを選択し、表 32 に従って設定項目を指定します。

図 24: 「ADC Information」タブのダイアログ・ボックス

表 32: 「ADC Information」の設定

設定	説明
ADC Server	データ・オブジェクトが含まれている ADC サーバー・マシンの名前を入力します。デフォルト名は Main です。
Data Object name	使用する既存のデータ・オブジェクトの名前を選択します。
Include in Global Transaction	「Include in Global Transaction」チェック・ボックスは、グローバル・トランザクションにこの Transform を含める場合に選択します。これを選択した場合は、フィールドにグローバル・トランザクションの名前を入力します。

## 「Record Selection」タブ

設定を定義するには、「Record Selection」タブを選択し、表 33 に従って設定項目を指定します。

図 25: 「Record Selection」タブのダイアログ・ボックス

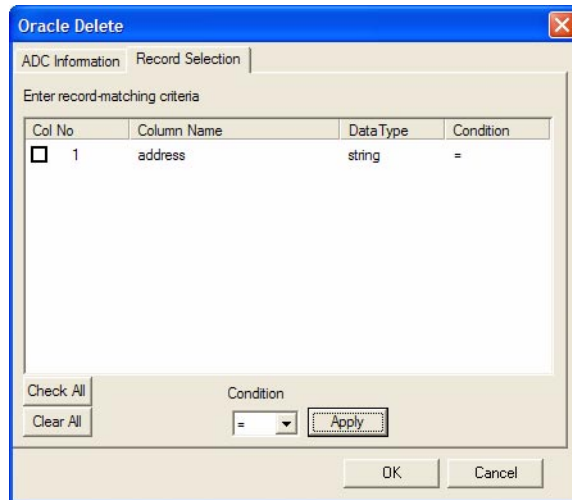


表 33: 「Record Selection」の設定

設定	説明
Col No.	このチェック・ボックスは、列を選択する場合に選択します。すべての列を選択する場合は「Check All」、すべての列を選択解除する場合は「Clear All」を選択します。
Condition	各列に対して、「Condition」ドロップダウン・リストから値を選択します。
Apply	「Apply」は、条件値を適用する場合にクリックします。

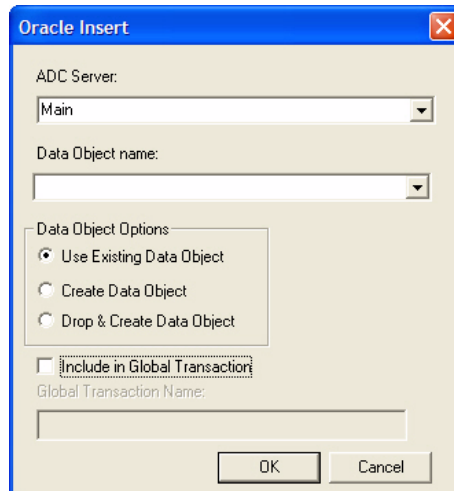
## Oracle BAM Delete の特性

パッケージ・タイプ	Basic
スキーマの変更	実行しない
マルチパスまたはシングルパス	シングルパス
コピーまたはインプレース	N/A
入力のソート	実行しない
プロパティの公開	一部

# Oracle BAM Insert

Oracle BAM Insert Transform では、受信したレコードが ADC 内のデータ・オブジェクトに挿入されます。

図 26: 「Oracle BAM Insert」 ダイアログ・ボックス



ここで説明する内容は次のとおりです。

- 「Oracle BAM Insert」 ダイアログ・ボックスの設定
- Oracle BAM Insert の特性

## 「Oracle BAM Insert」 ダイアログ・ボックスの設定

Oracle BAM Insert を定義するには、次の表に従って設定項目を指定します。

表 34: 「Oracle BAM Insert」 の設定

設定	説明
ADC Server	データ・オブジェクトが含まれている ADC サーバー・マシンの名前を入力します。デフォルト名は Main です。
Data Object name	使用するデータ・オブジェクトの名前を選択します。
Data Object Options	次のいずれかのアクションを選択します。 Use Existing Data Object: 指定したデータ・オブジェクトに挿入する場合。 Create Data Object: 指定した名前を使用して新しいデータ・オブジェクトを作成し、そのデータ・オブジェクトに挿入する場合。このオプションを使用して作成されたデータ・オブジェクトは編集できません。 Drop & Create Data Object: 指定したデータ・オブジェクトのコンテンツを消去し、新しいデータを挿入する場合。
Include in Global Transaction	「Include in Global Transaction」チェック・ボックスは、グローバル・トランザクションにこの Transform を含める場合に選択します。
Global Transaction Name	これを選択した場合は、フィールドにグローバル・トランザクションの名前を入力します。

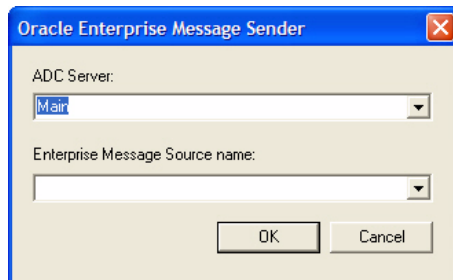
## Oracle BAM Insert の特性

パッケージ・タイプ	Basic
スキーマの変更	実行しない
マルチパスまたはシングルパス	シングルパス
コピーまたはインプレース	N/A
入力のソート	実行しない
プロパティの公開	一部

## Oracle BAM Message Sender

Oracle BAM Message Sender Transform では、受信したレコードが ADC 内のデータ・オブジェクトに挿入されます。Message Sender Transform では、Message Receiver Transform の逆が実行されます。Message Sender Transform は、送信するメッセージに XML がパックされます。

図 27: 「Oracle BAM Message Sender」 ダイアログ・ボックス



ここで説明する内容は次のとおりです。

- 「Oracle BAM Message Sender」 ダイアログ・ボックスの設定
- Oracle BAM Message Sender の特性

### 「Oracle BAM Message Sender」 ダイアログ・ボックスの設定

Oracle BAM Message Sender を定義するには、次の表に従って設定項目を指定します。

表 35: 「Oracle BAM Message Sender」 の設定

設定	説明
ADC Server	データ・オブジェクトが含まれている ADC サーバー・マシンの名前を入力します。デフォルト名は Main です。
Enterprise Message Source name	使用するエンタープライズ・メッセージ・ソースの名前を選択します。

## Oracle BAM Message Sender の特性

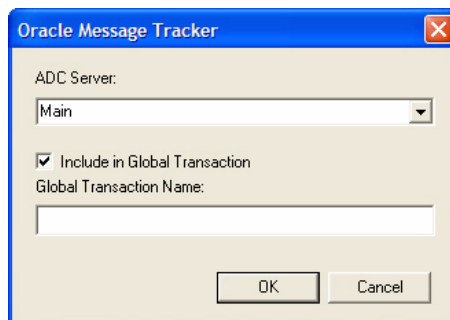
パッケージ・タイプ	Basic
スキーマの変更	実行しない
マルチパスまたはシングルパス	シングルパス
コピーまたはインプレース	N/A
入力のソート	実行しない
プロパティの公開	一部

## Oracle BAM Message Tracker

Oracle BAM Message Tracker Transform では、メッセージが追跡されます。また、Oracle BAM Message Tracker Transform は、グローバル・トランザクション・グループに含めることができます。

プラン内の複数の Transform は、データ処理用のグローバル・トランザクション・グループに含めることができます。メッセージは、処理が中断された場合に追跡されます。

図 28: 「Oracle BAM Message Tracker」ダイアログ・ボックス



ここで説明する内容は次のとおりです。

- 「Oracle BAM Message Tracker」ダイアログ・ボックスの設定
- Oracle BAM Message Tracker の特性

## 「Oracle BAM Message Tracker」 ダイアログ・ボックスの設定

Oracle BAM Message Tracker を定義するには、次の表に従って設定項目を指定します。

表 36: 「Oracle BAM Message Tracker」 の設定

設定	説明
ADC Server	データ・オブジェクトが含まれている ADC サーバー・マシンの名前を入力します。デフォルト名は Main です。
Include in Global Transaction	「Include in Global Transaction」 チェック・ボックスは、グローバル・トランザクションにこの Transform を含める場合に選択します。これを選択した場合は、フィールドにグローバル・トランザクションの名前を入力します。

## Oracle BAM Message Tracker の特性

パッケージ・タイプ	Basic
スキーマの変更	実行しない
マルチパスまたはシングルパス	シングルパス
コピーまたはインプレース	N/A
入力のソート	実行しない
プロパティの公開	一部

## Oracle BAM Update

Oracle BAM Update Transform では、データ・フローから取得したレコードに基づいて、ADC 内のデータ・オブジェクトのレコードが更新されます。

ここで説明する内容は次のとおりです。

- [「Oracle BAM Update」 ダイアログ・ボックスの設定](#)
- [Oracle BAM Update の特性](#)

## 「Oracle BAM Update」ダイアログ・ボックスの設定

Oracle BAM Update Transform を定義するには、ダイアログ・ボックスを表示し、次のタブを設定します。

- 「ADC Information」 タブ
- 「Record Selection」 タブ
- 「Update Specifications」 タブ

### 「ADC Information」 タブ

ソース・ファイルを定義するには、「ADC Information」タブを選択し、表 37 に従って設定項目を指定します。

図 29: 「ADC Information」タブのダイアログ・ボックス

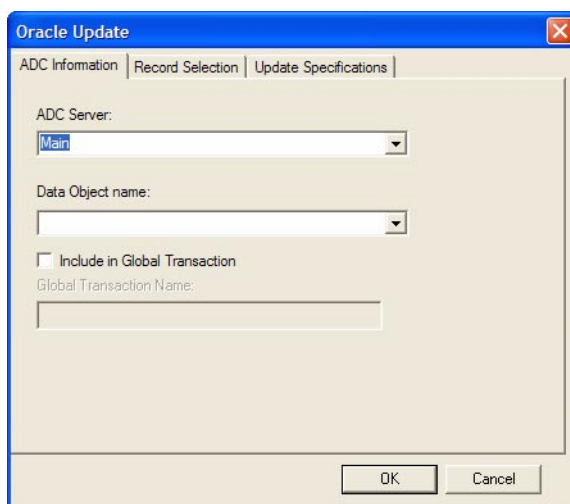


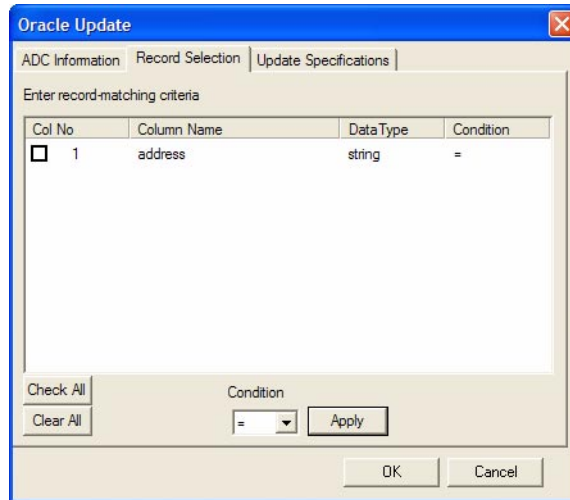
表 37: 「ADC Information」の設定

設定	説明
ADC Server	データ・オブジェクトが含まれている ADC サーバー・マシンの名前を入力します。デフォルト名は Main です。
Data Object name	使用する既存のデータ・オブジェクトの名前を選択します。
Include in Global Transaction	「Include in Global Transaction」チェック・ボックスは、グローバル・トランザクションにこの Transform を含める場合に選択します。これを選択した場合は、フィールドにグローバル・トランザクションの名前を入力します。

## 「Record Selection」タブ

設定を定義するには、「Record Selection」タブを選択し、表 38 に従って設定項目を指定します。

図 30: 「Record Selection」タブのダイアログ・ボックス



設定を定義するには、「Record Selection」タブを選択し、表 38 に従って設定項目を指定します。

表 38: 「Record Selection」の設定

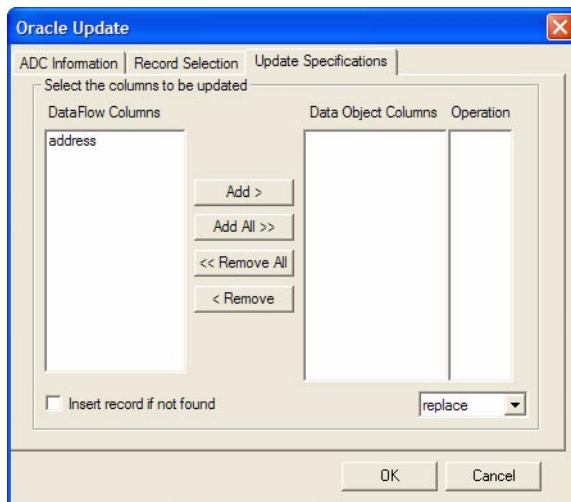
設定	説明
Col No.	このチェック・ボックスは、列を選択する場合に選択します。すべての列を選択する場合は「Check All」、すべての列を選択解除する場合は「Clear All」を選択します。
Condition	各列に対して、「Condition」ドロップダウン・リストから値を選択します。
Apply	「Apply」は、条件値を適用する場合にクリックします。



## 「Update Specifications」タブ

ソース・ファイルを定義するには、「Update Specifications」タブを選択し、表 39 に従って設定項目を指定します。

図 31: 「Update Specifications」タブのダイアログ・ボックス



Oracle BAM Update を定義するには、次の表に従って設定項目を指定します。

表 39: 「Update Specifications」の設定

設定	説明
Data Flow Service	データ・フロー列を選択し、「Add」をクリックしてデータ・オブジェクト内の列を更新します。
Data Object Columns	データ・オブジェクト列を選択し、置換以外の操作を実行する場合はドロップダウン・リストから「Add」または「Subtract」のいずれかを選択します。
Insert record if not found	「Insert record if not found」チェック・ボックスは、レコードを挿入および更新する場合に選択します。

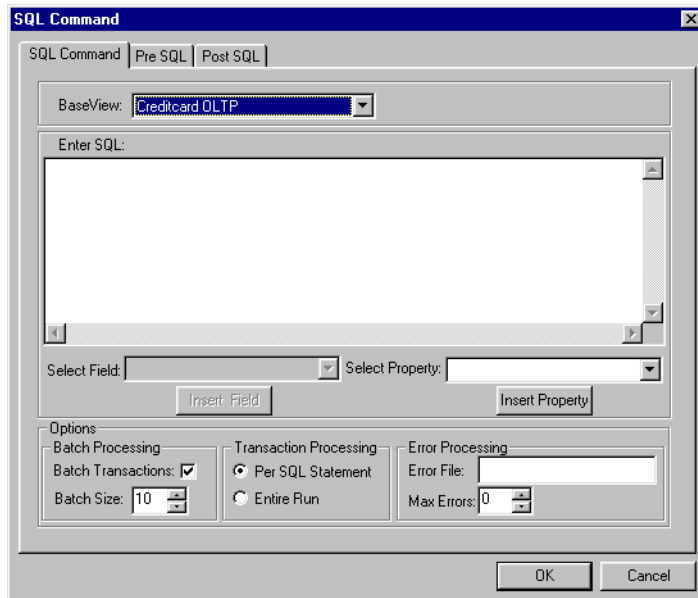
## Oracle BAM Update の特性

パッケージ・タイプ	Basic
スキーマの変更	実行しない
マルチパスまたはシングルパス	シングルパス
コピーまたはインプレース	N/A
入力のソート	実行しない
プロパティの公開	一部

## SQL Command Sink

SQL Command Sink は、プラン内の各行に対してカスタマイズされた SQL コマンドを実行する場合に使用します。この Transform によって、データ・フロー内の入力レコードの各バッチに対して指定した SQL が実行されます。この Transform を使用すると、ターゲットのディメンション表またはファクト表内のレコードを更新できます。

図 32: 「SQL Command Sink」ダイアログ・ボックス



ここで説明する内容は次のとおりです。

- [「SQL Command Sink」ダイアログ・ボックスの設定](#)
- [SQL Command Sink の特性](#)

### 「SQL Command Sink」ダイアログ・ボックスの設定

SQL Command Sink の設定を定義するには、次の表に従って設定項目を指定します。

表 40: 「SQL Command Sink」の設定

設定	説明
BaseView	SQL を実行する表（複数可）が含まれている BaseView を選択します。リポジトリ内の任意の BaseView を選択できます。
Enter SQL	実行する SQL 文を入力します。構文は、データベース・ベンダーの要件に準拠している必要があります。複数の文を実行する場合は、適切なデリミタ文字を使用して文を区切ります。単一の文の場合、デリミタは必要ありません。
Select Field	SQL に挿入するデータ・フロー内の各フィールドをドロップダウン・リストから選択して、「Insert Field」をクリックします。

表 40: 「SQL Command Sink」の設定 (続き)

設定	説明
Insert Field	選択したフィールドを文に挿入する場合にクリックします。
Select Property	SQL に挿入する SQL Command ステップの各設定をドロップダウン・リストから選択して、「Insert Property」をクリックします。
Insert Property	選択した設定を文に挿入する場合にクリックします。
Batch Transactions	トラブルシューティングのためにバッチ処理を無効にする場合は、「Batch Transactions」チェック・ボックスの選択を解除します。
Batch Size	必要に応じて「Batch Size」の値を調整します。デフォルト値は 10 入力行です。最適なバッチ・サイズは、SQL 文および処理されるレコードの数によって異なります。
Per SQL Statement	「Per SQL Statement」または「Entire Run」をクリックしてトランザクションを処理します。「Per SQL Statement」オプションでは、各バッチが個別のトランザクションとしてデータベースに送信されるため、バッチは個々にコミットされます。多数のレコードを処理する場合は、「Per SQL Statement」をクリックし、バッチ・サイズを適切に調整します。通常、バッチ・サイズを大きくすると、より効率的になります。
Entire Run	「Entire Run」は、1 つの大規模トランザクションを作成する場合に選択します。このトランザクションは、すべてのバッチが受信された後にコミットされます。
Error File	Data Load Server 上のエラー・ファイルのパスおよび名前を入力します。パスは、Oracle BAM Enterprise Link Server に相対して指定します。ファイル名を指定しなかった場合、エラー・データは書き込まれません。
Max Errors	SQL 処理を停止するエラーの最大数を「Max Errors」フィールドに指定します。すべてのエラーを許容するには、非常に大きい数値を指定します。
Pre SQL	「Pre SQL」タブで、レコードを処理する前に実行する SQL を入力します。「Pre SQL」タブで SQL 文を追加して、処理するレコードの表を作成できます。 / のみで行を区切って複数の文を指定することができます。
Post SQL	「Post SQL」タブで、レコードを処理した後に実行する SQL を入力します。「Post SQL」タブで SQL 文を追加して、レコードの処理後に索引を作成できます。 / のみで行を区切って複数の文を指定することができます。

## SQL Command Sink の特性

パッケージ・タイプ	Power Product
スキーマの変更	実行しない
マルチパスまたはシングルパス	シングルパス
コピーまたはインブレース	N/A
入力のソート	実行しない
プロパティの公開	すべて

## Terminal Sink

Terminal Sink は、結果の表示または後続のアクションの実行を行わずにデータ・フローを終了する場合に使用します。Terminal Sink が最後に実行されるプランを実行すると、結果がキャッシュに格納されないため、プランを更新するたびに新しい問合せが実行されます。

Terminal Sink にダイアログ・ボックスはありません。

パッケージ・タイプ	Basic
スキーマの変更	実行しない
マルチパスまたはシングルパス	シングルパス
コピーまたはインプレース	N/A
入力のソート	実行しない
プロパティの公開	N/A

## VBScriptSink

VBScriptSink Transform では、レコードが外部アプリケーション（Microsoft Excel、Microsoft Word など）に出力されます。ターゲット・アプリケーションは、OLE 準拠で、Data Flow Service が実行されているクライアント・コンピュータとシステムの両方にインストールされている必要があります。

VBScriptSink Transform に設定項目はありません。ダイアログ・ボックスには、テキスト・フィールドがあります。ここに VBScript を入力して「OK」ボタンをクリックし、ダイアログ・ボックスを閉じて変更を保存します。

VBScript Transform の詳細は、『Oracle BAM Enterprise Link Design Studio ユーザーズ・ガイド』または Oracle BAM Design Studio のオンライン・ヘルプを参照してください。

VBScript Transform を使用してバイナリ・データ型の列は作成できません。

パッケージ・タイプ	Power Product
スキーマの変更	実行しない
マルチパスまたはシングルパス	シングルパス
コピーまたはインプレース	N/A
入力のソート	実行しない
プロパティの公開	なし

---

# データ・フロー制御 Transform

ここで説明する内容は次のとおりです。

Transform 名	説明
<a href="#">Comparison</a>	2 セットの入力レコードを比較し、一致するレコードを識別します。
<a href="#">Conditional Splitter</a>	フィルタ条件を使用してデータ・フローを分割します。
<a href="#">Join</a>	2 つのデータ・フローの列を 1 つに結合します。
<a href="#">Round Robin Splitter</a>	スプリッタに含まれている出力にレコードを分散します。
<a href="#">Splitter</a>	データ・フロー出力の 2 つの同一コピーを作成します。
<a href="#">Subplan</a>	説明図を簡略化し、プランのステップを階層的にグループ化してプランを編成します。また、データ・フローの反復処理も提供します。
<a href="#">Union</a>	2 つのデータ・フローを 1 つに結合します。

## Comparison

Comparison Transform は、比較基準に基づいてデータ・フローを2つの入力データ・セットから3つのデータ・セットに分割する場合に使用します。出力の上部には、2つ目の入力には含まれていない1つ目の入力内のレコードが表示されます。出力の中央には、両方の入力に含まれているすべてのレコードが表示されます。出力の最後には、1つ目の入力には含まれていない2つ目の入力内のレコードが表示されます。

ここで説明する内容は次のとおりです。

- 「[Comparison](#)」 [ダイアログ・ボックスの設定](#)
- [Comparison Transform](#) に対するヒント
- [Comparison](#) の特性

### 「Comparison」ダイアログ・ボックスの設定

Comparison Transform を定義するには、ダイアログ・ボックスを表示し、次のタブを設定します。

- 「[General](#)」 [タブ](#)
- 「[Advanced](#)」 [タブ](#)

