

IBM InfoSphere Federation Server

バージョン 10.1

**フェデレーテッド・データ・
ソースの構成ガイド**

IBM

IBM InfoSphere Federation Server

バージョン 10.1

**フェデレーテッド・データ・
ソースの構成ガイド**

IBM

お願い

本書および本書で紹介する製品をご使用になる前に、423 ページの『特記事項および商標』に記載されている情報をお読みください。

お客様の環境によっては、資料中の円記号がバックスラッシュと表示されたり、バックスラッシュが円記号と表示されたりする場合があります。

原典： SC19-3699-00
IBM InfoSphere Federation Server
Version 10.1
Configuration Guide for Federated Data Sources

発行： 日本アイ・ビー・エム株式会社

担当： トランスレーション・サービス・センター

第1刷 2012.5

© Copyright IBM Corporation 1998, 2012.

目次

第 1 章 フェデレーテッド・サーバーの構成	1
ラッパー構成ツールの実行	1
IBM Data Studio のフェデレーテッド・オブジェクト	2
フェデレーテッド・オブジェクトの作成: 管理エク	
スプローラー	4
フェデレーテッド・オブジェクトの作成と管理:	
SQL および XQuery エディター	9
フェデレーテッド・システムでのコマンドの発行	11
フェデレーテッド・サーバーのセットアップの検証	12
フェデレーテッド・サーバーがデータ・ソースへ	
アクセスできるようにする	12
データ・ソース・クライアントの必須環境変数の	
設定	13
ライブラリー・ファイルがデータ・ソース・クラ	
イアント・ソフトウェアにリンクしていることの	
検証	15
製品ライセンス・キーとポリシーの管理	18
DB2 pureScale フィーチャーのフェデレーション	
の構成	18
フェデレーテッド・データベースの作成	19
コード・セット、照合シーケンス、および各国語	
サポート	19
データ・ソースへアクセスするためのフェデレーテ	
ッド・サーバーの構成	25
大/小文字の区別と引用符の正しい使用法	27
データ・ソースへアクセスするための複数のフェ	
デレーテッド・サーバーの構成	30
第 2 章 データ・ソースの構成	33
BioRS データ・ソースへのアクセスの構成	33
BioRS ラッパー	33
フェデレーテッド・サーバーへの BioRS デー	
タ・ソースの追加	34
カスタム関数と BioRS 照会	50
BioRS ラッパー・パフォーマンスの最適化	61
DB2 データ・ソースへのアクセスの構成	67
DB2 ノード項目のカタログ	67
リモート DB2 データベースのカタログ	67
DB2 ラッパーの登録	68
DB2 データ・ソースのサーバー定義の登録	70
DB2 データ・ソースのユーザー・マッピングの作	
成	72
DB2 データ・ソース・サーバーとの接続のテスト	74
DB2 表およびビューのニックネームの登録	76
Excel データ・ソースへのアクセスの構成	77
Excel ラッパー	77
フェデレーテッド・サーバーへの Excel データ・	
ソースの追加	82
Informix データ・ソースへのアクセスの構成	90

Informix クライアント構成ファイルのセットアップ	
およびテスト	90
Informix 環境変数の設定	91
Informix ラッパーの登録	95
Informix データ・ソースのサーバー定義の登録	97
Informix データ・ソースのユーザー・マッピン	
グの作成	100
Informix サーバーへの接続のテスト	101
Informix 表、ビュー、およびシノニムのニック	
ネームの登録	103
JDBC データ・ソースへのアクセスの構成	105
JDBC を通じてデータ・ソースへアクセスするた	
めのフェデレーテッド・サーバーの準備	106
JDBC ラッパーの登録	107
JDBC データ・ソースのサーバー定義の登録	109
JDBC データ・ソースのユーザー・マッピングの	
作成	112
JDBC データ・ソース・サーバーへの接続のテス	
ト	114
JDBC データ・ソースの表およびビューのニック	
ネームの登録	115
Microsoft SQL Server データ・ソースへのアクセス	
の構成	116
Microsoft SQL Server データ・ソースにアクセス	
するためのフェデレーテッド・サーバーの準備	
(Windows)	117
Microsoft SQL Server データ・ソースにアクセス	
するためのフェデレーテッド・サーバーの準備	
(Linux、UNIX)	117
Microsoft SQL Server 環境変数の設定	120
Microsoft SQL Server ラッパーの登録	122
Microsoft SQL Server データ・ソースのサーバー	
定義の登録	123
Microsoft SQL Server データ・ソース用のユーザ	
ー・マッピングの作成	126
Microsoft SQL Server リモート・サーバーへの接	
続のテスト	127
Microsoft SQL Server 表およびビューのニックネ	
ームの登録	128
ODBC トレース情報による、Microsoft SQL	
Server データ・ソースへの接続のトラブルシュー	
ティング	130
ODBC データ・ソースへのアクセスの構成	131
ODBC を通じてデータ・ソースにアクセスする	
ためのフェデレーテッド・サーバーの準備	
(Windows)	133
ODBC を通じてデータ・ソースにアクセスする	
ためのフェデレーテッド・サーバーの準備	
(Linux、UNIX)	134
ODBC ラッパーの登録	135
ODBC データ・ソースのサーバー定義の登録	139

ODBC データ・ソースのユーザー・マッピングの作成	141
ODBC データ・ソース・サーバーへの接続のテスト	142
ODBC データ・ソース表およびビューのニックネームの登録	143
ODBC チューニング・ユーティリティ (db2fedsvrcfg) による ODBC ラッパーのパフォーマンスの最適化	145
ODBC ラッパーを使用した Excel データへのアクセス	149
IBM InfoSphere Classic Federation Server for z/OS データ・ソースへの ODBC アクセスの構成	153
Netezza データ・ソースへの ODBC アクセスの構成	155
OLE DB データ・ソースへのアクセスの構成	156
OLE DB ラッパーの登録	157
OLE DB データ・ソースのサーバー定義の登録	158
OLE DB データ・ソースのユーザー・マッピングの作成	160
Oracle データ・ソースへのアクセスの構成	161
Oracle 環境変数の設定	161
Oracle クライアント構成ファイルのセットアップおよびテスト	166
Oracle ラッパーの登録	167
Oracle データ・ソースのサーバー定義の登録	168
Oracle データ・ソースのユーザー・マッピングの作成	170
Oracle サーバーへの接続のテスト	172
Oracle 表およびビューのニックネームの登録	174
データ・ソースとしてのスクリプトへのアクセスの構成	175
スクリプト・ラッパーの概要	176
データ・ソースとしてのスクリプトのフェデレーテッド・システムへの追加	177
スクリプトのニックネームの登録	186
スクリプト・ラッパーによる SQL 照会	190
スクリプト・ラッパー・パフォーマンスの最適化	191
Sybase データ・ソースへのアクセスの構成	192
Adaptive Server Enterprise (ASE) 用の Sybase ラッパー・サポート	192
Sybase 環境変数の設定	194
Sybase クライアント構成ファイルのセットアップおよびテスト (Windows)	197
Sybase クライアント構成ファイルのセットアップおよびテスト (UNIX)	198
Sybase ラッパーの登録	199
Sybase データ・ソースのサーバー定義の登録	201
Sybase データ・ソースのユーザー・マッピングの作成	203
Sybase サーバーへの接続のテスト	205
Sybase 表およびビューのニックネームの登録	206
Sybase ラッパー構成のトラブルシューティング	207
表構造ファイル・データ・ソースへのアクセスの構成	209
表構造ファイル - 概要	209

表構造ファイルの属性	210
表構造ファイル・ラッパー	210
フェデレーテッド・サーバーへの表構造ファイル・データ・ソースの追加	212
表構造ファイル・ラッパーのファイル・アクセス・コントロール・モデル	219
表構造ファイル・ラッパーの照会パフォーマンス最適化のためのガイドライン	220
Teradata データ・ソースへのアクセスの構成	221
Teradata サーバーへの接続のテスト	221
Teradata ライブラリーがランタイム・リンクに対応しているかどうかの検査 (AIX)	222
Teradata 環境変数の設定	223
Teradata ラッパーの登録	228
Teradata データ・ソースのサーバー定義の登録	229
Teradata データ・ソースのユーザー・マッピングの作成	232
Teradata サーバーへの接続のテスト	233
Teradata 表およびビューのニックネームの登録	235
Teradata データ・ソース構成のトラブルシューティング	237
Web サービス・データ・ソースへのアクセスの構成	239
Web サービスおよび Web サービス・ラッパー	239
Web サービス・ラッパーの登録	246
Web サービス・データ・ソースのサーバー定義の登録	247
Web サービス・ラッパーを使用したセキュリティの使用可能化	249
Web サービス・データ・ソースのニックネームの登録	250
Web サービス・ラッパーの照会制約事項	261
Web サービス・データ・ソース - 照会の例	264
XML データ・ソースへのアクセスの構成	267
XML ラッパー	267
フェデレーテッド・システムへの XML の追加	271
XML データ・ソースの照会	288

第 3 章 フェデレーテッド・フィーチャーのデータ・ソース・サポート 295

第 4 章 データ・ソース・オプション・リファレンス 299

BioRS オプション・リファレンス	299
DB2 データベース・オプション・リファレンス	303
Excel オプション・リファレンス	311
Informix オプション・リファレンス	312
JDBC オプション・リファレンス	318
Microsoft SQL Server オプション・リファレンス	326
ODBC オプション・リファレンス	331
Oracle オプション・リファレンス	338
スクリプト・オプション・リファレンス	344
Sybase オプション・リファレンス	349
Teradata オプション・リファレンス	356
表構造ファイル・オプション・リファレンス	360
Web サービス・オプション・リファレンス	363

XML オプション・リファレンス	371
第 5 章 フェデレーテッド情報を含むグ ローバル・カタログ表内のビュー	379
第 6 章 フェデレーテッド・システムの 関数マッピング・オプション	383
第 7 章 SQL ステートメントで有効な サーバーのタイプ	385
第 8 章 データ・タイプ・マッピング	387
デフォルトの順方向データ・タイプ・マッピング	387
DB2 Database for Linux, UNIX, and Windows デ ータ・ソースのデフォルトの順方向データ・タイ プ・マッピング	387
DB2 for System i データ・ソースのデフォルト の順方向データ・タイプ・マッピング	388
DB2 for VM and VSE データ・ソースのデフォ ルトの順方向データ・タイプ・マッピング	389
DB2 for z/OS データ・ソースのデフォルトの順 方向データ・タイプ・マッピング	390
Informix データ・ソースのデフォルトの順方向 データ・タイプ・マッピング	391
JDBC データ・ソースのデフォルトの順方向デー タ・タイプ・マッピング	392
Microsoft SQL Server データ・ソースのデフォ ルトの順方向データ・タイプ・マッピング	394
ODBC データ・ソースのデフォルトの順方向デー タ・タイプ・マッピング	395
Oracle NET8 データ・ソースのデフォルトの順 方向データ・タイプ・マッピング	397
Sybase データ・ソースのデフォルトの順方向デー タ・タイプ・マッピング	397
Teradata データ・ソースのデフォルトの順方向デー タ・タイプ・マッピング	399
順方向データ・タイプ・マッピングのサンプル	400
デフォルトの逆方向データ・タイプ・マッピング	402
DB2 Database for Linux, UNIX, and Windows デ ータ・ソースのデフォルトの逆方向データ・タイ プ・マッピング	403
DB2 for System i データ・ソースのデフォルト の逆方向データ・タイプ・マッピング	404
DB2 for VM and VSE データ・ソースのデフォ ルトの逆方向データ・タイプ・マッピング	405
DB2 for z/OS データ・ソースのデフォルトの逆 方向データ・タイプ・マッピング	406

Informix データ・ソースのデフォルトの逆方向 データ・タイプ・マッピング	406
JDBC データ・ソースのデフォルトの逆方向デー タ・タイプ・マッピング	407
Microsoft SQL Server データ・ソースのデフォ ルトの逆方向データ・タイプ・マッピング	408
ODBC データ・ソースのデフォルトの逆方向デー タ・タイプ・マッピング	409
Oracle NET8 データ・ソースのデフォルトの逆 方向データ・タイプ・マッピング	410
Sybase データ・ソースのデフォルトの逆方向デー タ・タイプ・マッピング	411
Teradata データ・ソースのデフォルトの逆方向デー タ・タイプ・マッピング	412
Unicode のデフォルトのデータ・タイプ・マッピン グ	413
JDBC データ・ソースのデフォルトの順方向デー タ・タイプ・マッピング (Unicode)	413
JDBC データ・ソースのデフォルトの逆方向デー タ・タイプ・マッピング (Unicode)	414
Microsoft SQL Server データ・ソースのデフォ ルトの順方向データ・タイプ・マッピング (Unicode)	414
Microsoft SQL Server データ・ソースのデフォ ルトの逆方向データ・タイプ・マッピング (Unicode)	414
NET8 データ・ソースのデフォルトの順方向デー タ・タイプ・マッピング (Unicode)	415
NET8 データ・ソースのデフォルトの逆方向デー タ・タイプ・マッピング (Unicode)	415
ODBC データ・ソースのデフォルトの順方向デー タ・タイプ・マッピング (Unicode)	416
ODBC データ・ソースのデフォルトの逆方向デー タ・タイプ・マッピング (Unicode)	416
Sybase データ・ソースのデフォルトの順方向デー タ・タイプ・マッピング (Unicode)	417
Sybase データ・ソースのデフォルトの逆方向デー タ・タイプ・マッピング (Unicode)	417
非リレーショナル・データ・ソースでサポートされ るデータ・タイプ	417
アクセシビリティ対応資料	421
特記事項および商標	423
索引	427

第 1 章 フェデレーテッド・サーバーの構成

データ・ソースを構成する前に、フェデレーテッド・サーバーを構成して、正しくセットアップされているかを検証します。

手順

フェデレーテッド・サーバーを構成するには、以下の作業を実行します。

ラッパー構成ツールの実行

Federation Server のラッパー構成ツールはバージョン 10 で導入されます。この構成ツールは、リレーショナル・ラッパーの構成に使用する GUI インターフェースを提供します。

始める前に

構成ツールをインストールして使用する前に、ご使用のフェデレーテッド・システムが正しくセットアップされていることを確認します。構成対象のデータ・ソースにラッパーがインストールされている必要があります。

このタスクについて

構成ツールは、既存の DB2[®] インスタンスがあるかどうかを検出します。構成したいインスタンスおよびラッパーを選択し、それぞれのラッパーに必須およびオプションの環境変数を入力します。この入力に基づき、構成ツールは db2dj.ini ファイルに環境変数を書き込み、DB2 インスタンスの .profile をエクスポートします。Informix[®] および Teradata ラッパーの場合、このツールは djxLink を実行します。構成が完了すると、DB2 の再始動を選択して、構成を有効にできます。

構成ツールを開始すると、フェデレーション・インストーラーはオペレーティング・システム、ユーザー権限、およびインストール済みの DB2 のバージョンを判別します。構成ツールはこの情報に基づいて、構成可能なラッパーを提示します。

制限: 構成ツールは Linux および UNIX オペレーティング・システムをサポートします。ユーザーは構成ツールを使用して以下のリレーショナル・ラッパーを構成できます。

- Informix
- Microsoft SQL Server
- Oracle
- Sybase
- Teradata

この構成ツールは ODBC および JDBC ラッパーをサポートしません。

手順

1. 構成ツールを起動します。

インストール・パスのbin・フォルダー /installation_path/bin にアクセスし、rwcfg.bin を実行して構成ツールを起動します。root ユーザーが製品をインストールした場合は、ユーザー ID にroot 権限が必要です。root ユーザー以外が製品をインストールした場合は、ユーザー ID に root ユーザー以外の権限が必要です。

2. 構成するインスタンスおよびリレーショナル・ラッパーを選択します。

「ようこそ」パネルで、構成するインスタンスを選択するようにプロンプトがでます。「構成するインスタンスおよびラッパーの選択 (Select instance and wrapper to configure)」パネルで次のようにします。

- 構成するインスタンスを選択します。選択したインスタンスに基づいてインストールされたラッパーがパネルに表示されます。
- 構成するラッパーを選択します。

3. リレーショナル・ラッパーを構成します。

それぞれのラッパーごとに、必須およびオプションのパラメーターを入力するようにプロンプトが出ます。構成ツールはすべてのシステム変数の設定および djxlink スクリプトの実行方法をガイドします。

4. 事前構成サマリーをレビューして構成を実行します。

事前構成サマリー・パネルには入力データが集約されます。なんらかの変更を行う必要がある場合、または選択されたコンポーネントの構成を続行する場合は、直前のパネルに戻ることができます。構成を実行すると、構成ツールは入力した環境変数を db2dj.ini およびインスタンスの .profile に設定し、必要な場合は .profile および djxlink を実行します。

5. フェデレーテッド・サーバーを再始動します。

構成が完了すると、「フェデレーション・サーバーの再始動 (Restart Federation Server)」パネルが表示されます。フェデレーテッド・サーバーをすぐに再始動することも、後で再始動することもできます。

6. 構成後のサマリーを確認してください。構成が完了すると、「構成サマリー (Configuration Summary)」パネルが表示されます。このパネルには以下の情報が表示されます。

- 選択されたコンポーネントの状況。
- 戻りコード、警告およびエラー・メッセージ、およびログ・ファイル情報などの構成状況の詳細。

IBM Data Studio のフェデレーテッド・オブジェクト

フェデレーテッド・サーバーをインストールしてセットアップし、フェデレーテッド・データベースを作成した後で、IBM® Data Studio を使用してフェデレーテッド・オブジェクトを作成して管理することができます。

管理エクスプローラーとオブジェクト・リストを使用して、フェデレーテッド・サーバーをセットアップし、リモート・データ・サーバー上のサポートされるデータ・ソースにアクセスできます。

IBM InfoSphere® Federation Serverの Data Studio サポートは、特定の Data Studio 機能に制限されます。使用できる機能は、アクセスするデータ・ソースにより異なります。これらの制限は、以下に要約されています。

- Data Studio は、データベース・オブジェクトのナビゲートに使用できる 2 つのビュー (管理エクスプローラーおよびデータ・ソース・エクスプローラー) を備えています。管理エクスプローラーを使用してください。データ・ソース・エクスプローラーはフェデレーテッド・オブジェクトをサポートしません。
- 管理エクスプローラーからフェデレーテッド・オブジェクトを作成および管理する Data Studio サポートでは、以下のデータ・ソースへの制限があります。
 - DB2 Database for Linux, UNIX, and Windows
 - DB2 for i
 - DB2 for z/OS®
 - Oracle
- 他のデータ・ソースについては、DB2 コマンド行から SQL ステートメントを発行する代わりに、Data Studio の SQL および XQuery Editor を使用してコマンドの発行と実装ができます。

次のトピックを使用して、作業対象のデータ・ソースのタイプに該当する Data Studio の機能に習熟してください。

- サポートされる DB2 データ・ソースおよび Oracle データ・ソースへのアクセス権限の構成については、4 ページの『フェデレーテッド・オブジェクトの作成: 管理エクスプローラー』を参照してください。
- 他のデータ・ソースへのアクセス権限の構成については、9 ページの『フェデレーテッド・オブジェクトの作成と管理: SQL および XQuery エディター』を参照してください。

Data Studio でのフェデレーテッド・オブジェクトのロードマップ

次のロードマップは、Data Studio でフェデレーテッド・オブジェクトを処理する、インストール、構成、および管理の各リソースを示しています。

表 1. Data Studio でのフェデレーテッド・オブジェクト管理へのロードマップ

タスク	リソース
Data Studio のインストール	Data Studio フル・クライアント または Data Studio 管理クライアントをインストールする必要があります。 SQL スクリプトの実行にジョブ・スケジューリングの使用を選択する場合は、オプションで、Data Studio Web コンソールをインストールできます。
接続の作成	データベースへの接続
オブジェクトの表示と管理	「オブジェクト・リスト」でのデータベース・オブジェクトの表示と管理
オブジェクト変更の管理	「プロパティ」ビューを使用したデータベース・オブジェクト変更の管理
コマンドの編集	SQL および XQuery エディターでのコマンドの編集 (手動)
特権の管理	特権の管理

表 1. Data Studio でのフェデレーテッド・オブジェクト管理へのロードマップ (続き)

タスク	リソース
変更プランを使用した作業	シナリオ: 変更プランを使用した基本的なデータベース・オブジェクト変更の作成
データベースの管理	タスク・アシストを使用したデータベースの管理

Data Studio の全機能の詳細情報は、IBM Data Studio バージョン 3.1.1 インフォメーション・センター内にあります。

フェデレーテッド・オブジェクトの作成: 管理エクスプローラー

サポートされている DB2 データ・ソースおよび Oracle データ・ソースの場合、Data Studio 管理エクスプローラーを使用してフェデレーテッド・サーバーをセットアップし、リモート・データ・サーバー上のデータ・ソースにアクセスできます。

フェデレーテッド・システムを構成するには、最初に、データ・ソースにアクセスするために必要な個々のフェデレーテッド・オブジェクトを作成します。その上で、オブジェクトを個々に管理する、または複数のフェデレーテッド・オブジェクトを構成する、さらに、追加の構成タスクと管理タスクを実行するために、管理エクスプローラー内でそれらのオブジェクトを処理することができます。

個々のフェデレーテッド・オブジェクトの作成

Data Studio の管理エクスプローラーを使用して、フェデレーテッド・サーバーのセットアップに必要なフェデレーテッド・オブジェクトをそれぞれ作成し、サポートされる DB2 データ・ソースおよび Oracle データ・ソースにアクセスすることができます。

このタスクについて

以下の手順では、管理エクスプローラーを使用して、データベース接続を作成し、サポートされる DB2 データ・ソースおよび Oracle データ・ソースのアクセスに必要なフェデレーテッド・オブジェクトを作成して実装する方法を説明します。

管理エクスプローラーはデータベース・オブジェクトが含まれるオブジェクト・ツリーです。オブジェクトは、オブジェクト・タイプに基づいて複数のフォルダーへグループ化されます。フェデレーテッド・オブジェクトを作成して管理するには、管理エクスプローラーの「フェデレーテッド・データベース・オブジェクト」フォルダーおよびその他のメイン・エリアで作業します。

オブジェクト・リスト

オブジェクト・リスト域を使用してオブジェクトの内容を表示したり、エディットすることができます。

「プロパティ」ビュー

「プロパティ」ビューは、オブジェクト・プロパティを表示したり、オブジェクト定義を作成、変更する場所です。

手順

- 以下のいずれかの方法を使用して、データベースに接続します。
 - 新規データベース接続を作成します。

- a. 管理エクスプローラーで、「新規」 > 「データベースへの新規接続」をクリックします。
 - b. 接続ダイアログで、パラメーターを入力して接続を構成します。
 - c. データベースに接続するために右クリックします。
- 既存のデータベース接続に接続します。
 - a. 管理エクスプローラーで、接続プロファイルを右クリックします。
 - b. 「接続」をクリックします。

注: フェデレーテッド・データベースまたは DB2 データベースが、Data Studio がインストールされたマシンにカタログ済みであれば、そのデータベース接続プロファイルは、Data Studio の起動時に自動的に作成されます。

2. 必要なフェデレーテッド・オブジェクトを作成します。

- ラッパー
- サーバー定義
- ユーザー・マッピング
- ニックネーム

またフェデレーテッド・ストアド・プロシージャを作成することもできます。

- a. 「フェデレーテッド・データベース・オブジェクト」フォルダーの内容を展開します。
- b. フェデレーテッド・オブジェクト・フォルダーを右クリックし、ドロップダウン・メニューで、作成するオブジェクトを選択します。

データベースに行う変更はローカル・ワークスペースの変更プランに保存されます。ユーザーは変更プランを使用して、フェデレーテッド・データベースまたは DB2 データベースに変更を定義し、その変更内容の実装方法を定義できます。

例えば、「ラッパー」フォルダーを右クリックし、「ラッパーの作成」を選択します。ラッパーを作成すると、「オブジェクト・リスト」域が開き、新規のラッパーが「デフォルトの変更プラン」ビューに表示されます。

- c. 「プロパティ」ビューでオブジェクト属性とオプションを指定します。

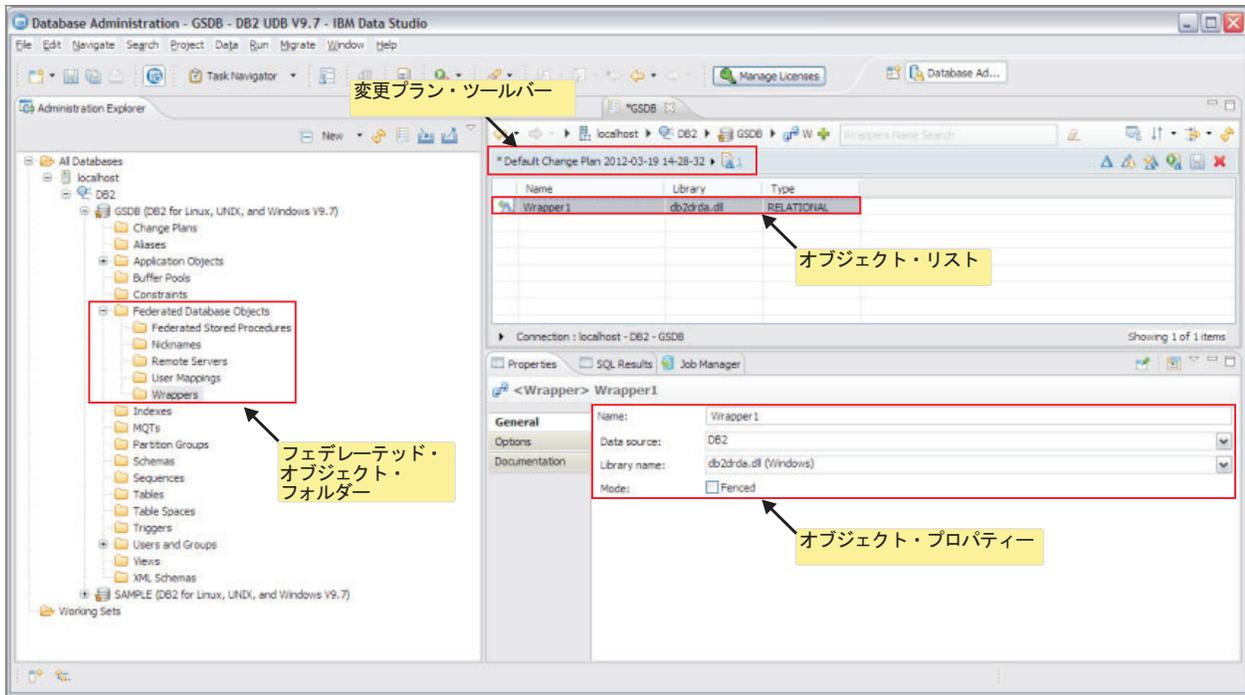


図 1. 管理エクスプローラー、オブジェクト・リスト、およびプロパティ・ビューの例

フェデレーテッド・オブジェクトには以下のプロパティが必要です。

表 2. 必要なオブジェクト・プロパティ

フェデレーテッド・オブジェクト	必要なオブジェクト・プロパティ
ラッパー	<ul style="list-style-type: none"> ライブラリー名 DBNAME オプション
サーバー	<ul style="list-style-type: none"> 名前 タイプ バージョン リモート・ユーザー ID リモート・パスワード
ユーザー・マッピング	<ul style="list-style-type: none"> ローカル・ユーザー ID サーバー名 リモート・ユーザー ID リモート・パスワード
ニックネーム	<ul style="list-style-type: none"> 名前 サーバー リモート・スキーマ リモート表

表2. 必要なオブジェクト・プロパティ (続き)

フェデレーテッド・オブジェクト	必要なオブジェクト・プロパティ
フェデレーテッド・ストアード・プロシージャ	<ul style="list-style-type: none"> 名前 サーバー リモート・スキーマ リモート・プロシージャ

3. コマンドを確認し、実装します。

オブジェクト定義の完了後、「変更内容の確認と実装 (Review and Deploy Changes)」をクリックし、生成された DDL を確認してオブジェクト定義を実装します。コマンドのエディット、コマンドの実行、または計画をスケジュールして後で実行することができます。

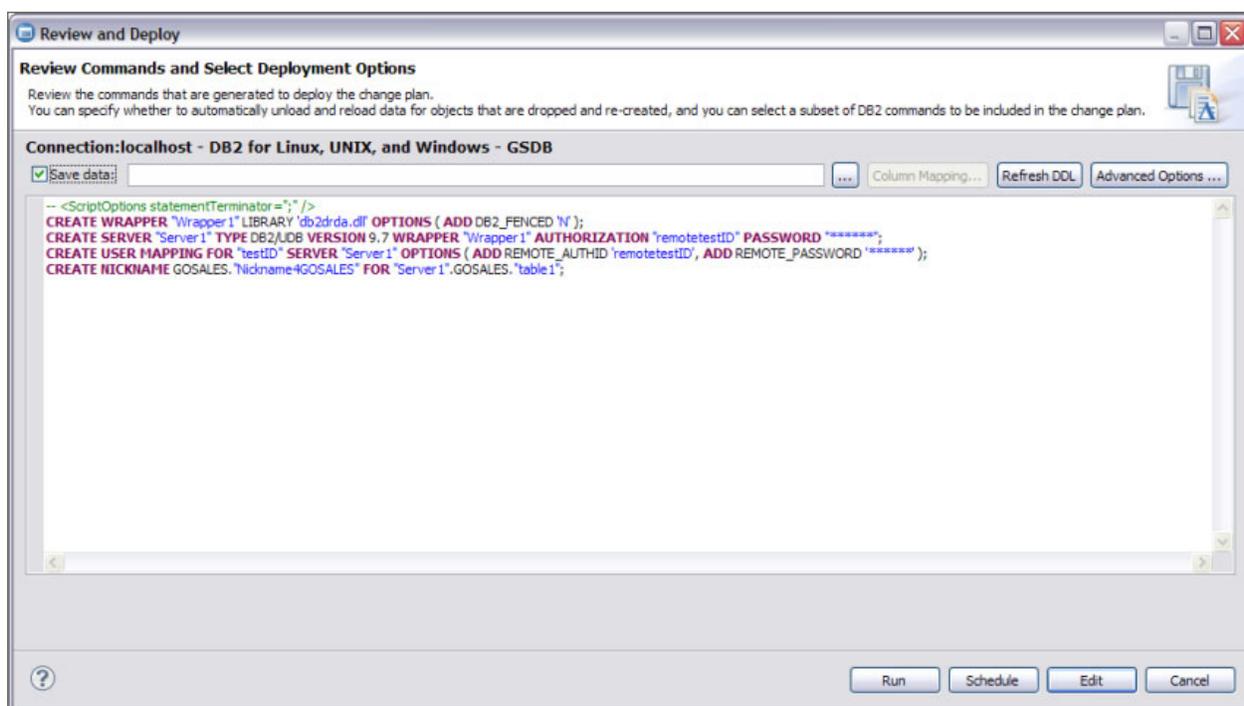


図2. ラッパー、サーバー、ユーザー・マッピング、およびニックネームを使用した「確認と実装 (Review and Deploy)」ダイアログの例。

タスクの結果

これで、オブジェクトを管理するために「フェデレーテッド・データベース・オブジェクト」フォルダーに作成したフェデレーテッド・オブジェクトにアクセスすることができます。例えば、任意のラッパーを選択し、「ラッパーの変更 (Alter wrapper)」選択してラッパー定義の変更を行うか、あるいは「ラッパーの除去 (Drop wrapper)」を選択してラッパーを削除することができます。

複数のフェデレーテッド・オブジェクトの作成と管理

フェデレーテッド・オブジェクトを作成した後は、個々の、あるいは複数のフェデレーテッド・オブジェクトを構成して管理するために、Data Studio の管理エクスプローラーのビューで作業することができます。

このタスクについて

フェデレーテッド・オブジェクトをより効率的に処理するために、複数のオブジェクトを作成または変更できます。

オブジェクトを作成した後、「SQL および XQuery エディター」にスクリプトを送り、次に、SQL ステートメントをカット・アンド・ペーストして複数のオブジェクトを作成することができます。例えば、管理エクスプローラー内でラッパーを作成できます。次に、そのラッパーからコマンドをコピーして、新規の複数のラッパーを作成できます。

以下の手順は、複数のオブジェクトの管理方法の説明です。

手順

1. 『4 ページの『フェデレーテッド・オブジェクトの作成: 管理エクスプローラー』』の手順を実行して、必要なフェデレーテッド・オブジェクトを作成します。

SQL コマンドの詳細例とデータ・ソースにアクセスするために作成する特定のオブジェクトに関しては、『33 ページの『第 2 章 データ・ソースの構成』』を参照してください。

2. 「**エディット**」オプションを選択し、作成したオブジェクトのコマンドを開きます。

「SQL および XQuery エディター」にコマンドが表示されます。オブジェクト定義を実装する前に、このエディターを使用して複数のオブジェクトを追加し、エディットします。

3. 残りのオブジェクトに対してコマンドをコピーし、エディットします。

以前に作成したオブジェクトからコマンドをコピーします。エディターでコマンドをコピーして貼り付け、新規のオブジェクトを作成します。

4. 「**SQL の実行**」をクリックして SQL スクリプトを実装します。

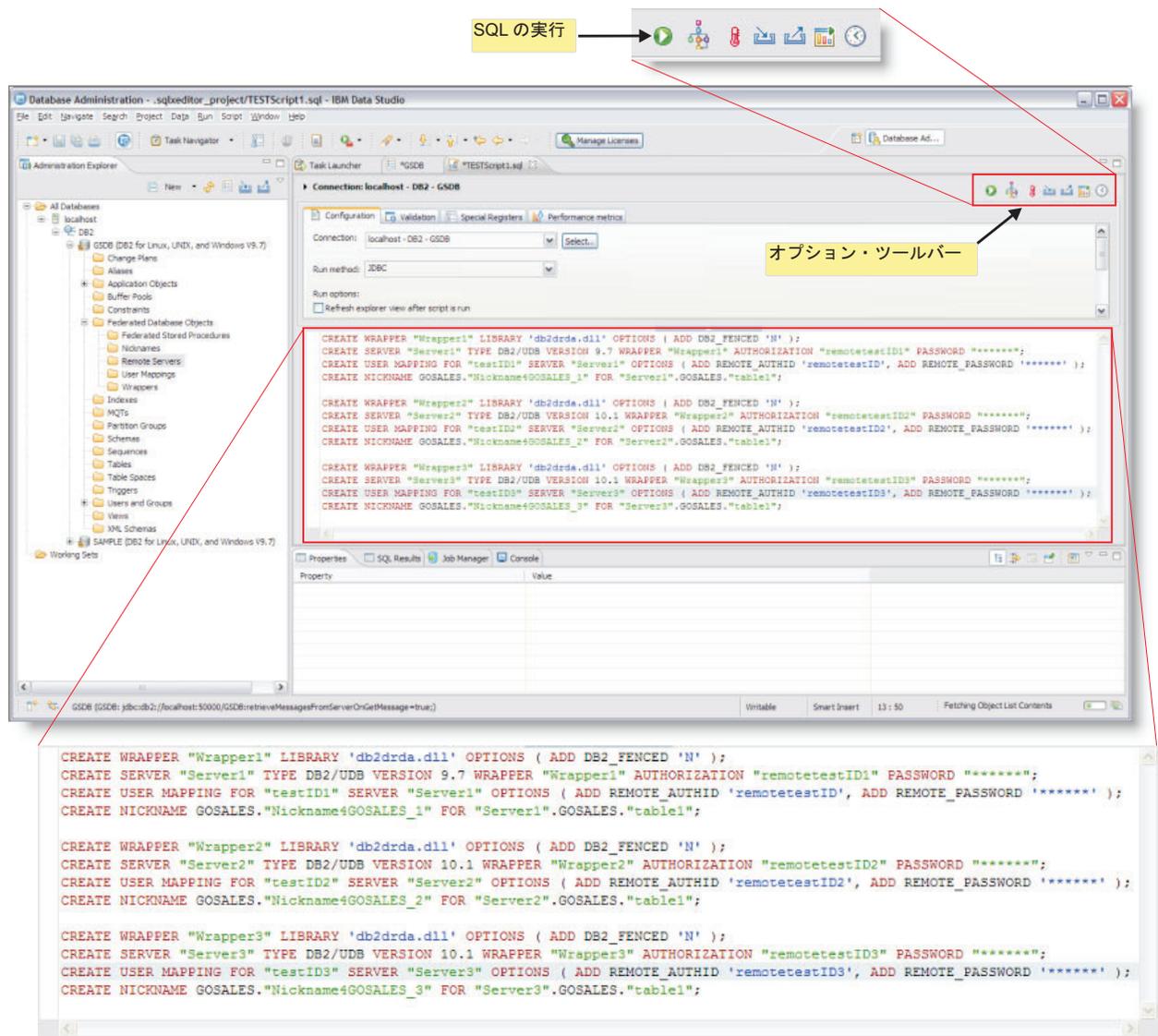


図3. 「SQL および XQuery エディター」での、複数のフェデレーテッド・オブジェクトの作成例

フェデレーテッド・オブジェクトの作成と管理: SQL および XQuery エディター

Data Studio の「SQL および XQuery エディター」を使用して、任意のデータ・ソースへのアクセスを構成するためのコマンドを発行および実装することができます。

このタスクについて

Data Studio の「SQL および XQuery エディター」での処理作業により、単独または複数のフェデレーテッド・オブジェクト定義を作成およびエディットしたり、コマンドを実装できます。以下の手順は、データベース接続の作成方法および「SQL および XQuery エディター」の起動と使用方法の説明です。

手順

1. 必要なフェデレーテッド・オブジェクトを作成します。

- ラッパー
 - サーバ定義
 - ユーザー・マッピング
 - ニックネーム
- a. 「SQL および XQuery エディター」を起動します。

「ファイル」メニューから、「ファイル」 > 「新規」 > 「その他」をクリックします。次に、「データ」 > 「SQL または XQuery スクリプト」を選択します。

- b. 「新規 SQL または XQuery スクリプト」ウィザードで必要事項を入力し、「SQL および XQuery エディター」を選択された状態のままにします。
- c. 「完了」をクリックします。

ウィザードで選択したデータ開発プロジェクトの「SQL スクリプト」フォルダーにステートメントが追加されます。「SQL および XQuery エディター」でステートメントが開きます。

- d. 「接続プロファイルの選択 (Select Connection Profile)」ウィザードを使用して、スクリプトをデータベースへ接続します。
- e. 作成するオブジェクトに対して SQL または XQuery ステートメントを完成します。

2. 残りのオブジェクトに対してコマンドをコピーし、エディットします。

3. オプション・ツールバーで「SQL の実行」をクリックし、SQL スクリプトを実行します。

例

以下の例は、「SQL および XQuery エディター」を使用したフェデレーテッド・オブジェクトの作成方法を示しています。

例 1: ODBC データ・ソースの複数のサーバーへのアクセスを構成するために、フェデレーテッド・オブジェクトを作成します。

1. 必要なフェデレーテッド・オブジェクトごとに作成します。

```
CREATE WRAPPER ODBC LIBRARY 'libdb2rcodbc.so' OPTIONS (MODULE '/opt/lib/odbc.so');
CREATE SERVER mysql_server TYPE mysql
  VERSION 4.0 WRAPPER ODBC
  OPTIONS (NODE 'odbc_node', DBNAME 'venice');
CREATE USER MAPPING FOR arturo SERVER mysql_server
  OPTIONS (REMOTE_AUTHID 'art', REMOTE_PASSWORD 'red4blue');
CREATE NICKNAME cust_europe FOR mysql_server."vinnie"."italy";
```

2. 残りのサーバーに複数のオブジェクトを作成するために、SQL をコピーして貼り付けます。

```
CREATE WRAPPER ODBC LIBRARY 'libdb2rcodbc.so' OPTIONS (MODULE '/opt/lib/odbc.so');
CREATE SERVER mysql_server TYPE mysql
  VERSION 4.0 WRAPPER ODBC
  OPTIONS (NODE 'odbc_node', DBNAME 'venice');
CREATE USER MAPPING FOR arturo SERVER mysql_server
  OPTIONS (REMOTE_AUTHID 'art', REMOTE_PASSWORD 'red4blue');
CREATE NICKNAME cust_europe FOR mysql_server."vinnie"."italy";
```

```
CREATE WRAPPER ODBC2 LIBRARY 'libdb2rcodbc.so' OPTIONS (MODULE '/opt/lib/odbc.so');
CREATE SERVER mysql_server2 TYPE mysql
  VERSION 4.0 WRAPPER ODBC
  OPTIONS (NODE 'odbc_node', DBNAME 'venice');
CREATE USER MAPPING FOR arturo SERVER mysql_server2
  OPTIONS (REMOTE_AUTHID 'art', REMOTE_PASSWORD 'red4blue');
CREATE NICKNAME cust_europe2 FOR mysql_server2."vinnie"."italy";
```

例 2: 複数のサーバー定義を作成します。

1. サーバー定義を作成します。

```
CREATE SERVER mysql_server2 TYPE mysql
  VERSION 4.0 WRAPPER ODBC;
```

2. 追加のサーバー定義を作成するために、DDL をコピーして貼り付け、エディットします。

```
CREATE SERVER mysql_server2 TYPE mysql
  VERSION 4.0 WRAPPER ODBC;
CREATE SERVER mysql2_server2 TYPE mysql
  VERSION 4.0 WRAPPER ODBC2;
CREATE SERVER mysql3_server2 TYPE mysql
  VERSION 4.0 WRAPPER ODBC3;
```

フェデレーテッド・システムでのコマンドの発行

フェデレーションを初めて操作する場合は、DB2 コマンド行の開始方法を確認してください。このインターフェースを使用して、フェデレーテッド・システムで多くのタスクを実行します。

このタスクについて

DB2 コマンド行を使用するには、タスクの実行に必要な DB2 コマンドを入力します。既に DB2 の操作に慣れている方は、コマンド行のほうが使いやすいでしょう。

IBM Data Studio を使用することもできます。サポートされているデータ・ソースに対しては、「管理エクスプローラー (Administration Explorer)」を使用できます。「管理エクスプローラー (Administration Explorer)」には、ダイアログ・ボックスとウィザードが用意され、ウィザードを使用すると、タスクの実行に必要な DB2 コマンドに関する知識がなくてもタスクを完了することができます。「SQL および XQuery エディター」もコマンドを発行およびデプロイするためのオプションで、DB2 コマンド行からの SQL ステートメント発行の代わりに使用できます。

タスクの実行方法として両方のインターフェースを使用できる場合、ドキュメンテーションでは、実行する固有のステップと入力するコマンドを併記しています。

手順

インターフェースにアクセスするには、以下のようになります。

- DB2 コマンド行を開始するには、オペレーティング・システムのプロンプトで db2 と入力します。すると、コマンドを連続して入力することができます。DB2 コマンド行の使用を終えた後は、quit と入力します。
- Data Studio を使用するには、アクセスの必要があるデータ・ソースに応じて Data Studio のフェデレーテッド・オブジェクトの指示に従ってください。

フェデレーテッド・サーバーのセットアップの検証

フェデレーションのインストール時に、ソフトウェアによりフェデレーテッド・サーバーの構成が試みられます。データ・ソースの構成時に問題が発生しないように、フェデレーテッド・サーバーが適切にセットアップされているか検証してください。

このタスクについて

フェデレーテッド・サーバーのセットアップを検証するには、以下の作業をすべて完了します。

フェデレーテッド・サーバーがデータ・ソースへアクセスできるようにする

フェデレーテッド・サーバーがデータ・ソースへアクセスできるようにするには、FEDERATED パラメーターを YES に設定しなければなりません。

このタスクについて

インストール・プログラムが FEDERATED パラメーターの設定を試行しますが、正しく YES に設定されているか確認するのが最善です。FEDERATED パラメーターが適切に設定されていない場合、データ・ソースにアクセスしようとすると、すべてメッセージ SQL20076、理由コード 1 で失敗します。

以下の手順は、FEDERATED パラメーターが適切に設定されているかを検証し、必要に応じて有効にする方法を説明します。

手順

1. DB2 コマンド行から、パラメーターとその現在の設定を表示する以下のコマンドを発行します。

```
GET DATABASE MANAGER CONFIGURATION
```
2. 以下のように、MAX_CONNECTIONS パラメーターの値をチェックして、コンセントレーターがオンかどうかを判別します。
 - MAX_CONNECTIONS が MAX_COORDAGENTS と等しい場合は、コンセントレーターはオフです。
 - MAX_CONNECTIONS が MAX_COORDAGENTS より大きい場合は、コンセントレーターはオンです。

FEDERATED パラメーターが YES に設定されていると、MAX_CONNECTIONS パラメーターをアクティブにすることはできません。コンセントレーターがオンの場合は、MAX_CONNECTIONS の値を、MAX_COORDAGENTS と等しい値に変更してください。

3. FEDERATED パラメーターの設定をチェックします。FEDERATED パラメーターが NO に設定されている場合は、設定を YES に変更します。設定を変更するには、以下のコマンドを発行します。

```
UPDATE DATABASE MANAGER CONFIGURATION USING FEDERATED YES
```

