

Oracle® Data Integrator

Knowledge Modules リファレンス・ガイド

10g リリース 3 (10.1.3)

部品番号 : B53289-01

2009 年 2 月

Oracle Data Integrator Knowledge Modules リファレンス・ガイド, 10g リリース 3 (10.1.3)

部品番号 : B53289-01

原本名 : Oracle Data Integrator Knowledge Modules Reference Guide, 10g Release 3 (10.1.3)

Copyright © 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、大量輸送、医療あるいはその他の本質的に危険を伴うアプリケーションで使用されることを意図しておりません。このプログラムをかかるとして使用する際、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

1 概要	
2 ファイル	
ナレッジ・モジュール	2-1
3 汎用 SQL	
ナレッジ・モジュール	3-1
4 Hyperion Essbase	
ナレッジ・モジュール	4-1
プラットフォームのサポート	4-1
固有の要件	4-1
5 Hyperion Financial Management	
ナレッジ・モジュール	5-1
プラットフォームのサポート	5-1
固有の要件	5-1
6 Hyperion Planning	
ナレッジ・モジュール	6-1
プラットフォームのサポート	6-1
固有の要件	6-1
7 Hypersonic SQL	
ナレッジ・モジュール	7-1
8 IBM DB2 UDB	
ナレッジ・モジュール	8-1
固有の要件	8-3
9 IBM DB2/400	
ナレッジ・モジュール	9-1
固有の要件	9-3

10 Informix	
ナレッジ・モジュール	10-1
11 JD Edwards EnterpriseOne	
概要	11-1
JDE EnterpriseOne ナレッジ・モジュール	11-1
プラットフォームのサポート	11-1
インストールおよび構成	11-2
Oracle Data Integrator JDE EnterpriseOne KM の使用	11-2
トポロジの定義	11-2
プロジェクトの設定	11-2
JDE 表のリバースエンジニアリング	11-3
統合インタフェースでの JDE のソースとしての使用	11-4
統合インタフェースでの JDE Z 表のターゲットとしての使用	11-4
IKM JDE Enterprise One Control Append (UBE) を使用した JDE へのデータの統合	11-4
IKM JDE Enterprise One Control Append (UBE) の制限	11-5
ナレッジ・モジュール・オプションのリファレンス	11-6
RKM JDE Enterprise One Oracle	11-6
IKM JDE Enterprise One Control Append (UBE)	11-7
12 JMS	
ナレッジ・モジュール	12-1
13 Microsoft Access	
ナレッジ・モジュール	13-3
14 Microsoft SQL Server	
ナレッジ・モジュール	14-1
固有の要件	14-3
BULK INSERT コマンドの使用方法	14-3
BCP コマンドの使用方法	14-3
リンク・サーバーの使用方法	14-4
15 Netezza	
ナレッジ・モジュール	15-1
16 Oracle AS レガシー・アダプタ /Attunity Stream	
概要	16-1
概要	16-1
Attunity Stream ナレッジ・モジュール	16-2
プラットフォームのサポート	16-2
インストールおよび構成	16-2
Attunity Stream KM の使用	16-2
トポロジの定義	16-3
Attunity Stream モデルの作成とリバースエンジニアリング	16-3

プロジェクトの設定	16-3
LKM Attunity to SQL を使用したインタフェースの設計	16-4
ナレッジ・モジュール・オプションのリファレンス	16-4
LKM Attunity to SQL	16-4
17 Oracle Database	
ナレッジ・モジュール	17-1
固有の要件	17-6
SQL*LOADER ユーティリティの使用法	17-6
外部表の使用法	17-6
Oracle Log Miner の使用法	17-6
18 Oracle Data Quality	
固有の要件	18-1
19 Oracle E-Business Suite	
概要	19-1
E-Business Suite ナレッジ・モジュール	19-1
プラットフォームのサポート	19-2
インストールおよび構成	19-2
EBS KM の使用	19-3
トポロジの定義	19-3
プロジェクトの設定	19-4
E-Business Suite 表のリバースエンジニアリング	19-4
E-Business Suite 表のリバースエンジニアリングの機能	19-5
統合インタフェースでの E-Business Suite のソースとしての使用	19-6
統合インタフェースでの E-Business Suite のターゲットとしての使用	19-6
OpenInterface を使用した E-Business Suite へのデータの統合	19-6
グループ ID の管理	19-6
OpenInterface プログラムの実行	19-7
ナレッジ・モジュール・オプションのリファレンス	19-8
RKM E-Business Suite	19-8
IKM E-Business Suite (Open Interface)	19-8
20 Oracle Enterprise Service Bus	
概要	20-1
相互参照	20-1
ESB 相互参照ナレッジ・モジュール	20-2
XREF KM プロセスの概要	20-2
ロード・フェーズ (LKM)	20-2
統合および相互参照フェーズ (IKM)	20-2
処理済レコードの更新 / 削除 (LKM)	20-3
インストールおよび構成	20-3
Oracle Data Integrator ESB 相互参照 KM を使用した XREF の利用	20-3
トポロジの定義	20-3
プロジェクトの設定	20-3

Oracle Data Integrator ESB 相互参照 KM とのインタフェースの設計	20-3
ナレッジ・モジュール・オプションのリファレンス	20-5
LKM SQL to SQL (ESB XREF)	20-5
LKM MSSQL to SQL (ESB XREF)	20-6
IKM SQL Control Append (ESB XREF)	20-6

21 Oracle OLAP

概要	21-1
Oracle OLAP ナレッジ・モジュール	21-1
プラットフォームのサポート	21-2
インストールおよび構成	21-3
Oracle OLAP KM の使用	21-3
プロジェクトの設定	21-3
OLAP キューブで使用されている Oracle 表のリバースエンジニアリング	21-3
統合インタフェースでの Oracle OLAP のソースとしての使用	21-4
統合インタフェースでの Oracle ROLAP のターゲットとしての使用	21-4
統合インタフェースでの Oracle MOLAP のターゲットとしての使用	21-4
ナレッジ・モジュール・オプションのリファレンス	21-5
RKM Oracle OLAP (Jython)	21-5
IKM Oracle AW Incremental Update	21-5

22 Oracle PeopleSoft

概要	22-1
PeopleSoft ナレッジ・モジュール	22-1
プラットフォームのサポート	22-1
インストールおよび構成	22-2
PeopleSoft KM の使用	22-2
トポロジの定義	22-2
プロジェクトの設定	22-2
PeopleSoft 表のリバースエンジニアリング	22-3
統合インタフェースでの PeopleSoft のソースとしての使用	22-4
ナレッジ・モジュール・オプションのリファレンス	22-4
RKM PeopleSoft ORACLE および RKM PeopleSoft MSSQL	22-4

23 Oracle Siebel CRM

概要	23-1
Siebel ナレッジ・モジュール	23-1
プラットフォームのサポート	23-2
Siebel からのデータ抽出の概要	23-2
インストールおよび構成	23-3
Siebel KM の使用	23-3
トポロジの定義	23-3
プロジェクトの設定	23-3
Siebel のリバースエンジニアリング	23-4
統合インタフェースでの Siebel のソースとしての使用	23-5
EIM 表を介した統合インタフェースでの Siebel のソースとしての使用	23-5
EIM 表を介した統合インタフェースでの Siebel のターゲットとしての使用	23-5

ナレッジ・モジュール・オプションのリファレンス	23-6
IKM SQL to Siebel Append (EIM)	23-6
LKM Siebel to SQL (EIM)	23-6
RKM Siebel Oracle / RKM Siebel MSSQL	23-7
RKM Siebel EIM Oracle / RKM Siebel EIM MSSQL	23-7
24 SalesForce	
ナレッジ・モジュール	24-1
固有の要件	24-1
25 SAS	
ナレッジ・モジュール	25-1
固有の要件	25-2
SAS のユーザー名およびパスワードの取得	25-2
SAS のための Oracle Data Integrator の構成	25-3
SAS のリモート表とローカル表	25-3
26 Sybase ASE	
ナレッジ・モジュール	26-1
固有の要件	26-3
27 Sybase IQ	
ナレッジ・モジュール	27-1
固有の要件	27-3
28 Teradata	
ナレッジ・モジュール	28-1
固有の要件	28-5
Teradata 用の KM の最適化	28-5
1 次索引および統計	28-5
Teradata ユーティリティのサポート	28-6
名前付きパイプのサポート	28-6
最適化された一時表管理	28-7

このドキュメントでは、Oracle Data Integrator 10g リリース 3 (10.1.3) に含まれるナレッジ・モジュールについて説明します。

ナレッジ・モジュールをプロジェクトにインポートした後、使用方法の詳細は必ずそれぞれの説明を参照してください。各ナレッジ・モジュールには、制限、およびオペレーティング・システムまたはデータベース固有のコマンドが含まれます。詳細は、該当するテクノロジーのドキュメント・セットを参照してください。

Oracle Data Integrator には、テクノロジーに依存しないナレッジ・モジュールが含まれています。汎用 SQL および JMS の各項に記載されているナレッジ・モジュールは、ほとんどのデータベースおよび JMS 準拠のほとんどのミドルウェアで動作するように設計されています。ただし、特定のテクノロジー用に事前作成されたナレッジ・モジュールが存在する場合は、そのテクノロジーのナレッジ・モジュールを常に使用することをお勧めします。

Oracle Data Integrator では、JDBC 接続を使用して異なるデータベースにアクセスします。可能なかぎり Type 4 JDBC ドライバを使用することをお勧めします。JDBC ドライバは、Oracle Data Integrator の UI またはエージェントを実行する各マシンの Java クラスパスで定義する必要があります。使用するドライバの jar ファイルを Oracle Data Integrator のインストール・フォルダの /drivers サブディレクトリにコピーすると簡単です。詳細は、JDBC ドライバに同梱されているドキュメントを参照してください。

前述のコメントは、JMS 固有のクライアント、また、一般的にはナレッジ・モジュールで要求されるすべての API にも適用されます。要求されるすべての Java アーカイブ・ファイル (jar ファイル) は、Java クラスパスで定義するか、もしくは /drivers ディレクトリにコピーする必要があります。

ナレッジ・モジュール

一部のナレッジ・モジュールは、ターゲット・ファイルへデータをエクスポートしたり、ソース・ファイルからデータを読み取るために提供されています。これらのナレッジ・モジュールについては他の項で説明しています。ここでは、参照用の一覧のみを記載します。

ファイルからの読取り：

- LKM File to SQL
- LKM File to DB2 UDB (LOAD)
- LKM File to MSSQL (BULK)
- LKM File to Netezza (EXTERNAL TABLE)
- LKM File to Oracle (EXTERNAL TABLE)
- LKM File to Oracle (SQLLDR)
- LKM File to Salesforce (Upsert)
- LKM File to SAS
- LKM File to Sybase IQ (LOAD TABLE)
- IKM File to Teradata (TTU)
- LKM File to Teradata (TTU)

ファイルへの書込み：

- IKM SQL to File Append
- IKM Netezza To File (EXTERNAL TABLE)
- IKM Salesforce to File (with filter)
- IKM Salesforce to File (without filter)
- IKM Teradata to File (TTU)

ファイル用のその他のナレッジ・モジュール：

種類	ナレッジ・モジュール	説明
リバースエンジニアリング	RKM File (FROM EXCEL)	Microsoft Excel スプレッドシートからファイルのメタデータを取得します。ファイル構造の定義を別個の Excel スプレッドシートで保持する場合は、この KM の使用を検討してください。
リバースエンジニアリング	RKM Oracle Data Quality	Oracle Data Quality DDX ファイルからファイルのメタデータを取得します。 DDX ファイルは、Oracle Data Quality により生成されます。

ナレッジ・モジュール

この項に記載されているナレッジ・モジュールは、一般的な SQL 準拠データベースのほとんどに適用されます。適用されるデータベースには、Oracle、Microsoft SQL Server、Sybase ASE、IBM DB2、Teradata、PostgreSQL、MySQL、Derby などが含まれます。また、これらの一部のデータベースに対しては、特定の SQL および最大限のパフォーマンスを引き出す強力なローダー・ユーティリティを利用するための、追加のナレッジ・モジュールも提供されています。

種類	ナレッジ・モジュール	説明
チェック	CKM SQL	<p>データストアで定義された制約と照合して、データの整合性をチェックします。無効なレコードを拒否して、動的に作成されるエラー表に格納します。フロー制御および静的制御に使用できます。</p> <p>SQL 準拠のデータベースでデータの整合性をチェックする場合は、この KM の使用を検討してください。使用するデータベース固有の CKM が存在する場合は、かわりにその CKM を使用してください。</p>
統合	IKM SQL Control Append	<p>置換 / 追加モードで任意の SQL 準拠ターゲット表にデータを統合します。CKM を使用してフロー・データをチェックする必要がある場合は、CKM を起動する前に一時的なステージング表が作成されます。</p> <p>置換モードで（データ整合性チェックを行って、もしくは行わずに）SQL 準拠のターゲット表をロードする場合は、この IKM の使用を検討してください。</p> <p>この IKM を使用するには、ステージング領域がターゲットと同じデータ・サーバー上にあることが必要です。</p>
統合	IKM SQL Incremental Update	<p>増分更新モードで任意の SQL 準拠ターゲット表にデータを統合します。この KM は、データ・フローをステージングするための一時的なステージング表を作成します。その後、ステージング表の内容をターゲット表と比較して、挿入するレコードおよびそれ以外の更新するレコードを推測します。また、CKM を起動してデータ整合性チェックを実行できます。すべてのデータベースで同じバルク更新構文がサポートされていないため、更新は行ごとに行われます。そのため、この KM を大量のデータに使用することはお薦めしません。</p> <p>SQL 準拠のターゲット表をロードして、欠落しているレコードを挿入し、既存のレコードを更新する場合は、この KM の使用を検討してください。固有の増分更新 IKM を使用できる場合は、その IKM を使用してください。より最適化されているためパフォーマンスに優れています。</p> <p>この IKM を使用するには、ステージング領域がターゲットと同じデータ・サーバー上にあることが必要です。</p>

種類	ナレッジ・モジュール	説明
統合	IKM SQL Incremental Update (row by row)	<p>増分更新モードで行ごとのロギングを使用して任意の ANSI-SQL92 準拠のデータベース・ターゲット表にデータを統合します。この KM は IKM SQL Incremental Update と似ていますが、その他に処理された各行の状態が示されます。処理アクティビティを記録するためにログ・ファイルを作成できます。</p> <p>次のオプションがロギング・メカニズムに使用されません。</p> <ul style="list-style-type: none"> ■ LOG_LEVEL: このオプションは記録されるデータの粒度の設定に使用されます。 次のログ・レベルを設定できます。 <ul style="list-style-type: none"> ■ 0: 何も記録されません。 ■ 1: 選択アクション、削除アクション、挿入アクションなどの JDBC アクションが示されます。 ■ 2: レベル 1 の他に、エラーが発生したすべてのレコードが記録されます。 ■ 3: レベル 2 の他に、処理されたすべてのレコードが記録されます。 ■ LOG_FILE_NAME: 使用されるログ・ファイルのフル・パス。 ■ MAX_ERRORS: エラーの最大数を指定します。このオプションで指定したエラーの最大数に達すると、IKM プロセスが停止します。 <p>重要な注意: 大量の場合、このナレッジ・モジュールはお勧めしません。バルク・ユーティリティ (SQL*LOADER、BULK INSERT など) または直接リンク (DBLINKS、リンク・サーバーなど) を使用する他のモジュールの方が、通常効率的です。</p>
統合	IKM SQL to File Append	<p>置換モードで任意の SQL 準拠ステージング領域からターゲット・ファイルにデータを統合します。</p> <p>データを変換してターゲット・ファイルにエクスポートする場合は、この IKM の使用を検討してください。ソース・データストアが同じデータ・サーバー上にある場合は、余分なロード・フェーズ (LKM) を発生させないために、このデータ・サーバーをステージング領域として使用することをお勧めします。</p> <p>この IKM を使用するには、ステージング領域がターゲットと異なる必要があります。</p>
統合	IKM SQL to SQL Append	<p>置換モードで任意の SQL 準拠ステージング領域から SQL 準拠のターゲット表にデータを統合します。</p> <p>ターゲットと異なるステージング領域を使用する場合は、この IKM の使用を検討してください。ほとんどのソース・データストアが同じデータ・サーバー上にある場合は、余分なロード・フェーズ (LKM) を発生させないために、このデータ・サーバーをステージング領域として使用することをお勧めします。</p> <p>この IKM を使用するには、ステージング領域がターゲットと異なる必要があります。</p>

種類	ナレッジ・モジュール	説明
ロード	LKM File to SQL	<p>ASCII ファイルまたは EBCDIC ファイルから、ステージング領域として使用される任意の SQL 準拠データベースへ、データをロードします。エージェントを使用してソース・ファイルから選択されたデータを読み取り、動的に作成された一時ステージング表に結果を書き込みます。</p> <p>ソース・データストアの 1 つが ASCII ファイルまたは EBCDIC ファイルの場合は、この LKM の使用を検討してください。ターゲット・ステージング領域固有の LKM を使用できる場合は、その LKM を使用してください。より最適化されているためパフォーマンスに優れています。たとえば、Oracle データベースにロードする場合は、LKM File to Oracle (SQLLDR) または LKM File to Oracle (EXTERNAL TABLE) を使用します。</p>
ロード	LKM SQL to SQL	<p>SQL 準拠のデータベースから SQL 準拠のステージング領域へデータをロードします。エージェントを使用してソース・データベースから選択されたデータを読み取り、動的に作成された一時ステージング表に結果を書き込みます。</p> <p>ソース・データストアがステージング領域と異なる SQL 準拠のデータベースにある場合は、この LKM の使用を検討してください。ソースおよびターゲットのステージング領域固有の LKM を使用できる場合は、その LKM を使用してください。より最適化されているためパフォーマンスに優れています。たとえば、Oracle ソース・サーバーから Oracle ステージング領域にロードする場合は、LKM Oracle to Oracle (dblink) を使用します。</p>
ロード	LKM SQL to SQL (JYTHON)	<p>SQL 準拠のデータベースから SQL 準拠のステージング領域へデータをロードします。Jython を使用してソース・データベースから選択されたデータを読み取り、動的に作成された一時ステージング表に結果を書き込みます。この LKM では、提供されている基本的な Jython コードを編集して、ソース・データベースとターゲット・ステージング領域の間のデフォルト JDBC データ型マッピングを変更できます。</p> <p>ソース・データストアがステージング領域と異なる SQL 準拠のデータベースにあり、独自のデータ型マッピング方法を指定する場合は、この LKM の使用を検討してください。</p> <p>ソースおよびターゲットのステージング領域固有の LKM を使用できる場合は、その LKM を使用してください。より最適化されているためパフォーマンスに優れています。たとえば、Oracle ソース・サーバーから Oracle ステージング領域にロードする場合は、LKM Oracle to Oracle (dblink) を使用します。</p>

種類	ナレッジ・モジュール	説明
ロード	LKM SQL to SQL (row by row)	<p>ISO-92 データベースから ISO-92 準拠のターゲット・データベースにデータをロードします。この LKM では Jython スクリプトを使用してデータベースから選択されたデータを読み取り、動的に作成されたターゲット一時表に結果を書き込みます。ステージング領域からターゲットへデータをロードし、処理された各行の状態を示します。</p> <p>次のオプションがロギング・メカニズムに使用されません。</p> <ul style="list-style-type: none"> ■ LOG_LEVEL: このオプションは記録されるデータの粒度の設定に使用されます。 次のログ・レベルを設定できます。 <ul style="list-style-type: none"> ■ 0: 何も記録されません。 ■ 1: 選択アクション、削除アクション、挿入アクションなどの JDBC アクションが示されません。 ■ 2: レベル 1 の他に、エラーが発生したすべてのレコードが記録されます。 ■ 3: レベル 2 の他に、処理されたすべてのレコードが記録されます。 ■ LOG_FILE_NAME: 使用されるログ・ファイルのフル・パス。 ■ MAX_ERRORS: エラーの最大数を指定します。このオプションで指定したエラーの最大数に達すると、LKM プロセスが停止します。 <p>重要な注意: 大量の場合、このナレッジ・モジュールはお勧めしません。バルク・ユーティリティ (SQL*LOADER、BULK INSERT など) または直接リンク (DBLINKS、リンク・サーバーなど) を使用する他のモジュールの方が、通常効率的です。</p>
リバースエンジニアリング	RKM SQL (JYTHON)	<p>任意の SQL 準拠データベースから、表、ビュー、システム表および列の JDBC メタデータを取得します。この RKM を使用して、JDBC メタデータを Oracle Data Integrator メタデータに変換するための独自の戦略を指定することも可能です。</p> <p>使用する JDBC ドライバの特異性により、標準の JDBC リバースエンジニアリング・プロセスで問題が発生した場合は、この RKM の使用を検討してください。この RKM では、JDBC ドライバの特異性に合せて、基礎となる Jython コードを編集できます。</p>
Web サービス	SKM SQL	<p>SQL 準拠データベースのデータ・アクセス Web サービスを生成します。データ・アクセス・サービスには、レコードの追加、削除、更新またはフィルタ処理などのデータ操作の他、変更されたデータの取得などのチェンジ・データ・キャプチャ操作が含まれます。データ操作は、データストアの制約による定義に従って、整合性チェックの影響を受けます。</p> <p>データ操作またはチェンジ・データ・キャプチャの Web サービスを生成してサービス指向アーキテクチャ・インフラストラクチャにデプロイする場合は、この SKM の使用を検討してください。使用するデータベース固有の SKM が存在する場合は、かわりにその SKM を使用してください。</p>

Hyperion Essbase

ナレッジ・モジュール

種類	ナレッジ・モジュール	説明
リバースエンジニアリング	RKM Hyperion Essbase	Essbase アプリケーションをリバース・エンジニアリングし、Oracle Data Integrator インタフェースでターゲットまたはソースとして使用されるデータ・モデルを作成します。
統合	IKM SQL to Hyperion Essbase (DATA)	Essbase アプリケーションにデータを統合します。
統合	IKM SQL to Hyperion Essbase (METADATA)	EEssbase アプリケーションにメタデータを統合します。
ロード	LKM Hyperion Essbase DATA to SQL	Essbase アプリケーションから、ステージング領域として使用される任意の SQL 準拠データベースへ、データをロードします。
ロード	LKM Hyperion Essbase METADATA to SQL	Essbase アプリケーションから、ステージング領域として使用される任意の SQL 準拠データベースへ、メタデータをロードします。

プラットフォームのサポート

Oracle Data Integrator Hyperion Essbase ナレッジ・モジュールは、Hyperion 11.1.1 との組合せで動作保証されています。

固有の要件

Hyperion Essbase KM に固有の要件の詳細は、次のドキュメントを参照してください。

- 『Oracle Data Integrator for Hyperion Essbase Readme』
- 『Oracle Data Integrator for Hyperion Essbase スタート・ガイド』
- 『Oracle Data Integrator for Hyperion Essbase ユーザーズ・ガイド』

『Oracle Data Integrator for Hyperion Essbase スタート・ガイド』に記載されているサンプル・ファイルは、Oracle Data Integrator インストール・フォルダの /demo/hyperion/ サブディレクトリにあります。

Hyperion Financial Management

ナレッジ・モジュール

種類	ナレッジ・モジュール	説明
リバースエンジニアリング	RKM Hyperion Financial Management	Financial Management アプリケーションをリバース・エンジニアリングし、Oracle Data Integrator インタフェースでターゲットまたはソースとして使用されるデータ・モデルを作成します。
統合	IKM SQL to Hyperion Financial Management Data	Financial Management アプリケーションにデータを統合します。
統合	IKM SQL to Hyperion Financial Management Dimension	Financial Management アプリケーションにメタデータを統合します。
ロード	LKM Hyperion Financial Management Data to SQL	Financial Management アプリケーションから、ステージング領域として使用される任意の SQL 準拠データベースへ、データをロードします。 RKM Hyperion Financial Management ナレッジ・モジュールでリバース・エンジニアリングされた HFMDData データ・ストアの列名を変更すると、このナレッジ・モジュールは機能しません。
ロード	LKM Hyperion Financial Management Members To SQL	Financial Management アプリケーションから、ステージング領域として使用される任意の SQL 準拠データベースへ、メンバー・リストをロードします。

プラットフォームのサポート

Oracle Data Integrator Hyperion Financial Management ナレッジ・モジュールは、Hyperion 11.1.1 との組合せで動作保証されています。

固有の要件

Financial Management KM に固有の要件の詳細は、次のドキュメントを参照してください。

- 『Oracle Data Integrator Adapter for Hyperion Financial Management Readme』
- 『Oracle Data Integrator Adapter for Hyperion Financial Management スタート・ガイド』
- 『Oracle Data Integrator Adapter for Hyperion Financial Management ユーザーズ・ガイド』

『Oracle Data Integrator Adapter for Hyperion Financial Management スタート・ガイド』に記載されているサンプル・ファイルは、Oracle Data Integrator インストール・フォルダの /demo/hyperion/ サブディレクトリにあります。

Hyperion Planning

ナレッジ・モジュール

種類	ナレッジ・モジュール	説明
リバースエンジニアリング	RKM Hyperion Planning	<p>Planning アプリケーションをリバース・エンジニアリングし、Oracle Data Integrator インタフェースでターゲットとして使用されるデータ・モデルを作成します。</p> <p>各ディメンション（標準ディメンションおよび属性ディメンション）は、適切な列を含むディメンションと同名のデータストアとしてリバースされます。UDA のロードでは「UDA」という名前のデータストアが作成されます。</p>
統合	IKM SQL to Hyperion Planning	<p>Planning アプリケーションにメタデータおよびデータをロードします。</p>

プラットフォームのサポート

Oracle Data Integrator Hyperion Planning ナレッジ・モジュールは、Hyperion 11.1.1 との組合せで動作保証されています。

固有の要件

Hyperion Planning KM に固有の要件の詳細は、次のドキュメントを参照してください。

- 『Oracle Data Integrator Adapter for Hyperion Planning Readme』
- 『Oracle Data Integrator Adapter for Hyperion Planning スタート・ガイド』
- 『Oracle Data Integrator Adapter for Hyperion Planning ユーザーズ・ガイド』

『Oracle Data Integrator Adapter for Hyperion Planning スタート・ガイド』に記載されているサンプル・ファイルは、Oracle Data Integrator インストール・フォルダの /demo/hyperion/ サブディレクトリにあります。

Hypersonic SQL

ナレッジ・モジュール

このデータベースと組み合わせて機能するその他のナレッジ・モジュールについては、[第3章「汎用 SQL」](#)を参照してください。

種類	ナレッジ・モジュール	説明
チェック	CKM HSQL	<p>Hypersonic SQL 表で定義された制約と照合して、データの整合性をチェックします。無効なレコードを拒否して、動的に作成されるエラー表に格納します。フロー制御および静的制御に使用できます。</p> <p>Hypersonic SQL データベースでデータの整合性をチェックする場合は、この CKM の使用を検討してください。</p> <p>この CKM は Hypersonic SQL 用に最適化されています。</p>
ジャーナル化	JKM HSQL Consistent	<p>トリガーを使用して Hypersonic SQL 表の一貫したジャーナル化を行うためのジャーナル化インフラストラクチャを作成します。Hypersonic SQL での一貫性のあるチェンジ・データ・キャプチャを有効にします。</p>
ジャーナル化	JKM HSQL Simple	<p>トリガーを使用して Hypersonic SQL 表の単純なジャーナル化を行うためのジャーナル化インフラストラクチャを作成します。</p> <p>Hypersonic SQL での単純なチェンジ・データ・キャプチャを有効にします。</p>
Web サービス	SKM HSQL	<p>Hypersonic SQL データベースのデータ・アクセス Web サービスを生成します。詳細は、第3章「汎用 SQL」の「SKM SQL」を参照してください。</p> <p>この SKM は Hypersonic SQL データベース用に最適化されています。</p>