#### 「General」タブ

Comparison Transform を定義するには、「General」タブを選択し、[表 41](#)に従って設定項目を指定します。

図 33: 「Comparison」ダイアログ・ボックス

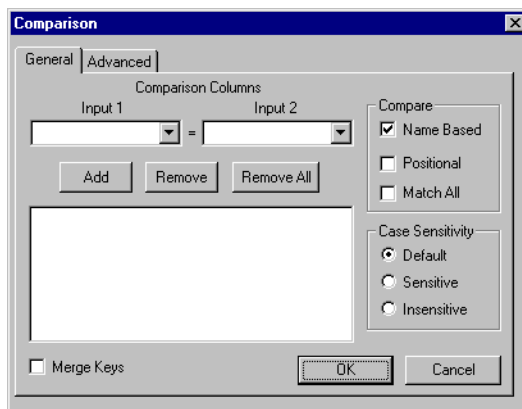


表 41: 「Comparison」 ダイアログ・ボックスの設定

設定	説明
Input 1	比較する列の名前を選択または入力します。
Input 2	比較する列の名前を選択または入力します。
Add	「Input 1」 および 「Input 2」 の値を比較する列のリストに追加します。
Remove	選択した比較をリストから削除します。
Remove All	すべての比較をリストから削除します。
Name Based	このオプションは、一致する名前を持つ列を比較のために選択する場合に選択します。一致する名前を持つ列が検出された場合、それらの列は比較のためにマップされます。
Positional	このオプションは、比較のために列を入力リスト内の位置に基づいて選択する場合に選択します。このステップの入力が「SQL Editor」ダイアログ・ボックスで編集された問合せの場合、列は表示されません。この場合、マッピングはプランの実行時に決定されます。
Match All	このオプションは、両方の入力のすべての列を比較する場合に選択します。入力ごとの列の数が同じである必要があります。「Name Based」が選択されている場合、1つ目と2つ目の入力とは同じである必要がありますが、任意の順序にできます。「Match All」が選択されていないかぎり、「Input 1」および「Input 2」には、異なる数の列を含めることができます。列の数が異なる場合は、マップされた列のみが比較されます。
Default	このオプションは、対象となる BaseView の大 / 小文字を処理するデフォルトの方法を使用してレコードを比較する場合に選択します。
Sensitive	このオプションは、比較で大 / 小文字を区別して一致を検出する場合に選択します。
Insensitive	このオプションは、大 / 小文字を区別せずに一致を検出する場合に選択します。
Merge Keys	このチェック・ボックスは、出力の中央でキー列値をマージする場合に選択します。このオプションを使用するには、2つのセットの入力列でキー列が共有されている必要があります。デフォルトの設定では、キー列はマージされません。

## 「Advanced」タブ

Comparison Transform に拡張機能を設定するには、「Advanced」タブを選択し、表 42 に従って設定項目を指定します。

図 34: 「Advanced」タブのダイアログ・ボックス

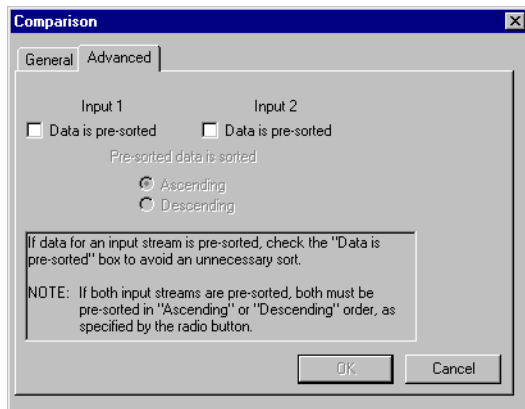


表 42: 「Advanced」タブの設定

設定	説明
Input 1 Data is pre-sorted	このチェック・ボックスは、「Input 1」のデータがすでにソートされている場合に選択します。
Input 2 Data is pre-sorted	このチェック・ボックスは、「Input 2」のデータがすでにソートされている場合に選択します。
Ascending	データがすでに昇順でソートされていることを示します。
Descending	データがすでに降順でソートされていることを示します。

## Comparison Transform に対するヒント

- 比較で複数の基準を指定するには、追加の Comparison Transform および Union Transform を使用します。
- 2 つ以上の列を比較できます。
- 異なるサイズのデータ・セットを比較する場合は、大きいほうのデータ・セットを Comparison Transform の入力の上部に追加するとパフォーマンスが向上します。
- 「Advanced」タブを使用して、すでにソートされているデータに対して不要なソートが実行されないようにします。不要なソートを行わないようにすると、パフォーマンスを向上させることができます。
- Comparison Transform は、ソートがデータベースにプッシュバックされないようにするデータ・フローに Transform を挿入しないかぎり、16 列に制限されています。



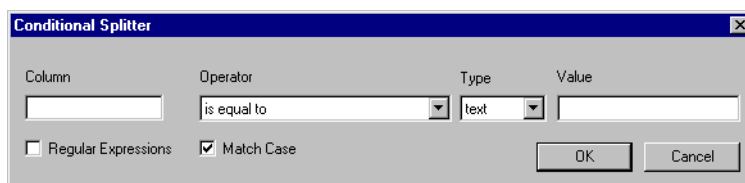
## Comparison の特性

パッケージ・タイプ	Basic
スキーマの変更	実行する
マルチパスまたはシングルパス	マルチパス
コピーまたはインプレース	インプレース
入力のソート	実行する
プロパティの公開	一部

## Conditional Splitter

Conditional Splitter は、フィルタ条件を使用してデータ・フローの出力を 2 つのセットに分割する場合に使用します。この Transform には、1 つの入力および 2 つの出力が含まれています。出力の上部がパススルーされた結果は、指定した条件と一致します。出力の下部がパススルーされた結果は、条件と一致しません。複数の条件を指定するには、追加の Conditional Splitter Transform および Union Transform を使用するか、または複数の Conditional Splitter Transform のかわりに反復が有効にされた Subplan Transform を使用します。

図 35: 「Conditional Splitter」ダイアログ・ボックス



ここで説明する内容は次のとおりです。

- [「Conditional Splitter」ダイアログ・ボックスの設定](#)
- [Conditional Splitter Transform に対するヒント](#)
- [Conditional Splitter の特性](#)

## 「Conditional Splitter」ダイアログ・ボックスの設定

Conditional Splitter を定義するには、次の表を参照してください。

表 43: 「Conditional Splitter」ダイアログ・ボックスの設定

設定	説明
Column	条件として使用する列を選択または入力します。 列名を入力する場合、その列名は Parts Bin 内の名前と一致している必要があります。
Operator	比較演算子を「Operator」ドロップダウン・リストから選択します。 値は次のとおりです。 is equal to is greater than is less than is less than or equal to is not equal to [and not null] is not equal to is null is not null
Type	条件として使用する列のデータ型を選択します。 値は次のとおりです。 text field number
Value	値の型を「Type」ドロップダウン・リストから選択します。 「number」を選択した場合は、フィルタする数値を「Value」フィールドに入力します。 「text」を選択した場合は、フィルタするテキストを「Value」フィールドに入力します。フィルタ基準を拡張するには、テキスト文字列に正規表現を使用します。正規表現は、ワイルドカードと類似していて、大/小文字が区別されます。 「field」を選択した場合は、列名を入力または選択します。
Regular Expressions	1 つ以上の正規表現を「Value」フィールドに含める場合に使用します。 このチェック・ボックスを選択した場合は、比較演算子として「is equal to」または「is not equal to」のいずれかを選択します。
Match Case	入力したテキストと大/小文字が一致する文字列を選択する場合に使用します。

## Conditional Splitter Transform に対するヒント

- 「Conditional Splitter」ダイアログ・ボックスでは、「フィールド」という語は「列」のシノニムです。

## 正規表現

正規表現は、検索またはフィルタ基準で使用されます。また、大 / 小文字が区別され、ワイルドカードと類似しています。次に、いくつかの一般的な正規表現を示します。

文字	説明	例
.	ピリオドは、任意の単一文字を戻します。	.uppy は、puppy または guppy を戻します。
[]	カッコ内の文字セットまたは範囲は、それらの文字セットのいずれかまたは文字範囲を戻します。	[ab] は、a または b のいずれかを戻します。 [a-e] は、a ~ e の任意の文字を戻します。
[^]	カッコ内のカレットは、カレットの後の文字セットのいずれかまたは文字範囲を除く任意の文字を戻します。	[^ab] は、a または b 以外の任意の文字を戻します。 [^a-e] は、a ~ e 以外の任意の文字を戻します。
+	文字の後のプラスは、その文字が 1 つ以上含まれている値を戻します。	[a-e]+ は、a ~ e の文字が 1 つ以上含まれている値を戻します。
*	文字の後のアスタリスクは、その文字が 0 個以上含まれている値を戻します。	[a-e]* は、a ~ e の文字が 0 個以上含まれている値を戻します。
?	文字の後の疑問符は、その文字が 0 個または 1 個含まれている値を戻します。	a?b は、b または ab のいずれかを戻します。
¥	特殊文字の前のバックスラッシュは、リテラル文字を戻します。	\+ は、+ を戻します。

## Conditional Splitter の特性

パッケージ・タイプ	Basic
スキーマの変更	実行する
マルチパスまたはシングルパス	シングルパス
コピーまたはインプレース	コピー
入力のソート	実行しない
プロパティの公開	すべて

# Join

Join Transform は、2つのデータ・フローの列を結合して1つの結果セットを生成する場合に使用します。結合された列では、新しいレコード構造が形成されます。Join Transform では、2つのデータ・フローに関連キー列が含まれている必要があります。

ここで説明する内容は次のとおりです。

- 「Join」 ダイアログ・ボックスの設定
- Join Transform に対するヒント
- Join の特性

## 「Join」 ダイアログ・ボックスの設定

Join Transform を定義するには、ダイアログ・ボックスを表示し、次のタブを設定します。

- 「General」 タブ
- 「Advanced」 タブ

### 「General」 タブ

Join を定義するには、「General」タブを選択し、[表 44](#)に従って設定項目を指定します。

図 36: 「General」タブのダイアログ・ボックス

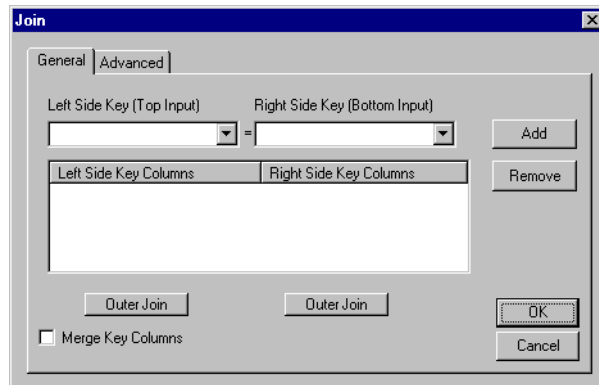


表 44: 「General」 タブの設定

設定	説明
Left Side Key (Top Input)	Join 文の一部として使用する入力列を選択または入力します。 <a href="#">Right Side Key (Bottom Input)</a> と同じデータ型である必要があります。 左側のキー列はソースの上部にあります。
Right Side Key (Bottom Input)	Join 文の一部として使用する入力列を選択または入力します。 <a href="#">Left Side Key (Top Input)</a> と同じデータ型である必要があります。 右側のキー列はソースの下部にあります。
Add	このボタンは、「Left Side Key (Top Input)」および「Right Side Key (Bottom Input)」で選択した列を Join 文のリストに追加する場合に使用します。 Join には、複数の列のペアを含めることができます。たとえば、複数の列内の値に対して制限を行う場合は、別の列のペアを Join に追加します。また、これらの列は同じデータ型である必要がありますが、最初のペアとは異なるデータ型を指定することができます。
Remove	このボタンは、Join 文を削除する場合に使用します。
Left Side Key Columns	このドロップダウン・リストには、データ・フローの上部（左側）の Join 文の列が表示されます。
Right Side Key Columns	このドロップダウン・リストには、データ・フローの下部（右側）の Join 文の列が表示されます。
Outer Join (Top or Left)	外部結合を使用して結果セットのいずれかまたは両方を結合する場合に選択します。 左側外部結合には、他の結果セットで一致していない結果セットの上部（左側）の行が含まれます。 (データ・フローの下部（右側）に一致が検出されていない場合でも) データ・フローの上部（左側）のレコードを含める場合は、結合表示ボックスの左側の直下にある「Outer Join」ボタンをクリックします。
Outer Join (Bottom or Right)	外部結合を使用して結果セットのいずれかまたは両方を結合する場合に選択します。 右側外部結合には、他の結果セットで一致していない結果セットの下部（右側）の行が含まれます。 (データ・フローの上部（左側）に一致が検出されていない場合でも) データ・フローの下部（右側）のレコードを含める場合は、結合表示ボックスの右側の直下にある「Outer Join」ボタンをクリックします。
Merge Key Columns	結合されたキー列を出力内の 1 つの列にマージする場合に使用します。これによって、自然内部結合が生成されます。 「Merge Key Columns」が選択されていない場合は、すべてのキー列が出力に含まれます。 「Merge Key Columns」チェック・ボックスを選択すると、1 つの列に結合された結合列が含まれている結果が出力されます。

**注意：** データ・フローの上部（左側）と下部（右側）の両方のレコードが必要な場合は、両方の「Outer Join」ボタンをクリックします。一致が検出されない場合でも、両方のデータ・フローのレコードが含まれます。

## 「Advanced」タブ

Join に拡張機能を設定するには、「Advanced」タブを選択し、表 45 に従って設定項目を指定します。

図 37: 「Advanced」タブのダイアログ・ボックス

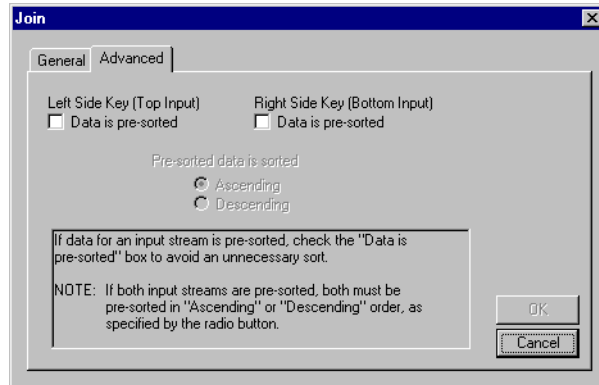


表 45: 「Advanced」タブの設定

設定	説明
Left Side Key (Top Input) Data is pre-sorted	このチェック・ボックスは、入力の上部がすでにソートされている場合に選択します。
Right Side Key (Bottom Input) Data is pre-sorted	このチェック・ボックスは、入力の下部がすでにソートされている場合に選択します。

## Join Transform に対するヒント

- 「Advanced」タブを使用して、すでにソートされているデータに対して不要なソートが実行されないようにします。不要なソートを行わないようにすると、パフォーマンスを向上させることができます。
- 異なるサイズのデータ・セットを結合する場合は、大きいほうのデータ・セットを Join Transform の入力の上部に追加するとパフォーマンスが向上します。
- デフォルトの設定は内部結合です。内部結合は、両方の結果セットで一致する列値が含まれている行のみを戻します。
- 結合する前に「Disk Sort」または「Memory Sort」を指定してデータをソートする際に「Data is pre-sorted」チェック・ボックスを選択した場合は、降順でソートする必要があります。
- プランのソース Transform が同じタイプの場合は、各ソースに一意の名前を指定します。

## Join の特性

パッケージ・タイプ	Basic
スキーマの変更	実行しない
マルチパスまたはシングルパス	マルチパス
コピーまたはインプレース	コピー
入力のソート	実行する
プロパティの公開	一部

## Round Robin Splitter

Round Robin Splitter Transform は、スプリッタに接続されている Transform にレコードを分散する場合に使用します。これによって、複数のセットを同時に処理できるようにデータおよび処理が分割されます。Round Robin Splitter Transform には、1つの入力接続点および複数の出力接続点が含まれています。データ・フロー・コネクタが追加されると、Round Robin Splitter Transform に新しい出力接続点が自動的に追加されます。

Round Robin Splitter では、より効率的に処理を行うためにデータ・フロー内のデータを分散する方法が提供されます。Round Robin Splitter の使用例としては、大規模データ・セットをバッチ・ローダーにロードする例があります。1つのバッチ・ローダーに対する単一のデータ・フローのかわりに、4つの出力を持つ Round Robin Splitter を追加できます。各出力に対して、ダイアログ・ボックスで指定した設定と同じ設定で追加のバッチ・ローダーを接続して、同じターゲット・データベースにロードすることができます。Round Robin Splitter では、各出力でレコードの 25% が受信されるように追加のバッチ・ローダー間でデータが共有されるため、データのセットの同時処理および高速ロードを行うことができます。

Round Robin Splitter Transform にダイアログ・ボックスはありません。

パッケージ・タイプ	Basic
スキーマの変更	実行する
マルチパスまたはシングルパス	N/A
コピーまたはインプレース	N/A
入力のソート	実行しない
プロパティの公開	N/A

## Splitter

Splitter Transform は、データ・フロー内の結果セットの 2 つの同一コピーを作成する場合に使用します。Splitter Transform には、1 つの入力接続点および複数の出力接続点が含まれています。データ・フロー・コネクタが追加されると、Splitter Transform に新しい出力接続点が自動的に追加されます。Splitter Transform で作成される重複データ・セットは、仮想データ・セットのみです。追加の物理リソースは必要ありません。

Splitter Transform にダイアログ・ボックスはありません。

パッケージ・タイプ	Basic
スキーマの変更	実行する
マルチパスまたはシングルパス	N/A
コピーまたはインプレース	N/A
入力のソート	実行しない
プロパティの公開	N/A

## Subplan

Subplan Transform は、階層化されたプランをデータ・フロー・エディタで作成する場合に使用します。Subplan Transform では、ステップをまとめてグループ化することによって、データ・フロー・エディタに一度に表示されるステップの数が削減されます。サブプラン処理の一部として、データ・フローをサブプラン上の反復接続点に接続できます。反復接続点は、データに対して反復処理を実行する場合に使用します。複数のステップおよびデータ・パスが必要なデータ抽出タスクおよびデータ変換タスクの多くは、Subplan Transform および反復処理を使用すると、より少ないステップで実行できます。

反復接続点を使用すると、処理をレコードごとに実行するか、レコード・グループごとに実行するかを定義できます。最初のレコードまたはレコード・グループに対するサブプランの処理が完了した後、サブプランに含まれているステップによって次のレコードまたはレコード・グループに対する処理が再開されます。

サブプランは、プランに無制限に追加できます。Subplan Transform には、複数の入力接続点および出力接続点が含まれています。データ・フロー・コネクタが追加されると、サブプランに新しい接続点が追加されます。サブプランの反復接続点には、1 つのデータ・フローのみを接続できます。非反復データ・フローは、サブプランに無制限に接続できます。



新しいサブプランを追加し、サブプランにデータ・フロー・コネクタ線を追加した後、そのサブプランを初めて表示すると、サブプラン入力記号およびサブプラン出力記号には、次のものが示されます。

- サブプランの反復入力接続点に対応する反復出力接続点（接続されている場合）
- 各サブプランの入力データ・フローに対応する入力接続点
- 各サブプランの出力データ・フローに対応する出力接続点

サブプランおよびサブプラン接続点では、処理の結合、マージまたは分割は実行されません。サブプラン内の各接続点は、上位レベルでサブプランに接続されているデータ・フローの継続です。プランを適切に処理するには、サブプラン内の各接続点がデータ・フローに接続されている必要があります。サブプラン内の **Join Transform**、**Union Transform** および **Splitter Transform** を使用して、データ・フローの数を増減できます。

ここで説明する内容は次のとおりです。

- 「**SubPlan Iteration**」 **ダイアログ・ボックスの設定**
- **Subplan Transform** に対するヒント
- サブプランの特性

## 「SubPlan Iteration」ダイアログ・ボックスの設定

サブプランの反復を定義するには、ダイアログ・ボックスを表示し、次のタブを設定します。

- 「**General**」 **タブ**
- 「**Property Assignments**」 **タブ**

## 「General」 タブ

反復タイプおよびソース・フィールドを定義するには、「General」タブを選択し、表 46 に従って設定項目を指定します。

図 38: 「General」タブのダイアログ・ボックス

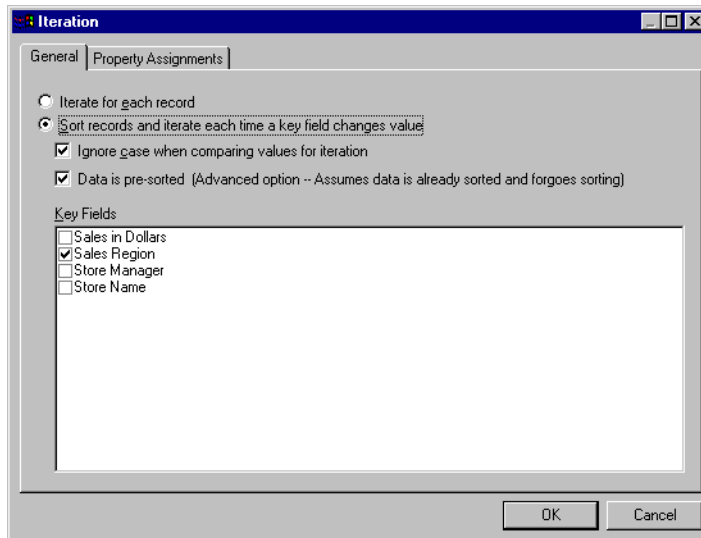


表 46: 「General」タブの設定

設定	説明
反復設定	データ・ソースの各レコードを処理するには、「 <b>Iterate For Each Record</b> 」を選択します。 選択したキー・フィールドに基づいてレコードをソートした後、値が変更されるたびにグループのレコードを処理するには、「 <b>Sort Records and Iterate Each Time A Key Field Changes Value</b> 」を選択します。
Key Fields	データ・ソース内のすべての Part が表示されます。反復設定が「 <b>Sort Records and Iterate Each Time A Key Field Changes Value</b> 」に設定されている場合は、キー・フィールドで使用する Part を選択します。
Ignore Case When Comparing Values For Iteration	反復設定を「 <b>Sort Records and Iterate Each Time A Key Field Changes Value</b> 」に設定して、レコード・グループを決定するキー・フィールド値を評価する際に大 / 小文字の違いを無視する場合に選択します。
Data is pre-sorted	データ・フローに含まれているデータが（先行するソート Transform などによって）すでにソートされている場合は、反復処理時に不要なソートが行われないように「 <b>Data is Pre-Sorted</b> 」を選択します。