データ・ソース・クライアントの必須環境変数の設定

必須の環境変数は、必ず `db2dj.ini` ファイルに設定されるようにします。このタスクは、Informix、Microsoft SQL Server、Oracle、Sybase、および Teradata の各データ・ソース、および JDBC ラッパーと ODBC ラッパーを使用するアプリケーションで必要です。

始める前に

システム管理者が、以下のタスクを実行します。

このタスクについて

フェデレーテッド・サーバーのインストール前にクライアント・ソフトウェアをインストールする場合、インストール・プログラムによって、選択するデータ・ソースに必要な環境変数が自動的に設定されます。フェデレーテッド・サーバーのインストール後にクライアント・ソフトウェアをインストール、新バージョンのクライアント・ソフトウェアにアップグレード、新しいハードウェアにマイグレーションのどれかを行う場合、次の方法のいずれかを使用して環境変数を設定できます。

- ラッパー構成ツールを使用して、環境変数を設定します。
- 環境変数を手動で設定します。

次の表に、必須の環境変数をデータ・ソース別にリストします。

表 3. データ・ソース別の必須の環境変数

データ・ソース	必須の変数	値
Informix	INFORMIXDIR INFORMIXSERVER	INFORMIXDIR の場合、Informix クライアント・ソフトウェアの含まれるディレクトリー INFORMIXSERVER の場合、デフォルトの Informix サーバーの名前
JDBC	インストールされる JDBC ドライバーにより異なります。CREATE SERVER ステートメントの DRIVER_PACKAGE サーバー・パラメーターで JDBC ドライバー・パッケージを指定しない場合、"CLASSPATH" システム環境変数で JDBC ドライバー・パッケージを指定しなければなりません。システム環境変数の設定に関する特定の情報は、JDBC ドライバーの資料を参照してください。	JDBC ドライバー・ライブラリーのファイル名とディレクトリー。
Microsoft SQL Server	DJX_ODBC_LIBRARY_PATH	ODBC ライブラリーの含まれるディレクトリー
ODBC	ODBC アプリケーションによって異なります。	ODBC アプリケーションによって異なります。

表 3. データ・ソース別の必須の環境変数 (続き)

データ・ソース	必須の変数	値
Oracle	ORACLE_HOME	Oracle クライアント・ソフトウェアの含まれるディレクトリー
Sybase	SYBASE、 SYBASE_OCS	SYBASE の場合は、Sybase クライアント・ソフトウェアの含まれるディレクトリー SYBASE_OCS の場合は、クライアント・ソフトウェアのディレクトリー、バージョン、およびリリース
Teradata	COPYLIB	Teradata クライアント・ソフトウェアの含まれるディレクトリー

手順

1. db2dj.ini ファイルを開きます。

インストール中に、フェデレーテッド・サーバーに db2dj.ini ファイルが追加されます。デフォルトでは、このファイルは DB レジストリー変数 DB2_DJ_INI が指定するロケーションにあります。DB2_DJ_INI レジストリー変数が設定されていない場合、このファイルは以下のロケーションにあります。

- UNIX の場合、このファイルは *instancehome*/sqllib/cfg/db2dj.ini にあります。*instancehome* はインスタンス所有者のホーム・ディレクトリーです。
- Microsoft Windows の場合、このファイルは %DB2PATH%\cfg\%db2dj.ini にあります。%DB2PATH% は DB2 データベース・システムがインストールされているディレクトリーです。例: C:\Program Files\IBM\sqllib。

2. 環境変数の値を設定するには、構文 *variable_name=variable_value* を使用します。 *variable_value* がファイルまたはディレクトリー名の場合は、完全修飾パスを指定する必要があります。例えば、INFORMIDIR 変数をホーム・ディレクトリー /home/user1 のディレクトリー informix に設定するには、db2dj.ini ファイルに次のエントリーを含めます。

```
INFORMIDIR=/home/user1/informix
```

3. DB2 のコマンド行から、以下のコマンドを発行します。

```
db2stop
db2start
```

環境変数を有効にするには、データベース・インスタンスを停止して、再開します。

4. 複数パーティション・システムを構成している場合は、各パーティションに db2dj.ini ファイルをコピーします。

db2dj.ini ファイルの制約事項

db2dj.ini ファイルには、データ・ソースに必須の環境変数とオプションの環境変数が含まれています。

db2dj.ini ファイルを変更するときは、以下の制約事項を守ってください。

それぞれの変数を指定するには、`variable_name=variable_value` という形式を使用します。

説明

- `variable_name` は、環境変数の名前です。
- `variable_value` は値です。

`variable_value` としてファイル名またはディレクトリー名を指定するには、完全修飾名を指定します。この名前には、ファイル名の `~` (波形記号) などのメタ文字や `$HOME` などの環境変数を入れることはできません。例えば、`INFORMIXDIR` 環境変数を `/home/user1` という名前のホーム・ディレクトリーの `informix` ディレクトリーに設定するには、次の値を `db2dj.ini` ファイル内で指定します。

```
INFORMIXDIR=/home/user1/informix
```

以下の制限を遵守してください。

- 環境変数名の最大長は 255 バイトです。
- 環境変数の値の最大長は 756 バイトです。
- `db2dj.ini` ファイルの 1 行の最大長は 1021 バイトで、この長さを超えるデータは無視されます。

ライブラリー・ファイルがデータ・ソース・クライアント・ソフトウェアにリンクしていることの検証

最適な結果を得るためには、`CREATE WRAPPER` コマンドを使用してラッパーを登録する前に、ライブラリー・ファイルがデータ・ソース・クライアント・ソフトウェアに正しくリンクしているか常に検証してください。このタスクは、フェデレーテッド・サーバーが `UNIX` を使用しており、データ・ソース・クライアントが `Informix`、`Microsoft SQL Server`、`Oracle`、`Sybase`、または `Teradata` である場合のみ適用されます。

このタスクについて

フェデレーションをインストールする際に、ラッパーを自動的に構成するよう選択できます。インストール・プログラム中に、フェデレーションをサーバーにインストールしてラッパーの自動構成を選択する前に、データ・ソース・クライアント・ソフトウェアをフェデレーテッド・サーバーにインストールすると、ラッパー・ライブラリー・ファイルが対応するデータ・ソース・クライアント・ソフトウェアにリンクされます。ラッパー・ライブラリー・ファイルへのリンクは、データ・ソースによって異なります。

- `Informix`、`Microsoft SQL Server`、`Sybase`、および `Teradata` のデータ・ソースの場合:

例えば新規バージョンのクライアントに更新するために、フェデレーションのインストール後にデータ・ソース・クライアント・ソフトウェアをインストールする場合には、ラッパー・ライブラリー・ファイルをデータ・ソース・クライアント・ソフトウェアに手動でリンクしなければなりません。

- `Oracle` データ・ソースの場合:

ラッパー・ライブラリー・ファイルが動的にリンクされます。動的にリンクすることで、実行時に Oracle クライアント・ライブラリーをロードし、Oracle Call Interface (OCI) 関数アドレスを自動的に解決します。

リンクが失敗する一般的な原因として、以下の原因があります。

- インストール中に必須の環境変数が使用できない。
- データ・ソース・クライアントがフェデレーテッド・サーバーにインストールされていない。
- フェデレーテッド・サーバーにインストールされているデータ・ソース・クライアントが、サポートされているレベルではない。

手順

ライブラリー・ファイルが正しくリンクされているか検証するには、以下のようになります。

手順

1. データ・ソースのメッセージ・ファイルをオープンします。メッセージ・ファイルは、フェデレーテッド・サーバーがインストールされているディレクトリー中の、lib32 または lib64 サブディレクトリー中にあります。以下の表には、データ・ソース、メッセージ・ファイル、およびライブラリー・ファイルがリストされています。

表4. データ・ソース、メッセージ・ファイル、およびライブラリー・ファイル

データ・ソース	メッセージ・ファイル	ライブラリー・ファイル
Informix	djxlinkInformix.out	AIX - libdb2informix.a HP-UX - libdb2informix.so Linux - libdb2informix.so Solaris - libdb2informix.so
Microsoft SQL Server	djxlinkMssql.out	AIX - libdb2mssql3.a HP-UX - libdb2mssql3.so Linux - libdb2mssql3.so Solaris - libdb2mssql3.so
Sybase	djxlinkSybase.out	AIX - libdb2ctlib.a HP-UX - libdb2ctlib.so Linux - libdb2ctlib.so Solaris - libdb2ctlib.so
Teradata	djxlinkTeradata.out	AIX - libdb2teradata.a HP-UX - libdb2teradata.so Linux - libdb2teradata.so Solaris - libdb2teradata.so

2. 以下のようにメッセージ・ファイルの内容を評価します。

- リンクが成功している場合は、ライブラリー・ファイルがフェデレーションのインストール先のディレクトリー内に表示され、メッセージ・ファイルは成功メッセージを表示します。
- リンクが失敗している場合は、メッセージ・ファイルはエラー・メッセージを表示します。ライブラリー・ファイルとメッセージ・ファイルのどちらもフェ

デレションのインストール先のディレクトリー内に表示されない場合は、ライブラリー・ファイルをデータ・ソース・クライアント・ソフトウェアに手動でリンクしなければなりません。

ライブラリー・ファイルをデータ・ソース・クライアント・ソフトウェアに手動でリンクする

ライブラリー・ファイルがディレクトリー・パス内に表示されない場合、スクリプトを実行して、データ・ソース・クライアント・ソフトウェアにリンクしなければなりません。

始める前に

- データ・ソース・クライアント・ソフトウェアがフェデレーテッド・サーバーにインストールされ、構成されていなければなりません。
- デフォルトでは、スクリプトはすべてのメッセージを英語で発行します。別の言語でメッセージを発行するには、1つのフェデレーテッド・データベースを、英語以外の言語のコード・ページを使用するように構成しなければなりません。
- 必須のデータ・ソース環境変数を設定しなければなりません。

表 5. データ・ソース、スクリプト、および必須の環境変数

データ・ソース	スクリプト	必須の環境変数
Informix	djxlinkInformix	INFORMIXDIR
Microsoft SQL Server	djxlinkMssql	DJX_ODBC_LIBRARY_PATH
Sybase	djxlinkSybase	SYBASE、SYBASE_OCS
Teradata	djxlinkTeradata	COPLIB

手順

1. UNIX コマンド・プロンプトをオープンし、以下のコマンドを入力します。

```
cd /opt/IBM/db2/V9.5/bin
```

2. DB2 コマンド行から、以下のコマンドを入力します。

```
db2 disconnect all  
db2stop
```

3. アクセスするデータ・ソースごとにスクリプトを実行します。スクリプトの名前については、上記の表を参照してください。

4. 複数パーティション環境をインストールしている場合は、データベース・インスタンスごとにステップ 1 から 3 までを繰り返します。

5. データベース・インスタンスごとに、ディレクトリー *install_dir/instance* (*install_dir* は DB2 のインストール・ディレクトリー) に進み、以下のコマンドを入力します。

```
./db2iupdt instance_name
```

このコマンドは、インスタンスの構成を更新し、データ・ソースにアクセスできるようにします。

6. データベース・インスタンスごとに、ライブラリー・ファイルに対するファイル許可を表示します。データベース・インスタンスの所有者に、ファイルを読み取って実行する許可があるか検証します。

製品ライセンス・キーとポリシーの管理

IBM InfoSphere Federation Server がインストールされているコンピューターには、それぞれ製品ライセンス・キーを指定するライセンス・ファイルが保管されている必要があります。ライセンス・ポリシーは、フェデレーテッド・サーバーに接続を許可されるユーザーの数を制御し、モニターします。

このタスクについて

ライセンス・キーは、インストール・ソフトウェアのライセンス・ディレクトリーにあります。ライセンス・キーが登録されていない場合は、手動で登録する必要があります。

手順

1. ライセンス・ファイルは、製品インストール・ソフトウェアの license ディレクトリーで見つけてください。

isfs.lic

は、製品版のライセンス・ファイルです

isfs_t.lic

は、試用版のライセンス・ファイルです

isfs_d.lic

は、開発者用エディションのライセンス・ファイルです

2. ディレクトリーにライセンス・ファイルがない場合、ライセンス・キーを手動で登録しなければなりません。を参照してください。
3. フェデレーテッド・サーバーに接続を許可されるユーザーを制御およびモニターするためのライセンス・ポリシーを指定します。

DB2 pureScale フィーチャーのフェデレーションの構成

DB2 pureScale 環境でフェデレーテッド機能を使用するには、クライアント・ライブラリーを DB2 pureScale メンバーすべてにインストールする必要があります。

このタスクについて

DB2 pureScale フィーチャーと共にフェデレーテッド機能を使用する場合、適用される制約事項は以下の通りです。

- フェデレーション XML と 2 フェーズ・コミット機能は、DB2 pureScale インスタンスではサポートされません。
- リモート照会取り消しのために、FORCE APPLICATION コマンドを発行する場合は、シームレス自動クライアント・リルートが無効にする必要があります。
- 非シームレス自動クライアント・リルート (ACR) に対して、FORCE APPLICATION コマンドを発行して長時間実行のフェデレーテッド・ステートメントを取り消す場合には、SQLCODE -30108 (SQL30108N) が戻されても、そのステートメントを再実行しないでください。ステートメントを再実行すると、さらなる停止を引き起こす可能性があります。

手順

クライアント・ライブラリーがすべての DB2 pureScale メンバーにインストールされていることを確認してください。お勧めは、すべてのクライアント・ライブラリーを同じリリースにインストールする構成です。

フェデレーテッド・データベースの作成

データ・ソースにアクセスするようにフェデレーテッド・サーバーを構成する前に、フェデレーテッド・データベースを作成する必要があります。

始める前に

- データベースを作成するには、SYSADM または SYSCTRL 権限がなければなりません。
- フェデレーションが、フェデレーテッド・サーバーとして機能するサーバー上にインストールされている必要があります。

このタスクについて

データベース・インスタンスが複数パーティションの構成を使用する場合、データベースを作成する際に、db2nodes.cfg ファイルにリストされているすべてのパーティションに影響があります。CREATE DATABASE コマンドの発行元のデータベース・パーティションが、新規データベースのカatalog・パーティションになります。

手順

1. フェデレーテッド・データベースの作成時に指定するコード・セットと照合シーケンスを判別します。
2. DB2 コマンド行から、CREATE DATABASE コマンドを発行します。

例えば、ブラジルのテリトリーでコード・セット ISO8859-15 を使用する federated という名前のデータベースを作成するには、以下のコマンドを発行します。

```
CREATE DATABASE federated
USING CODESET ISO8859-15
TERRITORY BR
```

コード・セット、照合シーケンス、および各国語サポート

フェデレーテッド・データベースを作成するとき、コード・セット、テリトリー、および照合シーケンスを指定します。この情報により、データが保管される言語と、文字データがソートされる順序が制御されます。

コード・セット とは、特定の自然言語の文字にマップする固有のビット・パターンのセットです。IBM 製品は、コード・ページ という用語を、コード・セットの同義語として使用しています。テリトリー はロケールを識別し、指定されたコード・セットに地域固有の情報を指定します。これらのオプションを指定しない場合、データベースの作成に使用された DB2 クライアントの言語と照合シーケンスがデータベースで使用されます。

フェデレーテッド・データベースを作成する前に、指定するコード・セット値とテリトリー値を決定します。データベースを作成した後でこれらの値を変更することはできません。フェデレーテッド・データベースのコード・セットを選択するには、フェデレーテッド・データベースのアクセス先であるリモート・データ・ソースによって指定されたコード・セットを評価します。リモート・データ・ソースが使用するコード・セットに対応するフェデレーテッド・データベース用のコード・セットを選択します。フェデレーテッド・データベースが複数のデータ・ソースにアクセスする場合は、すべてのリモート・データ・ソースで指定されたコード・セットを評価します。データ・ソースが別のコード・セットまたは互換性のないコード・セットを使用する場合は、フェデレーテッド・サーバーのコード・セットとして Unicode を指定します。

多くのデータ・ソースでは、ラッパーが最初にデータ・ソースに接続したときに、ラッパーによって以下のタスクが実行されます。

1. フェデレーテッド・データベースのコード・ページとテリトリーが決定される。
2. データ・ソースがデータ・ソース・クライアント・ロケールをサポートしている場合は、コード・セットとテリトリーがそこにマップされる。
3. 環境変数が設定されるか、データ・ソース API が呼び出されてデータ・ソースにクライアント・ロケールの種類が通知されるか、コード・セット変換の実行が準備される。

コード・ページの変換では、データ・ソース・データベースのコード・ページと、フェデレーテッド・データベースのコード・ページとの間で文字データの変換が行われます。一部のデータ・ソースでは、コード・ページの変換が実行されます。コード・ページの変換が実行されない一部のデータ・ソースでは、ラッパーが変換を実行します。例えば、フェデレーテッド・データベースがコード・ページに 819、テリトリーに US を使用する場合は、これに相当する Oracle クライアント・ロケールは American_America.WE8ISO8859P1 です。Oracle ラッパーは、Oracle クライアントのロケール値に自動的に NLS_LANG 環境変数を設定します。すると、Oracle データベースからこのラッパーにデータが送信されたときに、Oracle データベースがそのデータをコード・セット American_America.WE8ISO8859P1 からコード・ページ 819 に変換します。データがラッパーから Oracle データベースに送信されると、Oracle サーバーまたはクライアントは、このデータをコード・ページ 819 から Oracle データベースが使用するコード・セットに変換します。

照合シーケンスはフェデレーテッド・サーバーがサポートし、データ・ソース・サーバーがサポートする言語に関連付けられます。フェデレーテッド・データベースに照合シーケンスを指定するには、CREATE DATABASE コマンドに COLLATE USING オプションを含めます。フェデレーテッド・データベースとデータ・ソースが同一の照合シーケンスを使用する場合は、CREATE SERVER ステートメントの発行時に COLLATING_SEQUENCE サーバー・オプションを「Y」に設定します。フェデレーテッド・データベースに指定する照合シーケンスは、文字のソートまたは文字比較を行う照会がどこで実行されるかに影響します。デフォルトで、フェデレーテッド・データベースは大/小文字が区別される照合シーケンスを使用しています。ただし、一部のデータ・ソースでは、デフォルトで大/小文字を区別しない照合シーケンスを使用しています。データ・ソースでは、照合シーケンスのカスタマイズも許可される場合と、デフォルトのコード・ページを設定するオプションを複数提供している場合があります。フェデレーテッド・データベースとデータ・ソース

の照合シーケンスが異なる場合、照会から予想外の結果が戻されることがあります。例えば、照会で文字ソートを行う場合、正しい結果が戻されますが、予想する順序にはなりません。照会で文字比較を行う場合、不正確な結果が戻されることがあります。

文字ベースの操作がどこで実行されるかによって、照会のパフォーマンスが影響を受けます。照合シーケンスが異なる場合、フェデレーテッド・サーバーは文字ソートと文字比較をローカルに実行して、ソートに整合性のある結果セットが確実に戻されるようにします。最適の結果を得るには、フェデレーテッド・データベースの照合シーケンスを、データ・ソースが使用するものと同じシーケンスに設定します。次に、可能であれば、照会オプティマイザーは文字ベースの操作をデータ・ソースにプッシュダウンし、フェデレーテッド・データベースではなくデータ・ソースが操作を実行するようにします。

フェデレーテッド・システムの Unicode の設定

リレーショナル・ラッパー、非リレーショナル・ラッパー、およびユーザー定義関数は、Unicode コード・ページ (UTF-8) のデータベース上で実行できます。Unicode コード・ページのデータベースは、プラットフォームに依存しないフェデレーテッド・サーバー環境を提供します。

フェデレーテッド・システムの Unicode サポート

すべてのリレーショナル・ラッパー、非リレーショナル・ラッパー、およびユーザー定義関数は、Unicode コード・ページ (UTF-8) のデータベース上で実行できます。

Unicode コード・ページのデータベースは、プラットフォームに依存しないフェデレーテッド・サーバー環境を提供します。データベースは、異なるデータ・ソース上のさまざまなコード・ページに保管されるデータを操作できます。

22 ページの図 4 で、ある会社は異なる国々に支店を持っています。各支店は、独自のコード・ページの独自のデータベースに顧客データを保管しています。Microsoft SQL Server データベースは、データをコード・ページ A で保管します。Oracle データベースは、データをコード・ページ B で保管します。コード・ページ A とコード・ページ B は異なるテリトリーにあります。異なるテリトリーからのデータを統合するために、会社はフェデレーテッド・データベースのコード・ページを Unicode に設定することができます。そうすれば、この会社では表を結合して、テリトリーに関係なく合計購入注文数を参照できるようになります。

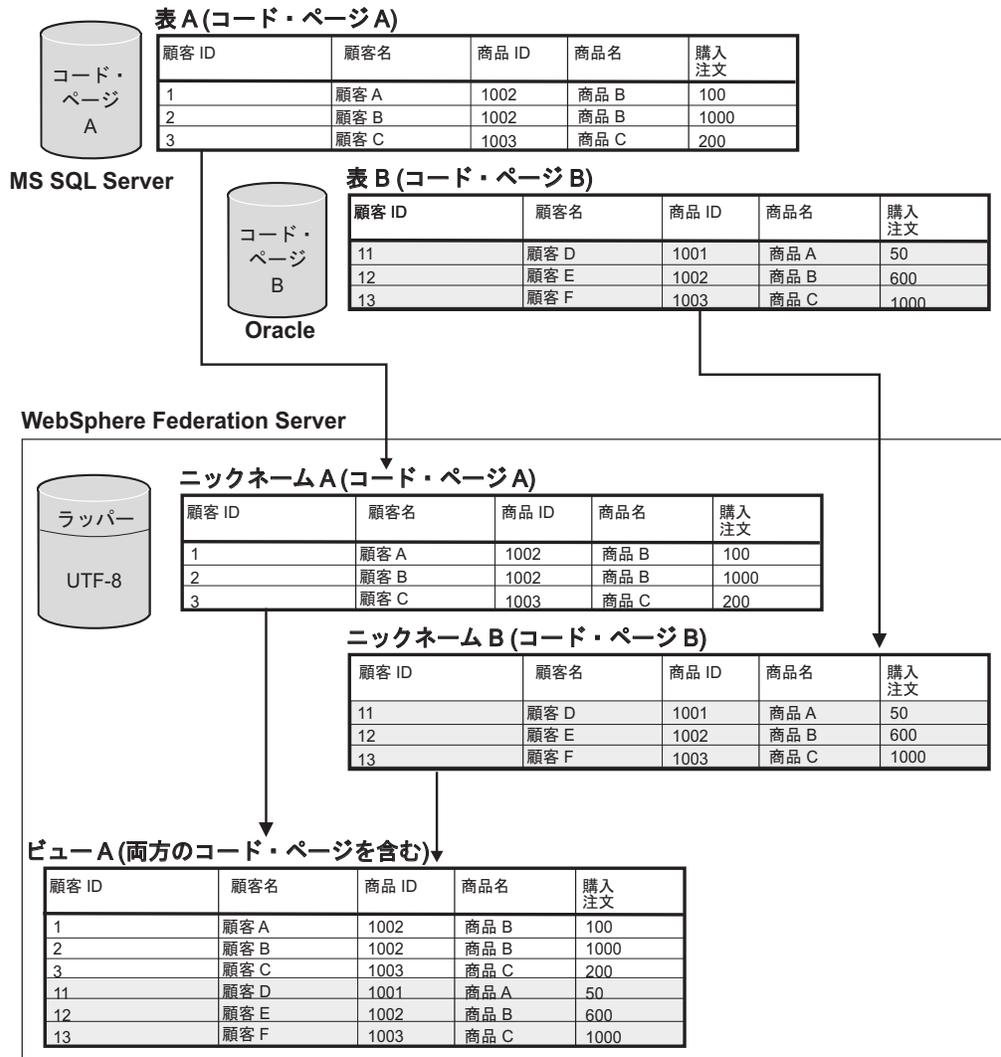


図 4. Unicode の例

Microsoft SQL Server データ・ソースおよび ODBC データ・ソースの Unicode サポートのためのクライアント・コード・ページの指定

Microsoft SQL Server データ・ソースおよび ODBC データ・ソースのコード・ページ変換が正しく行われるためには、クライアントのコード・ページがフェデレーテッド・データベースのコード・ページと異なる場合、クライアントのコード・ページを指定しなければなりません。

手順

クライアント・コード・ページを指定するには、CODEPAGE オプションにクライアント・コード・ページの値を設定した CREATE SERVER ステートメントを発行します。

クライアント・コード・ページは、データ・ソースのコード・ページです。クライアント・コード・ページの値は、コード・ページ番号です。

例:データ・ソースが Microsoft SQL Server で、フェデレーテッド・サーバーが Windows 上にあり、このオペレーティング・システムのデフォルト・システム・ロケールが日本語 (Shift-JIS) の場合は、CODEPAGE サーバー・オプションに 943 (Shift-JIS) または 1202 (UTF-16LE) を設定する必要があります。FEDSERVERW という名前の Microsoft SQL Server データ・ソースに対して 1202 コード・ページを指定するには、以下のステートメントを発行します。

```
CREATE SERVER FEDSERVERW TYPE MSSQLSERVER VERSION 2000 WRAPPER MSSQLODBC3
    OPTIONS(NODE 'SAMPLE', DBNAME 'TESTDB', CODEPAGE '1202');
```

例:データ・ソースが Microsoft SQL Server で、フェデレーテッド・サーバーが UNIX 上にあり、DataDirect Connect クライアントの IANAAppCodePage 設定値が 6 (Shift-JIS) の場合は、CODEPAGE サーバー・オプションに 943 (Shift-JIS) または 1208 (UTF-8) を設定する必要があります。FEDSERVERU という名前の Microsoft SQL Server データ・ソースに対して 1208 コード・ページを指定するには、以下のステートメントを発行します。

```
CREATE SERVER FEDSERVERU TYPE MSSQLSERVER VERSION 2000 WRAPPER MSSQLODBC3
    OPTIONS(NODE 'SAMPLE', DBNAME 'TESTDB', CODEPAGE '1208');
```

MSSQL および ODBC ラッパーの CODEPAGE オプションでサポートされる Unicode コード・ページ

有効なコード・ページ値には、DB2 Database for Linux, UNIX, and Windows がサポートするコード・ページ値に加えて、以下の表に示されているコード・ページ値があります。

表 6. MSSQL および ODBC ラッパーの CODEPAGE オプションでサポートされる Unicode コード・ページ

CODEPAGE オプションの値	説明
1200	Codepage1200 - UCS-2 (ビッグ・エンディアン)
1202	Codepage1202 - UCS-2 (リトル・エンディアン)
1208	Codepage1208 - UTF-8
1232	Codepage1232 - UTF-32 (ビッグ・エンディアン)
1234	Codepage1234 - UTF-32 (little-endian)

表構造ファイル・データ・ソースの Unicode サポートのためのファイル・コード・ページの指定

表構造ファイル・データ・ソースのコード・ページ変換が正しく行われるためには、ファイル・コード・ページがフェデレーテッド・データベースのコード・ページと異なる場合、ファイル・コード・ページを指定しなければなりません。

このタスクについて

有効な値には、DB2 Database for Linux, UNIX, and Windows がサポートするコード・ページ値があります。デフォルト値は、フェデレーテッド・データベースのコード・ページです。

制約事項

CODEPAGE オプションは、Unicode のフェデレーテッド・データベースでのみ使用できます。

手順

表構造ファイルのコード・ページを指定するには、CODEPAGE オプションに表構造ファイル内のデータのコード・ページ番号を設定した CREATE NICKNAME ステートメントを発行します。

例: DRUGDATA1.TXT という名前のファイル内のデータのコード・ページは、943 です。表構造ファイルのコード・ページを 943 として指定するには、以下の CREATE NICKNAME ステートメントを発行します。

```
CREATE NICKNAME DRUGDATA1(Dcode Integer NOT NULL, Drug CHAR(20),
    Manufactuer CHAR(20))
FOR SERVER biochem_lab
OPTIONS(FILE_PATH '/usr/pat/DRUGDATA1.TXT',CODEPAGE '943',
COLUMN_DELIMITER '.',
SORTED 'Y', KEY_COLUMN 'DCODE', VALIDATE_DATA_FILE 'Y');
```

表構造ファイル・データ・ソースの Unicode サポートのためのファイル・コード・ページの指定 - 例

次の例は、表構造ファイルのコード・ページを指定する方法を示しています。

DRUGDATA1.TXT という名前のファイル内のデータのコード・ページは、943 です。表構造ファイルのコード・ページを 943 として指定するには、以下の CREATE NICKNAME ステートメントを発行します。

```
CREATE NICKNAME DRUGDATA1(Dcode Integer NOT NULL, Drug CHAR(20),
    Manufactuer CHAR(20))
FOR SERVER biochem_lab
OPTIONS(FILE_PATH '/usr/pat/DRUGDATA1.TXT',CODEPAGE '943',
COLUMN_DELIMITER '.',
SORTED 'Y', KEY_COLUMN 'DCODE', VALIDATE_DATA_FILE 'Y');
```

リモート・コード・ポイント・サイズとフェデレーテッド・コード・ポイント・サイズが異なる場合のエラー

フェデレーテッド・データベースとリモート・データ・ソースの間でコード・ポイント・サイズが異なる場合、切り捨てられたデータが戻されたり、挿入障害あるいは更新障害が発生したりすることがあります。

リモート・データ・ソースからデータを選択した場合、文字ストリングを変換した結果、バイト数がニックネーム列のサイズより大きくなる場合は、データの切り捨てが行われます。切り捨てられたデータがぶら下がり文字で終了する場合、残りのバイトは空白で埋められます。また、変換後のデータ・サイズがリモート列のサイズ以下であれば、ニックネーム列のサイズより大きいデータを挿入または更新することも可能です。

フェデレーテッド・データベースのコード・ポイント・サイズがリモート・データ・ソースより小さい場合、データの挿入または更新は失敗する可能性があります。文字ストリングの変換の結果、バイト数がリモート・データ・ソース列のサイズより大きくなる場合は、挿入または更新は失敗します。

データ切り捨てあるいは打ち切り誤差が生じないようにするには、ニックネーム列のサイズまたはリモート表の列サイズを調整してください。

データ・ソースへアクセスするためのフェデレーテッド・サーバーの構成

フェデレーテッド・サーバーをインストールしてセットアップした後で、フェデレーテッド・データベースを作成してデータ・ソースへのアクセスを計画して構成します。

データ・ソースへのアクセスを構成するプロセスは、データ・ソースに関係なく同じです。このプロセスで異なる点は、各データ・ソースの構成タスクを完了するたびに適用する特定の設定値です。ここで説明するプロセスは一般的なものです。特定のデータ・ソースを構成するための完全な情報は、データ・ソースの詳細な構成情報を参照してください。

オブジェクトの命名方法と指定方法の検討

構成プロセス中は、オブジェクトの登録に DB2 SQL ステートメントを使用します。構成タスクを実行する前に、必ず、フェデレーテッド命名規則を理解しておき、指定するオブジェクトの大/小文字の区別に SQL ステートメント内での引用符の使用が与える影響を理解しておいてください。

ラッパーの登録

フェデレーション・インストール・プログラムを使用してデータ・ソースのサポートをインストールした後で、対応するラッパーを登録する必要があります。ラッパーとは、フェデレーテッド・サーバーがデータ・ソースと通信して、そこからデータを取り出すために使用するライブラリー・ファイルの集合です。アクセス先のデータ・ソースのタイプごとに、ラッパーを 1 つ登録します。例えば、DB2 for Linux, UNIX, and Windows で 1 つの表に、DB2 for iSeries® で 1 つの表に、Teradata で 1 つの表にアクセスするには、2 つのラッパーを登録します。つまり、DB2 データ・ソースには DRDA® ラッパーを、Teradata データ・ソースには Teradata ラッパーを登録します。

計画の一部として、デフォルトのラッパー名を使用するかどうか、ラッパーに別の名前を割り当てるかどうかを決定し、構成しているデータ・ソースごとに使用できるラッパー・オプションを検討します。各データ・ソースには、1 つ以上の必須ラッパー・オプションを設定しなければなりません。

サーバー定義の登録

特定のデータ・ソース・オブジェクトにアクセスできるようにするには、その前に 1 つ以上のサーバー定義を登録します。リレーショナル・データ・ソースの場合、サーバー定義はリモート・データベース、データベース・パーティション、またはノードを表します。非リレーショナル・データ・ソースの場合、サーバー定義はしばしば外部データ・オブジェクトの他のタイプにマップします。各データ・ソースには、サーバー定義を登録する際に指定が必要な必須パラメーターとオプション・パラメーターがあります。

計画の一部として、構成している特定のデータ・ソースに使用できるサーバー・オプションを検討します。各データ・ソースには、1 つ以上の必須サーバー・オプションを設定しなければなりません。

ユーザー・マッピングの登録

リモート・データ・ソースがユーザー認証を必要としている場合で、ユーザーのリモート・ユーザー ID とリモート・パスワードが、フェデレーテッド・データベースに接続する際にユーザーが使用するものと異なる場合は、ユーザー・マッピングを定義します。ユーザー・マッピングは、フェデレーション・サーバーの許可 ID とデータ・ソースのユーザー ID およびパスワードの関連性です。デフォルトで、ユーザー・マッピングはフェデレーテッド・サーバーのカタログに保管されます。

計画の一部として、LDAP サーバーやファイルなどの外部リポジトリにユーザー・マッピング情報を保管するかどうかを決定します。外部リポジトリを使用するには、フェデレーテッド・サーバーに対してこのリポジトリへのインターフェースを提供するプラグインを作成する必要があります。

データ・ソース統計の更新

アクセスしようとするリレーショナル・データ・ソースごとに、DB2 の RUNSTATS コマンドに相当するコマンドを使用して、リモート・データ・ソース側で統計を更新します。すると、ニックネームを作成するときに、フェデレーテッド・データベースのシステム・カタログに、最新の統計情報が追加されます。その後で、データ・ソースで照会を実行すると、照会オプティマイザーがこの情報を使用して、照会を実行するのに最も効率の良い方法を判別します。

ニックネームの作成後に、データ・ソース側の統計が変更される場合があります。リレーショナル・データ・ソースの統計が変更されたときは、SYSPROC.NNSTAT ストアード・プロシージャを使用して、システム・カタログの統計情報を更新します。非リレーショナル・データ・ソースの統計が変更されたときは、非リレーショナル・データ・ソースにより提供されるツールを使用するか、SYSTAT カタログ・ビューで統計を手動で更新します。

ニックネームの登録

アクセスするリレーショナル・データ・ソース・オブジェクトごとに、ニックネームを作成します。一部の非リレーショナルデータ・ソースでは、ニックネームを登録する際に入力列と出力列の固定リストを定義します。指定する各列は、データ・ソース・オブジェクト内の特定のフィールド、列、またはエレメントにマップされます。

計画の一部として、構成しているデータ・ソースに使用できるニックネームと列のオプションを検討します。一部のデータ・ソースでは、設定が必要な必須のニックネーム・オプションと列のオプションがあります。

追加的構成タスクの実行

データ・ソースの操作方法に応じて、以下の追加的構成タスクを実行します。

索引の仕様を作成する

索引がないオブジェクトには、索引の仕様を定義できます。例えば、表に新

しい索引ができた時に、またはデータ・ソース・オブジェクト (ビューなど) に索引がない場合に、索引の仕様を作成します。

代替データ・タイプ・マッピングを定義する

フェデレーテッド・システムでは、データ・ソースのデータ・タイプとフェデレーテッド・データベースのデータ・タイプとの間にデフォルトのマッピングがあります。リレーショナル・データ・ソースの場合は、代替データ・タイプ・マッピングを定義することができます。例えば、特定のサーバーに存在するすべてのデータ・ソース・オブジェクトのタイプ・マッピングを変更したり、特定のデータ・ソース・オブジェクト、データ・ソース・タイプ、またはデータ・ソース・オブジェクトとタイプに対するタイプ・マッピングを変更したりすることができます。

代替関数マッピングを定義する

フェデレーテッド・システムでは、組み込みデータ・ソース関数と組み込みフェデレーテッド・データベース関数の間でデフォルトの関数マッピングがあります。リレーショナル・データ・ソースの場合は、代替関数マッピングを定義することができます。代替関数マッピングは、例えば新しい組み込み関数を使用する場合や、データ・ソースでは使用できるがフェデレーテッド・データベース側に対応するマップ先関数がないユーザー定義関数を使用する場合に定義することができます。

大/小文字の区別と引用符の正しい使用方法

DB2 の SQL ステートメントでオプションの値とオブジェクトを指定する際、どのような場合に引用符が必要か、どのタイプを使用するか、および大/小文字の区別にどう影響するかについて知っておく必要があります。

オブジェクトを最初に作成する際の命名方法は、オブジェクト名の文字の大/小文字の別、およびコマンドでオブジェクト名とオプション値を指定する方法に影響します。例えば、ニックネームの作成時に名前を二重引用符で囲まなかった場合、オブジェクトの名前に使用したのが大文字か小文字かに関係なく、システム・カタログはこのニックネームを大文字で保管します。ニックネームの作成時に二重引用符を使用した場合、システム・カタログはオブジェクト名の文字を、指定したとおりの大/小文字で正確に保管します。次に、オブジェクト名をオプション値として使用するときに、その大/小文字を正確に指定しなければなりません。例えば、スクリプト、Web サービス、および XML ラッパーがサポートする FOREIGN_KEY 列オプションでは、オプションの値として外部キー列のニックネームを指定する必要があります。オプション値を入力する際は、フェデレーテッド・サーバー・カタログがニックネームの保管に使用するものと同じ大/小文字を使用する必要があります。

次の表に、DB2 の SQL ステートメントでオプション値とオブジェクトを指定する際の大/小文字と引用符の正しい使用方法を説明します。

表 7. 大/小文字と引用符の正しい使用法

識別子	大/小文字と引用符の使用法	例
オプション値	オプション値が必要な大/小文字を使用して、オプション値を単一引用符で囲みます。	<p>次のステートメントは、 remote_schema.remote_table (すべて小文字) という名前のデータ・ソース表を作成します。</p> <pre>CREATE TABLE newton.my_nick (c1 int) OPTIONS (remote_server 'MY_SERVER' remote_schema 'remote_schema', remote_tabname 'remote_table');</pre> <p>次のステートメントは、 REMOTE_SCHEMA.REMOTE_TABLE (すべて大文字) という名前のデータ・ソース表を作成します。</p> <pre>CREATE TABLE newton.my_nick (c1 int) OPTIONS (remote_server 'MY_SERVER' remote_schema 'REMOTE_SCHEMA', remote_tabname 'REMOTE_TABLE');</pre>
小文字のみで構成されるオブジェクト	すべて小文字を使用し、識別子を二重引用符で囲みます。	<p>次のステートメントは、 infx_user.remote_table (すべて小文字) という名前のデータ・ソース表にニックネームを作成します。</p> <pre>CREATE NICKNAME my_nick FOR infx_server. "infx_user"."remote_table";</pre> <p>注: Informix および Teradata などのような一部のデータ・ソースでは、デフォルトで小文字の名前を使用します。</p>
大文字、数字、および下線文字 () のみで構成されるオブジェクト	<p>次の 2 つの選択肢があります。</p> <ul style="list-style-type: none"> • すべて大文字を使用し、識別子を二重引用符で囲む。 • 大/小文字は任意のものを使用し、識別子を二重引用符で囲まない。 	<p>次のステートメントでは、いずれもニックネーム MY_NICK (すべて大文字) が作成されます。</p> <pre>CREATE NICKNAME my_nick FOR infx_server. "infx_user"."remote_table"; CREATE NICKNAME "MY_NICK" FOR infx_server. "infx_user"."remote_table";</pre>

データ・ソース許可 ID およびパスワードの場合は、サーバー・オプション FOLD_ID および FOLD_PW を使用して、この ID とパスワードを正しい大/小文字に変換することもできます。

UNIX オペレーティング・システムのコマンド・プロンプトからの場合

フェデレーテッド・サーバーの UNIX コマンド・プロンプトで大/小文字を区別する値を引用符で囲む場合は、引用符が必ず正確に構文解析されるようにしなければなりません。

二重引用符を含み、単一引用符を含まない SQL ステートメント

SQL ステートメントに二重引用符が含まれるが、単一引用符が含まれない場合、ステートメント全体を単一引用符で囲みます。

例えば、次の SQL ステートメントを発行する場合、

```
CREATE NICKNAME my_nickname FOR my_server."owner"."my_table"
```

UNIX コマンド・プロンプトで次のようにテキストを入力します。

```
db2 'CREATE NICKNAME my_nickname FOR my_server."owner"."my_table"'
```

単一引用符を含み、二重引用符を含まない SQL ステートメント

SQL ステートメントに単一引用符が含まれるが、二重引用符が含まれない場合、ステートメント全体を二重引用符で囲みます。

例えば、次の SQL ステートメントを発行する場合、

```
CREATE USER MAPPING FOR USER SERVER my_server  
  OPTIONS(REMOTE_AUTHID 'my_id', REMOTE_PASSWORD 'my_password')
```

UNIX コマンド・プロンプトで次のようにテキストを入力します。

```
db2 "CREATE USER MAPPING FOR USER SERVER my_server  
  OPTIONS(REMOTE_AUTHID 'my_id', REMOTE_PASSWORD 'my_password')"
```

二重引用符と単一引用符の両方を含む SQL ステートメント

SQL ステートメントに単一引用符と二重引用符の両方が含まれる場合は、以下のようにします。

- ステートメント全体を二重引用符で囲みます。
- ステートメント内の各二重引用符の前に円記号を付けます。

たとえば、次の SQL ステートメントを発行するとします。

```
CREATE USER MAPPING FOR "local_id" SERVER my_server  
  OPTIONS(REMOTE_AUTHID 'my_id', REMOTE_PASSWORD 'my_password')
```

UNIX コマンド・プロンプトで次のようにテキストを入力します。

```
db2 "CREATE USER MAPPING FOR \"local_id\" SERVER my_server  
  OPTIONS(REMOTE_AUTHID 'my_id', REMOTE_PASSWORD 'my_password')"
```

上述の例では、SQL ステートメントを UNIX コマンド・プロンプトから入力し、ステートメントを DB2 コマンドへ `-f` オプションを使用せずに渡すと仮定しています。SQL ステートメントを `-f` オプション付きの DB2 コマンドを使用してファイルから入力するには、それぞれの例で最初に示されているようにしてステートメントを入力します。

Windows オペレーティング・システムのコマンド・プロンプトからの場合

フェデレーテッド・サーバーの Microsoft Windows コマンド・プロンプトからコマンドを入力するときに大/小文字が区別される値を保持するには、各二重引用符の前に円記号を置きます。例えば、Microsoft SQL サーバーの表 `weekly_salary` のために

nick1 というニックネームを作成するとします。この表は、*NORBASE* データベースにあります。ローカル・スキーマは、*my_schema* です。

フェデレーテッド・サーバーの Windows コマンド・プロンプトで次のように入力します。

```
db2 CREATE NICKNAME nick1
      FOR NORBASE.%"my_schema%" .%"weekly_salary%"
```

DB2 コマンド行から、またはアプリケーションからの場合

DB2 コマンド行またはアプリケーションから値を指定する場合、値を正しい引用符で囲むことによって大/小文字が区別される値を保持することができます。

例えば、ユーザー ID *local_id* のためのユーザー・マッピングを作成するとします。リモート・ユーザー ID が *my_id* で、リモート・パスワードが *my_password* です。これら 3 つの値をみな小文字のまま保持したいとします。DB2 コマンド・プロンプトで次のように入力します。

```
CREATE USER MAPPING FOR "local_id" SERVER my_server
      OPTIONS(REMOTE_AUTHID 'my_id', REMOTE_PASSWORD 'my_password')
```

データ・ソースへアクセスするための複数のフェデレーテッド・サーバーの構成

フェデレーテッド・システムを複数のフェデレーテッド・サーバーで構成することができます。各フェデレーテッド・サーバーを個別に構成するのではなく、IBM Data Studio を使用してフェデレーテッド・サーバーを構成することによって、時間を節約できます。

始める前に

- フェデレーションが、フェデレーテッド・サーバーとして機能するサーバー上にインストールされている必要があります。
- フェデレーテッド・データベースはフェデレーテッド・サーバー上に存在している必要があります。

このタスクについて

サポートされている DB2 データ・ソースおよび Oracle データ・ソースの場合、Data Studio の「管理エクスプローラー (Administration Explorer)」を使用して、アクセスするデータ・ソースに対するフェデレーテッド・オブジェクトを構成できます。最初のサーバーを構成した後、これらのステートメントを再利用または変更し、追加のフェデレーテッド・サーバーをすぐに構成できます。

Data Studio の「SQL and XQuery エディター」を使用すると、任意のデータ・ソースに対して複数のサーバーを構成することも可能です。

手順

以下のいずれかの方法を使用して、複数のフェデレーテッド・サーバーを構成します。

- サポートされている DB2 データ・ソースおよび Oracle データ・ソースの場合、複数のフェデレーテッド・オブジェクトの作成と管理の指示に従ってください。