ナレッジ・モジュール

このデータベースと組み合わせて機能するその他のナレッジ・モジュールについては、[第3章「汎用 SQL」](#)を参照してください。

種類	ナレッジ・モジュール	説明
統合	IKM DB2 UDB Incremental Update	<p>増分更新モードで IBM DB2 UDB のターゲット表にデータを統合します。この IKM は、データ・フローをステージングするための一時的なステージング表を作成します。その後、ステージング表の内容をターゲット表と比較して、挿入するレコードおよびそれ以外の更新するレコードを推測します。また、CKM を起動してデータ整合性チェックを実行できます。</p> <p>挿入および更新は、最大のパフォーマンスを得るためにバルク・セットベースの処理で実行されます。そのため、この IKM は大量のデータ用に最適化されています。</p> <p>IBM DB2 UDB のターゲット表をロードして、欠落しているレコードを挿入し、既存のレコードを更新する場合は、この IKM の使用を検討してください。</p> <p>この IKM を使用するには、ステージング領域がターゲットと同じデータ・サーバー上にあることが必要です。</p>
統合	IKM DB2 UDB Slowly Changing Dimension	<p>データ・ウェアハウスでタイプ 2 の緩やかに変化するディメンションとして使用される IBM DB2 UDB のターゲット表に、データを統合します。この IKM は、ターゲット・データストアで設定される緩やかに変化するディメンション・メタデータに依存して、新しいバージョンとして挿入する必要があるレコード、または既存のバージョンとして更新する必要があるレコードを判別します。</p> <p>挿入および更新はバルク・セットベースの処理で実行されるため、この IKM は大量のデータ用に最適化されています。</p> <p>IBM DB2 UDB のターゲット表をタイプ 2 の緩やかに変化するディメンションとしてロードする場合は、この IKM の使用を検討してください。</p> <p>この IKM を使用するには、ステージング領域がターゲットと同じデータ・サーバー上にあり、適切な緩やかに変化するディメンション・メタデータがターゲット・データストア上で設定されていることが必要です。</p>

種類	ナレッジ・モジュール	説明
ジャーナル化	JKM DB2 UDB Consistent	<p>トリガーを使用して IBM DB2 UDB 表の一貫したジャーナル化を行うためのジャーナル化インフラストラクチャを作成します。</p> <p>IBM DB2 UDB での一貫性のあるチェンジ・データ・キャプチャを有効にします。</p>
ジャーナル化	JKM DB2 UDB Simple	<p>トリガーを使用して IBM DB2 UDB 表の単純なジャーナル化を行うためのジャーナル化インフラストラクチャを作成します。</p> <p>IBM DB2 UDB での単純なチェンジ・データ・キャプチャを有効にします。</p>
ロード	LKM DB2 UDB to DB2 UDB (EXPORT_IMPORT)	<p>固有の EXPORT / IMPORT コマンドを使用して、IBM DB2 UDB ソース・データベースから IBM DB2 UDB ステージング領域データベースヘデータをロードします。</p> <p>このモジュールでは、EXPORT CLP コマンドを使用して一時ファイルにデータを抽出します。その後、IMPORT CLP コマンドを使用してターゲットの DB2 UDB ステージング表にデータをロードします。この方法は通常、大量のデータを処理する場合に標準の LKM SQL to SQL より効率的です。</p> <p>ソース表が DB2 UDB データベースにあり、ステージング領域が異なる DB2 UDB データベースにある場合は、この LKM の使用を検討してください。</p>
ロード	LKM File to DB2 UDB (LOAD)	<p>固有の CLP LOAD コマンドを使用して、ファイルから DB2 UDB ステージング領域データベースヘデータをロードします。</p> <p>この LKM では、ファイルの種類（固定またはデリミタ付き）に応じて、適切な LOAD スクリプトが一時ディレクトリに生成されます。その後、CLP LOAD コマンドによってこのスクリプトが実行され、実行終了時に自動的に削除されます。この方法では固有の IBM DB2 ローダーが使用されるため、大量のデータを処理する場合は、標準の LKM File to SQL より効率的です。</p> <p>ソースが大きいフラット・ファイルで、ステージング領域が IBM DB2 UDB データベースの場合は、この LKM の使用を検討してください。</p>
ロード	LKM SQL to DB2 UDB	<p>任意の汎用 SQL ソース・データベースから IBM DB2 UDB ステージング領域ヘデータをロードします。この LKM は、第 3 章「汎用 SQL」に記載されている標準の LKM SQL to SQL と似ていますが、IBM DB2 UDB の追加パラメータを指定できる点で異なります。</p>
ロード	LKM SQL to DB2 UDB (LOAD)	<p>CLP LOAD コマンドを使用して、任意の汎用 SQL ソース・データベースから IBM DB2 UDB ステージング領域ヘデータをロードします。</p> <p>この LKM では、一時ファイルにソース・データがアンロードされ、CLP LOAD コマンドを使用して IBM DB2 固有のローダーがコールされて、ステージング表が移入されます。この方法では固有の IBM DB2 ローダーが使用されるため、通常、大量のデータを処理する場合は、LKM SQL to SQL または LKM SQL to DB2 UDB より効率的です。</p> <p>汎用データベースに大量のソース・データがあり、ステージング領域が IBM DB2 UDB データベースの場合は、この LKM の使用を検討してください。</p>

種類	ナレッジ・モジュール	説明
Web サービス	SKM IBM UDB	IBM DB2 UDB データベース用のデータ・アクセス Web サービスを生成します。詳細は、 第 3 章「汎用 SQL」 の「SKM SQL」を参照してください。 この SKM は IBM DB2 UDB データベース用に最適化されています。

固有の要件

IBM DB2 UDB 用の一部のナレッジ・モジュールでは、オペレーティング・システムのコールを使用して IBM CLP コマンド・プロセッサを起動し、効率的なロードを実行します。このようなナレッジ・モジュールの使用時には、次の制限が適用されます。

1. Oracle Data Integrator のエージェントを実行するマシン上に、IBM DB2 UDB Command Line Processor (CLP) および DB2 UDB Connect ソフトウェアがインストールされている必要があります。
2. トポロジで定義されるサーバー名は、これらのサーバーに使用される IBM DB2 UDB の接続文字列と一致する必要があります。

これらのトピックの追加情報は、IBM DB2 のドキュメントを参照してください。

ナレッジ・モジュール

このデータベースと組み合わせて機能するその他のナレッジ・モジュールについては、[第3章「汎用 SQL」](#)を参照してください。

種類	ナレッジ・モジュール	説明
統合	IKM DB2 400 Incremental Update	<p>増分更新モードで IBM DB2/400 のターゲット表にデータを統合します。この IKM は、データ・フローをステージングするための一時的なステージング表を作成します。その後、ステージング表の内容をターゲット表と比較して、挿入するレコードおよびそれ以外の更新するレコードを推測します。また、CKM を起動してデータ整合性チェックを実行できます。</p> <p>挿入および更新は、最大のパフォーマンスを得るためにバルク・セットベースの SQL 処理で実行されます。そのため、この IKM は大量のデータ用に最適化されています。</p> <p>SQL を使用して、IBM DB2/400 のターゲット表をロードし、欠落しているレコードを挿入して既存のレコードを更新する場合は、この IKM の使用を検討してください。</p> <p>この IKM を使用するには、ステージング領域がターゲットと同じデータ・サーバー上にあることが必要です。</p>
統合	IKM DB2 400 Incremental Update (CPYF)	<p>増分更新モードで IBM DB2/400 のターゲット表にデータを統合します。この IKM は IKM DB2 400 Incremental Update と似ていますが、セットベースの SQL 操作ではなく、CPYF 固有の OS/400 コマンドを使用してターゲット表に書き込む点で異なります。特定のケースでは、この IKM を使用する方が効率的です。</p> <p>SQL ではなく、固有の OS/400 コマンドを使用して、IBM DB2/400 のターゲット表をロードし、欠落しているレコードを挿入して既存のレコードを更新する場合は、この IKM の使用を検討してください。</p>

種類	ナレッジ・モジュール	説明
統合	IKM DB2 400 Slowly Changing Dimension	<p>データ・ウェアハウスでタイプ 2 の緩やかに変化するディメンションとして使用される IBM DB2/400 のターゲット表に、データを統合します。この IKM は、ターゲット・データストアで設定される緩やかに変化するディメンション・メタデータに依存して、新しいバージョンとして挿入する必要があるレコード、または既存のバージョンとして更新する必要があるレコードを判別します。</p> <p>挿入および更新はバルク・セットベースの処理で実行されるため、この IKM は大量のデータ用に最適化されています。</p> <p>IBM DB2/400 のターゲット表をタイプ 2 の緩やかに変化するディメンションとしてロードする場合は、この IKM の使用を検討してください。</p> <p>この IKM を使用するには、ステージング領域がターゲットと同じデータ・サーバー上にあり、適切な緩やかに変化するディメンション・メタデータがターゲット・データストア上で設定されていることが必要です。</p>
ジャーナル化	JKM DB2 400 Consistent	<p>トリガーを使用して IBM DB2/400 表の一貫したジャーナル化を行うためのジャーナル化インフラストラクチャを作成します。</p> <p>IBM DB2/400 での一貫性のあるチェンジ・データ・キャプチャを有効にします。</p>
ジャーナル化	JKM DB2 400 Simple	<p>トリガーを使用して IBM DB2/400 表の一貫したジャーナル化を行うためのジャーナル化インフラストラクチャを作成します。</p> <p>IBM DB2/400 での一貫性のあるチェンジ・データ・キャプチャを有効にします。</p>
ロード	LKM DB2 400 to DB2 400	<p>固有の CRTDDMF コマンドおよび CPYF コマンドを使用して、IBM DB2/400 ソース・データベースから IBM DB2/400 ステージング領域データベースヘデータをロードします。</p> <p>この LKM では、CRTDDMF を使用してターゲット上に DDM ファイルを作成し、CPYF を使用してソース・ステージング表からこの DDM ファイルヘデータを転送します。</p> <p>この方法は通常、大量のデータを処理する場合に標準の LKM SQL to SQL より効率的です。</p> <p>ソース表が DB2/400 データベースにあり、ステージング領域が異なる DB2/400 データベースにある場合は、この LKM の使用を検討してください。</p>
ロード	LKM SQL to DB2 400 (CPYFRMIMPF)	<p>CPYFRMIMPF コマンドを使用して、任意の汎用 SQL ソース・データベースから IBM DB2/400 ステージング領域ヘデータをロードします。</p> <p>この LKM では、一時ファイルにソース・データがアンロードされ、CPYFRMIMPF コマンドを使用して IBM DB2/400 固有のローダーがコールされて、ステージング表が移入されます。</p> <p>この方法では固有の IBM DB2/400 ローダーが使用されるため、通常、大量のデータを処理する場合は、LKM SQL to SQL より効率的です。</p> <p>汎用データベースに大量のソース・データがあり、ステージング領域が IBM DB2/400 データベースの場合は、この LKM の使用を検討してください。</p>

種類	ナレッジ・モジュール	説明
リバースエンジニアリング	RKM DB2 400	<p>IBM DB2/400 固有のメタデータを取得します。この RKM を使用すると、物理ファイル、データベース表、データベース・ビューおよび一意キーのメタデータを抽出することも可能です。また、基礎となる DB2/400 のカタログ表にアクセスするため、この RKM を使用して、標準の SQL リソースの詳細名のかわりにリソースの 10 桁の物理名を取得することもできます。</p> <p>表の名前が 10 桁より長く、OS/400 で固有のコマンドを使用する場合は、この RKM の使用を検討してください。</p>

固有の要件

IBM DB2/400 用の一部のナレッジ・モジュールでは、オペレーティング・システムのコールを使用して IBM iSeries のコマンドが起動され、効率的なロードが実行されます。このようなナレッジ・モジュールの使用時には、次の制限が適用されます。

1. OS/400 のコマンドは、10 桁の表名のみを受け入れます。SQL の表名ではなく、データストアの表の物理名を取得するには、特定の RKM DB2/400 を使用してください。
2. Oracle Data Integrator のエージェントがターゲットの iSeries マシンにインストールされている必要があります。適切な手順は、インストレーション・ガイドを参照してください。

ナレッジ・モジュール

このデータベースと組み合わせて機能するその他のナレッジ・モジュールについては、[第3章「汎用 SQL」](#)を参照してください。

種類	ナレッジ・モジュール	説明
統合	IKM Informix Incremental Update	<p>増分更新モードで IBM Informix のターゲット表にデータを統合します。この IKM は、データ・フローをステージングするための一時的なステージング表を作成します。その後、ステージング表の内容をターゲット表と比較して、挿入するレコードおよびそれ以外の更新するレコードを推測します。また、CKM を起動してデータ整合性チェックを実行できます。</p> <p>挿入および更新は、最大のパフォーマンスを得るためにバルク・セットベースの処理で実行されます。そのため、この IKM は大量のデータ用に最適化されています。</p> <p>IBM Informix のターゲット表をロードして、欠落しているレコードを挿入し、既存のレコードを更新する場合は、この IKM の使用を検討してください。</p> <p>この IKM を使用するには、ステージング領域がターゲットと同じデータ・サーバー上にあることが必要です。</p>
ジャーナル化	JKM Informix Consistent	<p>トリガーを使用して IBM Informix 表の一貫したジャーナル化を行うためのジャーナル化インフラストラクチャを作成します。</p> <p>IBM Informix での一貫性のあるチェンジ・データ・キャプチャを有効にします。</p>
ジャーナル化	JKM Informix Simple	<p>トリガーを使用して IBM Informix 表の単純なジャーナル化を行うためのジャーナル化インフラストラクチャを作成します。</p> <p>IBM Informix での単純なチェンジ・データ・キャプチャを有効にします。</p>

種類	ナレッジ・モジュール	説明
ロード	LKM Informix to Informix (SAME SERVER)	<p>同じサーバー内にあるソースの Informix データベースからターゲットの Informix ステージング領域へ、データをロードします。</p> <p>この LKM では、ソース・データベース内のビューおよびステージング領域データベース内のシノニムが作成されます。この方法は通常、大量のデータを処理する場合に標準の LKM SQL to SQL より効率的です。</p> <p>ソース表が IBM Informix データベースにあり、ステージング領域が同じ Informix サーバーの IBM Informix データベースにある場合は、この LKM の使用を検討してください。</p>
リバースエンジニアリング	RKM Informix	<p>表、ビュー、列、主キーおよび一意でない索引について、IBM Informix 固有のメタデータを取得します。この RKM は、基礎となる Informix カタログ表にアクセスしてメタデータを取得します。</p> <p>デフォルトの JDBC リバースエンジニアリング・プロセスによって提供されていない追加のメタデータを Informix カタログから抽出する場合は、この RKM の使用を検討してください。</p>
リバースエンジニアリング	RKM Informix SE	<p>表、ビュー、列、主キーおよび一意でない索引について、IBM Informix SE 固有のメタデータを取得します。この RKM は、基礎となる Informix SE カタログ表にアクセスしてメタデータを取得します。</p> <p>デフォルトの JDBC リバースエンジニアリング・プロセスによって提供されていない追加のメタデータを Informix SE カタログから抽出する場合は、この RKM の使用を検討してください。</p>
Web サービス	SKM Informix	<p>IBM Informix データベースのデータ・アクセス Web サービスを生成します。詳細は、第3章「汎用 SQL」の「SKM SQL」を参照してください。</p> <p>この SKM は IBM Informix データベース用に最適化されています。</p>

JD Edwards EnterpriseOne

概要

JD Edwards (JDE) EnterpriseOne は包括的な ERP ソフトウェアの統合型アプリケーション・スイートです。このスイートは、事業価値、標準ベースのテクノロジーおよび各業界の深い知見を組合せ、ビジネス・ソリューションを提供します。この項では、Oracle Data Integrator での JD Edwards EnterpriseOne ナレッジ・モジュールの使用の概要と方法について説明します。

JDE EnterpriseOne ナレッジ・モジュール

JDE EnterpriseOne 用 Oracle Data Integrator ナレッジ・モジュール (JDE KM) では、Oracle Data Integrator を介した JDE EnterpriseOne プラットフォームと任意のデータベース・アプリケーションの接続および統合が提供されます。

JDE KM では、次の処理を行うためにデータベースレベルの JDE EnterpriseOne 用の高度な統合方法を使用します。

- JDE EnterpriseOne データ構造のリバースエンジニアリング
- JDE EnterpriseOne からのデータの読取り (直接データベース統合)
- Z 表を介した JDE アプリケーションへのデータの書込み (インタフェース表統合)

ODI には、JDE EnterpriseOne データを処理するナレッジ・モジュールが 2 つ用意されています。

種類	ナレッジ・モジュール	説明
リバースエンジニアリング	RKM JDE Enterprise One Oracle	Oracle データベースにインストールされている JDE EnterpriseOne から、表およびインタフェース表などのアプリケーション・オブジェクトのメタデータを取得することによるデータストア定義の作成をサポートします。
統合	IKM JDE Enterprise One Control Append (UBE)	<p>任意のソースから JDE EnterpriseOne へのデータのロードをサポートします。データが制御追加モードで EnterpriseOne の Z 表に統合されます。</p> <ul style="list-style-type: none"> ■ データは制御されます。無効なデータはエラー表に分離され、リサイクルできます。 ■ RunUBE バッチ・コマンドを使用して JDE EnterpriseOne への統合が行われます。

プラットフォームのサポート

JDE KM は、次のプラットフォームとバージョンで動作保証されています。

- Oracle Database にインストールされた JDE EnterpriseOne 8.12

インストールおよび構成

JDE KM を使用するための特別な Oracle Data Integrator 構成はありません。

Oracle Data Integrator JDE EnterpriseOne KM の使用

JDE EnterpriseOne と Oracle Data Integrator JDE KM を組み合わせて使用するには、次の処理が必要です。

1. トポロジの定義
2. プロジェクトの設定
3. JDE 表のリバースエンジニアリング
4. 統合インタフェースで JDE データストアをソースまたはターゲットとして使用

トポロジの定義

Oracle Data Integrator JDE KM を使用したプロジェクトに固有の Oracle Data Integrator でのトポロジの作成手順は次のとおりです。

1. トポロジ・マネージャを使用してマスター・リポジトリに接続します。
2. Oracle Server テクノロジを使用してデータ・サーバーを作成します。このデータ・サーバーは、JDE データを格納する Oracle データベース・インスタンスを指します。
3. このデータ・サーバーに物理スキーマを作成します。このスキーマは、リバースエンジニアリングする JDE 表を含む Oracle スキーマ・データベースを指します。
4. 適切なコンテキストでこの物理スキーマに対する論理スキーマを作成します。

これで JDE のリバースエンジニアリングを行うことができます。

注意： JDE 表を格納する Oracle スキーマは、物理スキーマの定義で作業スキーマとして定義しないでください。また、このスキーマまたはデータベースは、インタフェースのステージング領域として使用しないでください。

プロジェクトの設定

次の KM を Oracle Data Integrator プロジェクトにインポートします。

- IKM JDE Enterprise One Control Append (UBE)
- RKM JDE Enterprise One Oracle
- この他に、Oracle データベースのデータ抽出およびデータ品質チェックを行う標準の Oracle LKM および CKM をインポートします。使用できる KM のリストは、このガイドの第 17 章「[Oracle Database](#)」を参照してください。

KM のインポートの詳細は、『Oracle Data Integrator ユーザーズ・ガイド』を参照してください。

JDE 表のリバースエンジニアリング

RKM JDE Enterprise One Oracle では JDE 表をリバースエンジニアリングできます。この RKM では、表およびインタフェース表などの JDE オブジェクトからメタデータが取得されます。

JDE のリバースエンジニアリングを行うには、次のようにします。

1. Oracle テクノロジーおよび JDE 接続の構成時に作成された論理スキーマに基づいてモデルを作成します。
2. このモデルで「リバース」タブを選択します。
 1. 「カスタマイズ済」を選択します。
 2. KM のリストから「RKM JDE Enterprise One Oracle」を選択します。
 3. 次のように RKM のオプションを設定します。
 - JDE_CENTRAL_OBJECTS: JDE セントラル・オブジェクトを格納する Oracle スキーマを指定します。
 - JDE_DATA_DICTIONARY: JDE データ・ディクショナリを格納する Oracle スキーマを指定します。
 - JDE_OBJECT_LIBRARIAN: JDE オブジェクト・ライブラリアンを格納する Oracle スキーマを指定します。
 - JDE_CONTROL_TABLES: 制御表スキーマを指定します。

注意: JDE_CENTRAL_OBJECTS、JDE_DATA_DICTIONARY、JDE_OBJECT_LIBRARIAN および JDE_CONTROL_TABLES のオプションに必要なスキーマを見つけるには、アプリケーション・マネージャにたずねるか、表 F98611 (データ・ソース・マネージャ) の問合せを行います。

- JDE_DATA_TABLES: データ表をリバースエンジニアリングする場合、このオプションを YES に設定します。
 - JDE_Z_TABLES: インタフェース表 (Z 表) をリバースエンジニアリングする場合、このオプションを YES に設定します。
 - JDE_MODULES: JDE システムの短縮名 (00: 基本環境、01: 住所録、02: 電子メール、など) を示します。
 - JDE_LANGUAGE: オブジェクト説明の取得に使用される言語 (E: 英語、F: フランス語、S: スペイン語、など) を示します。
3. リバースする表を選択するために「マスク」フィールドでリバース・マスクを指定します。「リバース」タブの「マスク」フィールドでは、リバースエンジニアリングされるオブジェクトが名前を基にフィルタ処理されます。「マスク」フィールドは空にしないでください。少なくともパーセント記号 (%) を含める必要があります。
 4. 「適用」をクリックして「リバース」をクリックします。
 5. 実行ログでリバースエンジニアリング・プロセスを確認できます。

注意: リバースエンジニアリング・プロセスには数分かかる場合があります。実行ログのツリーのリバースされたモジュールにリバースされたデータストアが表示されるまで待ってください。

リバースエンジニアリング・プロセスでは、データストアがモジュールごとにまとめて返されます。これらのデータストアをインタフェースのソースまたはターゲットとして使用できます。

統合インタフェースでの JDE のソースとしての使用

RKM JDE Enterprise One Oracle を使用してリバースエンジニアリングを行ったら、JDE データ表をインタフェースのソースとして使用し、JDE アプリケーションからデータを抽出して他のシステム（データ・ウェアハウス、他のデータベースなど）に統合できます。この状況で JDE をソースとして使用することは、インタフェースでソースとして Oracle データベースを使用することと同じです。標準の Oracle KM をこの用途で使用できます。

統合インタフェースでの JDE Z 表のターゲットとしての使用

RKM JDE Enterprise One Oracle を使用してリバースエンジニアリングを行ったら、JDE Z 表をインタフェースのターゲットとして使用し、IKM JDE Enterprise One Control Append (UBE) を介して任意のシステムから JDE アプリケーションにデータをロードできます。

IKM JDE Enterprise One Control Append (UBE) を使用した JDE へのデータの統合

JDE EnterpriseOne へのデータの統合は 2 段階で行われます。第 1 段階では、Z 表（ステージング表）のセットがロードされます。第 2 段階では、RunUBE コマンドが起動され、これらの Z 表から JDE EnterpriseOne にデータが統合されます。

Oracle Data Integrator では、RunUBE コマンドを自動的にコールして JDE に書き込むことができます。RunUBE コールは、JDE に移入するための適切な Z 表をロードした後でのみ IKM でアクティブ化できます。Z 表をロードする機能と RunUBE コマンドのコールは、IKM JDE Enterprise One Control Append (UBE) によって提供されます。

Oracle Data Integrator でデータを JDE に統合するには次のようにします。

1. Z 表をターゲット・データストアとして統合インタフェースを作成します。
2. 通常どおり結合、フィルタおよびマッピングを作成します。
3. 「フロー」タブで「IKM JDE Enterprise One Control Append (UBE)」を選択します。
4. 次のように RunUBE コマンドを実行する KM オプションを指定します。
 1. JDE_RUNUBE オプションを Yes に設定します。
 2. RunUBE コマンドを実行する JDE_DIRECTORY を指定します。
 3. パスワード・ファイルを作成する場合は、パスワードに関連するオプションを次のように設定します。
 - JDE_CREATE_PWD_FILE: Yes に設定します。

注意： UNIX または iSeries 環境で RunUBE のセキュリティを向上させるために、RunUBE コマンドの発行時、JDE_PWD_FILE オプションで指定されたテキスト・ファイルがシステムで読み取られ、テキスト・ファイルに示された JD Edwards EnterpriseOne のユーザー ID とパスワードが使用されます。

- JDE_PWD_FILE: パスワード・セキュリティ・ファイルの絶対パスを指定します。このファイルには、JDE_USER_ID と JDE_PWD のオプションで指定されたユーザー ID とパスワードが含まれています。
- JDE_DELETE_PWD_FILE: パスワード・ファイルを削除する場合、D と入力します。パスワード・ファイルを保持する場合、F と入力します。

注意： コマンドの実行後にパスワード・ファイルを削除する場合でも、ファイル・システムのセキュアな場所にファイルを保存します。

- JDE_USER_ID: JDE EnterpriseOne ユーザー ID。ユーザーはレポートを実行する権限を持っている必要があります。
 - JDE_PWD: ユーザー ID に対応する EnterpriseOne パスワード。
5. 次の RunUBE コマンドのパラメータも指定する必要があります。
- JDE_ENVIRONMENT: JDE EnterpriseOne 環境。
 - JDE_ROLE: JDE EnterpriseOne ロール。
 - JDE_REPORT: 処理するレポートのシステム名。フラット・ファイルの場合は APS アウトバウンド・プロセッサ (R34A400) および APS インバウンド・プロセッサ (R34A410)、XML ファイルの場合は APS SCBM 2.0 アウトバウンド・プロセッサ (R34A700) および APS マスター・インバウンド・プロセッサ (R34A820) などです。
 - JDE_VERSION: XJDE0001 などの処理するレポートのバージョンの名前。バージョン名を入力する必要があります。レポートのテンプレートは発行できません。
 - JDE_JOB_QUEUE: システムでバッチ・ジョブを送る QBATCH などのジョブ・キューの名前。
 - JDE_PROCESSING_MODE: 処理モード。バッチ処理を使用する場合、B と入力します。この場合、システムでジョブ制御ステータス・マスター表 (F986110) を使用して、レポートにキューでの位置が割り当てられます。インタラクティブ・モードを使用する場合、I と入力します。このモードでは、JDE EnterpriseOne キューイング・メカニズム外でレポートが即座に実行されます。
 - JDE_HOLD_CODE: 保留コード。ジョブの完了後すぐに出力をプリンタに送信する場合、P と入力します。処理されたファイルを印刷せずに保持する場合、H と入力します。ジョブはサーバーの使用プログラム (P986116) を使用して後で印刷できます。これはシステム管理ツール・メニュー (GH9011) からアクセスできます。
 - JDE_SAVE_CODE: 保存コード。処理の完了後ファイルを保存する場合、S と入力します。削除オプション (D) は、今後使用するための予備です。現在、削除オプションは無効です。