## 「Property Assignments」 タブ

プロパティ・マッピングに対するソース・フィールドを定義するには、「Property Assignments」タブを選択し、表 47 に従って設定項目を指定します。

図 39: 「Property Assignments」 タブのダイアログ・ボックス

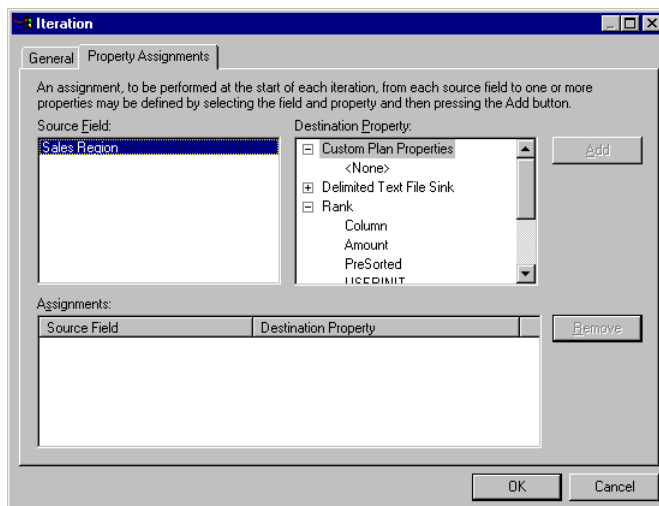


表 47: 「Property Assignments」 タブの設定

設定	説明
プロパティ・マッピングのフィールド	ソース・フィールドとカスタム・プラン・プロパティ間のマッピングを作成する際に使用します。マッピングを作成するには、「Source Field」および「Destination Property」に値を選択した後、「Add」をクリックします。
Source Field	データ・フローのデータ・ソース内の Part が表示されます。反復設定が「Iterate For Each Record」に設定されている場合は、データ・ソース内のすべての Part が表示されます。そうでない場合は、「Key Fields」で選択されている Part が表示されます。
Destination Property	プランに定義されているカスタム・プラン・プロパティおよび公開されているステップ・プロパティが表示されます。
AddSelectBold	「Source Field」で選択したフィールドと「Destination Property」で選択したプロパティ間のマッピングを作成する際にクリックします。
Assignments	サブプランに対するソース・フィールドおよびカスタム・プラン・プロパティの割当てが表示されます。
Remove	「Source Field」で選択したフィールドと「Destination Property」で選択したプロパティ間のマッピングの割当てを削除する際にクリックします。

## Subplan Transform に対するヒント

- サブプランの反復設定を編集するには、サブプランの反復接続点に接続されているデータ・ソース Transform が含まれているデータ・フローが必要です。
- サブプラン出力の各接続点のデータ・フローがサブプラン内部のデータ・フローに接続されている場合、このサブプラン出力の各接続点には、上位レベルのプランに接続される対応するデータ・フローが必要です。
- ソース・フィールドは、0（ゼロ）以上のプロパティに割り当てることができます。接続先プロパティは、1つのフィールドにのみ割り当てることができます。

## サブプランの特性

パッケージ・タイプ	Basic
スキーマの変更	N/A
マルチパスまたはシングルパス	N/A
コピーまたはインプレース	N/A
入力のソート	不定 <sup>1</sup>
プロパティの公開	N/A

1. レコード・グループごとの反復を選択すると、反復データ・フローのデータ・セットは前ステップの出力に基づいてソートされます。レコードごとの反復を選択すると、ソートは実行されません。サブプランがパススルーされる非反復データ・フローのデータ・セットは、サブプラン処理の一部としてはソートされません。

## Union

Union Transform は、複数のデータ・フローの行（重複行を含む）から1つの結果セットを生成する場合に使用します。Union Transform では、1つの行セットを他の行セットの上に配置して、複合行セットが形成されます。各データ・フロー内の行には同数の列が必要で、各列には同一の属性が必要です。

Union Transform には、複数の入力接続点および1つの出力接続点が含まれています。データ・フロー・コネクタが追加されると、Union Transform に新しい入力接続点が自動的に追加されます。

Union Transform にダイアログ・ボックスはありません。

パッケージ・タイプ	Basic
スキーマの変更	実行しない
マルチパスまたはシングルパス	N/A
コピーまたはインプレース	N/A
入力のソート	実行しない
プロパティの公開	N/A

# データ操作 Transform

ここで説明する内容は次のとおりです。

Transform 名	説明
<a href="#">Add Columns</a>	1 つ以上の空の列を追加します。
<a href="#">Column Select</a>	列の削除および並替えを実行します。
<a href="#">Concatenation</a>	列データと入力テキストを結合します。
<a href="#">Disk Sort</a>	ディスク・ファイルを使用して、列をソートします。
<a href="#">Expression Calculator</a>	式を使用して、計算された列を追加します。
<a href="#">Filter</a>	フィルタ条件を使用して、行を削除します。
<a href="#">Key Generation</a>	一意の数値キー値を生成します。
<a href="#">Key Lookup</a>	表内の値を検索して、別の表に挿入します。
<a href="#">Memory Sort</a>	メモリーを使用して、列内のレコードをソートします。
<a href="#">Message Digest</a>	任意の型の入力データに基づいてキーを作成します。
<a href="#">Oracle BAM Lookup</a>	データ・フロー内のデータおよびデータ・オブジェクト内のデータのキー・フィールドに対して検索を実行します。
<a href="#">Pivot</a>	列をクロス集計形式で集計および配置します。
<a href="#">Rank</a>	列の上位の行または下位の行を出力します。
<a href="#">Record Number</a>	順序番号を付けて列を追加します。
<a href="#">Remove Duplicate Rows</a>	データ内の重複行を削除します。
<a href="#">Rename Columns</a>	列名を変更します。
<a href="#">Search &amp; Replace</a>	文字列または部分文字列を別の文字列に置換します。
<a href="#">SQL Command</a>	各行に対してカスタマイズされた SQL コマンドを実行します。

---

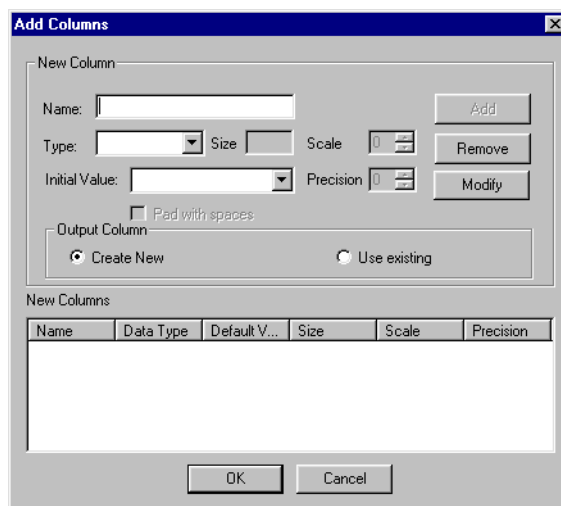
<b>Transform 名</b>	<b>説明</b>
<a href="#">Substring</a>	文字列値から部分文字列を抽出します。
<a href="#">Time Lookup</a>	時間ディメンションのファクト表でキー値を生成します。
<a href="#">Transpose</a>	列を行形式で配置します。
<a href="#">VBScriptCopy</a>	VBScript を使用して入力レコードをコピーします。
<a href="#">VBScriptInplace</a>	VBScript を使用して各入力レコードをインプレース変更します。

---

## Add Columns

Add Columns Transform は、新しい列をデータ・フローに追加する場合に使用します。新しい列のデータ型を指定する必要があります。列は、空にするか、または初期値を含めることができます。データ・フローの後半で Transform のドロップダウン・リストに新しい列名が自動的に表示されます。

図 40: 「Add Columns」ダイアログ・ボックス



ここで説明する内容は次のとおりです。

- [「Add Columns」ダイアログ・ボックスの設定](#)
- [Add Columns Transform に対するヒント](#)
- [Add Columns の特性](#)

### 「Add Columns」ダイアログ・ボックスの設定

Add Columns を定義するには、次の表に従って設定項目を指定します。

表 48: 「Add Columns」の設定

設定	説明
Name	追加する列の名前を入力します。
Type	新しい列のデータ型を選択します。「string」を選択した場合は、「Size」フィールドに文字列の長さ（文字数）を入力します。「decimal」を選択した場合は、「Scale」および「Precision」の値を指定します。
Initial Value	新しい列の初期値を指定します。
Size	列の長さを指定します。
Scale	小数の場合は、スケールを指定します。
Precision	小数の場合は、精度を指定します。

表 48: 「Add Columns」の設定

設定	説明
Add	このボタンは、「New Columns」に新しい列を追加する場合にクリックします。
Remove	このボタンは、選択した列を削除する場合にクリックします。列は、「New Columns」から削除されます。
Modify	このボタンは、列の書式設定に行われた変更を適用する場合にクリックします。列の新しい変更内容が「New Columns」に表示されます。
Pad with spaces	このチェック・ボックスは、データに空白を埋め込んで完全な長さにする場合に選択します。これは、string データ型専用です。
Create New	これは、新しい列を作成する場合に選択します。
Use existing	これは、新しい列で既存の列を上書きする場合に選択します。

表 49: サポートされている列データ型

データ型	説明	制限
String	英数値	なし
Int	整数値	± 64,000
Long	整数値	±2,000,000,000
DateTime	日時値	なし
Double	実数値 (小数を含む)	±2,000,000,000
Unit	正の整数値	64,000 未満
Decimal	記号および小数点付きの 10 進数文字列	なし

## Add Columns Transform に対するヒント

Add Columns Transform を使用して、結果で必要な列を追加します。これによって、プランのパフォーマンスを向上させることができます。他の Transform で、結果をコピーするかわりにインプレース生成できるためです。

## Add Columns の特性

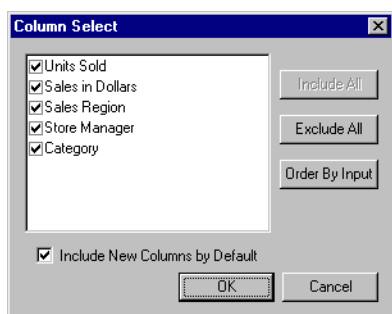
パッケージ・タイプ	Basic
スキーマの変更	実行する
マルチパスまたはシングルパス	シングルパス
コピーまたはインプレース	コピー
入力のソート	実行しない
プロパティの公開	一部



## Column Select

Column Select Transform は、不要な入力列をデータ・フローから削除する場合に使用します。この Transform によって、データ・フロー内の指定した入力列が削除され、残りの列が出力されます。削除された列はデータ・フローから破棄されるため、残りの列の処理パフォーマンスが向上します。ユーザー入力 of SQL がプランに含まれている場合は、別の「Column Select」ダイアログ・ボックスが表示されます。既存の入力列は表示されないため、選択する列の名前を入力する必要があります。

図 41: 「Column Select」ダイアログ・ボックス



ここで説明する内容は次のとおりです。

- [「Column Select」ダイアログ・ボックスの設定](#)
- [Column Select Transform に対するヒント](#)
- [Column Select の特性](#)

### 「Column Select」ダイアログ・ボックスの設定

Column Select を定義するには、次の表に従って設定項目を指定します。

表 50: 生成された SQL: 「Column Select」の設定

設定	説明
列チェック・ボックス	列をデータ・フローから除外するには、列名の横にあるチェック・ボックスの選択を解除します。
Include All	「Include All」は、すべての列チェック・ボックスを選択する場合に選択します。
Exclude All	「Exclude All」は、すべての列チェック・ボックスの選択を解除する場合に選択します。
Order by Input	列をクリックし、新しい位置にドラッグして列を並べ替えて、列の出力順序を変更します。入力列を並べ替えて、インポート順序にリストアする場合は、「Order By Input」をクリックします。
Include New Columns by Default	「Include New Columns by Default」は、データ・フローへの Part の追加時にそれらの Part を選択する場合に選択します。これを選択解除すると、プランに追加された新しい Part がデータ・フローに追加されなくなります。

ユーザー入力の SQL がプランに含まれている場合は、別のダイアログ・ボックスが表示されます。この場合は、次の表に従って設定項目を指定します。

表 51: ユーザー入力の SQL: 「Column Select」 の設定

設定	説明
Enter Name of Column	選択する各列の名前を入力して「Add」をクリックします。入力列が Part の場合は、Parts Bin に表示されているとおりの名前を入力します。
Add	「Add」は、列をデータ・フローに追加する場合にクリックします。
Include Only The Following Fields	下に表示されているリスト内のフィールドを含める場合にクリックします。
Exclude The Following Fields	下に表示されているリスト内のフィールドを除外する場合にクリックします。
Columns Included/Excluded	データ・フローに含められた列またはデータ・フローから除外された列のいずれかを表示します。

## Column Select Transform に対するヒント

- データ・フロー内の追加の処理を列名ではなく位置に基づいて実行する場合は、「Column Select」ダイアログ・ボックスで列を並べ替えます。たとえば、Batch Loader Transform では、名前または位置のいずれかに基づいて列をロードできます。データを表示するシンク・ステップ（Grid Transform など）では、元の順序が表示されます。

Grid Transform を削除し、再度データ・フローに追加すると、列は新しい順序で表示されます。この方法を使用して、Grid への出力を表示し、ロード用の出力順序を検証することができます。

- 入力列がデータ・フロー内でこの Transform の前の Expression Calculator Transform に含まれている場合は、不要な Column Select Transform を追加するかわりに Expression Calculator を使用してその列をデータ・フローから削除します。

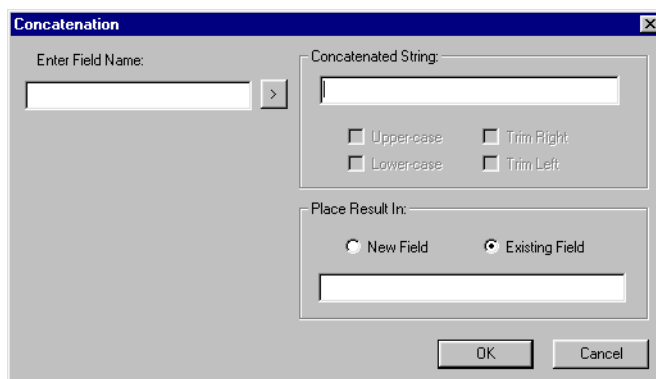
## Column Select の特性

パッケージ・タイプ	Basic
スキーマの変更	実行する
マルチパスまたはシングルパス	シングルパス
コピーまたはインプレース	コピー
入力のソート	実行しない
プロパティの公開	一部

# Concatenation

Concatenation Transform は、複数の列内の値の連結または単一の列内の入力テキストと列値の連結を行う場合に使用します。新しい列に結果を格納するか、または既存の列の値を置換することができます。連結の結果は、文字列値になります。

図 42: 「Concatenation」ダイアログ・ボックス



ここで説明する内容は次のとおりです。

- 「Concatenation」ダイアログ・ボックスの設定
- NULL 値に対する Concatenation の動作
- Concatenation の特性

## 「Concatenation」ダイアログ・ボックスの設定

Concatenation を定義するには、次の表に従って設定項目を指定します。

表 52: 「Concatenation」ダイアログ・ボックス

設定	説明
Enter Field Name	結合または書式設定する列（複数可）を選択する場合にダブルクリックします。
Concatenated String	文字列に文字（空白、カンマ、カッコなど）を追加する場合は、「Concatenated String」フィールドに入力します。列に下線が表示されます。入力する文字に下線は表示されません。
Upper-case	列値を大文字で指定するには、「Concatenated String」フィールド内の名前をクリックし、「Upper-case」を選択します。
Lower-case	列値を小文字で指定するには、「Concatenated String」フィールド内の名前をクリックし、「Lower-case」を選択します。
Trim Right	「Trim Right」は、連結文字列値の右側から不要な空白を切り捨てる場合に選択します。

表 52: 「Concatenation」 ダイアログ・ボックス (続き)

設定	説明
Trim Left	「Trim Left」は、連結文字列値の左側から不要な空白を切り捨てる場合に選択します。
New Field	「New Field」ラジオ・ボタンは、新しい列に結果を格納する場合に選択します。新しいフィールドの名前を入力します。
Existing Field	「Existing Field」は、このプランの結果が含まれている既存の列内の値を置換する場合に選択します。既存の入力が表示されていない場合は、Parts Bin に表示されているとおりの文字列列の名前を入力します。

## NULL 値に対する Concatenation の動作

Concatenation Transform と Expression Calculator では、NULL 値が含まれている文字列が異なる方法で連結されます。

- Concatenation Transform では、NULL 値が含まれている文字列値を連結すると文字列値になります。
- Expression Calculator では、NULL 値が含まれている文字列値を連結すると NULL 値になります。

この動作は、次のキーに ConcatWithNulls という DWORD Registry 設定を追加して変更することができます。

My Computer¥HKEY\_LOCAL\_MACHINE¥SOFTWARE¥Sagent¥DataMart¥TF  
値は、次のいずれかに設定します。

値	動作
0	デフォルト設定
1	Expression Calculator では、Concat(<NULL>, ABC) は ABC になります。
2	Expression Calculator および Concatenation Transform では、2 つのフィールドのいずれかが <NULL> の場合、結果は <NULL> になります。

## Concatenation の特性

パッケージ・タイプ	Basic
スキーマの変更	実行する
マルチパスまたはシングルパス	シングルパス
コピーまたはインプレース	コピー
入力のソート	実行しない
プロパティの公開	一部

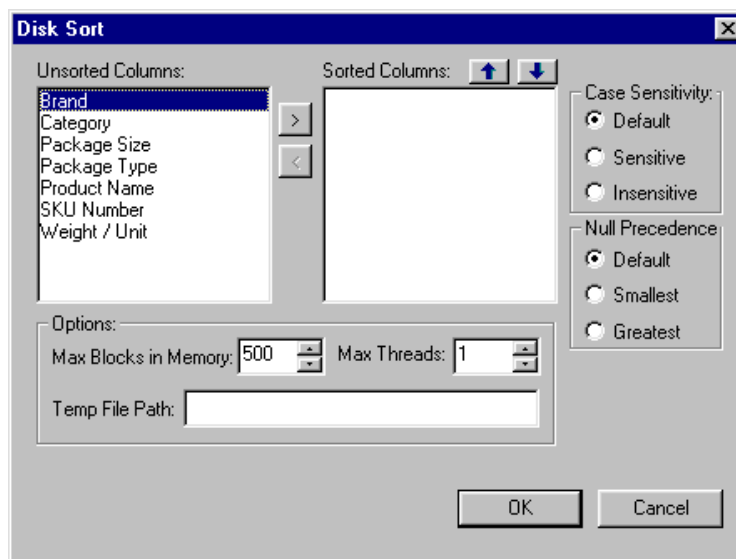
## Disk Sort

Disk Sort は、大規模な入力レコードのセットをソートしてからターゲット表にロードする場合に使用します。

ディスク・ソート時、小規模のレコードのセットは、少量のメモリーを使用して格納された後、Oracle BAM Enterprise Link Server 上の一時ファイルに書き込まれます。このプロセスは、すべての入力レコードがソートされるまで繰り返されます。その後、一時ファイルがマージされ、最終結果が戻されます。

**注意：**これは、パフォーマンス集中型のマルチパス Transform です。別の方法として、フィルタ制御を使用して、ソートを SQL 文にプッシュバックし、パフォーマンスを向上させる方法があります。

図 43: 「Disk Sort」ダイアログ・ボックス



ここで説明する内容は次のとおりです。

- 「Disk Sort」ダイアログ・ボックスの設定
- Disk Sort Transform に対するヒント

## 「Disk Sort」ダイアログ・ボックスの設定

Disk Sort を定義するには、「Disk Sort」ダイアログ・ボックスを表示し、表 53 に従って設定項目を指定します。

表 53: 「Disk Sort」の設定

設定	説明
Unsorted Columns	このフィールドには、この時点でのプランのデータ・フロー内の未ソート列の名前が表示されます。列名を選択し、右矢印ボタンをクリックして「Sorted Columns」フィールドに移動します。
Sorted Columns	このフィールドには、ソート優先順位の順序でソートするために選択された列が表示されます。また、データを昇順でソートするか、降順でソートするかも示されます。デフォルトのソート順序は昇順です。ソート順序を変更するには、次の操作を実行します。列名を選択し、上下矢印ボタンをクリックしてデータを昇順または降順でソートします。ソート優先順位の順序を変更するには、次の操作を実行します。列名をクリックして新しい位置にドラッグします。
「Case Sensitivity」の「Default」	このオプションは、データのソートに RDBMS のデフォルトの大 / 小文字の区別を使用する場合に選択します。
Sensitive	このオプションは、大 / 小文字を区別してデータをソートする場合に選択します。
Insensitive	このオプションは、大 / 小文字を区別しないでデータをソートする場合に選択します。
「Null Precedence」の「Default」	このオプションは、データのソートに RDBMS のデフォルトの NULL 優先順位を使用する場合に選択します。
Smallest	このオプションは、データのソート時に最小値を先頭、NULL 値を最後に配置する場合に選択します。
Greatest	このオプションは、データのソート時に最大値を先頭、NULL 値を最後に配置する場合に選択します。
Max Blocks in Memory	これは、レコードをソートするために割り当てる Oracle BAM Enterprise Link Server 上のメモリー・ブロックの数です。ブロックのサイズは、Oracle BAM Enterprise Link Admin の AgentBlockSize 設定によって決定されます。通常、大きい値を指定するとパフォーマンスが向上しますが、サーバー上の他のプロセスに影響を及ぼす場合があります。デフォルト値は 500 です。レコードのソートに必要なメモリー・ブロックの最小数は 300 です。最大値は約 32767 です。
Max Threads	これは、レコードをソートするために使用する Oracle BAM Enterprise Link Server 上のスレッドの数です。複数のスレッドで、複数のセットのレコードを同時にソートできます。デフォルト値は 1 で、最大値は 99 です。

表 53: 「Disk Sort」の設定 (続き)

設定	説明
Temp File Path	<p>ローカル・マシンからデータのソートに使用する一時ファイルへのパスを定義します。</p> <p>Data Flow Service が実行されているマシンにこのファイルが存在する場合に最適のパフォーマンスが実現されます。</p> <p>これは、ソートされたレコードが一時的にファイルに格納される Oracle BAM Enterprise Link Server 上の場所です。パスは、このマシンに相対して指定します。ファイルは、指定した名前および数値接尾辞付きで作成されます。</p> <p>たとえば、C:\Program Files\Oracle BAM Enterprise Link\sort を指定し、ソート処理で3つの一時ファイルが必要な場合は、sort1、sort2 および sort3 というファイルがこのディレクトリに作成されます。</p> <p>パスを指定しない場合は、Enterprise Link Server 上の TEMP 環境変数で指定されたディレクトリが使用されます。</p>