- その他のデータ・ソースの場合は、フェデレーテッド オブジェクトの作成: SQL および XQuery エディターの指示に従ってください。

第 2 章 データ・ソースの構成

リレーショナルおよび非リレーショナルのデータ・ソースにアクセスするように、フェデレーテッド・サーバーを構成できます。

BioRS データ・ソースへのアクセスの構成

フェデレーテッド・システムを使用すると、BioRS データ・バンクにあるデータと他のソースからの情報を統合させることができます。

このタスクについて

BioRS データ・ソースにアクセスするようにフェデレーテッド・サーバーを構成するには、アクセスするデータ・ソースとオブジェクトに関する情報をフェデレーテッド・サーバーに提供します。フェデレーテッド・サーバー構成後は、照会を作成し、カスタム関数を使用して BioRS データ・ソースにアクセスできます。

BioRS ラッパー

BioRS は、複数のデータ・ソースから情報を検索するのに使用できる照会および検索システムです。BioRS ラッパーは、BioRS の Web ベースのインターフェースと同じ API を使用して情報を照会します。

BioRS は、ドイツ Biomax 社が開発した照会および検索システムです。BioRS を使用すると、フラット・ファイルやリレーショナル・データベースを含む、複数のデータ・ソースから情報を検索できます。通常 SwissProt や GenBank のような公用データは、フラット・ファイルとして BioRS システムにダウンロードします。BioRS は公用データ・ソースと専用データ・ソース (組織独自で保守する専用データベースなど) を共通環境に統合できます。

BioRS システムに統合されたデータ・ソースは、データ・バンクと呼ばれます。各データ・バンクの項目に含まれるエレメントは、まとめてスキーマと呼ばれます。索引付けされているデータ・バンクのエレメントは、BIORS.CONTAINS、BIORS.CONTAINS_GE、および BIOR.S.CONTAINS_LE 関数で使用できます。BioRS 関数は SELECT ステートメントの WHERE 文節で指定されます。索引付けされていないエレメントは、SELECT リストおよび WHERE 文節内の他の述部で参照できます。索引付けされていないエレメントは、フェデレーテッド・サーバーによって処理されます。

データ・バンクの項目同士を関連付け、BioRS システムでデータ・バンクを結合することも可能です。

BioRS データ・バンクには、親子関係を指定できます (ネスト可能)。親子関係では、子データ・バンクは PARENT と呼ばれる Reference データ・タイプ・エレメントを持ちます。PARENT エレメントは親データ・バンクの _ID_ エレメントを参照します。ネスト化されたデータ・バンクとネスト化されていないデータ・バンクが収容するデータの違いは、この定義済み PARENT エレメントのみです。

BioRS は BioRS データ・バンクのデータを照会できる Web ベースのインターフェースを提供します。BioRS ラッパーは、BioRS の Web ベースのインターフェースと同じアプリケーション・プログラミング・インターフェース (API) を使用して情報を照会します。

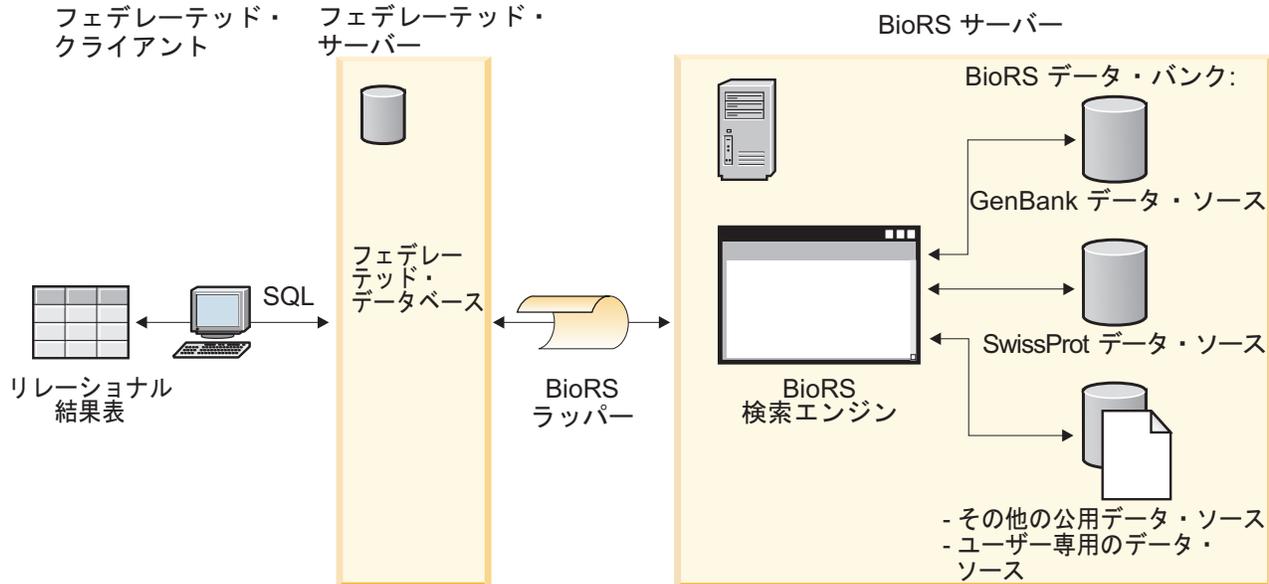


図 5. BioRS ラッパーが機能する構造

ユーザーまたはアプリケーションは、クライアントから SQL ステートメントを使用した照会をサブミットします。照会は BioRS ラッパーがインストールされたフェデレーテッド・システムに送信されます。照会の構成の仕方によっては、フェデレーテッド・サーバーと BioRS サーバーの両方が照会の処理に使用される場合があります。BioRS サーバーは、フェデレーテッド・システムとは異なるコンピュータ上にある場合もあります。ただし、各照会ごとに、フェデレーテッド・システムから BioRS サーバーに認証情報を提供する必要があります。認証情報は、ユーザー ID とパスワードの組み合わせか、非認証標識 (通常はゲスト・アカウント) のいずれかです。

BioRS 製品の詳細については、Biomax の Web サイト <http://www.biomax.com> を参照してください。

フェデレーテッド・サーバーへの BioRS データ・ソースの追加

BioRS データ・ソースにアクセスするようにフェデレーテッド・サーバーを構成するには、アクセスするデータ・ソースとオブジェクトに関する情報をフェデレーテッド・サーバーに提供します。

始める前に

- フェデレーションが、フェデレーテッド・サーバーとして機能するサーバー上にインストールされている必要があります。
- データベースがフェデレーテッド・サーバー上に存在している必要があります。

手順

1. BioRS ラッパーのカスタム関数を登録します。
2. BioRS ラッパーを登録します。
3. BioRS サーバ一定義を登録します。
4. オプション: BioRS ユーザー・マッピングを作成します。
5. BioRS データ・バンクのニックネームを登録します。

BioRS ラッパーのカスタム関数の登録

BioRS ラッパーを登録する前に、BioRS カスタム関数を登録する必要があります。BioRS カスタム関数は、BioRS ラッパーとともに使用され、BioRS 照会エンジンに述部をプッシュダウンします。

このタスクについて

BioRS ラッパーがインストールされている各フェデレーテッド・データベース・インスタンスにカスタム関数をすべて登録する必要があります。

BioRS ラッパーのカスタム関数はすべて、`biors` というスキーマ名を使用して登録する必要があります。

手順

BioRS カスタム関数ごとに、`CREATE FUNCTION` ステートメントを発行してください。カスタム関数ごとに、2 つのデータ・タイプを指定する必要があります。指定できる最初のデータ・タイプは、索引付けされた列です。指定できる 2 番目のデータ・タイプは、検索語です。`CREATE FUNCTION` ステートメントには、`AS TEMPLATE`、`DETERMINISTIC`、および `NO EXTERNAL ACTION` キーワードを含める必要があります。

`BIORS.CONTAINS` 関数の構文例は以下のとおりです。

```
CREATE FUNCTION biors.contains (column_data_type, search_term_data_type)
  RETURNS INTEGER AS TEMPLATE
  DETERMINISTIC NO EXTERNAL ACTION;
```

次のタスク

ヒント: BioRS カスタム関数を登録するには、サンプル・ファイル `create_function_mappings.ddl` を使用します。このサンプル・ファイルはフェデレーテッド・サーバーの `sqllib/samples/lifesci/biors` ディレクトリにあります。サンプル・ファイルには、考えられるデータ・タイプの組み合わせごとに `CREATE FUNCTION` ステートメントが含まれています。カスタム関数を登録するには、`create_function_mappings.ddl` ファイルを編集して、カスタム関数ごとに索引列および検索語のデータ・タイプを指定します。その後、BioRS ラッパーがインストールされている各フェデレーテッド・データベース・インスタンスで、`create_function_mappings.ddl` ファイルを実行する必要があります。

BioRS ラッパーのカスタム関数:

BioRS ラッパーとともに使用されるカスタム関数についての説明と例

フェデレーテッド環境は 2 つの照会エンジンを使用します。BioRS ラッパーの場合、フェデレーテッド・データベース照会エンジンと BioRS 照会エンジンがそれに当たります。BioRS カスタム関数を使用して、述部を BioRS エンジンにプッシュダウンすることを指定できます。

BioRS ラッパーのカスタム関数は以下のとおりです。

- BORS.CONTAINS
- BORS.CONTAINS_LE
- BORS.CONTAINS_GE
- BORS.SEARCH_TERM

BioRS カスタム関数を登録するには、CREATE FUNCTION ステートメントを使用します。

以下の表では、4 つの BioRS カスタム関数と、関数を登録する際に指定できる有効なデータ・タイプの例がリストされています。

表 8. BioRS ラッパーのカスタム関数

関数	説明
BIORS.CONTAINS (VARCHAR(), VARCHAR())	索引付けされている列を検索して、指定する値と等しい (BioRS 照会セマンティクスに従って) 値を探します。最初の引数は索引付けされている列への参照でなければならず、2 番目の引数は指定する検索値になります。
BIORS.CONTAINS (VARCHAR(), CHAR())	
BIORS.CONTAINS (VARCHAR(), DATE)	
BIORS.CONTAINS (VARCHAR(), TIMESTAMP)	
BIORS.CONTAINS_LE (VARCHAR(), VARCHAR())	索引付けされている列を検索して、指定する値以下の (BioRS 照会セマンティクスに従って) 値を探します。最初の引数は索引付けされている列への参照でなければならず、2 番目の引数は指定する検索値になります。
BIORS.CONTAINS_LE (VARCHAR(), SMALLINT)	
BIORS.CONTAINS_LE (VARCHAR(), BIGINT)	
BIORS.CONTAINS_LE (VARCHAR(), DECIMAL)	
BIORS.CONTAINS_LE (VARCHAR(), DOUBLE)	
BIORS.CONTAINS_LE (VARCHAR(), REAL)	
BIORS.CONTAINS_GE (CHAR(), CHAR())	索引付けされている列を検索して、指定する値以上の (BioRS 照会セマンティクスに従って) 値を探します。最初の引数は索引付けされている列への参照でなければならず、2 番目の引数は指定する検索値になります。
BIORS.CONTAINS_GE (CHAR(), DATE)	
BIORS.CONTAINS_GE (CHAR(), TIMESTAMP)	
BIORS.CONTAINS_GE (CHAR(), INTEGER)	
BIORS.CONTAINS_GE (CHAR(), SMALLINT)	
BIORS.CONTAINS_GE (CLOB(), DATE)	
BIORS.SEARCH_TERM (VARCHAR(), VARCHAR())	BioRS 照会式を BioRS 検索エンジンに渡します。
BIORS.SEARCH_TERM (VARCHAR(), CHAR())	
BIORS.SEARCH_TERM (CHAR(), VARCHAR())	
BIORS.SEARCH_TERM (CHAR(), CHAR())	

BioRS ラッパーの登録

BioRS データ・ソースにアクセスするためには、ラッパーを登録する必要があります。フェデレーテッド・サーバーは、ラッパーを使用してデータ・ソースとの通信やデータ・ソースからのデータの取得を行います。ラッパーは、ライブラリー・ファイルのセットとしてインプリメントされます。

このタスクについて

プロキシ・サーバーを使用して BioRS ファイルにアクセスする場合は、ラッパーの登録時にプロキシ情報をオプションとして指定する必要があります。デフォルトでは、BioRS ファイルに対する照会ではラッパー・オプションが使用されます。ただし、サーバー定義の登録時にプロキシ・オプションも指定する場合は、サーバー・オプションが優先されます。

手順

コマンド行から CREATE WRAPPER ステートメントを発行します。

例:

```
CREATE WRAPPER wrapper_name  
LIBRARY library_name;
```

BioRS 文書へのアクセスにプロキシ・サーバーを使用する場合、発行するステートメントは以下のとおりです。

```
CREATE WRAPPER wrapper_name  
LIBRARY library_name  
OPTIONS (PROXY_TYPE 'type',  
PROXY_SERVER_NAME 'server_name',  
PROXY_SERVER_PORT 'port_number');
```

CREATE WRAPPER ステートメントには LIBRARY パラメーターを指定する必要があります。指定するラッパー・ライブラリー・ファイルの名前は、フェデレーテッド・サーバーのオペレーティング・システムによって異なります。CREATE WRAPPER ラッパー・ステートメントで指定する正しいライブラリー名については、『BioRS ラッパー・ライブラリー・ファイル』のリストを参照してください。

BioRS ラッパー・ライブラリー・ファイル:

BioRS ラッパー・ライブラリー・ファイルは、フェデレーテッド・サーバーのインストール時にフェデレーテッド・サーバーに追加されます。

フェデレーテッド・サーバーのインストール時に、デフォルトのディレクトリー・パスに 3 つのライブラリー・ファイルが追加されます。例えば、フェデレーテッド・サーバーが AIX® 上で実行されている場合、ディレクトリー・パスには、libdb2lsbors.a、libdb2lsborsF.a、および libdb2lsborsU.a が追加されます。デフォルトのラッパー・ライブラリー・ファイルは libdb2lsbors.a です。他のラッパー・ライブラリー・ファイルは特定のラッパー・オプションで使用されます。

CREATE WRAPPER ステートメントに LIBRARY パラメーターを含め、デフォルトのラッパー・ライブラリー・ファイルの名前を指定する必要があります。

以下の表に、デフォルトのディレクトリー・パスとデフォルトのラッパー・ライブラリー・ファイルの名前をリストします。

表 9. BioRS ラッパーのライブラリーのロケーションおよびファイル名

オペレーティング・システム	ディレクトリー・パス	ラッパー・ライブラリー・ファイル名
AIX	/usr/opt/<install_path>/lib32/ /usr/opt/<install_path>/lib64/	libdb2lsbiors.a
Linux	/opt/IBM/db2/<install_path>/lib32 /opt/IBM/db2/<install_path>/lib64	libdb2lsbiors.so
Solaris	/opt/IBM/db2/<install_path>/lib32 /opt/IBM/db2/<install_path>/lib64	libdb2lsbiors.so
Windows	%DB2PATH%\%bin	db2lsbiors.dll

<install_path> は、UNIX または Linux 上でフェデレーテッド・サーバーがインストールされているディレクトリー・パスです。

%DB2PATH% は、Windows 上でフェデレーテッド・サーバーがインストールされているディレクトリー・パスを指定するのに使用する環境変数です。デフォルトの Windows ディレクトリー・パスは C:\Program Files\IBM\SQLLIB です。

CREATE WRAPPER ステートメント - BioRS ラッパーの例:

BioRS ラッパーを登録するには、CREATE WRAPPER ステートメントを使用します。例では、プロキシー・サーバーを介してまたは介さずに BioRS ドキュメントにアクセスするために必要なオプションを示します。

ラッパーの登録

BioRS 文書へのアクセスにプロキシー・サーバーを使用していない場合は、以下のステートメントを発行してラッパーを登録します。

```
CREATE WRAPPER biors_wrapper LIBRARY 'libdb2lsbiors.a';
```

biors_wrapper

BioRS ラッパーに割り当てる名前。重複するラッパー名は使用できません。

LIBRARY '*libdb2lsbiors.a*'

AIX オペレーティング・システムを使用するフェデレーテッド・サーバー用のラッパー・ライブラリー・ファイルの名前。

HTTP プロキシー・サーバーのラッパーの登録

ラッパーを登録して、HTTP プロキシー・サーバーを指定するには、以下のステートメントを使用します。

```
CREATE WRAPPER biors_proxy LIBRARY 'libdb2lsbiors.a'
  OPTIONS (PROXY_TYPE 'HTTP',
           PROXY_SERVER_NAME 'proxy.mysite.com',
           PROXY_SERVER_PORT '81');
```

PROXY_TYPE '*HTTP*'

フェデレーテッド・サーバーがファイアウォールで保護された状態でインターネットにアクセスする際に使用する、プロキシー・タイプを指定します。有効値は 'NONE'、'HTTP'、または 'SOCKS' です。

PROXY_SERVER_NAME 'proxy.mysite.com'

プロキシー・サーバー名または IP アドレスを指定します。 PROXY_TYPE サーバー・オプションの値が 'HTTP' または 'SOCKS' の場合、このオプションは必須です。

PROXY_SERVER_PORT '81'

プロキシー・サーバーのポート番号を指定します。 PROXY_TYPE サーバー・オプションの値が 'HTTP' または 'SOCKS' の場合、このオプションは必須です。

SOCKS プロキシー・サーバーのラッパーの登録

ラッパーを登録して、認証情報のない SOCKS プロキシー・サーバーを指定するには、以下のステートメントを使用します。

```
CREATE WRAPPER biors_wrapper LIBRARY 'libdb2lsbiors.so'  
  OPTIONS (PROXY_TYPE 'SOCKS',  
          PROXY_SERVER_NAME 'proxy_socks',  
          PROXY_SERVER_PORT '1081');
```

LIBRARY 'libdb2lsbiors.so'

Linux および Solaris オペレーティング・システムを使用するフェデレーテッド・サーバーに対応するラッパー・ライブラリー・ファイルの名前。

PROXY_TYPE 'SOCKS'

フェデレーテッド・サーバーがファイアウォールで保護された状態でインターネットにアクセスする際に使用する、プロキシー・タイプを指定します。有効値は 'NONE'、'HTTP'、または 'SOCKS' です。

PROXY_SERVER_NAME 'proxy_socks'

プロキシー・サーバー名または IP アドレスを指定します。 PROXY_TYPE サーバー・オプションの値が 'HTTP' または 'SOCKS' の場合、このオプションは必須です。

PROXY_SERVER_PORT '1081'

プロキシー・サーバーのポート番号を指定します。 PROXY_TYPE サーバー・オプションの値が 'HTTP' または 'SOCKS' の場合、このオプションは必須です。

BioRS データ・ソースのサーバー定義の登録

アクセスする各 BioRS サーバーをフェデレーテッド・データベースに登録する必要があります。

このタスクについて

プロキシー・サーバーを使用して BioRS ファイルにアクセスする場合は、サーバー定義を登録する際、プロキシー情報をオプションとして指定できます。プロキシー情報をサーバー定義で指定する場合、サーバー・オプションは、ラッパーの登録時に指定したオプションより優先されます。

手順

コマンド行から CREATE SERVER ステートメントを発行します。

例:

```
CREATE SERVER server_definition_name
VERSION version_number
WRAPPER wrapper_name
OPTIONS (NODE 'node_name');
```

プロキシ・サーバーを使用して BioRS ファイルにアクセスする場合は、BioRS ラッパーまたはサーバー定義の登録時にいくつかのオプションを指定する必要があります。BioRS ラッパーの登録時にプロキシ・サーバー情報を指定するには、以下のステートメントを発行します。

```
CREATE SERVER server_definition_name
VERSION version_number
WRAPPER wrapper_name
OPTIONS (PROXY_TYPE 'type',
PROXY_SERVER_NAME 'server_name',
PROXY_SERVER_PORT 'port_number');
```

サーバー定義の登録時に、CREATE SERVER ステートメントにサーバー・オプションを指定します。サーバー・オプションには、必須のオプションと任意指定のオプションとがあります。NODE サーバー・オプションは必須です。

次のタスク

サーバー定義を登録した後、サーバー・オプションを追加またはドロップするには、ALTER SERVER ステートメントを使用します。

CREATE SERVER ステートメント - BioRS ラッパーの例:

BioRS ラッパーにサーバー定義を登録するには、CREATE SERVER ステートメントを使用します。例には、必須パラメーター、オプション・パラメーター、および指定可能な追加サーバー・オプションが示されています。

必須パラメーターを使用したサーバー定義の登録

次に、CREATE SERVER ステートメントを実行して BioRS ラッパーのサーバー定義を登録する方法の例を示します。

```
CREATE SERVER biors_server VERSION '5.2' WRAPPER biors_wrapper
OPTIONS(NODE 'biors_myco.com');
```

biors_server

BioRS サーバーに割り当てる名前。重複するサーバー定義名は使用できません。

VERSION '5.2'

これは、アクセスしたい BioRS サーバーのバージョンです。サポートされている BioRS バージョンは 5.0.14 と 5.2 です。バージョン 5.2 の BioRS サーバーにアクセスする場合、VERSION パラメーターの値として '5.2' を指定する必要があります。バージョン 5.0.14 を使用している場合は、このオプションを指定する必要はありません。デフォルト値 '1.0' は、バージョン 5.0.14 と等価で、ユーザーが値を指定しなかった場合にこのパラメーターに使用されます。

WRAPPER *biors_wrapper*

CREATE WRAPPER ステートメントで指定したラッパー名。

NODE *'biors_myco.com'*

BioRS 照会ツールが使用可能なシステムのホスト名を指定します。デフォルト値は、*localhost* です。この値は大文字と小文字を区別します。

ノード名は CREATE SERVER ステートメントではオプションとなりますが、BioRS データ・ソースの場合は必要です。

オプション・パラメーターとサーバー・オプションを使用したサーバー定義の登録

以下の例には、BioRS ラッパーにサーバー定義を登録するときに指定できる追加パラメーターとサーバー・オプションが示されています。

```
CREATE SERVER biors_server TYPE BioRS VERSION '5.2'  
WRAPPER biors_wrapper  
OPTIONS (NODE 'biors_server2.com', PORT '5555', TIMEOUT 30 ,  
CASE_SENSITIVE 'N');
```

TYPE *BioRS*

アクセスを構成するデータ・ソース・サーバーのタイプを指定します。BioRS ラッパーの場合、サーバー・タイプは BioRS でなければなりません。このパラメーターはオプションです。

PORT *'5555'*

BioRS サーバーへの接続に使用するポート番号を指定します。デフォルト値は '*5014*' です。

TIMEOUT *30*

BioRS ラッパーの BioRS サーバーからの応答待機時間を分単位で指定します。デフォルト値は *10* です。このパラメーターはオプションです。

CASE_SENSITIVE *'N'*

BioRS サーバーが大/小文字を区別して名前を扱うかどうかを指定します。有効値は '*Y*' または '*N*' です。デフォルト値は '*Y*' です。

BioRS 製品では、BioRS サーバー・マシンに格納されているデータの大/小文字の区別を構成パラメーターで制御します。CASE_SENSITIVE オプションは、フェデレーテッド・サーバーにおける BioRS システム構成パラメーターに相当します。BioRS システムとフェデレーテッド・サーバーで、BioRS サーバーの大/小文字の区別の構成設定を同期させなければなりません。BioRS サーバーとフェデレーテッド・サーバーの間で大/小文字の区別に関する構成設定を同期させておかないと、フェデレーテッド・サーバーを介した BioRS データへのアクセスを試行した際にエラーが発生します。

重要: 新しい BioRS 定義を作成した後で、CASE_SENSITIVE オプションを変更または削除することはできません。CASE_SENSITIVE オプションを変更する必要がある場合は、サーバー定義をドロップしてから再作成する必要があります。BioRS サーバー定義をドロップした場合、そのサーバー定義を参照していたすべての BioRS ニックネームも作成する必要があります。フェデレーテッド・サーバーは、ドロップされたサーバーに対応するすべてのニックネームを自動的にドロップします。

プロキシ・サーバーを含むサーバー定義の登録

```
CREATE SERVER biors_proxy_serv VERSION 5.2 WRAPPER biors_proxy
  OPTIONS (NODE 'biors.mysite.com',
          PORT '5555',
          PROXY_TYPE 'HTTP'
          PROXY_SERVER_NAME 'proxy.mysite.com
          PROXY_SERVER_PORT '81'
```

PROXY_TYPE 'HTTP'

ファイアウォールで保護された状態でインターネットにアクセスする際に使用する、プロキシ・タイプを指定します。有効な値は 'NONE'、'HTTP' または 'SOCKS' です。

PROXY_SERVER_NAME 'proxy.mysite.com'

プロキシ・サーバー名または IP アドレスを指定します。PROXY_TYPE サーバー・オプションの値が 'HTTP' または 'SOCKS' の場合、このオプションは必須です。

PROXY_SERVER_PORT '81'

プロキシ・サーバーのポート番号を指定します。PROXY_TYPE サーバー・オプションの値が 'HTTP' または 'SOCKS' の場合、このオプションは必須です。

認証情報のあるプロキシ・サーバーを含むサーバー定義の登録

サーバー定義を登録して、認証情報のある SOCKS プロキシ・サーバーを指定するには、以下のステートメントを使用します。

```
CREATE SERVER biors_proxy_serv VERSION 5.2 WRAPPER biors_proxy
  OPTIONS (NODE 'biors.mysite.com',
          PORT '5555',
          PROXY_TYPE 'SOCKS'
          PROXY_SERVER_NAME 'proxy_socks',
          PROXY_SERVER_PORT '1081',
          PROXY_AUTHID 'argle'
          PROXY_PASSWORD 'bargle')
```

PROXY_AUTHID 'argle'

プロキシ・サーバーのユーザー ID を指定します。PROXY_TYPE サーバー・オプションの値が 'SOCKS' の場合、このサーバー・オプションは必須です。

PROXY_PASSWORD 'bargle'

ユーザー名 '*argle*' に関連したプロキシ・サーバー上のパスワードを指定します。PROXY_TYPE サーバー・オプションの値が 'SOCKS' の場合、このサーバー・オプションは必須です。

BioRS データ・ソースのユーザー・マッピングの作成

BioRS サーバーにアクセスする場合、フェデレーテッド・サーバーは BioRS サーバーへの接続を確立します。BioRS システムで使用しているアカウントへのアクセス方式により、ユーザー・マッピングを作成する必要がある場合があります。

始める前に

ユーザー・マッピングは、各フェデレーテッド・サーバー・ユーザー ID およびパスワードと、対応するデータ・ソース・ユーザー ID およびパスワードとの間の関連付けです。

ユーザー・マッピングをフェデレーテッド・システムで指定するには 2 つの方式があります。LDAP などの外部リポジトリを使用してユーザー・マッピングを保管するか、またはフェデレーテッド・データベース・カタログにユーザー・マッピングを作成することができます。

BioRS システムで使用しているアカウント・アクセス方式により、ユーザー・マッピングを作成する必要があるかどうかが決まります。

- BioRS サーバーですべてのユーザー・アカウントにゲスト・アクセスを設定している場合、ユーザー・マッピングを作成する必要はありません。フェデレーテッド・サーバーは、ゲスト・アカウントを使用して BioRS サーバーにアクセスします。
- LDAP など、ユーザー・マッピングを保管できる外部リポジトリがある場合は、ユーザー・マッピングを作成する必要はありません。BioRS ラッパーでは DB2_UM_PLUGIN オプションを指定する必要があります。このオプションは、ラッパーの登録または変更の際に指定できます。外部リポジトリのスキーマには、ゲスト・アクセスが含まれている必要があります。
- BioRS サーバーが、ユーザー・アカウントを ID とパスワードで認証するよう構成されている場合、BioRS データ・ソースへのアクセスに BioRS ラッパーを使用するあらゆるユーザー・アカウントに対応するユーザー・マッピングをフェデレーテッド・データベースに作成する必要があります。
- BioRS サーバーが、ゲストおよび認証ユーザー・アカウントの両方を使用するよう構成されている場合、BioRS データ・ソースへのアクセスに BioRS ラッパーを使用する認証ユーザー・アカウントに対応するユーザー・マッピングを作成する必要があります。

このタスクについて

ユーザー・マッピングは、BioRS ラッパーを使用して BioRS データ・ソースを照会するユーザーまたはアプリケーションのアクセスを認証します。ユーザー・マッピングがユーザーまたはアプリケーションに対して定義されていない場合、フェデレーテッド・サーバーはゲスト・アカウントを使用します。照会対象のデータ・バンクで認証が必要な場合、エラー・メッセージが戻される可能性があります。

正しいユーザー ID とパスワードを確実に BioRS サーバーに渡すには、BioRS データ・ソースの検索を許可されたユーザーのユーザー・マッピングをフェデレーテッド・データベースに作成します。ユーザー・マッピングを作成する際、リモート・パスワードは暗号化された形式でフェデレーテッド・データベースのシステム・カタログ表に格納されます。

手順

コマンド行から CREATE USER MAPPING ステートメントを発行します。例えば、次のようにします。

```
CREATE USER MAPPING FOR local_userID
SERVER server_definition_name
OPTIONS (REMOTE_AUTHID 'remote_userID',
REMOTE_PASSWORD 'remote_password')
PROXY_AUTHID 'proxy_server_userID',
PROXY_PASSWORD 'proxy_server_password';
```

例

サーバー定義とユーザー・マッピングの登録に際してプロキシ・サーバーの認証情報を指定する場合は、CREATE USER MAPPING ステートメントで指定した値が CREATE SERVER ステートメントで指定した値よりも優先されます。

例えば、組織に 10 人の人がいて、サーバー定義を登録する際に認証情報を指定するとします。ユーザー・マッピングは 10 人のうち 3 人について作成します。この 3 人がフェデレーテッド・システムにアクセスするときには、ユーザー・マッピング作成時に指定した認証情報が使用されます。残りの 7 人に関しては、サーバー定義の登録時に指定した認証情報が使用されます。

CREATE USER MAPPING ステートメント - BioRS ラッパーの例:

CREATE USER MAPPING ステートメントを使用して、フェデレーテッド・サーバーのユーザー ID を BioRS サーバーのユーザー ID およびパスワードにマップしてください。

ゲスト・ユーザー・アカウント、認証ユーザー・アカウント、USER 特殊レジスターまたはプロキシ・サーバーの指定により、ユーザー・マッピングを作成できます。

ゲスト・ユーザー・アカウントのユーザー・マッピングの作成

GUEST ユーザー・オプションでは、BioRS ラッパーが BioRS サーバーにアクセスするのにゲスト・アカウントを使用するかどうかを指定します。

以下の例では、BioRS サーバーへのアクセスにゲスト・ユーザー・アカウントを使用することを指定する方法を示します。

```
CREATE USER MAPPING FOR charlie SERVER biors_server
  OPTIONS (GUEST 'Y');
```

charlie BioRS サーバーで定義されたりリモート・ユーザー ID にマップするローカルの許可 ID を指定します。

SERVER *biors_server*

CREATE SERVER ステートメントで登録した BioRS サーバーのサーバー定義名を指定します。

GUEST 'Y'

BioRS ラッパーがこのユーザーの認証にゲスト・ユーザー・アカウントを使用することを指定します。

認証ユーザー・アカウントに対応するユーザー・マッピングの作成

以下に示すのは、フェデレーテッド・サーバーのユーザー ID を BioRS サーバーのユーザー ID およびパスワードにマップする方法を示す例です。

```
CREATE USER MAPPING FOR charlie SERVER biors_server
  OPTIONS (REMOTE_AUTHID 'charlene',
  REMOTE_PASSWORD 'all4one');
```

charlie BioRS サーバーで定義されたりリモート・ユーザー ID およびパスワードにマップするローカルの許可 ID を指定します。

SERVER *biors_server*

CREATE SERVER ステートメントで登録した BioRS サーバーのサーバー定義名を指定します。

REMOTE_AUTHID '*charlene*'

charlie をマップする BioRS サーバー側のユーザー ID を指定します。このリモート ID は、BioRS サーバーで受け入れ可能な形式でなければなりません。

REMOTE_PASSWORD '*all4one*'

'*charlene*' に関連するパスワードを指定します。

特殊レジスターを使用してのユーザー・マッピングの作成

フェデレーテッド・データベース特殊レジスター USER を使用して、CREATE USER MAPPING ステートメントを実行する人の許可 ID を、REMOTE_AUTHID ユーザー・オプションに指定されるデータ・ソースの許可 ID にマップすることができます。

以下に、特殊レジスター USER を含む CREATE USER MAPPING ステートメントの例を示します。

```
CREATE USER MAPPING FOR USER SERVER biors_server
  OPTIONS (REMOTE_AUTHID 'charlene', REMOTE_PASSWORD 'all4one');
```

プロキシ・サーバーのユーザー・マッピングの作成

以下に示すのは、フェデレーテッド・サーバーのユーザー ID を BioRS サーバーのユーザー ID およびパスワードにマップする方法を示す例です。

サーバー定義を登録して、認証情報のある SOCKS プロキシ・サーバーを指定するには、以下のステートメントを使用します。

```
CREATE USER MAPPING FOR charlie SERVER biors_proxy
  OPTIONS (REMOTE_AUTHID 'charlene',
  REMOTE_PASSWORD 'all4one'
  PROXY_AUTHID 'chuck'
  PROXY_PASSWORD 'them2us');
```

PROXY_AUTHID '*chuck*'

プロキシ・サーバーのユーザー ID を指定します。プロキシ・サーバーで認証が必要である場合に、このユーザー・マッピング・オプションは必須です。

PROXY_PASSWORD '*them2us*'

ユーザー名 '*chuck*' に関連したプロキシ・サーバー上のパスワードを指定します。プロキシ・サーバーで認証が必要である場合に、このユーザー・マッピング・オプションは必須です。

BioRS データ・ソースのニックネームの登録

登録するそれぞれの BioRS サーバー定義で、アクセスしたいそれぞれのデータ・バンクごとにニックネームを登録する必要があります。BioRS サーバーを照会するときには、データ・バンクの名前ではなく、これらのニックネームを使用します。

始める前に

- CREATE NICKNAME ステートメントが必要とする構文に、BioRS データ・バンク名が準拠していない場合、ニックネームを登録する際に REMOTE_OBJECT ニックネーム・オプションを使用しなければなりません。
- CREATE NICKNAME ステートメントが必要とする構文に、BioRS エlement名が準拠していない場合、ニックネームを登録する際に ELEMENT_NAME 列オプションを使用しなければなりません。

このタスクについて

制約事項

BioRS AllText エlementをニックネームの 1 列目に使用しないでください。その他の列 (2 列目や 3 列目) では使用できません。

BioRS システムに統合されたデータ・ソースを、BioRS では データ・バンク と呼びます。BioRS 内のデータ・バンクはフェデレーテッド・システム上のニックネームと同じです。

ニックネームに指定できる名前の長さは 128 文字までです。

手順

コマンド行から CREATE NICKNAME ステートメントを発行します。

例:

```
CREATE NICKNAME nickname
(
  column_name data_type
    OPTIONS (nickname_column_options),
  column_name data_type
    OPTIONS (nickname_column_options),
  column_name data_type
    OPTIONS (nickname_column_options)
)
FOR SERVER server_definition_name
OPTIONS (nickname_options);
```

BioRS ニックネームを作成する場合、ニックネーム列のリストを定義してください。指定したニックネーム列は、特定の BioRS データ・バンク形式のElementに対応する必要があります。BioRS では、Elementに対して、使用可能な 5 つのデータ・タイプ (Text、Number、Date、Author、および Reference) を定義しています。BioRS データ・タイプは、フェデレーテッド・データベースによって使用される CHAR、CLOB、または VARCHAR データ・タイプにのみマップできます。

次のタスク

ニックネームを作成する BioRS データ・バンクごとに、このステップを繰り返します。

CREATE NICKNAME ステートメント - BioRS ラッパーの例:

CREATE NICKNAME ステートメントを使用して、アクセスしたい BioRS データ・バンクのニックネームを登録します。

データ・バンクの簡単なニックネームの作成 (制限あり)

BioRS データ・バンクのニックネームを登録する最も簡単な方法は、BioRS データ・バンクと同じ名前のニックネームを付けることです。

例:

```
CREATE NICKNAME SwissProt
  (ID VARCHAR(32) OPTIONS (ELEMENT_NAME '_ID_'),
  ALLTEXT VARCHAR(128),
  ENTRYDATE VARCHAR (64))
FOR SERVER biors_server;
```

ニックネームは *SwissProt* で、これは対応する BioRS データ・バンクの名前と同じです。

この簡単な CREATE NICKNAME 構文を使用した場合、次の 2 つの制限が生じます。

1. 各フェデレーテッド・データベース・スキーマに対して使用できるニックネームのファミリーは、1 つに制限されます。例えば、親子関係になっている 2 つのデータ・バンクがあるとします。データ・バンクは SWISSPROT と SPFEAT です。これらのデータ・バンクはファミリーを形成します。CREATE NICKNAME ステートメントのデフォルトの構文を使用する場合は、SWISSPROT データ・バンクのニックネームとして SWISSPROT を、SPFEAT データ・バンクのニックネームとして SPFEAT を持つようになります。スキーマ内で SWISSPROT に複数のニックネームを持つには、ニックネームを登録する際に REMOTE_OBJECT ニックネーム・オプションを使用しなければなりません。
2. 使用できるデータ・バンクは、ニックネームとして使用できる名前を持つものに限定されます。データ・バンク名は、フェデレーテッド・サーバーによってサポートされている構文に従う必要があります。例えば、データ・バンク名にピリオドやスペースが含まれている場合、ニックネームの登録に際しては、REMOTE_OBJECT ニックネーム・オプションを使用する必要があります。

同一のデータ・バンクに対する複数のニックネームの作成

REMOTE_OBJECT ニックネーム・オプションでは、ニックネームに関連する BioRS データ・バンク名を指定します。REMOTE_OBJECT ニックネーム・オプションで指定する名前によって、ニックネームのスキーマと BioRS データ・バンクが決まります。REMOTE_OBJECT ニックネーム・オプションはニックネーム間の関係も指定します。

以下の例は、前の例と同じニックネーム特性の設定を使用していますが、ニックネームを変えています。例では REMOTE_OBJECT ニックネーム・オプションを使用して、ニックネームを定義する BioRS データ・バンクを指定しています。

```
CREATE NICKNAME NewSP
  (ID VARCHAR(32) OPTIONS (ELEMENT_NAME '_ID_'),
  ALLTEXT VARCHAR(128),
  ENTRYDATE VARCHAR (64))
FOR SERVER biors_server
  OPTIONS (REMOTE_OBJECT 'SwissProt');
```

フェデレーテッド構文に適合しないデータ・バンクのニックネームの作成

以下の例は、フェデレーテッド・サーバーが要求する構文に適合しないリモート BioRS データ・バンクのニックネームを作成する方法を示します。

```
CREATE NICKNAME SwissFT
  (ID VARCHAR(32) OPTIONS (ELEMENT_NAME '_ID_'),
  ALLTEXT VARCHAR (128),
  ENTRYDATE VARCHAR (64),
  FtLength VARCHAR (16))
  FOR SERVER biors1
  OPTIONS (REMOTE_OBJECT 'SwissProt.Features');
```

SwissFT

BioRS データ・バンクを識別するために使用する固有なニックネームです。

ID VARCHAR(32) OPTIONS (ELEMENT_NAME '_ID_')

表の列の名前とデータ・タイプ。ELEMENT_NAME 列オプションは ID 列に対して指定します。

ELEMENT_NAME 列オプションは BioRS エlement名を指定します。この名前で大/小文字の区別をすることは、BioRS サーバーで大/小文字の区別をすることは、および CASE_SENSITIVE サーバー・オプションの値によって決まります。BioRS エlement名と列名が異なる場合のみ、Element名を指定する必要があります。列オプション値は、単一引用符で囲む必要があります。

通常、以下のような場合に ELEMENT_NAME 列オプションを使用します。

- BioRS エlement名に、有効なフェデレーテッド構文に適合しない、ピリオドやスペースなどの文字が含まれている場合。例えば、データ・バンクに Pub.Date という名前のElementが含まれている場合、そのElement名は列名として使用できません。ピリオドやスペースなどの文字はサポートされていません。Element名は有効な列名にマップする必要があります。
- BioRS エlement名の構文が、ユーザーあるいはユーザーの属する組織がフェデレーテッド・システムに関して規定した規格に適合していない場合。例えば、ユーザーの組織で、スキーマ、ニックネーム、列には必ず接頭部をつけるという規則になっていれば、BioRS エlement名を列名としては使用できないことになります。
- BioRS エlement名がフェデレーテッド・ユーザーにとって明白ではない可能性がある場合。

ALLTEXT VARCHAR(128)

表の列の名前とデータ・タイプ。

ENTRYDATE VARCHAR(64)

表の列の名前とデータ・タイプ。

FtLength VARCHAR(16)

表の列の名前とデータ・タイプ。

SERVER biors1

CREATE SERVER ステートメントで BioRS サーバーに割り当てた名前。

OPTIONS (REMOTE_OBJECT 'SwissProt.Features')

ニックネームに関連する BioRS データ・バンク名を指定します。この名前

によって、ニックネームのスキーマと BioRS データ・バンクが決まります。この名前によって、そのニックネームと他のニックネームの関係も指定されます。

この名前で大/小文字の区別をするかどうかは、BioRS サーバーで大/小文字の区別をするかどうか、および CASE_SENSITIVE サーバー・オプションの値によって決まります。

重要: ALTER NICKNAME ステートメントではこの名前を変更または削除できません。BioRS データ・バンクの名前を変更した場合、ニックネームを削除してから、再びそれを再作成しなければなりません。

BioRS データ・バンク名が、有効なフェデレーテッド構文の規格に適合しない場合、REMOTE_OBJECT ニックネーム・オプションを指定する必要があります。この例では、データ・バンク名 "SwissProt.Features" はいくつかの理由で適合していません。データ・バンク名に、フェデレーテッド構文に適合しない文字、ピリオドが含まれており、大文字と小文字が混在しています。

通常、以下のような場合に REMOTE_OBJECT ニックネーム・オプションを使用します。

- BioRS データ・バンク名に、有効なフェデレーテッド構文に適合しない、ピリオドやスペースのような文字が含まれている場合。データ・バンク名は有効なフェデレーテッド名にマップする必要があります。
- BioRS データ・バンク名の大/小文字の区別が、ユーザーあるいはユーザーの属する組織がフェデレーテッド・システムに関して規定した規格に適合していない場合。例えば、ユーザーの組織で、スキーマ、ニックネーム、列には必ず接頭部をつけるという規則になっていれば、BioRS データ・バンク名はそれらの名前としては使用できないことになります。
- BioRS データ・バンク名がフェデレーテッド・ユーザーにとって明白ではない可能性がある場合。

他の BioRS データ・バンクにリンクされたデータ・バンクのニックネームの作成

以下の例は、他の BioRS データ・バンクにリンクする BioRS データ・バンクを使用する表のニックネームを作成する方法を示します。

```
CREATE NICKNAME SwissFT2
  (ID VARCHAR(32) OPTIONS (ELEMENT_NAME '_ID_'),
  ALLTEXT VARCHAR (1200),
  FtKey VARCHAR (32),
  FtLength VARCHAR (64),
  FtDescription VARCHAR (128),
  Parent VARCHAR (32) OPTIONS (REFERENCED_OBJECT 'SwissProt'))
FOR SERVER biors1
OPTIONS (REMOTE_OBJECT 'SwissProt.Features');
```

ニックネームは SwissFT2 です。表列は ID、ALLTEXT、FtKey、FtLength、FtDescription、および Parent です。ELEMENT_NAME 列オプションは ID 列に対して指定します。REMOTE_OBJECT オプションは、ニックネームが対応する BioRS データ・バンクの名前の指定にも利用します。

さらに、Parent 列は REFERENCED_OBJECT オプションを使用します。このオプションは BioRS Reference データ・タイプ・エレメントに対応する列に指定する必要

があります。REFERENCED_OBJECT オプションは、列が参照する BioRS データ・バンクの名前を指定します。この場合、Parent エlementは BioRS SwissProt データ・バンクを参照します。

カスタム関数と BioRS 照会

BioRS カスタム関数は、BioRS ラッパーとともに使用され、BioRS 照会エンジンに述部をプッシュダウンします。

フェデレーテッド環境は 2 つの照会エンジンを使用します。BioRS ラッパーの場合、照会エンジンはフェデレーテッド・データベースと BioRS です。4 つの BioRS カスタム関数を使用して、述部を BioRS エンジンにプッシュダウンすることを指定できます。

BioRS ラッパーのカスタム関数はすべて、BIORS というスキーマ名を使用して登録する必要があります。この種の関数を使用するときには、常に BIORIS スキーマを含める必要があります。

BIORS ラッパーのカスタム関数は以下のとおりです。

- BIORIS.CONTAINS
- BIORIS.CONTAINS_LE
- BIORIS.CONTAINS_GE
- BIORIS.SEARCH_TERM

BioRS CONTAINS 関数

カスタム関数 BIORIS.CONTAINS、BIORS.CONTAINS_LE、および BIORIS.CONTAINS_GE には検索語列引数と照会テキスト引数が必要です。以下に、BIORS.CONTAINS ステートメントの例を示します。

```
BIORS.CONTAINS (search_term_column,query_term)
```

検索語列引数の値は、索引付けされた BioRS 列を参照します。索引付けされていない列を使用すると、エラー・メッセージ SQL30090N (「操作がアプリケーション実行環境で無効です。」) が表示されます。