IKM JDE Enterprise One Control Append (UBE) の制限

- ターゲット表が別の表 (外部キー) によって参照されている場合、TRUNCATE オプションは機能しません。
- RECYCLE_ERRORS オプションを使用する場合、インタフェース用に更新キーを設定する必要があります。
- このモジュールをジャーナル化されたソース表と組み合わせて使用する場合、ソースの削除を含めないようデータが自動的にフィルタ処理されます。
- FLOW_CONTROL および STATIC_CONTROL オプションは、無効なデータを分離するためにチェック・ナレッジ・モジュールをコールします (CKM が設定されていない場合、エラーが発生します)。統合インタフェースで TEMPORARY ターゲット・データストアに移入される場合は、この 2 つのオプションを NO に設定する必要があります。
- RunUBE コマンドは JDE サーバーで実行する必要があります。
- Oracle Data Integrator ランタイム・エージェントがこのサーバーにインストールされている必要があります。
- RunUBE コマンドが起動されたかどうかの情報以外、RunUBE コマンドでプログラムの実行の詳細は示しません。プログラムの実行について確認するには、JDE サーバーで作成されたログ・ファイルを見るか、JDE アプリケーションに接続してジョブ・ステータスの表示アプリケーション (アプリケーション = P986110、フォーム = W986116A) を検索します。

ナレッジ・モジュール・オプションのリファレンス

RKM JDE Enterprise One Oracle

オプション	値	必須	説明												
JDE_CENTRAL_OBJECTS	PD812	はい	JDE セントラル・オブジェクトを格納する Oracle スキーマ。												
JDE_DATA_DICTIONARY	DD812	はい	JDE データ・ディクショナリを格納する Oracle スキーマ。												
JDE_OBJECT_LIBRARIAN	OL812	はい	JDE オブジェクト・ライブラリアンを格納する Oracle スキーマ。												
JDE_CONTROL_TABLES	PRODCTL	はい	制御表スキーマ。												
JDE_DATA_TABLES	<u>Yes</u> No	はい	データ表をリバースするためのフラグ。このオプションを Yes に設定すると、データ表がリバースされます。												
JDE_Z_TABLES	<u>Yes</u> No	はい	インタフェース表をリバースするためのフラグ。このオプションを Yes に設定すると、インタフェース表 (Z 表) がリバースされます。												
JDE_MODULES	01	はい	次のマッピングに基づいて、JDE モジュールの JDE システム短縮名を指定します。 <table border="1" data-bbox="967 934 1373 1199"> <thead> <tr> <th>短縮名</th> <th>完全名</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>基本環境</td> </tr> <tr> <td>01</td> <td>住所録</td> </tr> <tr> <td>02</td> <td>電子メール</td> </tr> <tr> <td>03</td> <td>売掛管理</td> </tr> <tr> <td>他</td> <td>...</td> </tr> </tbody> </table>	短縮名	完全名	00	基本環境	01	住所録	02	電子メール	03	売掛管理	他	...
短縮名	完全名														
00	基本環境														
01	住所録														
02	電子メール														
03	売掛管理														
他	...														
JDE_LANGUAGE	<u>E</u> F S	はい	オブジェクト説明とコメントの言語。 サポートされている言語の例は次のとおりです。 <table border="1" data-bbox="967 1474 1373 1606"> <tbody> <tr> <td>E</td> <td>英語</td> </tr> <tr> <td>F</td> <td>フランス語</td> </tr> <tr> <td>S</td> <td>スペイン語</td> </tr> </tbody> </table>	E	英語	F	フランス語	S	スペイン語						
E	英語														
F	フランス語														
S	スペイン語														

IKM JDE Enterprise One Control Append (UBE)

オプション	値	必須	説明
INSERT	<u>Yes</u> No	はい	インタフェースのターゲット・データストアへのデータの挿入を自動的に試行します。
COMMIT	<u>Yes</u> No	はい	統合後コミット。 ターゲット・データストアに挿入されたすべてのデータをコミットします。
FLOW_CONTROL	<u>Yes</u> No	はい	フロー制御のアクティブ化。 フロー制御を行う場合、このオプションを Yes に設定します。
RECYCLE_ERRORS	Yes <u>No</u>	はい	以前のエラーのリサイクル。 以前の制御から拒否されたデータをリサイクルする場合、このオプションを Yes に設定します。
STATIC_CONTROL	Yes <u>No</u>	はい	統合後制御。 ターゲット・データの挿入または更新後にターゲット表を制御する場合、このオプションを Yes に設定します。
TRUNCATE	Yes <u>No</u>	はい	ターゲット・データストアの切捨て。 ターゲット・データストアを切り捨てる場合、このオプションを Yes に設定します。
DELETE_ALL	Yes <u>No</u>	はい	ターゲット・データストアのすべての行を削除する場合、このオプションを YES に設定します。
CREATE_TARG_TABLE	Yes <u>No</u>	はい	ターゲット表を作成する場合、このオプションを Yes に設定します。
ODI_ERR_FILE	c:¥temp¥err.txt	いいえ	ODI で生成されるエラー・ファイルの完全なパス。
JDE_RUNUBE	<u>Yes</u> No	はい	RunUBE コマンドを実行する場合、 YES に設定します。
JDE_DIRECTORY	C:¥JDEdwards¥DDP¥812¥system¥bin32	JDE_RUNUBE が Yes に設定されている場合、 はい	RunUBE コマンドを実行するディレクトリ。
JDE_CREATE_PWD_FILE	Yes <u>No</u>	JDE_RUNUBE が Yes に設定されている場合、 はい	ODI でパスワード・ファイルを生成する場合、 Yes に設定します。
JDE_PWD_FILE	C:¥JDEdwards¥DDP¥812¥pwd.txt	JDE_CREATE_PWD_FILE が Yes に設定されている場合、 はい	接続用セキュリティ・ファイル。 ユーザー ID とパスワードを含むパスワード・セキュリティ・ファイルの完全なパス。
JDE_DELETE_PWD_FILE	d D f F	JDE_RUNUBE が Yes に設定されている場合、 はい	パスワード・ファイルを削除するかどうか。 d D は、パスワード・ファイルの自動削除を表します。 f F では、パスワード・ファイルは削除されません。

オプション	値	必須	説明
JDE_USER_ID	JDE	JDE_CREATE_PWD_FILE が Yes に設定されている場合、はい	JD Edwards EnterpriseOne ユーザー ID。レポートを実行する権限を持っている必要があります。
JDE_PWD	password	JDE_CREATE_PWD_FILE が Yes に設定されている場合、はい	JD Edwards EnterpriseOne ユーザー ID に対応する JD Edwards EnterpriseOne パスワード。
JDE_ENVIRONMENT	PD812	JDE_RUNUBE が Yes に設定されている場合、はい	JD Edwards EnterpriseOne 環境。
JDE_ROLE	*ALL	JDE_RUNUBE が Yes に設定されている場合、はい	JD Edwards EnterpriseOne ロール。
JDE_REPORT	R014021	JDE_RUNUBE が Yes に設定されている場合、はい	処理する JDE E1 レポートのシステム名。
JDE_VERSION	XJDE0001	JDE_RUNUBE が Yes に設定されている場合、はい	XJDE0001 などの処理するレポートの JDE E1 バッチ・バージョンの名前。バージョン名を入力する必要があります。レポート・テンプレートは発行できません。
JDE_JOB_QUEUE	QBATCH	JDE_RUNUBE が Yes に設定されている場合、はい	システムでバッチ・ジョブを送る QBATCH などの JDE E1 ジョブ・キューの名前。
JDE_PROCESSING_MODE	B I	JDE_RUNUBE が Yes に設定されている場合、はい	JDE E1 のインタラクティブまたはバッチ処理モード。バッチ処理を使用する場合、B と入力します。この場合、システムでレポートにキューでの位置が割り当てられます。インタラクティブ・モードの場合、I と入力します。この場合、JD Edwards EnterpriseOne キューイング・メカニズム外でレポートが即座に実行されます。
JDE_HOLD_CODE	H P	JDE_RUNUBE が Yes に設定されている場合、はい	JDE E1 実行モード。 保留コード。ジョブの完了後すぐに出力をプリンタに送信する場合、P と入力します。処理されたファイルを印刷せずに保持する場合、H と入力します。ジョブはサーバーの使用プログラム (P986116) を使用して後で印刷できます。これはシステム管理ツール・メニュー (GH9011) からアクセスできます。
JDE_SAVE_CODE	D S	JDE_RUNUBE が Yes に設定されている場合、はい	JDE E1 の保存または削除オプション。 保存コード。処理の完了後ファイルを保存する場合、S と入力します。ファイルを削除する場合、D と入力します。

ナレッジ・モジュール

この項に記載されているナレッジ・モジュールは、最も一般的な JMS 準拠ミドルウェアのほとんどに適用されます。適用されるミドルウェアには、Oracle JMS、Sonic MQ、IBM Websphere MQ などが含まれます。これらのほとんどのナレッジ・モジュールには、メッセージを確実に配信するためのトランザクション処理が含まれます。また、これらのすべてのナレッジ・モジュールで、Oracle Data Integrator の JMS 用 JDBC ドライバが使用されます。この JDBC ドライバには、メッセージのキューおよびトピックを問い合わせるための簡略化された SQL 構文が含まれます。したがって、ドライバでは、SQL SELECT 文は一連の JMS メッセージの取得と解釈され、SQL INSERT 文は一連の JMS メッセージの入力と解釈されます。

種類	ナレッジ・モジュール	説明
統合	IKM SQL to JMS Append	<p>任意の SQL 準拠ステージング領域から JMS 準拠のメッセージのキューまたはトピックへ、テキストまたはバイナリ形式でデータを統合します。</p> <p>データを変換してターゲットの JMS キューまたはトピックにエクスポートする場合は、この IKM の使用を検討してください。ほとんどのソース・データストアが同じデータ・サーバー上にある場合は、余分なロード・フェーズ (LKM) を発生させないために、このデータ・サーバーをステージング領域として使用することをお勧めします。</p> <p>この IKM を使用するには、ステージング領域がターゲットと異なる必要があります。</p>
統合	IKM SQL to JMS XML Append	<p>任意の SQL 準拠ステージング領域から JMS 準拠のメッセージのキューまたはトピックへ、XML 形式でデータを統合します。</p> <p>データを変換してターゲットの JMS キューまたはトピックに XML 形式でエクスポートする場合は、この IKM の使用を検討してください。ほとんどのソース・データストアが同じデータ・サーバー上にある場合は、余分なロード・フェーズ (LKM) を発生させないために、このデータ・サーバーをステージング領域として使用することをお勧めします。</p> <p>この IKM を使用するには、ステージング領域がターゲットと異なる必要があります。</p>

種類	ナレッジ・モジュール	説明
ロード	LKM JMS to SQL	<p>テキストまたはバイナリの JMS 準拠メッセージのキューまたはトピックから、ステージング領域として使用される任意の SQL 準拠データベースへ、データをロードします。エージェントを使用してソース・キュー / トピックから選択されたメッセージを読み取り、動的に作成された一時ステージング表に結果を書き込みます。</p> <p>メッセージを確実に配信するために、メッセージのコンシューマ（またはサブスクライバ）は、IKM によってデータが実際にターゲットに統合されるまで、読取りをコミットしません。</p> <p>ソース・データストアの 1 つがテキストまたはバイナリの JMS メッセージの場合は、この LKM の使用を検討してください。</p>
ロード	LKM JMS XML to SQL	<p>XML の JMS 準拠メッセージのキューまたはトピックから、ステージング領域として使用される任意の SQL 準拠データベースへ、データをロードします。エージェントを使用してソース・キュー / トピックから選択されたメッセージを読み取り、動的に作成された一時ステージング表に結果を書き込みます。</p> <p>メッセージを確実に配信するために、メッセージのコンシューマ（またはサブスクライバ）は、IKM によってデータが実際にターゲットに統合されるまで、読取りをコミットしません。</p> <p>ソース・データストアの 1 つが XML の JMS メッセージの場合は、この LKM の使用を検討してください。</p>

Microsoft Access

ナレッジ・モジュール

このデータベースと組み合わせて機能するその他のナレッジ・モジュールについては、[第3章「汎用 SQL」](#)を参照してください。

種類	ナレッジ・モジュール	説明
統合	IKM Access Incremental Update	<p>増分更新モードで Microsoft Access のターゲット表にデータを統合します。この IKM は、データ・フローをステージングするための一時的なステージング表を作成します。その後、ステージング表の内容をターゲット表と比較して、挿入するレコードおよびそれ以外の更新するレコードを推測します。また、CKM を起動してデータ整合性チェックを実行できます。</p> <p>Microsoft Access のターゲット表をロードして、欠落しているレコードを挿入し、既存のレコードを更新する場合は、この KM の使用を検討してください。</p> <p>この IKM を使用するには、ステージング領域がターゲットと同じデータ・サーバー上にあることが必要です。</p>

Microsoft SQL Server

ナレッジ・モジュール

このデータベースと組み合わせて機能するその他のナレッジ・モジュールについては、[第3章「汎用 SQL」](#)を参照してください。

種類	ナレッジ・モジュール	説明
統合	IKM MSSQL Incremental Update	<p>増分更新モードで Microsoft SQL Server のターゲット表にデータを統合します。この KM は、データ・フローをステージングするための一時的なステージング表を作成します。その後、ステージング表の内容をターゲット表と比較して、挿入するレコードおよびそれ以外の更新するレコードを推測します。また、CKM を起動してデータ整合性チェックを実行できます。</p> <p>挿入および更新は、最大のパフォーマンスを得るためにバルク・セットベースの処理で実行されます。そのため、この IKM は大量のデータ用に最適化されています。</p> <p>Microsoft SQL Server のターゲット表をロードして、欠落しているレコードを挿入し、既存のレコードを更新する場合は、この IKM の使用を検討してください。</p> <p>この IKM を使用するには、ステージング領域がターゲットと同じデータ・サーバー上にあることが必要です。</p>
統合	IKM MSSQL Slowly Changing Dimension	<p>データ・ウェアハウスでタイプ 2 の緩やかに変化するディメンションとして使用される Microsoft SQL Server のターゲット表に、データを統合します。この IKM は、ターゲット・データストアで設定される緩やかに変化するディメンション・メタデータに依存して、新しいバージョンとして挿入する必要があるレコード、または既存のバージョンとして更新する必要があるレコードを判別します。</p> <p>挿入および更新はバルク・セットベースの処理で実行されるため、この IKM は大量のデータ用に最適化されています。</p> <p>Microsoft SQL Server のターゲット表をタイプ 2 の緩やかに変化するディメンションとしてロードする場合は、この IKM の使用を検討してください。</p> <p>この IKM を使用するには、ステージング領域がターゲットと同じデータ・サーバー上にあり、適切な緩やかに変化するディメンション・メタデータがターゲット・データストア上で設定されていることが必要です。</p>

種類	ナレッジ・モジュール	説明
ジャーナル化	JKM MSSQL Consistent	<p>トリガーを使用して Microsoft SQL Server 表の一貫したジャーナル化を行うためのジャーナル化インフラストラクチャを作成します。</p> <p>Microsoft SQL Server での一貫性のあるチェンジ・データ・キャプチャを有効にします。</p>
ジャーナル化	JKM MSSQL Simple	<p>トリガーを使用して Microsoft SQL Server 表の単純なジャーナル化を行うためのジャーナル化インフラストラクチャを作成します。</p> <p>Microsoft SQL Server での単純なチェンジ・データ・キャプチャを有効にします。</p>
ロード	LKM File to MSSQL (BULK)	<p>固有の BULK INSERT SQL コマンドを使用して、ファイルから Microsoft SQL Server ステージング領域データベースへデータをロードします。</p> <p>この方法では固有の BULK INSERT コマンドが使用されるため、大量のデータを処理する場合は、標準の LKM File to SQL より効率的です。ただし、ロードされるファイルは、Microsoft SQL Server マシンからアクセス可能であることが必要です。</p> <p>ソースが大きいフラット・ファイルで、ステージング領域が Microsoft SQL Server データベースの場合は、この LKM の使用を検討してください。</p>
ロード	LKM MSSQL to MSSQL (BCP)	<p>固有の BCP out/BCP in コマンドを使用して、Microsoft SQL Server ソース・データベースから Microsoft SQL Server ステージング領域データベースへ、データをロードします。</p> <p>このモジュールでは、固有の BCP (バルク・コピー・プログラム) コマンドを使用して一時ファイルにデータを抽出します。データはその後、固有の BCP コマンドを再び使用して、ターゲットの Microsoft SQL Server ステージング表にロードされます。この方法は通常、大量のデータを処理する場合に標準の LKM SQL to SQL より効率的です。</p> <p>ソース表が Microsoft SQL Server インスタンスにあり、ステージング領域が異なる Microsoft SQL Server インスタンスにある場合は、この LKM の使用を検討してください。</p>
ロード	LKM MSSQL to MSSQL (LINKED SERVERS)	<p>固有のリンク・サーバー機能を使用して、Microsoft SQL Server ソース・データベースから Microsoft SQL Server ステージング領域データベースへ、データをロードします。</p> <p>このモジュールでは、固有のリンク・サーバー機能を使用して、ターゲットの Microsoft SQL Server ステージング領域からソース・データにアクセスします。BCP を使用する方法のように、間にステージング・ファイルは作成されません。このモジュールは、MSDTC のメカニズムに依存します。また、大量のデータにも適しています。</p> <p>ソース表が Microsoft SQL Server インスタンスにあり、ステージング領域が異なる Microsoft SQL Server インスタンスにある場合は、この LKM の使用を検討してください。</p>

種類	ナレッジ・モジュール	説明
ロード	LKM SQL to MSSQL	任意の汎用 SQL ソース・データベースから Microsoft SQL Server ステージング領域へデータをロードします。この LKM は、第 3 章「汎用 SQL」に記載されている標準の LKM SQL to SQL と似ていますが、Microsoft SQL Server の追加パラメータを指定できる点で異なります。
ロード	LKM SQL to MSSQL (BULK)	固有の BULK INSERT SQL コマンドを使用して、任意の汎用 SQL ソース・データベースから Microsoft SQL Server ステージング領域データベースへデータをロードします。 この LKM では、一時ファイルにソース・データがアンロードされ、Microsoft SQL Server の BULK INSERT SQL コマンドがコールされて、ステージング表が移入されます。この方法では固有の BULK INSERT が使用されるため、通常、大量のデータを処理する場合は、LKM SQL to SQL または LKM SQL to MSSQL より効率的です。 汎用データベースに大量のソース・データがあり、ステージング領域が Microsoft SQL Server データベースの場合は、この LKM の使用を検討してください。

固有の要件

Microsoft SQL Server 用の一部のナレッジ・モジュールでは、このデータベース固有の機能が使用されます。このようなナレッジ・モジュールの使用時には、次の制限が適用されます。これらのトピックの追加情報は、Microsoft SQL Server のドキュメントを参照してください。

BULK INSERT コマンドの使用方法

1. BULK INSERT コマンドでロードするファイルは、Microsoft SQL Server インスタンスのマシンからアクセスできることが必要です。サーバーのファイル・システム上に置くか、UNC (Unique Naming Convention) パスからアクセスできるようにします。
2. UNC ファイル・パスはサポートされていますが、パフォーマンスの低下につながるためお薦めしません。
3. パフォーマンスの点から、ターゲット・サーバー・マシン上に Oracle Data Integrator のエージェントをインストールすることをお薦めします。

BCP コマンドの使用方法

1. BCP ユーティリティおよび Microsoft SQL Server のクライアント・ネットワーク・ユーティリティは、Oracle Data Integrator のエージェントを実行するマシン上にインストールする必要があります。
2. トポロジで定義されるサーバー名は、これらのサーバーに使用される Microsoft SQL Server クライアントの接続文字列と一致する必要があります。
3. クライアント・ユーティリティで定義されるサーバー名には、空白を使用できません。
4. UNC ファイル・パスはサポートされていますが、パフォーマンスの低下につながるためお薦めしません。
5. ターゲットのステージング領域データベースには、select into/bulk copy オプションが必要です。
6. BCP プログラムによって生成されたファイルが空の場合は、実行が保留状態のままになります。

7. パフォーマンスの点から、ターゲット・サーバー・マシン上に Oracle Data Integrator のエージェントをインストールすることをお勧めします。

リンク・サーバーの使用法

1. Microsoft SQL Server のインスタンスに接続するためにトポロジで定義されたユーザーには、次の権限が必要です。
 - ステージング領域データベースの `db_owner` であることが必要です。
 - `db_ddladmin` ロールを割り当てられている必要があります。
 - リンク・サーバーを自動作成するために、`sysdamin` 権限が必要です。
2. MSDTC サービスは、両方の SQL Server インスタンス（ソースおよびターゲット）で開始する必要があります。このサービスの構成に役立つヒントは次のとおりです。
 - MSDTC サービスのログオン・アカウントは、ネットワーク・サービス・アカウントです（`LocalSystem` アカウントではありません）。
 - MSDTC は、ネットワーク・トランザクションに対して有効であることが必要です。
 - Windows ファイアウォールは、ネットワーク上で MSDTC サービスが許可されるように構成する必要があります。デフォルトでは、Windows ファイアウォールによって MSDTC プログラムがブロックされます。
 - Microsoft SQL Server は、MSDTC のスタートアップが完了した後に起動する必要があります。
 - 次のリンクは、MSDTC サービスの詳細な構成に役立ちます。
<http://support.microsoft.com/?kbid=816701> および
<http://support.microsoft.com/?kbid=839279>

ナレッジ・モジュール

このデータベースと組み合わせて機能するその他のナレッジ・モジュールについては、[第3章「汎用SQL」](#)を参照してください。

種類	ナレッジ・モジュール	説明
チェック	CKM Netezza	<p>Netezza 表で定義された制約と照合して、データの整合性をチェックします。無効なレコードを拒否して、動的に作成されるエラー表に格納します。フロー制御および静的制御に使用できます。</p> <p>Netezza データベースでデータの整合性をチェックする場合は、この KM の使用を検討してください。</p> <p>この CKM は Netezza 用に最適化されています。</p>
統合	IKM Netezza Control Append	<p>置換 / 追加モードで Netezza のターゲット表にデータを統合します。CKM を使用してフロー・データをチェックする必要がある場合は、CKM を起動する前に一時的なステージング表が作成されます。</p> <p>置換モードで（データ整合性チェックを行って、もしくは行わずに）Netezza のターゲット表をロードする場合は、この IKM の使用を検討してください。</p> <p>この IKM を使用するには、ステージング領域がターゲットと同じデータ・サーバー上にあることが必要です。</p>
統合	IKM Netezza Incremental Update	<p>増分更新モードで Netezza のターゲット表にデータを統合します。この IKM は、データ・フローをステージングするための一時的なステージング表を作成します。その後、ステージング表の内容をターゲット表と比較して、挿入するレコードおよびそれ以外の更新するレコードを推測します。また、CKM を起動してデータ整合性チェックを実行できます。</p> <p>挿入および更新は、最大のパフォーマンスを得るためにバルク・セットベースの処理で実行されます。そのため、この IKM は大量のデータ用に最適化されています。</p> <p>Netezza のターゲット表をロードして、欠落しているレコードを挿入し、既存のレコードを更新する場合は、この IKM の使用を検討してください。</p> <p>この IKM を使用するには、ステージング領域がターゲットと同じデータ・サーバー上にあることが必要です。</p>

種類	ナレッジ・モジュール	説明
統合	IKM Netezza To File (EXTERNAL TABLE)	<p>置換モードで Netezza のステージング領域からターゲット・ファイルにデータを統合します。この IKM では、Netezza 固有の EXTERNAL TABLE 機能が使用されるため、ステージング領域が Netezza 上にあることが必要です。</p> <p>データを変換して Netezza サーバーからターゲット・ファイルへエクスポートする場合は、この IKM の使用を検討してください。</p> <p>この IKM を使用するには、ステージング領域がターゲットと異なる必要があります。ステージング領域は、Netezza の場所に設定する必要があります。</p>
ロード	LKM File to Netezza (EXTERNAL TABLE)	<p>EXTERNAL TABLE 機能 (dataobject) を使用して、ファイルから Netezza サーバーのステージング領域データベースヘデータをロードします。</p> <p>この方法では固有の EXTERNAL TABLE コマンドが使用されるため、大量のデータを処理する場合は、標準の LKM File to SQL より効率的です。ただし、ロードされるファイルは、Netezza サーバー・マシンからアクセス可能であることが必要です。</p> <p>ソースが大きいフラット・ファイルで、ステージング領域が Netezza サーバーの場合は、この LKM の使用を検討してください。</p>
リバースエンジニアリング	RKM Netezza (JYTHON)	<p>Netezza データベースから JDBC メタデータを取得します。この RKM を使用して、Netezza JDBC メタデータを Oracle Data Integrator メタデータに変換するための独自の戦略を指定することも可能です。</p> <p>Netezza JDBC ドライバの特異性により、標準の JDBC リバースエンジニアリング・プロセスで問題が発生した場合は、この RKM の使用を検討してください。</p>

Oracle AS レガシー・アダプタ /Attunity Stream

概要

この項では、レガシー・ソースで取得された変更を Oracle Data Integrator を使用して統合するためのバージョン 10.1.3.4 以上の Oracle Application Server レガシー・アダプタおよび Attunity Stream の使用の概要と方法について説明します。

概要

Oracle Application Server レガシー・アダプタでは、CICS、VSAM、Tuxedo、IMS DB および IMS TM などのエンタープライズ・データ・ソースのログベースのチェンジ・データ・キャプチャが提供されます。取得された変更は、ステージング領域と呼ばれる領域に格納されます。このステージング領域は Oracle Data Integrator インタフェースのステージング領域とは異なります。

Attunity Stream は Attunity Integration Suite (AIS) の一部で、Oracle Application Server レガシー・アダプタと同じ機能を提供します。この項では、両製品とも Attunity Stream と呼びます。

Attunity Stream のステージング領域には、ソースから取得された変更の格納に Attunity Stream で使用される変更表が含まれます。これには、Oracle Data Integrator で読み取られた最後の位置が保持され（これは Attunity Stream のコンテキストと呼ばれ、Oracle Data Integrator のコンテキストの概念とは異なります）、次に Oracle Data Integrator からのリクエストを受け取った際ここから開始されます。変更表は Attunity Stream データソースを使用してアクセスされます。

Attunity Stream の概念は Oracle Data Integrator で次のようにマップされます。

- ポートをリスニングしている Attunity Agent (デーモン) 内の 1 つの作業領域は、1 つの ODI データ・サーバーに対応します。
- このデーモン内で、各データソース (データソースと所有者のペア) は、1 つの ODI 物理スキーマに対応します。
- 各データソースで、変更表は、Attunity テクノロジーに基づいた ODI モデルの ODI データストアとして表されます。