**注意:** ユーザー入力の SQL を作成する場合は、別のダイアログ・ボックスが表示されます。「Enter Column To Sort」フィールドに正確な列名を入力し、「Add」をクリックする必要があります。

## Disk Sort Transform に対するヒント

- 「Max Blocks in Memory」を設定する場合は、Data Flow Service の設定および実行しているその他のプログラムについて考慮します。この値は、実行しているプログラムに対応できる十分大きい値に設定してください。
- 「Max Threads」を設定する場合は、Data Flow Service を調べて使用可能なスレッドの数を確認します。最大スレッド数を使用すると、パフォーマンスが最大になります。
- パフォーマンスを最大にするには、Data Flow Service と同じマシンに一時ファイルを配置する必要があります。
- Windows NT パフォーマンス モニタとともに Data Flow Service カウンタを使用すると、メモリーおよびスレッドの値が Oracle BAM Enterprise Link Server 上のパフォーマンスに影響します。

Data Flow Service での監視方法の詳細は、『Oracle Business Activity Monitoring Enterprise Link 管理者ガイド』を参照してください。

## Disk Sort の特性

パッケージ・タイプ	Basic
スキーマの変更	実行しない
マルチパスまたはシングルパス	マルチパス
コピーまたはインプレース	コピー
入力のソート	実行する
プロパティの公開	一部

## Expression Calculator

次のタスクを実行する場合に Expression Calculator を使用します。

- データを操作して必要な正確なデータ出力を取得する式の作成
- 列の追加、列の並替えおよび出力に送信する列の選択のすべての 1 ステップでの実行

ここで説明する内容は次のとおりです。

- 「Expression Calculator」ダイアログ・ボックスの設定
- 式の作成
- Expression Calculator に対するヒント
- Expression Calculator の特性

### 「Expression Calculator」ダイアログ・ボックスの設定

Expression Calculator を定義するには、「Expression Calculator」ダイアログ・ボックスを表示し、表 54 に従って設定項目を指定します。

図 44: 「Expression Calculator」ダイアログ・ボックス

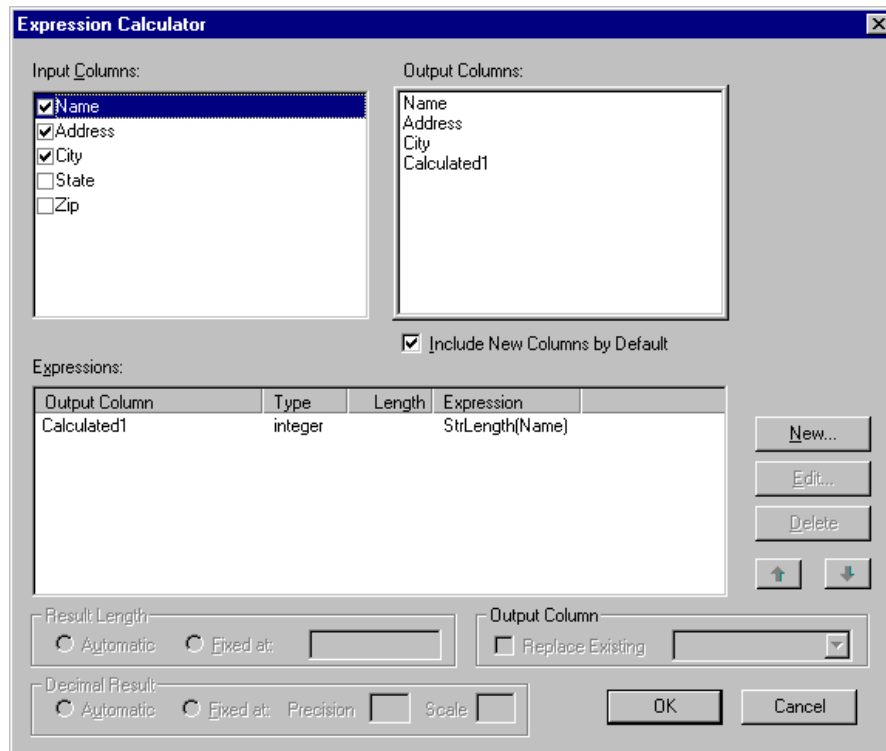




表 54: 「Expression Calculator」の設定

設定	説明
Input Columns	ここには、この時点でのデータ・フロー内のすべての列が表示されます。入力列のチェック・ボックスを選択解除すると、その列は、出力に含まれなくなるため、データ・フローから削除されます。式には含めたままにしておくことができます。出力列は、すべてのチェック・ボックスを選択解除した後、出力する順序で再選択して、並べ替えることができます。デフォルトでは、すべてのレコードが選択されています。
Output Columns	ここには、データ・フローに含められる列が表示されます。「Input Columns」リスト内のチェック・ボックスによって、列が「Output Columns」リストに表示されるかどうか指定されます。
Include New Columns by Default	このオプションによって、出力を変更せずにプランに新しい列を追加できます。デフォルトでは、このチェック・ボックスは選択されていません。
Expressions	ここには、計算されたすべての列の名前、データ型、長さ（該当する場合）および式が表示されます。decimal フィールドの場合、「Length」列の値は「Precision」の値より2（記号用に1列、小数点用に1列）大きくなります。ただし、「Precision」が「Scale」と等しい場合、「Length」の値は「Precision」の値より3大きくなります（追加の列に、小数点の左側に0（ゼロ）が保持されます）。
New	このボタンは、Expression Builder を使用して新しく計算された列を作成する場合にクリックします。同じ Transform 内に複数の計算された列を作成できます。
Edit	計算された列を式リストから選択し、このボタンをクリックして Expression Builder を表示します。Expression Builder を使用して式を編集します。Expression Builder は、列名をダブルクリックして表示することもできます。
Delete	このボタンは、選択した式を削除する場合にクリックします。
矢印ボタン	これらのボタンは、式を上下に移動する場合に使用します。順序は非常に重要です。式は、計算された別の列によって参照される場合、その式を参照する計算された列の前（上）に位置している必要があります。
Automatic	これは、結果に基づいてフィールドの長さを平均する場合に選択します。
Fixed at	これは、結果を固定長にする場合に選択します。フィールドの長さを設定する必要があります。string データ型の列の場合、列は、最大結果値にデフォルトで自動的にサイズ変更されます。列サイズを固定文字数に設定するには、「Output Column」リスト内の列名をクリックします。その後、「Fixed at」ラジオ・ボタンを選択し、値を入力します。この長さを超えるすべての文字が切り捨てられます。

表 54: 「Expression Calculator」 の設定 (続き)

設定	説明
Replace Existing	このチェック・ボックスは、既存の出力列を新しく計算された列の結果と置換する場合に選択します。置換する列をドロップダウン・リストから選択する必要があります。
Decimal Result	「Precision」および「Scale」の値は、手動で指定するか、または自動的に計算できます。 このセクションは、decimal フィールドを選択し、Decimal Math Provider を有効にするとアクティブになります。コンピュータで「Precision」および「Scale」を決定する場合は「Automatic」、「Precision」および「Scale」に独自の値を入力する場合は「Fixed at」を選択します。 decimal フィールドおよび decimal フィールドの計算の詳細は、150 ページの「10 進演算計算」を参照してください。

**注意:** ユーザー入力の SQL を作成する場合は、別のダイアログ・ボックスが表示されます。「Enter Column To Sort」フィールドに正確な列名を入力し、「Add」をクリックする必要があります。

## 式の作成

式を作成するには、「New」をクリックして「Expression Builder」ダイアログ・ボックスを表示し、表 55 に従って設定項目を指定します。

図 45: 「Expression Builder」ダイアログ・ボックス

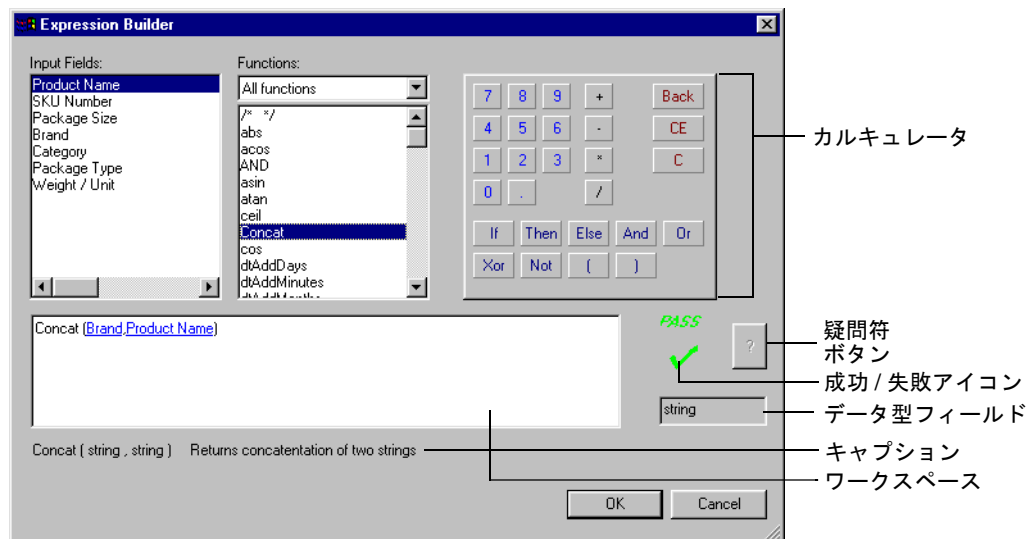


表 55: 「Expression Builder」 の設定

設定	説明
Input Fields	ここには、すべての入力フィールドが表示されます。「Input Fields」リスト内のアイテムをダブルクリック、入力またはドラッグしてワークスペースに移動して式で使用します。
Functions	ここには、作成する式に適用可能な関数が表示されます。「Functions」リスト内のアイテムをダブルクリック、入力またはドラッグして式に追加します。各関数の詳細は、付録 A「Expression Calculator の関数」を参照してください。
カルキュレータ	表示されている関数を使用して、計算式を作成できます。カルキュレータ・キーボードを使用して、式に定数、セパレータおよび論理演算子を追加します。このキーボードは、他の構文要素を追加する場合も使用します。 「Back」で、直前の文字を削除します。 「CE」で、式を保存して以降最新の変更を消去します。 「C」で、この式のすべての構文を削除します。
ワークスペース	この領域には、作成中の式が表示されます。
キャプション	関数を選択すると、正しい構文がキャプションに表示されます。入力フィールドを選択すると、そのフィールドのデータ型がキャプションに表示されます。
成功 / 失敗アイコン	これで、構文が正しいかどうかを確認します。式に正しい構文が使用されている場合は、緑色のチェック・マークが表示されます。式に正しくない構文が使用されている場合は、赤色の失敗アイコンが表示されます。式の構文が正しい場合は、自動解析機能によって、ダイアログ・ボックスの右側に緑色のチェック・マークが表示されます。式の構文が正しくない場合は、赤色の失敗アイコンが表示されます。構文の修正方法に関するヒントを表示するには、疑問符ボタンをクリックします。構文が正しい場合、疑問符ボタンはグレーアウトされています。
疑問符ボタン	式が失敗した場合は、このボタンをクリックして対応する構文に関するヒントを表示します。
データ型フィールド	このボックスには、式のデータ型が表示されます。

**注意：**「OK」をクリックして式を保存すると、Calculatedn というデフォルト名 (n は正の整数) の新しい列が「Output Column」リストに表示されます。この列は、ワークスペース内の Grid またはその他のシンクに表示されません。

## Expression Calculator に対するヒント

- 「Expression Calculator」では、プランのプロパティが入力列として表示されます。
- フィールドの名前を変更する場合は、Rename Columns Transform を使用してリソースを保存します。
- 大規模な If-Then-Else 文を作成する場合は、各セクションで自動解析が正常に実行され、新しいセクションが開始される前に緑色のチェック・マークが表示されていることを確認します。
- 正規表現を含める必要がある場合は、Search & Replace Transform を使用します。
- 連結レジストリ設定については、96 ページの「NULL 値に対する Concatenation の動作」を参照してください。
- If-Then-Else 文または Switch 文を使用する場合は、すべての出力が同じデータ型であることを確認します。

### 式の名前の変更

列の名前を変更するには、次の手順を実行します。

1. デフォルト名を右クリックし、「Rename」を選択します。名前が編集可能なフィールドになります。
2. 新しい名前を入力し、[Enter] を押します。

### 式の編集

式を編集するには、次の手順を実行します。

1. 「Expression」リスト内の列名をダブルクリックするか、または列名を選択して「Edit」をクリックします。

Expression Builder を使用して列を編集します。

2. 編集の完了後、「OK」をクリックします。

式を削除するには、次の手順を実行します。

- 列名を選択して「Delete」をクリックするか、または列名をダブルクリックして「Delete」を選択します。

## 式の順序の変更

式は、「Expressions」リストに表示されている順序で処理されます。式で他の式の結果を参照できます。このため、他のデータに対する依存性が含まれている結果を生成する場合は、式の順序を並べ替えます。

式の順序を変更するには、次の手順を実行します。

1. 「Expression Calculator」ダイアログ・ボックスを表示し、列を選択します。
2. 上下矢印ボタンをクリックして、リスト内で列を上下に移動します。
3. 「OK」をクリックして列を保存し、ダイアログ・ボックスを閉じます。

## Switch 文の使用

ネストされた If-Then-Else 文を置換するには、Switch 文を Expression Builder に入力します。

たとえば、次の文を置換するとします。

```
IF (column_name=="business")
THEN ("B")
ELSE (IF (column_name=="mod-cook")
THEN ("M")
ELSE (IF (column_name=="popular_comp")
THEN ("P")
ELSE (IF (column_name=="psychology")
THEN ("H") ELSE ("NULL_STRING"))))
```

この文は、次の CASE 文として記述することができます。

```
SWITCH (column_name)
Case ("business"): ("B")
Case ("mod-cook"): ("M")
Case ("popular_comp"): ("P")
Case ("psychology"): ("H")
Default (NULL_STRING)
```

## Expression Calculator の構文

次の一般的なルールは、すべての式構文に適用されます。

- 値を文字列値またはリテラルとして指定し、入力フィールドとして解釈されないようにするには、その文字列を一重引用符または二重引用符で囲みます。

**注意:** 文字列に一重引用符が含まれている式文字列またはリテラル値を使用している場合は、二重引用符を使用します。たとえば、'Southeast's Regional RTMs' ではなく、"Southeast's Regional RTMs" を使用します。この場合、一重引用符を使用するとエラーが戻されます。

- 演算子にオペランドが必要な場合は、そのオペランドをカッコで囲みます。他の式にネストされた式では、優先順位または式が評価される順位を示すために追加のカッコが必要です。自動解析機能では、カッコが適切に配置されるまで式は正常に解析されません。

式のタイプは、式によって戻されるデータの種類によって決まります。式によって整数値が戻される場合、その式は **integer** 型です。式によって文字列値が戻される場合、その式は **string** 型です。Expression Calculator Transform では、各式の出力タイプが評価され、正しいデータ型が自動的に割り当てられます。

## Expression Calculator の特性

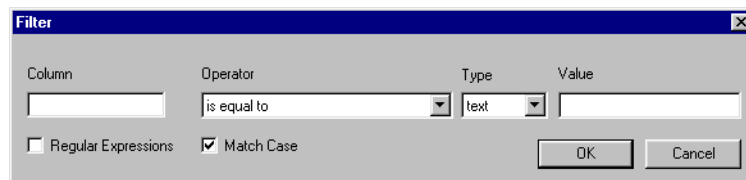
パッケージ・タイプ	Basic
スキーマの変更	実行する
マルチパスまたはシングルパス	シングルパス
コピーまたはインプレース	両方
入力のソート	実行しない
プロパティの公開	なし

## Filter

Filter は、データ・フロー内の次のステップに渡される入力列に対して条件または制約を指定する場合に使用します。このタイプのフィルタは、SQL の WHERE 句に制約を追加する場合と類似していますが、リレーショナル・データベースとは関係ありません。プランに複数の Filter Transform が含まれている場合、指定したすべての条件が適用されます。

リレーショナル・データベースのコマンドを使用して SQL フィルタを作成するには、「Filters」ツールバーの「Show/Hide Filters」ボタンをクリックしてプランにフィルタを追加します。

図 46: 「Filter」ダイアログ・ボックス



ここで説明する内容は次のとおりです。

- 「Filter」ダイアログ・ボックスの設定
- Filter Transform に対するヒント
- Filter の特性

## 「Filter」 ダイアログ・ボックスの設定

Filter を定義するには、次の表に従って設定項目を指定します。

表 56: 「Filter」 の設定

設定	説明
Column	フィルタする列を選択します。このドロップダウン・リストにすべての列（集計された列を含む）が表示されます。
Operator	比較演算子を選択します。
Type	比較タイプを選択します。表示されるタイプは、選択した列のデータ型によって異なります。
Value	選択したタイプに基づいて、比較値を指定します。 「number」タイプを選択した場合は、フィルタする数値を「Value」フィールドに入力します。 「column」タイプを選択した場合は、比較する列を「Value」ドロップダウン・リストから選択します。 「text」タイプを選択した場合は、フィルタするテキストを「Value」フィールドに入力します。フィルタ基準を拡張するには、テキスト文字列に正規表現を使用します。正規表現は、ワイルドカードと類似していて、大/小文字が区別されます。
Regular Expressions	「Regular Expressions」チェック・ボックスは、1つ以上の正規表現を値に含める場合に選択します。このチェック・ボックスを選択した場合は、比較演算子として「is equal to」または「is not equal to」のいずれかを選択します。
Match Case	「Match Case」チェック・ボックスは、結果の大/小文字の区別を入力の大/小文字の区別に一致させる必要がある場合に選択します。

## Filter Transform に対するヒント

- 比較を実行する Filter Transform およびその他の Transform では、Oracle の浮動小数点がサポートされていません。
- 正規表現の詳細は、79 ページの「正規表現」を参照してください。

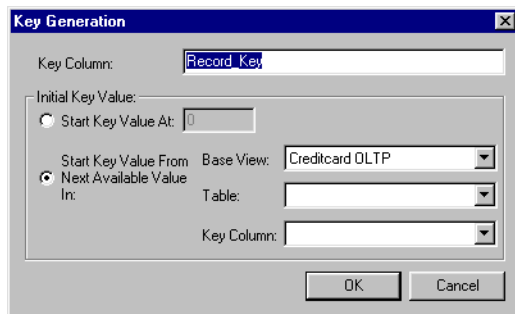
## Filter の特性

パッケージ・タイプ	Basic
スキーマの変更	実行しない
マルチパスまたはシングルパス	シングルパス
コピーまたはインプレース	コピー
入力のソート	実行しない
プロパティの公開	すべて

## Key Generation

Key Generation は、ディメンション表への移入時に使用する主キーを生成する場合に使用します。

図 47: 「Key Generation」 ダイアログ・ボックス



ここで説明する内容は次のとおりです。

- 「Key Generation」 ダイアログ・ボックスの設定
- Key Generation Transform に対するヒント
- Key Generation の特性

### 「Key Generation」 ダイアログ・ボックスの設定

Key Generation を定義するには、次の表に従って設定項目を指定します。

表 57: 「Key Generation」 の設定

設定	説明
Key Column	ディメンション表内のキー列の名前を入力します。この列は、integer 型である必要があります。ディメンション表にこの列が存在しない場合は生成されます。
Start Key Value At	新しいキーを生成するか、または既存のキー値を上書きする場合は、「Start Key Value At」ラジオ・ボタンを選択します。0（ゼロ）以外の値でディメンション・キーの生成を開始するか、またはディメンション・レコードを追加するには、開始値を入力します。
Start Key Value From Next Available Value in	レコードをディメンション表に追加する場合は、キー列から継承された値でディメンション・キーの生成を開始します。「Start Key Value From Next Available Value in」ボタンを選択します。
BaseView	キー列を持つ表が含まれている BaseView を「BaseView」ドロップダウン・リストから選択します。
Table	キー列が含まれている表の名前を「Table」ドロップダウン・リストから選択します。
Key Column	キー値を開始する列の名前を「Key Column」ドロップダウン・リストから選択します。 Key Generation Transform では、この列に最大整数値が検出されます。新しい行は、この値に続く値を使用してディメンション表に追加されます。



## Key Generation Transform に対するヒント

Key Generation Transform が含まれているプランを実行する場合は、データ・フロー内の各入力レコードに対して新しい連続整数値が生成されます。

## Key Generation の特性

パッケージ・タイプ	Basic
スキーマの変更	実行する
マルチパスまたはシングルパス	シングルパス
コピーまたはインプレース	コピー
入力のソート	実行しない
プロパティの公開	一部

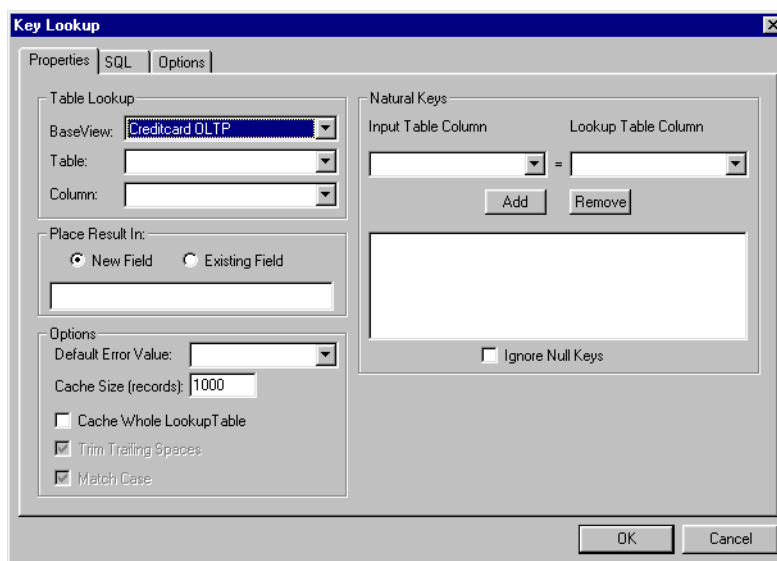
## Key Lookup

Key Lookup Transform では、ターゲット・ディメンション表内の主キー値を使用して外部キー列に移入します。ファクト表の前にディメンション表に移入するため、これらの主キー値はすでに存在します。ファクト表にロードするレコードは、古いキー値で識別されます。レコードをロードする前に、古いキー値を対応する新しいキー値に変換する必要があります。