照会語引数の値は、検索語列引数に指定された索引付きエレメントの検索に使用される値です。

照会語引数の値は、リテラル、ホスト変数、または列参照のいずれかになります。演算またはストリング連結は使用できません。また、使用する検索語列が NULL 値を許可するよう定義済みでも、照会語引数値は NULL を取れません。

照会語引数では、大/小文字は区別されません。

有効なデータ・タイプ

照会語引数で有効なデータ・タイプと形式は、使用される検索語列の BioRS データ・タイプによって決まります。BioRS では、使用可能な 5 つのデータ・タイプ (Text、Author、Date、Number、および Reference) を定義しています。

BioRS データ・タイプおよびデータ・タイプごとの有効な関数照会語を、以下の表にリストします。

表 10. BioRS データ・タイプと有効なカスタム関数照会語

検索語列のデータ・タイプ	有効な照会語	形式
Text	VARCHAR() または CHAR()	BioRS テキスト用語 (ワイルドカードを含む)。
Author	VARCHAR() または CHAR()	"<last>, <init>" 書式の BioRS 作成者参照。"<last>" は作成者の姓です。"<init>" は作成者のイニシャルですが、ピリオドはつけません。コンマとイニシャルの間には空白文字を挿入できます。 また、コンマやイニシャルを含めずに <last> だけを指定することもできます。
Date	VARCHAR(), CHAR(), DATE, または TIMESTAMP	文字ストリングの場合、フェデレーテッド・データベースに有効な日付形式、yyyy/mm/dd
Number	VARCHAR() または CHAR()、INTEGER、SMALLINT、BIGINT REAL、DOUBLE、DECIMAL	フェデレーテッド・データベースに有効な数字フォーマット
Reference	VARCHAR() または CHAR()	BioRS テキスト用語。

BioRS データ・タイプ検索語列と照会語引数などのその他の組み合わせを使用すると、エラー・メッセージ SQL30090N (「操作がアプリケーション実行環境で無効です。」) が表示されます。

ワイルドカード文字の使用

Text、Author、および Reference データ・タイプ検索語列の照会語引数は、BioRS 照会言語パターンと一致する必要があります。BioRS では、英数字ストリングとワイルドカードで照会語引数を作成できます。BIORS.CONTAINS は、疑問符 (?) とアスタリスク (*) の 2 つのワイルドカードをサポートします。

? ワイルドカードは英数字 1 文字に相当します。例えば、述部 BioRS.CONTAINS (description, 'bacteri?')=1 は、用語 bacteria と一致しますが、bacterial とは一致しません。

ワイルドカード文字 * は、ゼロまたは複数の文字に相当します。例えば、述部 BioRS.CONTAINS (description, 'bacteri*')=1 は、用語 bacteri、bacteria、および bacterial のすべてと一致します。

BioRS 照会言語パターンの詳細については、ご使用の BioRS の資料を参照してください。

照会での BioRS CONTAINS 関数の指定

BIORS.CONTAINS 関数はすべての BioRS 列型に指定できます。

BIORS.CONTAINS_GE と BIOR.S.CONTAINS_LE カスタム関数は基本 BioRS データ・タイプが Number もしくは Date である列にのみ指定できます。

BIORS.CONTAINS_GE 関数は、照会語引数が表す値より大きいか等しい値が含まれている列の行を選択します。BIORS.CONTAINS_LE 関数は、照会語引数が表す値より小さいか等しい値が含まれている列の行を選択します。

BIORS.CONTAINS、BIORS.CONTAINS_GE、および BIOR.S.CONTAINS_LE 関数は整数の結果を戻します。3 つの CONTAINS 関数のいずれかを述部で使用する場
合、戻り値は = または <> 演算子を使用して値 1 と比較されます。例：

```
SELECT * FROM s.MySP WHERE BIOR.S.CONTAINS (s.AllText, 'muscus') = 1;
```

表現 NOT (BioRS.Contains (col,value) = 1) は、表現 BioRS.CONTAINS (col,value) <> 1 と同じです。

BioRS SEARCH_TERM 関数

BIORS.SEARCH_TERM カスタム関数は、検索語列引数および照会語を必要とします。SEARCH_TERM カスタム関数の構文例は以下のとおりです。

```
BIORS.SEARCH_TERM (search_term_column,query_term)
```

検索語列引数の値は、_ID_ エレメントに対応する列を参照する必要があります。

照会語引数の値は、複数のエレメントを参照できる式です。

照会語引数の値は、リテラル、ホスト変数、または列参照のいずれかになります。演算またはストリング連結は使用できません。また、使用する検索語列が NULL 値を許可するよう定義済みでも、照会語引数値は NULL を取れません。

照会語引数では、大/小文字は区別されません。

照会での BioRS SEARCH_TERM 関数の指定

BIORS.SEARCH_TERM 関数を実行すると、別の方法では実行が難しい照会を実行
できます。BioRS 形式を使用した検索語の指定にこの関数を使用できます。

BIORS.SEARCH_TERM 関数には 2 つの引数が必要です。最初の引数は、用語が適用されるニックネームの _ID_ 列への参照です。2 番目の引数は、データ・バンク名を持たない用語を含む文字ストリングです。

以下の例では、SeqLength エレメントが 100 以上の値を含む MyEMBL データ・バンクの項目に対する列のすべてを選択しています。

```
SELECT * FROM MyEMBL s WHERE  
  BIOR.S.SEARCH_TERM (s.ID, '[SeqLength GREATER number:100;]') = 1;
```

以下の例では、Swiss ニックネームから MolWeight エレメントの値が 100368 以上の MolWeight 列を選択しています。

```
SELECT s.molweight FROM Swiss s WHERE  
  BIOR.S.SEARCH_TERM (s.ID, '[MolWeight GREATER number:100368;]') = 1;
```

BioRS 照会における等価演算

等価演算子 (=) はリテラル式または結合照会で使用できますが、一定の制限があります。

リテラル式または結合照会で等価演算子を使用する場合、照会が BioRS サーバーにプッシュダウンされるためには、等価演算子が BioRS データ・バンクの `_ID_` エlementを参照している必要があります。等価演算子を含んでいても `_ID_` Elementを参照しない照会は、BioRS サーバーによる処理のためにプッシュダウンされることはありません。

等価演算子は、リテラル式の中で使用できます。例：

```
ID = 'swissprot:100K_RAT'
```

等価述部は、BioRS データ・バンクと他のローカル表または nonBioRS ニックネームとの結合の中で使用できます。例：

```
SELECT n.ID, n.EntryDate, t.C1 FROM w46851_n1 n, w46851_t1 t WHERE t.ID = n.ID
```

BioRS データ・バンクの間の結合では、一方のデータ・バンクの `_ID_` Elementと、もう一方のデータ・バンクの参照タイプ・Elementを参照する必要があります。

ただし、等価述部を使用すると、以下のような条件で、予期したものとは異なる結果が戻される場合があります。

大/小文字を区別しないマッチング

演算では大/小文字を区別しません。例えば、`ID='100k_rat'` は以下の両方のストリングに一致します。

- '100k_rat'
- '100K_RAT'

ワイルドカード・マッチング

ステートメント `ID='100K_R*'` は、`'100K_RAT'` と `'100K_RODENT'` の両方に一致します。

データ・バンク・プレフィックス

演算により、ソース・データ・バンクを示すプレフィックスが戻されます。例えば、SwissProt データ・バンクに対する結合に `ID='100K_RAT'` と指定すると、値 `'swissprot:100K_RAT'` が戻される可能性があります。

注：こうした動作のいずれかに依存するようなアプリケーションは作成しないでください。

以下では、結合での等価述部の振る舞いの例を示します。

ローカル表 `w46851_t1` には、以下の値が含まれます。

ID	C1
swissprot:100K_RAT	0
swissprot:RAT	1
swissprot:100K_R	2
swissprot:100K_R*	3
swissprot:100k_rat	4
100K_RAT	100

RAT	101
100K_R	102
100K_R*	103
100k_rat	104

表 w46851_t1 を SwissProt データ・バンクを基にしたニックネーム w46851_n1 と結合できます。以下のステートメントは、等価演算を含む結合照会を示しています。

```
SELECT n.ID, n.EntryDate, t.C1 FROM w46851_n1 n, w46851_t1 t WHERE t.ID = n.ID
```

戻される結果は以下のようになります。

ID	ENTRYDATE	C1
swissprot:100K_RAT	01-NOV-1997	0
swissprot:100K_RAT	01-NOV-1997	3
swissprot:100K_RAT	01-NOV-1997	4
swissprot:100K_RAT	01-NOV-1997	100
swissprot:100K_RAT	01-NOV-1997	103
swissprot:100K_RAT	01-NOV-1997	104

6 record(s) selected.

ただし、予期された振る舞いは、行 0 のみが戻されるというものです。

BioRS ラッパーの等価結合述部

BioRS カスタム関数を使用する場合、BioRS エンジンに対する述部を指定しなければなりません。ただし、例外が 1 つあります。それは、照会中に等価結合操作を行う場合です。

結合操作では、一致する列の値を基にして複数の表からデータを取り出します。等価結合は、式 = 式形式の結合条件をとる結合操作です。BioRS 照会では、等価結合条件は片方のデータ・バンクの `_ID_` エレメントともう一方のデータ・バンクの `Reference` タイプ・エレメントを含む必要があります。

ニックネーム定義と等価結合照会の例

ここでは、ニックネーム定義のサンプルとサンプルのニックネームを使用した等価結合照会を例示します。

2 つの BioRS データ・バンク、SwissProt と SwissProt.features を照会します。SwissProt.features データ・バンクは SwissProt データ・バンクの子で、Parent と呼ばれるエレメントを含みます。Parent エレメントは SwissProt の `_ID_` エレメントが示す項目への参照を含みます。2 つのデータ・バンクに対して 2 つのニックネーム定義を登録します。

SwissProt データ・バンクのニックネーム定義

SwissProt データ・バンクは親データ・バンクです。

```
CREATE NICKNAME tc600sprot (
  ID          VARCHAR (32) OPTIONS (ELEMENT_NAME '_ID_'),
  AllText     VARCHAR (128),
  EntryDate   VARCHAR (128),
  Update      VARCHAR (128),
  Description VARCHAR (1200),
  Crossreference VARCHAR (32),
  Authors     VARCHAR (256),
```

```

Journal          VARCHAR (256),
JournalIssue     VARCHAR (64) OPTIONS (IS_INDEXED 'N'),
PublicationYear  VARCHAR (1024),
Gene             VARCHAR (20) OPTIONS (IS_INDEXED 'Y'),
Remarks         VARCHAR (1200),
RemarkType      CHAR (20),
CatalyticActivity VARCHAR (20),
CoFactor         VARCHAR (64),
Disease          VARCHAR (128),
Function         VARCHAR (128),
Pathway          VARCHAR (128),
Similarity       VARCHAR (128),
Complex          VARCHAR (64),
FtKey            VARCHAR (32),
FtDescription    VARCHAR (128),
FtLength         VARCHAR (256),
MolWeight        VARCHAR (64),
ProteinLen       VARCHAR (32) OPTIONS (ELEMENT_NAME 'Protein_length'),
Sequence         CLOB,
AccNumber        VARCHAR (32),
Taxonomy         VARCHAR (128),
Organelle        VARCHAR (128),
Organism         VARCHAR (128),
Keywords         VARCHAR (1200),
Localization     VARCHAR (128),
FtKey_count      VARCHAR (32))
FOR SERVER biors_server_600
  OPTIONS (REMOTE_OBJECT 'SwissProt');

```

SwissProt.features データ・バンクのニックネーム定義

SwissProt.features データ・バンクは SwissProt データ・バンクの子データ・バンクです。このニックネームには **Parent** エレメントが含まれています。

```

CREATE NICKNAME tc600feat (
  ID          VARCHAR (32) OPTIONS (ELEMENT_NAME '_ID_'),
  AllText     VARCHAR (1200),
  FtKey       VARCHAR (32),
  FtLength    VARCHAR (64),
  FtDescription VARCHAR (128),
  Parent      VARCHAR (32) OPTIONS (REFERENCED_OBJECT 'SwissProt'))
FOR SERVER biors_server_600
  OPTIONS (REMOTE_OBJECT 'SwissProt.features');

```

1 つの等価結合で 2 つのニックネームを参照する照会

この照会では、2 つの述部が tc600sprot ニックネーム (SwissProt データ・バンク) に適用されます。2 つの述部は用語 *anopheles* を含み、発行年が 1997 年である行をフィルターに掛けます。1 つの述部が tc600feat ニックネーム (SwissProt.features データ・バンク) に適用され、用語 *signal* を含む FtKey エレメントのある行をフィルターに掛けます。2 つのニックネームは条件 *f.Parent = s.ID* を使用して結合されます。

```

SELECT s.ID, f.ID, f.FtKey FROM tc600sprot s, tc600feat f
  WHERE BIORS.CONTAINS (s.AllText, 'anopheles') = 1
  AND BIORS.CONTAINS (s.PublicationYear, 1997) = 1
  AND BIORS.CONTAINS (f.FtKey, 'signal') = 1
  AND f.Parent = s.ID;

```

最終結果セットには、これらの基準に適合する行、および SwissProt データ・バンクの一致する項目を参照する SwissProt.features データ・バンクの項目のみが含まれます。

BioRS AllText エlement

BioRS システムのデータ・バンクにはそれぞれ、AllText と呼ばれるElementがあります。AllText Elementは、BioRS がすべてのデータ・バンクに自動的に作成する、索引付けされたElementです。

AllText Elementを使用することにより、特定の索引付けされたElementのみでなく、項目のテキストすべてを検索できます。例えば、`muscus` という用語を検索すると、単語 `muscus` が表題、要約、説明、有機体に使用されている項目を戻せます。

AllText Elementをフェデレーテッド照会で使用するには、AllText Elementをニックネーム列にマップする必要があります。CREATE NICKNAME ステートメントで列を指定する場合、AllText Elementをニックネーム列にマップします。AllText Elementへマップされているニックネーム列は、SELECT ステートメントで NULL 値を戻します。AllText Elementとして列を指定する場合、その列を CREATE NICKNAME ステートメントで宣言される最初の列にしてはなりません。

AllText Elementをニックネーム列に適切にマップすると、そのニックネーム列を `BIORS.CONTAINS` カスタム関数で使用できます。

BioRS データ・ソース - 照会の例

以下の例には、BioRS データ・ソースへのアクセスに使用できるサンプル照会の包括的なセットが含まれており、例の中で使用されるニックネームを作成するのに必要なステートメントも示されます。

以下の例では、次のことを行う方法を示します。

- システム・パフォーマンスを最適化するための照会の構成
- 照会でのカスタム関数とワイルドカードの使用
- 特定の BioRS データ・タイプ列にアクセスする照会の使用
- 関係述部を使用した、親ニックネームと子ニックネームの間の等価結合の構成

サンプル照会では、ニックネーム `swiss` と `swissft` を使用します。

swiss ニックネームのための CREATE NICKNAME ステートメント

親ニックネーム `swiss` は、以下の CREATE NICKNAME ステートメントを使用して、SwissProt データ・バンクに対して登録されています。

```
CREATE NICKNAME swiss
(
  ID                CHAR (30) OPTIONS (ELEMENT_NAME '_ID_'),
  EntryDate         VARCHAR (15),
  Update            CLOB (15),
  Description        CLOB (15),
  Crossreference     CLOB (15),
  Authors            CLOB (15),
  Journal            VARCHAR (15),
  JournalIssue       VARCHAR (15),
  PublicationYear    CLOB (15),
  PublicationTitle   CLOB (15),
  Gene               CLOB (15),
  Remarks            CLOB (15),
  RemarkType         VARCHAR (15),
  CatalyticActivity VARCHAR (15),
```

```

CoFactor          VARCHAR (15),
Disease           VARCHAR (15),
Function          CLOB (15),
Pathway           VARCHAR (15),
Similarity        CLOB (15),
Complex           VARCHAR (15),
FtKey             VARCHAR (15),
FtDescription     CLOB (15),
FtLength          VARCHAR (15),
MolWeight         CHAR (15),
Protein_Length   VARCHAR (15),
Sequence          CLOB (15),
AccNumber         VARCHAR (15),
Taxonomy          CLOB (15),
Organelle         VARCHAR (15),
Organism          VARCHAR (15),
Keywords          VARCHAR (15),
Localization      VARCHAR (15),
FtKey_count       VARCHAR (15),
AllText           CLOB (15)
)
FOR SERVER biors_server
OPTIONS (REMOTE_OBJECT 'swissprot');

```

swissft ニックネームのための CREATE NICKNAME ステートメント

子ニックネーム `swissft` は、以下の CREATE NICKNAME ステートメントを使用して、SwissProt.Features データ・バンクに対して登録されています。

```

CREATE NICKNAME swissft
(
  ID          VARCHAR (30) OPTIONS (ELEMENT_NAME '_ID_'),
  FtKey       VARCHAR (15),
  FtLength    VARCHAR (15),
  FtDescription VARCHAR (15),
  Parent      VARCHAR (30) OPTIONS (REFERENCED_OBJECT 'swissprot'),
  AllText     CLOB (15)
)
FOR SERVER biors_server
OPTIONS (REMOTE_OBJECT 'swissprot.features');

```

照会構造によるフェデレーテッド・サーバーのパフォーマンスへの影響

以下の表の照会と結果は、照会を構造化して、フェデレーテッド・システムと BioRS サーバーとの間のワークロードを最適化する方法を示しています。

表 11. 異なる照会で同じ結果が得られる例

照会	結果
<pre> SELECT s.id FROM Swiss s WHERE BIORS.CONTAINS(s.id, '100K_RAT') = 1 FETCH FIRST 3 ROWS ONLY; </pre>	<pre> ID ----- 100K_RAT 1 record(s) selected. </pre>
<pre> SELECT s.id FROM Swiss s WHERE s.id LIKE '%100K_RAT%' FETCH FIRST 3 ROWS ONLY; </pre>	<pre> ID ----- 100K_RAT 1 record(s) selected. </pre>

この表に示されている照会は、いずれも同じ結果となります。ただし、1 番目の照会は、2 番目の照会と比べかなり短い時間で実行されます。最初の照会は、入力述

部の指定に `BIORS.CONTAINS` 関数を使用しています。その結果、BioRS サーバーは SwissProt データ・バンクのデータの選択を行い、選択したデータをフェデレーテッド・サーバーに渡します。2 番目の照会では、LIKE 入力述部が `swiss` ニックネームに直接指定されています。そのため、BioRS サーバーは SwissProt データ・バンク全体をフェデレーテッド・サーバーに転送します。データ・バンクの内容が転送された後で、フェデレーテッド・サーバーがデータを選択します。

通常、処理の述部がデータ・ソースに送られた場合に、照会パフォーマンスは大きく向上します。

BIORS.CONTAINS カスタム関数にワイルドカードを使用した照会

以下の表に、`BIORS.CONTAINS` カスタム関数にワイルドカード文字を使用した照会と結果の例を示します。使用されているワイルドカード文字はそれぞれ異なりますが、照会結果はすべて同じになります。

表 12. `BIORS.CONTAINS` カスタム関数にワイルドカードを使用した照会の例

照会	結果
<pre>SELECT s.crossreference FROM Swiss s WHERE BIORS.CONTAINS(s.crossreference, 'MEDLINE') = 1 FETCH FIRST 3 ROWS ONLY;</pre>	<pre>CROSSREFERENCE ----- NCBI_TaxID=1011 NCBI_TaxID=5875 NCBI_TaxID=4081 3 record(s) selected.</pre>
<pre>SELECT s.crossreference FROM Swiss s WHERE BIORS.CONTAINS(s.crossreference, '?ED?IN?') = 1 FETCH FIRST 3 ROWS ONLY;</pre>	<pre>CROSSREFERENCE ----- NCBI_TaxID=1011 NCBI_TaxID=5875 NCBI_TaxID=4081 3 record(s) selected.</pre>
<pre>SELECT s.crossreference FROM Swiss s WHERE BIORS.CONTAINS(s.crossreference, '*D*N*') = 1 FETCH FIRST 3 ROWS ONLY;</pre>	<pre>CROSSREFERENCE ----- NCBI_TaxID=1011 NCBI_TaxID=5875 NCBI_TaxID=4081 3 record(s) selected</pre>

BioRS Author データ・タイプ列にアクセスする照会

以下の表の照会と結果は、`BIORS.CONTAINS` カスタム関数を使用して BioRS Author データ・タイプ・エレメントの情報にアクセスする方法を示します。すべての照会構文はほとんど同じです。照会語の先頭のイニシャルの有無、および、ファーストネームと最後のイニシャルの間のスペースの数のみが異なります。

表 13. BioRS Author データ・タイプ列にアクセスする照会の例

照会	結果
SELECT s.authors FROM Swiss s WHERE BIORS.CONTAINS(s.authors, 'Mueller') = 1 FETCH FIRST 3 ROWS ONLY;	AUTHORS ----- Mueller D. Rehb Mayer K.F.X. Sc Zemmour J. Litt 3 record(s) selected.
SELECT s.authors FROM Swiss s WHERE BIORS.CONTAINS(s.authors, 'Mueller,D') = 1 FETCH FIRST 3 ROWS ONLY;	AUTHORS ----- 0 record(s) selected.
SELECT s.authors FROM Swiss s WHERE BIORS.CONTAINS(s.authors, 'Mueller ,D') = 1 FETCH FIRST 3 ROWS ONLY;	AUTHORS ----- 0 record(s) selected.
SELECT s.authors FROM Swiss s WHERE BIORS.CONTAINS(s.authors, 'Mueller, D') = 1 FETCH FIRST 3 ROWS ONLY;	AUTHORS ----- Mueller D. Rehb Zou P.J. Borovo Davies J.D. Mue 3 record(s) selected.

BioRS Date データ・タイプ列にアクセスする照会

以下の表の照会と結果は、BIORS.CONTAINS カスタム関数を使用して BioRS Date タイプ・エレメントの情報にアクセスする方法を示します。

BioRS Date タイプのフィールドが日付のシーケンスを含む場合、表の 2 例目に見られるように、結果に余分な情報が含まれます。BioRS Numeric データ・タイプ・エレメント (Date と Number) には、複数の値を含めることができます。そのため、BioRS Date または Number エレメントに対して実行した照会結果も複数の値を含みます。複数の値は常にスペースで区切ります。

表 14. BioRS Date データ・タイプ列にアクセスする照会の例

照会	結果
SELECT e.entrydate FROM embl e WHERE BIORS.CONTAINS(e.entrydate, date('11/01/1997')) = 1 FETCH FIRST 3 ROWS ONLY;	ENTRYDATE ----- 01-NOV-1997 01-NOV-1997 01-NOV-1997 3 record(s) selected.
SELECT g.update FROM gen g WHERE BIORS.CONTAINS(g.update, date('11/01/1997')) = 1 FETCH FIRST 3 ROWS ONLY;	UPDATE ----- 01-NOV-1997 11- 01-NOV-1997 12- 01-NOV-1997 06- 3 record(s) selected.

BIORS.CONTAINS_LE および BIOR.S.CONTAINS_GE カスタム関数を使用した照会

以下の表の照会と結果は、BIORS.CONTAINS_LE および BIOR.S.CONTAINS_GE カスタム関数の使用方法を示します。

表 15. BIOR.S.CONTAINS_LE および BIOR.S.CONTAINS_GE カスタム関数を使用した照会の例

照会	結果
SELECT s.molweight FROM Swiss s WHERE BIORS.CONTAINS_LE(s.molweight, 100368) = 1 FETCH FIRST 3 ROWS ONLY;	MOLWEIGHT ----- 100368 10576 8523 3 record(s) selected.
SELECT s.molweight FROM Swiss s WHERE BIORS.CONTAINS_GE(s.molweight, 100368) = 1 FETCH FIRST 3 ROWS ONLY;	MOLWEIGHT ----- 100368 103625 132801 3 record(s) selected.
SELECT s.journalissue FROM Swiss s WHERE BIORS.CONTAINS_GE(s.journalissue, 172) = 1 FETCH FIRST 3 ROWS ONLY;	JOURNALISSUE ----- 172 21 242 196 3 record(s) selected.

BIORS.SEARCH_TERM カスタム関数を使用した照会

以下の表の照会と結果は、BioRS 形式を使用した検索語の指定に BIOR.S.SEARCH_TERM カスタム関数を使用する方法を示します。

表 16. BIOR.S.SEARCH_TERM カスタム関数を使用した照会の例

照会	結果
SELECT s.publicationyear FROM Swiss s WHERE BIORS.SEARCH_TERM (s.id, '[PublicationYear EQ number:1997;]')=1 FETCH FIRST 10 ROWS ONLY;	PUBLICATIONYEAR ----- 1997 1997 2000 1988 1991 1997 1994 1997 1997 1998 1994 1995 1997 1997 1999 1997 1994 1994 1995 1993 1992 1997 10 record(s) selected.
SELECT s.molweight FROM Swiss s WHERE BIORS.SEARCH_TERM (s.id, '[MolWeight EQ number:100368;]') = 1 FETCH FIRST 10 ROWS ONLY;	MOLWEIGHT ----- 100368 100368 2 record(s) selected.

表 16. *BIORS.SEARCH_TERM* カスタム関数を使用した照会の例 (続き)

照会	結果
SELECT s.molweight FROM Swiss s WHERE BIORS.SEARCH_TERM (s.id, '[MolWeight GREATER number:100368;]') = 1 FETCH FIRST 10 ROWS ONLY;	MOLWEIGHT ----- 100368 103625 132801 194328 130277 287022 289130 135502 112715 112599
	10 record(s) selected.

関係述部を使用して、親子関係のある 2 つのデータ・バンク間で等価結合を構成する

以下の照会は、関係述部を使用して、親子関係のある 2 つのデータ・バンク間で等価結合を構成する方法を示します。

```
SELECT s.id, f.id, f.parent FROM Swiss s, Swissft f
WHERE (f.parent = s.id) FETCH FIRST 10 ROWS ONLY;
```

次の照会結果では、100K_RAT レコードが 9 つの子レコードの親になります (100K_RAT.1 から 100K_RAT.9)。

ID	ID	PARENT
100K_RAT	100K_RAT.1	swissprot:100K_RAT
100K_RAT	100K_RAT.2	swissprot:100K_RAT
100K_RAT	100K_RAT.3	swissprot:100K_RAT
100K_RAT	100K_RAT.4	swissprot:100K_RAT
100K_RAT	100K_RAT.5	swissprot:100K_RAT
100K_RAT	100K_RAT.6	swissprot:100K_RAT
100K_RAT	100K_RAT.7	swissprot:100K_RAT
100K_RAT	100K_RAT.8	swissprot:100K_RAT
100K_RAT	100K_RAT.9	swissprot:100K_RAT
104K_THEPA	104K_THEPA.1	swissprot:104K_THEPA

10 record(s) selected.

BioRS ラッパー・パフォーマンスの最適化

BioRS データ・ソースに対する照会のパフォーマンスは、BioRS ラッパーのパフォーマンスを最適化することによって改善できます。

BioRS ラッパー・パフォーマンスの最適化方法

照会の構造と BioRS データ・バンクについての統計情報は、照会のパフォーマンスに影響します。

検索エンジン間の転送データ量を最小化する。

フェデレーテッド環境は 2 つの検索エンジンを使用します。BioRS ラッパーの場合、検索エンジンはフェデレーテッド・データベースと BioRS です。構成によっては、複数のフェデレーテッド・データベース・エンジンが含まれる場合もあります。フェデレーテッド・データベース・エンジンは二

ニックネーム列に指定された述部 (=、BETWEEN、LIKE、および <> などの関係演算子) を処理します。BioRS エンジンは、BioRS ラッパーの 4 つのカスタム関数を使用して指定された述部を処理します。

2 つの検索エンジン間で転送するデータ量を最小化するには、可能な場合は常にデータ処理を BioRS システムにプッシュダウンするよう照会を構成します。

照会で結合操作を行う必要がある場合、可能な場合は BioRS データ・バンクの既存の親子関係を活用して等価結合操作を行います。等価結合操作は BioRS で処理され、ここでもフェデレーテッド・データベースと BioRS 検索エンジン間のデータ転送量を最小化できます。

重要: BioRS へのフェデレーテッド照会を中断しないでください (例えば、コマンド行プロセッサで **Ctrl-D** または **Ctrl-Z** を使用する、あるいはアプリケーション・プログラムを停止する、など)。照会を中断すると、BioRS サーバーで実行中の処理がデッド状態になります。これらの「非活動」プロセスは、BioRS サーバーおよびフェデレーテッド・サーバーのパフォーマンスを急激に低下させます。また、「デッド」プロセスの数が増えると、フェデレーテッド照会処理中に予期しないエラーが発生する恐れがあります。例えば、行があると予想される場合に、有効な照会が 0 行を戻す、などです。極端な場合には、BioRS サーバーまたはフェデレーテッド・サーバー、あるいはその両方が停止または異常終了する恐れがあります。

フェデレーテッド環境で BioRS 統計情報を保守する。

フェデレーテッド・システムでは、フェデレーテッド・データベースは、ニックネームが付けられたオブジェクトのカatalog統計に基づいて、照会処理を最適化します。BioRS データ・ソースに関する現行統計の保守は、BioRS ラッパーのパフォーマンス最適化に不可欠です。ニックネームが定義されたリモート・オブジェクトの統計データまたは構造の特性を変更した場合、フェデレーテッド・システムで対応するニックネーム列カーディナリティー統計を更新する必要があります。

BioRS ラッパーのパフォーマンスを最適化するには、これらの更新をフェデレーテッド・サーバーで定期的に行う必要があります。

BioRS 統計情報

現行の BioRS 統計情報は、BioRS ラッパーのパフォーマンスの最適化には不可欠です。

フェデレーテッド・システムでは、フェデレーテッド・データベースは、ニックネームが付けられたオブジェクトのカatalog統計に基づいて、照会処理を最適化します。これらの統計は、CREATE NICKNAME ステートメントを使用したニックネーム登録時に BioRS データ・ソースから抽出されます。フェデレーテッド・データベースはデータ・ソース側にオブジェクトが存在するかチェックし、次に既存のデータ・ソース統計データを収集します。情報はデータ・ソース・カatalogから読み取られ、フェデレーテッド・データベース内のシステム・カatalogに置かれます。

BioRS データ・ソースでは、主要な統計情報は以下のものを含みます。

- ニックネームのカーディナリティー。BioRS データ・ソースでは、ニックネーム・カーディナリティーは対応する BioRS データ・バンクの項目数と同じです。

- BioRS `_ID_` エlementに対応する列のカーディナリティー。この列のカーディナリティーは、列を参照するニックネームのカーディナリティーと一致する必要があります。
- BioRS ラッパーが使用する必要があるすべての列のカーディナリティー。

ヒント: BioRS ラッパーのパフォーマンスを最適化するには、BioRS データ・ソースに関する現行の統計を維持するようにします。ニックネームが定義されたりモート・オブジェクトの統計データまたは構造の特性を変更した場合、フェデレーテッド・システムで対応するカーディナリティー統計を更新する必要があります。カーディナリティー統計は `SYSSTAT.TABLES` ビューおよび `SYSSTAT.COLUMNS` ビューのフェデレーテッド・データベース・システム・カタログに保管されます。

フェデレーテッド・システムの BioRS カーディナリティー統計を保守するには、以下のようにします。

1. 必要に応じて、ニックネームのカーディナリティー統計を確認します。
2. カタログ・ビューで、ニックネームのカーディナリティー統計を更新します。
3. カタログ・ビューで、列のカーディナリティー統計を更新します。

BioRS データ・バンク・カーディナリティー統計の確認

ニックネーム統計を更新、または、BioRS `_ID_` Elementに対応する列のカーディナリティーを更新する前に BioRS データ・バンク・カーディナリティー統計を確定する必要があります。

手順

BioRS `admin_find` ユーティリティー・プログラムまたは `www_find.cgi` ユーティリティー・プログラムのいずれかを使用し、カーディナリティーを有効にする `-c` オプションを指定します。これら 2 つの BioRS ユーティリティー・プログラムの詳細については、ご使用の BioRS の資料を参照してください。

BioRS ニックネーム・カーディナリティー統計の更新

BioRS データ・バンクのコンテンツを大幅に変更する場合、ニックネーム・カーディナリティー統計を更新する必要があります。

始める前に

統計を更新するニックネームに対応する BioRS データ・バンクのカーディナリティー番号を判別してください。

このタスクについて

フェデレーテッド・システムの BioRS ニックネーム・カーディナリティー統計を更新するには、`SYSSTAT.TABLES` カタログ・ビューを変更する必要があります。

ニックネームの正しいカーディナリティー統計を維持することで、オブティマイザーと BioRS ラッパーが最適なデータ・アクセス・プランを選択できます。

手順

UPDATE ステートメントを発行して SYSSTAT.TABLES カタログ・ビューを変更し、正しいカーディナリティー番号を指定します。 UPDATE ステートメントの構文は以下のとおりです。

```
UPDATE SYSSTAT.TABLES SET CARD=cardinality_number
WHERE TABSCHEMA=nickname_schema
AND TABNAME=nickname_name;
```

例えば、ニックネームが JONES.SWISS であれば、以下の UPDATE ステートメントを使用して統計を更新します。

```
UPDATE SYSSTAT.TABLES SET CARD=15312191
WHERE TABSCHEMA='JONES'
AND TABNAME='SWISS';
```

SYSSTAT.TABLES

ニックネーム統計が保管される、フェデレーテッド・データベースのシステム・カタログ・ビュー。

SET CARD=15312191

統計を更新するニックネームの BioRS データ・バンク・カーディナリティー番号。

TABSCHEMA= 'JONES'

更新するニックネームのスキーマの名前。

TABNAME='SWISS'

更新するニックネームの名前。

BioRS 列カーディナリティー統計

フェデレーテッド・システムの BioRS 列カーディナリティー統計を更新するには、SYSSTAT.COLUMNS カタログ・ビューを変更する必要があります。

このタスクについて

BioRS 列カーディナリティー統計の更新は、BioRS ニックネームの作成前に行うことも、BioRS データ・ソースの照会パフォーマンスの向上を図るために行うこともできます。

制約事項

この手順は、BioRS `_ID_` エレメントに対応する列のカーディナリティー統計の更新には使用しないでください。

オプティマイザーおよび BioRS ラッパーが照会処理時に最良のデータ・アクセス・プランを選択できるよう、BioRS 列のカーディナリティー統計が最新のものであることを確認してください。

手順

UPDATE ステートメントを発行して、SYSSTAT.COLUMNS カタログ・ビューを変更します。 UPDATE ステートメントの構文は以下のとおりです。

```
UPDATE SYSSTAT.COLUMNS SET COLCARD=(SELECT COUNT(DISTINCT column_name)
FROM nickname_schema.nickname_name)
WHERE
TABSCHEMA=nickname_schema
AND TABNAME=nickname_name
AND COLNAME=column_name;
```

SYSSTAT.COLUMNS

列統計が保管される、フェデレーテッド・データベースのシステム・カタログ・ビュー。

COLCARD=(SELECT COUNT(DISTINCT column_name
統計を更新するニックネームの列名。

TABSCHEMA= nickname_schema
更新するニックネームのスキーマの名前。

TABNAME=nickname_name
更新するニックネームの名前。

COLNAME=column_name
更新対象のカーディナリティー統計を持つ列の名前。

ニックネームに関連するデータ・バンクの項目をすべて検索するため、照会の実行には数分かかる場合があります。

列に複数の値を含む場合 (SwissProt データベース形式の PublicationYear エレメントなど)、SQL 照会を使用するには計算が複雑になり過ぎます。このような列には、カーディナリティー値を手動で計算後、SYSSTAT.COLUMNS カタログ・ビューを更新する必要があります。カーディナリティー値を計算するには、列の別個の値を、行ごとの値の平均値で割ります。計算したカーディナリティー値は、表のカーディナリティーより小さくなります。

例えば、ニックネームが以下に示す 3 行の値を PublicationYear 列に持つ場合、互いに異なる 9 種類の値があり、1 行あたりの値の個数の平均は 4 です。

- 1997 1992 1985
- 1997 1992 1982
- 1992 1991 1990 1976 1974 1971

PublicationYear 列のカーディナリティーは 9/4 すなわち 3 (2.25 をすぐ上の整数に切り上げ) です。次の UPDATE ステートメントを使用して、SYSSTAT.COLUMNS カタログ・ビューを更新します。

```
UPDATE SYSSTAT.COLUMNS SET CARDCOL=3
WHERE
TABSCHEMA=nickname_schema
AND TABNAME=nickname_name
AND COLNAME=column_name;
```

BioRS _ID_ 列カーディナリティーの更新

BioRS _ID_ エレメントにマップする列に対応する BioRS 列カーディナリティー統計を更新するには、SYSSTAT.COLUMNS カタログ・ビューを変更する必要があります。

始める前に

列を参照するニックネームに対応する BioRS データ・バンクのカーディナリティー番号を判別してください。BioRS _ID_ エlementにマップする列のカーディナリティー番号は、列を参照するニックネームのカーディナリティーと一致する必要があります。

BioRS _ID_ エlementにマップする列に対応するカーディナリティー統計は、常に最新であることを確認するようにします。オプティマイザーおよび BioRS ラッパーは、これらの統計を、照会処理のための最良のデータ・アクセス・プランを選択するのに使用します。

このタスクについて

BioRS _ID_ 列カーディナリティーを更新するには、ELEMENT_NAME オプションを含む SYSCAT.COLOPTIONS ビュー中の項目を選択する必要があります。BioRS ラッパーは、このオプションを、フェデレーテッド・データベースのニックネーム列名と BioRS サーバーのエlement名のマッピングに使用します。

手順

UPDATE ステートメントを発行してカタログ・ビューを変更します。

例:

```
UPDATE SYSSTAT.COLUMNS SET COLCARD=cardinality_number
WHERE
  TABSCHEMA=nickname_schema
  AND TABNAME=nickname_name
  AND COLNAME=column_name
  IN (SELECT column_name FROM SYSCAT.COLOPTIONS
      WHERE
        TABSCHEMA=nickname_schema
        AND TABNAME=nickname_name
        AND OPTION='ELEMENT_NAME'
        AND SETTING='_ID_')
```

SYSSTAT.COLUMNS

列統計が保管される、フェデレーテッド・データベースのシステム・カタログ・ビュー。

SET COLCARD=*cardinality_number*

統計を更新する列のニックネームに対応する BioRS データ・バンク・カーディナリティー番号。

TABSCHEMA= *nickname_schema*

更新するニックネームのスキーマの名前。

TABNAME=*nickname_name*

更新するニックネームの名前。

COLNAME=*column_name*

更新対象のカーディナリティー統計を持つ列の名前。

IN (SELECT *column_name* FROM SYSCAT.COLOPTIONS

この SELECT ステートメントは、BioRS _ID_ エlementにマップする列の名前を決定します。SYSCAT.COLOPTIONS は、列オプションが保管されるフェデレーテッド・データベースのシステム・カタログ・ビューです。

OPTION='ELEMENT_NAME'

ニックネーム列名が BioRS エlement名にマップされることを示す、
SYSCAT.COLOPTIONS ビューの行の値。

SETTING='_ID_'

ELEMENT_NAME オプションに対応する列は '_ID_' であると指定します。

DB2 データ・ソースへのアクセスの構成

DB2 ファミリー・データ・ソースにアクセスするフェデレーテッド・サーバーを構成するには、アクセスするデータ・ソースとオブジェクトに関する情報をフェデレーテッド・サーバーに提供する必要があります。

始める前に

- フェデレーテッド・サーバーのセットアップを検証します。

手順

1. DB2 ノード項目のカタログ
2. リモート DB2 データベースのカタログ
3. DB2 ラッパーの登録
4. DB2 データ・ソースのサーバー定義の登録
5. DB2 データ・ソースのユーザー・マッピングの作成
6. DB2 データ・ソース・サーバーとの接続のテスト
7. DB2 表およびビューのニックネームの登録

DB2 ノード項目のカタログ

フェデレーテッド・サーバーで DB2 データ・ソースとの接続に使用されるプロトコルを指定するノード項目をカタログする必要があります。

手順

CATALOG TCPIP NODE コマンドを発行します。

例:

```
CATALOG TCPIP NODE db2_node REMOTE system42 SERVER db2tcp42
```

ここで、

- *db2_node* はノードに割り当てる名前
- *system42* はデータ・ソースが含まれるシステムのホスト名
- *db2tcp42* は、サーバー・データベース・マネージャー・インスタンスのサービス名または 1 次ポート番号です。

リモート DB2 データベースのカタログ

フェデレーテッド・サーバーの接続先データベースを特定するには、フェデレーテッド・サーバー・システム・データベース・ディレクトリーの中にリモート DB2 データベースをカタログする必要があります。

手順

1. 構成アシスタントを使用することによってデータベースをカタログするか、または **CATALOG DATABASE** コマンドを発行します。

たとえば、次のようにします。

```
CATALOG DATABASE DB2DB390 AS CLIENTS390 AT NODE DB2NODE AUTHENTICATION SERVER
```

ここで、

- **DB2DB390** は、カタログするリモート・データベースの名前
 - **CLIENTS390** は、カタログするリモート・データベースのための別名です。別名を指定しない場合、データベース・マネージャーは、リモート・データベースの名前を別名として使用します。
 - **DB2NODE** は、それ以前にカタログしたノードの名前です。
 - **AUTHENTICATION SERVER** は、認証が DB2 データ・ソース・ノード側で実行されることを指定します。
2. リモート・データベースの名前の長さが 8 文字を超える場合は、**CATALOG DCS DATABASE** コマンドを発行することによって DCS ディレクトリー・エントリーを作成してください。

例:

```
CATALOG DCS DATABASE SALES400 AS SALES_DB2DB400
```

ここで、

- **SALES400** はカタログするリモート・データベースの別名です。この別名は、リモート・ノードに関連するフェデレーテッド・サーバー・システム・データベース・ディレクトリーの中のエントリーの名前と一致していなければなりません。この別名は、**CATALOG DATABASE** コマンドに指定する名前と同じです。
- **SALES_DB2DB400** は、カタログするターゲット・ホスト・データベースの名前です。

DB2 ラッパーの登録

DB2 ファミリー・データ・ソースにアクセスするラッパーを登録する必要があります。フェデレーテッド・サーバーは、そのラッパーを使用することにより、データ・ソースと通信したりデータ・ソースからデータを取り出したりします。ラッパーは、一連のライブラリー・ファイルとしてインプリメントされます。

手順

ラッパーを登録するには、以下の方法のいずれかを使用します。

方法	説明
Data Studio の「管理エクスプローラー (Administration Explorer)」を使用します。	「フェデレーテッド・データベース・オブジェクト」フォルダーで、「ラッパー (Wrappers)」フォルダーを右クリックし、「ラッパーの作成 (Create Wrapper)」を選択します。

方法	説明
<p>CREATE WRAPPER ステートメントを発行し、ラッパーの名前を指定します。</p>	<p>DB2 ファミリー・データ・ソースのデフォルトのラッパー名は DRDA です。例:</p> <pre>CREATE WRAPPER DRDA</pre> <p>ラッパーを登録するためにデフォルトの名前を使用する場合、ライブラリー名を指定する必要はありません。その場合、ラッパーに関連するデフォルトのライブラリー名がフェデレーテッド・サーバーにより自動的に使用されるからです。ラッパー名がフェデレーテッド・データベース中の既存のラッパー名と競合する場合には、任意の名前を選択してデフォルトのラッパー名と置き換えることができます。しかし、そうする場合、CREATE WRAPPER ステートメントに LIBRARY パラメーターを含める必要があります。</p> <p>例えば、AIX オペレーティング・システムが使用されているフェデレーテッド・サーバーで db2_wrapper という名前のラッパーを登録するには、次のステートメントを発行します。</p> <pre>CREATE WRAPPER db2_wrapper LIBRARY 'libdb2drda.a'</pre>

デフォルトのライブラリー名は、フェデレーテッド・サーバーのオペレーティング・システムに固有のものです。詳細は、DB2 ラッパー・ライブラリー・ファイルのリストを参照してください。

DB2 ラッパー・ライブラリー・ファイル

フェデレーテッド・サーバーをインストールすると、デフォルト・ディレクトリー・パスにラッパー・ライブラリー・ファイルが追加されます。

ラッパーを登録する際にデフォルト・ラッパー名を使用しない場合は、**CREATE WRAPPER** ステートメントに **LIBRARY** パラメーターを含め、デフォルト・ラッパー・ライブラリー・ファイル名を指定する必要があります。

以下の表に、デフォルト・ディレクトリー・パスとデフォルト・ラッパー・ライブラリー・ファイル名のリストを示します。この表において、*install_path* は UNIX または Linux におけるフェデレーション・サーバーのインストール先ディレクトリー・パス、そして *%DB2PATH%* は Microsoft Windows におけるフェデレーション・サーバーのインストール先ディレクトリー・パスを指定する環境変数です。Windows でのデフォルトのディレクトリー・パスは *C:\Program Files\IBM\sqllib* です。

表 17. オペレーティング・システム別の DB2 ディレクトリー・パスおよびライブラリー・ファイル名

オペレーティング・システム	ディレクトリー・パス	ライブラリー・ファイル名
AIX	/usr/opt/install_path/lib32/ /usr/opt/install_path/lib64/	libdb2drda.a
HP-UX	/opt/IBM/db2/install_path/lib32 /opt/IBM/db2/install_path/lib64	libdb2drda.so
Linux	/opt/IBM/db2/install_path/lib32 /opt/IBM/db2/install_path/lib64	libdb2drda.so
Solaris	/opt/IBM/db2/install_path/lib32 /opt/IBM/db2/install_path/lib64	libdb2drda.so
Windows	%DB2PATH%\bin	db2drda.dll