Attunity Stream ナレッジ・モジュール

Attunity Stream 用 Oracle Data Integrator ナレッジ・モジュールでは、JDBC インタフェースを介した Attunity ステージング領域からの統合が提供されます。次のことが可能です。

- Attunity Stream データソースから Attunity Stream データを読み取ります。
- この Attunity Stream データを Oracle Data Integrator ステージング領域へロードします。
- 読み取られた変更の一貫した使用を保証する Attunity Stream コンテキストを処理します。

Attunity KM では Attunity ステージング領域で提供されるデータを使用して、作業環境のクリーニング（一時表の削除）、Attunity Stream コンテキスト情報の決定と保存、ジャーナル化されたデータの収集表へのロードおよびロードされたデータのジャーナルからのパージが行われます。

ODI には、Attunity Stream データを処理するナレッジ・モジュール（KM）が1つ用意されています。

種類	ナレッジ・モジュール	説明
ロード	LKM Attunity to SQL	ステージング領域として使用されている SQL 準拠のデータベースに Attunity Stream データをロードします。

注意： Attunity Stream はソース・システムでの変更の取得に使用されますが、通常の JDBC ソースとして使用されます（LKM のみ使用されません）。このテクノロジーには Oracle Data Integrator ジャーナル化フレームワークが使用されません。

プラットフォームのサポート

Oracle Data Integrator Attunity Stream ナレッジ・モジュールは、次のプラットフォームとバージョンで動作保証されています。

- バージョン 10.1.3.4 以上の Oracle Application Server レガシー・アダプタ
- バージョン 5.0.1 以上の Attunity Stream

インストールおよび構成

Attunity Stream テクノロジーを使用するには、まず、Oracle Data Integrator インストールの `oracledi/driver/` ディレクトリに Attunity ドライバをインストールし、ODI を再起動する必要があります。ドライバ・ファイルには `nvjdbc2.jar`、`nvapispy2.jar`、`nvlog2.jar` が含まれます。

Attunity Stream KM の使用

Oracle Data Integrator Attunity Stream KM では、Attunity Stream で取得された変更への読取りアクセスが可能になります。

Oracle Data Integrator 統合プロジェクトで Attunity Stream データを使用するには、次の処理が必要です。

1. トポロジの定義
2. Attunity Stream モデルの作成
3. プロジェクトの設定
4. Oracle Data Integrator Attunity Stream KM を使用したインタフェースの設計

トポロジの定義

Attunity Stream KM を使用したプロジェクトに固有の Oracle Data Integrator でのトポロジの作成手順は次のとおりです。

1. トポロジ・マネージャを使用してマスター・リポジトリに接続します。
2. トポロジ・マネージャに Attunity テクノロジがない場合は、シノニム挿入更新モードを使用して Attunity テクノロジをインポートします。
3. Attunity テクノロジを使用してデータ・サーバーを作成します。このデータ・サーバーは、Attunity Stream データソースに格納されているサーバーと作業領域を表します。このデータ・サーバーのパラメータを次のように設定します。

- **JDBC ドライバ:** com.attunity.jdbc.NvDriver
- **JDBC URL:**

```
jdbc:attconnect://<host_name>:<port>/<workspace>
[;AddDefaultSchema=1] [;<parameter>=<value>]
```

JDBC URL の詳細を次の表に示します。

<host_name>	Attunity デーモンを実行しているマシンの名前。
<port>	デーモンがリスニングするポート。
<workspace>	デーモンの作業領域（デフォルトは Navigator）。
AddDefaultSchema=1	このパラメータは、データソースで本来所有者をサポートしない場合にスキーマでデフォルトの所有者名 ("public") を示すことを指定します。Oracle Data Integrator で所有者の値を使用することがあるため、これが必要です。
<parameter>=<value>	JDBC ドライバに使用できる任意のパラメータ。Oracle Data Integrator では完全修飾構文 (DATASOURCE:OWNER.TABLE_NAME) を使用して変更表にアクセスするため、DefTdpName ドライバ・パラメータを使用してデータソースを指定する必要がないことに注意してください。

JDBC URL 接続の詳細は、Attunity Stream のドキュメントを参照してください。

- **JDBC ユーザー:** 作業領域に接続するユーザーのプロファイル。匿名アクセスを使用する場合、または URL でユーザーとパスワードを指定する場合、このフィールドと「JDBC パスワード」フィールドを空のままにします。
 - **JDBC パスワード:** ユーザー・プロファイルのマスター・パスワード。
4. このデータ・サーバーに、変更されたデータを読み取る Attunity Stream データソースに対応する物理スキーマを作成します。データ・サーバーが正しく構成されている場合、物理スキーマを定義する際、作業領域で使用できるデータソースと所有者のリスト（ない場合は "public"）が表示されます。
 5. 適切なコンテキストでこの物理スキーマに対する論理スキーマを作成します。

Attunity Stream モデルの作成とリバースエンジニアリング

Attunity Stream モデルを作成するには、ODI の標準のモデル作成手順を使用します。標準リバースエンジニアリングでは、データソースに格納されている変更表がデータストアとして返されます。変更表には、統合に使用されるデータ列の他に一部の CDC ヘッダー列も含まれます。これらの列には、タイムスタンプ、表名、処理、トランザクション ID、コンテキストなどが含まれます。詳細は、Attunity Stream のドキュメントを参照してください。

プロジェクトの設定

LKM Attunity to SQL が ODI プロジェクトにまだない場合はインポートします。KM のインポートの詳細は、『Oracle Data Integrator ユーザーズ・ガイド』を参照してください。

LKM Attunity to SQL を使用したインタフェースの設計

Attunity Stream データを Oracle Data Integrator 統合プロジェクトにロードする統合インタフェースを作成するには、次のようにします。

1. Attunity Stream ソース・データストアと SQL 準拠のターゲット・データストアを使用して統合インタフェースを作成します。
2. 通常どおり結合、フィルタおよびマッピングを作成します。変更表間の結合はソースで許可されないことに注意してください。インタフェースのステージング領域で行う必要があります。
3. インタフェースの「フロー」タブでソース変更表を含むソース・セットを選択し、「**LKM Attunity to SQL**」を選択します。
4. 次のように KM のオプションを設定します。
 - **DELETE_TEMPORARY_OBJECTS**: 一時オブジェクト（ファイルとスクリプト）を統合後保持する場合、このオプションを **No** に設定します。
 - **PK_LIST**: ジャーナル化された表の主キーを含むソース列のリストを指定します。SQL 構文を使用し、「**ORDER_ID, CUSTOMER_ID**」などのように各列名は先頭に表の別名を付けず、カンマ (,) で区切ります。
5. 「**OK**」をクリックし、インタフェースを保存して閉じます。

注意： この LKM を使用してインタフェースを実行する場合、変更表からの変更が使用されます。この KM では、同じ変更を 2 回読み取ることはサポートされません。

ナレッジ・モジュール・オプションのリファレンス

LKM Attunity to SQL

オプション	値	必須	説明
DELETE_TEMPORARY_OBJECTS	Yes No	はい	このオプションでは、処理後に一時オブジェクトを削除するかどうかを指定できます。 デバッグに便利です。
PK_LIST	文字列	はい	このオプションには、主キーとして使用される列のリストが含まれます。

ナレッジ・モジュール

このデータベースと組み合わせて機能するその他のナレッジ・モジュールについては、[第3章「汎用 SQL」](#)を参照してください。

種類	ナレッジ・モジュール	説明
チェック	CKM Oracle	<p>Oracle 表で定義された制約と照合して、データの整合性をチェックします。無効なレコードを拒否して、動的に作成されるエラー表に格納します。フロー制御および静的制御に使用できます。</p> <p>Oracle データベースでデータの整合性をチェックする場合は、この CKM の使用を検討してください。</p> <p>この CKM は Oracle 用に最適化されています。</p>
統合	IKM Oracle Incremental Update	<p>増分更新モードで Oracle のターゲット表にデータを統合します。この IKM は、データ・フローをステージングするための一時的なステージング表を作成します。その後、ステージング表の内容をターゲット表と比較して、挿入するレコードおよびそれ以外の更新するレコードを推測します。また、CKM を起動してデータ整合性チェックを実行できます。</p> <p>挿入および更新は、最大のパフォーマンスを得るためにバルク・セットベースの処理で実行されます。そのため、この IKM は大量のデータ用に最適化されています。</p> <p>Oracle のターゲット表をロードして、欠落しているレコードを挿入し、既存のレコードを更新する場合は、この IKM の使用を検討してください。大量のデータを含む表にこの IKM を使用する場合は、MINUS セットベース演算子を削除してこの IKM を拡張することを検討してください。</p> <p>この IKM を使用するには、ステージング領域がターゲットと同じデータ・サーバー上にあることが必要です。</p>

種類	ナレッジ・モジュール	説明
統合	IKM Oracle Spatial Incremental Update	<p>MERGE dml 文を使用して、増分更新モードで Oracle (9i 以上) のターゲット表にデータを統合します。この IKM は、データ・フローをステージングするための一時的なステージング表を作成します。その後、ステージング表の内容をターゲット表と比較して、挿入するレコードおよびそれ以外の更新するレコードを推測します。また、CKM を起動してデータ整合性チェックを実行できます。</p> <p>存在しない行を挿入し、既存の行を更新します。</p> <p>データを管理できます。無効なデータはエラー表に分離され、リサイクルできます。</p> <p>このモジュールをジャーナリ化ソース表で使用する場合、削除を同期化できます。</p> <p>このモジュールでは、SDO_GEOMETRY データ型を処理できます。SDO_GEOMETRY データ型は、トポロジで宣言する必要があります。</p> <p>制限事項：</p> <p>ジャーナリ化データを使用する場合、「ジャーナルから削除を同期化」を実行すると、ターゲット上の削除済の行がコミットされます。</p> <p>Oracle 9i データベース以上とのみ互換性があります。データの比較には、インタフェースに定義された更新キーが使用されます。更新キーの設定が必要です。</p> <p>ターゲット表が別の表（外部キー）によって参照されている場合、TRUNCATE オプションは機能しません。</p> <p>FLOW_CONTROL および STATIC_CONTROL オプションは、無効なデータを分離するためにチェック・ナレッジ・モジュールをコールします（CKM が設定されていない場合、エラーが発生します）。統合インタフェースによって TEMPORARY ターゲット・データストアが移入される場合は、この 2 つのオプションを NO に設定する必要があります。</p> <p>COMMIT オプションには関係なく、削除をコミットします。ANALYZE_TARGET オプションは、COMMIT が Yes に設定されている場合にのみ正しい統計を評価します。そうでない場合、IKM は古いデータに基づいて統計を採取します。</p> <p>この IKM を使用するには、ステージング領域がターゲットと同じデータ・サーバー上にあることが必要です。</p>

種類	ナレッジ・モジュール	説明
統合	IKM Oracle Incremental Update (MERGE)	<p>増分更新モードで Oracle のターゲット表にデータを統合します。この IKM は、データ・フローをステージングするための一時的なステージング表を作成します。その後、Oracle MERGE 固有の SQL コマンドを使用して、ステージング表の内容をターゲット表と比較し、レコードを適宜ロードします。また、CKM を起動してデータ整合性チェックを実行できます。</p> <p>挿入および更新は、最大のパフォーマンスを得るためにバルク・セットベースの MERGE 演算子によって実行されます。そのため、この IKM は大量のデータ用に最適化されています。</p> <p>Oracle のターゲット表をロードして、欠落しているレコードを挿入し、既存のレコードを更新する場合は、この IKM の使用を検討してください。</p> <p>この IKM を使用するには、ステージング領域がターゲットと同じデータ・サーバー上にあることが必要です。</p>
統合	IKM Oracle Incremental Update (PL SQL)	<p>増分更新モードで Oracle のターゲット表にデータを統合します。この IKM は、データ・フローをステージングするための一時的なステージング表を作成します。その後、ステージング表の内容をターゲット表と比較して、挿入するレコードおよびそれ以外の更新するレコードを推測します。また、CKM を起動してデータ整合性チェックを実行できます。</p> <p>挿入および更新は、行ごとの PL/SQL 処理で行われます。そのため、この IKM を大量のデータに使用することはお薦めしません。</p> <p>Oracle のターゲット表をロードして欠落しているレコードの挿入と既存のレコードの更新を行い、ロングまたはバイナリ・ロング・オブジェクト (BLOB) のデータ型がレコードに含まれる場合は、この IKM の使用を検討してください。</p> <p>この IKM を使用するには、ステージング領域がターゲットと同じデータ・サーバー上にあることが必要です。</p>
統合	IKM Oracle Slowly Changing Dimension	<p>データ・ウェアハウスでタイプ 2 の緩やかに変化するディメンションとして使用される Oracle のターゲット表に、データを統合します。この IKM は、ターゲット・データストアで設定される緩やかに変化するディメンション・メタデータに依存して、新しいバージョンとして挿入する必要があるレコード、または既存のバージョンとして更新する必要があるレコードを判別します。</p> <p>挿入および更新はバルク・セットベースの処理で実行されるため、この IKM は大量のデータ用に最適化されています。Oracle のターゲット表をタイプ 2 の緩やかに変化するディメンションとしてロードする場合は、この IKM の使用を検討してください。</p> <p>この IKM を使用するには、ステージング領域がターゲットと同じデータ・サーバー上にあり、適切な緩やかに変化するディメンション・メタデータがターゲット・データストア上で設定されていることが必要です。</p>

種類	ナレッジ・モジュール	説明
ジャーナル化	JKM Oracle 10g Consistent (LOGMINER)	<p>Oracle 10g 表の一貫したジャーナル化を行うためのジャーナル化インフラストラクチャを作成します。変更されたデータは、Oracle 10g Log Miner 固有のユーティリティによって取得されます。</p> <p>Oracle での一貫性のあるチェンジ・データ・キャプチャを有効にします。</p>
ジャーナル化	JKM Oracle 11g Consistent (LOGMINER)	<p>Oracle 11g 表の一貫したジャーナル化を行うためのジャーナル化インフラストラクチャを作成します。変更されたデータは、Oracle 11g Log Miner 固有のユーティリティによって取得されます。</p> <p>Oracle での一貫性のあるチェンジ・データ・キャプチャを有効にします。</p>
ジャーナル化	JKM Oracle 9i Consistent (LOGMINER)	<p>Oracle 9i 表の一貫したジャーナル化を行うためのジャーナル化インフラストラクチャを作成します。変更されたデータは、Oracle 9i Log Miner 固有のユーティリティによって取得されます。</p> <p>Oracle での一貫性のあるチェンジ・データ・キャプチャを有効にします。</p>
ジャーナル化	JKM Oracle Consistent	<p>トリガーを使用して Oracle 表の一貫したジャーナル化を行うためのジャーナル化インフラストラクチャを作成します。</p> <p>Oracle での一貫性のあるチェンジ・データ・キャプチャを有効にします。</p>
ジャーナル化	JKM Oracle Simple	<p>トリガーを使用して Oracle 表の単純なジャーナル化を行うためのジャーナル化インフラストラクチャを作成します。</p> <p>Oracle での単純なチェンジ・データ・キャプチャを有効にします。</p>
ロード	LKM File to Oracle (EXTERNAL TABLE)	<p>EXTERNAL TABLE SQL コマンドを使用して、ファイルから Oracle ステージング領域へデータをロードします。</p> <p>この方法では固有の EXTERNAL TABLE コマンドが使用されるため、大量のデータを処理する場合は、標準の LKM File to SQL より効率的です。ただし、ロードされるファイルは、Oracle サーバー・マシンからアクセス可能であることが必要です。</p> <p>ソース・ファイルのデータは Oracle のステージング領域表には複製されないことに注意してください。この表はファイルのシノニムとしてのみ機能します。そのため、同じファイルがインタフェース内で他の大きい表と結合している場合は、パフォーマンスの問題が発生する可能性があります。最適化するために、ファイルの内容を Oracle ステージング領域にある実際の物理表にコピーする手順を追加することで、この LKM を拡張できます。</p> <p>ソースが大きいフラット・ファイルで、ステージング領域が Oracle データベースの場合は、この LKM の使用を検討してください。</p>

種類	ナレッジ・モジュール	説明
ロード	LKM File to Oracle (SQLLDR)	<p>固有の SQL*LOADER コマンドライン・ユーティリティを使用して、ファイルから Oracle ステージング領域へデータをロードします。</p> <p>この LKM では、ファイルの種類（固定またはデリミタ付き）に応じて、一時ディレクトリに適切な制御スクリプト（CTL）が生成されます。その後、SQLLDR オペレーティング・システム・コマンドによってこのスクリプトが実行され、実行終了時に自動的に削除されます。この方法では固有の Oracle ロードャーが使用されるため、大量のデータを処理する場合は、標準の LKM File to SQL より効率的です。</p> <p>ソースが大きいフラット・ファイルで、ステージング領域が Oracle データベースの場合は、この LKM の使用を検討してください。</p>
ロード	LKM Oracle to Oracle (DBLINK)	<p>固有のデータベース・リンク機能を使用して、Oracle ソース・データベースから Oracle ステージング領域データベースへデータをロードします。</p> <p>このモジュールでは、固有の Oracle データベース・リンクを使用して、ターゲットの Oracle ステージング領域からソース・データにアクセスします。ソース Oracle サーバーによってソースの結合および式がすべて適切に実行されるように、ソース上にソース・ステージング・スキーマでビューが作成されます。ソース・データはターゲット・ステージング領域には複製されません。Oracle のシノニムを介して単に参照されます。この LKM は大量のデータに適しています。最適化するために、リモート・シノニムの内容を Oracle ステージング領域にある実際の物理表にコピーする手順を追加することで、この LKM を拡張できます。</p> <p>ソース表が Oracle データベースにあり、ステージング領域が異なる Oracle データベースにある場合は、この LKM の使用を検討してください。</p>
ロード	LKM SQL to Oracle	<p>任意の汎用 SQL ソース・データベースから Oracle ステージング領域へデータをロードします。この LKM は、第 3 章「汎用 SQL」に記載されている標準の LKM SQL to SQL と似ていますが、Oracle の追加パラメータを指定できる点で異なります。</p>
リバースエンジニアリング	RKM Oracle	<p>表、ビュー、列、主キー、一意でない索引および外部キーについて、Oracle 固有のメタデータを取得します。この RKM は、基礎となる Oracle カタログ表にアクセスしてメタデータを取得します。この RKM はサンプルとしてのみ提供されています。</p> <p>通常は、この RKM のかわりに標準の JDBC リバースエンジニアリングを使用することを検討してください。ただし、この RKM を土台として使用し、拡張して独自のメタデータのリバースエンジニアリング動作を追加することが可能です。</p>
Web サービス	SKM Oracle	<p>Oracle データベースのデータ・アクセス Web サービスを生成します。詳細は、第 3 章「汎用 SQL」の「SKM SQL」を参照してください。</p> <p>この SKM は Oracle データベース用に最適化されています。</p>

固有の要件

Oracle 用の一部のナレッジ・モジュールでは、このデータベース固有の機能が使用されます。このようなナレッジ・モジュールの使用時には、次の制限が適用されます。これらのトピックの追加情報は、Oracle のドキュメントを参照してください。

SQL*LOADER ユーティリティの使用方法

1. Oracle Data Integrator のエージェントを実行するマシンに、Oracle クライアントおよび SQL*LOADER ユーティリティがインストールされている必要があります。
2. トポロジで定義されるサーバー名は、Oracle インスタンスへのアクセスに使用される Oracle TNS 名と一致する必要があります。
3. 特定のログ・ファイルが SQL*LOADER によって作成されます。エラーが発生した場合は、このファイルを参照することをお勧めします。制御ファイル (CTL)、ログ・ファイル (LOG)、廃棄ファイル (DSC) および不良ファイル (BAD) は、ソース・ファイルの物理スキーマで定義された作業ディレクトリにあります。
4. ダイレクト・モードを使用するには、Oracle Data Integrator のエージェントをターゲットの Oracle サーバー・マシン上で実行する必要があります。ソース・ファイルもそのマシン上にある必要があります。

外部表の使用方法

1. External Table コマンドでロードするファイルは、Oracle インスタンスのマシンからアクセスできる必要があります。サーバー・マシンのファイル・システム上に置くか、UNC (Unique Naming Convention) パスからアクセス可能にするか、もしくはローカルでマウントする必要があります。
2. パフォーマンスの点から、ターゲット・サーバー・マシン上に Oracle Data Integrator のエージェントをインストールすることをお勧めします。

Oracle Log Miner の使用方法

1. AUTO_CONFIGURATION オプションを使用すると、Oracle データベースを自動で構成し、すべての前提条件を確実に満たすことができます。このオプションでは、データベースの初期化パラメータが自動的に変更されるため、本番環境での使用はお勧めしません。Oracle Data Integrator 実行ログの「ジャーナルの作成」ステップをチェックして、正しく実行されなかった構成タスク（警告ステータス）を検出する必要があります。
2. 非同期モードでは、ジャーナル化されたシステムで最高のパフォーマンスが得られますが、追加の Oracle Database 初期化構成および追加の構成用権限が必要です。
3. 非同期モードでは、ジャーナル化されたデータベースが ARCHIVELOG に含まれている必要があります。このオプションをオンにする前に、Oracle のドキュメントで詳細を確認してください。これは、アーカイブを正しく管理し、アーカイブ・ファイルがアーカイブ・リポジトリから定期的に削除されていない場合に Oracle インスタンスが停止するなどの一般的な問題を回避するのに役立ちます。
4. ASYNCHRONOUS_MODE が No に設定されている場合、ジャーナル化されたデータはコミット直後に使用可能になります。それ以外の場合は、コミットしてからジャーナル化されたデータが使用可能になるまでに時間がかかります。この時間は場合によって 1 秒から数分です。
5. ASYNCHRONOUS_MODE オプションを変更する場合は、必ずジャーナルを停止してから行ってください。

6. 非同期モードを使用する場合は、インスタンスに接続するユーザーに、Oracle Streams の管理認可を付与する必要があります。これは、この権限をすでに持っているユーザー (SYSTEM ユーザーなど) としてログインしているときに、`DMBS_STREAMS_AUTH.GRANT_ADMIN_PRIVILEGE` プロシージャを使用して実行できます。
7. `AUTO_CONFIGURATION` オプションを使用する場合、ログイン中のユーザーは、他のユーザーに権限を付与し、インスタンス・パラメータを変更するための `DBA` ロールも割り当てられている必要があります。
8. 作業スキーマには、他のスキーマに格納されている表を参照するビューを作成できるように、`SELECT ANY TABLE` 権限を付与する必要があります。
9. その他の前提条件の詳細は、Oracle のドキュメント (『Oracle Database データ・ウェアハウス・ガイド』の「チェンジ・データ・キャプチャ」) を参照してください。

Oracle Data Quality

リバースエン 지니어リング	RKM Oracle Data Quality DDX	Oracle Data Quality DDX 記述ファイルのリバースエン 지니어リングからファイル定義を取得します。
------------------	--------------------------------	--

固有の要件

Oracle Data Quality の DQ プロセスの生成 : プロファイリングで検出されたデータ品質の問題に基づいて、データ品質の修復フローが自動的に生成されます。変更およびカスタマイズ可能です。

ランタイム・バッチ・スクリプトの生成

DQ プロセスのステップごとに (trsfmr、globtr、parser など)、1 つの OS コマンド。コードの生成時に実行プラットフォームを選択します。

Oracle E-Business Suite

概要

この項では、Oracle Data Integrator での E-Business Suite ナレッジ・モジュールの使用の概要と方法について説明します。

Oracle E-Business Suite (EBS) は、オラクル社カスタマのビジネス・ニーズに最適のソリューションを提供する統合ソフトウェア・アプリケーションのスイートです。

E-Business Suite ナレッジ・モジュール

E-Business Suite 用 Oracle Data Integrator ナレッジ・モジュールでは、Oracle Data Integrator と E-Business Suite 間の包括的で双方向の接続が提供され、データの抽出とロードが可能になります。ナレッジ・モジュールでは、E-Business Suite のすべてのモジュールがサポートされ、EBS オブジェクト表 / ビューとインタフェース表を介した双方向の接続が提供されます。EBS ナレッジ・モジュールでは、次の機能がサポートされます。

- **EBS からのデータの抽出**: 表、ビューおよびキー・フレックスフィールドなどのオブジェクトを使用した E-Business Suite からのデータの抽出に、標準 Oracle または SQL LKM を使用できます。
- **EBS オブジェクトのリバースエンジニアリング**: RKM E-Business Suite を使用して E-Business Suite データ構造をリバースエンジニアリングできます。
- **EBS へのデータ統合**: OpenInterface 表を使用した E-Business Suite へのデータの統合に IKM E-Business Suite を使用できます。OpenInterface API では、多くの Oracle 固有のインタフェースがカプセル化され、データの整合性が保たれます。OpenInterface は、次のもので構成されます。
 - ロードされるインタフェース表。これらの表は、E-Business Suite への入力データ・エントリ・ポイントです。
 - インタフェース表から E-Business Suite へのデータの挿入の検証および処理を行うプログラム。

Oracle E-Business Suite 用 Oracle Data Integrator ナレッジ・モジュールではデータベース層と対話して、メタデータの抽出とデータのロードを行います。データのロード時、アプリケーション層の同時処理サーバーとも対話します。

Oracle Data Integrator には、E-Business Suite データを処理するナレッジ・モジュール (KM) が 2 つ用意されています。

種類	ナレッジ・モジュール	説明
統合	IKM E-Business Suite (Open Interface)	<p>IKM E-Business Suite は、EBS インタフェース表へのデータのロードと同時にリクエスト（これによってインタフェース表からベース表へロードされる）の発行に使用されます。</p> <p>この統合ナレッジ・モジュールでは、次の処理が行われます。</p> <ul style="list-style-type: none"> ■ 増分更新モードで任意のソースからインタフェース表にデータを統合します。 ■ データ制御を有効にします。無効なデータはエラー表に分離され、リサイクルできます。 <p>インタフェース表のロードの他に、次のオプションのアクションが提供されます。</p> <ul style="list-style-type: none"> ■ バッチの最初のインタフェースにグループ ID を作成します。 ■ 後続のインタフェースでこのグループ ID を使用します。 ■ バッチの最後の表のロード時にこのグループ ID を削除します。 ■ バッチの任意の時点で E-Business Suite インタフェース・プログラムをコールする必要がある場合、および必要なインタフェース表がすべてロードされた後、OpenInterface プログラムが実行されます。 <p>注意: インタフェース表のロードは、IKM E-Business Suite (Open Interface) KM を使用して行います。E-Business Suite の物理表に直接書き込むことはサポートされていません。</p>
リバースエンジニアリング	RKM E-Business Suite	<p>この KM では、E-Business Suite データ構造がリバースエンジニアリングされます。表、ビュー、フレックスフィールドおよび E-Business Suite のインタフェース表構造（列、主キーおよび外部キー）などの EBS オブジェクトがリバースされます。</p>