Key Lookup Transform は、次のタスクにも使用できます。

- 検証表に対するデータの検証
- 緩やかに変化しているディメンション表内の特定の行の確認
- 状態コードを使用した状態名の検出などの単純な検索の実行

図 48: 「Key Lookup」 ダイアログ・ボックス



ここで説明する内容は次のとおりです。

- 「Key Lookup」ダイアログ・ボックスの設定
- Key Lookup Transform に対するヒント
- Key Lookup の特性

## 「Key Lookup」ダイアログ・ボックスの設定

Key Lookup を定義するには、次の表に従って設定項目を指定します。

表 58: 「Key Lookup」の設定

設定	説明
BaseView	検索表が含まれている BaseView を「BaseView」ドロップダウン・リストから選択します。通常、これはターゲット・ディメンション表です。
Table	ディメンション表の名前を「Table」ドロップダウン・リストから選択します。
Column	検索する値が含まれている列の名前を「Column」ドロップダウン・リストから選択します。ほとんどの移入では、これは、ターゲット・ディメンション表内の主キー列です。
New Field	「New Field」は、ファクト表内の新しい列に結果値を格納する場合に選択します。列名を指定します。「New Field」を選択し、この名前の列が存在する場合は、既存のフィールドが使用されます。存在しない場合は作成されます。
Existing Field	「Existing Field」は、既存の列を使用する場合に選択します。列名を指定します。
Default Error Value	検索エラーが発生したファクト表のキー・フィールドに値を挿入する場合にデフォルトのエラー値を指定します。検索エラーは、ディメンション表の主キー列内のフィールド値がソース表内のフィールド値と一致しない場合に発生します。この値によって、検索エラーが発生したレコードにフラグが設定されます。デフォルトは NULL 値です。任意の値を指定できます。
Cache Size (records)	キャッシュ・サイズをレコード数で指定します。
Cache Whole Lookup Table	「Cache Whole Lookup Table」は、データ・フローのこのステップの実行時に、Oracle BAM Enterprise Link Server 上のメモリーにディメンション表をロードする場合に選択します。キー列のレコードがメモリーに収まる場合は、これによって検索パフォーマンスが向上します。
Trim Trailing Spaces	値の後の空白を削除する場合に選択します。
Match Case	「Match Case」チェック・ボックスは、結果の大 / 小文字の区別を入力の大 / 小文字の区別に一致させる必要がある場合に選択します。
Input Table Column	ソース表内の自然キー列の名前を「Input Table Column」ドロップダウン・リストから選択して、検索処理の入力として使用します。これは、ソース内の古い自然キーです。

表 58: 「Key Lookup」の設定 (続き)

設定	説明
Lookup Table Column	ターゲット・ディメンション表内の自然キー列の名前を「Lookup Table Column」ドロップダウン・リストから選択します。これは、ターゲット・ディメンション表での参照用に格納されている古いキー列です。
Add	「Add」は、ソース内の古いキー列をターゲット内の古いキー列にマップする場合にクリックします。
Remove	マッピングを削除する場合にクリックします。
Ignore Null Keys	NULL キーが含まれているレコードを無視する場合に選択します。

## Key Lookup Transform に対するヒント

キー検索処理では、ソース・レコードの古いキー値とディメンション表に現在格納されている古いキー値が比較されます。一致が検出されると、Key Lookup Transform によって、そのレコードの新しいキー値が判別され、その新しい値がファクト表に挿入されます。

ファクト表に移入する場合、一連のキー検索によって、外部キー列に挿入された値が確認されます。各入力レコードに、各ディメンション表からの主キー値が割り当てられます。このキー値の組合せによって、レコードが一意に識別され、ファクト表の連結キーが形成されます。ファクト表内のキー列は、ディメンション表のキーが検出されるまで空です。

ターゲット・ディメンション表によって、古いキー値と新しいキー値の間の接続が提供されます。ディメンション表に移入する場合、古いキーを表す Part をプランに含めます。新しいキーが生成され、古いキーは新しい表での参照用に格納されます。

Key Lookup Transform は、debug モードで実行できます。このモードでは、Transform はデータベースにアクセスせずに実行されます。各行に対してデフォルト値が戻されます。これは、プランをデバッグし、パフォーマンスの問題を検出する場合に有効です。

「SQL」および「Properties」タブの値には、一貫性が保持されている必要があります。検索 SQL を編集した後に「Properties」内の値を変更すると、「Properties」タブ内の新しい値に基づいて SQL を再生成するように求められます。

- Key Lookup Transform では、各検索に対して 0 (ゼロ) または 1 のいずれかが戻される必要があります。複数の行が戻された場合は、エラーが発生します。この場合、検索表内の重複行を削除するか、または 1 行のみが戻されるように生成された SQL を変更します。たとえば、文に MAX を含めます。
- 新しいレコードに対して Key Lookup を実行するディメンション・ロード・プランでは、SQL Query によって後続の空白が切り捨てられません。SQL Query の後に Expression Calculator をデータ・フローに追加し、Key Lookup 用の入力フィールドで Right Trim を実行できます。

## Key Lookup の特性

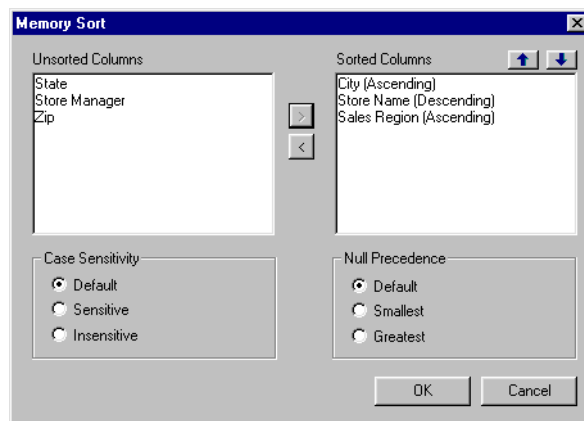
パッケージ・タイプ	Basic
スキーマの変更	実行する
マルチパスまたはシングルパス	シングルパス
コピーまたはインプレース	両方
入力のソート	実行しない
プロパティの公開	一部

## Memory Sort

Memory Sort は、指定した列に基づいて昇順または降順でデータ・フロー内のレコードをソートする場合に使用します。

Data Flow Service では、データ・フロー内の SQL Query ソースと Memory Sort ステップ間のステップに応じて、レコードのソートがメモリー内で実行されるか、または SQL の ORDER BY 句を SQL Query ソースに追加して実行されます。Memory Sort Transform は、最良のソート方法を選択するように最適化されています。

図 49: 「Memory Sort」 ダイアログ・ボックス



ここで説明する内容は次のとおりです。

- [「Memory Sort」 ダイアログ・ボックスの設定](#)
- [Memory Sort Transform に対するヒント](#)
- [Memory Sort の特性](#)

## 「Memory Sort」ダイアログ・ボックスの設定

Memory Sort を定義するには、次の表に従って設定項目を指定します。

表 59: 「Memory Sort」の設定

設定	説明
Unsorted Columns	結果セットの順序付けの基準となる列の名前をクリックします。その列を「Sorted Columns」リストにドラッグするか、または右矢印ボタンをクリックします。
Sorted Columns	列のソート順序を変更するには、列をクリックした後、「Ascending」または「Descending」ボタンのいずれかをクリックします。昇順がデフォルト値です。
「Case Sensitivity」の「Default」	データベース・サーバーでデフォルトの大 / 小文字の区別方法を使用してレコードをソートする場合（推奨）は、「Default」を選択します。
「Sensitive」または「Insensitive」	大 / 小文字の区別を指定する必要がある場合にのみ「Sensitive」または「Insensitive」を選択します。この値が RDBMS のデフォルト値と競合する場合、ソートは Data Flow Service で実行されます。
「Null Precedence」の「Default」	REBMS でデフォルトの NULL 処理方法を使用してレコードをソートする場合（推奨）は、「Default」を選択します。
「Smallest」または「Greatest」	NULL 値の順序付けを指定する必要がある場合にのみ「Smallest」または「Greatest」を選択します。この値が RDBMS のデフォルト値と競合する場合、ソートは Data Flow Service で実行されます。

## Memory Sort Transform に対するヒント

- 「Sorted Columns」リスト内の列の順序によって、ソートが適用される順序が決定されます。各列に、異なるソート順序を指定できます。「Sorted Columns」リスト内の最初の列が最初にソートされます。他の列は、この列に相対して順序付けられます。
- RDBMS でソートを実行できない場合、ソート順序には Oracle BAM Enterprise Link Admin 内のこの BaseView の CaseSensitiveSort 設定が反映されます。詳細は、『Oracle Business Activity Monitoring Enterprise Link 管理者ガイド』を参照してください。
- RDBMS でソートを実行できない場合は、Oracle BAM Enterprise Link Admin 内の該当する BaseView の NullsFirst 設定によって、NULL 値がソート順序に配置される場所が決定されます。詳細は、『Oracle Business Activity Monitoring Enterprise Link 管理者ガイド』を参照してください。
- Splitter を Memory Sort Transform の前に配置した場合、Memory Sort Transform の後に配置した場合とは異なる Memory Sort 結果を受信することがあります。これは、ソートが、ある場合は内部で、その他の場合はデータベースで実行されるためです。
- 編集した SQL で Memory Sort Transform を使用するには、主ソート列として使用する列の名前を入力し、「Add」をクリックする必要があります。列の修飾名を使用する必要があります。この列名は、「Sorted Columns」リストに表示されます。

## Memory Sort の特性

パッケージ・タイプ	Basic
スキーマの変更	実行しない
マルチパスまたはシングルパス	マルチパス
コピーまたはインプレース	コピー
入力のソート	実行する
プロパティの公開	一部

## Message Digest

Message Digest Transform は、任意の型の入力データに基づいてキーを作成します。キーは、複数の異なるフィールドの値から導出された単一フィールドの値です。この新しいキー・フィールドは、索引フィールドとして使用されます。Message Digest Transform によって生成されたキーは（ほとんどの場合）主キーと同様に機能します。キーの作成後、そのキーを使用して、ファイル内のレコードのソートまたは1つ以上のデータベース間のレコードの一致検索を実行できます。

Message Digest Transform で生成されたキーを使用するレコード一致検索は、従来のマージ / パージ・レコード一致検索とは異なります。従来のマージ / パージ一致検索の利点は、ファジー・マッチングです。ファジー・マッチングでは、多くの異なるフィールドを参照してレコードが比較され、それらのレコードが同じかどうかに関して値が評価されます。Message Digest Transform で生成されたキーを使用してレコードを一致検索する場合は、各レコードのキー・フィールドのみが比較されます。キーのみを比較することによって、両方のレコードのデータが厳密に同じかどうかを確認されます。Message Digest Transform で生成されたキーを使用するレコード一致検索の利点として、レコードを構成するすべてのフィールドではなく、1つのフィールドのみを調べることによって、レコードが変更されているかどうかを確認できることがあげられます。1つのフィールドを比較してレコードを一致検索すると、各レコード内の複数のフィールドを比較する場合より大幅に高速になります。

また、Message Digest Transform は、別の Transform で使用するために RAW データを暗号化する場合にも使用できます。たとえば、別の Transform 内のフィールド・データを比較し、その RAW データを一般公開された状態で送信しない場合は、関連フィールドのキーを作成し、それらのキーを入力として他の Transform に送信します。次に、それらの入力キーが他の Transform の常駐キーと比較されます。入力データが傍受された場合でも、それらのキーによって元のデータは公開されません。生成されたキーから元のレコードへのリバース・エンジニアリングの可能性が大幅に低くなります。

## キーの構成

Message Digest Transform では、パブリック・ドメイン MD5 アルゴリズムを使用して RAW 入力データ・フィールドからキーが生成されます。これらのアルゴリズムによって、入力データを少し変更することで（1つのフィールドで小さな変更を行った場合でも）大幅に異なるキーを生成できる程度まで入力データが処理されます。ファイルをこのキー・フィールドでソートすると、少し異なる類似レコードは近接して配置されません。

逆に、類似キーは、非常に異なる初期データから生成することができます。異なるデータから同じキーが生成される可能性は非常に低いため、2つのキーが一致した場合は、それらのキーが同じデータから生成された可能性が高くなります。

ここで説明する内容は次のとおりです。

- [Message Digest Transform の設定](#)
- [Message Digest Transform の使用方法](#)

## Message Digest Transform の設定

「Message Digest Properties」ダイアログ・ボックスには、次のセクションがあります。

- Input Fields
- Output Field

図 50: 「Message Digest Properties」ダイアログ・ボックス: データ・ソースが定義されている場合

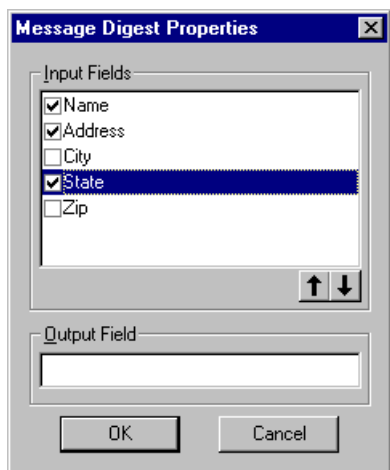


図 51: 「Message Digest Properties」 ダイアログ・ボックス: データ・ソースが定義されていない場合



「Message Digest Properties」の設定を定義するには、次の表に従って設定項目を指定します。

表 60: 「Message Digest Properties」の設定

設定	説明
Input Fields	<p>この領域は、キーの生成時に使用するフィールドを Transform に指示する場合に使用します。入力データ・ソースが定義されている場合は、入力データ・ソースのすべてのフィールドがこの領域に表示されます。これらのフィールドの1つ以上を選択して、キー・フィールドを作成できます。</p> <p>データ・ソースが定義されていない場合、Message Digest Transform では、入力データの構造（フィールド名など）が認識されません。この場合、この領域に表示されるフィールドは、キーを作成するために以前選択したフィールドです。</p> <p>選択したフィールドが連結され、処理されて出力キー・フィールドが生成されます。出力キー・フィールドは、選択した入力フィールドの数に関係なく、常に同じサイズ（32 バイト）です。任意の数のフィールドを選択できます。</p> <p>フィールドに含まれているデータの型が何であるかは問題ではありません。Message Digest Transform では、すべてのデータ型が処理されます。</p> <p>出力キーは、入力フィールドが連結された順序によって異なります。同じフィールド・データを異なる順序で連結すると、異なるキーが生成されます。</p>
Output Field	<p>このフィールドは、Message Digest Transform によって作成され、出力キーを含みます。テキスト・ボックスに名前を入力してフィールドの名前を選択します。</p>



## Message Digest Transform の使用方法

### Input Fields

データ・ソースを定義していない場合は、次のオプションがあります。

表 61: 入力フィールド・オプション（データ・ソースが定義されていない場合）

オプション	アクション
フィールド名の追加	キーの作成に使用する各フィールドに対して、データ・ソースのフィールドの名前を入力し、「Add」をクリックします。 データ・ソースの定義後、「Message Digest Properties」ダイアログ・ボックスに入力データ・ソース内のすべてのフィールドが表示されますが、手動で追加したフィールドのみがキーの作成時に使用するために選択されます。
フィールド名の削除	フィールド名をリストから削除するには、フィールドを選択し、「Remove」をクリックします。
連結順序の変更	フィールドが表示されている順序によって、フィールドの連結順序が決定されます。連結順序を変更するには、フィールドを選択し、上下矢印ボタンを押します。

データ・ソースを定義している場合は、次のオプションがあります。

表 62: 入力フィールド・オプション（データ・ソースが定義されている場合）

オプション	アクション
フィールド名の選択	フィールド名の左にあるボックスを選択して、キーの作成時に使用するフィールドを選択します。1つ以上のフィールドを選択できます。
連結順序の変更	フィールドが表示されている順序によって、フィールドの連結順序が決定されます。連結順序を変更するには、フィールド名をクリックしてフィールドを選択し、上下矢印ボタンを押します。

### Output Field

- 「Output Field」テキスト・ボックスに名前を入力して出力キー・フィールドの名前を選択します。

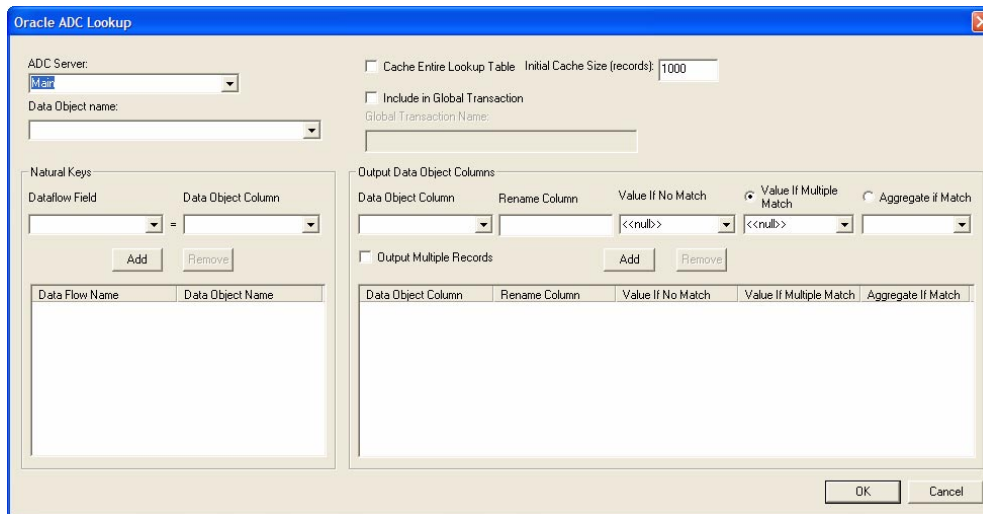
## Oracle BAM Lookup

Oracle BAM Lookup Transform を使用して、データ・フロー内のデータおよびデータ・オブジェクト内のデータからキー・フィールドに対して検索を実行します。入力データ・フロー・フィールドを複数の出力データ・オブジェクト列にマップできます。一致が検出されない場合または複数の一致が検出された場合は、出力する列を選択し、マップされたデータ・オブジェクト列の値を指定できます。

Oracle BAM Lookup Transform は、次のタスクにも使用できます。

- 検証表に対するデータの検証
- 緩やかに変化しているデータ・オブジェクト内の特定の行の確認
- 顧客コードを使用した顧客名の検出などの単純な検索の実行

図 52: 「Oracle BAM Lookup」 ダイアログ・ボックス



ここで説明する内容は次のとおりです。

- 「Oracle BAM Lookup」 ダイアログ・ボックスの設定
- Oracle BAM Lookup の特性

## 「Oracle BAM Lookup」 ダイアログ・ボックスの設定

Oracle BAM Lookup を定義するには、次の表に従って設定項目を指定します。

表 63: 「Oracle BAM Lookup」 の設定

設定	説明
ADC Server	ADC サーバー・マシン名を入力します。デフォルト名は Main です。
Data Object	検索処理に使用する既存のデータ・オブジェクトの名前を選択します。
Cache Entire Lookup Table	毎回ソース・データを使用するかわりに、検索結果をキャッシュする場合は、「Cache Entire Lookup Table」チェック・ボックスを選択します。これによって、緩やかに変化しているディメンションのパフォーマンスが向上します。ディメンションが絶えず変化している場合は、毎回新しいデータが使用されるようにキャッシュを無効にしておきます。
Initial Cache Size	キャッシュが有効になっている場合は、初期キャッシュ・サイズにレコード数を入力します。パフォーマンスを向上させるには、素数を選択します。競合を最小にするには、サイズを最大データ・セットより約 20% 大きくする必要があります。
Data Flow Field	ソース内の自然キー列の名前を「Data Flow Field」ドロップダウン・リストから選択して、検索処理の入力として使用します。
Data Object Column	ADC 内の自然キー列の名前を「Data Object Column」ドロップダウン・リストから選択します。「Add」をクリックします。ADC 内のデータ・オブジェクト列に対するソース内のデータ・フロー・フィールドのマッピングが「Add」ボタンの下のフィールドに表示されます。マップされた不要なキーを削除するには、「Remove」をクリックします。マップされたペアを編集するには、そのアイテムを削除し、再度追加します。

表 63: 「Oracle BAM Lookup」の設定

設定	説明
Output Data Object Columns	<p>「Output Data Object Columns」セクションでは、マップされたキー・ペアのみでなく、データ・オブジェクト・スキーマから出力するすべての列を追加する必要があります。一致が検出されない場合または複数の一致が検出された場合は、マップされたデータ・オブジェクト列の値を指定できます。その他の出力列に対しては、値を &lt;&lt;null&gt;&gt; のままにしておきます。</p> <p>データ・オブジェクト列を「Output Data Object Columns」ドロップダウン・リストから選択します。データ・フロー・フィールドとデータ・オブジェクト列間で一致が検出されない場合に挿入する値を選択します。この値は、NULL、-1、NA または任意に入力することができます。この値のデータ型は、列のデータ型と一致している必要があります。</p> <p>データ・フロー・フィールドとデータ・オブジェクト列間で複数の一致が検出された場合に挿入する値を選択します。この値は、NULL、-1、NA または任意に入力することができます。この値のデータ型は、列のデータ型と一致している必要があります。「Add」をクリックします。</p>
Remove	<p>データ・オブジェクト列の名前および指定した値が「Add」ボタン下のフィールドに表示されます。マップされた不要な出力を削除するには、「Remove」をクリックします。値設定を編集するには、そのアイテムを削除し、再度追加します。</p>

## Oracle BAM Lookup の特性

パッケージ・タイプ	Basic
スキーマの変更	実行しない
マルチパスまたはシングルパス	マルチパス
コピーまたはインプレース	コピー
入力のソート	実行する
プロパティの公開	一部

## Pivot

Pivot Transform は、データ・フロー内の列値を集計および入れ替える場合に使用します。この Transform を使用すると、行、列および値のサマリーが表示されるクロス集計形式に入力列を配置できます。また、フィールドを入力として使用して、表示されないようにすることもできます。