オペレーティング・システムごとに、デフォルト・ライブラリー・ファイルが 1 個、そして特定のラッパー・オプションが指定された場合のみ使用される付加的なファイルが 2 個、合計 3 個のライブラリー・ファイルがインストールされます。

DB2 データ・ソースのサーバー定義の登録

フェデレーテッド・サーバーでは、各 DB2 サーバーに接続するために許可情報とパスワード情報が必要になります。その許可情報およびパスワード情報はグローバル・カタログには保管されていないため、それを各サーバー定義に含める必要があります。

手順

サーバー定義を登録するには、以下のいずれかの方法を使用します。

方法	説明
Data Studio の「管理エクスプローラー(Administration Explorer)」を使用します。	「フェデレーテッド・データベース・オブジェクト」フォルダーで、「リモート・サーバー (Remote Servers)」フォルダーを右クリックし、「リモート・サーバーの作成 (Create Remote Servers)」を選択します。
CREATE SERVER ステートメントを発行します。	<p>サーバーを登録するには、特定の必須サーバー・オプションを含める必要があります。この例には、DB2 サーバーの登録に必要なサーバー・オプションだけが含まれています。</p> <pre>CREATE SERVER server_definition_name TYPE server_type VERSION version_number WRAPPER DRDA AUTHORIZATION "userid" PASSWORD "password" OPTIONS (DBNAME 'database_name')</pre> <p>DBNAME は必須サーバー・オプションです。DBNAME の値は、アクセス対象の DB2 データベースの別名です。この別名は、データベースをカタログする際に定義します。</p>

注: DB2 for z/OS バージョン 8 を使用して互換モードでデータベースを作成した場合、VERSION には、バージョン 7 を指定する必要があります。

サーバーを登録する際、CREATE SERVER ステートメントに付加的なサーバー・オプションを指定できます。それらのオプションには、汎用サーバー・オプションと、データ・ソースに固有のサーバー・オプションとが含まれます。InfoSphere Federation Server バージョン 9.7 フィックスパック 2 以降では、CREATE SERVER ステートメントを実行すると、以下のサーバー・オプションがデータ・ソースの構成に基づいて自動的に構成されます。

- DATE_COMPAT
- NUMBER_COMPAT
- SAME_DECFLT_ROUNDING
- VARCHAR2_COMPAT

これらのサーバー・オプションを手動で構成しようとする、SQL1841N メッセージを受け取ります。

CREATE SERVER ステートメントで指定できるオプションの詳細については、DB2 オプションのリファレンス情報を参照してください。

次のタスク

サーバーを登録した後で、サーバー・オプションを追加したり既存のサーバー・オプションを削除したりするには、ALTER SERVER ステートメントを使用します。

CREATE SERVER ステートメント - DB2 ラッパーの場合の例

CREATE SERVER ステートメントは、DB2 サーバー定義を登録するために使用します。このトピックには、必須指定のオプションを指定した完結した例、また付加的なサーバー・オプションの使用法を示す例が含まれています。

完全指定の例

以下の例は、CREATE SERVER ステートメントを使用することによって DRDA ラッパーのサーバー定義を登録する方法を示すものです。

```
CREATE SERVER DB2SERVER TYPE DB2/ZOS VERSION 7 WRAPPER DRDA
AUTHORIZATION "spalten" PASSWORD "db2guru"
OPTIONS (DBNAME 'CLNTS390')
```

DB2SERVER

DB2 データベース・サーバーに割り当てる名前。重複するサーバー定義名は使用できません。これは、必須のサーバー・オプションです。

TYPE *DB2/ZOS*

アクセスを構成するデータ・ソース・サーバーのタイプを指定します。

VERSION 7

アクセスする DB2 データベース・サーバーのバージョン。

注: 互換モードでデータベースを作成するために DB2 for z/OS バージョン 8 を使用した場合は、バージョン 7 を指定する必要があります。

WRAPPER *DRDA*

CREATE WRAPPER ステートメントで指定した名前。

AUTHORIZATION "spalten"

データ・ソースでの許可 ID。この ID には、データ・ソースにおける BINDADD 権限が付与されていなければなりません。この値は大文字と小文字を区別します。

PASSWORD "db2guru"

データ・ソースで許可 ID に関連付けるパスワード。この値は大文字と小文字を区別します。

DBNAME 'CLNTS390'

アクセスする DB2 データベースの別名。この別名は、**CATALOG DATABASE** コマンドを使用することによってデータベースをカタログした際に定義したものです。この値は大文字と小文字を区別します。

データベース名変数は CREATE SERVER ステートメントのオプションとして指定されますが、DB2 データ・ソースに関しては、これは必須です。

サーバー・オプションの例

サーバー定義を登録する際、CREATE SERVER ステートメントに付加的なサーバー・オプションを指定できます。それらのオプションには、汎用サーバー・オプションと、DB2 データ・ソース固有のサーバー・オプションとが含まれます。詳細については、オプションのリファレンス情報を参照してください。

CPU_RATIO オプションは、データ・ソース CPU の実行速度がフェデレーテッド CPU に比べてどれだけ速いか (または遅いか) を示します。CPU_RATIO オプションを 0.001 に設定した場合、それは、リモート・データ・ソース側の CPU の能力がフェデレーテッド・サーバー CPU の 1000 倍であるということになります。

例：

```
CREATE SERVER DB2SERVER TYPE DB2/CS VERSION 9.7 WRAPPER DRDA
AUTHORIZATION "spalten" PASSWORD "db2guru"
OPTIONS (DBNAME 'CLNTS390', CPU_RATIO '0.001')
```

DB2 データ・ソースのユーザー・マッピングの作成

ユーザー・マッピングは、フェデレーテッド・サーバー側のユーザー ID およびパスワードと、データ・ソース・サーバー側の対応するユーザー ID およびパスワードとの間の関連を定義します。

このタスクについて

DB2 データ・ソースのためにユーザー・マッピングが必須かどうかは、フェデレーテッド環境の構成によって異なります。フェデレーテッド・トラステッド・コンテキストおよびプロキシ認証を使用する環境の場合、必要となるユーザー・マッピングは全くないか、または少数だけです。最善の結果を得るには、ユーザー・マッピングを作成する前に、十分にフェデレーテッド・トラステッド・コンテキストの計画を立ててセットアップしてください。

手順

Oracle データ・ソースのユーザー・マッピングを作成するには、以下の方法のいずれかを使用します。

方法	説明
Data Studio の「管理エクスプローラー (Administration Explorer)」を使用します。	「フェデレーテッド・データベース・オブジェクト」フォルダーで、「ユーザー・マッピング (User Mappings)」フォルダーを右クリックし、「ユーザー・マッピングの作成 (Create User Mapping)」を選択します。
CREATE USER MAPPING ステートメントを発行して、ローカル・ユーザー ID を DB2 サーバー・ユーザー ID とパスワードにマップします。	例: <pre>CREATE USER MAPPING FOR local_userID SERVER server_definition_name OPTIONS (REMOTE_AUTHID 'remote_userID', REMOTE_PASSWORD 'remote_password')</pre> <p>REMOTE_AUTHID は、バインド許可 ID ではなく接続許可 ID です。</p>

CREATE USER MAPPING ステートメント - DB2 ラッパーの場合の例

CREATE USER MAPPING ステートメントは、フェデレーテッド・サーバーの許可 ID を、DB2 サーバーのユーザー ID およびパスワードにマップするために使用します。

完全指定の例

以下の例は、フェデレーテッド・サーバーの許可 ID を、リモート DB2 のユーザー ID およびパスワードにマップする方法を示すものです。

```
CREATE USER MAPPING FOR ALONZO SERVER DB2SERVER
OPTIONS (REMOTE_AUTHID 'al', REMOTE_PASSWORD 'day2night')
```

ALONZO

ローカルの許可 ID をリモート・ユーザー ID およびパスワードにマップします。

SERVER DB2SERVER

CREATE SERVER ステートメントで定義した DB2 ファミリー・データ・ソース・サーバーの名前を指定します。

REMOTE_AUTHID 'al'

ALONZO のマッピング先となる DB2 ファミリー・データ・ソース・サーバー側の接続ユーザー ID を指定します。CREATE SERVER ステートメントで FOLD_ID サーバー・オプションを「U」または「L」に設定している場合を除き、この値は大文字小文字を区別します。

REMOTE_PASSWORD 'day2night'

'al' に関連するパスワードを指定します。CREATE SERVER ステートメントで FOLD_PW サーバー・オプションを「U」または「L」に設定している場合を除き、この値は大文字小文字を区別します。

特殊レジスターの例

以下に、特殊レジスター USER を含む CREATE USER MAPPING ステートメントの例を示します。

```
CREATE USER MAPPING FOR USER SERVER DB2SERVER
  OPTIONS (REMOTE_AUTHID 'al', REMOTE_PASSWORD 'day2night')
```

DB2 特殊レジスター USER を使用することにより、CREATE USER MAPPING ステートメントを発行するユーザーの許可 ID を、REMOTE_AUTHID ユーザー・オプションで指定されているデータ・ソース・ユーザー ID にマップできます。

パブリック・ユーザー・マッピングの例

InfoSphere Federation Server バージョン 9.7 フィックスバック 2 以降では、すべてのローカル・データベース・ユーザーが 1 つのリモート・ユーザー ID とパスワードでデータ・ソースにアクセスできるようにするパブリック・ユーザー・マッピングを作成できます。

この例では、まず DRDA ラッパーおよび DB2 for z/OS データ・ソース・サーバー SERVER390 が作成されます。次に、CREATE USER MAPPING ステートメントが PUBLIC と共に指定され、サーバー SERVER390 へのパブリック・ユーザー・マッピングが作成されます。ローカル・データベース・ユーザーすべてを、サーバー SERVER390 の許可 ID にマップするために、CREATE SERVER ステートメントで指定される許可 ID は CREATE USER MAPPING ステートメントの OPTIONS でも指定されます。

例

```
CREATE WRAPPER DRDA;

CREATE SERVER SERVER390
  TYPE DB2/ZOS VERSION 7.1 WRAPPER DRDA
  AUTHORIZATION "APP_USER" PASSWORD "secret"
  OPTIONS (DBNAME 'remotedb');

CREATE USER MAPPING FOR PUBLIC SERVER SERVER390
  OPTIONS (REMOTE_AUTHID 'APP_USER', REMOTE_PASSWORD 'secret');
```

DB2 データ・ソース・サーバーとの接続のテスト

DB2 データ・ソース・サーバーとの接続をテストし、DB2 データ・ソース・サーバーにアクセスするよう、フェデレーテッド・サーバーが適切に構成されているかどうかを確認します。

手順

DB2 サーバーとの接続をテストするには、パススルー・セッションを開き、DB2 システム表に対して SQL SELECT ステートメントを発行します。

DB2 データ・ソース	例
DB2 for z/OS	SET PASSTHRU <i>server_definition_name</i> SELECT count(*) FROM sysibm.systables SET PASSTHRU RESET

DB2 データ・ソース	例
DB2 for Linux, UNIX and Windows	SET PASSTHRU <i>server_definition_name</i> SELECT count(*) FROM syscat.systables SET PASSTHRU RESET
DB2 for System i	SET PASSTHRU <i>server_definition_name</i> SELECT count(*) FROM qsys2.systables SET PASSTHRU RESET

SQL SELECT ステートメントがカウント結果を戻したなら、データ・ソースへのアクセスは適切に構成されています。

データ・ソース接続エラーのトラブルシューティング

データ・ソース・サーバーへのテスト接続では、いくつかの理由でエラーが戻されることがあります。以下の処置により、エラーが発生した理由を判別することができます。

症状

データ・ソースに接続しようとしたときにエラーが戻されます。

原因

接続の問題には、いくつかの原因が考えられます。

問題の解決方法

データ・ソース接続エラーのトラブルシューティングを行うには、以下の項目で問題がないかどうかをチェックします。

- データ・ソースが使用可能であることを確認します。
- 該当する場合は、データ・ソース・サーバーが着信接続に対して構成されていることを確認します。
- REMOTE_AUTHID と REMOTE_PASSWORD のオプションに関するユーザー・マッピング設定を調べ、データ・ソースとの接続に有効な設定になっているか確認します。必要に応じて、ユーザー・マッピングを変更するか、別のユーザー・マッピングを作成します。
- 該当する場合は、フェデレーテッド・サーバー上のデータ・ソース・クライアント・ソフトウェアがインストールされていて、データ・ソースに接続できるように正しく構成されていることを確認します。
- ODBC データ・ソースの場合、フェデレーテッド・サーバー上の ODBC ドライバーがインストールされていて、ODBC データ・ソース・サーバーに接続できるように正しく構成されていることを確認します。Windows が稼働するフェデレーテッド・サーバーでは、ドライバーをチェックするときに「ODBC データ ソース アドミニストレータ」ツールを使用します。UNIX が稼働するフェデレーテッド・サーバーの場合は、ODBC クライアントのベンダーが提供している資料を参照してください。
- フェデレーテッド・サーバー上で設定されている変数の設定がデータ・ソースについて適切であることを確認します。これらの変数には、システム環境変数、db2dj.ini ファイル、および DB2 Profile Registry (db2set) 変数が含まれます。

- サーバー定義を調べます。必要であれば、サーバー定義をドロップして再作成します。

DB2 表およびビューのニックネームの登録

DB2 サーバー定義ごとに、アクセス対象の表およびビューのそれぞれについてニックネームを登録します。この操作をした後、DB2 データベースに対する照会を実行する際には、データ・ソース・オブジェクトの名前ではなく、このニックネームを使用します。

始める前に

ニックネームを登録する前に、DB2 **RUNSTATS** コマンドを使用することによって、DB2 データ・ソースにおいて統計情報を更新してください。データ・ソース統計情報は、フェデレーテッド・サーバーが照会処理を最適化するのに使用されます。

このタスクについて

ニックネームを作成する場合、固有の名前およびリモート・オブジェクトの名前（例えば、表またはビューの名前）を指定します。DB2 Database for Linux, UNIX, and Windows の場合は、表、ビュー、またはニックネームの別名も指定できます。DB2 for z/OS または DB2 for i/OS の場合は、表またはビューの別名を指定できます。

制約事項

DB2 データベース別名に対してニックネームを作成することはできません。

手順

ニックネームを登録するには、以下の方法のいずれかを使用します。

方法	説明
Data Studio の「管理 エクスプローラー (Administration Explorer)」を使用しま す。	「フェデレーテッド・データベース・オブジェクト」フォルダー で、「ニックネーム (Nicknames)」フォルダーを右クリックし、 「ニックネームの作成 (Create Nickname)」を選択します。
CREATE NICKNAME ステート メントを発行します。	例: <pre>CREATE NICKNAME <i>nickname</i> FOR <i>server_definition_name</i>."<i>remote_schema</i>."<i>remote.table</i> ;</pre>

ニックネームを作成すると、フェデレーテッド・サーバーはニックネームを使用してデータ・ソース・カタログを照会します。この照会は、データベース・ソースの表またはビューへの接続をテストします。接続ができない場合、エラー・メッセージが出ます。

次のタスク

この作業を完了すると、ニックネームを使用してデータ・ソース・オブジェクトへのアクセスや、他にもテンポラル表でニックネームを作成するなど、さまざまな操作を実行できます。

CREATE NICKNAME ステートメント - DB2 データ・ソースの場合の例

CREATE NICKNAME ステートメントとその必須ニックネーム・オプションは、アクセスする DB2 表またはビューのためのニックネームを登録するために使用します。

この例は、CREATE NICKNAME ステートメントの必須指定のオプションの使い方を示すものです。さらにいくつかのニックネームおよび列オプションを含めることもできます。

```
CREATE NICKNAME DB2SALES FOR DB2SERVER.VINNIE.EUROPE
```

DB2SALES

DB2 表またはビューを特定する固有のニックネーム。このニックネームには、スキーマとニックネームの両方を含めることができます。スキーマを省略した場合、このニックネームを登録するユーザーの許可 ID が使用されます。

DB2SERVER.VINNIE.EUROPE

リモート・オブジェクトを表す、3 つの部分からなる ID。

- *DB2SERVER* は、CREATE SERVER ステートメントで DB2 データベース・サーバーに割り当てた名前です。
- *VINNIE* は、表またはビューの所有者のユーザー ID です。この値は大文字と小文字を区別します。
- *EUROPE* は、アクセスするリモート表またはビューの名前です。

Excel データ・ソースへのアクセスの構成

フェデレーテッド・システムを使用すると、Excel データ・ソースにあるデータと他のソースからの情報を統合させることができます。

このタスクについて

フェデレーテッド・サーバーが Excel データ・ソースにアクセスするように構成するには、アクセスするデータ・ソースとオブジェクトの情報をフェデレーテッド・サーバーに渡す必要があります。フェデレーテッド・サーバーを構成したら、照会を作成して Excel データ・ソースにアクセスすることができます。

Excel ラッパー

Excel ワークブックとは、Microsoft Excel アプリケーションを使用して作成されるファイルで、xls というファイル拡張子を持ちます。Excel ラッパーは、Excel ファイルで検索を実行するのに使用します。

Excel ファイルは、対応する行と列が含まれる表として表示するのが最適な情報を保管するのに使用します。Excel ワークブックは 1 つ以上のスプレッドシート・ページ (ワークシート) で構成されます。ワークシートはしばしば、計算を実行するために使用されます。

以下の図は、Excel ラッパーがワークシートをフェデレーテッド・システムに接続する方法を示しています。

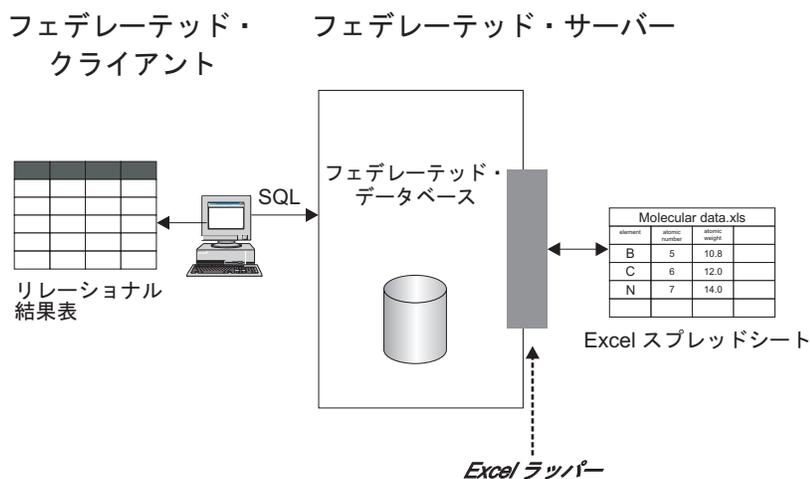


図 6. Excel ラッパーがどのように機能するか

Excel ラッパーは CREATE NICKNAME ステートメントを使用して、Excel ワークシート内の列をフェデレーテッド・システム内の列にマップします。以下の表は、Compound_Master.xls というファイルに保管されているワークシート・データのサンプルを示しています。

表 18. Compound_Master.xls のサンプル・ワークシート・データ

	A	B	C	D
1	COMPOUND_NAME	WEIGHT	MOL_COUNT	WAS_TESTED
2	compound_A	1.23	367	tested
3	compound_G		210	
4	compound_F	0.000425536	174	tested
5	compound_Y	1.00256		tested
6	compound_Q		1024	
7	compound_B	33.5362		
8	compound_S	0.96723	67	tested
9	compound_O	1.2		tested

通常、Excel ワークシート内の情報は、標準 SQL コマンドでは使用できません。Excel ラッパーがフェデレーテッド・サーバーにインストールされて登録されると、この情報が典型的なリレーショナル・データ・ソースであるかのようにアクセスすることができます。例えば、分子数が 100 より大きいすべての化合物データを知る必要がある場合、以下の SQL 照会を実行します。

```
SELECT * FROM compound_master WHERE mol_count > 100
```

照会の結果が、以下の表に示されています。

表 19. 照会の結果

COMPOUND_NAME	WEIGHT	MOL_COUNT	WAS_TESTED
compound_A	1.23	367	tested
compound_G		210	
compound_F	0.000425536	174	tested
compound_Q		1024	

Excel データにアクセスする方法

Excel ラッパーまたは ODBC ラッパーのどちらかを使用して、Microsoft Excel ワークシートのデータにアクセスできます。

Excel データを照会するには、両方のラッパーとも、Excel ワークブック中のワークシートを開いて読み取れるフェデレーテッド・サーバーが必要です。そのため、Excel ワークブックは、フェデレーテッド・サーバーと同じコンピューター上か、またはネットワーク・アクセス可能なドライブ上になければなりません。

Excel ラッパーを使用する場合は、Excel アプリケーションがフェデレーテッド・サーバー上にインストールされていなければなりません。

ODBC ラッパーを使用する場合は、Excel ODBC ドライバーがフェデレーテッド・サーバー上になければなりません。このドライバーは、Microsoft Windows® と共に自動的にインストールされます。Excel アプリケーションをフェデレーテッド・サーバー上にインストールする必要はありません。

各ラッパーには、Excel ワークブック中のデータの場所とレイアウトに関するいくつかの要件があります。Excel ラッパーを使用する場合は、ワークブック中の最初のワークシート中のデータしかアクセスできません。ODBC ラッパーを使用する場合は、ワークブック中のどのワークシートのデータでもアクセスできます。

以下の例は、これらの 2 つのラッパーに関するワークシートのレイアウトの要件を示しています。

複数のラベル行と数式があるワークシートの例

この例は、ワークシートの上部に複数のラベル行があり、ブランク行を含み、13 行目に数式がある例を示しています。このワークシートのデータにアクセスするには、アクセスしたいセルの範囲を示さなければなりません。

	A	B	C	D
1	Compound Analysis			
2				
3	Compound Name	Weight	Molecular Count	Tested?
4	compound_A	1.23	367	tested
5	compound_G		210	
6	compound_F	0.000425536	174	tested
7	compound_Y	1.000256		tested
8	compound_Q		1024	
9	compound_B	33.5362		
10	compound_S	0.96723	67	tested
11	compound_O	1.2		tested
12				
13	Total Compounds Tested			5

図7. 複数のラベル行と数式を含むワークシート

Excel ラッパーを使用する場合

CREATE NICKNAME ステートメント中でセルの範囲を指定するには、RANGE オプションを使用します。指定する範囲には、データだけを含めてください。列ラベルをこの範囲に含めないでください。SUM などの数式を含むセルは、数式ではなく数式の結果を戻します。数式の結果を戻したいのでなければ、数式を含むセルをこの範囲に含めないでください。この例では、セルの範囲として A4:D11 を RANGE オプションに含めます。

ODBC ラッパーを使用する場合

セルの範囲に名前を作成して、ワークシート中のデータの場所を明示的に指定しなければなりません。Excel は、このセルの範囲を名前付き範囲として参照します。Excel ODBC ドライバーは、範囲内の先頭行である 1 行のラベル行のみ認識します。ラベルとデータの間には空白行は許されません。名前付き範囲には、列ラベルの行を 1 行だけ含めなければなりません。CREATE NICKNAME ステートメントで名前付き範囲を指定します。名前を付ける範囲中には、列ラベルの行は 1 行だけ含めなければなりません。名前付き範囲内に 1 つの列ラベル行を含めない場合は、最初のデータ行が列ラベルとして扱われます。SUM などの数式を含むセルは、数式ではなく数式の結果を戻します。数式の結果を戻したいのでなければ、数式を含むセルをこの範囲に含めないでください。この例では、名前を付けるセルの範囲は A3:D11 です。

1 つのラベル行を含むワークシートの例

この例は、ワークシートの上部に 1 つの列ラベル行のみある例を示しています。このレイアウトには、追加のラベル行、空白行、数式を含むセルはありません。

	A	B	C	D
1	Compound Name	Weight	Molecular Count	Tested?
2	compound_A	1.23	367	tested
3	compound_G		210	
4	compound_F	0.000425536	174	tested
5	compound_Y	1.000256		tested
6	compound_Q		1024	
7	compound_B	33.5362		
8	compound_S	0.96723	67	tested
9	compound_O	1.2		tested
10				
11				

図 8. 1 行目に 1 つの列ラベル行を含むワークシート

Excel ラッパーを使用する場合

CREATE NICKNAME ステートメント中でセルの範囲を指定するには、RANGE オプションを使用しなければなりません。この範囲に 1 行目の列ラベルを含めることはできません。セルの範囲として A2:D9 を指定します。

ODBC ラッパーを使用する場合

名前付き範囲を作成せずにこのデータにアクセスできます。CREATE NICKNAME ステートメント中にワークシート名を指定します。ラッパーは最初の非ブランク行をラベルとして読み取り、その情報をニックネームのための列名として使用します。後続の行はデータとして読み取られます。

データのみを含むワークシートの例

この例は、データのみを含むワークシートを示しています。列ラベルの行、ブランク行、数式を含むセルはありません。

	A	B	C	D
1	compound_A	1.23	367	tested
2	compound_G		210	
3	compound_F	0.000425536	174	tested
4	compound_Y	1.000256		tested
5	compound_Q		1024	
6	compound_B	33.5362		
7	compound_S	0.96723	67	tested
8	compound_O	1.2		tested
9				
10				

図9. データのみを含むワークシート

Excel ラッパーを使用する場合

ワークブックの最初のワークシートにデータがある場合は、RANGE オプションを使用しなくてもこのラッパーはデータにアクセスできます。ワークブックのそれ以外のワークシートにデータがある場合は、CREATE NICKNAME ステートメント中に RANGE オプションを指定しなければなりません。

ODBC ラッパーを使用する場合

ODBC ラッパーを使用して Excel データにアクセスする場合は、このラッパーは Excel ODBC ドライバーのサポート内容による制限を受けます。Excel ODBC ドライバーには、特定のワークシートの形式が必要になります。このドライバーは、最初の非ブランク行に列ラベルが含まれていることを想定しています。最初の非ブランク行にデータが含まれている場合は、この行のデータは残りのデータの列ラベルとして扱われます。ワークシートに列ラベルの行が含まれていない場合、最初の行はデータではなくラベルとして使用されます。その結果、最初のデータ行は失われます。ワークシートに変更を加えると、この要件を満たすことができます。データの前に新しい行を挿入し、各データ列のラベルを追加すると、1つのラベル行を含む例のようになります。

フェデレーテッド・サーバーへの Excel データ・ソースの追加

フェデレーテッド・サーバーが Excel データ・ソースにアクセスするように構成するには、アクセスするデータ・ソースとオブジェクトの情報をフェデレーテッド・サーバーに渡す必要があります。

始める前に

- Excel ワークシート内のデータは、Excel ラッパーがアクセスできるように適切に構造化されている必要があります。
- フェデレーションが、フェデレーテッド・サーバーとして機能するサーバー上にインストールされている必要があります。

- データベースがフェデレーテッド・サーバー上に存在している必要があります。

このタスクについて

制約事項

- Excel ラッパーは、フェデレーテッド・サーバーでサポートされているバージョンの Microsoft Windows オペレーティング・システムでのみ使用可能です。
- Excel アプリケーションがフェデレーテッド・サーバー上にインストールされている必要があります。
- Excel ワークブックは、フェデレーテッド・サーバーと同じコンピューター上か、またはネットワーク・アクセス可能なドライブ上になければなりません。
- Excel ラッパーは、Excel ワークブック中の最初のワークシート中のデータしかアクセスできません。
- フェデレーテッド・データベースのコード・ページ・セットが Excel ファイルの文字セットと一致していなければなりません。そうでない場合、照会から予期しない結果が戻される可能性があります。
- パススルー・セッションは使用できません。

手順

1. 『Excel ラッパーの登録』。
2. 84 ページの『Excel データ・ソースのサーバー定義の登録』。
3. 85 ページの『Excel データ・ソースのニックネームの登録』。

Excel ラッパーの登録

Excel データ・ソースにアクセスするには、ラッパーを登録する必要があります。

このタスクについて

ラッパーは、データ・ソースとの通信やデータの検索のためにフェデレーテッド・サーバーによって使用されます。ラッパーは、ライブラリー・ファイルのセットとしてインプリメントされます。

手順

コマンド行から CREATE WRAPPER ステートメントを発行します。

例:

```
CREATE WRAPPER excel_wrapper  
LIBRARY 'db2lsxls.dll';
```

LIBRARY パラメーターを CREATE WRAPPER ステートメント内に指定する必要があります。

Excel ラッパーのライブラリー・ファイル:

Excel ラッパー・ライブラリー・ファイルはフェデレーテッド・サーバーのインストール時にフェデレーテッド・サーバーに追加されます。

フェデレーテッド・サーバーをインストールすると、デフォルトのディレクトリー・パスに 3 つのライブラリー・ファイルが追加されます。例えば、フェデレーテ

ッド・サーバーが Windows 上で稼働している場合、ディレクトリー・パスに追加されるラッパー・ライブラリー・ファイルは db2lsx1s.dll、db2lsx1sF.dll、および db2lsx1sU.dll です。デフォルトのラッパー・ライブラリー・ファイルは、db2lsx1s.dll です。他のラッパー・ライブラリー・ファイルは、特定のラッパー・オプションで使用されます。

Excel ラッパーの登録時には、CREATE WRAPPER ステートメントに LIBRARY パラメーターを含め、デフォルトのラッパー・ライブラリー・ファイルの名前を指定する必要があります。

以下の表に、デフォルトのディレクトリー・パスとデフォルトのラッパー・ライブラリー・ファイルの名前がリストされています。

表 20. Excel ラッパーのライブラリー・ロケーションおよびファイル名

オペレーティング・システム	ディレクトリー・パス	ラッパー・ライブラリー・ファイル名
Windows	%DB2PATH%\%bin	db2lsx1s.dll

%DB2PATH% は、Windows 上でフェデレーテッド・サーバーがインストールされているディレクトリー・パスを指定するのに使用する環境変数です。デフォルトの Windows ディレクトリー・パスは C:\Program Files\IBM\SQLLIB です。

Excel データ・ソースのサーバー定義の登録

フェデレーテッド・オブジェクトの階層では、ニックネームにより識別される Excel ワークブック・ファイルが特定のサーバー定義オブジェクトに関連付けられている必要があるため、サーバー定義を登録する必要があります。

手順

コマンド行から CREATE SERVER ステートメントを発行します。

例:

```
CREATE SERVER server_definition_name
WRAPPER excel_wrapper;
```

CREATE SERVER ステートメント - Excel ラッパーの例:

Excel ラッパーにサーバー定義を登録するには、CREATE SERVER ステートメントを使用します。

以下の例は、生化学データを含むワークブックについてサーバー定義 *biochem_lab* を登録する方法を示します。発行する CREATE SERVER ステートメントは次のとおりです。

```
CREATE SERVER biochem_lab WRAPPER excel_wrapper;
```

biochem_lab

Excel サーバー定義に割り当てる名前。重複するサーバー定義名は使用できません。

```
WRAPPER Excel_wrapper
```

CREATE WRAPPER ステートメントで指定したラッパー名。

Excel データ・ソースのニックネームの登録

登録するそれぞれの Excel サーバ定義で、アクセスするそれぞれの Excel ワークシートごとにニックネームを登録する必要があります。Excel データ・ソースを照会するときには、ワークシートの名前ではなく、これらのニックネームを使用します。

このタスクについて

Excel ワークシートのニックネームを作成すると、ワークシート・データ内の情報がリレーショナル表にマップされます。

ワークシート内のブランク・セルは、NULL として解釈されます。

ワークシート内には最大で連続 10 個のブランクを含めることができ、データ・セット内に含まれます。10 個を超える連続ブランクは、データ・セットの終了として解釈されます。

ワークシートに、ブランク列を含めることができます。しかし、これらの列は、たとえそれが使用されない場合でも、有効なフィールドとして登録され、記述されなければなりません。

手順

コマンド行から CREATE NICKNAME ステートメントを発行します。

例:

```
CREATE NICKNAME nickname
(
  column_name data_type
    OPTIONS (nickname_column_options),
  column_name data_type
    OPTIONS (nickname_column_options),
  column_name data_type
    OPTIONS (nickname_column_options)
)
FOR SERVER server_definition_name
OPTIONS (nickname_options);
```

ニックネームの長さは 128 文字までです。

次のタスク

ニックネームを作成する Excel ワークシートごとに、このステップを繰り返します。

CREATE NICKNAME ステートメント - Excel ラッパーの例:

CREATE NICKNAME ステートメントを使用して、アクセスする Excel ワークシートのニックネームを登録します。以下の例は、必須パラメータと任意指定のニックネーム・オプションを示しています。

```
CREATE NICKNAME Compounds
(
  Compound_ID INTEGER,
  CompoundName VARCHAR(50),
  MolWeight FLOAT
```

```
)  
FOR SERVER biochem_lab  
OPTIONS (FILE_PATH 'C:\My Documents\CompoundMaster.xls',  
        RANGE 'B2:D25');
```

Compounds

Excel ワークシートを識別するために使用する固有のニックネームです。

重要: ニックネームは、2 つの部分 (スキーマとニックネーム) からなる名前です。ニックネームの登録時にスキーマを省略すると、ニックネームを登録するユーザーの許可 ID がニックネームのスキーマとして使用されます。

Compound_ID *INTEGER*

化合物の ID を含むワークシート列の名前とデータ・タイプ。

CompoundNAME *VARCHAR(50)*

化合物の名前を含むワークシート列の名前とデータ・タイプ。

MolWeight *FLOAT*

化合物の分子量を含むワークシート列の名前とデータ・タイプ。

FOR SERVER *biochem_lab*

CREATE SERVER ステートメントで Excel サーバー定義に割り当てた名前。

FILE_PATH 'C:\My Documents\CompoundMaster.xls')

アクセスするデータを含む Excel ワークブックの完全修飾ディレクトリ・パスとファイル名を指定します。このデータはワークブックの最初のワークシートに入っていないければなりません。

OPTIONS (RANGE 'B2:D25')

FILE_PATH ニックネーム・オプションで指定したワークブック内でアクセスするセルの範囲を指定します。

範囲オプション値の構文または意味エラーはどれも SQL1882E メッセージとなります。次のようなエラーがあります。

- 範囲が有効な範囲ではない。例えば、範囲で指定した左上のセルが、右下のセルの下側または右側にある場合です。
- 範囲値によって指定された列の数が CREATE NICKNAME ステートメントで指定された列の数に対応しない。
- 無効文字または他の構文エラーが見つかった。

例えば、以下に示す文字は無効です。

- #DIV/0!
- #N/A
- #NAME?
- #NULL!
- #NUM!
- #REF!
- #VALUE!

Excel データ・ソース - 照会の例

Excel データにアクセスするには、通常の名と表の列を使用する場合と同様に、SQL ステートメントでニックネームと定義済みのニックネーム列を使用します。

以下の例では、ニックネーム `compounds` を使用して Excel データにアクセスする照会を構成する方法を示します。

特定の情報列の選択

以下の照会は、分子量が 200 より大きいすべての `compound_ID` を表示します。

```
SELECT compound_ID FROM compounds
WHERE molweight > 200;
```

SELECT ステートメントでの OR 条件の使用

以下の照会は、化合物名または分子量が NULL になっているすべてのレコードを表示します。

```
SELECT * FROM compounds
WHERE compoundname IS NULL OR molweight IS NULL;
```

SELECT ステートメントでの LIKE および AND 条件の使用

以下の照会は、化合物名に文字列 `ase` が含まれており、分子量が 300 以上のすべてのレコードを表示します。

```
SELECT * FROM compounds
WHERE compoundname LIKE '%ase%' AND molweight >= 300;
```

Excel データ・ソース - サンプル・シナリオ

このシナリオでは、Excel ワークシートへのアクセスに使用するフェデレーテッド・オブジェクトを登録するのに必要な SQL ステートメントを示します。このシナリオには、作成したニックネームを使用して実行できるいくつかの照会が含まれています。

Excel ワークシートについての情報

このシナリオでは、最初にさまざまな化合物についての情報を含むワークシートを使用します。ワークシートが入っているワークブックの名前は `Compound_Master.xls` であり、このワークブックは Excel で作成されています。ワークブックの完全修飾パス名は、`C:\%Data%\Compound_Master.xls` です。

ワークブック内の最初のワークシートには 4 列× 9 行のデータが含まれています。列には、化合物の名前、化合物の重量、化合物の分子数、およびその化合物がテスト済みかどうかが表示されています。

ワークシートの内容は以下の表に示されています。

表 21. サンプル・ワークシート *Compound_Master.xls*

Å	A	B	C	D
1	COMPOUND_NAME	WEIGHT	MOL_COUNT	WAS_TESTED
2	compound_A	1.23	367	tested
3	compound_G	Å	210	Å

表 21. サンプル・ワークシート *Compound_Master.xls* (続き)

Å	A	B	C	D
4	compound_F	0.000425536	174	tested
5	compound_Y	1.00256	Å	tested
6	compound_Q	Å	1024	Å
7	compound_B	33.5362	Å	Å
8	compound_S	0.96723	67	tested
9	compound_O	1.2	Å	tested

フェデレーテッド・オブジェクトを登録する

Excel ラッパーを使用してワークシートにアクセスするには、オブジェクトをフェデレーテッド・サーバーに登録する必要があります。

1. Excel ラッパーを登録します。

例:

```
CREATE WRAPPER Excel LIBRARY 'db2lsxls.dll';
```

2. サーバ一定義を登録します。

例:

```
CREATE SERVER biochem_lab WRAPPER Excel;
```

3. Excel ワークシートを参照するニックネームを登録します。

たとえば、次のようにします。

```
CREATE NICKNAME Compound_Master
  (compound_name VARCHAR(40),
   weight          FLOAT,
   mol_count       INTEGER,
   was_tested      VARCHAR(20))
FOR SERVER biochem_lab
OPTIONS (FILE_PATH 'C:\Data\Compound_Master.xls');
```

登録プロセスが完了します。これで、Excel ワークシートはフェデレーテッド・システムの一部となり、SQL 照会で使用できるようになります。

以下の例は、SQL 照会と、*Compound_Master* ニックネームから戻される結果を示しています。

特定の WHERE 文節条件に一致するすべてのデータを戻す照会

分子数が 100 より大きい化合物のすべてのデータを戻すには、以下の照会を発行します。

```
SELECT * FROM Compound_Master
WHERE mol_count > 100;
```

ワークシートから特定の列を戻す照会

分子数がまだ判別されていないすべての化合物の名前と分子数を戻すには、以下の照会を発行します。

```
SELECT compound_name, mol_count FROM Compound_Master
WHERE mol_count IS NULL;
```

行 5、7、および 9 から、*compound_name* および *mol_count* 列が戻されます。

特定の WHERE 文節条件に一致する行の数を数える照会

重量が 1 より大きく、テスト済みでない化合物の数を戻すには、以下の照会を発行します。

```
SELECT count(*) FROM Compound_Master
WHERE was_tested IS NULL AND weight > 1
```

レコード・カウント 1 が戻されます。行 7 の化合物が照会基準に一致しています。

ワークシートから特定の列を戻す、副選択ステートメントを含む照会

分子数が判別済みで、その分子数が平均の分子数よりも小さいすべての化合物の名前と分子数を戻すには、以下の照会を発行します。

```
SELECT compound_name, mol_count FROM Compound_Master
WHERE mol_count IS NOT NULL
AND mol_count <
(SELECT AVG(mol_count) FROM Compound_Master
WHERE mol_count IS NOT NULL AND was_tested IS NOT NULL);
```

副照会は分子数の平均として 368 を戻します。主照会はこの平均を使用して、以下の表で示されている照会結果を戻します。

表 22. 照会の結果

COMPOUND_NAME	MOL_COUNT
compound_A	367
compound_G	210
compound_F	174
compound_S	67

Excel ラッパーのファイル・アクセス・コントロール・モデル

Excel ファイルにアクセスする際、ラッパーはセキュリティ上の目的でユーザー ID が必要です。Excel ラッパーは、フェデレーテッド・データベース・サービスに関連したユーザー ID を使用します。フェデレーテッド・データベース・サービスの名前は、データベース・インスタンスの名前によって異なります。例えば、データベース・インスタンスの名前が DB2 である場合、サービス名は DB2 - DB2 となります。フェデレーテッド・データベース・サービスに関連したユーザー ID を判別するには、Windows で「コントロール パネル」を使用してサービスを表示してください。サービス名をダブルクリックして、「ログオン」プロパティ・ページを表示します。

Informixデータ・ソースへのアクセスの構成

Informixデータ・ソースにアクセスするようにフェデレーテッド・サーバーを構成するには、アクセスするデータ・ソースとオブジェクトに関する情報をフェデレーテッド・サーバーに提供する必要があります。

始める前に

- InformixClient SDK ソフトウェアは、フェデレーテッド・サーバーとして機能するサーバー上にインストールされ、構成されている必要があります。
- フェデレーテッド・サーバーのセットアップを確認します。
- フェデレーテッド・パラメーターをチェックして、フェデレーションが有効になっていることを確認します。
- AIX フェデレーテッド・サーバーでは、AIXBase Application Development Math Library がインストールされている必要があります。このライブラリーがインストールされているかどうかは、AIX コマンド `lsipp -l bos.add.libm` で確認できます。

このタスクについて

制約事項

SuSE Linux Enterprise Server バージョン 9.2 以降のトラステッド・モードでは、Informix ラッパーはサポートされていません。

手順

1. Informix クライアント構成ファイルをセットアップしてテストします。
2. Informix 環境変数を設定します。
3. ラッパーを登録します。
4. サーバー定義を登録します。
5. ユーザー・マッピングを作成します。
6. サーバーへの接続をテストします。
7. Informix の表、ビュー、およびシノニムのニックネームを登録します。

Informix クライアント構成ファイルのセットアップおよびテスト

Informixクライアント構成ファイルは、フェデレーテッド・サーバーにインストールされているクライアント・ライブラリーを使用して、Informixデータベースに接続するために使用されます。

始める前に

Informix Client SDK ソフトウェアは、フェデレーテッド・サーバー上にインストールされている必要があります。

このタスクについて

このクライアント構成ファイルは、各 Informix データベース・サーバーのロケーション、およびそのデータベース・サーバーへの接続のタイプ (プロトコル) を指定します。

クライアント構成ファイルのデフォルトのロケーションは、フェデレーテッド・サーバーによって使用されるオペレーティング・システムによって異なります。

- UNIXが稼働するフェデレーテッド・サーバーの場合、構成ファイルのデフォルトの場所と名前は `$INFORMIXDIR/etc/sqlhosts` です。sqlhosts ファイルは、Informix Client SDK でインストールされます。
- Windows が稼働するフェデレーテッド・サーバーの場合、sqlhosts レジストリーのデフォルトのロケーションはローカル・コンピューターです。

sqlhosts のフォーマットは、「*Administrator's Guide for Informix Dynamic Server*」に記載されています。

手順

1. Informix Client SDK を構成します。
 - UNIX が稼働するフェデレーテッド・サーバーの場合は、sqlhosts ファイルを編集することによって Informix Client SDK を構成できます。sqlhosts ファイルは、Informix Connect または Informix Client SDK がインストールされている別のシステムからコピーすることもできます。
 - Windowsが稼働するフェデレーテッド・サーバーの場合は、Informix Setnet32 ユーティリティーを使用することによってInformix Client SDK を構成できます。Setnet32 ユーティリティーは、sqlhosts レジストリーをセットアップします。
2. sqlhosts ファイルまたはレジストリーの位置を検査します。
 - UNIX稼働するフェデレーテッド・サーバーの場合は、sqlhosts ファイルは `$INFORMIXDIR/etc/` ディレクトリーにあります。
 - Windowsが稼働するフェデレーテッド・サーバーの場合は、sqlhosts 情報は Windows レジストリーの次のキーに保持されます。
`HKEY_LOCAL_MACHINE\SOFTWARE\INFORMIX\SQLHOSTS`
3. sqlhosts ファイルまたはレジストリーをデフォルトの検索パス以外のパスに置く場合は、`INFORMIXSQLHOSTS` 環境変数を設定してファイルの場所を指定してください。以下のいずれかのオプションを使用して、`INFORMIXSQLHOSTS` 環境変数を設定します。
 - UNIXが稼働するフェデレーテッド・サーバーの場合は、`INFORMIXSQLHOSTS` 環境変数を sqlhosts ファイルの完全修飾名に設定します。
 - Windows が稼働するフェデレーテッド・サーバーの場合は、Setnet32 ユーティリティーを使用して、`INFORMIXSQLHOSTS` 環境変数をレジストリーを保管する Windowsコンピューターの名前に設定します。
4. 接続をテストして、クライアント・ソフトウェアが Informix サーバーに接続できることを確認してください。Informixdbaccess ユーティリティーがフェデレーテッド・サーバーにある場合は、そのツールを使用して接続をテストしてください。ない場合は、Informix デモ・プログラムを実行して、クライアントのセットアップをテストしてください。