プラットフォームのサポート

Oracle Data Integrator E-Business Suite ナレッジ・モジュールは、次のプラットフォームとバージョンで動作保証されています。

- E-Business Suite 11i
- E-Business Suite 12

インストールおよび構成

E-Business Suite KM を使用するための特別な Oracle Data Integrator 構成はありません。

EBS KM の使用

Oracle Data Integrator E-Business Suite KM では、Oracle Database に格納されている E-Business Suite データへの読取りアクセスと書き込みアクセスが可能になります。

Oracle Data Integrator E-Business Suite と Oracle Data Integrator E-Business Suite KM を組み合わせて使用するには、次のようにします。

1. トポロジの定義
2. プロジェクトの設定
3. E-Business Suite のリバースエンジニアリング
4. 統合インタフェースで E-Business Suite データストアをソースまたはターゲットとして使用

トポロジの定義

E-Business Suite KM を使用したプロジェクトに固有の Oracle Data Integrator でのトポロジの作成手順は次のとおりです。

1. トポロジ・マネージャを使用してマスター・リポジトリに接続します。
2. Oracle テクノロジーに基づきデータ・サーバーを作成します。このデータ・サーバーは、E-Business Suite データを含む Oracle インスタンスを表します。
3. このデータ・サーバーに物理スキーマを作成します。このスキーマは、E-Business Suite 表を指すシノニムを含む Oracle スキーマです。

注意： この物理スキーマは、E-Business Suite 表を指すシノニムを含む Oracle スキーマを表す必要があります。このスキーマは通常 APPS と呼ばれています。アプリケーションの物理表を含む Oracle スキーマを直接指しません。関連するアプリケーションにちなんで名前が付けられます。

注意： リバースエンジニアリングでは、物理スキーマがアタッチされているデータ・サーバーで指定された Oracle ユーザーは、APPLSYS 表から選択する権限を持っている必要があります。

4. 適切なコンテキストでこの物理スキーマに対する論理スキーマを作成します。

注意： E-Business Suite 表を含む Oracle スキーマおよびこれらの表を指すシノニムを含む Oracle スキーマは、物理スキーマの定義で作業スキーマとして定義しないでください。また、これらの Oracle スキーマはインタフェースのステージング領域として使用しないでください。

Oracle Data Integrator でのトポロジの作成の詳細は、『Oracle Data Integrator ユーザーズ・ガイド』を参照してください。

プロジェクトの設定

次の KM が Oracle Data Integrator プロジェクトにまだない場合はインポートします。

- IKM E-Business Suite (Open Interface)
- RKM E-Business Suite

この他に、Oracle データベースのデータ抽出およびデータ品質チェックを行う標準の Oracle LKM および CKM をインポートします。使用できる KM のリストは、このガイドの第 17 章「Oracle Database」を参照してください。

KM のインポートの詳細は、『Oracle Data Integrator ユーザーズ・ガイド』を参照してください。

E-Business Suite 表のリバースエンジニアリング

RKM E-Business Suite では、インストールされている E-Business Suite 表をリバースエンジニアリングし、E-Business Suite リポジトリから取得された情報を付加できます。

リバースエンジニアリング・プロセスでは、次の情報が返されます。

- インストールされている E-Business Suite (モジュール) をサブモデルとして。
- モジュールのサブモデルごとに、表、ビュー、フレックスフィールドおよびインタフェース表のサブモデルがあります。
- データストアの表、列、主キーおよび外部キー。
- リバースされた表でのコメント。

E-Business Suite のリバースエンジニアリングを行うには、次のようにします。

1. Oracle テクノロジーおよび E-Business Suite 接続の構成時に作成された論理スキーマに基づいてモデルを作成します。
2. このモデルで「リバース」タブを選択します。
 1. 「カスタマイズ済」を選択します。
 2. KM のリストから「RKM E-Business Suite」を選択します。
 3. 次のように RKM オプションを設定します（詳細は「ナレッジ・モジュール・オプションのリファレンス」を参照し、RKM E-Business Suite の制限を考慮に入れてください）。
 - **Applications List:** 'INV' などのアプリケーションの短縮名のリストを入力します。
 - **Only Installed Applications:** インストールされて共有されているアプリケーションのみをリバースエンジニアリングする場合、このオプションを YES に設定します。このオプションを NO に設定すると、すべてのアプリケーションがリバースエンジニアリングされます。
 - **Min Rows:** すべての表をリバースエンジニアリングする場合、デフォルト値の '0' のままにします。最小行数を持つ表のみをリバースエンジニアリングする場合、このオプションで最小行数を指定します。
 - **Description Mask:** リバースエンジニアリングされるオブジェクトを、E-Business Suite での摘要に基づいてフィルタ処理するための摘要マスクを指定します。
 - **Flexfields:** このオプションを YES に設定すると、アプリケーションのフレックスフィールドがリバースエンジニアリングされます。
 - **Interface Tables:** このオプションを YES に設定すると、アプリケーションのインタフェース表がリバースエンジニアリングされます。

4. リバースする表を選択するために「マスク」フィールドでリバース・マスクを指定します。「リバース」タブの「マスク」フィールドでは、リバースエンジニアリングされるオブジェクトが名前を基にフィルタ処理されます。

注意：「マスク」フィールドおよび Description Mask オプションは、SQL Like を使用して実装されています。

選択できるパターンは次のとおりです。

- ※ パーセント記号を使用すると、任意の長さ（長さ 0 を含む）の任意の文字列と一致します。

空の値は、1 文字と一致します。

3. 「適用」をクリックして「リバース」をクリックします。

実行ログでリバースエンジニアリング・プロセスを確認できます。

注意：リバースエンジニアリング・プロセスには数分かかる場合があります。実行ログのツリーのリバースされたモジュールにリバースされたデータストアが表示されるまで待ってください。

リバースエンジニアリングの最後に、アプリケーションと表がサブモデルとデータストアとして表示されます。これで Oracle Applications をインタフェースでソースまたはターゲットとして使用できます。

E-Business Suite 表のリバースエンジニアリングの機能

E-Business Suite 表のリバースエンジニアリングには、次の機能があります。

- E-Business Suite モジュールがサブモデルとしてリバースされます。サブモデル名はアプリケーション名に対応します。
- 各アプリケーション・サブモデルは、表、ビュー、フレックスフィールドおよびインタフェース表のサブモデルに分けられます。
- 表 / ビュー、列、主キーおよび外部キーがデータストアでリバースされます。
- <AppName> のフレックスフィールドと呼ばれるサブモデルがアプリケーションごとに作成されます。フレックスフィールド・サブモデルのデータストアは、アプリケーション用に登録されているキー・フレックスフィールドの Concatenated_Segment_Views に対応します。これらのオブジェクトはビューのサブセットです。フレックスフィールド・サブフォルダのデータストアは、フレックスフィールドにちなんで名前が付けられます。
- インタフェース表サブモデルのデータストアは、名前に INTERFACE パターンを含む表に対応します。これらのオブジェクトは表のサブセットです。

注意：OpenInterface (EBS 統合リポジトリで指定) には、名前に INTERFACE パターンを含まないインタフェース表を持つものがあります。

この項では、E-Business Suite 表のリバースエンジニアリングの制限について説明します。

- タブ選択リバースはこのナレッジ・モジュールに使用できません。
- Min Rows オプションを使用するには、すべての表で Oracle 統計が算出されている必要があります。
- Oracle Data Integrator データ・サーバーで定義された Oracle ユーザーが、リバースする表の所有者でない場合、リバースするすべての表でこのユーザーのシノニムを定義する必要があります。

- キー・フレックスフィールドのみがサポートされています。付加フレックスフィールドはサポートされていません。

統合インタフェースでの E-Business Suite のソースとしての使用

E-Business Suite をソースとして使用する場合、アプリケーションからデータを抽出して別のシステム（データ・ウェアハウス、他のデータベースなど）に統合します。

E-Business Suite からのデータの抽出は、Oracle Database をソースとする通常の統合インタフェースを使用して行います。このような統合インタフェースに使用できる KM のリストは、このマニュアルの第 17 章「Oracle Database」を参照してください。

統合インタフェースでの E-Business Suite のターゲットとしての使用

E-Business Suite IKM は、E-Business Suite インタフェース表をターゲットとして持つ統合インタフェースで使用されます。この IKM は IKM Oracle Incremental Update と同様に機能し、OpenInterface 用のオプションがあります。IKM Incremental Update の使用方法と制限については、このマニュアルの第 17 章「Oracle Database」を参照してください。

OpenInterface を使用した E-Business Suite へのデータの統合

Oracle Data Integrator では、OpenInterface API を使用して E-Business Suite への書込みが行われます。Oracle アプリケーションをロードするトランザクションはグループ ID で特定されるバッチです。たとえば、インタフェース表をロードして E-Business Suite で結果を出す場合、これらのすべてのロード処理およびプログラムの検証と処理へのコールには、このバッチのグループ ID が使用されます。OpenInterface の詳細は、各 E-Business Suite のモジュールの API と OpenInterface のガイドまたは E-Business Suite リポジトリを参照してください。

E-Business Suite に固有のアクション（グループ ID 処理、プログラムの実行）に対する統合インタフェースの構成は次に説明します。

グループ ID の管理

インタフェース表のグループをロードする 1 つのバッチ内の最初の統合インタフェースで、グループ ID を作成する必要があります。

統合インタフェースでグループ ID を作成するには、次のようにします。

1. KM オプションに次の設定を行います。
 - OA_CREATE_NEW_GROUP_ID を YES に設定します。
 - OA_GROUP_ID_NAME オプションにグループ ID 名を指定します。
 - **警告：**グループ ID 名は特定のインスタンスで一意である必要があります。バッチ処理の最後にグループ ID を削除する場合、OA_REMOVE_GROUP_ID オプションを使用する必要があります。
 - OA_GROUP_ID_EXPRESSION オプションにグループ ID 値に対する有効な SQL 式（インスタンスでは、Oracle Database 順序値（<SEQUENCE_NAME>.NEXTVAL）を使用）を指定します。
2. 統合インタフェース・マッピングで、グループ ID 値を使用してロードするインタフェース表のすべての列に対して UD1 フラグを選択し、マッピング値を 0 に設定します。

バッチに属する次の統合インタフェースでは、既存のグループ ID を使用する必要があります。

統合インタフェースで既存のグループ ID を使用するには、次のようにします。

1. OA_USE_EXISTING_GROUP_ID IKM オプションを Yes に設定します。
2. OA_GROUP_ID_NAME IKM オプションにグループ ID 名を指定します。
3. 統合インタフェース・マッピングで、グループ ID 値を使用してロードするすべての列に対して UD1 フラグを選択し、マッピング値を 0 に設定します。

インタフェース表のグループをロードする最後の統合インタフェースで、必要ないグループ ID を削除できます。

既存のグループ ID を削除するには、次のようにします。

1. OA_REMOVE_GROUP_ID オプションを選択します。
2. OA_GROUP_ID_NAME オプションにグループ ID 名を指定します。
3. 統合インタフェース・マッピングで、グループ ID 値を使用してロードするインタフェース表のすべての列に対して UD1 フラグを選択し、マッピング値を 0 に設定します。

注意： グループ ID は、Oracle Applications インタフェース表を指す物理スキーマで指定された作業スキーマで作成される SNP_OA_GROUP 表に格納されます。グループ ID は、Oracle Data Integrator では一意のグループ ID 名で参照されます。

OpenInterface プログラムの実行

Oracle Data Integrator 統合インタフェースでは、インタフェース表のセットをロードする場合、E-Business Suite インタフェース表でデータを検証および処理するために OpenInterface プログラムをコールする必要があります。このコールでは既存のグループ ID を使用できます（既存のグループ ID の使用に関する記述を参照）。OpenInterface に 1 つの表のみが含まれる場合、同じ統合インタフェースで作成することもできます（グループ ID の作成に関する記述を参照）。OpenInterface プログラムの実行は、パッケージの最後の統合インタフェースで起動されます。この統合インタフェースで OpenInterface 表のセットに移入が行われ、必要ない場合は通常グループ ID が削除されます。

OpenInterface プログラムを実行するには、次のようにします。

1. SUBMIT_PROGRAM オプションを YES に設定します。
2. OA_PROGRAM オプションに、コールするプログラムの名前を指定します。

注意： 使用可能な OpenInterface プログラムとパラメータのリストは、E-Business Suite のモジュールの API と OpenInterface のドキュメントまたは E-Business Suite リポジトリを参照してください。

3. OA_ARGUMENTS オプションにプログラムのパラメータを指定します。パラメータは、`argument_name => 'argument value', argument_name => 'argument value' ...` の形式で指定します。ある引数がグループ ID の値をとる場合、`argument Name => v_group_id` と指定する必要があります。
4. 次のオプションに値を設定して、プログラムを実行するセッションのコンテキスト・パラメータも指定する必要があります。
 - OA_USER_NAME: E-Business Suite ユーザー名
 - OA_REPONSIBILITY: E-Business Suite 職責名
 - OA_LANGUAGE: 職責に使用される言語
 - OA_APPLICATION: 職責が属するアプリケーション

ナレッジ・モジュール・オプションのリファレンス

RKM E-Business Suite

オプション	値	必須	説明
Applications List	文字列。デフォルトは 'INV'。	はい	APPLSYS.FND_APPLICATION 表に記述されているアプリケーションの短縮名のリスト。 これらの名前は引用符で囲み、カンマで区切る必要があります。 例: 'INV', 'PO', 'GL'
Only Installed Applications	<u>Yes</u> No	はい	インストールされているアプリケーションのみリバースします。 このオプションを選択すると、インストールされて共有されているアプリケーションのみがリバースされます。そうでない場合、すべてのアプリケーションがリバースされます。
Min Rows	デフォルトは 0	はい	最小行数を持つ表のみがリバースされます。 値 0 はすべての表を表します。このオプションを使用するには、統計が算出されている必要があります。
Description Mask	デフォルトは %	はい	フィルタ処理の説明マスク。 マスク・パターンと一致する説明のオブジェクトのみがリバースされます。
Flexfields	<u>Yes</u> No	はい	フレックスフィールドをリバースするためのフラグ。 このオプションを選択すると、アプリケーションのフレックスフィールドがリバースされます。
Interface Tables	<u>Yes</u> No	はい	インタフェース表をリバースするためのフラグ。 このオプションを選択すると、アプリケーションのインタフェース表がリバースされます。

IKM E-Business Suite (Open Interface)

オプション	値	必須	説明
INSERT	<u>Yes</u> No	はい	新規の行を挿入します。 インタフェースのターゲット・データストアへのデータの挿入を自動的に試行します。
UPDATE	<u>Yes</u> No	はい	ターゲット表を更新します。 ターゲット・データストア・レコードの UPDATE_KEY 列の値に応じてターゲット・データストアの行を特定し、更新します。
COMMIT	<u>Yes</u> No	はい	統合後コミット。 ターゲット・データストアで挿入または更新されたすべてのデータをコミットします。

オプション	値	必須	説明
SYNC_JRN_DELETE	<u>Yes</u> No	はい	ジャーナル化された削除の同期をとります。 ジャーナル化された削除の同期をとる場合、このオプションを選択します。このオプションは、インタフェースで1つのソース表がジャーナル化される場合のみ有効です。
FLOW_CONTROL	<u>Yes</u> No	はい	フロー制御をアクティブにします。 フロー制御を行う場合、このオプションを選択します。
RECYCLE_ERRORS	Yes <u>No</u>	はい	以前のエラーをリサイクルします。 以前の制御から拒否されたデータをリサイクルする場合、このオプションを選択します。
STATIC_CONTROL	Yes <u>No</u>	はい	統合後制御。 ターゲット・データの挿入または更新後にターゲット表を制御する場合、このオプションを選択します。
TRUNCATE	Yes <u>No</u>	はい	ターゲット・データストアを切り捨てる場合、このオプションを選択します。
DELETE_ALL	Yes <u>No</u>	はい	ターゲット・データストアのすべての行を削除する場合、このオプションを選択します。
DELETE_TEMPORARY_OBJECTS	<u>Yes</u> No	はい	一時オブジェクト（表、ファイルおよびスクリプト）を統合後保持する場合、このオプションをNOに設定します。注意：デバッグに便利です。
FLOW_TABLE_OPTIONS	文字列。デフォルトはNOLOGGING。	いいえ	フロー表の作成オプション。このオプションを使用して統合表の作成時およびパフォーマンスの向上に使用される属性を指定します。 このオプションはデフォルトではNOLOGGING（Oracle v8からのみ有効）に設定されています。 このオプションは空のままにできます。
COMPATIBLE	デフォルトは9	いいえ	ステージング / ターゲットのRDBMSのバージョン。このオプションは、PURGEキーワードの使用および統計の収集方法に影響を及ぼします。

値	PURGE	STATS
10	はい	DBMS_STATS
9	いいえ	DBMS_STATS
8	いいえ	ANALYZE

オプション	値	必須	説明
VALIDATE	Yes <u>No</u>	はい	<p>KM オプションを検証します。このオプションでは、開発時の追加検証処理が生成されます。行われる検証は次のとおりです。</p> <ul style="list-style-type: none"> - KM オプション COMPATIBLE の検証 - ステージング・データベースの RDBMS のバージョンの検証 - KM オプション DETECTION_STRATEGY の検証 <p>このオプションは、すべての本番使用で (実行頻度が高い場合は特に) 無効にしません。本番では付加価値はありません。処理のオーバーヘッドが増すだけです。</p>
DETECTION_STRATEGY	文字列。デフォルトは NOT_EXISTS。	はい	<p>無用な更新を特定する方法。有効な値は次のとおりです。</p> <ul style="list-style-type: none"> - MINUS: フロー表の移入時、ターゲットに同じものが存在するレコードを除外するために使用します。 - NOT_EXISTS: フロー表の移入時、ターゲットに同じものが存在するレコードを除外するために使用します。 - POST_FLOW: ソースからのすべてのレコードがフロー表にロードされます。その後、更新文が使用され、ターゲットに同じものが存在するフロー表内のすべての行にフラグが付けられます。 - NONE: ソースからのすべてのレコードがフロー表にロードされます。ターゲット・レコードがフロー表のレコードとまったく同じ場合でもすべてのターゲット・レコードが更新されます。
ANALYZE_TARGET	Yes <u>No</u>	はい	<p>統合表にデータをロードする前にターゲット表を分析する場合、このオプションを選択します。</p>
OPTIMIZER_HINT		いいえ	<p>このオプションを使用して統合表のロード時に使用されるヒント (文字列) を指定します。</p>
SUBMIT_PROGRAM	<u>Yes</u> No	はい	<p>Oracle Application OpenInterface プログラムを発行する場合、このオプションを選択します。</p>
OA_USER_NAME	文字列。デフォルトは myname。	はい	<p>コンテキスト初期化用の Oracle Application ユーザー名。</p>
OA_RESPONSIBILITY	文字列。デフォルトは ResponsibilityName。	はい	<p>コンテキストの初期化に使用する Oracle Application 職責。</p>
OA_LANGUAGE	文字列。デフォルトは US。	はい	<p>職責に使用されるデフォルト言語。</p>
OA_APPLICATION	文字列。デフォルトは INV。	はい	<p>このインタフェース用の Oracle Application 名。</p>

オプション	値	必須	説明
OA_PROGRAM	文字列。デフォルトは INCOIN。	はい	発行する E-Business Suite プログラム。
OA_ARGUMENTS	文字列。デフォルトは <code>,argument1 => 'BATCH', argument2 => v_group_ id。</code>	はい	次の構文を使用した E-Business Suite プログラム引数。 <code>, argumentX => 'value' , argumentX => 'value' , ...</code> CREATE_NEW_GROUP_ID オプションまたは USE_EXISTING_GROUP_ID オプションを使用する場合、次のように v_group_id 変数を引数として使用できます。 <code>, argumentX => 'value' , argumentX=> v_group_id , ...</code>
OA_GROUP_ID_NAME	文字列。デフォルトは myname。	はい	新規グループ ID の格納または既存のグループ ID の取得に使用するグループ ID 名。
OA_GROUP_ID_EXPRESSION	文字列。デフォルトは <code>group_id_ sequence. nextval。</code>	はい	最初のグループ ID を取得するためのグループ ID 式。 通常これは Oracle 順序です。
OA_CREATE_NEW_GROUP_ID	文字列。デフォルトは NOT_EXISTS。	はい	新規グループ ID を作成し、その値を今後使用するために格納する場合、このオプションを選択します。このオプションを YES に設定すると、OA_USE_EXISTING_GROUP_ID の値は無視されます。
OA_USE_EXISTING_GROUP_ID	Yes No	はい	名前で指定された既存のグループ ID の値を使用する場合、このオプションを選択します。(OA_GROUP_ID_NAME)
OA_REMOVE_GROUP_ID	Yes No	はい	ジョブの最後にグループ ID の定義を削除する場合、このオプションを選択します。

Oracle Enterprise Service Bus

概要

この項では、Oracle Data Integrator での ESB 相互参照ナレッジ・モジュールの使用の概要と方法について説明します。

相互参照

相互参照は、サービス指向アーキテクチャ (SOA) に真に基づいて構築された Oracle Fusion Middleware の機能で、Enterprise Service Bus (ESB) コンポーネントを介して使用できます。通常疎結合の統合で利用されます。統合に加わる各種アプリケーション間の実行時関連の管理に使用されます。

Oracle SOA Suite の相互参照機能では、異なるアプリケーションで作成された対応するエンティティの識別子を関連付けることができます。たとえば、相互参照を使用して、あるアプリケーションで作成された顧客エンティティ (ネイティブ ID: Cust_100) と別のアプリケーションの同じ顧客のエンティティ (ネイティブ ID: CT_001) を関連付けることができます。

相互参照 (XRef) 機能を使用すると、エンティティのネイティブ・キーのアプリケーション間でのマッピングが簡単になります。たとえば、異なる ERP システムの同じ注文を相互に関連付けます。

相互参照の実装では、Oracle データベース・スキーマを使用して、システムとデータストアの間でレコードを参照するための情報を格納する相互参照表 (XREF_DATA と呼ばれる) を格納します。

データのターゲット表へのロード後にソース表データを更新または削除するオプション機能も統合に必要です。これによって、ステータス・フィールドなどの一部の属性の更新またはターゲット・システムへ正常に処理された後のソース・レコードのページがバルク統合でサポートされる必要があります。

ESB 相互参照ナレッジ・モジュール

ESB 相互参照ナレッジ・モジュール (KM) は、Oracle ESB 10g との組合せで動作保証されています。Oracle Data Integrator には、ESB 相互参照を処理するナレッジ・モジュールが 3 つ用意されています。

種類	ナレッジ・モジュール	説明
ロード	LKM SQL to SQL (ESB XREF)	この KM では、標準 ISO ソースからのデータのロード時の相互参照がサポートされます。Oracle および DB2 の両方がサポートされます。LKM SQL to SQL (ESB XREF) は、同じインタフェースの IKM SQL Control Append (ESB XREF) と組み合わせて使用する必要があります。
ロード	LKM MSSQL to SQL (ESB XREF)	この KM は、Microsoft SQL Server 用に最適化されたバージョンの LKM SQL to SQL (ESB XREF) です。
統合	IKM SQL Control Append (ESB XREF)	この KM では、Oracle、DB2 または Microsoft SQL Server ターゲットへのデータの統合時の相互参照がサポートされます。切捨て / 更新 (追加) モードでデータがターゲット表に統合され、データ・チェックがサポートされます。

XREF KM プロセスの概要

全プロセスは次の 3 つの主なフェーズに分けられます。

ロード・フェーズ (LKM)

ロード・フェーズでは、ソース表からの列を使用してソース主キーが作成されます。このソース主キーは、VARCHAR 値を返すユーザー定義の SQL 式を使用して導出されます。この式は SRC_PK_EXPRESSION KM オプションで指定されます。

たとえば、ソースの注文明細表 (インタフェースでは OLINE という別名) に TO_CHAR(OLINE.ORDER_ID) || '-' || TO_CHAR(OLINE.LINE_ID) という式を使用できます。

この値を最終的に使用して相互参照表が移入されます。

統合および相互参照フェーズ (IKM)

統合フェーズでは、ターゲット表に対して共通 ID が作成されます。共通 ID の値は、XREF_SYS_GUID KM オプションの式から導出されます。この式の例は次のとおりです。

- データベース順序 (<SEQUENCE_NAME>.NEXTVAL)
- グローバル一意 ID を返す関数 (Oracle の場合は SYS_GUID(), SQL Server の場合は NewID())

この共通 ID は、UD1 フラグでマークされたターゲット表のターゲット列にプッシュされます。

共通 ID もソース主キーも相互参照表 (XREF_DATA) にプッシュされます。また、ソース主キーと共通 ID との間の相互参照を作成する一意の行番号値が、IKM で相互参照表にプッシュされます。この行番号値は、XREF_ROWNUMBER_EXPRESSION KM オプションから導出されず、このオプションでは共通 ID と同様の一意の識別子を生成するための式をとります。

相互参照 KM の複数インタフェースで同一ソース行が複数ターゲット表のロードに使用される場合、同じ共通 ID が再利用されます (再導出されません)。これによって、一意のソース行と異なるターゲット行との間の相互参照を作成できます。

処理済レコードの更新 / 削除 (LKM)

オプションのこのフェーズ (SRC_UPDATE_DELETE_ACTION KM オプションでパラメータ化) では、正常に処理されたソース・レコードを基にソース・レコードの削除または更新を行います。

- SRC_UPDATE_DELETE_ACTION が DELETE 値をとる場合、インタフェースで処理されたソース・レコードは削除されます。
- SRC_UPDATE_DELETE_ACTION が UPDATE 値をとる場合、インタフェースで処理されたソース・レコードのソース列が、SRC_UPD_EXPRESSION KM オプションで指定された SQL 式で更新されます。ソース列の名前は SRC_UPD_COL KM オプションで指定する必要があります。

インストールおよび構成

ESB 相互参照 KM を使用するための特別な Oracle Data Integrator 構成はありません。

Oracle Data Integrator ESB 相互参照 KM を使用した XREF の利用

トポロジの定義

ESB 相互参照 KM を使用したプロジェクトに固有の Oracle Data Integrator でのトポロジの作成手順は次のとおりです。

1. ソースおよびターゲットに対応するデータ・サーバー、物理スキーマおよび論理スキーマを作成します。
2. XREF_DATA という名前の相互参照表に含まれるスキーマに対し、ESB_XREF と呼ばれる Oracle データ・サーバー、物理スキーマおよび論理スキーマを作成します。すでに宣言されているデータ・サーバーにこの表が格納されている場合、スキーマの作成のみが必要です。

注意： Oracle Data Integrator でトポロジを作成する一般的な手順については、『Oracle Data Integrator ユーザーズ・ガイド』を参照してください。

プロジェクトの設定

次の KM が Oracle Data Integrator プロジェクトにまだない場合はインポートします。

- IKM SQL Control Append (ESB XREF)。
- LKM SQL to SQL (ESB XREF)。Microsoft SQL Server を使用している場合は LKM MSSQL to SQL (ESB XREF)。