移入プランでは、この Transform は、データ・セット内の一意の値の識別、一意の値の集計および列への結果の配置を行う場合に有効です。

ここで説明する内容は次のとおりです。

- 「Pivot」ダイアログ・ボックスの設定
- Pivot Transform に対するヒント
- Pivot の特性

## 「Pivot」 ダイアログ・ボックスの設定

Pivot Transform を定義するには、ダイアログ・ボックスを表示し、次のタブを設定します。

- 「General」 タブ
- 「Advanced」 タブ

### 「General」 タブ

Pivot Transform を定義するには、「General」タブを選択し、表 64 に従って設定項目を指定します。

図 53: 「Pivot」 ダイアログ・ボックス

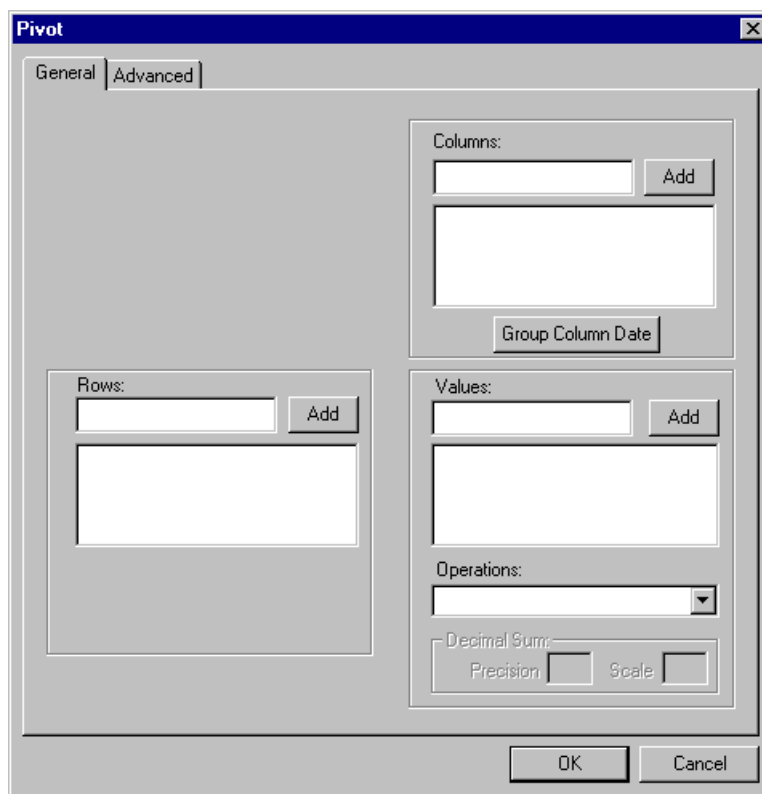


表 64: 「Pivot」 の設定

設定	説明
Columns	列名を入力します。
「Add」 ボタン	「Add」 ボタンは、アイテムをリストに追加する場合にクリックします。
Group Column Data	「Column」 リスト・フィールドが日時データ型の場合は、「Group Column Date」 をクリックして「Grid」列の日付のグループ化を選択します。
Rows	行名を入力します。

表 64: 「Pivot」 の設定 (続き)

設定	説明
Values	複数のフィールドを値として追加できます。また、同じ入力フィールドを値として複数回追加することもできるため、それらが検出されるたびに、別の処理を実行できます。
Operations	「Operations」フィールドの各値のサマリーを作成する処理を選択します。
Decimal Sum	このセクションでは、出力 decimal フィールドの精度を高める場合に decimal フィールドの「Precision」および「Scale」の値を手動で設定できます。 「Decimal Sum」セクションは、10 進演算が有効になっているかどうかに関係なく、decimal フィールドに対して合計処理がリクエストされるとアクティブになります。スキーマが不明の場合に表示されるダイアログ・ボックスでは、(フィールドが decimal 型の場合にスキーマが不明となるため) すべてのフィールドに対する合計処理に対して「Decimal Sum」セクションが有効になります。この情報は、出力フィールドが decimal 型でない場合、実行時に無視されます。 デフォルト値を変更する場合は、「Precision」および「Scale」に別の値を入力します。 decimal フィールドおよび decimal フィールドの計算の詳細は、150 ページの「10 進演算計算」を参照してください。

## 「Advanced」 タブ

Pivot Transform に拡張機能を設定するには、「Advanced」タブを選択し、表 65 に従って設定項目を指定します。

図 54: 「Advanced」タブのダイアログ・ボックス

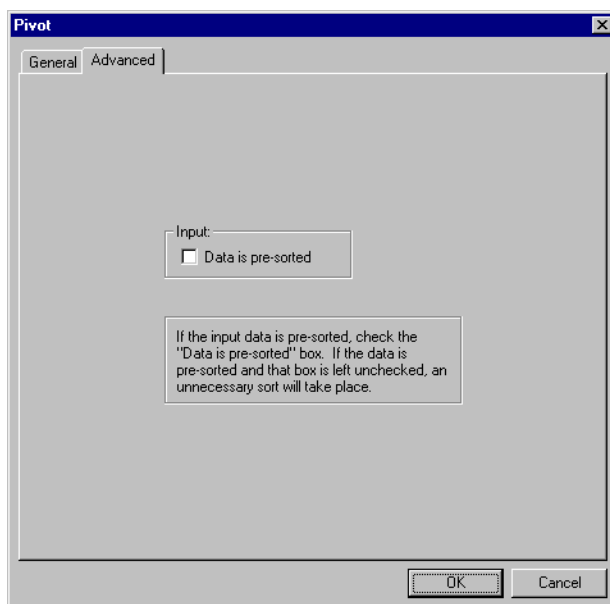


表 65: 「Advanced」タブの設定

設定	説明
Data is pre-sorted	このチェック・ボックスは、入力データがすでにソートされている場合に選択します。

## Pivot Transform に対するヒント

- 「Column」の値が指定されている場合、出力スキーマは表示されません。Add Columns Transform を使用して、Pivot Transform の後に出力スキーマを作成できます。
- 「Column」の値が指定されていない場合、出力スキーマが表示されます。
- 「Advanced」タブを使用して、すでにソートされているデータに対して不要なソートが実行されないようにします。不要なソートを行わないようにすると、パフォーマンスを向上させることができます。

## Pivot の特性

パッケージ・タイプ	Basic
スキーマの変更	実行する
マルチパスまたはシングルパス	マルチパス
コピーまたはインプレース	コピー
入力のソート	実行する
プロパティの公開	一部

## Rank

Rank Transform は、数値列の上位の行または下位の行を出力する場合に使用します。指定した数の上位または下位の列値、あるいはすべての行のうちの指定した上位または下位の割合（パーセント）に該当する行をランク付けできます。たとえば、列内の上位 10 個の値、または列内のすべての行のうちの上位 10% に該当する行を戻すことができます。

ここで説明する内容は次のとおりです。

- [「Rank」ダイアログ・ボックスの設定](#)
- [Rank の特性](#)

### 「Rank」ダイアログ・ボックスの設定

Rank Transform を定義するには、ダイアログ・ボックスを表示し、次のタブを設定します。

- [「General」タブ](#)
- [「Advanced」タブ](#)

## 「General」タブ

Rank Transform を定義するには、「General」タブを選択し、表 66 に従って設定項目を指定します。

図 55: 「Rank」ダイアログ・ボックス

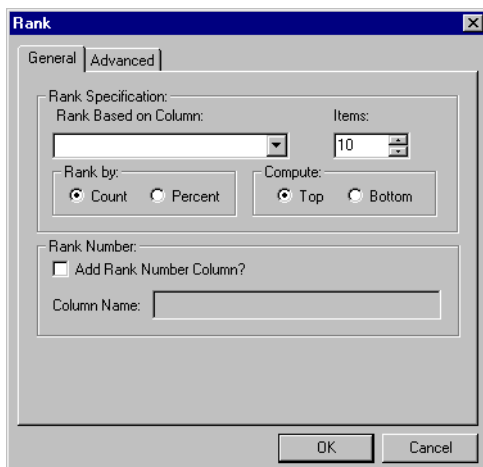


表 66: 「Rank」の設定

設定	説明
Rank Based on Column	ランク付けする列をドロップダウン・リストから選択します。
Items	出力するランク付けされた行の数、または「Items」フィールドに出力する行の割合（パーセント）を入力します。
Count	「Count」は、指定した数の上位または下位の値を戻す場合に選択します。
Percent	「Percent」は、行の合計数に対する指定した割合（パーセント）に該当する行を戻す場合に選択します。
Top	「Top」は、出力するセットの上位を指定する場合に選択します。
Bottom	「Bottom」は、出力するセットの下位を指定する場合に選択します。
Add Rank Number Column	「Add Rank Number Column」チェック・ボックスは、新しい列内のランク付けされた結果に番号を付ける場合に選択します。
Column Name	「Column Name」フィールドには、新しい列の名前を入力します。列の名前に、空白は含めないでください。

## 「Advanced」タブ

Rank Transform に拡張機能を設定するには、「Advanced」タブを選択し、表 67 に従って設定項目を指定します。

図 56: 「Advanced」タブのダイアログ・ボックス

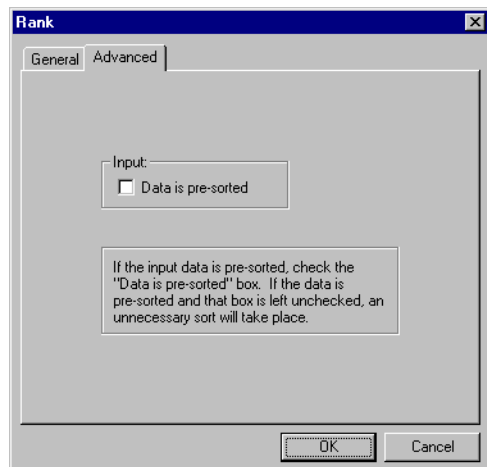


表 67: 「Advanced」タブの設定

設定	説明
Data is pre-sorted	このチェック・ボックスは、入力データがすでにソートされている場合に選択します。

## Rank の特性

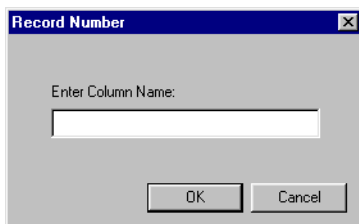
パッケージ・タイプ	Basic
スキーマの変更	実行する
マルチパスまたはシングルパス	マルチパス
コピーまたはインプレース	コピー
入力のソート	実行する
プロパティの公開	すべて



## Record Number

Record Number は、1 から始まる順序番号を表示するプランの列を追加する場合に使用します。Record Number Transform は、データ内の行数を表示する場合に有効です。

図 57: 「Record Number」 ダイアログ・ボックス



ここで説明する内容は次のとおりです。

- 「Record Number」 ダイアログ・ボックスの設定
- Record Number の特性

### 「Record Number」 ダイアログ・ボックスの設定

Record Number Transform を定義するには、ダイアログ・ボックスを表示し、次の設定を指定します。

表 68: 「Record Number」 の設定

設定	説明
Enter Column Name	新しいレコード番号列の名前を入力します。

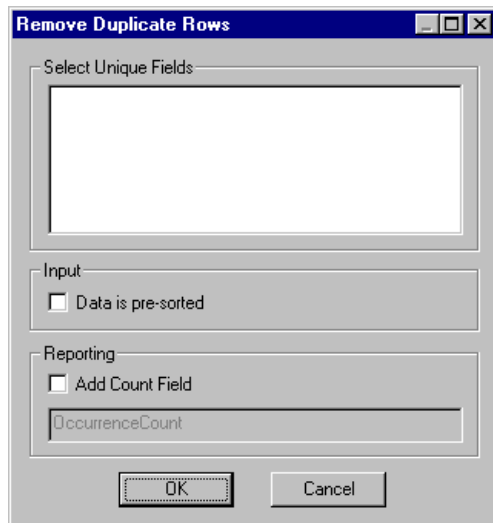
### Record Number の特性

パッケージ・タイプ	Basic
スキーマの変更	実行する
マルチパスまたはシングルパス	シングルパス
コピーまたはインプレース	コピー
入力のソート	実行しない
プロパティの公開	すべて

## Remove Duplicate Rows

Remove Duplicate Rows Transform は、重複レコードをレコードのリストから削除する場合に使用します。

図 58: 「Remove Duplicate Rows」 ダイアログ・ボックス



ここで説明する内容は次のとおりです。

- 「Remove Duplicate Rows」 ダイアログ・ボックスの設定
- Remove Duplicate Rows Transform に対するヒント
- Remove Duplicate Rows の特性

## 「Remove Duplicate Rows」ダイアログ・ボックスの設定

「Remove Duplicate Rows」の設定を定義するには、次の表に従って設定項目を指定します。

表 69: ダイアログ・ボックスの設定

設定	説明
Select Unique Fields	このオプションで、すべてのレコードに対して一意にする必要があるフィールドを選択できます。 たとえば、「PhoneNumber」フィールドを選択し、この Transform によって同じ電話番号を持つ複数のレコードが検出された場合は、最初のレコードが保存され、他のすべてのレコードは破棄されます。
Input	「Data is pre-sorted」オプションが選択されている場合、この Transform では、データが一意のフィールドに基づいてすでにソートされているとみなされます。データが事前にソートされている場合にこのオプションを選択しないと、不要なソートが実行され、実行時間が長くなります。
Reporting	「Add Count Field」が選択されている場合は、名前を定義した列が追加されます。このフィールドの値は、選択した一意のフィールドの値と同じ値が含まれているレコードの数です。 たとえば、1つのレコードに一意のフィールド内の一意の値が含まれている場合、このフィールドの値は1になります。3つのレコードに一意のフィールドの値と同じ値が含まれている場合、このフィールドの値は3になります。 この列の合計が、入力レコードの数になります。

## Remove Duplicate Rows Transform に対するヒント

- 一意であるとタグ付けするフィールドを選択します。1つ以上のフィールドを選択できます。複数のフィールドを選択した場合、2つのレコードが重複しているとみなされるには、選択したすべてのフィールドが一致する必要があります。
- レコードが一意のフィールドに基づいてすでにソートされている場合は、「Data is pre-sorted」チェック・ボックスを選択します。これによって、Transform のパフォーマンスが向上します。
- 選択した一意のフィールドの値と同じ値が含まれているレコードの数をレポートするフィールドを追加する場合は、「Add Count Field」チェック・ボックスを選択します。

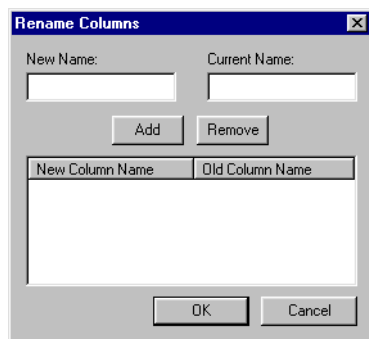
## Remove Duplicate Rows の特性

パッケージ・タイプ	Power Product
スキーマの変更	実行する
マルチパスまたはシングルパス	マルチパス
コピーまたはインプレース	コピー
入力のソート	実行する
プロパティの公開	一部

## Rename Columns

Rename Columns は、データ・フロー内の入力列の新しい名前を指定する場合に使用します。

図 59: 「Rename Columns」ダイアログ・ボックス



ここで説明する内容は次のとおりです。

- [「Rename Columns」ダイアログ・ボックスの設定](#)
- [Rename Columns Transform に対するヒント](#)
- [Rename Columns の特性](#)

## 「Rename Columns」ダイアログ・ボックスの設定

「Rename Columns」の設定を定義するには、次の表に従って設定項目を指定します。

表 70: 「Rename Columns」の設定

設定	説明
New Name	リスト内の新しい列名をクリックし、このフィールドに新しい名前を入力します。 編集した SQL に対して処理を行う場合、フィールド名は表示されません。
Current Name	編集した SQL に対して処理を行う場合は、「Current Name」フィールドに、Parts Bin に表示されているとおりに列名を入力します。
Apply	「Apply」は、マッピングを完了する場合にクリックします。
Add	編集した SQL に対して処理を行う場合は、「Add」をクリックしてマッピングを完了します。
Remove	編集した SQL に対して処理を行う場合は、「Remove」をクリックしてマッピングをリストから削除します。

## Rename Columns Transform に対するヒント

- この Transform は、名前ベースのマッピングを使用してデータベースにデータをロードし、ソース列とターゲット列に異なる名前が使用されている場合に有効です。この Transform をデータ・フロー内のシンク Transform の直前に追加して、新しい名前をそのシンクに渡します。また、この Transform は、計算フィールドの名前変更にも使用できます。
- この Transform は、単一の BaseView を使用して複数のターゲット表をロードし、それらのターゲット表の列に異なる名前が使用されている場合にも有効です。
- このステップの入力が「SQL Editor」ダイアログ・ボックスで編集された問合せの場合、既存の入力列は表示されないため、現行の列名および新しい列名を入力する必要があります。また、前の Transform によって Part の名前が Rename Columns Transform に適切に渡されていない場合またはワークスペースに Part が存在しない場合も、列名を入力する必要があります。

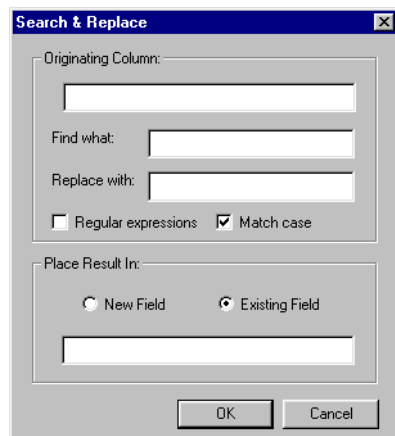
## Rename Columns の特性

パッケージ・タイプ	Basic
スキーマの変更	実行する
マルチパスまたはシングルパス	シングルパス
コピーまたはインプレース	インプレース
入力のソート	実行しない
プロパティの公開	一部

## Search & Replace

Search & Replace は、指定した文字列または部分文字列を列内で検索し、検出されたすべての個所でその文字列を新しい列または既存の列内の別の文字列に置換する場合に使用します。検索文字列の定義には正規表現を使用できます。また、検索文字列の大 / 小文字を区別して一致検索するかどうかも指定できます。

図 60: 「Search & Replace」 ダイアログ・ボックス



ここで説明する内容は次のとおりです。

- 「Search & Replace」 ダイアログ・ボックスの設定
- Search & Replace Transform に対するヒント
- Search & Replace の特性

## 「Search & Replace」ダイアログ・ボックスの設定

「Search & Replace」の設定を定義するには、次の表に従って設定項目を指定します。

表 71: 「Search & Replace」の設定

設定	説明
Originating Column	検索する入力列を「Originating Column」ドロップダウン・リストから選択します。スキーマが不明の場合または編集した SQL が存在する場合は、入力列の名前を入力します。
Find What	検索文字列を入力します。
Replace with	置換文字列を入力します。
Regular Expressions	「Regular expressions」チェック・ボックスは、検索文字列に正規表現を使用する場合に選択します。
Match case	「Match case」チェック・ボックスは、大 / 小文字を区別して「Find What」フィールドに入力したとおりに検索する場合に選択します。
New Field	「New Field」は、置換文字列を新しいフィールドに指定する場合に選択します。新しい列の名前を入力します。列の名前に、空白は含めないでください。
Existing Field	「Existing Field」は、置換文字列を既存のフィールドに指定する場合に選択します。列名を入力します。

## Search & Replace Transform に対するヒント

- 「Originating Column」ドロップダウン・リストには、string 型の列のみが表示されます。
- 検索基準を拡張するには、検索文字列に正規表現を使用します。
- 複数の検索置換処理を実行する場合は、Expression Calculator を使用するとパフォーマンスが向上します。

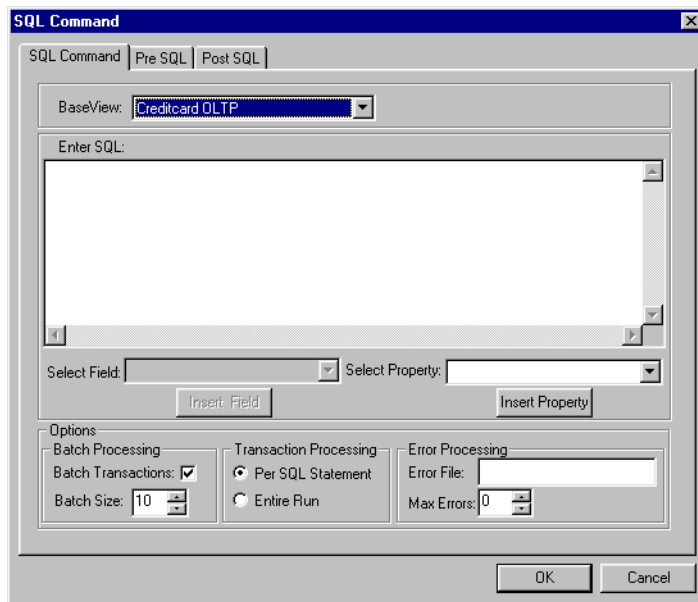
## Search & Replace の特性

パッケージ・タイプ	Basic
スキーマの変更	実行する
マルチパスまたはシングルパス	シングルパス
コピーまたはインプレース	コピー
入力のソート	実行しない
プロパティの公開	すべて

## SQL Command

SQL Command は、プラン内の各行に対してカスタマイズされた SQL コマンドを実行する中間ステップとして使用します。たとえば、この Transform を使用すると、ターゲットのディメンション表またはファクト表内のレコードを更新できます。この Transform によって、データ・フロー内の入力レコードの各バッチに対して指定した SQL が実行されます。

図 61: 「SQL Command」 ダイアログ・ボックス



ここで説明する内容は次のとおりです。

- 「SQL Command」 ダイアログ・ボックスの設定
- SQL Command の特性

### 「SQL Command」 ダイアログ・ボックスの設定

「SQL Command」の設定を定義するには、次の表に従って設定項目を指定します。

表 72: 「SQL Command」の設定

設定	説明
BaseView	SQL を実行する表（複数可）が含まれている BaseView を選択します。リポジトリ内の任意の BaseView を選択できます。
Enter SQL	実行する SQL 文を入力します。構文は、データベース・ベンダーの要件に準拠している必要があります。複数の文を実行する場合は、適切なデリミタ文字を使用して文を区切ります。単一の文の場合、デリミタは必要ありません。



表 72: 「SQL Command」の設定 (続き)