Informix 環境変数の設定

Informix環境変数は、フェデレーテッド・サーバーの `db2dj.ini` ファイルに設定する必要があります。

このタスクについて

Informixデータ・ソース用の環境変数には、必須の変数とオプションの変数があります。Informixラッパーをインストールする前に Informixクライアント・ソフトウェアをインストールしてある場合には、必須の Informix環境変数は db2dj.ini ファイルに設定されています。

Informixラッパーをインストールする前に Informixクライアント・ソフトウェアをインストールしていない場合や、オプションの環境変数を設定する場合は、このタスクのステップにしたがって、環境変数を設定する必要があります。

手順

1. 以下のいずれかの方法で、使用する Informix 環境変数を設定します。

方法	ステップ
自動的に環境変数を設定するには、次のようにします。	DB2セットアップ・プログラムを再び実行して、「カスタム」インストール・オプションを指定します。ウィザードの指示に従ってください。 重要: インストール・プログラムを再び実行した場合は、必須の環境変数のみが設定されます。オプションの環境変数は手動で設定する必要があります。
ラッパー構成ツールを使用して環境変数を設定するには、次のようにします。	ラッパー構成ツールを実行して、必須の環境変数とオプションの環境変数を設定します。1 ページの『ラッパー構成ツールの実行』の説明に従ってください。
手動で環境変数を設定するには、次のようにします。	db2dj.ini ファイルを編集します。 <ul style="list-style-type: none">• UNIX が稼働するフェデレーテッド・サーバーでは、db2dj.ini ファイルは sql1lib/cfg ディレクトリにあります。• Windows が稼働するフェデレーテッド・サーバーでは、db2dj.ini ファイルは %DB2PATH%\cfg ディレクトリにあります。

db2dj.ini ファイルには、ご使用のフェデレーテッド・サーバーにインストールされている Informix クライアント・ソフトウェアに関する構成情報が含まれます。このファイルが存在しない場合は、任意のテキスト・エディターを使用して、db2dj.ini という名前のファイルを作成することができます。db2dj.ini ファイルには、環境変数の完全修飾パスを指定する必要があります。そうしないと、エラーが発生します。次の環境変数は、UNIX 上の db2dj.ini ファイル内の項目の一例です。

```
INFORMIXDIR=/informix/csdk  
INFORMIXSERVER=inf10
```

2. Informix コード・ページ変換環境変数を (必要に応じて) 設定します。
3. フェデレーテッド・サーバーで環境変数を確実に設定するために、フェデレーテッド・データベース・インスタンスをリサイクルします。

フェデレーテッド・データベース・インスタンスをリサイクルするには、次のコマンドを発行します。

```
db2stop
db2start
```

Informix 環境変数

Informixデータ・ソース用の環境変数には、必須の変数とオプションの変数があります。これらの変数は `db2dj.ini` ファイルで設定されます。

Informix の有効な環境変数は以下のとおりです。

- INFORMIXDIR
- INFORMIXSERVER
- INFORMIXSQLHOSTS (オプション)
- CLIENT_LOCALE (オプション)
- DB_LOCALE (オプション)
- DBNLS (オプション)

CLIENT_LOCALE、DB_LOCALE、および DBNLS 環境変数はコード・ページ環境変数です。

変数の説明

INFORMIXDIR

InformixClient SDK ソフトウェアがインストールされているディレクトリー・パスを指定します。

例:

- UNIX が稼働するフェデレーテッド・サーバーの場合、パスを次のように設定します。

```
INFORMIXDIR=/informix/csdk
```

- Windowsが稼働するフェデレーテッド・サーバーの場合、パスを次のように設定します。

```
INFORMIXDIR=C:\informix\csdk
```

INFORMIXSERVER

デフォルトの Informix サーバーの名前を識別します。この設定は、`sqlhosts` ファイル (UNIX) または `SQLHOSTS` レジストリー・キー (Windows) で有効な項目でなければなりません。INFORMIXSERVER の値を入手するには、`sqlhosts` ファイルを読み取ります。`dbservername` 値のいずれかを選択します。`dbservername` は、`sqlhosts` ファイル内の各項目の最初の値です。

例:

```
INFORMIXSERVER=inf10
```

要件: Informix ラッパーがこの環境変数の値を使用しない場合でも、Informix クライアントによってこの環境変数を設定するよう要求されます。ラッパーは、アクセスしたい Informix データベース・サーバーを指定する NODE サーバー・オプションの値を使用します。

INFORMIXSQLHOSTS

Informix `sqlhosts` ファイルのデフォルトのパスを使用している場合、この環境変数を設定する必要はありません。ただし、Informix `sqlhosts` ファイルの他のパスを使用している場合、この環境変数を設定する必要があります。INFORMIXSQLHOSTS 変数を Informix `sqlhosts` ファイルがある絶対パスの名前に設定する必要があります。

- UNIXが稼働するフェデレーテッド・サーバーの場合、デフォルトのパスは `$INFORMIXDIR/etc/sqlhosts` です。
- Windowsが稼働するフェデレーテッド・サーバーの場合、SQLHOSTS レジストリー・キーがローカル・コンピューター上にない場合には、レジストリーを保管する Windows コンピューターの名前が INFORMIXSQLHOSTS 環境変数の値になります。

この環境変数を別のパスに設定する UNIX の例は、次のとおりです。

```
INFORMIXSQLHOSTS=/informix/csdk/etc/my_sqlhosts
```

Informix コード・ページ変換:

Informix ラッパーが Informix データ・ソースに接続するたび、ラッパーは、その接続に使用するコード・ページ値を判別します。Informix ラッパーが設定したコード・ページ値を使用することもできますし、CLIENT_LOCALE 環境変数を設定してコード・ページを指定することもできます。

Informix コード・ページ変換を指定する環境変数は、フェデレーテッド・サーバーの `db2dj.ini` ファイルに設定されます。

Informix コード・ページ変換の場合、以下のオプション環境変数を設定することができます。

- CLIENT_LOCALE
- DB_LOCALE
- DBNLS

Informix コード・ページ環境変数は以下のとおりです。

CLIENT_LOCALE

使用する Informix ロケールを指定します。Informix ラッパーが変数設定を自動的に決定することのないようにするには、この変数を使用します。

例:

```
CLIENT_LOCALE=Informix_client_locale_value
```

- CLIENT_LOCALE 変数がフェデレーテッド・サーバー上の `db2dj.ini` ファイルに設定されている場合、ラッパーは `db2dj.ini` ファイルのコード・ページ値を使用します。
- フェデレーテッド・サーバー上で CLIENT_LOCALE 変数が設定されていない場合、ラッパーはフェデレーテッド・データベースのテリトリーとコード・ページを判断します。ラッパーは、CLIENT_LOCALE 変数を最もよく一致する Informix ロケールに設定します。一致する Informix ロケールがない場合、ラッパーは、CLIENT_LOCALE 変数を UNIX システムの場合には `en_us.8859-1` ロケールに、Windows システムの場合には `en_us.CP1252` ロケールに設定します。

Informix サーバー上で **glfiles** コマンドを発行することによって、有効な Informix ロケールのリストを表示できます。

コード・ページ変換についての詳細は、『*Informix Guide to GLS Functionality*』を参照してください。

DB_LOCALE

Informix データベースが、クライアント・ロケールとは異なるコード・ページを使用することを指定します。Informix に 2 つのコード・ページ間で変換を実行させるには、この変数を使用します。DB_LOCALE 環境変数を Informix データベース・ロケールの名前に設定します。

例:

```
DB_LOCALE=Informix_db_locale_value
```

DBNLS

DB_LOCALE 設定が Informix データベースの実際のロケールと一致することを Informix が検証するように指定します。この環境変数を 1 に設定します。

例:

```
DBNLS=1
```

Informixによるコード・ページ変換の強制実行

Informix データベースが、クライアント・ロケールとは異なるコード・ページを使用しており、Informix が 2 つのコード・ページ間で変換を実行するようにしたいとします。その場合、以下のことを行う必要があります。

1. Informix 環境変数 DB_LOCALE を Informix データベース・ロケールの名前に設定します。フェデレーテッド・サーバー上の db2dj.ini ファイル内で、この変数を設定します。
2. DB_LOCALE 設定が Informix データベースの実際のロケールと一致することを検証するには、Informix 環境変数 DBNLS を 1 に設定する必要があります。この変数は、フェデレーテッド・サーバー上の db2dj.ini ファイル内で設定します。

中国語コード・ページ GB 18030 を使用する Informix データ

中国語コード・ページ GB 18030 を使用するデータにアクセスするには、フェデレーテッド・データベースで UTF-8 コード・ページを使用し、次の設定を db2dj.ini ファイルに追加することによって、Informix で GB 18030 データが正しく Unicode に変換されるようにします。

```
DB_LOCALE=zh_cn.GB18030-2000
```

Informix ラッパーの登録

Informix データ・ソースにアクセスするためには、ラッパーを登録する必要があります。ラッパーは、データ・ソースとの通信やデータの取得のためにフェデレーテッド・サーバーによって使用されます。ラッパーは、ライブラリー・ファイルのセットとしてインプリメントされます。

手順

CREATE WRAPPER ステートメントを発行し、Informix ラッパーのデフォルト名を指定します。

例:

```
CREATE WRAPPER INFORMIX
```

推奨: デフォルトのラッパー名を使用します。Informix用のラッパーのデフォルト名は **INFORMIX** です。このデフォルト名を使用してラッパーを登録する場合、フェデレーテッド・サーバーは稼働しているオペレーティング・システムに適した Informix ラッパー・ライブラリーを自動的に使用します。

デフォルト・ラッパー名がフェデレーテッド・データベース中の既存のラッパー名と競合する場合には、任意の名前を選択してデフォルトのラッパー名と置き換えることができます。デフォルト名以外の名前を使用する場合、CREATE WRAPPER ステートメントに **LIBRARY** パラメーターを含める必要があります。

例えば、AIX オペレーティング・システムを使用するフェデレーテッド・サーバーで、`informix_wrapper` という名前のラッパーを登録するには、以下のステートメントを発行します。

```
CREATE WRAPPER informix_wrapper LIBRARY 'libdb2informix.a';
```

指定するラッパー・ライブラリー・ファイルの名前は、フェデレーテッド・サーバーのオペレーティング・システムによって異なります。CREATE WRAPPER ラッパー・ステートメントで指定する正しいライブラリー名については、『Informix ラッパー・ライブラリー・ファイル』のリストを参照してください。

Informix ラッパー・ライブラリー・ファイル

Informix ラッパー・ライブラリー・ファイルは、フェデレーションのインストール時にフェデレーテッド・サーバーに追加されます。

フェデレーションをインストールすると、デフォルトのディレクトリー・パスに 3 つのライブラリー・ファイルが追加されます。例えば、フェデレーテッド・サーバーが AIX 上で実行されている場合、ディレクトリー・パスには、ラッパー・ライブラリー・ファイル `libdb2informix.a`、`libdb2informixF.a`、および `libdb2informixU.a` が追加されます。デフォルトのラッパー・ライブラリー・ファイルは `libdb2informix.a` です。それ以外のラッパー・ライブラリー・ファイルは、デフォルトのラッパー・ライブラリーによって内部的に使用されます。

ラッパーの登録時にデフォルトのラッパー名を使用しないことにした場合、CREATE WRAPPER ステートメントに **LIBRARY** パラメーターを含めて、デフォルトのラッパー・ライブラリー・ファイルの名前を指定する必要があります。

以下の表に、デフォルトのディレクトリー・パスとデフォルトのラッパー・ライブラリー・ファイルの名前がリストされています。

表 23. Informix ラッパー・ライブラリー・ロケーションおよびファイル名

オペレーティング・システム	ディレクトリー・パス	ライブラリー・ファイル名
AIX	<code>/usr/opt/install_path/lib32/</code> <code>/usr/opt/install_path/lib64/</code>	<code>libdb2informix.a</code>

表 23. Informix ラッパー・ライブラリー・ロケーションおよびファイル名 (続き)

オペレーティング・システム	ディレクトリー・パス	ライブラリー・ファイル名
HP-UX	 /opt/IBM/db2/install_path/lib32 /opt/IBM/db2/install_path/lib64	libdb2informix.so
Linux	 /opt/IBM/db2/install_path/lib32 /opt/IBM/db2/install_path/lib64	libdb2informix.so
Solaris	 /opt/IBM/db2/install_path/lib32 /opt/IBM/db2/install_path/lib64	libdb2informix.so
Windows	%DB2PATH%\bin	db2informix.dll

- `install_path` は、UNIX または Linux 上でフェデレーションがインストールされているディレクトリーのパスです。
- `%DB2PATH%` は、Windows でフェデレーションがインストールされているディレクトリーのパスを指定するのに使用する環境変数です。デフォルトの Windows ディレクトリー・パスは、`C:\Program Files\IBM\SQLLIB` です。

Informix データ・ソースのサーバー定義の登録

フェデレーテッド・データベース内に、アクセスする各 Informixサーバーを登録する必要があります。

手順

1. Informix `sqlhosts` ファイルまたはレジストリーでノード名を見つけてます。

`sqlhosts` ファイルの例:

```
inf10an onsoctcp anaconda inmx10
inf10bo onsoctcp boa ifmx10
inf10py onsoctcp python ifmx10
```

- 各行の最初の値は `inf10an` などの `node_name` です。
- 各行の 2 番目の値は `nettype`、または接続のタイプです。この例の場合、`onsoctcp` はこれが TCP/IP 接続であることを示します。
- 各行の 3 番目の値は `anaconda`、`boa`、`python` などのホスト名です。
- 各行の 4 番目の値は `inmx10` などのサービス名です。サービス名フィールドは、2 番目の値にリストされた `nettype` により異なります。

`sqlhosts`ファイルのフォーマットおよび、それらのフィールドの意味の詳細は、Informix 資料「*Administrators Guide for Informix Dynamic Server*」を参照してください。

2. コマンド行から `CREATE SERVER` ステートメントを発行します。

例:

```
CREATE SERVER server_definition_name TYPE informix
VERSION version_number WRAPPER INFORMIX
OPTIONS (NODE 'node_name', DBNAME 'database_name');
```

'node_name' および 'database_name' 変数の指定は、CREATE SERVER ステートメントではオプションですが、Informix のデータ・ソースにアクセスする場合には必須です。

サーバー定義を登録した後、サーバー・オプションを追加またはドロップするには、ALTER SERVER ステートメントを使用します。

CREATE SERVER ステートメント - Informixラッパーの例

Informixラッパーのサーバー定義を登録するには、CREATE SERVER ステートメントを使用します。必須パラメーターを使用する完全なステートメントの例、およびその他のサーバー・オプションを使用する例が、このトピックには含まれていません。

完全な例

次の例は、CREATE SERVER ステートメントを発行して、Informix ラッパーのサーバー定義を登録する方法を示しています。

```
CREATE SERVER asia TYPE informix VERSION 10 WRAPPER INFORMIX
OPTIONS (NODE 'abc', DBNAME 'sales');
```

asia Informix データベース・サーバーに割り当てる名前。重複するサーバー定義名は使用できません。

TYPE informix

アクセスを構成するデータ・ソース・サーバーのタイプを指定します。Informix ラッパーの場合、サーバー・タイプは informix でなければなりません。

VERSION 10

これは、アクセスする Informix データベース・サーバーのバージョンです。

WRAPPER INFORMIX

CREATE WRAPPER ステートメントで指定したラッパー名。

NODE 'abc'

Informix データベース・サーバーが存在するノードの名前。ノード名は sqlhosts ファイルから入手します。この値は大文字と小文字を区別します。

ノード名は CREATE SERVER ステートメントではオプションとなっていますが、Informix データ・ソースの場合は必要です。

DBNAME 'sales'

アクセスする Informixデータベースの名前。この値は大/小文字の区別があります。

データベースの名前の指定は、CREATE SERVER ステートメントではオプションですが、Informix データ・ソースの場合は必須です。

追加のサーバー・オプション

サーバー定義の作成時に、CREATE SERVER ステートメントに追加のサーバー・オプションを指定することができます。サーバー・オプションには、汎用のサーバー・オプションと Informix に特有のサーバー・オプションを指定できます。

FOLD_ID および FOLD_PW サーバー・オプション

データ・ソースに接続する時に、フェデレーテッド・サーバーは、ユーザー ID とパスワードについて、入力された文字だけでなく、大文字と小文字のすべての可能な組み合わせを使用して接続を試みます。フェデレーテッド・サーバーはデータ・ソース・サーバーとの接続に成功するまで、最高 9 回、接続を試みます。このような試みは接続時間を長引かせることになり、ユーザー ID がロックアウトされる結果になる可能性もあります。FOLD_ID および FOLD_PW サーバー・オプションに値を指定することで、ロックアウトを防ぐことができます。FOLD_ID と FOLD_PW のサーバー・オプションを 'N' (ユーザー ID またはパスワードを大文字変換しない) に設定することができます。

FOLD_ID と FOLD_PW のサーバー・オプションを 'N' に設定する場合は、ユーザー ID とパスワードを大/小文字を間違えずに指定する必要があります。これらのサーバー・オプションを 'N' にすると、指定したユーザー ID やパスワードが間違っている場合に、ラッパーがさまざまな大文字と小文字の組み合わせを試行しないで済むという利点があります。これらの 2 つのサーバー・オプションにより、ログインに何回も失敗して限度を超え、ID がロックアウトされる可能性を減らすことができます。

以下に、サーバー・オプションを追加した Informix サーバー定義の例を示します。

```
CREATE SERVER asia TYPE informix VERSION 10 WRAPPER INFORMIX
  OPTIONS (NODE 'abc', DBNAME 'sales', FOLD_ID 'N', FOLD_PW 'N');
```

IUD_APP_SVPT_ENFORCE サーバー・オプションの例

IUD_APP_SVPT_ENFORCE オプションは、フェデレーテッド・サーバーがアプリケーション・セーブポイント・ステートメントの検出または作成を実施すべきかどうかを指定します。Informix はアプリケーション・セーブポイント・ステートメントをサポートしません。'N' にすると、エラーが検出されても、フェデレーテッド・サーバーはトランザクションをロールバックしません。エラー・リカバリーは、アプリケーションが処理する必要があります。

IUD_APP_SVPT_ENFORCE サーバー・オプションは、Informix データ・ソースとの間で複製が行えるよう、'N' に設定する必要があります。以下に、IUD_APP_SVPT_ENFORCE サーバー・オプションを追加した Informix サーバー定義の例を示します。

```
CREATE SERVER asia TYPE informix VERSION 10 WRAPPER INFORMIX
  OPTIONS (NODE 'abc', DBNAME 'sales', IUD_APP_SVPT_ENFORCE 'N');
```

INFORMIX_DB_LOCALE および INFORMIX_CLIENT_LOCALE オプション

INFORMIX_DB_LOCALE オプションは、フェデレーテッド・サーバーとデータ・ソース・サーバーの接続に使用するデータベース・ロケール環境変数 (DB_LOCALE) を設定します。INFORMIX_DB_LOCALE オプションを指定しない場合は、InformixDB_LOCALE 環境変数が db2dj.ini ファイルに指定されている値に設定されます。db2dj.ini ファイルに DB_LOCALE 環境変数が指定されていない場合には、InformixDB_LOCALE 環境変数は設定されません。有効な値は、任意の有効な Informix ロケールです。このオプションは任意です。デフォルト設定は「なし」です。

INFORMIX_CLIENT_LOCALE オプションは、フェデレーテッド・サーバーとデータ・ソース・サーバーの接続に使用するクライアント・ロケール環境変数 (CLIENT_LOCALE) を設定します。INFORMIX_CLIENT_LOCALE オプションを指定しない場合は、InformixCLIENT_LOCALE 環境変数が db2dj.ini ファイルに指定されている値に設定されます。db2dj.ini に CLIENT_LOCALE が指定されていない場合は、InformixCLIENT_LOCALE 環境変数がフェデレーテッド・データベースのコード・ページとテリトリに最も近い Informixロケールに設定されます。有効な値は任意の有効な Informixロケールです。このオプションは任意です。デフォルト設定は「なし」です。

以下に、INFORMIX_DB_LOCALE および INFORMIX_CLIENT_LOCALE オプションを追加した Informix サーバー定義の例を示します。

```
CREATE SERVER asia TYPE informix VERSION 10 WRAPPER INFORMIX
  OPTIONS (NODE 'abc', DBNAME 'sales', INFORMIX_DB_LOCALE 'en_us.8859-1',
  INFORMIX_CLIENT_LOCALE 'en_us.CP1252');
```

Informix データ・ソースのユーザー・マッピングの作成

Informix サーバーにアクセスする場合、フェデレーテッド・サーバーは、データ・ソースに対して有効なユーザー ID とパスワードを使用して、Informix サーバーへの接続を確立します。それぞれのフェデレーテッド・サーバーのユーザー ID とパスワード、および対応するデータ・ソースのユーザー ID とパスワードの間の関連付け (ユーザー・マッピング) を定義しなければなりません。

このタスクについて

フェデレーテッド・システムにアクセスする、各ユーザー ID のユーザー・マッピングを作成して、Informix データ・ソースへの分散要求を送信します。

手順

CREATE USER MAPPING ステートメントを発行します。

例えば、次のようにします。

```
CREATE USER MAPPING FOR local_userID SERVER server_definition_name
  OPTIONS (REMOTE_AUTHID 'remote_userID', REMOTE_PASSWORD 'remote_password');
```

REMOTE_AUTHID および REMOTE_PASSWORD 変数の指定は、CREATE USER MAPPING ステートメントではオプションですが、Informix データ・ソースにアクセスする場合には必須です。

CREATE USER MAPPING ステートメント - Informixラッパーの例

CREATE USER MAPPING ステートメントを使用して、フェデレーテッド・サーバーのユーザー ID を Informix のユーザー ID およびパスワードにマップしてください。このトピックには、必要パラメーターを使用する完全指定の例と、CREATE USER MAPPING ステートメントで DB2 特殊レジスター USER を使用する方法を示す例が含まれます。

完全な例

以下に示すのは、フェデレーテッド・サーバーのユーザー ID を Informix サーバーのユーザー ID およびパスワードにマップする方法を示す例です。

```
CREATE USER MAPPING FOR VINCENT SERVER asia
  OPTIONS (REMOTE_AUTHID 'vinnie', REMOTE_PASSWORD 'close2call');
```

VINCENT

Informix サーバーで定義されるユーザー ID にマップする、ローカルのユーザー ID を指定します。

SERVER asia

CREATE SERVER ステートメントで登録した Informixサーバーのサーバー定義名を指定します。

REMOTE_AUTHID 'vinnie'

VINCENT のマップ先となる Informix データベース・サーバー側のユーザー ID を指定します。CREATE SERVER ステートメントで FOLD_ID サーバー・オプションを「U」または「L」に設定している場合を除き、この値は大文字小文字を区別します。

リモート・ユーザー ID の指定は、CREATE USER MAPPING ステートメントではオプションとなっていますが、Informix データ・ソースの場合は必要です。

REMOTE_PASSWORD 'close2call'

'vinnie' に関連するパスワードを指定します。CREATE SERVER ステートメントで FOLD_PW サーバー・オプションを「U」または「L」に設定している場合を除き、この値は大文字小文字を区別します。

リモート・パスワードの指定は、CREATE USER MAPPING ステートメントではオプションとなっていますが、Informix データ・ソースの場合は必要です。

特殊レジスターの例

DB2 特殊レジスター USER を使用して、CREATE USER MAPPING ステートメントを実行するユーザーの許可 ID を、REMOTE_AUTHID ユーザー・オプションに指定されるデータ・ソースの許可 ID にマップすることができます。

以下に、特殊レジスター USER を含む CREATE USER MAPPING ステートメントの例を示します。

```
CREATE USER MAPPING FOR USER SERVER asia
  OPTIONS (REMOTE_AUTHID 'vinnie', REMOTE_PASSWORD 'close2call');
```

Informix サーバーへの接続のテスト

Informix データ・ソースへアクセスするようにフェデレーテッド・サーバーが適正に構成されているかどうかを判別するには、Informixデータ・ソース・サーバーへの接続をテストします。

このタスクについて

サーバー定義および定義したユーザー・マッピングを使用して、Informix サーバーへの接続をテストできます。

手順

パススルー・セッションを開き、Informix システム表に対して SELECT ステートメントを発行します。SELECT ステートメントがカウントを戻す場合、サーバー定義およびユーザー・マッピングは適切にセットアップされています。

例:

```
SET PASSTHRU server_definition_name
SELECT count(*) FROM informix.systables
SET PASSTHRU RESET
```

SELECT ステートメントがエラーを戻す場合には、接続エラーのトラブルシューティングを行う必要があります。

データ・ソース接続エラーのトラブルシューティング

データ・ソース・サーバーへのテスト接続では、いくつかの理由でエラーが戻されることがあります。以下の処置により、エラーが発生した理由を判別することができます。

症状

データ・ソースに接続しようとしたときにエラーが戻されます。

原因

接続の問題には、いくつかの原因が考えられます。

問題の解決方法

データ・ソース接続エラーのトラブルシューティングを行うには、以下の項目で問題がないかどうかをチェックします。

- データ・ソースが使用可能であることを確認します。
- 該当する場合は、データ・ソース・サーバーが着信接続に対して構成されていることを確認します。
- REMOTE_AUTHID と REMOTE_PASSWORD のオプションに関するユーザー・マッピング設定を調べ、データ・ソースとの接続に有効な設定になっているか確認します。必要に応じて、ユーザー・マッピングを変更するか、別のユーザー・マッピングを作成します。
- 該当する場合は、フェデレーテッド・サーバー上のデータ・ソース・クライアント・ソフトウェアがインストールされていて、データ・ソースに接続できるように正しく構成されていることを確認します。
- ODBC データ・ソースの場合、フェデレーテッド・サーバー上の ODBC ドライバーがインストールされていて、ODBC データ・ソース・サーバーに接続できるように正しく構成されていることを確認します。Windows が稼働するフェデレーテッド・サーバーでは、ドライバーをチェックするときに「ODBC データ ソース アドミニストレータ」ツールを使用します。UNIX が稼働するフェデレーテッド・サーバーの場合は、ODBC クライアントのベンダーが提供している資料を参照してください。

- フェデレーテッド・サーバー上で設定されている変数の設定がデータ・ソースについて適切であることを確認します。これらの変数には、システム環境変数、db2dj.ini ファイル、および DB2 Profile Registry (db2set) 変数が含まれます。
- サーバー定義を調べます。必要であれば、サーバー定義をドロップして再作成します。

Informixラッパーのパフォーマンス調整

FOLD_ID および FOLD_PW サーバー・オプションを使用することで、フェデレーテッド・サーバーと Informixデータ・ソースの接続を改善することができます。

データ・ソースに接続する時に、フェデレーテッド・サーバーは、ユーザー ID とパスワードについて、大文字と小文字のすべての可能な組み合わせを使用して接続を試みます。サーバーはデータ・ソース・サーバーとの接続に成功するまで、最高 9 回、接続を試みます。このような試みは接続時間を長引かせることになり、ユーザー ID がロックアウトされる結果になる可能性もあります。

FOLD_ID および FOLD_PW サーバー・オプションに値を指定することにより、パフォーマンスを改善することができます。

- Informix のすべてのユーザー ID とパスワードが小文字である場合、FOLD_ID および FOLD_PW サーバー・オプションを値 'L' を使用して設定することにより、接続時間が改善されます。

例:

```
ALTER SERVER TYPE INFORMIX
  OPTIONS (ADD FOLD_ID 'L');
ALTER SERVER TYPE INFORMIX
  OPTIONS (ADD FOLD_PW 'L');
```

- フェデレーテッド・サーバーは、ユーザー ID およびパスワードの大文字と小文字の値のそれぞれの組み合わせを試みます。これらのオプションを 'N' (ユーザー ID とパスワードを大文字に変換しない) にすることにより、何回もログインを試みて失敗し、失敗ログインの最大数を越えるという可能性を減らすことができます。このような設定をした場合は、常に大/小文字を正しく使用して、ユーザー ID とパスワードを指定する必要があります。誤ったユーザー ID とパスワードが指定されても、ラッパーは組み合わせを変えて試行することはありません。

例:

```
ALTER SERVER TYPE INFORMIX
  OPTIONS (ADD FOLD_ID 'N');
ALTER SERVER TYPE INFORMIX
  OPTIONS (ADD FOLD_PW 'N');
```

Informix 表、ビュー、およびシノニムのニックネームの登録

登録する Informixサーバー定義ごとに、アクセスする各表、ビュー、またはシノニムのニックネームを登録する必要があります。Informix サーバーを照会するときには、データ・ソース・オブジェクトの名前ではなく、これらのニックネームを使用します。

始める前に

ニックネームを登録する前に、Informix データ・ソース側で統計を更新します。フェデレーテッド・データベースは、データ・ソースのカatalog統計に基づいて、照会処理を最適化します。DB2 **RUNSTATS** コマンドに相当する Informix **UPDATE STATISTICS** コマンドを使用して、データ・ソースの統計を更新することができます。

手順

CREATE NICKNAME ステートメントを発行します。ニックネームの長さは 128 バイトまでです。

例:

```
CREATE NICKNAME nickname FOR server_definition_name."remote_schema"."remote.table" ;
```

ニックネームを作成すると、フェデレーテッド・サーバーはニックネームを使用してデータ・ソース・カatalogを照会します。この照会は、データベース・ソースの表、ビュー、またはシノニムへの接続をテストします。接続不可能な場合、エラー・メッセージが出ます。

次のタスク

このステップを、ニックネームを作成したいそれぞれの Informix 表、ビュー、またはシノニムごとに繰り返します。

CREATE NICKNAME ステートメント - Informixラッパーの例

アクセス先の Informix 表、ビュー、またはシノニムのニックネームを登録するには、**CREATE NICKNAME** ステートメントを使用します。必須パラメーターを使用する完全なステートメントの例が、このトピックには含まれています。

完全な例

```
CREATE NICKNAME JPSALES FOR asia."vinnie"."japan" ;
```

JPSALES

Informix の表、ビュー、またはシノニムを識別するために使用する固有なニックネーム。

重要: ニックネームは、2 つの部分 (スキーマとニックネーム) からなる名前です。ニックネームの登録時にスキーマを省略すると、そのニックネームのスキーマはニックネームを登録するユーザーの許可 ID になります。

asia."*vinnie*".*"japan"*

リモート・オブジェクトを表す、3 つの部分からなる ID。

- *asia* は、**CREATE SERVER** ステートメントで Informix データベース・サーバーに割り当てたサーバー定義名です。
- *vinnie*は、データベースが ANSI 準拠ではない場合に、表、ビュー、またはシノニムが属する所有者の名前です。ANSI 準拠のデータベースでは、スキーマ名です。
- *japan*は、アクセスするリモートの表、ビュー、またはシノニムの名前です。

フェデレーテッド・サーバーは、引用符で名前を囲まない限り、Informix スキーマおよび表の名前を大文字に変換します。

JDBC データ・ソースへのアクセスの構成

JDBC データ・ソースにアクセスするようにフェデレーテッド・サーバーを構成するには、アクセスするデータ・ソースとオブジェクトの情報をフェデレーテッド・サーバーに指定しなければなりません。

始める前に

JDBC API を通じてアクセスするデータ・ソースを、この資料では JDBC データ・ソースと言います。

- JDBC ドライバーが、フェデレーテッド・サーバーとして機能するコンピュータ上にインストールされ、構成されている必要があります。
- フェデレーテッド・サーバーのセットアップを検証します。
- フェデレーテッド・パラメーターをチェックして、フェデレーションが有効になっていることを確認します。
- ご使用のデータ・ソース・クライアントに応じて、以下の変数がシステム環境と db2dj.ini ファイルの両方で指定されていることを確認する必要があります。
 - CLASSPATH システム環境変数
 - データ・ソース・クライアントに必要な他のシステム環境変数
 - DB2 プロファイル・レジストリーの DB2LIBPATH 変数 (db2set)

必要な変数を判別するには、データ・ソース・クライアントに付属の資料を参照してください。

例えば、ランタイム・ライブラリーを指定する必要がある場合は、以下の環境変数を指定しなければならないことがあります。

- AIX オペレーティング・システムの LIBPATH
- Windows オペレーティング・システムの PATH
- Linux および Solaris オペレーティング・システムの LD_LIBRARY_PATH

注: システム環境変数を構成するか、db2dj.ini ファイルを変更する場合は、InfoSphere Federation Server を停止してから再始動する必要があります。

このタスクについて

制約事項

- JDBC ラッパーがサポートされるのは、fenced モードの場合だけです。
- JDBC ラッパーは以下の機能とステートメントをサポートしません。
 - ニックネームでの LOCK TABLE ステートメント
 - ステートメント・レベルの分離
- JDBC データ・ソースは位置指定の更新操作および削除操作をサポートしません。
- JDBC ラッパーは、接続当たりのアクティブなステートメントの数を制限するデータ・ソースに対する INSERT、UPDATE、DELETE ステートメントをサポート

しません。お使いのデータ・ソースが接続当たりのアクティブなステートメントの数を制限するかどうかを判断するには、そのデータ・ソースの資料を参照してください。

- JDBC ラッパーは、ドライバー固有の SQL データ・タイプ標識を使用するデータ・タイプの列がある表に対する操作をサポートしません。サポートされないタイプの操作には、パススルー・セッションでの CREATE NICKNAME ステートメントと SELECT ステートメントが含まれます。JDBC ラッパーがサポートするのは、JDBC 仕様 3.0 以降で定義された SQL データ・タイプ標識だけです。JDBC 仕様については、JDBC ドライバーの資料を参照してください。
- JDBC ラッパーは、パススルー・セッションでは LOB をサポートしません。
- タイプ・マッピングとデータ・タイプ変換の制限:
 - サポートされないデータ・タイプ:
ARRAY、DATALINK、DISTINCT、JAVA_OBJECT、REF、STRUCT、および OTHER
 - サポートが限定されているデータ・タイプ:
 - XML データ・タイプのサポートは制限があります。フェデレーテッド・サーバーが CLOB データ・タイプを処理するのは、関連した JDBC データ・タイプが CLOB と SQLXML (JDBC 4.0) のいずれかのデータ・タイプである場合だけです。それ以外は、XML データ・タイプのサポートはありません。
 - DBCS および UNICODE データ・タイプを JDBC ラッパーは UCS-2 として保管します。
 - DB2 データベースの DECFLOAT データ・タイプと Oracle データベースの NUMBER データ・タイプは、対応する JDBC ラッパー・データ・タイプと比べて、有効範囲が大きく、形式が異なっている場合があります。DECFLOAT や NUMBER データ・タイプへのマッピングは不正確な結果を返す可能性があります。

手順

1. JDBC を通じてデータ・ソースへアクセスするためのフェデレーテッド・サーバーを準備します。
2. JDBC ラッパーを登録します。
3. JDBC データ・ソースのサーバー定義を登録します。
4. JDBC データ・ソースのユーザー・マッピングを作成します。
5. JDBC データ・ソース・サーバーへの接続をテストします。
6. JDBC データ・ソースの表およびビューのニックネームを登録します。

JDBC を通じてデータ・ソースへアクセスするためのフェデレーテッド・サーバーの準備

フェデレーテッド・サーバーは、JDBC データ・ソースにアクセスすることが必要です。フェデレーテッド・サーバーを準備するには、CLASSPATH 環境変数を設定する必要があるかどうかを判断しなければなりません。

始める前に

db2jcc.jar ファイル内の DB2 のデフォルト JDBC ドライバー以外の JDBC ドライバーを使用する場合、CLASSPATH 環境変数に JDBC ドライバーの情報を追加することが必要な場合があります。オプションで、サーバー定義の登録時に、使用する JDBC ドライバー・パッケージを CREATE SERVER ステートメントの DRIVER_PACKAGE パラメーターで指定することができます。

手順

使用する JDBC ドライバーを含む Java .jar ファイルを CLASSPATH 環境変数に登録します。

オプション	説明
Linux および UNIX の場合	<p>export コマンドを実行して、使用する JDBC ドライバーを登録します。例えば、DB2 JDBC ドライバーを指定する場合は次のコマンドを実行します。</p> <pre>export CLASSPATH=\$CLASSPATH:db2_instance_dir/sql1lib/java/db2jcc.jar</pre> <p>ここで、<i>db2_instance_dir</i> は、使用する DB2 データベース・システムのインスタンスがインストールされている場所の完全修飾ファイル・パスです。</p>
Windows の場合	<p>以下のように、使用する JDBC ドライバーを CLASSPATH システム環境変数に設定します。</p> <ol style="list-style-type: none">1. 管理者としてログオンします。2. 「コントロール パネル」を開き、「システム」 > 「環境変数」までナビゲートします。3. JDBC ドライバーのファイル名とディレクトリーを CLASSPATH システム変数に追加します。新しい項目と既存のすべての項目はセミコロンを使って区切ってください。 <p>システム環境変数の登録に関する特定の情報は、JDBC ドライバーの資料を参照してください。</p>

次のタスク

この作業が終了したら、ラッパーを登録しなければなりません。

JDBC ラッパーの登録

JDBC データ・ソースにアクセスするためのラッパーを登録する必要があります。フェデレーテッド・サーバーは、ラッパーを使用してデータ・ソースとの通信やデータ・ソースからのデータの取得を行います。ラッパーは、ライブラリー・ファイルのセットとしてインプリメントされます。

手順

次の方法のいずれかを使用します。

方法	説明
コマンド行から CREATE WRAPPER ステートメントを実行し、 JDBC ラッパーのデフォルト名を指定します。	例: <pre>CREATE WRAPPER JDBC;</pre> デフォルト名 JDBC を使用してラッパーを登録する場合、フェデレーテッド・サーバーは自動的に、フェデレーテッド・サーバーが稼働するオペレーティング・システムに対応する適切な JDBC ラッパー・ライブラリーを使用します。
コマンド行から CREATE WRAPPER ステートメントを実行し、 JDBC ラッパーの代替名を指定します。	デフォルト・ラッパー名がフェデレーテッド・データベース中の既存のラッパー名と競合する場合には、任意の名前を選択してデフォルトのラッパー名と置き換えることができます。デフォルト名以外の名前を使用する場合、 CREATE WRAPPER ステートメントに LIBRARY パラメーターを含める必要があります。 例えば、 AIX を使用するフェデレーテッド・サーバー上で、 <i>jdbc_Wrapper</i> という名前でもラッパーを登録するには、次のステートメントを実行します。 <pre>CREATE WRAPPER jdbc_Wrapper LIBRARY 'libdb2rcjdbc.a';</pre> 指定するラッパー・ライブラリー・ファイルは、フェデレーテッド・サーバーのオペレーティング・システムによって異なります。

次のタスク

この作業が終了したら、サーバー定義を登録しなければなりません。

JDBC ラッパー・ライブラリー・ファイル

JDBC ラッパー・ライブラリー・ファイルは、ラッパーのインストール時にフェデレーテッド・サーバーに追加されます。

JDBC ラッパーのインストール時に、デフォルトのディレクトリー・パスにライブラリー・ファイルが追加されます。例えば、フェデレーテッド・サーバーが **AIX** で稼働している場合、ディレクトリー・パスに追加されるラッパー・ライブラリー・ファイルは、*libdb2rcjdbc.a*、*libdb2rcjdbcF.a*、*libdb2rcjdbcU.a*、および *db2qgjdbc.jar* です。デフォルトのラッパー・ライブラリー・ファイルは *libdb2rcjdbc.a* です。他のラッパー・ライブラリー・ファイルは **JDBC** ラッパーによって内部的に使用されます。

ラッパーの登録時にデフォルトのラッパー名を使用しない場合は、**CREATE WRAPPER** ステートメントに **LIBRARY** パラメーターを指定する必要があります。

CREATE WRAPPER ステートメントの **LIBRARY** パラメーターを指定するには、以下のデフォルト・ディレクトリー・パスとラッパー・ライブラリー・ファイル名を使用します。

表 24. JDBC クライアント・ライブラリーのディレクトリー・パスおよびファイル名

オペレーティング・システム	ディレクトリー・パス	ラッパー・ライブラリー・ファイル
AIX	<code>/usr/opt/install_path/lib32/</code> <code>/usr/opt/install_path/lib64/</code>	<code>libdb2rcjdbc.a</code>
Linux	<code>/opt/IBM/db2/install_path/lib32</code> <code>/opt/IBM/db2/install_path/lib64</code>	<code>libdb2rcjdbc.so</code>
Solaris	<code>/opt/IBM/db2/install_path/lib32</code> <code>/opt/IBM/db2/install_path/lib64</code>	<code>libdb2rcjdbc.so</code>
Windows	<code>%DB2PATH%\%bin</code>	<code>db2rcjdbc.dll</code>

`install_path` は、Linux または UNIX でフェデレーションがインストールされているディレクトリー・パスです。

CREATE WRAPPER ステートメント - JDBC ラッパーの例

CREATE WRAPPER ステートメントを使用して、JDBC ラッパーを登録します。

フェデレーテッド・サーバーでデフォルトの名前のラッパーを登録するには、例えば次のように JDBC ラッパー名を指定して CREATE WRAPPER ステートメントを実行します。

```
CREATE WRAPPER JDBC;
```

次の例では、`jdbc_Wrapper` がフェデレーテッド・データベースに登録するラッパーに割り当てる代替名です。

Linux および Solaris のフェデレーテッド・サーバー

以下に示すのは、代替名を指定してラッパーを登録する方法の例です。

```
CREATE WRAPPER jdbc_Wrapper LIBRARY 'libdb2rcjdbc.so';
```

AIX のフェデレーテッド・サーバー

以下に示すのは、代替名を指定してラッパーを登録する方法の例です。

```
CREATE WRAPPER jdbc_Wrapper LIBRARY 'libdb2rcjdbc.a';
```

Windows のフェデレーテッド・サーバー

以下に示すのは、代替名を指定してラッパーを登録する方法の例です。

```
CREATE WRAPPER jdbc_Wrapper LIBRARY 'db2rcjdbc.dll';
```

JDBC データ・ソースのサーバー定義の登録

フェデレーテッド・データベース内に、アクセスする JDBC サーバーをそれぞれ登録する必要があります。

始める前に

DB2 サーバーのデフォルト JDBC ドライバー以外の JDBC ドライバーを使用する場合、CLASSPATH 環境変数を設定して、使用する JDBC ドライバー・パッケージ・ファイルを指定する必要がある場合があります。オプションで、

DRIVER_PACKAGE サーバー・オプションを指定した CREATE SERVER ステートメントで、使用する JDBC ドライバー・パッケージを指定することができます。

このタスクについて

JDBC サーバーを登録するには、JDBC ドライバー・ライブラリーからの JDBC ドライバー・パッケージ名とリモート・サーバーの JDBC 接続ストリングを指定しなければなりません。

手順

登録する JDBC サーバーごとにコマンド行から CREATE SERVER ステートメントを実行します。

例:

```
CREATE SERVER server_Definition_Name
  TYPE jdbc_Datasource_Type
  VERSION version_Number
  WRAPPER jdbc_Wrapper_Name
  OPTIONS (
    DRIVER_CLASS 'jdbc_Driver_Class_Path',
    URL 'jdbc_Url_Connect_String');
```

重要: CREATE SERVER ステートメントを実行しても、データ・ソースへの接続は実際には CREATE NICKNAME ステートメントを実行するまで作成されません。OPTIONS パラメーターで間違った接続情報を指定した場合、CREATE NICKNAME ステートメントかパススルー・ステートメントのいずれかを実行するまで、エラーは通知されません。

次のタスク

この作業が終了したら、ユーザー・マッピングを作成しなければなりません。

CREATE SERVER ステートメント - JDBC ラッパーの例

JDBC ラッパーにサーバー定義を登録するには、CREATE SERVER ステートメントを使用します。この例は、必須パラメーターと追加のサーバー・パラメーターの例を示します。

以下に示すのは、CREATE SERVER ステートメントを実行して、DB2 データ・ソースのサーバー定義を登録する方法の例です。

```
CREATE SERVER jdbc_server1
  TYPE JDBC
  VERSION 3.0
  WRAPPER jdbc_wrapper1
  OPTIONS (
    DRIVER_PACKAGE '/home/My_LIB/JDBC_driver/derbyclient.jar',
    DRIVER_CLASS 'com.ibm.db2.jcc.DB2Driver',
    URL 'jdbc:db2://server.example.com:50471/testdb');
```

パラメーター値

jdbc_server1

JDBC データ・ソース・サーバーに割り当てる名前を指定します。重複するサーバー定義名は使用できません。

TYPE JDBC

アクセスするデータ・ソース・サーバーのタイプを指定します。このパラメーターはオプションです。

VERSION 3.0

アクセスする JDBC データ・ソースのバージョンを指定します。このパラメーターはオプションです。

WRAPPER jdbc_wrapper1

CREATE WRAPPER ステートメントで指定したラッパー名を指定します。

DRIVER_PACKAGE '/home/My_LIB/JDBC_driver/derbyclient.jar'

JDBC ドライバー・パッケージを指定します。

DRIVER_CLASS 'com.ibm.db2.jcc.DB2Driver'

JDBC ドライバー・ライブラリーを指定します。

URL 'jdbc:db2://matthaus.cn.ibm.com:50471/testdb'

リモート・サーバーの JDBC 接続ストリングを指定します。

サーバー・パラメーター

サーバー定義の作成時に、CREATE SERVER ステートメントに追加のサーバー・オプションを指定することができます。サーバー・オプションには、汎用のサーバー・オプションと JDBC 固有のサーバー・パラメーターの両方を含めることができます。