Oracle Data Integrator ESB 相互参照 KM とのインタフェースの設計

複数のソース表からターゲット表へのロードとソースの 1 つとターゲットとの間の相互参照の処理の両方を行う統合インタフェースを作成するには、次のようにします。

1. 相互参照を持つソースとターゲット・データストアとのインタフェースを作成します。
2. 通常どおり結合、フィルタおよびマッピングを作成します。「ダイアグラム」タブで、共通 ID のプレースホルダになるターゲット・データストアの列に UD1 フラグを必ず選択します。この列をマップする必要はありません。
3. インタフェースの「フロー」タブで、相互参照のソース表を含むソース・セットを選択し、「LKM SQL to SQL (ESB XREF)」(ソース・データストアが Microsoft SQL Server にある場合は「LKM MSSQL to SQL (ESB XREF)」)を選択します。

4. 次のように KM のオプションを指定します。
 - SRC_PK_EXPRESSION
XREF 表に格納するソース主キー値を表す式を指定します。キーとして定義された 1 列のみがソース表に含まれる場合、列名 (SEQ_NO など) を入力します。ソース・キーに複数の列が含まれる場合、キー値の導出に使用する式を指定します。たとえば、表に 2 つのキー列があり、これらの列を連結した値をソース値として XREF 表に格納する場合、SEQ_NO|DOC_DATE と入力します。このオプションは必須です。
 - SRC_UPDATE_DELETE_ACTION
データのターゲットへの統合後ソース・レコードに対して行うアクションを示します。
 - ソース・レコードに何もしない場合は NONE と指定します。
 - SRC_UPD_COL および SRC_UPD_EXPRESSION に従ってソース・レコード・フラグを更新する場合、UPDATE と入力します。
UPDATE オプションを選択する場合、次のオプションも指定する必要があります。SRC_PK_LOGICAL_SCHEMA、SRC_PK_TABLE_NAME、SRC_PK_TABLE_ALIAS、SRC_UPD_COL および SRC_UPD_EXPRESSION
 - 統合後ソース・レコードを削除する場合、DELETE と入力します。
UPDATE オプションを選択する場合、次のオプションも指定する必要があります。SRC_PK_LOGICAL_SCHEMA、SRC_PK_TABLE_NAME および SRC_PK_TABLE_ALIAS
5. インタフェースの「フロー」タブでステージング領域を選択し、「**IKM SQL Control Append (ESB XREF)**」を選択します。
6. 次のように KM のオプションを指定します。
 - XREF_TABLE_NAME: 参照表に格納されるソース表の名前を入力します。
 - XREF_COLUMN_NAME: これは、XREF 表に格納されるソース主キーの名前です。
 - XREF_SYS_GUID_EXPRESSION: 共通 ID の導出に使用される式。式の例は次のとおりです。
データベース順序 (<SEQUENCE_NAME>.NEXTVAL)
グローバル一意 ID を返す関数 (Oracle の場合は SYS_GUID()、SQL Server の場合は NewID())
 - XREF_ROWNUMBER_EXPRESSION: これは、XREF_DATA 表の行番号列にプッシュされる値です。順序に変更する必要がない場合、デフォルト値 GUID を使用します。
 - FLOW_CONTROL: CKM Oracle を使用できるようにする場合、YES に設定します。

注意: ターゲット表に共通 ID のプレースホルダがなく、たとえば、ターゲット列の 1 つにソース識別子を移入する場合、どのソース識別子をどの列に移入するかを示すには、Oracle Data Integrator の標準マッピング・ルールを使用する必要があります。

共通 ID をロードするターゲット列がターゲット表の一意キーの場合、マップされる必要があります。その列にダミーのマッピングを設定する必要があります。実行時、このダミー・マッピングは、統合ナレッジ・モジュールで生成された共通識別子で上書きされます。このターゲット列に UD1 フラグを必ず設定します。

7. オプションでインタフェースの「制御」タブで「CKM Oracle」を選択します。
8. 「OK」をクリックし、インタフェースを保存して閉じます。

ナレッジ・モジュール・オプションのリファレンス

LKM SQL to SQL (ESB XREF)

オプション	値	必須	説明
SRC_UPDATE_DELETE_ACTION	NONE UPDATE DELETE	はい	データのターゲットへの統合後ソース・レコードに対して行うアクションを示します。 このオプションに対する有効な値は次のとおりです。 - NONE: ソース・レコードに何のアクションも行われません。 - UPDATE: SRC_UPD_COL および SRC_UPD_EXPRESSION に従ってソース・レコード・フラグが更新されます。 - DELETE: 統合後ソース・レコードが削除されます。
SRC_PK_EXPRESSION	連結式	はい	1つの大きな varchar 列に合うよう PK からの値を連結する式。 例: ソースの注文明細表 (インタフェースでは OLINE という別名) に次の式を使用できます。 TO_CHAR (OLINE.ORDER_ID) '-' TO_CHAR (OLINE.LINE_ID)
SRC_PK_LOGICAL_SCHEMA	ソース表の論理スキーマの名前	いいえ	ソース表の論理スキーマの名前を示します。ソース表は、処理後レコードを削除または更新する表です。論理スキーマは、実行時にコンテキストに従って実際の物理スキーマを解決するために使用されません。 例: ORDER_BOOKING このオプションは、SRC_UPDATE_DELETE_ACTION が UPDATE または DELETE に設定されている場合にのみ必要です。
SRC_PK_TABLE_NAME	ソース表名。デフォルトは MY_TABLE。	いいえ	処理後にレコードを削除するソース表の名前を示します。 例: ORDERS このオプションは、SRC_UPDATE_DELETE_ACTION が UPDATE または DELETE に設定されている場合にのみ必要です。
SRC_PK_TABLE_ALIAS	ソース表の別名。デフォルトは MY_ALIAS。	いいえ	このインタフェース内でのソース表の別名を示します。ソース表は、処理後レコードを削除または更新する表です。 例: ORD このオプションは、SRC_UPDATE_DELETE_ACTION が UPDATE または DELETE に設定されている場合にのみ必要です。

オプション	値	必須	説明
SRC_UPD_COL	別名が付けられたソース列名	いいえ	更新フラグ・インジケータを保持し、別名が付けられたソース列名。 SRC_UPDATE_DELETE_ACTION が式リテラル SRC_UPD_EXPRESSION で UPDATE に設定されている場合、この列の値は、統合後更新されます。列に使用される別名は、ソース表に対して定義されているものと一致する必要があります。 例: ORD.LOADED_FLAG このオプションは、SRC_UPDATE_DELETE_ACTION が UPDATE に設定されている場合にのみ必要です
SRC_UPD_EXPRESSION	リテラルまたは式	いいえ	SRC_UPD_COL の更新に使用されるリテラルまたは式。この値は、SRC_UPDATE_DELETE_ACTION が UPDATE に設定されている場合、統合後のこの列の更新に使用されます。 例: RECORDS PROCESSED このオプションは、SRC_UPDATE_DELETE_ACTION が UPDATE に設定されている場合にのみ必要です。
DELETE_TEMPORARY_OBJECTS	Yes No	はい	一時オブジェクト（ファイルおよびスクリプト）を統合後保持する場合、このオプションを NO に設定します。 デバッグに便利です。

LKM MSSQL to SQL (ESB XREF)

LKM MSSQL to SQL (ESB XREF) のオプションの詳細は、前述の「[LKM SQL to SQL \(ESB XREF\)](#)」を参照してください。

IKM SQL Control Append (ESB XREF)

オプション	値	必須	説明
INSERT	<u>Yes</u> No	はい	インタフェースのターゲット・データストアへのデータの挿入を自動的に試行します。
COMMIT	<u>Yes</u> No	はい	ターゲット・データストアに挿入されたすべてのデータをコミットします。
FLOW_CONTROL	<u>Yes</u> No	はい	フロー制御を行う場合、このオプションを選択します。
RECYCLE_ERRORS	Yes <u>No</u>	はい	以前の制御から拒否されたデータをリサイクルする場合、このオプションを選択します。
STATIC_CONTROL	Yes <u>No</u>	はい	ターゲット・データの挿入または更新後にターゲット表を制御する場合、このオプションを選択します。
TRUNCATE	Yes <u>No</u>	はい	ターゲット・データストアを切り捨てる場合、このオプションを選択します。
DELETE_ALL	Yes <u>No</u>	はい	ターゲット・データストアのすべての行を削除する場合、このオプションを選択します。
CREATE_TARG_TABLE	Yes <u>No</u>	はい	ターゲット表を作成する場合、このオプションを選択します。

オプション	値	必須	説明
DELETE_ TEMPORARY_ OBJECTS	Yes No	はい	一時オブジェクト（表、ファイルおよびスクリプト）を統合後保持する場合、このオプションを NO に設定します。デバッグに便利です。
XREF_TABLE_NAME	XREF 表名	はい	XREF 表で使用する表名。 例: ORDERS
XREF_COLUMN_ NAME	列名	はい	XREF 表でリテラルとして使用する主キー列名
XREF_LAST_ ACCESSED	ODI	はい	XREF 表で使用するリテラルのうち最後にアクセスされたもの。
XREF_SYS_GUID_ EXPRESSION	SYS_GUID()	はい	XREF 表（列名 "VALUE"）の共通 ID の移入に使用される式を入力します。 有効な値は、SYS_GUID()、 MY_SEQUENCE.NEXTVAL などです。

概要

この項では、Oracle Data Integrator での Oracle OLAP ナレッジ・モジュールの使用の概要と方法について説明します。

Oracle OLAP ナレッジ・モジュール

Oracle OLAP 用 Oracle Data Integrator ナレッジ・モジュールでは、Oracle Data Integrator と Oracle OLAP のキューブとの間の統合と接続が提供されます。Oracle Data Integrator では Oracle OLAP KM を使用し、キューブの格納モードに応じて異なる 2 種類のキューブを処理できます。

- **ROLAP** (リレーショナル・オンライン分析処理) キューブは、リレーショナル格納モデルに基づいています。
- ROLAP キューブでは大量のデータを処理でき、リレーショナル・データベースの機能をすべて利用できます。**MOLAP** (多次元オンライン分析処理) データは、多次元キューブの形式で格納されます。MOLAP モデルでは、限られた量のデータに対する問合せのパフォーマンスがよく、データの取得が高速です。

Oracle OLAP 用 Oracle Data Integrator KM では、次の処理を行うために Oracle OLAP 用の高度な統合方法を使用します。

- Oracle OLAP データ構造 (ROLAP または MOLAP のキューブで使用されているすべての表) をリバースエンジニアリングします。
- 増分更新モードで Oracle Analytic Workspace ターゲットにデータを統合します。

Oracle Data Integrator には、Oracle OLAP データを処理するナレッジ・モジュール (KM) が 2 つ用意されています。

種類	ナレッジ・モジュール	説明
リバースエンジニアリング	RKM Oracle OLAP (Jython)	<p>Oracle Database から表、ビュー、列、主キー、一意キーおよび外部キーを取得するためのリバース・エンジニアリング・ナレッジ・モジュール。ROLAP または MOLAP のキューブで使用されます。この KM には、ロギング (ログの使用およびログ・ファイル名) オプションが用意されています。次のようにオプションを設定します。</p> <ul style="list-style-type: none"> ■ MOLAP: アナリティック・ワークスペースをリバースする場合、YES に設定します。 ■ AW_NAME: アナリティック・ワークスペースの名前を示します。 ■ AW_URL: アナリティック・ワークスペースの URL を指定します。 ■ AW_OWNER: アナリティック・ワークスペースの所有者の名前を示します。 ■ AW_PASSWORD: アナリティック・ワークスペースの所有者のパスワードを示します。 ■ ROLAP: ROLAP スキーマから表をリバースする場合、YES に設定します。 ■ USE_LOG: リバースエンジニアリング・プロセスのログ詳細をログ・ファイルに書き込む場合、YES に設定します。 ■ LOG_FILE_NAME: ログ・ファイルの名前を指定します。
統合	IKM Oracle AW Incremental Update	<p>この KM は IKM Oracle Incremental Update と似ています。MOLAP キューブを処理するための追加オプションが 4 つあります。</p> <ul style="list-style-type: none"> ■ AW_NAME: アナリティック・ワークスペースの名前を示します。 ■ AW_OWNER: アナリティック・ワークスペースの所有者の名前を示します。 ■ CUBE_NAME: キューブの名前を示します。 ■ REFRESH_CUBE: アナリティック・ワークスペースのキューブをリフレッシュする場合、このオプションを YES に設定します。

プラットフォームのサポート

Oracle Data Integrator Oracle OLAP ナレッジ・モジュールは、次のプラットフォームとバージョンで動作保証されています。

- 10gR2 以上の Oracle

インストールおよび構成

RKM Oracle OLAP (Jython) では、リバースエンジニアリング・プロセス時、AWXML.jar API が使用されます。この API は Oracle で提供されます。AWXML.jar ファイルを ORACLE_HOME/olap/api/lib フォルダから oracledi/drivers フォルダにコピーし、ODI を再起動します。

注意： AWXML.jar ファイルはオラクル社の Web サイトからダウンロードすることもできます。

Oracle OLAP KM の使用

Oracle Data Integrator Oracle OLAP KM は標準 Oracle KM 同様に使用できます。Oracle OLAP KM に固有の手順を次の項で説明します。

プロジェクトの設定

次の KM が Oracle Data Integrator プロジェクトにまだない場合はインポートします。

- IKM Oracle AW Incremental Update
- RKM Oracle OLAP (Jython)

KM のインポートの詳細は、『Oracle Data Integrator ユーザーズ・ガイド』を参照してください。

OLAP キューブで使用されている Oracle 表のリバースエンジニアリング

RKM Oracle OLAP (Jython) では、Oracle OLAP キューブで使用されているトポロジで定義された物理スキーマに格納されている Oracle 表をリバースエンジニアリングできます。この RKM では、表、ビュー、列および制約があれば、これらが取得されます。

Oracle OLAP のリバースエンジニアリングを行うには、次のようにします。

1. Oracle テクノロジおよび Oracle OLAP 接続の構成時に作成された論理スキーマに基づいてモデルを作成します。
2. このモデルで「リバース」タブを選択します。
 1. 「カスタマイズ済」を選択します。
 2. KM のリストから「RKM Oracle OLAP (Jython)」を選択します。
 3. 次のように RKM のオプションを設定します。
 - **MOLAP:** アナリティック・ワークスペースをリバースする場合、YES に設定します。このオプションを YES に設定する場合、次のオプションは必須です。
 - **AW_NAME:** アナリティック・ワークスペースの名前を示します。
 - **AW_URL:** アナリティック・ワークスペースの URL を指定します。
 - **AW_OWNER:** アナリティック・ワークスペースの所有者の名前を示します。
 - **AW_PASSWORD:** アナリティック・ワークスペースの所有者のパスワードを示します。
 - **ROLAP:** ROLAP スキーマから表をリバースする場合、YES に設定します。
 - **USE_LOG:** リバースエンジニアリング・プロセスのログ詳細をログ・ファイルに書き込む場合、YES に設定します。
 - **LOG_FILE_NAME:** ログ・ファイルの名前を指定します。

- リバースする表を選択するために「マスク」フィールドでリバース・マスクを指定します。「リバース」タブの「マスク」フィールドでは、リバースエンジニアリングされるオブジェクトが名前を基にフィルタ処理されます。「マスク」フィールドは空にしないでください。少なくともパーセント記号 (%) を含める必要があります。
- 「適用」をクリックして「リバース」をクリックします。

実行ログでリバースエンジニアリング・プロセスを確認できます。

注意： リバースエンジニアリングする表の数によっては、リバースエンジニアリング・プロセスに数分以上かかる場合があります。

リバースエンジニアリング・プロセスの最後に、キューブで使用された表がデータストアとして示されます。これらのデータストアをインタフェースのソースまたはターゲットとして使用できます。

統合インタフェースでの Oracle OLAP のソースとしての使用

RKM Oracle OLAP (Jython) を使用してリバースエンジニアリングを行ったら、Oracle OLAP データ表を統合インタフェースのソースとして使用し、Oracle OLAP データベースからデータを抽出して他のシステム（データ・ウェアハウス、他のデータベースなど）に統合できます。この状況で Oracle OLAP をソースとして使用することは、統合インタフェースでソースとして Oracle データストアを使用することと同じです。汎用 SQL および Oracle Database の KM をこの用途で使用できます。

統合インタフェースでの Oracle ROLAP のターゲットとしての使用

RKM Oracle OLAP (Jython) を使用してリバースエンジニアリングを行ったら、Oracle ROLAP データ表を統合インタフェースのターゲットとして使用し、任意のシステムから Oracle ROLAP データベースにデータをロードできます。この状況で Oracle ROLAP をターゲットとして使用することは、統合インタフェースでターゲットとして Oracle データストアを使用することと同じです。汎用 SQL および Oracle Database の KM をこの用途で使用できます。

統合インタフェースでの Oracle MOLAP のターゲットとしての使用

Oracle MOLAP を統合インタフェースのターゲットとして使用することは、Oracle ROLAP をターゲットとして使用することと似ています。統合プロセスの標準機能の他に、統合インタフェースの実行時に IKM Oracle AW Incremental Update を使用して MOLAP キューブをリフレッシュできる点が異なります。

注意： 統合インタフェース処理を行うたびにキューブがリフレッシュされないようにするには、パッケージの最後の統合インタフェースでのみキューブのリフレッシュ・オプションを指定して IKM Oracle AW Incremental Update を使用します。最後の統合インタフェースで、キューブをリフレッシュするオプションを次のように設定します。REFRESH_CUBE オプションを YES に設定し、AW_OWNER、AW_NAME および CUBE_NAME オプションに値を指定します。

ナレッジ・モジュール・オプションのリファレンス

RKM Oracle OLAP (Jython)

オプション	値	必須	説明
MOLAP	<u>Yes</u> No	はい	AW から表をリバースします。
AW_NAME	文字列	MOLAP が YES に設定されている場合、はい	アナリティック・ワークスペース名。
AW_URL	<server>: 1521:<SID>	MOLAP が YES に設定されている場合、はい	アナリティック・ワークスペースの UR。 JDBC 接続が jdbc:oracle:thin:@<server>: 1521:<SID> の場合、AW_URL は <server>:1521:<SID> です。
AW_OWNER	<%=odiRef.ge tModel ("SCHEMA_ NAME")>	MOLAP が YES に設定されている場合、はい	アナリティック・ワークスペースの所有者。
AW_PASSWORD	文字列	MOLAP が YES に設定されている場合、はい	アナリティック・ワークスペースの所有者のパスワード。
ROLAP	<u>Yes</u> No	はい	ROLAP スキーマから表をリバースしません。
USE_LOG	<u>Yes</u> No	はい	ログ・ファイルを使用する場合、このオプションを Yes に設定します。
LOG_FILE_NAME	/temp/ reverse.log	USE_LOG が Yes に設定されている場合、はい	ログ・ファイル名。

IKM Oracle AW Incremental Update

オプション	値	必須	説明
INSERT	<u>Yes</u> No	はい	Yes に設定すると、インタフェースのターゲット・データストアへのデータの挿入を自動的に試行します。
UPDATE	<u>Yes</u> No	はい	Yes に設定すると、ターゲット・データストア・レコードの UPDATE_KEY 列の値に応じてターゲット・データストアの行を特定し、更新します。
COMMIT	<u>Yes</u> No	はい	Yes に設定すると、ターゲット・データストアで挿入または更新されたすべてのデータをコミットします。
SYNC_JRN_DELETE	<u>Yes</u> No	はい	Yes に設定すると、ジャーナル化された削除の同期をとります。このオプションは、インタフェースで1つのソース表がジャーナル化される場合のみ有効です。
FLOW_CONTROL	<u>Yes</u> No	はい	Yes に設定すると、フロー制御を行います。
RECYCLE_ERRORS	Yes <u>No</u>	はい	以前の制御から拒否されたデータをリサイクルする場合、このオプションを YES に設定します。

オプション	値	必須	説明												
STATIC_CONTROL	Yes <u>No</u>	はい	ターゲット・データの挿入または更新後にターゲット表を制御する場合、このオプションを YES に設定します。												
TRUNCATE	Yes <u>No</u>	はい	ターゲット・データストアを切り捨てる場合、このオプションを YES に設定します。												
DELETE_ALL	Yes <u>No</u>	はい	ターゲット・データストアのすべての行を削除する場合、このオプションを YES に設定します。												
CREATE_TARG_TABLE	Yes <u>No</u>	はい	ターゲット表を作成する場合、このオプションを YES に設定します。												
DELETE_TEMPORARY_OBJECTS	<u>Yes</u> No	はい	一時オブジェクト（表、ファイルおよびスクリプト）を統合後保持する場合、このオプションを NO に設定します。デバッグに便利です。												
FLOW_TABLE_OPTIONS	NOLOGGING	いいえ	フロー表の作成オプション。このオプションを使用して統合表の作成時およびパフォーマンスの向上に使用される属性を指定します。 このオプションはデフォルトでは NOLOGGING（Oracle v8 からのみ有効）に設定されています。 このオプションは空のままにできます。												
COMPATIBLE	9	はい	ステージング / ターゲットの RDBMS のバージョン。このオプションは、PURGE キーワードの使用および統計の収集方法に影響を及ぼします。 <table border="1" data-bbox="987 1056 1391 1232"> <thead> <tr> <th>値</th> <th>PURGE</th> <th>STATS</th> </tr> </thead> <tbody> <tr> <td>10</td> <td>はい</td> <td>DBMS_STATS</td> </tr> <tr> <td>9</td> <td>いいえ</td> <td>DBMS_STATS</td> </tr> <tr> <td>8</td> <td>いいえ</td> <td>ANALYZE</td> </tr> </tbody> </table>	値	PURGE	STATS	10	はい	DBMS_STATS	9	いいえ	DBMS_STATS	8	いいえ	ANALYZE
値	PURGE	STATS													
10	はい	DBMS_STATS													
9	いいえ	DBMS_STATS													
8	いいえ	ANALYZE													
VALIDATE	Yes <u>No</u>	はい	KM オプションを検証します。このオプションでは、開発時の追加検証処理が生成されます。行われる検証は次のとおりです。 <ul style="list-style-type: none"> ■ KM オプション COMPATIBLE の検証 ■ ステージング・データベースの RDBMS のバージョンの検証 ■ KM オプション DETECTION_STRATEGY の検証 このオプションは、すべての本番使用で（実行頻度が高い場合は特に）無効にします。本番では付加価値はありません。処理のオーバーヘッドが増すだけです。												

オプション	値	必須	説明
DETECTION_STRATEGY	NOT_EXISTS	はい	<p>無用な更新を特定する方法。有効な値は次のとおりです。</p> <ul style="list-style-type: none"> - MINUS: フロー表の移入時、ターゲットに同じものが存在するレコードを除外するために使用します。 - NOT_EXISTS: フロー表の移入時、ターゲットに同じものが存在するレコードを除外するために使用します。 - POST_FLOW: ソースからのすべてのレコードがフロー表にロードされます。その後、更新文が使用され、ターゲットに同じものが存在するフロー表内のすべての行にフラグが付けられます。 - NONE: ソースからのすべてのレコードがフロー表にロードされます。ターゲット・レコードがフロー表のレコードとまったく同じ場合でもすべてのターゲット・レコードが更新されます。
ANALYZE_TARGET	Yes <u>No</u>	はい	統合表にデータをロードする前にターゲット表を分析する場合、このオプションを選択します。
OPTIMIZER_HINT		いいえ	このオプションを使用して統合表のロード時に使用されるヒント（文字列）を指定します。
REFRESH_CUBE	<u>Yes</u> No	いいえ	<p>アナリティック・ワークスペースの MOLAP キューブをリフレッシュする場合、このオプションを Yes に設定します。このオプションを Yes に設定する場合、次のキューブに固有のオプションは必須です。</p> <p>AW_OWNER および CUBE_NAME</p>
AW_OWNER	文字列	REFRESH_CUBE が Yes に設定されている場合、はい	アナリティック・ワークスペースの所有者。
AW_NAME	文字列	REFRESH_CUBE が Yes に設定されている場合、はい	アナリティック・ワークスペース名。
CUBE_NAME	文字列	REFRESH_CUBE が Yes に設定されている場合、はい	MOLAP キューブ名。

Oracle PeopleSoft

概要

この項では、Oracle Data Integrator での PeopleSoft ナレッジ・モジュールの使用の概要と方法について説明します。

PeopleSoft ナレッジ・モジュール

PeopleSoft 用 Oracle Data Integrator ナレッジ・モジュールでは、Oracle Data Integrator と PeopleSoft プラットフォームとの間の統合と接続が提供されます。

これらの KM で PeopleSoft のデータレベルの統合が可能になります。データの抽出が PeopleSoft ビジネス・オブジェクト表で直接行われます。この方法は読取り専用です。

PeopleSoft 用 Oracle Data Integrator KM では、次の処理を行うために PeopleSoft 用の高度な統合方法を使用します。

- PeopleSoft データ構造（ビジネス・オブジェクト、表、ビュー、列、キーおよび外部キー）をリバースエンジニアリングします。
- データレベルの統合方法を使用して PeopleSoft からデータを抽出します。

Oracle Data Integrator には、PeopleSoft データを処理するナレッジ・モジュール（KM）が 2 つ用意されています。

種類	ナレッジ・モジュール	説明
リバースエンジニアリング	RKM PeopleSoft ORACLE	PeopleSoft からビジネス・オブジェクト、表、ビュー、列、キーおよび外部キーを取得するためのリバースエンジニアリング・ナレッジ・モジュール。PeopleSoft 表をホストするデータベースは Oracle です。
リバースエンジニアリング	RKM PeopleSoft MSSQL	PeopleSoft からビジネス・オブジェクト、表、ビュー、列、キーおよび外部キーを取得するためのリバースエンジニアリング・ナレッジ・モジュール。PeopleSoft 表をホストするデータベースは Oracle です。

プラットフォームのサポート

Oracle Data Integrator PeopleSoft ナレッジ・モジュールは、次のプラットフォームとバージョンで動作保証されています。

- 8.x 以上の Oracle PeopleSoft

インストールおよび構成

PeopleSoft KM を使用するための特別な Oracle Data Integrator 構成はありません。

PeopleSoft KM の使用

Oracle Data Integrator PeopleSoft KM では、PeopleSoft データへの読取り専用アクセスができます。

PeopleSoft と Oracle Data Integrator PeopleSoft KM を組み合わせて使用するには、次の処理が必要です。

1. トポロジの定義
2. プロジェクトの設定
3. PeopleSoft のリバースエンジニアリング
4. 統合インタフェースで UPeopleSoft データストアをソースとして使用

トポロジの定義

PeopleSoft KM を使用したプロジェクトに固有の Oracle Data Integrator でのトポロジの作成手順は次のとおりです。

1. トポロジ・マネージャを使用してマスター・リポジトリに接続します。
2. Oracle または Microsoft SQL Server の適切なテクノロジーを使用してデータ・サーバーを作成します。このデータ・サーバーは、PeopleSoft データを格納するデータベース・インスタンスを表します。
3. このデータ・サーバーに物理スキーマを作成します。たとえば、このスキーマは、リバースエンジニアリングする PeopleSoft 表を含む Oracle スキーマまたは Microsoft SQL Server データベースです。
4. 適切なコンテキストでこの物理スキーマに対する論理スキーマを作成します。これで PeopleSoft のリバースエンジニアリングを行うことができます。