設定	説明
Select Field	SQL に挿入するデータ・フロー内の各フィールドをドロップダウン・リストから選択して、「Insert Field」をクリックします。
Insert Field	選択したフィールドを文に挿入する場合にクリックします。
Select Property	SQL に挿入する SQL Command ステップの各設定をドロップダウン・リストから選択して、「Insert Property」をクリックします。
Insert Property	選択した設定を文に挿入する場合にクリックします。
Batch Transactions	トラブルシューティングのためにバッチ処理を無効にする場合は、「Batch Transactions」チェック・ボックスの選択を解除します。
Batch Size	必要に応じて「Batch Size」の値を調整します。デフォルト値は 10 入力行です。最適なバッチ・サイズは、SQL 文および処理されるレコードの数によって異なります。
Per SQL Statement	「Per SQL Statement」または「Entire Run」をクリックしてトランザクションを処理します。「Per SQL Statement」オプションでは、各バッチが個別のトランザクションとしてデータベースに送信されるため、バッチは個々にコミットされます。多数のレコードを処理する場合は、「Per SQL Statement」をクリックし、バッチ・サイズを適切に調整します。通常、バッチ・サイズを大きくすると、より効率的になります。
Entire Run	「Entire Run」は、1 つの大規模トランザクションを作成する場合に選択します。このトランザクションは、すべてのバッチが受信された後にコミットされます。
Error File	Data Load Server 上のエラー・ファイルのパスおよび名前を入力します。パスは、Oracle BAM Enterprise Link Server に相対して指定します。ファイル名を指定しなかった場合、エラー・データは書き込まれません。
Max Errors	SQL 処理を停止するエラーの最大数を「Max Errors」フィールドに指定します。すべてのエラーを許容するには、非常に大きい数値を指定します。
Pre SQL	「Pre SQL」タブで、レコードを処理する前に実行する SQL を入力します。「Pre SQL」タブで SQL 文を追加して、処理するレコードの表を作成できます。
Post SQL	「Post SQL」タブで、レコードを処理した後に実行する SQL を入力します。「Post SQL」タブで SQL 文を追加して、レコードの処理後に索引を作成できます。

SQL 文の例を次に示します。

```
update jobs set    MAX_LVL = [[FIELD:MAX_LVL]], MIN_LVL =
  [[FIELD:MIN_LVL]] where  JOB_ID = [[FIELD:JOB_ID]]
```

SQL は、各行に対して 1 回生成されます。行内のジョブ ID が jobs という表内のジョブ ID と一致する場合は、最小レベルおよび最大レベルの値が更新されます。

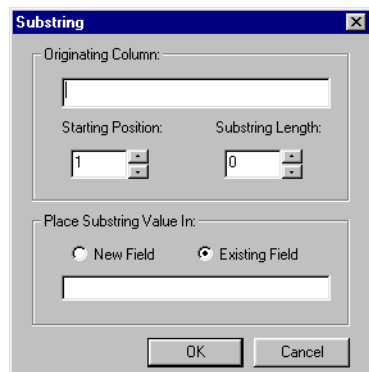
## SQL Command の特性

パッケージ・タイプ	Power Product
スキーマの変更	実行しない
マルチパスまたはシングルパス	シングルパス
コピーまたはインプレース	インプレース
入力のソート	実行しない
プロパティの公開	すべて

## Substring

Substring は、文字列値から部分文字列を抽出し、その結果を新しい列または既存の列に格納する場合に使用します。部分文字列は、開始位置および長さを指定して定義します。

図 62: 「Substring」 ダイアログ・ボックス



ここで説明する内容は次のとおりです。

- 「Substring」 ダイアログ・ボックスの設定
- Substring Transform に対するヒント
- Substring の特性

## 「Substring」ダイアログ・ボックスの設定

「Substring」の設定を定義するには、次の表に従って設定項目を指定します。

表 73: 「Substring」の設定

設定	説明
Originating Column	検索する入力列を「Originating Column」ドロップダウン・リストから選択します。
Starting Position	部分文字列の先頭の文字の位置（数値）を指定します。
Substring Length	部分文字列の長さを指定します。文字列内の残りのすべての文字を選択する場合は0（ゼロ）を入力します。
New Field	「New Field」は、計算された値の出力に列を追加する場合に選択します。新しいフィールドのデフォルトの名前を編集できます。
Existing Field	「Existing Field」は、既存のフィールドに出力を格納する場合に選択します。「Existing Field」で部分文字列より短い値を選択すると、部分文字列は切り捨てられます。

## Substring Transform に対するヒント

- 「Originating Column」ドロップダウン・リストには、string 型の列のみが表示されます。
- たとえば、ソース文字列 (164) 187-7451 から部分文字列 164 を定義する場合、1 はソース文字列の 2 番目の文字であるため、開始位置は 2 になります。

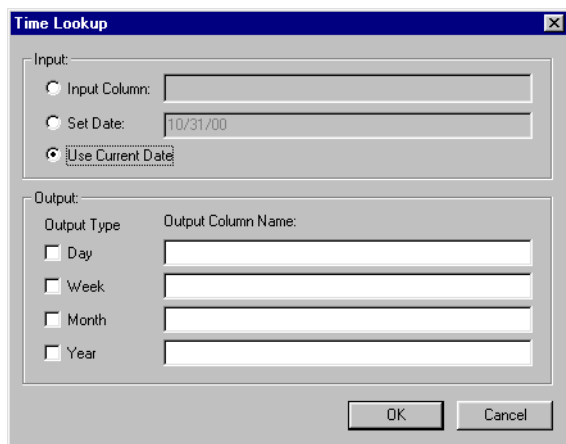
## Substring の特性

パッケージ・タイプ	Basic
スキーマの変更	実行する
マルチパスまたはシングルパス	シングルパス
コピーまたはインプレース	コピー
入力のソート	実行しない
プロパティの公開	すべて

## Time Lookup

「Time Lookup」は、ファクト表内の時間ディメンションの外部キー列に移入する場合に使用します。この Transform では、入力の日付値または日時値が読み取られ、指定した型の日付値および同等のユリウス日付値が出力されます。入力に対して、時間ディメンション・レコードの開始日、ファクト表の粒度および出力列名を指定します。Time Lookup Transform によって、同等の値が生成され、指定したファクト表の列に挿入されます。

図 63: 「Time Lookup」ダイアログ・ボックス



ここで説明する内容は次のとおりです。

- [Time Lookup Transform の設定](#)
- [Time Lookup Transform に対するヒント](#)
- [Time Lookup の特性](#)

### Time Lookup Transform の設定

「Time Lookup」の設定を定義するには、次の表に従って設定項目を指定します。

表 74: 「Time Lookup」の設定

設定	説明
Input Column	入力に対して、入力日時列の名前を入力します。
Set Date	すべての出力日の値を指定した日付に設定するには、「Set Date」を選択し、日付値を入力します。
Use Current Date	すべての出力日の値を現行の日付に設定するには、「Use Current Date」を選択します。
Output Type	出力に対して、生成する出力のタイプを選択し、ファクト表内の該当する列の名前を入力します。ファクト表の粒度と一致する出力のタイプを選択します。すべての出力のタイプが、4713 年 B.C.1 月から始まるユリウス日付値です。
Output Column Name	選択した各型の列の名前を入力します。ユリウス日付値を使用して「Grid」に表示する場合、出力のタイプには任意の名前を使用できます。

## Time Lookup Transform に対するヒント

- Key Lookup Transform または Time Lookup Transform のいずれかを使用して、ファクト表にユリウス日付値を移入できます。これらの2つの Transform では、同じタスクが異なる方法で実行されます。
- Time Lookup Transform では、日付値および同等のユリウス日付値がファクト表の外部キーとして生成されます。時間ディメンション・データに対してこの Transform を使用すると、パフォーマンスが向上します。
- ほとんどのファクト表には1つの日付列のみが含まれていますが、Time Lookup Transform を使用すると、4つのタイプの出力を生成することができます。たとえば、ユーザーが年に関して頻繁に問い合わせる場合は、日付値のみでなく年値も生成し、ファクト表と時間ディメンション表のこれらの列間の BaseView に連結結合を作成できます。この方法によって、ファクト表と時間ディメンション表間の結合の数が削減され、特定のタイプのプランのパフォーマンスが向上します。

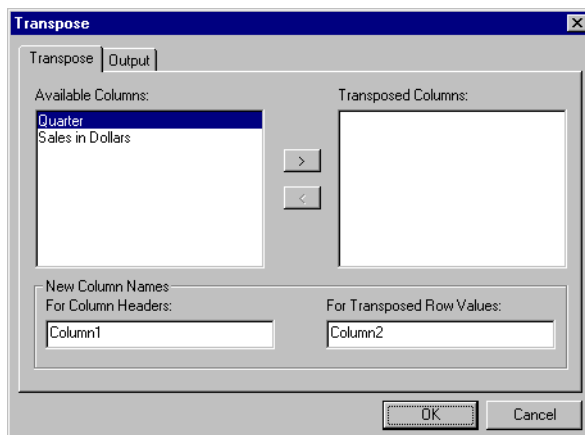
## Time Lookup の特性

パッケージ・タイプ	Power Product
スキーマの変更	実行する
マルチパスまたはシングルパス	シングルパス
コピーまたはインプレース	コピー
入力のソート	実行しない
プロパティの公開	なし

## Transpose

Transpose は、列データをデータ・フロー内の行に移す場合に使用します。データの入替えは、Pivot Transform を使用したデータのピボットの反対の処理です。これによって、行データが列に変換されます。

図 64: 「Transpose」ダイアログ・ボックス



ここで説明する内容は次のとおりです。

- 「Transpose」ダイアログ・ボックスの設定
- Transpose Transform に対するヒント
- Transpose の特性

## 「Transpose」ダイアログ・ボックスの設定

Transpose Transform を定義するには、ダイアログ・ボックスを表示し、次の設定を指定します。

表 75: 「Transpose」の設定

設定	説明
Available Columns	入れ替える列の名前をクリックし、その列を「Transposed Columns」リストにドラッグするか、または右矢印ボタンをクリックします。複数の列を選択するには、[Shift] と [Ctrl] キーを使用します。 この列名は、「Transposed Columns」リストに表示されます。複数の列を入れ替える場合は、それらの列の型に互換性がある必要があります。
Type Name of Columns to Transpose	ユーザー入力の SQL が存在する場合は、入れ替える列の名前を入力する必要があります。複数の列を入れ替える場合は、それらの列の型に互換性がある必要があります。
Add	ユーザー入力の SQL が存在する場合は、「Add」をクリックして、列を「Transposed Columns」リストに追加します。
Transposed Columns	「Transposed Columns」リストには、入れ替えられた列の名前が表示されます。
For Column Headers	既存の列ヘッダーが含まれている列の名前を入力します。この新しい列は、データ・フローに自動的に追加されます。この列のデフォルト名は Column1 です。
For Transposed Row Values	入れ替えられた行値が含まれている列の名前を入力します。この新しい列は、データ・フローに自動的に追加されます。この列のデフォルト名は Column2 です。
「Output」タブ	「Output」タブで、データ・フロー内の次の Transform に渡さない列を選択解除します。デフォルトでは、すべての入力列および生成された列がデータ・フロー内の次のステップに渡されます。

## Transpose Transform に対するヒント

- 1つ以上の列を入れ替えると、2つの新しい列がデータ・フローに追加されます。1つ目の列には、入れ替えられたすべての列のヘッダー（Part名）が含まれます。2つ目の列には、入れ替えられた列の値が含まれます。たとえば、「Sales in Dollars」を入れ替えると、プランの実行時に1つ目の新しい列には「Sales in Dollars」というヘッダーが含まれます。2つ目の列には、ドル値が含まれます。
- 複数の列を入れ替える場合は、それらの列の型に互換性がある必要があります。
- 同じデータ・フロー内で複数の入替え処理を実行するには、**Splitter Transform** を使用して重複データ・セットを作成します。データの各セットを入れ替えた後、**Union Transform** を使用してそれらのデータ・セットを1つにマージします。

### 列型の互換性に関する表

Transpose Transform の列型を一致検索するには、次の表を使用します。

表 76: 列型の互換性に関するマトリックス

	int	long	String	datetime	double	uint	binary	decimal
int	True	True	False	False	True	True	False	True
long	True	True	False	False	True	True	False	True
String	False	False	True	False	False	True	True	False
datetime	False	False	False	True	False	False	False	False
double	True	True	False	False	True	True	False	True
uint	True	True	False	False	True	True	False	True
binary	False	False	True	False	False	False	True	False
decimal	True	True	False	False	True	True	False	True

### Transpose の特性

パッケージ・タイプ	Basic
スキーマの変更	実行する
マルチパスまたはシングルパス	シングルパス
コピーまたはインプレース	コピー
入力のソート	実行しない
プロパティの公開	すべて

## VBScriptCopy

VBScriptCopy Transform は、次のタスクに使用します。

- 入力レコードのコピーおよび入力レコードのデータおよびその列属性の操作
- 列の追加または削除、あるいはサイズ、型、名前などの列属性の変更

VBScriptCopy Transform に設定項目はありません。ダイアログ・ボックスには、テキスト・フィールドがあります。ここに **VBScript** を入力して「OK」ボタンをクリックし、ダイアログ・ボックスを閉じて変更を保存します。

VBScript Transform の詳細は、『Oracle BAM Enterprise Link Design Studio ユーザーズ・ガイド』または Oracle BAM Design Studio のオンライン・ヘルプを参照してください。

VBScript Transform を使用してバイナリ・データ型の列は作成できません。

パッケージ・タイプ	Basic
スキーマの変更	実行する
マルチパスまたはシングルパス	シングルパス
コピーまたはインプレース	コピー
入力のソート	実行しない
プロパティの公開	なし

## VBScriptInplace

この Transform は、各レコードのインプレース処理に使用します。そのため、入力レコードのデータのみを変更できます。列の追加または削除、あるいは列属性の変更は実行できません。たとえば、10 文字の文字列が含まれている入力レコード内の列は、10 文字を超えて出力することはできません。ただし、10 文字未満で出力することはできます。

VBScriptInplace Transform に設定項目はありません。ダイアログ・ボックスには、テキスト・フィールドがあります。ここに **VBScript** を入力して「OK」ボタンをクリックし、ダイアログ・ボックスを閉じて変更を保存します。

VBScript Transform の詳細は、『Oracle BAM Enterprise Link Design Studio ユーザーズ・ガイド』または Oracle BAM Design Studio のオンライン・ヘルプを参照してください。

VBScript Transform を使用してバイナリ・データ型の列は作成できません。

パッケージ・タイプ	Basic
スキーマの変更	実行しない
マルチパスまたはシングルパス	シングルパス
コピーまたはインプレース	インプレース
入力のソート	実行しない
プロパティの公開	なし



---

# Expression Calculator の関数

ここで説明する内容は次のとおりです。

- 日時関数
- 文字列関数
- 数学関数
- 論理演算子
- バイナリ演算子
- その他の演算子
- カルキュレータの演算子

## 日時関数

次の関数は、日時値の一部を戻すか、2つの値間の間隔を計算するか、または新しい値を作成する場合に使用します。

表 77: 日時関数

演算子	説明	データ型の制限
dtAddDays (date, n)	date の値に日数 n を追加します。月の日数およびうるう年が考慮されます。	datetime のみです。n は整数です。
dtAddMinutes (date, n)	date の値に分数 n を追加します。月の日数およびうるう年が考慮されます。	datetime のみです。n は整数です。
dtAddMonths (date, n)	date の値に月数 n を追加します。月の日数およびうるう年が考慮されます。	datetime のみです。n は整数です。
dtAddYears (date, n)	date の値に年数 n を追加します。うるう年が考慮されます。	datetime のみです。n は整数です。
dtAge (date)	年齢 (date の値と現在の日付間の年数) を戻します。	datetime のみです。
dtCurrentDT ()	現在の日時を戻します。	入力はありません。
dtDateFromDouble (double)	double の値から日時値を作成します。月の日数またはうるう年は考慮されません。	double のみです。
dtDateFromJulian (long)	ユリウス日付値から日時値を作成します。時間値は 00:00:00 です。	long integer のみです。
dtDateFromPStr (str, format)	packed string の str から日時値を作成します。format は、mmddy または mmddyyyy の場合は 1、yyyymmdd の場合は 2 です。	string のみです。ここで、str の長さは 6 文字または 8 文字のいずれかです。
dtDateToDouble (date)	1899 年 12 月 30 日午前 0 時で始まる日時値から double 値を作成します。時間値は、1 日の小数部分として表現されます。月の日数またはうるう年は考慮されません。	datetime のみです。0100 年 1 月 1 日から 9999 年 12 月 31 日までの日付値です。
dtDateToJulian (date)	日時値からユリウス日付値を作成します。ユリウス日付は、4713 年 B.C. 1 月 1 日から始まります。時間値は失われます。	datetime のみです。

表 77: 日時間数 (続き)

演算子	説明	データ型の制限
dtDateToString (dateval, format)	指定した日付を文字列値として戻します。戻される文字列の書式は、format パラメータによって異なります。 format 文字列は、次のいずれかにできます。 MM/DD/YYYY MM/DD/YY M/D/YY DD-MM-YYYY DD-MM-YY D-M-YY MMDDYYYY MMDDYY DDMMYYYY DDMMYY YYYYMMDD YYMMDD	dateval は、datetime 型である必要があります。 format は、string である必要があります。
dtDayOfWeek (date)	曜日の値を戻します。1 が日曜日です。	datetime のみです。
dtDayOfYear (date)	通年での日付を表す数値を戻します。	datetime のみです。
dtDaysInMonth (month, year)	month の日数を戻します。	month および year は integer です。year が 2 桁の場合、 $\leq 29$ は 20xx、 $\geq 30$ は 19xx とみなされます。
dtDaysInYear (year)	year の日数を戻します。うるう年が考慮されます。	year が 2 桁の場合、 $\leq 29$ は 20xx、 $\geq 30$ は 19xx とみなされます。
dtDiffDays (d1, d2)	d1 と d2 の差 (日数) を double として戻します。	datetime のみです。
dtDiffHours (d1, d2)	d1 と d2 の差 (時間) を double として戻します。	datetime のみです。
dtDiffMinutes (d1, d2)	2 つの日付の差 (分数) を double として戻します。	datetime のみです。
dtDiffMonthss (d1, d2)	2 つの日付の差 (月数) を double として戻します。	datetime のみです。
dtDiffSeconds (d1, d2)	2 つの日付の差 (秒数) を double として戻します。	datetime のみです。
dtDiffYears (d1, d2)	2 つの日付の差 (年数) を double として戻します。	datetime のみです。
dtGetDay (date)	月単位での日付を表す値を戻します。	datetime のみです。
dtGetHour (date)	時間値を戻します。	datetime のみです。
dtGetMinute (date)	分値を戻します。	datetime のみです。
dtGetMonth (date)	月値を戻します。	datetime のみです。
dtGetSecond (date)	秒値を戻します。	datetime のみです。

表 77: 日時関数 (続き)

演算子	説明	データ型の制限
dtGetYear (date)	年値を戻します。	datetime のみです。
dtIsLeapYear (year or date)	year がうるう年の場合は 1、それ以外は 0 を戻します。	date は datetime です。year が 2 桁の場合、 $\leq 29$ は 20xx、 $\geq 30$ は 19xx とみなされます。
MakeDate (month, day, year)	日付のみで日時値を作成します。時間値は 00:00:00 です。	month、day および year は integer です。year が 2 桁の場合、 $\leq 29$ は 20xx、 $\geq 30$ は 19xx とみなされます。
MakeDT (month, day, year, hours, minutes, seconds)	時間および日付で日時値を作成します。	month、day、year、hours、minutes および seconds は integer です。year が 2 桁の場合、 $\leq 29$ は 20xx、 $\geq 30$ は 19xx とみなされます。
MakeTime (hours, minutes, seconds)	時間のみで日時値を作成します。日付値は、Dec. 30, 1899 です。	hours、minutes および seconds は integer です。

## 文字列関数

次の関数で文字列値を操作します。次の関数を使用すると、大 / 小文字の変換、部分文字列の抽出、文字列値の切捨てまたは埋込み、長さの計算および 2 つの文字列値の比較を実行できます。

表 78: 文字列関数

演算子	説明
Concat(str1, str2)	str1 と str2 を連結します。
IsDateTime(str)	文字列が datetime 値の場合は 1、それ以外は 0 (ゼロ) を戻します。
IsDecimal(str)	文字列が decimal 値の場合は 1、それ以外は 0 (ゼロ) を戻します。
IsDouble(str)	文字列が double 値の場合は 1、それ以外は 0 (ゼロ) を戻します。
IsInt(str)	文字列が integer 値の場合は 1、それ以外は 0 (ゼロ) を戻します。
Left(str,n)	str の左から n 個の文字を抽出します。
Mid(str,n,m)	str 内の n から始まる m 個の文字で構成される部分文字列を抽出します。
Right(str,n)	str の右から n 個の文字を抽出します。
ToUpper(str)	str を大文字に変換します。
ToLower(str)	str を小文字に変換します。
TrimLeft(str)	str の左側から空白を切り捨てます。
TrimRight(str)	str の右側から空白を切り捨てます。
PadLeft (str1, str2, n)	str2 内の文字列を str1 の左側に n 回埋め込みます。

表 78: 文字列関数

演算子	説明
PadRight (str1, str2, n)	str2 内の文字列を str1 の右側に n 回埋め込みます。
StrCmp(str1, str2)	str1 と str2 を比較します。ASCII 照合順序で、str1 が str2 より優先される場合は -1 を返します。str2 が str1 より優先される場合は 1 を返します。いずれも優先されない場合は 0 を返します。
StrLength (str)	str の長さ (文字数) を返します。
Substring (str1, str2)	str1 内の str2 の部分を返します。