一般的に、サーバー・パラメーターのデフォルト設定では限られた機能しかありません。サーバー・パラメーターを使用して、構成を最適化することができます。

JDBC データ・ソースにアクセスするには、CREATE SERVER ステートメントで DRIVER_CLASS および URL サーバー・パラメーターを指定しなければなりません。DRIVER_PACKAGE および JDBC_LOG サーバー・パラメーターはオプションです。次の OPTIONS パラメーター構文は、すべての JDBC 固有のサーバー・パラメーターを指定しています。

OPTIONS (

```
DRIVER_PACKAGE '/path1/file1.jar: /path2/file2.jar',  
DRIVER_CLASS 'com.ibm.db2.jcc.DB2Driver',  
URL 'jdbc:db2://server.example.com:50471/testdb',  
JDBC_LOG 'Y');
```

パラメーター

DRIVER_PACKAGE '/path1/file1.jar: /path2/file2.jar'

JDBC ドライバー・パッケージを指定し、CLASSPATH 環境変数を設定します。

DRIVER_CLASS 'com.ibm.db2.jcc.DB2Driver'

DB2 JDBC ドライバー・ライブラリーを指定します。

URL 'jdbc:db2://server.example.com:50471/testdb'

コロンで区切られる以下の 3 つの部分からなる JDBC 接続ストリングを指定します。

- データベース・プロトコル
- データベース・タイプ名、または接続ドライバー名

- 別名またはサブネームを使用したデータベース ID

例

DB2 データベースの場合

```
jdbc:db2://server.example.com:50471/testdb
```

jdbc はプロトコル、db2 はデータベースのタイプ、//server.example.com:50471/testdb はデータベース別名で、DB2 クライアント上の DB2 データベース・カタログ項目を参照しています。

Oracle の場合

```
jdbc:oracle:thin:@//myhost:1521/orcl
```

jdbc はプロトコル、oracle はデータベースのタイプ、thin:@//myhost:1521/orcl は Oracle サーバーにアクセスするための Oracle JDBC クライアント情報です。

JDBC_LOG 'Y'

エラー・トレースのためのログ・ファイルを作成することを指定します。このサーバー・オプションのデフォルト値は N です。

例

```
CREATE SERVER jdbc_server1
  TYPE JDBC
  VERSION 3.0
  WRAPPER jdbc_wrapper1
  OPTIONS (
    DRIVER_PACKAGE '/home2/JDBC_driver/derbyclient.jar',
    DRIVER_CLASS 'org.apache.derby.jdbc.ClientDriver',
    URL 'jdbc:derby://9.181.139.129:1527/testdb9;create=true;',
    JDBC_LOG 'Y');
```

JDBC データ・ソースのユーザー・マッピングの作成

各フェデレーテッド・サーバー・ユーザー ID と対応するデータ・ソース・ユーザー ID の間の関連 (ユーザー・マッピング) を定義しなければなりません。

このタスクについて

JDBC サーバーにアクセスしようとする、フェデレーテッド・サーバーはそのデータ・ソースのユーザー ID とパスワードを使用して、JDBC サーバーへの接続を確立します。

フェデレーテッド・システムにアクセスする各ユーザー ID が JDBC データ・ソースに分散要求を送信するためのユーザー・マッピングを作成します。

手順

CREATE USER MAPPING ステートメントを実行して、ローカル・ユーザー ID を JDBC データ・ソースのユーザー ID およびパスワードにマップします。

```
CREATE USER MAPPING FOR local_userID
  SERVER server_definition_name
  OPTIONS (
    REMOTE_AUTHID 'remote_userID',
    REMOTE_PASSWORD 'remote_password');
```

REMOTE_AUTHID および **REMOTE_PASSWORD** ユーザー・マッピング・パラメーターは必須です。

次のタスク

この作業が終了したら、JDBC データ・ソースへの接続をテストできます。

CREATE USER MAPPING ステートメント - JDBC ラッパーの例

この例は、CREATE USER MAPPING ステートメントで DB2 特殊レジスター USER を使用する方法を示します。

以下の例は、フェデレーテッド許可 ID を JDBC データ・ソースのユーザー ID およびパスワードにマップする方法を示しています。

```
CREATE USER MAPPING FOR arturo
  SERVER jdbc_server1
  OPTIONS (
    REMOTE_AUTHID 'art',
    REMOTE_PASSWORD 'red4blue');
```

パラメーター

SERVER arturo

JDBC データ・ソースで定義されているリモート・ユーザー ID およびパスワードにマップするローカル許可 ID を指定します。

OPTIONS jdbc_server1

JDBC データ・ソースに対する CREATE SERVER ステートメントで定義したサーバー定義名を指定します。

REMOTE_AUTHID 'art'

arturo をマップするリモート・ユーザー ID を指定します。CREATE SERVER ステートメントで FOLD_ID サーバー・パラメーターを「U」または「L」にしている場合を除き、この値は大文字と小文字を区別します。

REMOTE_PASSWORD 'red4blue'

'art' と関連付けられたリモート・パスワードを指定します。CREATE SERVER ステートメントで FOLD_PW サーバー・オプションを「U」または「L」に設定している場合を除き、この値は大文字小文字を区別します。

DB2 特殊レジスター USER

DB2 特殊レジスター USER を使用して、CREATE USER MAPPING ステートメントを実行する人の許可 ID を、REMOTE_AUTHID ユーザー・オプションに指定されるデータ・ソースの許可 ID にマップすることができます。

以下に、特殊レジスター USER を含む CREATE USER MAPPING ステートメントの例を示します。

```
CREATE USER MAPPING FOR USER
  SERVER jdbc_server1
  OPTIONS (
    REMOTE_AUTHID 'art',
    REMOTE_PASSWORD 'red4blue');
```

JDBC データ・ソース・サーバーへの接続のテスト

JDBC データ・ソース・サーバーへの接続のテストは、定義したサーバー定義とユーザー・マッピングを使用して行うことができます。

手順

パススルー・セッションを開き、JDBC データ・ソースのシステム表に対して SELECT ステートメントを実行します。SELECT ステートメントがカウントを戻す場合、サーバー定義およびユーザー・マッピングは適切にセットアップされています。

```
SET PASSTHRU server_definition_name
SELECT count(*) FROM schema_name.table_name
SET PASSTHRU RESET
```

SELECT ステートメントがエラーを戻す場合には、接続エラーのトラブルシューティングを行います。

次のタスク

この作業が終了したら、JDBC データ・ソースの表およびビューのニックネームを登録しなければなりません。

データ・ソース接続エラーのトラブルシューティング

データ・ソース・サーバーへのテスト接続では、いくつかの理由でエラーが戻されることがあります。以下の処置により、エラーが発生した理由を判別することができます。

症状

データ・ソースに接続しようとしたときにエラーが戻されます。

原因

接続の問題には、いくつかの原因が考えられます。

問題の解決方法

データ・ソース接続エラーのトラブルシューティングを行うには、以下の項目で問題がないかどうかをチェックします。

- データ・ソースが使用可能であることを確認します。
- 該当する場合は、データ・ソース・サーバーが着信接続に対して構成されていることを確認します。
- REMOTE_AUTHID と REMOTE_PASSWORD のオプションに関するユーザー・マッピング設定を調べ、データ・ソースとの接続に有効な設定になっているか確認します。必要に応じて、ユーザー・マッピングを変更するか、別のユーザー・マッピングを作成します。
- 該当する場合は、フェデレーテッド・サーバー上のデータ・ソース・クライアント・ソフトウェアがインストールされていて、データ・ソースに接続できるように正しく構成されていることを確認します。

- ODBC データ・ソースの場合、フェデレーテッド・サーバー上の ODBC ドライバーがインストールされていて、ODBC データ・ソース・サーバーに接続できるように正しく構成されていることを確認します。Windows が稼働するフェデレーテッド・サーバーでは、ドライバーをチェックするときに「ODBC データ ソース アドミニストレータ」ツールを使用します。UNIX が稼働するフェデレーテッド・サーバーの場合は、ODBC クライアントのベンダーが提供している資料を参照してください。
- フェデレーテッド・サーバー上で設定されている変数の設定がデータ・ソースについて適切であることを確認します。これらの変数には、システム環境変数、db2dj.ini ファイル、および DB2 Profile Registry (db2set) 変数が含まれます。
- サーバー定義を調べます。必要であれば、サーバー定義をドロップして再作成します。

JDBC データ・ソースの表およびビューのニックネームの登録

登録するそれぞれの JDBC サーバー定義で、アクセスするそれぞれの表またはビューごとにニックネームを登録する必要があります。JDBC データ・ソースを照会するときには、データ・ソース・オブジェクトの名前ではなく、これらのニックネームを使用します。

始める前に

ニックネームを登録する前に、JDBC データ・ソース側の統計を更新します。フェデレーテッド・データベースは、データ・ソースのカatalog統計に基づいて、照会処理を最適化します。DB2 **RUNSTATS** コマンドに相当するデータ・ソース・コマンドを使用して、データ・ソースの統計を更新します。

手順

コマンド行から CREATE NICKNAME ステートメントを発行します。

例:

```
CREATE NICKNAME nickname
FOR server_definition_name."remote_schema"."remote.table";
```

ニックネームの長さは 128 文字までです。

ニックネームの作成時に、フェデレーテッド・サーバーはデータ・ソース・カatalogを照会します。この照会は、データベース・ソースの表またはビューへの接続をテストします。接続ができない場合、エラー・メッセージが出ます。

次のタスク

ニックネームを作成する JDBC の表またはビューごとに、このステップを繰り返します。

CREATE NICKNAME ステートメント - JDBC ラッパーの例

この例で示すのは、CREATE NICKNAME ステートメントを使用して、JDBC 表またはビューのニックネームを登録する方法です。

このステートメントは、サーバー定義およびリモート・スキーマと表を指定しています。

```
CREATE NICKNAME cust_europe FOR jdbc_server."vinnie"."italy"
```

cust_europe

JDBC の表またはビューを識別するために使用する固有なニックネームです。ニックネームはスキーマ内では固有でなければなりません。

重要: ニックネームは、スキーマとニックネームという 2 部構成の名前です。ニックネームの登録時にスキーマを省略すると、そのニックネームのスキーマはニックネームを登録するユーザーの許可 ID に設定されます。

お使いの JDBC データ・ソースがスキーマをサポートしない場合、CREATE NICKNAME ステートメントでスキーマを指定しないでください。以下に例をあげます。

```
CREATE NICKNAME cust_europe FOR jdbc_server."italy"
```

jdbc_server."vinnie"."italy"

リモート・オブジェクトを表す、3 つの部分からなる ID。

jdbc_server

CREATE SERVER ステートメントで JDBC データ・ソース・サーバーに割り当てたサーバー定義名です。

vinnie 表またはビューが属する所有者のユーザー ID です。

italy アクセスするリモート表またはビューの名前です。

フェデレーテッド・サーバーは、名前を二重引用符で囲んでいない限り、JDBC スキーマと表の名前を大文字に変換します。

Microsoft SQL Server データ・ソースへのアクセスの構成

Microsoft SQL Server データ・ソースにアクセスするようにフェデレーテッド・サーバーを構成するには、アクセスしたいデータ・ソースとオブジェクトの情報をフェデレーテッド・サーバーに提供しなければなりません。

始める前に

- ODBC ドライバーがフェデレーテッド・サーバー上にインストール・構成されていないければなりません。
- フェデレーションが、フェデレーテッド・サーバーとして機能するサーバー上にインストールされている必要があります。
- フェデレーテッド・サーバーのセットアップを確認します。
- フェデレーテッド・パラメーターをチェックして、フェデレーションが有効になっていることを確認します。

手順

1. オペレーティング・システムに応じて、以下の方法のいずれかを使用して、フェデレーテッド・サーバーとフェデレーテッド・データベースを準備します。
 - フェデレーテッド・サーバーとフェデレーテッド・データベースを準備します (Windows)。
 - フェデレーテッド・サーバーとフェデレーテッド・データベースを準備します (UNIX)。
2. Microsoft SQL Server ラッパー用の環境変数を設定します。

3. ラッパーを登録します。
4. サーバー定義を登録します。
5. ユーザー・マッピングを作成します。
6. Microsoft SQL Server リモート・サーバーへの接続をテストします。
7. Microsoft SQL Server の表およびビューのニックネームを登録します。

Microsoft SQL Server データ・ソースにアクセスするためのフェデレーテッド・サーバーの準備 (Windows)

Windows を実行するフェデレーテッド・サーバーは、Microsoft SQL Server データ・ソースにアクセスできなければなりません。フェデレーテッド・サーバーを準備するには、ODBC システム DSN 内の設定を検査し、Microsoft SQL Server データ・ソースへの接続をテストする必要があります。

手順

1. ODBC システム DSN が Microsoft SQL Server データ・ソースに接続するように設定されていることを確認します。コントロール・パネルで、Microsoft SQL Server リモート・サーバーの既存の DSN 項目を見付けるか、DSN 項目を作成します。

Microsoft SQL Server リモート・サーバーの DSN 項目は、フェデレーテッド・データベースにサーバー定義を登録するときに NODE サーバー・オプションに使用する値です。

2. 以下の方法のいずれかを使用して、Microsoft SQL Server データ・ソースへの接続をテストします。
 - 「ODBC データ ソース アドミニストレータ」ウィンドウから、「構成」を選択します。
 - Microsoft SQL Server 照会ツールを使用します。

次のタスク

この作業が完了したら、環境変数を設定することができます。

Microsoft SQL Server データ・ソースにアクセスするためのフェデレーテッド・サーバーの準備 (Linux、UNIX)

Linux または UNIX を実行するフェデレーテッド・サーバーは、Microsoft SQL Server データ・ソースにアクセスできなければなりません。フェデレーテッド・サーバーを準備するには、odbc.ini ファイル内の設定を検査し、シンボリック・リンクを作成してから、Microsoft SQL Server データ・ソースへの接続をテストする必要があります。

手順

1. フェデレーテッド・サーバー上の odbc.ini ファイルが更新されていることを確認してください。odbc.ini ファイルがフェデレーテッド・サーバーに存在しない場合は、テキスト・エディターで作成することができます。odbc.ini ファイルについては、ODBC クライアント・ベンダーの資料を参照してください。

要確認: `odbc.ini` ファイルやこのファイルのコピーは、DB2 インスタンス所有者のホーム・ディレクトリーに置きます。これにより、インスタンス所有者が `root` ユーザーでない場合にもアクセスできます。

2. `odbc.ini` へのパスが `ODBCINI` 環境変数にあることを確認します。

オペレーティング・システムのコマンド・プロンプトから次のコマンドを発行します。

```
export ODBCINI=$HOME/.odbc.ini
```

3. 適切なシンボリック・リンクを作成します。

フェデレーテッド・サーバーのオペレーティング・システム	ステップ
Linux	<p>インストールする ODBC ドライバーに応じて、シンボリック・リンクを作成します。以下に例を示します。</p> <pre>ln -s \$DJX_ODBC_LIBRARY_PATH/.. /locale/usr/local/locale</pre> <pre>ln -s \$DJX_ODBC_LIBRARY_PATH /libodbcinst.so/ usr/lib/libodbcinst.so</pre> <p>例</p> <ul style="list-style-type: none"> • IBM Branded DataDirect ODBC ドライバーの場合: <ul style="list-style-type: none"> - バージョン 5.2 のシンボリック・リンクは次のように作成します。 <pre>ln -s \$DJX_ODBC_LIBRARY_PATH/ libVMicu22.so /usr /lib/libVMicu22.so</pre> - バージョン 5.3 のシンボリック・リンクは次のように作成します。 <pre>ln -s \$DJX_ODBC_LIBRARY_PATH /libVMicu23.so /usr /lib/libVMicu23.so</pre> • DataDirect Technologies Connect for ODBC ドライバーの場合: <ul style="list-style-type: none"> - バージョン 4.2 のシンボリック・リンクは次のように作成します。 <pre>ln -s \$DJX_ODBC_LIBRARY_PATH /libivicu19.so/usr /lib/libivicu19.so</pre> - バージョン 5.0 のシンボリック・リンクは次のように作成します。 <pre>ln -s \$DJX_ODBC_LIBRARY_PATH/ libivicu20.so/usr/lib /lib/libivicu20.so</pre> <p>シンボリック・リンクを作成しない場合、CREATE WRAPPER MSSQLODBC3 ステートメントが失敗し、以下のエラー・メッセージが表示される可能性があります。</p> <pre>SQL10013N 指定した ライブラリー (name) を ロードできませんでした。</pre>

フェデレーテッド・サーバーのオペレーティング・システム	ステップ
Solaris	以下のシンボリック・リンクを作成します。 <pre>In -s \$DJX_ODBC_LIBRARY_PATH/./ locale \$HOME/sqllib/locale</pre> ここで、\$HOME は、DB2 インスタンス所有者のホーム・ディレクトリーです。

以下の表に、各 ODBC ドライバー・バージョンごとのライブラリーの名前をリストします。

表 25.

ODBC ドライバーのバージョン	ライブラリー名
4.2	libodbcinst.so libVMicu19.so
5.0	libodbcinst.so libVMicu20.so
5.1	libodbcinst.so libVMicu21.so
5.3	libodbcinst.so libVMicu23.so
6.0	libodbcinst.so libVMicu24.so

4. ODBC ドライバーをインストールします。ODBC ドライバーをインストールする方法については、『ODBC ドライバーのインストール』を参照してください。

次のタスク

この作業が完了したら、環境変数を設定することができます。

Microsoft SQL Server 環境変数の設定

フェデレーテッド・サーバー上の db2dj.ini ファイルで、Microsoft SQL Server 環境変数を設定する必要があります。

このタスクについて

制約事項

この作業を始める前に、db2dj.ini ファイルに関する制限を確認してください。

db2dj.ini ファイルには、フェデレーテッド・サーバーにインストールされている Microsoft SQL Server ODBC ドライバーに関する構成情報が含まれます。

Microsoft SQL Server データ・ソースには、必須の環境変数とオプションの環境変数があります。

Microsoft SQL Server ラッパーをインストールする前に Microsoft SQL Server クライアント・ソフトウェアをインストールした場合、必要な Microsoft SQL Server 環境変数は db2dj.ini ファイルで設定されています。

Microsoft SQL Server ラッパーをインストールする前に Microsoft SQL Server クライアント・ソフトウェアをインストールしなかった場合や、いずれかのオプションの環境変数を設定する場合は、この作業のステップを使用して環境変数を設定する必要があります。

手順

1. 次の方法のいずれかを使用します。

方法	ステップ
ラッパー構成ツールで環境変数を設定します。	ラッパー構成ツールを実行し、必須の環境変数とオプションの環境変数を設定します。ラッパー構成ツールの実行の指示に従ってください。
環境変数を手動で設定します。	<p>db2dj.ini ファイルを編集します。</p> <ul style="list-style-type: none"> UNIX が稼働するフェデレーテッド・サーバーの場合、このファイルは sql1lib/cfg ディレクトリーにあります。 Windows を実行しているフェデレーテッド・サーバーの場合、このファイルは %DB2PATH%\cfg ディレクトリー内にあります。 <p>このファイルが存在しない場合、任意のテキスト・エディターを使用して、db2dj.ini という名前のファイルを作成しても構いません。db2dj.ini ファイルには、環境変数の値に完全修飾パスを指定する必要があります。そうしないと、エラーが発生します。</p> <p>例:</p> <pre>DJX_ODBC_LIBRARY_PATH=/opt/odbc/lib ODBCINI=/opt/odbc/.odbc.ini</pre>

2. 環境変数がフェデレーテッド・サーバーで設定されていることを確認するには、DB2 インスタンスをリサイクルします。

```
db2stop
db2start
```

次のタスク

この作業が完了したら、ラッパーを登録することができます。

Microsoft SQL Server 環境変数

Microsoft SQL Server データ・ソースには、必須の環境変数とオプションの環境変数があります。これらの変数は db2dj.ini ファイルに設定されます。

Microsoft SQL Server では、以下の環境変数が有効です。

- DJX_ODBC_LIBRARY_PATH
- ODBCINI
- LD_LIBRARY_PATH (Solaris のみ)

変数の説明

DJX_ODBC_LIBRARY_PATH

ODBC ライブラリー・ファイルへのディレクトリー・パスを指定します。
この変数は、Solaris を実行するフェデレーテッド・サーバーでも指定する必要があります。

例:

```
DJX_ODBC_LIBRARY_PATH=ODBC_driver_directory/lib
```

ODBC_driver_directory は、ODBC ドライバーがインストールされているディレクトリー・パスです。

ODBCINI

ODBC 構成ファイル (*odbc.ini*) があるディレクトリー・パスを指定します。

例:

```
ODBCINI=/home/db2inst1/.odbc.ini
```

ODBCINI 環境変数はシステム変数として設定してはなりません。

LD_LIBRARY_PATH (Solaris のみ)

Solaris を実行するフェデレーテッド・サーバーで ODBC ライブラリー・ファイルへのディレクトリー・パスを指定します。

例:

```
LD_LIBRARY_PATH=ODBC_driver_directory/lib
```

Microsoft SQL Server ラッパーの登録

Microsoft SQL Server データ・ソースにアクセスするためのラッパーを登録しなければなりません。フェデレーテッド・サーバーは、ラッパーを使用してデータ・ソースとの通信やデータ・ソースからのデータの取得を行います。ラッパーは、ライブラリー・ファイルのセットとしてインプリメントされます。

手順

コマンド行から CREATE WRAPPER ステートメントを発行し、Microsoft SQL Server ラッパーのデフォルト名を指定します。

例:

```
CREATE WRAPPER MSSQLODBC3;
```

要確認: デフォルト名 MSSQLODBC3 を使用してラッパーを登録する場合、フェデレーテッド・サーバーは、フェデレーテッド・サーバーが実行されているオペレーティング・システムに適した Microsoft SQL Server ラッパー・ライブラリーを自動的に使用します。

デフォルトのラッパー名がフェデレーテッド・データベース内の既存のラッパー名と競合する場合、デフォルトのラッパー名を、指定の名前で置き換えることができます。デフォルト名とは別の名前を使用する場合は、CREATE WRAPPER ステートメントに LIBRARY パラメーターを含める必要があります。

たとえば、AIX を使用するフェデレーテッド・サーバー上で *sqlserver_wrapper* という名前のラッパーを登録するには、次のステートメントを発行します。

```
CREATE WRAPPER sqlserver_wrapper
  LIBRARY 'libdb2mssql3.a';
```

指定するラッパー・ライブラリー・ファイルは、フェデレーテッド・サーバーのオペレーティング・システムによって異なります。

次のタスク

この作業が完了したら、サーバー定義を登録することができます。

Microsoft SQL Server ラッパー・ライブラリー・ファイル

Microsoft SQL Server ラッパー・ライブラリー・ファイルはラッパーのインストール時にフェデレーテッド・サーバーに追加されます。

Microsoft SQL Server ラッパーをインストールすると、デフォルトのディレクトリー・パスに 3 つのライブラリー・ファイルが追加されます。例えば、フェデレーテッド・サーバーが AIX 上で実行されている場合、ディレクトリー・パスに追加されるラッパー・ライブラリー・ファイルは libdb2mssql3.a、libdb2mssql3F.a、および libdb2mssql3U.a です。デフォルトのラッパー・ライブラリー・ファイルは libdb2mssql3.a です。他のラッパー・ライブラリー・ファイルは Microsoft SQL Server ラッパーによって内部で使用されます。

ラッパーを登録する際にデフォルトのラッパー名を使用しない場合は、CREATE WRAPPER ステートメントに LIBRARY パラメーターを含め、デフォルトのラッパー・ライブラリー・ファイル名を指定する必要があります。

以下の表に、デフォルトのディレクトリー・パスとデフォルトのラッパー・ライブラリー・ファイルの名前がリストされています。

表 26. Microsoft SQL Server クライアント・ライブラリーの場所とファイル名

オペレーティング・システム	ディレクトリー・パス	ライブラリー・ファイル名
AIX	/usr/opt/install_path/lib32/ /usr/opt/install_path/lib64/	libdb2mssql3.a
Linux	/opt/IBM/db2/install_path/lib32 /opt/IBM/db2/install_path/lib64	libdb2mssql3.so
Solaris	/opt/IBM/db2/install_path/lib32 /opt/IBM/db2/install_path/lib64	libdb2mssql3.so
Windows	%DB2PATH%\bin	db2mssql3.dll

install_path は、UNIX または Linux 上でフェデレーテッド・サーバーがインストールされているディレクトリー・パスです。

Microsoft SQL Server データ・ソースのサーバー定義の登録

フェデレーテッド・データベースでアクセスする各 Microsoft SQL Server リモート・サーバーを登録する必要があります。

手順

1. Microsoft SQL Server のノード名を見付けます。

- Windows を実行するフェデレーテッド・サーバーの場合、このノード名は、アクセスするMicrosoft SQL Server リモート・サーバーに指定したシステム DSN 名です。
- UNIX を実行するフェデレーテッド・サーバーの場合、このノード名は .odbc.ini ファイルに定義されています。

.odbc.ini ファイルの先頭に ODBC Data Sources とラベルの付いたセクションがあり、そこにノードがリストされています。ノードごとに、各ノードを記述するセクションが .odbc.ini ファイルにあります。

以下の例は、AIX での .odbc.ini ファイルです。ノード名は [rawilson] と [medusa] です。

```
[ODBC Data Sources]
rawilson=MS SQL Server 2000
medusa=MS SQL Server 2000
[rawilson]
Driver=/opt/odbc/lib/ddmsss20.so
Description=MS SQL Server Driver for AIX
Address=9.112.30.39,1433
[medusa]
Driver=/opt/odbc/lib/ddmsss20.so
Description=MS SQL Server Driver for AIX
Address=9.112.98.123,1433
[ODBC]
InstallDir=/opt/odbc
```

2. コマンド行から CREATE SERVER ステートメントを発行し、サーバー定義を作成します。

例:

```
CREATE SERVER server_definition_name
TYPE MSSQLSERVER
VERSION version_number
WRAPPER wrapper_name
OPTIONS (NODE 'node_name',
DBNAME 'database_name');
```

'*node_name*' および '*db_name*' 変数は CREATE SERVER ステートメントではオプションとして指定されますが、Microsoft SQL Server データ・ソースの場合は必須です。

サーバー定義を登録した後、サーバー・オプションを追加またはドロップするには、ALTER SERVER ステートメントを使用します。

次のタスク

この作業を完了したら、ユーザー・マッピングを作成することができます。

CREATE SERVER ステートメント - Microsoft SQL Server ラッパーの例

Microsoft SQL Server ラッパーにサーバー定義を登録するには、CREATE SERVER ステートメントを使用します。このトピックでは、必須パラメーターを使用する完全指定の例と、追加のサーバー・オプションを使用する例を提供します。

以下の例は、CREATE SERVER ステートメントを発行して Microsoft SQL の Server ラッパーのサーバー定義を登録する方法を示しています。

```
CREATE SERVER sqlserver TYPE MSSQLSERVER VERSION 2000 WRAPPER wrapper_name  
OPTIONS (NODE 'sqlnode', DBNAME 'africa');
```

sqlserver

Microsoft SQL Server リモート・サーバーに割り当てる名前。重複するサーバー定義名は使用できません。

TYPE MSSQLSERVER

アクセスを構成しているデータ・ソースのタイプを指定します。Microsoft SQL Server ラッパーの場合、サーバー・タイプは MSSQLSERVER でなければなりません。

VERSION 2000

アクセスしたい Microsoft SQL Server データベース・サーバーのバージョン。

WRAPPER wrapper_name

CREATE WRAPPER ステートメントで指定したラッパー名。

NODE 'sqlnode'

Microsoft SQL Server リモート・サーバーが存在するノードの名前。Windows を実行するフェデレーテッド・サーバーの場合、アクセスする Microsoft SQL Server リモート・サーバーに指定したシステム DSN 名です。UNIX を実行するフェデレーテッド・サーバーの場合、.odbc.ini ファイルに定義されたノードです。

この値は大/小文字の区別があります。

ノード名は CREATE SERVER ステートメントではオプションとなっていますが、Microsoft SQL Server データ・ソースの場合は必要です。

DBNAME 'africa'

アクセスしたい Microsoft SQL Server データベースの名前。この値は大/小文字の区別があります。

データベース名は CREATE SERVER ステートメントではオプションとなっていますが、Microsoft SQL Server データ・ソースの場合は必要です。

サーバー・オプション

サーバー定義の作成時に、CREATE SERVER ステートメントに追加のサーバー・オプションを指定することができます。サーバー・オプションには、汎用サーバー・オプションと Microsoft SQL Server に特有のサーバー・オプションを指定できます。

COLLATING_SEQUENCE サーバー・オプションは、データ・ソースがフェデレーテッド・サーバーと同じ照合シーケンスを使用しているか、異なる照合シーケンスを使用しているかを指定します。Windows NT または Windows 2000 で稼働する Microsoft SQL Server データベース・サーバーでは、デフォルトの照合シーケンスは大文字小文字を区別しません (例: 'STEWART' と 'StewART' は等しいと見なされます)。フェデレーテッド・サーバーからの正しい結果を保証するには、

COLLATING_SEQUENCE サーバー・オプションを 'I' にしてください。この設定は、Microsoft SQL Server データ・ソースで大文字小文字を区別しないことを指示しています。

データ・ソースから戻される結果がフェデレーテッド・サーバーで照会を処理するときに戻される結果と異なる場合、フェデレーテッド・サーバーは照会をプッシュダウンしません。COLLATING_SEQUENCE サーバー・オプションを 'I' にすると、フェデレーテッド・サーバーは、ストリング・データまたは式を含み、以下の文節、述部、または関数を含む照会をプッシュダウンしません。

- GROUP BY 文節
- DISTINCT 文節
- 基本述部 (例: 等しい (=))
- 集約関数 (例: MIN や MAX)

以下に示すのは、COLLATING_SEQUENCE サーバー・オプションの指定方法の例です。

```
CREATE SERVER sqlserver TYPE MSSQLSERVER VERSION 2000 WRAPPER mssqlodbc3
OPTIONS (NODE 'sqlnode', DBNAME 'africa', COLLATING_SEQUENCE 'I');
```

Microsoft SQL Server データ・ソース用のユーザー・マッピングの作成

Microsoft SQL Server リモート・サーバーへのアクセスを試行する際、フェデレーテッド・サーバーはそのデータ・ソースで有効なユーザー ID とパスワードを使用して、Microsoft SQL Server リモート・サーバーへの接続を確立します。

手順

CREATE USER MAPPING ステートメントを発行します。
たとえば、次のようにします。

```
CREATE USER MAPPING FOR local_userID SERVER server_definition_name
OPTIONS (REMOTE_AUTHID 'remote_userID', REMOTE_PASSWORD 'remote_password');
```

次のタスク

この作業を完了したら、Microsoft SQL Server の表とビューへの接続をテストすることができます。

CREATE USER MAPPING ステートメント - Microsoft SQL Server ラッパーの例

CREATE USER MAPPING ステートメントを使用して、フェデレーテッド許可 ID をリモート Microsoft SQL Server のユーザー ID およびパスワードにマップします。このトピックには、必須パラメーターを使用する完全指定の例と、CREATE USER MAPPING ステートメントで DB2 特殊レジスター USER を使用する方法を示す例が含まれています。

以下に示すのは、フェデレーテッド許可 ID を Microsoft SQL Server リモート・サーバーのユーザー ID およびパスワードにマップする方法を示す例です。

```
CREATE USER MAPPING FOR elizabeth SERVER sqlserver
OPTIONS (REMOTE_AUTHID 'liz', REMOTE_PASSWORD 'abc123')
```

elizabeth

Microsoft SQL Server リモート・サーバーで定義されたりモート・ユーザー ID およびパスワードにマップする許可 ID を指定します。

SERVER *sqlserver*

CREATE SERVER ステートメントで Microsoft SQL Server リモート・サーバー用に登録したサーバー定義名を指定します。

REMOTE_AUTHID '*liz*'

elizabeth をマップするリモート・ユーザー ID を指定します。CREATE SERVER ステートメントで FOLD_ID サーバー・オプションを「U」または「L」に設定している場合を除き、この値は大文字小文字を区別します。

REMOTE_PASSWORD '*abc123*'

liz に関連するリモート・パスワードを指定します。CREATE SERVER ステートメントで FOLD_PW サーバー・オプションを「U」または「L」に設定している場合を除き、この値は大文字小文字を区別します。

DB2 特殊レジスター USER

DB2 特殊レジスター USER を使用して、CREATE USER MAPPING ステートメントを実行する人の許可 ID を、REMOTE_AUTHID オプションに指定されるリモート・ユーザー ID にマップすることができます。

以下に、特殊レジスター USER を含む CREATE USER MAPPING ステートメントの例を示します。

```
CREATE USER MAPPING FOR USER SERVER sqlserver
  OPTIONS (REMOTE_AUTHID 'liz', REMOTE_PASSWORD 'abc123');
```

Microsoft SQL Server リモート・サーバーへの接続のテスト

Microsoft SQL Server リモート・サーバーへの接続をテストして、フェデレーテッド・サーバーが MicrosoftSQL Server データ・ソースにアクセスするように適切に構成されているかどうかを判別します。

このタスクについて

Microsoft SQL Server リモート・サーバーへの接続のテストは、定義したサーバー定義とユーザー・マッピングを使用して行うことができます。

手順

パススルー・セッションを開き、Microsoft SQL Server システム表に対して SQL SELECT ステートメントを発行します。SELECT ステートメントがカウントを戻す場合、サーバー定義およびユーザー・マッピングは適切にセットアップされています。

例:

```
SET PASSTHRU server_definition_name
SELECT count(*) FROM dbo.sysobjects
SET PASSTHRU RESET
```

次のタスク

この作業を完了したら、ニックネームを登録することができます。

データ・ソース接続エラーのトラブルシューティング

データ・ソース・サーバーへのテスト接続では、いくつかの理由でエラーが戻されることがあります。以下の処置により、エラーが発生した理由を判別することができます。

症状

データ・ソースに接続しようとしたときにエラーが戻されます。

原因

接続の問題には、いくつかの原因が考えられます。

問題の解決方法

データ・ソース接続エラーのトラブルシューティングを行うには、以下の項目で問題がないかどうかをチェックします。

- データ・ソースが使用可能であることを確認します。
- 該当する場合は、データ・ソース・サーバーが着信接続に対して構成されていることを確認します。
- REMOTE_AUTHID と REMOTE_PASSWORD のオプションに関するユーザー・マッピング設定を調べ、データ・ソースとの接続に有効な設定になっているか確認します。必要に応じて、ユーザー・マッピングを変更するか、別のユーザー・マッピングを作成します。
- 該当する場合は、フェデレーテッド・サーバー上のデータ・ソース・クライアント・ソフトウェアがインストールされていて、データ・ソースに接続できるように正しく構成されていることを確認します。
- ODBC データ・ソースの場合、フェデレーテッド・サーバー上の ODBC ドライバーがインストールされていて、ODBC データ・ソース・サーバーに接続できるように正しく構成されていることを確認します。Windows が稼働するフェデレーテッド・サーバーでは、ドライバーをチェックするときに「ODBC データ ソース アドミニストレータ」ツールを使用します。UNIX が稼働するフェデレーテッド・サーバーの場合は、ODBC クライアントのベンダーが提供している資料を参照してください。
- フェデレーテッド・サーバー上で設定されている変数の設定がデータ・ソースについて適切であることを確認します。これらの変数には、システム環境変数、db2dj.ini ファイル、および DB2 Profile Registry (db2set) 変数が含まれます。
- サーバー定義を調べます。必要であれば、サーバー定義をドロップして再作成します。

Microsoft SQL Server 表およびビューのニックネームの登録

登録するそれぞれの Microsoft SQL Server リモート・サーバー定義に対して、アクセスしたい表またはビューごとにニックネームを登録する必要があります。

Microsoft SQL Server リモート・サーバーを照会するときには、データ・ソース・オブジェクトの名前ではなく、これらのニックネームを使用します。

始める前に

フェデレーテッド・データベースの統計が現行の、完全なものとなるようにするには、ニックネームを作成する前に、Microsoft SQL Server の `sp_createstats` ストアド・プロシージャおよび Microsoft SQL Server の **CREATE STATISTICS** コマンドを Microsoft SQL Server データベースから実行します。

`sp_createstats` ストアド・プロシージャは Microsoft SQL Server データ・ソースの表にあるすべてのデフォルトの列に関する統計を収集しますが、索引内で最初に現れる列の統計は収集しません。フェデレーテッド・データベースが Microsoft SQL Server 表に関する完全な統計を持つようにするには、Microsoft SQL Server の **CREATE STATISTICS** コマンドをさらに使用して、索引内で最初に現れる各列の統計を収集しなければなりません。

Microsoft SQL Server データベースから **CREATE STATISTICS** コマンドを使用する場合、統計の名前を、その統計が収集される列の名前と同じにしなければなりません。統計の名前を列の名前と同じにすることにより、**CREATE NICKNAME** ステートメントを使ってニックネームを登録するときに Microsoft SQL Server の **CREATE STATISTICS** コマンドによって収集される統計をフェデレーテッド・データベースが読み取るようにします。

手順

コマンド行から **CREATE NICKNAME** ステートメントを発行します。

例:

```
CREATE NICKNAME nickname  
  
FOR server_definition_name."remote_schema"."remote.table";
```

ニックネームを作成すると、フェデレーテッド・サーバーはニックネームを使用してデータ・ソース・カタログを照会します。この照会は、データベース・ソースの表、ビュー、またはシノニムへの接続をテストします。接続ができない場合、エラー・メッセージが出ます。

次のタスク

ニックネームを作成したい Microsoft SQL Server の表またはビューごとに、このステップを繰り返します。

CREATE NICKNAME ステートメント - Microsoft SQL Server ラッパーの例

CREATE NICKNAME ステートメントを使用して、アクセスする Microsoft SQL Server の表またはビューのニックネームを登録します。必須パラメーターを使用した完全指定の例がこのトピックには含まれています。

以下の例は、**CREATE NICKNAME** ステートメントを使用して Microsoft SQL Server の表またはビューのニックネームを登録する方法を示しています。

```
CREATE NICKNAME cust_africa FOR sqlserver."vinnie"."egypt"
```

cust_africa

Microsoft SQL Server の表またはビューを識別するために使用する固有なニックネームです。

重要: ニックネームは、2 つの部分 (スキーマとニックネーム) からなる名前です。ニックネームの登録時にスキーマを省略すると、そのニックネームのスキーマはニックネームを登録するユーザーの許可 ID になります。

sqlserver."vinnie"."egypt"

リモート・オブジェクトを表す、3 つの部分からなる ID。

- *sqlserver* は、CREATE SERVER ステートメントで Microsoft SQL Server リモート・サーバーに割り当てたサーバー定義名です。
- *vinnie* は、表またはビューが属する所有者のユーザー ID です。
- *egypt* は、アクセスしたいリモートの表またはビューの名前です。

フェデレーテッド・サーバーは、引用符で名前を囲まない限り、Microsoft SQL Server スキーマおよび表の名前を大文字に変換します。

ODBC トレース情報による、Microsoft SQL Server データ・ソースへの接続のトラブルシューティング

データ・ソースへの接続に問題がある場合、ODBC トレース情報を入手して問題を分析し、解決することができます。

症状

しかしながら、トレースをオンにすると、システム・パフォーマンスに影響が出ます。接続に関する問題が解決したら、トレースはオフにすべきです。

Microsoft SQL Server ラッパーでデータ・ソースに接続できない場合、トレースの実行は問題の診断に役立つことがあります。

原因

この問題の原因は、ラッパー構成の誤りである可能性があります。

問題の診断

Windows を実行するフェデレーテッド・サーバーの問題を診断する場合:

1. 「コントロール パネル」で「管理ツール」フォルダーを開きます。
2. 「データ ソース (ODBC)」をクリックし、「ODBC データ ソース アドミニストレータ」ウィンドウを開きます。
3. 「トレース」タブをクリックします。
4. 「トレースの開始」をクリックし、トレース・ユーティリティを開始します。

UNIX を実行するフェデレーテッド・サーバーの場合:

1. `odbc.ini` ファイルを変更します。

たとえば、DataDirect ODBC 3.x ドライバーを使用している場合、クライアント・ディレクトリで `odbc.ini` ファイルの例を見つけてください。 `odbc.ini` ファイルには、トレース・ファイルを活動化するのに必要な設定の例が含まれています。

```
[ODBC]
Trace=1
TraceFile=/home/user1/trace_dir/filename.xxx
TraceDll=ODBC_driver_directory/odbc32.dll
InstallDir=/opt/odbc
```

トレースをオンにするには、1 行目を `Trace=1` に設定します。トレースをオフにするには、1 行目を `Trace=0` に設定します。 `TraceFile` 設定の値は、フェデレーテッド・データベース・インスタンスが書き込みアクセス権を持っているパスおよびファイル名です。

問題の解決方法

トレース・ログ・ファイルに問題がないかどうか調べます。

Windows の場合、「ODBC データ ソース アドミニストレータ」を開き、「トレース」タブをクリックします。トレース・ログ・ファイルへのパスは「ログ ファイルのパス」フィールドに示されています。

UNIX の場合、`odbc.ini` ファイルを開きます。トレース・ログ・ファイルへのパスは `TraceFile` 設定で示されています。

ODBC データ・ソースへのアクセスの構成

ODBC データ・ソースにアクセスするようにフェデレーテッド・サーバーを構成するには、アクセスするデータ・ソースとオブジェクトの情報をフェデレーテッド・サーバーに提供しなければなりません。

始める前に

- ODBC ドライバーが、フェデレーテッド・サーバーとして機能するサーバー上にインストールされ、構成されている必要があります。
- フェデレーションが、フェデレーテッド・サーバーとして機能するサーバー上にインストールされている必要があります。
- フェデレーテッド・サーバーのセットアップを検証します。
- フェデレーテッド・パラメーターをチェックして、フェデレーションが有効になっていることを確認します。
- 以下の変数がデータ・ソース・クライアントに応じて、ご使用のシステム環境と `db2dj.ini` ファイルの両方に指定されていることを確認する必要があります。
 - データ・ソース・クライアントが必要とするシステム環境変数
 - DB2 プロファイル・レジストリー (`db2set`) の `DB2LIBPATH` 変数

必要な変数を判別するには、データ・ソース・クライアントにより提供される資料を参照してください。

例えば、ランタイム・ライブラリーを指定する必要がある場合は、以下の環境変数を指定することが必要になります。

- LIBPATH (AIX オペレーティング・システム)
- PATH (Windows オペレーティング・システム)
- LD_LIBRARY_PATH (Linux および Solaris オペレーティング・システム)

注: システム環境変数の構成や db2dj.ini ファイルの変更を行う場合には、InfoSphere Federation Server を一度停止してから再始動する必要があります。

このタスクについて

制約事項

- ODBC ラッパーを使用して DB2 ファミリーのデータ・ソースにアクセスすることはできません。DB2 ファミリーのデータ・ソースにアクセスするには、DRDA ラッパーを使用してください。
- ODBC ラッパーは、以下の機能やステートメントをサポートしていません。
 - 2 フェーズ・コミット・トランザクション
 - ニックネームでの LOCK TABLE ステートメント
 - ODBC 3.x で推奨されない機能
 - X/Open または SQL/CLI ドライバー
 - ストアード・プロシージャのニックネーム
 - リモート・セーブポイント・ステートメントを使用した、ステートメント・レベルの原子性の保証
 - WITH HOLD カーソル
- 位置指定更新および削除操作をサポートしないデータ・ソースの場合、非 NULL 可能な列に対するユニーク索引がニックネームや対応するリモート表に存在しないと、ニックネームによる位置指定 UPDATE および DELETE ステートメントや、一部の検索 UPDATE および DELETE ステートメントは失敗します。こうしたステートメントが失敗した場合は、エラー SQL30090 (理由コード 21) が戻されます。
- ODBC ラッパーは、接続ごとのアクティブなステートメントの数を制限するデータ・ソースに対する INSERT、UPDATE、DELETE ステートメントをサポートしません。データ・ソースが接続ごとのアクティブ・ステートメントの数を制限するかどうかを判断するには、そのデータ・ソースのための文書を調べてください。この制限が当てはまる ODBC データ・ソースの 1 つとして、IBM Red Brick® Warehouse があります。
- ODBC ラッパーは、ドライバー固有の SQL データ・タイプ標識を持つ列を含む表に対する操作をサポートしません。また、パススルー・モードでの CREATE NICKNAME ステートメントおよび SELECT ステートメントもサポートされません。ODBC ラッパーは、Microsoft ODBC プログラマーズ・リファレンス にある ODBC 規格によって定義されている SQL データ・タイプ標識のみをサポートします。

ODBC API を通してアクセスされるデータ・ソースのことを、ここでは ODBC データ・ソースと呼びます。

必要に応じて、Excel ラッパーを使用する代わりに ODBC ラッパーを使用して Excel データにアクセスすることができます。Excel データにアクセスするように ODBC ラッパーを構成するには、ODBC ラッパーを使用した Excel データへのアクセスを参照してください。