注意： PeopleSoft 表を格納する Oracle スキーマまたは Microsoft SQL Server データベースは、物理スキーマの定義で作業スキーマとして定義しないでください。また、このスキーマまたはデータベースは、インタフェースのステージング領域として使用しないでください。

プロジェクトの設定

次の KM が Oracle Data Integrator プロジェクトにまだない場合はインポートします。

- RKM PeopleSoft ORACLE
- RKM PeopleSoft MSSQL

KM のインポートの詳細は、『Oracle Data Integrator ユーザーズ・ガイド』を参照してください。

PeopleSoft 表のリバースエンジニアリング

RKM PeopleSoft <database> では、PeopleSoft データ構造をデータレベルでリバースエンジニアリングし、PeopleSoft ディクショナリから取得された情報を付加できます。

データの抽出は、PeopleSoft ビジネス・オブジェクト表で直接行われます。このアクセス方法は読取り専用です。

リバースエンジニアリング・プロセスでは、次の情報が返されます。

- ビジネス・オブジェクトをサブモデルとして
- ビジネス・オブジェクト表を関連付けられている列と制約を持つデータストアとして
- リバースされる表と列にアタッチされているコメント

PeopleSoft のリバースエンジニアリングを行うには、次のようにします。

1. Oracle または Microsoft SQL Server のテクノロジーおよびトポロジの定義時に作成された論理スキーマに基づいてモデルを作成します。
2. このモデルで「リバース」タブを選択します。
 1. 「カスタマイズ済」を選択します。
 2. KM のリストから「RKM PeopleSoft ORACLE」または「RKM PeopleSoft MSSQL」を選択します。
 3. 次のように **BUSINESS OBJECT RKM** オプションを設定します。

CCM、DBI、DPO などのビジネス・オブジェクト・コードを入力します。

ビジネス・オブジェクト・コードは、PeopleSoft のオブジェクト・オーナー ID に対応します。異なるオブジェクト・オーナー ID が PeopleSoft ビュー EO_BCOWNRID_VW にリストされます。このフィールドは、リバースエンジニアリングするビジネス・オブジェクトをフィルタ処理するためのマスクとして使用されます。このフィールドは空にしないでください。少なくともパーセント記号 (%) を含める必要があります。

詳細は、後述の「ナレッジ・モジュール・オプションのリファレンス」を参照してください。

4. リバースする表を選択するために「マスク」フィールドでリバースエンジニアリング・マスクを指定します。「リバース」タブの「マスク」フィールドでは、リバースエンジニアリングされるオブジェクトが名前を基にフィルタ処理されます。「マスク」フィールドは空にしないでください。少なくともパーセント記号 (%) を含める必要があります。
3. 「適用」をクリックして「リバース」をクリックします。

実行ログでリバースエンジニアリング・プロセスを確認できます。

注意： リバースエンジニアリングするビジネス・オブジェクトの数によっては、リバースエンジニアリング・プロセスに数分以上かかる場合があります。

リバースエンジニアリング・プロセスの最後に、アプリケーションと表がサブモデルとデータストアとして表示されます。これで、PeopleSoft を統合インターフェースでソースとして使用できます。

統合インタフェースでの PeopleSoft のソースとしての使用

RKM PeopleSoft <database> を使用してリバースエンジニアリングを行ったら、PeopleSoft データ表をインタフェースのソースとして使用し、PeopleSoft データベースからデータを抽出して他のシステム（データ・ウェアハウス、他のデータベースなど）に統合できます。この状況で PeopleSoft をソースとして使用することは、統合インタフェースでソースとして Oracle または Microsoft SQL Server のデータストアを使用することと同じです。汎用 SQL、Oracle Database または Microsoft SQL Server の KM をこの用途で使用できます。

ナレッジ・モジュール・オプションのリファレンス

RKM PeopleSoft ORACLE および RKM PeopleSoft MSSQL

オプション	値	必須	説明
BUSINESS OBJECT	文字列。デフォルトは %。	はい	<p>ビジネス・オブジェクト名。</p> <p>リバースエンジニアリングされるビジネス・オブジェクトのフィルタ処理に使用されるマスク。 '%' ですべてのビジネス・オブジェクトが返されます。</p> <p>ビジネス・オブジェクトは、PeopleSoft のオブジェクト・オーナー ID に対応します。</p> <p>PeopleSoft ビュー EO_BCOWNRID_VW には異なるオブジェクト・オーナー ID があります。このビューにオブジェクト・オーナー ID がリストされない表もあります。これらの表は、"NA" という名前のデフォルト・ビジネス・オブジェクト（サブモデル）でリバースエンジニアリングされます。</p>

Oracle Siebel CRM

概要

この項では、Oracle Data Integrator での Oracle Siebel CRM ナレッジ・モジュールの使用の概要と方法について説明します。

Siebel ナレッジ・モジュール

Oracle Data Integrator Siebel ナレッジ・モジュール (KM) では、次の処理を行うために Siebel 用の高度な統合方法を使用します。

- Siebel データ構造 (ビジネス・コンポーネントおよびビジネス・オブジェクト) のリバースエンジニアリング
- EIM (Enterprise Integration Manager) 表のリバースエンジニアリング
- データレベル統合を使用した Siebel からのデータの読取り
- EIM 表を使用した Siebel データの読取りおよび書込み

Oracle Data Integrator には Siebel を処理する次のナレッジ・モジュールが用意されています。

種類	ナレッジ・モジュール	説明
統合	IKM SQL to Siebel Append (EIM)	ANSI-SQL92 準拠のステージング領域から Siebel EIM (Enterprise Integration Manager) 表にデータを統合し、適切な EIM 構成ファイル (.ifb) を生成します。Siebel サーバー・マネージャを使用してインポート・プロセスを実行します。 切捨て / 挿入モードでターゲット表に移入されます。
ロード	LKM Siebel to SQL (EIM)	Siebel EIM (Enterprise Integration Manager) 表から ISO-92 準拠のターゲット・データベースにデータをロードします。 このモジュールでは、ランタイム・エージェントを使用して EIM (Enterprise Integration Manager) 表からステージング領域へデータが抽出されます。 適切な EIM 構成ファイル (.ifb) を生成し、Siebel サーバー・マネージャを使用してエクスポート・プロセスを実行できます。
リバースエンジニアリング	RKM Siebel Oracle	Siebel 用リバースエンジニアリング・ナレッジ・モジュール。ビジネス・オブジェクトはサブモデルとしてリバースされ、ビジネス・コンポーネントは列と制約 (主キーおよび外部キー) を持つデータストアとしてリバースされます。

種類	ナレッジ・モジュール	説明
リバースエンジニアリング	RKM Siebel EIM Oracle	Siebel EIM (Enterprise Integration Manager) 表用リバースエンジニアリング・ナレッジ・モジュール。 Siebel プロジェクトはサブモデルとしてリバースされ、EIM 表は列と主キーを持つデータストアとしてリバースされます。
リバースエンジニアリング	RKM Siebel MSSQL	この RKM は、Microsoft SQL Server 上にインストールされた Siebel に RKM Siebel Oracle と同じ機能を提供します。
リバースエンジニアリング	RKM Siebel EIM MSSQL	この RKM は、Microsoft SQL Server 上にインストールされた Siebel に RKM Siebel EIM Oracle と同じ機能を提供します。

プラットフォームのサポート

Oracle Data Integrator Siebel ナレッジ・モジュールは、次のプラットフォームとバージョンで動作保証されています。

- 7.7 以上の Oracle Siebel CRM

Siebel からのデータ抽出の概要

Oracle Data Integrator には、Siebel からデータを抽出する方法が 2 つ用意されています。

1. **データレベルの統合**: データの抽出が Siebel ビジネス・コンポーネント表で直接行われま
す。Siebel データベースからデータを抽出して他のシステムに統合することで、Siebel
データ・モデルを統合インタフェースのソースとして使用できます。この状況で Siebel を
ソースとして使用することは、統合インタフェースでソースとして通常の表を使用するこ
とと同じです。この統合方法は読取り専用です。
 - **リバース**: Siebel ビジネス・コンポーネントをリバースエンジニアリングするには、
RKM Siebel <database> (<database> は Siebel 表をホストするデータベースの名前)
を使用します。この RKM では、Siebel データ構造をリバースエンジニアリングし、
Siebel ディクショナリから取得された情報を付加できます。
 - **抽出**: Siebel から Siebel データを抽出する各種ナレッジ・モジュールにアクセスできま
す。汎用 SQL、Oracle Database または Microsoft SQL Server の KM をこの用途で使
用できます。
2. **EIM 表を使用した統合**: Siebel からのデータの抽出および Siebel へのデータのロードに
EIM 表を使用します。EIM 表が、Siebel アプリケーションと他のアプリケーション (他の
Siebel もこれらのアプリケーションの 1 つ) との間のステージング領域として機能します。
この方法では、読取りと書込みがサポートされます。
 - **リバース**: Siebel EIM 表をリバースエンジニアリングするには、RKM Siebel EIM
<database> ナレッジ・モジュールを使用します。この RKM では、Siebel EIM 表をリ
バースエンジニアリングし、Siebel ディクショナリから取得された情報を付加できま
す。
 - **抽出**: Siebel アプリケーション表から EIM 表をロードするエクスポート・スクリプトの
自動実行後、EIM 表でデータ抽出が行われます。Siebel EIM 表からデータを抽出して
SQL ステージング領域にロードするには、LKM Siebel to SQL (EIM) を使用します。

この LKM では、まず、適切な EIM 構成ファイル (.ifb: インタフェース・ビルダー・
ファイル) が生成され、サーバー・マネージャを使用してエクスポート・プロセスが
実行されます。その後、選択したデータが EIM Siebel 表からステージング領域に抽出
されます。

- **統合:** データ統合が EIM 表で行われます。インポート・スクリプトが生成されて実行され、EIM 表から Siebel アプリケーション表へデータがインポートされます。ステージング領域からこのような統合を行うには、IKM SQL to Siebel Append (EIM) を使用します。

この IKM では、まず、ステージング領域から EIM 表にデータがロードされます。次に、EIM 構成ファイル (.ifb) が生成され、サーバー・マネージャを使用してインポート・プロセスが実行されます。

警告: Siebel への書込みは、EIM 表を使用して行います。Siebel 物理表に直接書き込むことは推奨されません。

インストールおよび構成

EIM KM を使用する場合、ランタイム・エージェントをホストするマシンに Siebel Srvmgr ユーティリティをインストールする必要があります。

Siebel KM の使用

トポロジの定義

Siebel KM を使用したプロジェクトに固有の Oracle Data Integrator でのトポロジの作成手順は次のとおりです。

1. トポロジ・マネージャを使用してマスター・リポジトリに接続します。
2. (データベースのテクノロジーに応じて) Oracle または Microsoft SQL Server のテクノロジーに基づき、Siebel データを含むインスタンスを指すデータ・サーバーを作成します。
3. このデータ・サーバーに物理スキーマを作成します。このスキーマは、リバースエンジニアリングする Siebel 表または EIM データ構造を含む Oracle スキーマまたは Microsoft SQL Server データベースを指す必要があります。
4. 適切なコンテキストでこの物理スキーマに対する論理スキーマを作成します。

トポロジ・マネージャでのこれらの要素の作成の詳細は、『Oracle Data Integrator ユーザーズ・ガイド』を参照してください。

警告: Siebel 表を含む Oracle スキーマまたは Microsoft SQL Server データベースは、物理スキーマの定義で作業スキーマとして定義しないでください。また、このスキーマまたはデータベースは、統合インタフェースのステージング領域として使用しないでください。

プロジェクトの設定

プロジェクトで使用するデータベースのテクノロジーに応じて、次の KM を Oracle Data Integrator プロジェクトにインポートします (KM がプロジェクトにない場合)。

- RKM Siebel <database> または RKM Siebel EIM <database> KM
- LKM Siebel to SQL (EIM)
- IKM SQL to Siebel Append (EIM)

Siebel のリバースエンジニアリング

Siebel のリバースエンジニアリングを行うには、次のようにします。

1. Oracle または Microsoft SQL Server のテクノロジーおよびトポロジの定義時に作成された論理スキーマに基づいてモデルを作成します。
2. このモデルで「リバース」タブを選択します。
 1. 「カスタマイズ済」を選択し、使用する統合方法に応じて RKM Siebel <database> または RKM Siebel EIM <database> KM を選択します。
 2. 次のように RKM オプションを構成します。
 - a. RKM Siebel to Oracle または RKM Siebel MSSQL:
 - **Business Object:** リバースエンジニアリングするビジネス・オブジェクトをフィルタ処理するマスクを指定します。たとえば、Account、Acc%、Customer、Employee、%mpl% などです。パーセント記号 (%) ですべてのビジネス・オブジェクトが返されます。
 - b. RKM Siebel EIM Oracle または RKM Siebel EIM MSSQL:
 - **USE_PROJECT:** Oracle Data Integrator でプロジェクトをサブモデルとしてリバースエンジニアリングする場合、このオプションを YES に設定します。
 - **REPOSITORY:** Siebel リポジトリ名を指定します。デフォルトは Siebel Repository です。
 - **PROJECT_NAME:** リバースエンジニアリングする Siebel プロジェクトをフィルタ処理するマスクを指定します。たとえば、EIM Accounts and Quotes、EIM Activity、EIM A% などです。パーセント記号 (%) ですべての Siebel プロジェクトが返されます。
3. リバースする表を選択するために「マスク」フィールドでリバースエンジニアリング・マスクを指定します。「リバース」タブの「マスク」フィールドでは、リバースエンジニアリングされるオブジェクトが名前を基にフィルタ処理されます。「マスク」フィールドは空にしないでください。少なくともパーセント記号 (%) を含める必要があります。

3. 「適用」をクリックして「リバース」をクリックします。

実行ログでリバースエンジニアリング・プロセスを確認できます。

注意： リバースエンジニアリングするビジネス・オブジェクトの数によっては、リバースエンジニアリング・プロセスに数分以上かかる場合があります。

リバースエンジニアリングの最後に、アプリケーションと表がサブモデルとデータストアとして表示されます。

リバースエンジニアリングの機能

RKM Siebel <database> プロセスでは次のものが返されます。

- インストールされているビジネス・オブジェクトをサブモデルとして
- ビジネス・コンポーネントを列と制約（主キーと外部キー）を持つデータストアとして
- リバースされた表と列でのコメント

RKM Siebel <database> EIM プロセスでは次のものが返されます。

- プロジェクトをサブモデルとして
- EIM 表を列と制約（主キーと外部キー）を持つデータストアとして

統合インタフェースでの Siebel のソースとしての使用

RKM Siebel <database> を使用してリバースエンジニアリングを行ったら、Siebel データ表を統合インタフェースのソースとして使用できます。

この状況で Siebel をソースとして使用することは、インタフェースでソースとして標準の Oracle または MSSQL のデータベースを使用することと同じです。汎用 SQL、Oracle Database または Microsoft SQL Server の KM をこの用途で使用できます。

EIM 表を介した統合インタフェースでの Siebel のソースとしての使用

EIM 表を使用して Siebel からデータを抽出するには、EIM 表をソースとして統合インタフェースを作成します。LKM Siebel to SQL (EIM) を選択し、次のように KM オプションを設定します。

- IFB_PATH: EIM 構成ファイル (.ifb) を作成するパスを指定します。
- SRVRMGR_PATH: Siebel srvmgr バイナリの場所を指定します。このパラメータは必須です。
- SIEBEL_GATEWAY: ゲートウェイ・サーバー・マシンのネットワーク・アドレスを指定します。
- SIEBEL_ENTERPRISE: エンタープライズ・サーバーの名前を示します。
- SIEBEL_SERVER: Siebel サーバーの名前を示します。
- SERVER_USER: サーバー管理者のユーザー名を示します。
- SERVER_USER_PWD: サーバー管理者のパスワードを示します。

LKM Siebel to SQL (EIM) では、次の処理が自動的に行われます。

1. EIM 表へのエクスポート・プロセスが記述された EIM 構成ファイルが生成されます。
2. Siebel srvmgr コマンドラインなどを使用して EIM プロセスが実行されます。
3. EIM 表から他のアプリケーションにデータを抽出、変換およびロードします。

EIM 表を介した統合インタフェースでの Siebel のターゲットとしての使用

EIM 表を使用して Siebel にデータを挿入するには、EIM 表をターゲットとして統合インタフェースを作成します。IKM SQL to Siebel Append (EIM) を選択し、次のように KM オプションを設定します。

- IFB_PATH: EIM 構成ファイル (.ifb) を作成するパスを指定します。
- SRVRMGR_PATH: Siebel srvmgr バイナリの場所を指定します。このパラメータは必須です。
- SIEBEL_GATEWAY: ゲートウェイ・サーバー・マシンのネットワーク・アドレスを指定します。
- SIEBEL_ENTERPRISE: エンタープライズ・サーバーの名前を示します。
- SIEBEL_SERVER: Siebel サーバーの名前を示します。
- SERVER_USER: サーバー管理者のユーザー名を示します。
- SERVER_USER_PWD: サーバー管理者のパスワードを示します。

IKM SQL to Siebel Append (EIM) では、次の処理が自動的に行われます。

1. 適切な EIM 表をロードします。
2. EIM 表からのインポート・プロセスが記述された EIM 構成ファイルが生成されます。
3. Siebel srvmgr コマンドラインなどを使用して EIM プロセスが実行されます。

ナレッジ・モジュール・オプションのリファレンス

IKM SQL to Siebel Append (EIM)

オプション	値	必須	説明
INSERT	<u>Yes</u> No	はい	統合インタフェースのターゲット・データストアへのデータの挿入を自動的に試行します。
TRUNCATE	Yes <u>No</u>	はい	ターゲット・データストアを切り捨てる場合、このオプションを選択します。
DELETE_ALL	Yes <u>No</u>	はい	ターゲット・データストアのすべての行を削除する場合、このオプションを選択します。
IFB_PATH	/temp	はい	EIM 構成ファイル (.ifb) を作成するパス。
SRVRMGR_PATH	/temp	はい	srvmgr バイナリのパス。
SIEBEL_GATEWAY	GATEWAY	はい	ゲートウェイ・サーバー・マシンのネットワーク・アドレス。
SIEBEL_ENTERPRISE	SIEBEL_ENTERPRISE	はい	エンタープライズ・サーバー名。
SIEBEL_SERVER	SERVER	はい	Siebel サーバー名。
SERVER_USER	SERVER_USER	はい	サーバー管理者のユーザー名。
SERVER_USER_PWD	SERVER_USER_PWD	はい	サーバー管理者のパスワード。

LKM Siebel to SQL (EIM)

オプション	値	必須	説明
DELETE_TEMPORARY_OBJECTS	<u>Yes</u> No	はい	一時オブジェクト (表、ファイルおよびスクリプト) を統合後保持する場合、このオプションを NO に設定します。デバッグに便利です。
IFB_PATH	/temp	はい	EIM 構成ファイル (.ifb) を作成するパス。
SRVRMGR_PATH	/temp	はい	srvmgr バイナリのパス。このパラメータは必須です。
SIEBEL_GATEWAY	GATEWAY	はい	ゲートウェイ・サーバー・マシンのネットワーク・アドレス。
SIEBEL_ENTERPRISE	SIEBEL_ENTERPRISE	はい	エンタープライズ・サーバー名。
SIEBEL_SERVER	SERVER	はい	Siebel サーバー名 (デフォルトはすべてのサーバー)。
SERVER_USER	SERVER_USER	はい	サーバー管理者のユーザー名。
SERVER_USER_PWD	SERVER_USER_PWD	はい	サーバー管理者のパスワード。

RKM Siebel Oracle / RKM Siebel MSSQL

オプション	値	必須	説明
BUSINESS_OBJECT	%	はい	ビジネス・オブジェクト名。 リバースエンジニアリングされるビジネス・オブジェクトのフィルタ処理に使用されるマスク。 %'ですべてのビジネス・オブジェクトが返されます。

RKM Siebel EIM Oracle / RKM Siebel EIM MSSQL

オプション	値	必須	説明
USE_PROJECT	<u>Yes</u> No	はい	このオプションを YES に設定すると、Oracle Data Integrator でプロジェクトがサブモデルとしてリバースエンジニアリングされます。
REPOSITORY	Siebel Repository	はい	Siebel リポジトリ名。
PROJECT_NAME	%	はい	プロジェクト名。 リバースエンジニアリングされるプロジェクトのフィルタ処理に使用されるマスク。 %'ですべてのプロジェクトが返されます。

ナレッジ・モジュール

種類	ナレッジ・モジュール	説明
統合	IKM File to Salesforce (Upsert)	挿入 / 更新モードで、ファイルから Salesforce ヘッダーデータを統合します。
統合	IKM Salesforce to File (with filter)	フィルタを使用して、Salesforce からファイルヘッダーデータを取得します。
統合	IKM Salesforce to File (without filter)	フィルタを使用せずに、Salesforce からファイルヘッダーデータを取得します。
リバースエンジニアリング	RKM Salesforce.com	Sales Force から、表、ビュー、システム表および列を取得するリバース・エンジニアリング・ナレッジ・モジュールです。

固有の要件

SalesForce 用のナレッジ・モジュールを使用するには、Sforce 6.0 Java QuickStart および AppExchange Data Loader の各ユーティリティが必要です。

これらのコンポーネントは、次の手順に従ってインストールできます。

- Sforce 6.0 Java QuickStart: このパッケージをダウンロードして、quickstart.jar ファイルを Oracle Data Integrator の /drivers ディレクトリにコピーします。
- AppExchange Data Loader: このユーティリティは、Salesforce との間のバルク・ロードの実行に使用します。Salesforce シナリオを実行するエージェントが稼働するマシンにインストールする必要があります。

これらのコンポーネントは、どちらも Salesforce.com からダウンロードできます。詳細は、それぞれのドキュメントを参照してください。

注意: Oracle Data integrator では JDBC/ODB ドライバは使用されません。使用するトポロジで JDBC ドライバおよび URL を提供する必要はありません。

これらのナレッジ・モジュールの使用時には、次の制限が適用されます。

1. リバースエンジニアリングでは、特定のバージョンの JVM を実行する必要がある Sforce 6.0 Java QuickStart コンポーネントが使用されます。必ず、適切な JVM を使用してください。
2. Salesforce のスキーマをステージング領域として使用することはできません。
3. Salesforce との間のデータ統合は、ファイルを通じて実行されます。これらのファイルは、1 行のヘッダーにフィールド名が含まれるカンマ区切りのファイル・データストアとして、ファイル・モデル内で定義する必要があります。このヘッダーは、これらのフィールドを識別するために KM によって使用されます。たとえば、[Ctrl] キーを押しながら Salesforce モデルからファイル・モデルヘッダーデータストアをドラッグ・アンド・ドロップすることで、Salesforce データストアをファイル・データストアとしてすばやくコピーできます。
4. データを Salesforce にロードするには、まず、1 つのインタフェースを使用してファイル・モデルをロードし、その後で IKM File to Salesforce (Upsert) を使用してターゲットの Salesforce システムをロードする必要があります。この IKM では、Salesforce システムへの挿入および更新が管理されます。
5. Salesforce からデータをアンロードするには、まず、IKM Salesforce to File を使用してファイル・モデルをロードし、その後でこのファイル・モデルをインタフェースのソースとして使用する必要があります。この IKM には、Salesforce でデータのフィルタ処理が要求されるかどうかに応じて、2 つのバージョンが用意されています。Salesforce で実行されるフィルタ処理は、プレーン SQL ではなく、Sforce Object Query Language (SOQL) でコード化される必要があります。

注意： Salesforce IKM では、AppExchange Data Loader コンポーネントを使用して、Salesforce との間のデータのロードが行われます。このユーティリティは、Salesforce との間のデータ統合セッションを実行するマシンにインストールする必要があります。また、KM オプションでユーティリティの場所を指定する必要があります。

ナレッジ・モジュール

種類	ナレッジ・モジュール	説明
チェック	CKM SAS	<p>SAS 表で定義された制約と照合して、データの整合性をチェックします。無効なレコードを拒否して、動的に作成されるエラー表に格納します。フロー制御および静的制御に使用できます。</p> <p>SAS 表でデータの整合性をチェックする場合は、この CKM の使用を検討してください。</p> <p>この CKM は SAS に対応しています。</p>
統合	IKM SAS Control Append	<p>置換 / 追加モードで SAS のターゲット表にデータを統合します。CKM を使用してフロー・データをチェックする必要がある場合は、CKM を起動する前に一時的なステージング表が作成されます。</p> <p>置換モードで（データ整合性チェックを行って、もしくは行わずに）SAS のターゲット表をロードする場合は、この IKM の使用を検討してください。</p> <p>この IKM を使用するには、ステージング領域がターゲットと同じデータ・サーバー上にあることが必要です。</p>
統合	IKM SAS Incremental Update	<p>増分更新モードで SAS のターゲット表にデータを統合します。この IKM は、データ・フローをステージングするための一時的なステージング表を作成します。その後、ステージング表の内容をターゲット表と比較して、挿入するレコードおよびそれ以外の更新するレコードを推測します。また、CKM を起動してデータ整合性チェックを実行できます。</p> <p>挿入および更新は、最大のパフォーマンスを得るためにバルク・セットベースの処理で実行されます。</p> <p>SAS のターゲット表をロードして、欠落しているレコードを挿入し、既存のレコードを更新する場合は、この IKM の使用を検討してください。</p> <p>この IKM を使用するには、ステージング領域がターゲットと同じデータ・サーバー上にあることが必要です。</p>
ロード	LKM File to SAS	<p>ファイルから SAS ステージング領域データベースへデータをロードします。</p> <p>ソースがフラット・ファイルで、ステージング領域が SAS サーバーの場合は、この LKM の使用を検討してください。</p>

種類	ナレッジ・モジュール	説明
ロード	LKM SAS to SQL	SAS のソースから任意の汎用 SQL ステージング領域へデータをロードします。この LKM は、第 3 章「汎用 SQL」に記載されている標準の LKM SQL to SQL と似ていますが、ソースとしての SAS 専用である点で異なります。
ロード	LKM SQL to SAS	任意の汎用 SQL ソース・データベースから SAS ステージング領域へデータをロードします。この LKM は、第 3 章「汎用 SQL」に記載されている標準の LKM SQL to SQL と似ていますが、ターゲットとしての SAS 専用である点で異なります。
リバースエンジニアリング	RKM SAS	SAS の表およびビューの構造（列のみ）を取得するリバース・エンジニアリング・ナレッジ・モジュールです。

固有の要件

SAS 用のナレッジ・モジュールは、SAS/CONNECT および SAS/BASE の各ユーティリティを利用します。したがって、これらのモジュールは、どちらも Oracle Data Integrator のエージェントを実行するマシンにインストールする必要があります。

また、これらのナレッジ・モジュールでは、SAS インタプリタを使用して SAS のコマンドが実行されます。SAS インタプリタの場所（Windows のプラットフォームでは `sas.exe` と呼ばれる）を把握し、ナレッジ・モジュールの `SAS_COMMAND KM` オプションで指定する必要があります。

SAS/CONNECT および SAS/BASE の構成の詳細は、SAS のドキュメントを参照してください。

注意： Oracle Data integrator では SAS/SHARE パッケージに含まれる JDBC/ODBC ドライバは使用されません。このコンポーネントは SAS のすべてのバージョンで使用できません。