## 数学関数

数学関数は、数値 (複数可) に対して数学演算を実行し、結果として数値を返します。

表 79: 数学関数

演算子	説明	データ型の制限
Min(x,y)	x と y を比較し、小さいほうの値を返します。	datetime または string ではありません。
Max(x,y)	x と y を比較し、大きいほうの値を返します。	datetime または string ではありません。
Range(x,y,z)	x が y と z の間の場合は 1、それ以外の場合は 0 (ゼロ) を返します。	3 つすべてのパラメータですべての数値型を使用できます。
Round(x)	x を最も近い整数に四捨五入して返します。	double または decimal のみです。
RoundTo(x,y)	小数点以降の小数を scale に指定された桁数で四捨五入します。パラメータ (decimal, scale) を使用します。 scale パラメータに 0 (ゼロ) を指定すると、最も近い整数値に四捨五入されます。	decimal のみです。
Trunc(x)	x を整数値部分を残して切り捨てます。	double または decimal のみです。
TruncateTo (x,y)	小数点以降の 10 進数を scale に指定された桁数で切り捨てます。この関数では、2 つのパラメータ (decimal, scale) を使用します。 scale パラメータに 0 (ゼロ) を指定すると、小数点の右側のすべての数値が切り捨てられます。	decimal のみです。
Ceil(x)	x 以上の最小値を返します。戻されるデータ型は double です。	double または decimal のみです。
Floor(x)	x 以下の最大値を返します。戻されるデータ型は double です。	double または decimal のみです。
Abs(x)	x の絶対値を返します。	long、double、decimal または int のみです。

表 79: 数学関数 (続き)

演算子	説明	データ型の制限
Sqrt(x)	x の平方根を戻します。	double または decimal のみです。x は正数値である必要があります。
Mod(x,y)	x を y で割った場合の剰余を戻します。	
Sin(x)	x のサインを戻します。	double のみです。x はラジアンで表現する必要があります。
Cos(x)	x のコサインを戻します。	double のみです。x はラジアンで表現する必要があります。
Tan(x)	x のタンジェントを戻します。	double のみです。x はラジアンで表現する必要があります。
Asin(x)	x のアークサインを戻します。	double のみです。x はラジアンで表現する必要があります。
Acos(x)	x のアークコサインを戻します。	double のみです。x はラジアンで表現する必要があります。
Atan(x)	x のアークタンジェントを戻します。	double のみです。x はラジアンで表現する必要があります。
Random(x)	0 から x-1 までの整数の乱数を戻します。	double のみです。

## 論理演算子

論理演算子は、1 つ以上の値に条件を課し、指定したロジックに基づいて単一の結果を戻します。オペランドには、定数、カスタム・プロパティまたは式を使用できます。これらの演算子は、関数リスト、キーパッドまたはキーボードから指定できます。

表 80: 論理演算子

演算子	説明
IF(x)THEN(y)ELSE(z)	条件 x を検索します。検出された場合は y を適用します。検出されなかった場合は z を適用します。
SWITCH(x)CASE(y) : (z) DEFAULT(w)	x = y の場合は z を実行します。それ以外の場合は w を実行します。
(x)AND(y)	条件 x および y を検索します。x および y が検出され、両方とも 0 (ゼロ) 以外の値の場合は 1 を戻します。それ以外の場合は 0 (ゼロ) を戻します。
(x)OR(y)	条件 x および y を検索します。x または y のいずれかが検出され、その値が 0 (ゼロ) 以外の値の場合は 1 を戻します。それ以外の場合は 0 (ゼロ) を戻します。
(x)XOR(y)	条件 x および y を検索します。x または y のいずれかのみが検出され、その値が 0 (ゼロ) 以外の値の場合は 1 を戻します。それ以外の場合は 0 (ゼロ) を戻します。
NOT(x)	x が 0 (ゼロ) の場合は 1 を戻します。それ以外の場合は 1 を戻します。

## バイナリ演算子

バイナリ演算子には、2つのオペランドが演算子の両側に1つずつ必要です。オペランドには、評価時に結果が数値として示される定数、カスタム・プロパティまたは式を使用できます。指数演算子または剰余演算子は、キーボードで入力して指定します。

表 81: バイナリ演算子

演算子	説明
+	加算: 2つの値を加算します。
-	減算: 1つの値から別の値を減算します。
*	乗算: 2つの値を乗算します。
/	除算: 2つの値を除算します。
$x^n$ または $x^{**}n$	指数計算: $x$ を $n$ 乗します。
$n\%x$	モジュロ: $n$ を $x$ で割った場合の剰余を戻します。

## その他の演算子

変換演算子は、値のデータ型を別のデータ型に変換します。結果の値の操作方法または表現方法に対する制限は、新しいデータ型の属性によって異なります。

表 82: その他の演算子

演算子	説明
ToDateTime(x)	$x$ を datetime 型に変換します。
ToDecimal(x,p,s)	$x$ を精度 $p$ 、スケール $s$ の decimal 型に変換します。
ToDouble(x)	$x$ を double 型に変換します。
ToInt(x)	$x$ を integer 型に変換します。
ToLong(x)	$x$ を long 型に変換します。
ToString(x)	$x$ を string 型に変換します。
ToUInt(x)	$x$ を unsigned integer 型に変換します。
NULL_INT	NULL の integer 値を戻します。
NULL_UINT	NULL の unsigned integer 値を戻します。
NULL_LONG	NULL の long integer 値を戻します。
NULL_DOUBLE	NULL の double 値を戻します。
NULL_STRING	NULL の string 値を戻します。
NULL_DATETIME	NULL の datetime 値を戻します。
NULL_DECIMAL	NULL の decimal 値を戻します。

## カルキュレータの演算子

### 関係演算子および等価演算子

関係演算子および等価演算子によって、2つの値の論理的な比較が可能になります。これらの演算子は、式の両側の値に対して動作します。関数リストに存在する NULL 値テストを除き、これらの演算子を式に追加するには、キーボードで入力します。

表 83: 関係演算子および等価演算子

演算子	説明
<	左側の値が右側の値より小さいことをテストします。
<=	左側の値が右側の値以下であることをテストします。
>	左側の値が右側の値より大きいことをテストします。
>=	左側の値が右側の値以上であることをテストします。
==	左側の値が右側の値と等しいことをテストします。
!= または <>	左側の値が右側の値と等しくないことをテストします。
IsNull(x)	NULL 値であることをテストします。x が NULL の場合は 1 を返します。それ以外の場合は 0 (ゼロ) を返します。
IsNotNull (x)	NULL 値以外の値であることをテストします。x が NULL でない場合は 1 を返します。それ以外の場合は 0 (ゼロ) を返します。

### 単項演算子

単項演算子は、カッコ内の値の記号を変更します。この演算子を式に追加するには、キーボードで入力して指定します。

表 84: 単項演算子

演算子	説明	データ型の制限
-(x)	x の負の値を返します。	int、long、double および uint のみです。



---

# データ型

この付録では、10 進演算について説明し、Flat File Source Transform でサポートされているバイナリ入力データ型を示します。

ここで説明する内容は次のとおりです。

- [10 進演算計算](#)
- [サポートされているバイナリ入力データ型](#)

## 10 進演算計算

パフォーマンス向上のために、decimal 値は、計算または比較に使用される場合、double 型に変換されます。ただし、このプロセスによって、精度が損なわれます。decimal 値の結果の特定の精度を確保するには、10 進演算機能を有効にする必要があります。decimal フィールドでソートまたは比較を実行する Transform は、10 進演算を有効にすると影響を受ける可能性があります。これは、decimal 値が double 型に変換されなくなるためです。

decimal データ値は、Precision および Scale の 2 つのパラメータで定義されます。Precision は有効桁数、Scale は小数点の位置を示します。たとえば、値 123.4567 の有効桁数は 7 であるため、Precision は 7 になります。小数点の後は 4 桁であるため、Scale は 4 になります。

次の表に、decimal フィールドのルールを示します。

表 85: decimal フィールドのルール

Precision の最大値は 38 です。
Scale の最大値は 38 です。
Scale は Precision 以下です。このルールは強制適用されます。
計算された decimal 値の場合： Precision のデフォルト値は、計算で使用されるすべての decimal フィールドの Precision の最大値です。 Scale のデフォルト値は、次の関数で計算されます。 Min[38 - Precision, Max[Scale of each decimal operand]]
Scale は、RoundTo、TruncateTo および ToDecimal 以外のすべての Expression Calculator Transform 関数で無視されます。
Precision で許容される桁数を超える結果は、Precision に適合するように四捨五入されます。たとえば、計算された値 123.456 の Precision が 5 の場合、この計算結果は 123.46 に四捨五入されます。
計算された結果の整数部分の長さに Precision で許容される桁数より多くの桁数が必要な場合は、ランタイム・エラーが生成されます。たとえば、計算結果 123.456 の Precision が 2 の場合、四捨五入して意味のある結果を得ることはできません。
計算結果の小数点の右側に Scale の桁数より多くの桁数が存在し、小数点の左側に Precision の値から Scale の値を引いた桁数より少ない桁数が存在する場合、Oracle BAM Design Studio では、Precision で許容可能な最大桁数まで小数点の右側の桁数が戻されます。たとえば、計算された値 123.456 の Precision が 6 で、Scale が 2 であるとし、Precision から Scale を引いた値は 4 で、小数点の左側は 3 桁のみであるため、小数点の右側の 3 桁すべての精度が保存されます。ただし、計算結果の Precision が 5 で、Scale が 2 の場合、この結果は 123.46 に四捨五入されます。

## Precision および Scale の自動設定と手動設定

値を decimal 型として定義する場合は、Oracle BAM Design Studio で Precision および Scale を判別するか、またはこれらの値を手動で設定できます。

### Precision の手動設定例

Precision および Scale の手動設定は、計算結果が Precision のデフォルト値に適合しない際にフィールドの Precision の値を増加する場合などに使用できます。

たとえば、decimal フィールド Dec1 の Precision が 2、Scale が 1 で、値が 1.2 であるとして、この場合、次のように計算フィールドを定義します。

```
Dec2 = Dec1 * 9
```

新しいフィールド Dec2 の Precision も 2 で、Scale も 1 です。ただし、Dec2 の値 (10.8) は、Precision が 2、Scale が 1 の decimal フィールドでは表すことができません。したがって、Precision を 3 以上にして、Precision および Scale のデフォルト設定を上書きする必要があります。Scale は 1 のままにしておくことができます。

### Precision および Scale の手動設定の問題

decimal フィールドを計算する場合、オペランドの Precision とは異なる結果の Precision を指定すると、計算が行われる前に結果の Precision がオペランドに適用されます。計算結果の Precision が計算フィールドに適用され、元のオペランドの Precision は計算で使用されないことに注意してください。

次に、この問題の例を示します。

Price という decimal フィールドと Cost という別の decimal フィールドを定義し、各フィールドの Precision を 10、Scale を 3 にするとします。各フィールドには、次の値が割り当てられています。

```
Price: = 123.40
```

```
Cost: = 123.35
```

Expression Calculator Transform を使用して、次のように Profit を計算するとします。

```
Profit = Price - Cost
```

Profit が常に分数 (常に 1 未満) になることがわかっている場合に (答が約 0.05 になるとわかっているため) Profit の Precision を 4 に設定すると、Profit の計算値は正しくなくなります。

これは、新しい Precision の値 4 が次に示すように Price および Cost に適用され、Profit の値が新しい Precision の値 4 に適合するように切り捨てられるため発生します。

```
Price = 123.4
```

```
Cost: = 123.4
```

Profit の計算結果は 0 (ゼロ) となるため、正しくなくなります。

ただし、Profit の値を自動的に判別すると、計算結果は、正しい答の 0.05 になります。

**注意：** この例は、decimal オペランドの Precision とは異なる値で結果の Precision を手動で設定する場合にのみ適用されます。Precision の値を自動的に判別する場合、この例は適用されません。また、Scale の設定はこの例には関係ありません。

## リテラルの処理

Expression Calculator では、リテラル数値は double として処理されます。たとえば、次のような decimal フィールド X が式に含まれているとします。

```
X = 1.1
```

システムによって、リテラルが double 型 (1.0999999999999999 などの値) に変換されます。フィールド X に不適切な値が割り当てられたことは明らかであると考えられます。

リテラル値が decimal 型として処理されるようにする適切な方法としては、**ToDecimal** 関数を次のように起動する方法があります。

```
X = ToDecimal("1.1", <precision>, <scale>)
```

## 10 進演算の有効化と無効化の切替え

Windows のレジストリ設定 UseDecimalMath は、DC セクションに定義されている DWORD 設定です。すべての decimal 型の演算計算を 10 進精度で実行するには、値を 1 に設定します。decimal フィールドを double フィールドとして管理するには、値を 0 (ゼロ) に設定します。

**注意：** デフォルトでは、10 進演算は無効になっています。

また、「Plan Properties」ダイアログ・ボックスには、「**Use Decimal Math Provider**」チェック・ボックスがあります。このオプションによって、対象となっているプランの 10 進演算を有効および無効にします。

## decimal フィールドを使用する Expression Calculator 関数

次の表に、decimal フィールドを使用する Expression Calculator 関数の動作の詳細を示します。

表 86: decimal フィールドを使用する Expression Calculator 関数

演算子	説明
+ - * /	いずれかのオペランドが decimal 型の場合、結果は decimal 型になります。
If-Then-Else	Then および Else の両方の句が decimal 型の場合、結果は decimal 型になります。1 つの句のみが decimal 型の場合は、エラーが発生します。
Switch-Case-Default	すべての Switch-Case 条件式は同じデータ型 (decimal 型も可能) である必要があります。すべての Case-Default 実行式は同じデータ型 (decimal 型も可能) である必要があります。また、この実行式によって関数の出力が指定されます。たとえば、すべての実行式が decimal 型の場合、出力は decimal 型になります。 すべての条件式のデータ型およびすべての実行式のデータ型は同じである必要がありますが、各セクションのデータ型は異なる場合があります。
Min Max	すべてのオペランドが decimal 型の場合、結果は decimal 型になります。一部のオペランドが同じ型でない場合は、エラーが発生します。
Range	Range 関数は、整数 (0 または 1) を戻します。各引数は、数値データ型にすることができます。
Unary Abs Ceil Floor Sqrt	各関数で、厳密に 1 つのオペランドが使用されます。そのオペランドが decimal 型の場合、結果は decimal 型になります。
ToDecimal	別のデータ型 (datetime 以外) から decimal 型を作成します。また、この関数は、decimal 形式を別の Precision および Scale で別の decimal 形式に変換します。
And Or Xor Not == > >= < <= != Round Trunc ToDouble ToInt ToLong ToUInt ToString	これらの関数のそれぞれで、decimal 引数を使用されます。

表 86: decimal フィールドを使用する Expression Calculator 関数 (続き)

演算子	説明
RoundTo	この関数では、2つのパラメータ (decimal, scale) を使用します。小数点以降の 10 進数を scale に指定された桁数で四捨五入します。 0 (ゼロ) は scale パラメータに指定可能な値です。0 (ゼロ) を指定すると、最も近い整数値に四捨五入されます。
TruncateTo	この関数では、2つのパラメータ (decimal, scale) を使用します。小数点以降の 10 進数を scale に指定された桁数で切り捨てます。 0 (ゼロ) は scale パラメータに指定可能な値です。0 (ゼロ) を指定すると、小数点の右側のすべての数値が切り捨てられます。
NULL_DECIMAL	NULL の decimal 値を戻します。

## サポートされているバイナリ入力データ型

Flat File Source Transform では、次のデータ型がサポートされています。

表 87: Flat File Source Transform でサポートされているデータ型

データ型	説明
Packed Decimal	可変長シーケンスの 2 進化 10 進数。各バイトの上位 4 ビット内の数値の各ペアのうちより有効な数値で 2 桁が 1 バイトにパックされます。最後のバイトの下位 4 ビットに、数値の記号 (正の値に対しては 16 進数 C または F、負の値に対しては 16 進数 D) を示すコードが格納されます。
IBM 370 Small Integer	有効性が低い順にバイトが格納された 2 バイトの 2 進数。範囲は -32,768 ~ +32,767 です。
IBM 370 Integer	有効性が低い順にバイトが格納された 4 バイトの 2 進数。範囲は次のとおりです。 -2,147,483,648 ~ +2,147,483,647
IBM 370 Long Integer	有効性が低い順にバイトが格納された 8 バイトの 2 進数。範囲は次のとおりです。 -9,223,372,036,854,775,808 ~ +9,223,372,036,854,775,807
IBM 370 Small Unsigned Integer	有効性が低い順にバイトが格納された 2 バイトの 2 進数。範囲は 0 ~ +65,535 です。
IBM 370 Unsigned Integer	有効性が低い順にバイトが格納された 4 バイトの 2 進数。範囲は 0 ~ _4,294,967,295 です。
IBM 370 Long Unsigned Integer	有効性が低い順にバイトが格納された 8 バイトの 2 進数。範囲は 0 ~ +18,446,744,073,609,551,615 です。
IBM 370 Float	IBM 370 形式での 4 バイトの 2 進浮動小数値。
IBM 370 Double	IBM 370 形式での 8 バイトの 2 進浮動小数値。
IBM 370 Extended	IBM 370 形式での 16 バイトの 2 進浮動小数値。
8086 Small Integer	有効性が高い順にバイトが格納された 2 バイトの 2 進数。範囲は -32,768 ~ +32,767 です。
8086 Integer	有効性が高い順にバイトが格納された 4 バイトの 2 進数。範囲は -2,147,483,648 ~ +2,147,483,647 です。

表 87: Flat File Source Transform でサポートされているデータ型 (続き)

データ型	説明
8086 Long Integer	有効性が高い順にバイトが格納された 8 バイトの 2 進数。範囲は -2,147,483,648 ~ +2,147,483,647 です。
8086 Small Unsigned Integer	有効性が高い順にバイトが格納された 2 バイトの 2 進数。範囲は 0 ~ +65,535 です。
8086 Unsigned Integer	有効性が高い順にバイトが格納された 4 バイトの 2 進数。範囲は 0 ~ +4,294,967,295 です。
8086 Long Unsigned Integer	有効性が高い順にバイトが格納された 8 バイトの 2 進数。範囲は 0 ~ +18,446,744,609,551,615 です。
8086 Float	IEEE 形式での 4 バイトの 2 進浮動小数値。
8086 Double	IEEE 形式での 8 バイトの 2 進浮動小数値。

サポートされているバイナリ入力データ型



---

# 索引

## 数字

2つのセットの列の結合 80

## A

Add Columns Transform 91

## C

CASE 文 105

Column Select Transform 93

Command Link Sink Transform 56

Comparison Transform 74

Concatenation Transform 95

Conditional Splitter Transform 77

## D

Delimited Text File Sink Transform 57

Delimited Text File Source Transform 16

Disk Sort 97

使用方法 98

設定 98

ヒントおよび方法 99

## E

Expression Builder 102

Expression Calculator 100

構文 105

使用方法 100

設定 100

ヒントおよび方法 104

## F

Filter Transform 106

Flat File Source 21

「Attributes」タブ 23

「Columns」タブ 27

「Errors」タブ 29

「Processing」タブ 24

「Tags」タブ 26

サポートされている出力データ型 32

サポートされている入力データ型 31

使用方法 21

設定 21

ヒントおよび方法 31

## G

Grid 54

## I

「Iteration」ダイアログ・ボックスの設定 85

## J

Join Transform

使用方法 80

設定 80

ヒントおよび方法 82

## K

Key Generation Transform 108

Key Lookup Transform 109

## M

Memory Sort Transform 112

## O

OLE DB プロバイダへの接続 32

OLEDB Source Transform 32

Oracle BAM Alert Transform 59

Oracle BAM Delete Transform 60

Oracle BAM Insert Transform 63

Oracle BAM Lookup Transform 117

Oracle BAM Message Sender Transform 64

Oracle BAM Message Tracker Transform 65

Oracle BAM Update Transform 66

## P

Pivot Transform 119

## R

Rank Transform 122

Record Number Transform 125

Remove Duplicate Rows Transform 126

Rename Columns Transform 128

Round Robin Splitter Transform 83

## S

Search & Replace Transform 130

Splitter Transform 84

SQL Command Sink Transform 70

SQL Command Transform 132

SQL Query 37

使用方法 38

設定 38

ヒントおよび方法 40

編集済 9, 39

Substring Transform 134

## T

Terminal Sink 72

Time Generation Transform 41

Time Lookup Transform 136

Transform

コピーおよびインプレース 6

定義 2

マルチパスおよびシングルパス 6

Transform の再利用 12

Transform の特性

完全な表 10

定義済 9

Transpose Transform 137

## U

Union Transform 88

## V

VBScriptCopy 140

VBScriptInplace 140

VBScriptSink 72

VBScriptSource 43

## あ

値のランク付け 122

## い

インプレース処理 6

## お

大 / 小文字の無視 86

## き

キー

検索 109

生成 108

キーの生成 108

キー・フィールド 86

行, 重複の削除 126

## け

結果の表示

Grid 54

検索

キー 109

時間 136

## こ

構文, Expression Calculator 105

コピー処理およびインプレース処理 6

## さ

削除

重複行 126

列 93

サブプラン

記述 84

終了 84

使用方法に関するヒント 88

特性 88

## し

時間ディメンション表への移入 41  
式

作成 102

順序の変更 105

名前の変更 104

編集 104

式の作成 102

式の編集 104

使用方法

CASE 文 105

Expression Calculator の構文 105

比較基準 74

シングルパス処理 6

## す

スキーマ

情報 9

不明 9

## せ

制限

Grid 54

接続先プロパティ 87

## そ

ソース Transform

Delimited Text File Source 16

Flat File Source 21

OLEDB Source 32

SQL Query 37

Time Generation 41

VBScriptSource 43

ソース・フィールド 87

ソート

Disk Sort 97

Join の使用 76, 82, 121, 124

Memory Sort 112

様々なタイプ 7

必要な Transform 7

## つ

追加

Transform へのコメント 12

レコード番号列 125

列 91

## て

データ

比較 74

フィルタ 106

データの結合 88

データの検証 109

データの比較 74

データのフィルタ 106

データ・フロー

アーキテクチャ 2

結合 80

データの結合 88

分割 84

列の削除 93

列の追加 91

データ・フローの分割

サブプラン 85

条件付き 77

単純 84

比較の使用 74

ラウンドロビンの使用 83

## な

名前の変更

式 104

列 128

## は

パフォーマンス

向上 8

バッチ・ローダーのテスト 8

パフォーマンスの向上 8

反復 84

## ひ

表示シンク

Grid 54

ヒントおよび方法

Disk Sort 99

Expression Calculator 104

Flat File Source 31

Join 82

SQL Query 40

## ふ

フィールド, プロパティ・マッピング 87

部分文字列の抽出 134

不明スキーマ 9

プラン

実行モデル 2

データ・フロー 2

パフォーマンスの向上 8

## ほ

保存

再利用する Transform 12

## ま

マルチパス処理 6

## ゆ

ユーザー入力の SQL 9

## れ

列

入替え 137

名前の変更 128

ピボット 119

連結 95

列の入替え 137

列のピボット 119

列の連結 95

## わ

割当て 87