次の各トピックは、これらの SAS 用ナレッジ・モジュールの構成に役立ちます。

SAS のユーザー名およびパスワードの取得

SAS サーバーに接続するための有効なユーザー名およびパスワードを、SAS 管理者に問い合せてください。このユーザーには、データ・ライブラリへのデータの読み取り / 書き込み、および作業ライブラリのオブジェクトの読み取り / 書き込みおよび作成に必要な権限が付与されている必要があります。通常は管理者アカウントが使用されます。また、SAS サーバー名も問い合せてください。

SAS のための Oracle Data Integrator の構成

SAS ナレッジ・モジュールの使用時には、次の制限が適用されます。

1. SAS サーバーの JDBC URL (トポロジの JDBC タブ) には、SAS サーバーへの接続に使用する SAS/CONNECT のサインオン・スクリプトのパスが含まれている必要があります。

例 : p:¥oracle¥sas¥saslink¥tcpunix.scr

2. SAS サーバーへの接続に使用する SAS/CONNECT のサインオン・スクリプトは、Oracle Data Integrator のエージェントを実行するマシンからアクセスできることが必要です。次に示すように、このファイルでは、userid およびという変数を使用して、ユーザー名およびパスワードが設定されている必要があります。これらの変数は、サーバーへの接続時にナレッジ・モジュールによって実行時に設定され、スクリプトに渡されます。

```
/*-----UNIX LOGON-----*/
/*-----*/
*input 'userid?';
type "&userid" LF;
waitfor 'Password', 30 seconds : nolog;
*input nodisplay 'Password?';
type "&password" LF;
```

3. 物理スキーマ定義のすべての接頭辞の \$ 記号を削除します。たとえば、統合表の接頭辞は、I\$_ ではなく、I_ にする必要があります。
4. SAS には主キーおよび外部キーは存在しません。必要に応じて、独自のモデルに手動で追加できます。

SAS のリモート表とローカル表

ナレッジ・モジュールによるデータの処理は、Oracle Data Integrator のエージェントの場所によって異なります。エージェントが SAS データをホストするマシンにインストールされている場合、SAS 表はローカル表とみなされます。それ以外の場合は、リモート表とみなされます。

インタフェースによって処理される SAS 表がローカルかリモートかは、REMOTE_LIB KM オプションを介して指定されます。Yes に設定されている場合、ナレッジ・モジュールはその表をリモートとみなします。

ナレッジ・モジュール

このデータベースと組み合わせて機能するその他のナレッジ・モジュールについては、[第3章「汎用 SQL」](#)を参照してください。

種類	ナレッジ・モジュール	説明
統合	IKM Sybase ASE Incremental Update	<p>増分更新モードで Sybase Adaptive Server Enterprise のターゲット表にデータを統合します。この IKM は、データ・フローをステージングするための一時的なステージング表を作成します。その後、ステージング表の内容をターゲット表と比較して、挿入するレコードおよびそれ以外の更新するレコードを推測します。また、CKM を起動してデータ整合性チェックを実行できます。</p> <p>挿入および更新は、最大のパフォーマンスを得るためにバルク・セットベースの処理で実行されます。そのため、この IKM は大量のデータ用に最適化されています。</p> <p>Sybase Adaptive Server Enterprise のターゲット表をロードして、欠落しているレコードを挿入し、既存のレコードを更新する場合は、この IKM の使用を検討してください。</p> <p>この IKM を使用するには、ステージング領域がターゲットと同じデータ・サーバー上にあることが必要です。</p>
統合	IKM Sybase ASE Slowly Changing Dimension	<p>データ・ウェアハウスでタイプ 2 の緩やかに変化するディメンションとして使用される Sybase Adaptive Server Enterprise のターゲット表に、データを統合します。この IKM は、ターゲット・データストアで設定される緩やかに変化するディメンション・メタデータに依存して、新しいバージョンとして挿入する必要があるレコード、または既存のバージョンとして更新する必要があるレコードを判別します。</p> <p>挿入および更新はバルク・セットベースの処理で実行されるため、この IKM は大量のデータ用に最適化されています。</p> <p>Sybase Adaptive Server Enterprise のターゲット表をタイプ 2 の緩やかに変化するディメンションとしてロードする場合は、この IKM の使用を検討してください。</p> <p>この IKM を使用するには、ステージング領域がターゲットと同じデータ・サーバー上にあり、適切な緩やかに変化するディメンション・メタデータがターゲット・データストア上で設定されていることが必要です。</p>

種類	ナレッジ・モジュール	説明
ジャーナル化	JKM Sybase ASE Consistent	<p>トリガーを使用して Sybase Adaptive Server Enterprise 表の一貫したジャーナル化を行うためのジャーナル化インフラストラクチャを作成します。</p> <p>Sybase Adaptive Server Enterprise での一貫性のあるチェンジ・データ・キャプチャを有効にします。</p>
ジャーナル化	JKM Sybase ASE Simple	<p>トリガーを使用して Sybase Adaptive Server Enterprise 表の単純なジャーナル化を行うためのジャーナル化インフラストラクチャを作成します。</p> <p>Sybase Adaptive Server Enterprise での単純なチェンジ・データ・キャプチャを有効にします。</p>
ロード	LKM SQL to Sybase ASE (BCP)	<p>BCP (バルク・コピー・プログラム) ユーティリティを使用して、任意の SQL 準拠データベースから Sybase Adaptive Server Enterprise のステージング領域データベースへデータをロードします。</p> <p>この LKM では、一時ファイルにソース・データがアンロードされ、Sybase BCP ユーティリティがコールされて、ステージング表が移入されます。この方法では固有の BCP ユーティリティが使用されるため、通常、大量のデータを処理する場合は、LKM SQL to SQL より効率的です。</p> <p>汎用データベースに大量のソース・データがあり、ステージング領域が Sybase Adaptive Server Enterprise データベースの場合は、この LKM の使用を検討してください。</p>
ロード	LKM Sybase ASE to Sybase ASE (BCP)	<p>固有の BCP out/BCP in コマンドを使用して、Sybase Adaptive Server Enterprise のソース・データベースから Sybase Adaptive Server Enterprise のステージング領域データベースへ、データをロードします。</p> <p>このモジュールでは、固有の BCP (バルク・コピー・プログラム) コマンドを使用して一時ファイルにデータを抽出します。データはその後、固有の BCP コマンドを再び使用して、ターゲットの Sybase Adaptive Server Enterprise ステージング表にロードされます。この方法は通常、大量のデータを処理する場合に標準の LKM SQL to SQL より効率的です。</p> <p>ソース表が Sybase Adaptive Server Enterprise インスタンスにあり、ステージング領域が異なる Sybase Adaptive Server Enterprise インスタンスにある場合は、この LKM の使用を検討してください。</p>

固有の要件

Sybase Adaptive Server Enterprise 用の一部のナレッジ・モジュールでは、BCP 固有のロード・ユーティリティが使用されます。このようなナレッジ・モジュールの使用時には、次の制限が適用されます。これらのトピックの追加情報は、Sybase Adaptive Server Enterprise のドキュメントを参照してください。

1. BCP ユーティリティおよび Sybase Adaptive Server Enterprise クライアントは、Oracle Data Integrator のエージェントを実行するマシン上にインストールする必要があります。
2. トポロジで定義されるサーバー名は、これらのサーバーに使用される Sybase Adaptive Server Enterprise クライアントの接続文字列と一致する必要があります。
3. クライアントで定義されるサーバー名には、空白を使用できません。
4. ターゲットのステージング領域データベースには、`select into/bulk copy` オプションが必要です。
5. BCP プログラムによって生成されたファイルが空の場合は、実行が保留状態のままになります。
6. パフォーマンスの点から、ターゲット・サーバー・マシン上に Oracle Data Integrator のエージェントをインストールすることをお勧めします。

ナレッジ・モジュール

このデータベースと組み合わせて機能するその他のナレッジ・モジュールについては、[第3章「汎用 SQL」](#)を参照してください。

種類	ナレッジ・モジュール	説明
チェック	CKM Sybase IQ	<p>Sybase IQ 表で定義された制約と照合して、データの整合性をチェックします。無効なレコードを拒否して、動的に作成されるエラー表に格納します。フロー制御および静的制御に使用できます。</p> <p>Sybase IQ データベースでデータの整合性をチェックする場合は、この KM の使用を検討してください。</p> <p>この CKM は Sybase IQ 用に最適化されています。</p>
統合	IKM Sybase IQ Incremental Update	<p>増分更新モードで Sybase IQ のターゲット表にデータを統合します。この IKM は、データ・フローをステージングするための一時的なステージング表を作成します。その後、ステージング表の内容をターゲット表と比較して、挿入するレコードおよびそれ以外の更新するレコードを推測します。また、CKM を起動してデータ整合性チェックを実行できます。</p> <p>挿入および更新は、最大のパフォーマンスを得るためにバルク・セットベースの処理で実行されます。そのため、この IKM は大量のデータ用に最適化されています。</p> <p>Sybase IQ のターゲット表をロードして、欠落しているレコードを挿入し、既存のレコードを更新する場合は、この IKM の使用を検討してください。</p> <p>この IKM を使用するには、ステージング領域がターゲットと同じデータ・サーバー上にあることが必要です。</p>

種類	ナレッジ・モジュール	説明
統合	IKM Sybase IQ Slowly Changing Dimension	<p>データ・ウェアハウスでタイプ 2 の緩やかに変化するディメンションとして使用される Sybase IQ のターゲット表に、データを統合します。この IKM は、ターゲット・データストアで設定される緩やかに変化するディメンション・メタデータに依存して、新しいバージョンとして挿入する必要があるレコード、または既存のバージョンとして更新する必要があるレコードを判別します。</p> <p>挿入および更新はバルク・セットベースの処理で実行されるため、この IKM は大量のデータ用に最適化されています。</p> <p>Sybase IQ のターゲット表をタイプ 2 の緩やかに変化するディメンションとしてロードする場合は、この IKM の使用を検討してください。</p> <p>この IKM を使用するには、ステージング領域がターゲットと同じデータ・サーバー上にあり、適切な緩やかに変化するディメンション・メタデータがターゲット・データストア上で設定されていることが必要です。</p>
ロード	LKM File to Sybase IQ (LOAD TABLE)	<p>LOAD TABLE SQL コマンドを使用して、ファイルから Sybase IQ ステージング領域データベースヘデータをロードします。</p> <p>この方法では固有の LOAD TABLE コマンドが使用されるため、大量のデータを処理する場合は、標準の LKM File to SQL より効率的です。ただし、ロードされるファイルは、Sybase IQ マシンからアクセス可能であることが必要です。</p> <p>ソースが大きいフラット・ファイルで、ステージング領域が Sybase IQ データベースの場合は、この LKM の使用を検討してください。</p>
ロード	LKM SQL to Sybase IQ (LOAD TABLE)	<p>固有の LOAD TABLE SQL コマンドを使用して、任意の汎用 SQL ソース・データベースから Sybase IQ ステージング領域データベースヘデータをロードします。</p> <p>この LKM では、一時ファイルにソース・データがアンロードされ、Sybase IQ の LOAD TABLE SQL コマンドがコールされて、ステージング表が移入されます。この方法では固有の LOAD TABLE ユーティリティが使用されるため、通常、大量のデータを処理する場合は、LKM SQL to SQL より効率的です。</p> <p>汎用データベースに大量のソース・データがあり、ステージング領域が Sybase IQ データベースの場合は、この LKM の使用を検討してください。</p>

固有の要件

Sybase IQ 用の一部のナレッジ・モジュールでは、固有の LOAD TABLE コマンドが使用されません。このようなナレッジ・モジュールの使用時には、次の制限が適用されます。これらのトピックの追加情報は、Sybase IQ のドキュメントを参照してください。

1. LOAD TABLE コマンドでロードされるファイルは、Sybase IQ マシンからアクセスできる必要があります。サーバーのファイル・システム上に置くか、UNC (Unique Naming Convention) パスからアクセス可能にするか、もしくはリモート・ファイル・システムからマウントする必要があります。
2. UNC ファイル・パスはサポートされていますが、パフォーマンスの低下につながるためお薦めしません。
3. パフォーマンスの点から、ターゲット・サーバー・マシン上に Oracle Data Integrator のエージェントをインストールすることをお薦めします。

ナレッジ・モジュール

このデータベースと組み合わせて機能するその他のナレッジ・モジュールについては、[第3章「汎用 SQL」](#)を参照してください。Oracle Data Integrator KM には、Teradata 専用の最適化が含まれます。これらの最適化および構成は、「[Teradata 用の KM の最適化](#)」で説明されています。

種類	ナレッジ・モジュール	説明
チェック	CKM Teradata	<p>Teradata 表で定義された制約と照合して、データの整合性をチェックします。無効なレコードを拒否して、動的に作成されるエラー表に格納します。フロー制御および静的制御に使用できます。</p> <p>Teradata データベースでデータの整合性をチェックする場合は、この KM の使用を検討してください。</p> <p>この KM では、次の Teradata の最適化がサポートされます。</p> <ul style="list-style-type: none"> ■ 1 次索引および統計
統合	IKM File to Teradata (TTU)	<p>この IKM は、Teradata ユーティリティの能力を利用してファイルを直接ターゲットにロードするように設計されています。制限により、ソースとして1つのファイル、およびターゲットとして1つの Teradata 表のみを使用できます。</p> <p>選択するユーティリティによって、置換モードまたは増分更新モードでデータを統合できます。</p> <p>1つのフラット・ファイルをターゲット表にロードする場合は、この IKM の使用を検討してください。この IKM では Teradata ユーティリティが使用されるため、大量のデータの処理に適しています。</p> <p>この IKM を使用するには、ステージング領域をソース・ファイルのスキーマに設定する必要があります。</p> <p>この KM では、次の Teradata の最適化がサポートされます。</p> <ul style="list-style-type: none"> ■ 1 次索引および統計 ■ Teradata ユーティリティのサポート ■ 最適化された一時表管理

種類	ナレッジ・モジュール	説明
統合	IKM SQL to Teradata (TTU)	<p>Teradata ユーティリティを使用して、SQL 準拠のデータベースから Teradata データベースのターゲット表へ、データを統合します。</p> <p>この IKM は、Teradata ユーティリティの能力を利用してソース・データを直接ターゲットにロードするように設計されています。この IKM を使用できるのは、すべてのソース表が同じデータ・サーバーに属し、このデータ・サーバーがステージング領域（ソース上のステージング領域）として使用される場合にかぎられません。ソース・データはファイルまたは名前付きパイプにアンロードされ、その後、選択された Teradata ユーティリティによってターゲット表に直接ロードされます。名前付きパイプを使用すると、データをファイルに書き込む必要がありません。この IKM は、非常に大量の場合にお勧めします。</p> <p>選択するユーティリティによって、置換モードまたは増分更新モードでデータを統合できます。</p> <p>次のような場合は、この IKM の使用を検討してください。</p> <ul style="list-style-type: none"> ■ ソースで数回の変換を実行してターゲットをロードする。 ■ すべてのソース表が（ステージング領域として使用される）同じデータ・サーバー上にある。 ■ ソースとターゲットの間でデータをステージングしたくない。 <p>この IKM を使用するには、ステージング領域をソース・データ・サーバーのスキーマに設定する必要があります。</p> <p>この KM では、次の Teradata の最適化がサポートされます。</p> <ul style="list-style-type: none"> ■ 1 次索引および統計 ■ Teradata ユーティリティのサポート ■ 名最適化された一時表管理前付きパイプのサポート
統合	IKM Teradata Control Append	<p>置換 / 追加モードで Teradata のターゲット表にデータを統合します。CKM を使用してフロー・データをチェックする必要がある場合は、CKM を起動する前に一時的なステージング表が作成されます。</p> <p>追加モードで（データ整合性チェックを行って、もしくは行わずに）Teradata のターゲット表をロードする場合は、この IKM の使用を検討してください。</p> <p>この IKM を使用するには、ステージング領域がターゲットの Teradata 表と同じデータ・サーバー上にあることが必要です。</p> <p>この KM では、次の Teradata の最適化がサポートされます。</p> <ul style="list-style-type: none"> ■ 1 次索引および統計 ■ 最適化された一時表管理

種類	ナレッジ・モジュール	説明
統合	IKM Teradata Incremental Update	<p>増分更新モードで Teradata のターゲット表にデータを統合します。この IKM は、データ・フローをステージングするための一時的なステージング表を作成します。その後、内容をターゲット表と比較して、挿入するレコードおよびそれ以外の更新するレコードを導出します。また、CKM を起動してデータ整合性チェックを実行できます。</p> <p>挿入および更新は、最大のパフォーマンスを得るためにバルク・セットベースの処理で実行されます。そのため、この IKM は大量のデータ用に最適化されています。</p> <p>Teradata のターゲット表をロードして、欠落しているレコードを挿入し、既存のレコードを更新する場合は、この IKM の使用を検討してください。</p> <p>この IKM を使用するには、ステージング領域がターゲットと同じデータ・サーバー上にあることが必要です。</p> <p>この KM では、次の Teradata の最適化がサポートされます。</p> <ul style="list-style-type: none"> ■ 1 次索引および統計 ■ 最適化された一時表管理
統合	IKM Teradata Slowly Changing Dimension	<p>データ・ウェアハウスでタイプ 2 の緩やかに変化するディメンションとして使用される Teradata のターゲット表に、データを統合します。この IKM は、ターゲット・データストアで設定される緩やかに変化するディメンション・メタデータに依存して、新しいバージョンとして挿入する必要があるレコード、または既存のバージョンとして更新する必要があるレコードを判別します。</p> <p>挿入および更新はバルク・セットベースの処理で実行されるため、この IKM は大量のデータ用に最適化されています。</p> <p>Teradata のターゲット表をタイプ 2 の緩やかに変化するディメンションとしてロードする場合は、この IKM の使用を検討してください。</p> <p>この IKM を使用するには、ステージング領域がターゲットと同じデータ・サーバー上にあり、適切な緩やかに変化するディメンション・メタデータがターゲット・データストア上で設定されていることが必要です。</p> <p>この KM では、次の Teradata の最適化がサポートされます。</p> <ul style="list-style-type: none"> ■ 1 次索引および統計 ■ 最適化された一時表管理 <p>この KM には、COMPATIBLE オプションも含まれません。このオプションは、Teradata エンジンのメジャー・バージョン番号に対応します。このバージョンが 12 以上の場合、標準の INSERT 文と UPDATE 文ではなく MERGE 文を使用して入力データ・フローがターゲット表にマージされます。</p>

種類	ナレッジ・モジュール	説明
統合	IKM Teradata to File (TTU)	<p>置換モードで Teradata のステージング領域からターゲット・ファイルにデータを統合します。この IKM では、ステージング領域が Teradata にあることが必要です。Teradata のネイティブのユーティリティを使用してデータをターゲット・ファイルにエクスポートします。</p> <p>データを変換して Teradata サーバーからターゲット・ファイルへエクスポートする場合は、この IKM の使用を検討してください。</p> <p>この IKM を使用するには、ステージング領域がターゲットと異なる必要があります。ステージング領域は、Teradata の場所に設定する必要があります。</p> <p>この KM では、次の Teradata の最適化がサポートされます。</p> <ul style="list-style-type: none"> ■ Teradata ユーティリティのサポート
ロード	LKM File to Teradata (TTU)	<p>Teradata のバルク・ユーティリティを使用して、ファイルから Teradata ステージング領域データベースへデータをロードします。</p> <p>この方法では、固有の Teradata ユーティリティを使用してステージング領域にファイルをロードするため、大量のデータを処理する場合は、標準の LKM File to SQL より効率的です。</p> <p>ソースが大きいフラット・ファイルで、ステージング領域が Teradata データベースの場合は、この LKM の使用を検討してください。</p> <p>この KM では、次の Teradata の最適化がサポートされます。</p> <ul style="list-style-type: none"> ■ 統計 ■ 最適化された一時表管理
ロード	LKM SQL to Teradata (TTU)	<p>Teradata のネイティブ・バルク・ユーティリティを使用して、SQL 準拠のソース・データベースから Teradata ステージング領域データベースへデータをロードします。</p> <p>この LKM では、ソース・データがファイルまたは名前付きパイプにアンロードされ、指定された Teradata ユーティリティがコールされてこのファイルまたはパイプからステージング表に移入されます。名前付きパイプを使用すると、データをファイルに書き込む必要がありません。この LKM は、非常に大量の場合にお薦めします。</p> <p>次のような場合は、この IKM の使用を検討してください。</p> <ul style="list-style-type: none"> ■ SQL 準拠のデータベースのソース・データが大量にある。 ■ ソースとターゲットの間でデータをステージングしたくない。 ■ ステージング領域が Teradata データベースである。 <p>この KM では、次の Teradata の最適化がサポートされます。</p> <ul style="list-style-type: none"> ■ Teradata ユーティリティのサポート ■ 名前付きパイプのサポート ■ 最適化された一時表管理

種類	ナレッジ・モジュール	説明
リバースエンジニアリング	RKM Teradata	<p>DBC システム・ビューを使用して、Teradata データベースからメタデータを取得します。RKM では、UNICODE の列がサポートされます。</p> <p>メタデータには、表、ビュー、列、キー（1 次索引および 2 次索引）および外部キーが含まれます。記述情報（列タイトルおよび簡単な説明）もリバースエンジニアリングされます。</p> <p>一意索引は次のようにリバースされることに注意してください。</p> <ul style="list-style-type: none"> ■ 一意の 1 次索引は主キーとみなされます。 ■ 1 次索引は一意でない索引とみなされます。 ■ 一意の 2 次索引は、代替キーとみなされます。 ■ 一意でない 2 次索引は、一意でない索引とみなされます。 <p>標準の JDBC インタフェースでサポートされていない特定の Teradata メタデータ（1 次索引など）を取得するには、この RKM を使用できます。</p>

固有の要件

これらのナレッジ・モジュールには、次の要件および制限が適用されます。

1. KM に対応する Teradata ユーティリティを使用する場合、適切な Teradata ユーティリティが Oracle Data Integrator エージェントを実行するマシンにインストールされている必要があります。
2. トポロジで定義される Teradata サーバーのサーバー名は、このサーバーに使用される Teradata 接続文字列（COP_n postfix を除く）と一致する必要があります。
3. エージェントは、ターゲットの Teradata ホストとは別のプラットフォームにインストールすることをお勧めします。エージェントがインストールされるマシンには、ターゲット Teradata サーバーへの非常に高いネットワーク・バンド幅が必要です。

Teradata 用の KM の最適化

Oracle Data Integrator KM には、Teradata 専用の最適化が含まれます。これらの最適化について次に説明します。

1 次索引および統計

Teradata のパフォーマンスは、1 次索引に大きく依存します。Teradata KM では、一時表に対するカスタマイズされた 1 次索引（PI）がサポートされます。これは Teradata IKM および CKM に適用されます。一時表に対する 1 次索引は、これらの KM で PRIMARY_INDEX KM オプションを使用して定義できます。これは次の値をとります。

- [PK]: PI はターゲット表の主キーです。これは CKM のデフォルト値です。
- [UK]: PI はインタフェースの更新キーです。これは IKM のデフォルト値です。
- <Column list>: これは、カンマ区切りの列名のリストに基づく任意の PI です。
- <Empty string>: 1 次索引が指定されていません。Teradata エンジンで PI のデフォルト・ルール（一時表の最初の列）が使用されます。

選択した PI 列について Teradata MultiColumnStatistics がオプションで収集されます。これは、COLLECT_STATS KM オプションで制御されます。デフォルトでは true に設定されています。

Teradata ユーティリティのサポート

Teradata ユーティリティ (TTU) では、Teradata エンジンから、あるいは Teradata エンジンヘデータを転送する効率的な方法が提供されます。TTU をサポートする LKM または IKM を使用する場合、TERADATA_UTILITY オプションを使用してデータをロードするための方法を設定できます。

Teradata ターゲット (IKM) またはステージング領域 (LKM) ヘデータをプッシュする場合、このオプションは次の値をとります。

- FASTLOAD: Teradata FastLoad を使用します。
- MLOAD: Teradata MultiLoad を使用します。
- TPUMP: Teradata TPump を使用します。
- TPT-LOAD: Teradata Parallel Transporter (ロード・オペレータ) を使用します。
- TPT-SQL-INSERT: Teradata Parallel Transporter (SQL 挿入オペレータ) を使用します。

Teradata からファイルヘデータをプッシュする場合、このオプションは次の値をとります。

- FEXP: Teradata FastExport を使用します。
- TPT: Teradata Parallel Transporter を使用します。

TTU KM を使用する場合、KM パラメータも考慮する必要があります。

- REPORT_NB_ROWS: このオプションを使用すると、ユーティリティで処理された行数が統合インタフェースの「警告」ステップでレポートされます。
- SESSIONS: FastLoad セッションの数。
- MAX_ALLOWED_ERRORS: 許容エラーの最大数。これは、FastLoad/MultiLoad/TPump の ERRLIMIT コマンドおよび TPT の ErrorLimit 属性に相当します。
- MULTILOAD_TPUMP_TYPE: MultiLoad または TPump ユーティリティで行われる処理。有効な値は INSERT、UPSERT および DELETE です。UPSERT および DELETE の場合、インタフェースで更新キーが必要です。

ユーティリティとロード・オペレータの詳細と適切な選択については、Teradata のドキュメントを参照してください。

名前付きパイプのサポート

TTU KM を使用して SQL ソースと Teradata との間でデータを移動する場合、アンロード / ロード・プロセスでファイルのかわりに名前付きパイプを使用することでパフォーマンスが向上します。名前付きパイプは、NP_USE_NAMED_PIPE オプションを YES に設定するとアクティブにできます。名前付きパイプを使用する場合、次のオプションも考慮する必要があります。

- NP_EXEC_ON_WINDOWS: ランタイム・エージェントが Windows プラットフォームで稼働する場合、このオプションを YES に設定します。
- NP_ACCESS_MODULE: 名前付きパイプで使用されるアクセス・モジュール。このアクセス・モジュールはプラットフォームに依存します。値のリストは、ナレッジ・モジュール・オプションのヘルプを参照してください。
- NP_TTU_STARTUP_TIME: TTU でパイプを使用してデータを受信できる秒数。これは、KM で TTU を起動した時点と KM で名前付きパイプへのデータのプッシュを始めた時点との遅延です。この遅延は、マシンの負荷によって異なります。

最適化された一時表管理

Data Integrator 一時ステージング表の作成および削除は、Teradata エンジンのリソースを消費する処理でもあります。ODI_DDL KM オプションでこれらの DDL 操作を制御する方法が提供されます。次の値をとります。

- **DROP_CREATE**: 実行のたびに常にすべての一時表を削除して再作成します（デフォルトの動作）。
- **CREATE_DELETE_ALL**: 必要な場合（通常最初の実行時のみ）一時表を作成し、**DELETE ALL** を使用して一時表の内容を削除します。一時表は後続の実行で再利用されます。
- **DELETE_ALL**: 一時表を作成しません。すべての一時表について **DELETE ALL** を発行するのみです。
- **NONE**: 一時表に対して **DDL** を発行しません。一時表は別途処理される必要があります。