推奨: Microsoft SQL Server の場合、ODBC ラッパーを使用するのではなく、Microsoft SQL Server ラッパーを使用して、Microsoft SQL Server データ・ソースにアクセスしなければなりません。Microsoft SQL Server ラッパーでは、Microsoft SQL Server でのより優れた照会パフォーマンスとより優れた機能が提供されます。Microsoft SQL Server ラッパーの構成については、Microsoft SQL Server データ・ソースへのアクセスの構成を参照してください。

手順

1. オペレーティング・システムに応じて、以下の方法のいずれかを使用して、フェデレーテッド・サーバーとフェデレーテッド・データベースを準備します。
 - ODBC (Windows) を介してデータ・ソースにアクセスするためにフェデレーテッド・サーバーを準備します。
 - ODBC (Linux、UNIX) を通してデータ・ソースにアクセスするためにフェデレーテッド・サーバーを準備します。
2. ODBC ラッパーを登録します。
3. ODBC データ・ソースのサーバー定義を登録します。
4. ODBC データ・ソースのユーザー・マッピングを作成します。
5. ODBC データ・ソース・サーバーとの接続をテストします。
6. ODBC データ・ソースの表およびビューのニックネームを登録します。

ODBC を通してデータ・ソースにアクセスするためのフェデレーテッド・サーバーの準備 (Windows)

Windows が稼働するフェデレーテッド・サーバーの場合、フェデレーテッド・サーバーは、ODBC データ・ソースにアクセスする必要があります。フェデレーテッド・サーバーの準備をするには、ODBC System DSN の設定を確認する必要があります。

手順

フェデレーテッド・サーバー上に、ODBC 3.x ドライバーがインストールされていて構成されているか調べてください。

ODBC データ・ソースのノード名が、システム DSN に定義されている必要があります。インストールと構成の手順については、ODBC ドライバーの資料を参照してください。

Microsoft ODBC データ ソース アドミニストレータを使用して DSN を構成した場合は、「コントロール パネル」でこの設定を検査できます。「システム DSN」として ODBC データ・ソースが登録されているか確認してください。そうならないと、フェデレーテッド・サーバーが DSN を見つけられない可能性があります。

次のタスク

この作業が完了したら、ラッパーを登録することができます。

ODBC を通してデータ・ソースにアクセスするためのフェデレーテッド・サーバーの準備 (Linux、UNIX)

Linux または UNIX が稼働するフェデレーテッド・サーバーの場合、フェデレーテッド・サーバーは ODBC データ・ソースにアクセスできる必要があります。フェデレーテッド・サーバーを準備するには、odbc.ini ファイルの設定を確認し、シンボリック・リンクを作成し、ODBC データ・ソースへの接続をテストする必要があります。

始める前に

Linux for System z オペレーティング・システムでは、ODBC ラッパーを使用して、DataDirect Sybase Wire Protocol、Oracle Wire Protocol、Oracle Client ドライバーで Sybase または Oracle データ・ソースに接続する場合、DSN の odbc.ini 構成ファイルで **EnableDescribeParam** パラメーターを 1 に設定しなければなりません。

```
EnableDescribeParam=1
```

EnableDescribeParam パラメーターを 1 に設定しないと、ODBC SERVER オブジェクトを PUSHDOWN モードに設定したときに、非 char および varchar のホスト変数のバインディングに関して、1822N エラー・メッセージがスローされます。

手順

1. フェデレーテッド・サーバーの odbc.ini ファイルが更新されていることを確認します。このファイルがない場合は、任意のテキスト・エディターを使用して作成することができます。odbc.ini ファイルについては、ODBC クライアント・ベンダーの資料を参照してください。
2. ODBC クライアントを構成します。

ODBC クライアントの構成方法については、ODBC クライアントのベンダーからの資料を参照してください。

3. クライアントが DataDirect ODBC か RedBrick である場合、適切なシンボリック・リンクが作成されていることを確認します。以下のシンボリック・リンクで、**ODBC_CLIENT_DIR** は ODBC クライアントがインストールされるディレクトリーです。

フェデレーテッド・サーバーのオペレーティング・システム	ステップ
Linux	以下のシンボリック・リンクを作成します。 <pre>ln -s \$ODBC_CLIENT_DIR/./locale /usr/local/locale</pre> <pre>ln -s \$ODBC_CLIENT_DIR/libodbcinst.so /usr/lib/libodbcinst.so</pre> <p>DataDirect Technologies Connect for ODBC 4.2 ドライバーを使用している場合、次のシンボリック・リンクも作成しなければなりません。</p> <pre>ln -s \$ODBC_CLIENT_DIR/libivicu19.so /usr/lib/libivicu19.so</pre>

フェデレーテッド・サーバーのオペレーティング・システム	ステップ
Solaris	以下のシンボリック・リンクを作成します。 <pre>ln -s \$ODBC_CLIENT_DIR/../../locale \$HOME/sql/lib/locale</pre> \$HOME は、DB2 インスタンス所有者のホーム・ディレクトリーです。

4. クライアントが DataDirect ODBC の場合、フェデレーテッド・サーバーからデータ・ソースへの接続は、DataDirect Connect ODBC demoodbc ツールを使用してテストできます。
 - a. /opt/odbc/odbc.sh スクリプトを実行します。このスクリプトは、いくつかのオペレーティング・システム特有の環境変数をセットアップします。
 - b. DataDirect Connect ODBC demoodbc ツールを使用して、ODBC データ・ソースへの接続をテストします。 demoodbc ツールは、Connect ODBC ライブラリーの /demo サブディレクトリーにあります。

次のタスク

この作業が完了したら、ラッパーを登録することができます。

ODBC ラッパーの登録

ODBC データ・ソースにアクセスするためには、ラッパーを登録する必要があります。フェデレーテッド・サーバーは、ラッパーを使用してデータ・ソースとの通信やデータ・ソースからのデータの取得を行います。ラッパーは、ライブラリー・ファイルのセットとしてインプリメントされます。

手順

次の方法のいずれかを使用します。

方法	説明
CREATE WRAPPER ステートメントをコマンド行から発行し、 ODBC ラッパー のデフォルト名を指定します。	<p>例:</p> <pre>CREATE WRAPPER ODBC;</pre> <p>要確認: このデフォルト名、ODBC を使用してラッパーを登録する場合、フェデレーテッド・サーバーはフェデレーテッド・サーバーが稼働するオペレーティング・システムにとって適切な ODBC ラッパー・ライブラリーを自動的に採用します。</p> <p>Linux または UNIX が稼働するフェデレーテッド・サーバーでは、MODULE ラッパー・オプションを指定する必要があります。MODULE ラッパー・オプションでは、ODBC Driver Manager を含むライブラリーの絶対パスを指定します。</p> <p>AIX および Solaris では、IBM InfoSphere Classic Federation Server for z/OS データ・ソースへのアクセスを構成し、32 ビットの IBM InfoSphere Classic Federation Server for z/OS クライアントを使用する場合、DB2_FENCED および DB2_SOURCE_CLIENT_MODE オプションを、以下の例に示すように指定する必要があります。</p> <pre>CREATE WRAPPER ODBC LIBRARY 'libdb2rcodbc.a' OPTIONS (DB2_FENCED 'Y', DB2_SOURCE_CLIENT_MODE '32BIT', MODULE '/opt/IBM/DB2IIClassic82/cli/lib/cacsqcli.so')</pre> <p>DB2_SOURCE_CLIENT_MODE がサポートされているのは、AIX および Solaris のみです。</p>
CREATE WRAPPER ステートメントをコマンド行から発行し、 ODBC ラッパー の代替名を指定します。	<p>デフォルトのラッパー名がフェデレーテッド・データベース内の既存のラッパー名と競合する場合、デフォルトのラッパー名を、指定の名前で置き換えることができます。デフォルト名とは別の名前を使用する場合は、CREATE WRAPPER ステートメントに LIBRARY パラメーターを含める必要があります。</p> <p>たとえば、AIX を使用するフェデレーテッド・サーバー上で odbc_wrapper という名前のラッパーを登録するには、次のステートメントを発行します。</p> <pre>CREATE WRAPPER odbc_wrapper LIBRARY 'libdb2rcodbc.a';</pre> <p>指定するラッパー・ライブラリー・ファイルは、フェデレーテッド・サーバーのオペレーティング・システムによって異なります。</p>

次のタスク

この作業が完了したら、サーバー定義の登録ができます。

ODBC ラッパー・ライブラリー・ファイル

ODBC ラッパー・ライブラリー・ファイルは、ラッパーのインストール時にフェデレーテッド・サーバーに追加されます。

ODBC ラッパーをインストールすると、デフォルトのディレクトリー・パスに、3つのライブラリー・ファイルが追加されます。たとえば、フェデレーテッド・サーバーが AIX 上で稼働している場合、ディレクトリー・パスに追加されるラッパー・ライブラリー・ファイルは、libdb2rcodbc.a、libdb2rcodbcF.a、libdb2rcodbcU.a です。デフォルトのラッパー・ライブラリー・ファイルは、libdb2rcodbc.a です。他のラッパー・ライブラリー・ファイルは、ODBC ラッパーによって内部で使用されます。

ラッパーを登録する際にデフォルトのラッパー名を使用しない場合は、CREATE WRAPPER ステートメントに LIBRARY パラメーターを含め、デフォルトのラッパー・ライブラリー・ファイル名を指定する必要があります。

以下の表に、デフォルトのディレクトリー・パスとデフォルトのラッパー・ライブラリー・ファイルの名前をリストします。

表 27. ODBC クライアント・ライブラリーの場所とファイル名

オペレーティング・システム	ディレクトリー・パス	ラッパーのライブラリー・ファイル
AIX	/usr/opt/install_path/lib32/ /usr/opt/install_path/lib64/	libdb2rcodbc.a
Linux	/opt/IBM/db2/install_path/lib32 /opt/IBM/db2/install_path/lib64	libdb2rcodbc.so
Solaris	/opt/IBM/db2/install_path/lib32 /opt/IBM/db2/install_path/lib64	libdb2rcodbc.so
Windows	%DB2PATH%\%bin	db2rcodbc.dll

install_path は、UNIX または Linux 上でフェデレーションがインストールされているディレクトリー・パスです。

CREATE WRAPPER ステートメント - ODBC ラッパーの例

ODBC ラッパーを登録するには、CREATE WRAPPER ステートメントを使用します。このトピックでは、Linux、UNIX および Windows に関する例を提供します。

以下の例では、*odbc_wrapper* は、フェデレーテッド・データベースに登録するラッパーに割り当てる名前です。

Linux および Solaris のフェデレーテッド・サーバー

以下の例では、Linux または Solaris が稼働するフェデレーテッド・サーバーにラッパーをデフォルト名で登録する方法を示します。

```
CREATE WRAPPER odbc OPTIONS (MODULE '/opt/lib/odbc.so');
```

Linux または UNIX が稼働するフェデレーテッド・サーバーでは、MODULE ラッパー・オプションを指定する必要があります。MODULE ラッパー・オプションでは、ODBC Driver Manager を含むライブラリーの絶対パスを指定します。

以下の例では、Linux または Solaris が稼働するフェデレーテッド・サーバーで代替名を使用してラッパーを登録する方法を示します。

```
CREATE WRAPPER odbc_wrapper LIBRARY 'libdb2rcodbc.so'  
OPTIONS (MODULE '/opt/lib/odbc.so');
```

AIX フェデレーテッド・サーバー

以下の例では、AIX が稼働するフェデレーテッド・サーバーでデフォルト名を使用してラッパーを登録する方法を示します。

```
CREATE WRAPPER odbc
  OPTIONS (MODULE '/usr/lib/odbc.a');
```

UNIX が稼働するフェデレーテッド・サーバーでは、MODULE ラッパー・オプションを指定する必要があります。MODULE ラッパー・オプションでは、ODBC Driver Manager を含むライブラリーの絶対パスを指定します。

以下の例では、AIX が稼働するフェデレーテッド・サーバーで代替名を使用してラッパーを登録する方法を示します。

```
CREATE WRAPPER odbc_wrapper LIBRARY 'libdb2rcodbc.a'
  OPTIONS (MODULE '/usr/lib/odbc.a');
```

IBM InfoSphere Classic Federation Server for z/OS (AIX)

以下に示すのは、AIX オペレーティング・システムで CREATE WRAPPER ステートメントを発行して ODBC ラッパーを登録する方法の例です。64 ビット AIX および Solaris では、IBM InfoSphere Classic Federation Server for z/OS クライアントを使用する場合、DB2_FENCED オプションと DB2_SOURCE_CLIENT_MODE オプションは、以下の例で示すように指定する必要があります。

```
CREATE WRAPPER odbc
  OPTIONS (DB2_FENCED 'Y', DB2_SOURCE_CLIENT_MODE '32BIT',
  MODULE '/opt/IBM/DB2IIClassic82/cli/lib/cacsqlcli.so');
```

DB2_SOURCE_CLIENT_MODE がサポートされているのは、AIX および Solaris のみです。

UNIX が稼働するフェデレーテッド・サーバーでは、MODULE ラッパー・オプションを指定する必要があります。MODULE ラッパー・オプションでは、ODBC Driver Manager を含むライブラリーの絶対パスを指定します。

以下の例では、代替名を使用して IBM InfoSphere Classic Federation Server for z/OS データ・ソースにアクセスするためのラッパーを登録する方法を示しています。

```
CREATE WRAPPER odbc_wrapper LIBRARY 'libdb2rcodbc.a'
  OPTIONS (DB2_FENCED 'Y', DB2_SOURCE_CLIENT_MODE '32BIT',
  MODULE '/opt/IBM/DB2IIClassic82/cli/lib/cacsqlcli.so');
```

Windows フェデレーテッド・サーバー

以下の例では、Windows が稼働するフェデレーテッド・サーバーでデフォルト名を使用してラッパーを登録する方法を示します。

```
CREATE WRAPPER odbc;
```

以下の例では、Windows が稼働するフェデレーテッド・サーバーで代替名を使用してラッパーを登録する方法を示します。

```
CREATE WRAPPER odbc_wrapper LIBRARY 'db2rcodbc.dll';
```

ODBC データ・ソースのサーバー定義の登録

フェデレーテッド・データベース内に、アクセスする ODBC サーバーをそれぞれ登録する必要があります。

手順

1. コマンド行から CREATE SERVER ステートメントを発行し、必須サーバー・オプションを指定します。

例:

```
CREATE SERVER server_definition_name TYPE data_source_type
  VERSION version_number WRAPPER wrapper_name
  OPTIONS (NODE 'node_name', CODEPAGE 'codepage_name', DBNAME 'database_name');
```

NODE、CODEPAGE、および DBNAME は CREATE SERVER ステートメントのサーバー・オプションですが、ODBC データ・ソースで必須です。

- NODE サーバー・オプションは、cac.ini ファイルの DATASOURCE キーワードで指定されたデータ・ソースに設定しなければなりません。

例 cac.ini ファイルで DATASOURCE = CACSAMP tcp/150.45.37.49/5000 が指定されている場合、NODE オプションは CACSAMP に設定しなければなりません。

- CODEPAGE サーバー・オプションは、cac.ini ファイルで指定されたクライアント・コード・ページのコード・ページ番号に設定しなければなりません。クライアント・コード・ページは、データ・ソースのコード・ページです。

例 cac.ini ファイルで CLIENT CODEPAGE = IBM-850 が指定されている場合、CODEPAGE オプションは 850 に設定しなければなりません。

- DBNAME サーバー・オプションは、アクセスするデータ・ソース・データベースの名前に設定しなければなりません。

例 ODBC データ・ソース名が venice の場合、DBNAME オプションは venice に設定しなければなりません。

2. サーバー定義を作成した後、サーバー・オプションを追加またはドロップするには、ALTER SERVER ステートメントを使用します。

次のタスク

この作業が完了したら、ユーザー・マッピングを作成できます。

CREATE SERVER ステートメント - ODBC ラッパーの例

ODBC ラッパーにサーバー定義を登録するには、CREATE SERVER ステートメントを使用します。このトピックでは、必須パラメーターを使用する完全指定の例と、追加のサーバー・オプションを使用する例を提供します。

次に、CREATE SERVER ステートメントを実行して MySQL データ・ソースのサーバー定義を登録する方法の例を示します。

```
CREATE SERVER mysql_server TYPE mysql
  VERSION 4.0 WRAPPER wrapper_name
  OPTIONS (NODE 'odbc_node', DBNAME 'venice');
```

mysql_server

ODBC データ・ソース・サーバーに割り当てる名前。重複するサーバー定義名は使用できません。

TYPE *mysql*

アクセスを構成するデータ・ソース・サーバーのタイプを指定します。このパラメーターはオプションです。

VERSION *4.0*

アクセスする ODBC データ・ソースのバージョン。このパラメーターはオプションです。

WRAPPER *wrapper_name*

CREATE WRAPPER ステートメントで指定したラッパー名。

NODE *'odbc_node'*

DSN を定義した時に ODBC データ・ソースに割り当てたノード名 (システム DSN 名)。Windows が稼働するフェデレーテッド・サーバーの場合、この値は、「ODBC データ ソース アドミニストレータ (ODBC Data Source Administrator)」ウィンドウにあるシステム DSN の名前である必要があります。UNIX が稼働するフェデレーテッド・サーバーの場合、ノードの名前は ODBC 構成ファイルで定義されている DSN になります。ODBC 構成ファイルの名前は通常 *odbc.ini* です。

ノード名は CREATE SERVER ステートメントではオプションとなっていますが、ODBC データ・ソースの場合は必要です。

DBNAME *'venice'*

オプション。アクセスする ODBC データ・ソースの名前。

サーバー・オプション

サーバー定義の作成時に、CREATE SERVER ステートメントに追加のサーバー・オプションを指定することができます。サーバー・オプションには、一般的なサーバー・オプションと ODBC に特有のサーバー・オプションを指定できます。

一部の ODBC データ・ソース (例: MySQL) は、SQL ステートメントで表名や列名を囲んでいる引用符を処理できません。これらのデータ・ソースにアクセスするには、CREATE SERVER ステートメントに以下のサーバー・オプションを含める必要があります。

- DB2_TABLE_QUOTE_CHAR '``'
- DB2_ID_QUOTE_CHAR '``'
- DB2_AUTHID_QUOTE_CHAR '``'

`` 文字はスキーマ名、表名、列名などの ID の区切り文字です。

例:

```
CREATE SERVER mysql_server TYPE mysql
  VERSION 4.0 WRAPPER wrapper_name
  OPTIONS (NODE 'mysql_node', DB2_TABLE_QUOTE_CHAR '``',
  DB2_ID_QUOTE_CHAR '``', DB2_AUTHID_QUOTE_CHAR '``')
```

ODBC データ・ソースのユーザー・マッピングの作成

ODBC サーバーにアクセスする場合、フェデレーテッド・サーバーは、データ・ソースに対して有効なユーザー ID とパスワードを使用して、ODBC サーバーへの接続を確立します。ユーザー・マッピングを必要とするデータ・ソースの場合、それぞれのフェデレーテッド・サーバーのユーザー ID とパスワード、および対応するデータ・ソースのユーザー ID とパスワードの間の関連付け (ユーザー・マッピング) を定義しなければなりません。

このタスクについて

フェデレーテッド・システムにアクセスして ODBC データ・ソースへの分散要求を送信するユーザー ID ごとに、ユーザー・マッピングを作成します。

手順

CREATE USER MAPPING ステートメントを発行します。
たとえば、次のようにします。

```
CREATE USER MAPPING FOR local_userID SERVER server_definition_name  
    OPTIONS (REMOTE_AUTHID 'remote_userID', REMOTE_PASSWORD 'remote_password');
```

REMOTE_AUTHID および REMOTE_PASSWORD は CREATE USER MAPPING ステートメントではユーザー・マッピング・オプションですが、ODBC データ・ソースにアクセスするには必須です。

ユーザー・マッピングでは、cac.ini ファイルで指定されているユーザー ID およびパスワードに DB2 許可 ID をマップしなければなりません。

例 cac.ini ファイルで USERID = *MY_USERID* および USERPASSWORD = *MY_PASSWORD* が指定されている場合、CREATE USER MAPPING ステートメントのオプションを以下のように指定しなければなりません。

```
REMOTE_AUTHID = MY_USERID  
REMOTE_PASSWORD = MY_PASSWORD
```

次のタスク

この作業が完了したら、ODBC データ・ソースへの接続をテストできます。

CREATE USER MAPPING ステートメント - ODBC ラッパーの例

CREATE USER MAPPING ステートメントを使用して、フェデレーテッド・サーバーのユーザー ID を ODBC データ・ソースのユーザー ID およびパスワードにマップしてください。このトピックには、必須パラメーターを使用する完全指定の例と、CREATE USER MAPPING ステートメントで DB2 特殊レジスター USER を使用する方法を示す例が含まれています。

以下に示すのは、フェデレーテッド許可 ID を ODBC データ・ソースのユーザー ID とパスワードにマップする方法の例です。

```
CREATE USER MAPPING FOR arturo SERVER mysql_server  
    OPTIONS (REMOTE_AUTHID 'art', REMOTE_PASSWORD 'red4blue')
```

arturo ODBC データ・ソースで定義されたリモート・ユーザー ID およびパスワードにマップするローカル許可 ID を指定します。

mysql_server

CREATE SERVER ステートメントで定義した ODBC データ・ソースのサーバー定義名を指定します。

'art' *arturo* をマップするリモート・ユーザー ID を指定します。CREATE SERVER ステートメントで FOLD_ID サーバー・オプションを「U」または「L」に設定している場合を除き、この値は大文字小文字を区別します。

'red4blue'

'art' に関連するリモート・パスワードを指定します。CREATE SERVER ステートメントで FOLD_PW サーバー・オプションを「U」または「L」に設定している場合を除き、この値は大文字小文字を区別します。

DB2 特殊レジスター USER

DB2 特殊レジスター USER を使用して、CREATE USER MAPPING ステートメントを実行しているユーザーの許可 ID を、REMOTE_AUTHID ユーザー・オプションに指定されているデータ・ソースの許可 ID にマップすることができます。

以下に、特殊レジスター USER を含む CREATE USER MAPPING ステートメントの例を示します。

```
CREATE USER MAPPING FOR USER SERVER mysql_server
  OPTIONS (REMOTE_AUTHID 'art', REMOTE_PASSWORD 'red4blue');
```

ODBC データ・ソース・サーバーへの接続のテスト

フェデレーテッド・サーバーが ODBC データ・ソースへのアクセスのために適切に構成されているかを確認するために、ODBC データ・ソース・サーバーへの接続をテストします。

このタスクについて

ODBC データ・ソース・サーバーへの接続のテストは、定義したサーバー定義とユーザー・マッピングを使用して行うことができます。

手順

パススルー・セッションを開き、ODBC データ・ソース・システム表に対して SELECT ステートメントを発行します。SELECT ステートメントがカウントを戻す場合、サーバー定義およびユーザー・マッピングは適切にセットアップされています。

```
SET PASSTHRU server_definition_name
SELECT count(*) FROM schema_name.table_name
SET PASSTHRU RESET
```

SELECT ステートメントがエラーを戻す場合には、接続エラーのトラブルシューティングをします。

次のタスク

この作業が完了したら、ODBC データ・ソース表およびビューのニックネームを登録できます。

データ・ソース接続エラーのトラブルシューティング

データ・ソース・サーバーへのテスト接続では、いくつかの理由でエラーが戻されることがあります。以下の処置により、エラーが発生した理由を判別することができます。

症状

データ・ソースに接続しようとしたときにエラーが戻されます。

原因

接続の問題には、いくつかの原因が考えられます。

問題の解決方法

データ・ソース接続エラーのトラブルシューティングを行うには、以下の項目で問題がないかどうかをチェックします。

- データ・ソースが使用可能であることを確認します。
- 該当する場合は、データ・ソース・サーバーが着信接続に対して構成されていることを確認します。
- REMOTE_AUTHID と REMOTE_PASSWORD のオプションに関するユーザー・マッピング設定を調べ、データ・ソースとの接続に有効な設定になっているか確認します。必要に応じて、ユーザー・マッピングを変更するか、別のユーザー・マッピングを作成します。
- 該当する場合は、フェデレーテッド・サーバー上のデータ・ソース・クライアント・ソフトウェアがインストールされていて、データ・ソースに接続できるように正しく構成されていることを確認します。
- ODBC データ・ソースの場合、フェデレーテッド・サーバー上の ODBC ドライバーがインストールされていて、ODBC データ・ソース・サーバーに接続できるように正しく構成されていることを確認します。Windows が稼働するフェデレーテッド・サーバーでは、ドライバーをチェックするときに「ODBC データ ソース アドミニストレータ」ツールを使用します。UNIX が稼働するフェデレーテッド・サーバーの場合は、ODBC クライアントのベンダーが提供している資料を参照してください。
- フェデレーテッド・サーバー上で設定されている変数の設定がデータ・ソースについて適切であることを確認します。これらの変数には、システム環境変数、db2dj.ini ファイル、および DB2 Profile Registry (db2set) 変数が含まれます。
- サーバー定義を調べます。必要であれば、サーバー定義をドロップして再作成します。

ODBC データ・ソース表およびビューのニックネームの登録

登録する各 ODBC サーバー定義で、アクセスする表またはビューごとにニックネームを登録する必要があります。ODBC データ・ソースを照会するときには、データ・ソース・オブジェクトの名前ではなく、これらのニックネームを使用します。

始める前に

ニックネームを登録する前に、ODBC データ・ソース側の統計を更新します。フェデレーテッド・データベースは、データ・ソースのカatalog統計に基づいて、照会処理を最適化します。DB2 **RUNSTATS** コマンドに相当するデータ・ソース・コマンドを使用して、データ・ソースの統計を更新します。

手順

コマンド行から CREATE NICKNAME ステートメントを発行します。

例:

```
CREATE NICKNAME nickname  
FOR server_definition_name."remote_schema"."remote.table" ;
```

ニックネームの長さは 128 文字までです。

ニックネームを作成すると、フェデレーテッド・サーバーはニックネームを使用してデータ・ソース・カatalogを照会します。この照会は、データベース・ソースの表またはビューへの接続をテストします。接続ができない場合、エラー・メッセージが出ます。

次のタスク

ニックネームを作成する ODBC の表またはビューごとに、このステップを繰り返します。

CREATE NICKNAME ステートメント - ODBC ラッパーの例

CREATE NICKNAME ステートメントを発行して、アクセスする ODBC 表またはビューのニックネームを登録します。必須パラメーターを使用した完全指定の例がこのトピックには含まれています。

以下の例では、CREATE NICKNAME ステートメントを使用して、ODBC 表またはビューのニックネームを登録する方法を示します。

```
CREATE NICKNAME cust_europe FOR mysql_server."vinnie"."italy"
```

cust_europe

ODBC の表またはビューを識別するために使用する固有なニックネームです。ニックネームはスキーマ内で固有でなければなりません。

重要: ニックネームは、2 つの部分 (スキーマとニックネーム) からなる名前です。ニックネームの登録時にスキーマを省略すると、そのニックネームのスキーマはニックネームを登録するユーザーの許可 ID になります。

ODBC データ・ソースがスキーマをサポートしていない場合、CREATE NICKNAME ステートメントからスキーマを省略してください。

```
mysql_server."vinnie"."italy"
```

リモート・オブジェクトを表す、3 つの部分からなる ID。

- *mysql_server* は、ユーザーが CREATE SERVER ステートメントで ODBC データ・ソース・サーバーに割り当てたサーバー定義名です。
- *vinnie* は、表またはビューが属する所有者のユーザー ID です。
- *italy* は、アクセスするリモートの表またはビューの名前です。

フェデレーテッド・サーバーは、名前を引用符で囲んでいない限り、ODBC スキーマおよび表の名前を大文字に変換します。

ODBC チューニング・ユーティリティ (db2fedsvrcfg) による ODBC ラッパーのパフォーマンスの最適化

ODBC ラッパーのパフォーマンスは、ODBC チューニング・ユーティリティ db2fedsvrcfg によって最適化できます。このユーティリティでは、データ・ソースに対して定義済み照会のセットを実行し、結果の正確さをテストします。ユーティリティは ALTER SERVER ステートメントのセットを作成します。このセットをサーバーに対して実行して、最適なパフォーマンスが得られるサーバー・オプションを設定できます。

このタスクについて

フェデレーテッド・サーバーで以下の項目の構成が必要になります。

- フェデレーション
- ODBC ラッパー
- ODBC クライアント・ソフトウェア
- システム環境の変数。ODBCINI と LIBPATH 変数が必要な場合があります。

インストール、構成、ODBC 要件については、個々の製品に添付されている資料を参照してください。

手順

1. ODBC サーバーが定義されていない場合は、DB2 Control Center または DB2 コマンド行プロセッサを使用してフェデレーテッド・サーバーに接続し、ODBC ラッパーおよびサーバーを作成します。
2. オプション: テスト表が ODBC データ・ソースにすでに存在する場合は、ODBC データ・ソースに接続してドロップします。
3. ODBC チューニング・ユーティリティに必要なオプションを指定して DB2 コマンド行から実行します。ユーティリティが完了するにはしばらく時間がかかります。
4. ユーティリティが正常に実行されたことを確認します。ユーティリティが正常に実行された場合は、コマンド・ウィンドウまたは指定の出力ファイルで以下のメッセージを見ることができます。

```
ALTER SERVER "DS1" OPTIONS (ADD option1, 'value1')
ALTER SERVER "DS1" OPTIONS (ADD option2, 'value2')
ALTER SERVER "DS1" OPTIONS (ADD option3, 'value3')
.....
```

db2fedsvrcfg コマンドが正常に終了しました。

コマンドが正常に実行されなかった場合には、エラーの理由を知らせるエラー・メッセージが表示されます。問題を訂正し、コマンドを再度実行してください。

以下の場合には、テスト表を手動で作成する必要があります。

- デフォルトとは異なる表名を使用する場合

- ODBC 経由でアクセスするデータ・ソースが読み取り専用の場合
 - ODBC チューニング・ユーティリティーが、ユーティリティーに必要とされるテスト表を作成できない場合
5. データ・ソースが定義されたフェデレーテッド・サーバーに接続します。
 6. ODBC チューニング・ユーティリティーによって作成されたファイルを実行する前に、db2look ユーティリティーを使用して、既存のサーバー設定を保管します。既存のサーバー設定の保管に関しては、db2look ユーティリティーの資料を参照してください。
 7. オプション: ODBC サーバーが定義されている場合、フェデレーテッド・サーバーに接続して、サーバー・オプションをドロップすることができます。ユーティリティーは、ステップ 4 に記述されているフォーマットの ALTER SERVER ステートメントを作成します。これらのサーバー・オプションがすでに追加されている場合は、ALTER SERVER ステートメントが正常に実行されなくなります。
 8. 以下のコマンドを使用して、ユーティリティーによって生成された ALTER SERVER ステートメントを、フェデレーテッド・データベースに対して実行します。ODBC チューニング・ユーティリティーによって作成された ALTER SERVER ステートメントは、db2fedsvrcfg.sql ファイルに含まれます。
db2 -tvf db2fedsvrcfg.sql
 9. ALTER SERVER ステートメントの結果を確認します。いずれかのステートメントが失敗した場合は、db2fedsvrcfg.sql ファイルのステートメントを修正して再度実行してみるということを、正常に実行されるようになるまで繰り返します。
 10. ODBC チューニング・ユーティリティーによるサーバーのチューニングが完了したら、PUSHDOWN サーバー・オプションを 'Y' に設定し、最適化プロセスを完了します。

db2fedsvrcfg コマンド構文 - ODBC チューニング・ユーティリティー

ODBC ラッパーのパフォーマンスを向上させるには、**db2fedsvrcfg** コマンドを使用します。

構文

```
db2fedsvrcfg-s serverName [-m odbcDriverManagerLibrary] -dsn odbcDSNname
[-dbname dsDBname] [-u userid] [-p password] [-noprep] [-prefix tableNamePrefix]
[-suffix tableNameSuffix] [-dscp codePage] [-v] [-o outputFile] [-h]
```

パラメーター

AIX または Solaris で 32 ビットの ODBC ドライバーを使用する場合は、コマンド **db2fedsvrcfg32** を使用します。それ以外の場合は、コマンド **db2fedsvrcfg** を使用します。

```
-dbname dsDBname
    データ・ソースのデータベース名
```

- dscp codePage**
データ・ソースのコード・ページ ID。このオプションが指定されていないと、ユーティリティーはユーザーの環境のコード・ページを使用します。このパラメーターはオプションです。
- dsn odbcDSNname**
データ・ソースのシステム・データ・ソース名 (DSN)
- h** 詳細ヘルプが表示されます。このパラメーターはオプションです。
- m odbcDriverManagerLibrary**
ODBC ドライバー・マネージャー・ライブラリーの完全修飾ファイル名。
Windowsの場合、ODBC ドライバー・マネージャー・ライブラリーのファイル名はオプションです。
- noprep**
テスト前にテスト表がデータ・ソースで作成されないようにする。このパラメーターはオプションです。
- o outputFile**
ODBC チューニング・ユーティリティー出力ファイルの完全修飾ファイル名。
出力ファイルには、ODBC ラッパー・パフォーマンスのチューニングに使用される ALTER SERVER ステートメントが含まれます。このパラメーターはオプションです。このパラメーターが指定されていないと、出力はコマンド・ウィンドウに表示されます。
- p password**
データ・ソースへの接続のためのパスワード。このパラメーターはオプションです。
- prefix tableNamePrefix**
ユーティリティーが分析に使用する ODBC データ・ソース表名の接頭部。接頭部が指定されていないと、デフォルトの接頭部である IITEST が使用されます。このパラメーターはオプションです。
- s serverName**
フェデレーテッド・サーバーの名前。
- suffix tableNameSuffix**
ユーティリティーが分析に使用する ODBC データ・ソース表名の接尾部。接尾部が指定されなければ、空ストリングが使用されます。
- u userid**
データ・ソースへの接続のためのユーザー名。このパラメーターはオプションです。
- v** ユーティリティーの出力が詳細になるよう指定します。このパラメーターはオプションです。

例

以下は、データ・ストア、つまり ODBC データ・ソースに対して実行されるコマンドの例です。この例では、テスト表の名前は、ABC n XYZ となっています。 n には 1 から 7 の数字が入ります。

```
db2fedsvr cfg -s DS1 -m "/usr/lib/odbc.a"
-dsn datastore -dbname db1 -u authid -p password -noprep
-prefix ABC -suffix XYZ -o "/home/user1/db2fedsvr cfg.sql"
```

ODBC チューニング・ユーティリティー (db2fedsvr cfg) のためのテスト表定義

場合によっては、ODBC チューニング・ユーティリティーのためのテスト表を手動で作成する必要があります。このトピックでは、テスト表定義について説明します。

以下の場合には、ODBC チューニング・ユーティリティーのためのテスト表をユーザーが作成する必要があります。

- デフォルトとは異なる表名を使用する場合
- ODBC 経由でアクセスするデータ・ソースが読み取り専用の場合
- ODBC チューニング・ユーティリティーが、ユーティリティーに必要とされるテスト表を作成できない場合

必要となる 7 つのテスト表 (IITEST1 から IITEST7) すべてに、以下の表定義が適用されます。デフォルトの表名の接頭部は IITEST で、デフォルトの接尾部は空ストリングです。それ以外の接頭部や接尾部を指定する場合は、ODBC チューニング・ユーティリティーの実行時に `-prefix` および `-suffix` オプションを指定する必要があります。

表 28. 表 IITEST1 のための ODBC チューニング・ユーティリティー・テスト表定義

列名	SQL データ・タイプ	SQL データ・タイプ ID	長さ
IT1C1	integer	SQL_INTEGER	
IT1C2	integer	SQL_INTEGER	
IT1C3	char(1)	SQL_CHAR	1
IT1C4	char(3)	SQL_CHAR	3
IT1C5	char(10)	SQL_CHAR	10
IT1C6	varchar(10)	SQL_VARCHAR	10
IT1C7	char(100)	SQL_CHAR	100

表 29. 表 IITEST2 のための ODBC チューニング・ユーティリティー・テスト表定義

列名	SQL データ・タイプ	SQL データ・タイプ ID	長さ
IT2C1	integer	SQL_INTEGER	
IT2C2	integer	SQL_INTEGER	
IT2C3	char(30)	SQL_CHAR	30

表 30. 表 IITEST3 から IITEST7 のための ODBC チューニング・ユーティリティー・テスト表定義

各表の列数	列名	SQL データ・タイプ	SQL データ・タイプ ID	長さ
66	ITxCn	char(100)	SQL_CHAR	100

x 定義される表に対応する番号

n 列番号

たとえば、IT3C1 は、表 ITEST3 の最初の列の列名です。

ODBC ラッパーを使用した Excel データへのアクセス

Excel ODBC ドライバーを使用すると、ODBC ラッパーで Microsoft Excel ワークブックにアクセスできます。

始める前に

- Excel ODBC ドライバーがフェデレーテッド・サーバー上になければなりません。
- フェデレーテッド・サーバーは Excel ワークブック中のワークシートを開き読み取って、データを取り出せなければなりません。したがって、Excel ワークブックはフェデレーテッド・サーバーと同じコンピューター上か、またはアクセス可能なマップされたネットワーク・ドライブ上になければなりません。
- 予期される列のデータ・タイプに応じて、列のフォーマットを設定します。
- 列に挿入されるデータは、その列に指定されたフォーマット・タイプに適合するものでなければなりません。
- スプレッドシートの最初の 8 行にデータがない場合は、空であるか確認します。セルを確実に空にするには、Microsoft Excel でスプレッドシートを開いて、「編集」 > 「すべてクリア」を選択します。
- スプレッドシートの列に挿入されるデータは、必ず指定されたタイプに適合するものにします。

このタスクについて

Excel アプリケーションをフェデレーテッド・サーバー上にインストールする必要はありません。Excel ODBC ドライバーは、自動的に Windows と共にインストールされます。

ODBC ラッパーと Excel ODBC ドライバーを使用すると、ワークブック中のすべてのワークシート中のデータにアクセスできます。Excel ODBC ドライバーは、ワークブックをデータベースと解釈し、ワークブック中の個々のワークシートを表と解釈します。

ワークブックを作成したバージョンの Excel アプリケーションがサポートされなくなっても、Excel ODBC ドライバーは旧バージョンの Excel ワークブックをサポートしています。たとえば、Microsoft 社は Excel バージョン 4.0 で作成されたワークシートをサポートしなくなりましたが、このドライバーはこのバージョンで作成した Excel ワークシートをサポートしています。

制約事項

- ワークブックがユーザーまたはアプリケーションによって読み取り/書き込みモードですでに開かれている場合、ODBC ラッパーはワークシートにアクセスできません。しかし、ユーザーまたはアプリケーションがワークブックを開く前に ODBC ラッパーがワークブックを開いた場合は、ユーザーまたはアプリケーションは読み取り専用モードでワークブックを開くことができます。
- Excel ODBC ドライバーは、最初の非ブランク行にワークシート列のラベルが含まれていることを予期しています。ワークシートにラベルがない場合は、ワークシート中に列ラベルの行を挿入しなければなりません。

- Excel ODBC ドライバーは Windows オペレーティング・システムでだけ入手可能なので、ODBC ラッパーを使用してアクセスできるのは、Windows が稼働しているフェデレーテッド・サーバー上の Excel データだけです。
- Excel ワークシート上で挿入操作と更新操作を実行できますが、削除操作は実行できません。Excel ODBC ドライバーは、削除操作をサポートしていません。ワークシートからデータを削除するには、Excel でワークシートを開いて変更を加えなければなりません。

手順

1. アクセスする Excel ワークブックが、フェデレーテッド・サーバー上にあるか、アクセス可能なマップされたネットワーク・ドライブにあるか確認します。
2. ワークグループを使用する、移行期の Windows ネットワークを介して Excel データを共有する場合、Excel データ・ソースに対するアクセス許可を構成しなければなりません。
3. 必要に応じて、Excel ワークシート中のデータのレイアウトに変更を加えて、Excel ODBC ドライバーの要件を満たすようにします。アクセスするワークシートまたは名前付き範囲ごとに、このステップを繰り返します。
4. 必要に応じて、アクセスする名前付き範囲を作成します。
5. アクセスするワークブックのシステム DSN を作成します。システム DSN の構成には、「ODBC データ ソース アドミニストレータ」を使用できます。システム DSN の作成時に指定した名前が、CREATE SERVER ステートメント中の NODE オプションの値として割り当てられます。

ワークグループを使用する Windows ネットワークを介して Excel データ・ソースを共有する場合、以下の構文を使用して、システム DSN のデータベース名を指定しなければなりません。

```
¥¥computer_Name¥filename_Subdirectory
```

ここで、*computer_Name* は Excel データ・ソースのコンピューター名、*filename_Subdirectory* は Excel ファイルのサブディレクトリーおよびファイル名です。

例 Excel データ・ソースのコンピューター名が XLSQLS で、Excel ファイルへのネットワーク・ディレクトリーが E:¥share¥test.xls の場合、以下の DSN データベース名を指定します。

```
¥¥XLSQLS¥share¥test.xls
```

ここで、ネットワーク・ディレクトリー E: のルート・ディレクトリーが ¥¥ およびコンピューター名 XLSQLS に置き換えられています。

6. CREATE WRAPPER ステートメントを発行します。
7. フェデレーテッド・データベース・システム・カタログ中にサーバー・オブジェクトを登録して、ワークブックの場所を指定します。ODBC ラッパーの場合には、DSN ごとにサーバー・オブジェクトが必要です。Excel ODBC ドライバーの使用時に DSN がワークブックと関連付けられます。NODE *compounds_workbook_dsn* は、作成したシステム DSN です。ODBC ラッパーが Excel ワークシートにアクセスするには、NODE オプションが必要です。

ワークブックの場所を指定するには、CREATE SERVER ステートメントを発行し、NODE オプションのシステム DSN として DSN を使用します。

たとえば、次のようにします。

```
CREATE SERVER compounds_workbook WRAPPER odbc  
  OPTIONS (NODE 'compounds_workbook_dsn', PASSWORD 'n')
```

アクセスしようとしているワークブックごとに、このステップを繰り返します。

8. CREATE NICKNAME ステートメントを発行して、アクセスするワークシートのニックネームを作成します。構文は次のとおりです。

```
CREATE NICKNAME nickname FOR server_name.remote_table
```

9. 名前付き範囲を作成してデータにアクセスする場合は、CREATE NICKNAME ステートメントの *remote_table* の部分に範囲の名前を指定します。

たとえば、範囲の名前が *testing* の場合は、CREATE NICKNAME ステートメントは以下のようになります。

```
CREATE NICKNAME compounds_nickname FOR compounds_workbook.testing
```

範囲ではなくワークシート全体のデータにアクセスするには、ワークシートの名前の後に \$ 記号を指定します。

たとえば、ワークシートの名前が *Sheet1* の場合は、CREATE NICKNAME ステートメントは以下のようになります。

```
CREATE NICKNAME compounds_nick FOR compounds_workbook.Sheet1$
```

ODBC ラッパーを使用する場合のワークグループ内の Excel データへのアクセス許可の構成

ワークグループを使用する Windows ネットワークを介して共有されるリモート Excel データへアクセスするために、アクセス許可を構成できます。

このタスクについて

この作業の例はすべて、Windows XP Professional でのものです。ユーザー・アカウント許可の設定方法に関する特定の情報については、データ・ソースが置かれているオペレーティング・システムの資料を参照してください。

手順

1. 「ネットワーク アクセス：ローカル アカウントの共有とセキュリティ モデル」パネルで共有およびセキュリティー・モデルを設定します。以下の共有およびセキュリティー・モデルのいずれかを選択できます。

Guest のみ - ローカル ユーザーが Guest として認証する

Guest ユーザー・アカウントのみでデータ・ソースに接続します。Guest ユーザー・アカウント用に構成し、DB2 ユーザー・パスワードを使用しなければならない、アクセス・レベルを受け取ります。

クラシック - ローカル ユーザーがローカル ユーザーとして認証する

DB2 ユーザー ID およびパスワードを使用して、データ・ソースに接続できます。または、Guest ユーザー・アカウントを使用することもできます。DB2 ユーザー ID の場合、DB2 ユーザー ID のアクセス・レベルに従って、データ・ソースへのアクセス・レベルを受け取ります。

Guest ユーザー・アカウントの場合、Guest ユーザー・アカウント用に構成し、DB2 ユーザー・パスワードを使用しなければならない、アクセス・レベルを受け取ります。

例 「クラシック - ローカル ユーザーがローカル ユーザーとして認証する」共用およびセキュリティー・モデルを選択するには、「管理ツール」を開き、「ローカル セキュリティ ポリシー」 > 「ローカル ポリシー」 > 「セキュリティ オプション」 > 「ネットワーク アクセス：ローカル アカウントの共有とセキュリティ モデル」までナビゲートします。

2. 「ネットワーク経由でコンピュータへアクセス」プロパティ・パネルでユーザー・アカウントを作成します。

オプション	説明
「Guest のみ - ローカル ユーザーが Guest として認証する」セキュリティー・モデルの場合	Windows ワークグループを指定し、Guest ユーザー・アカウントを作成します。パスワードは、DB2 パスワードを使用して指定しなければなりません。
「クラシック - ローカル ユーザーがローカル ユーザーとして認証する」セキュリティー・モデルの場合	<ul style="list-style-type: none"> Windows ワークグループを指定し、DB2 ユーザー ID およびパスワードを使用してユーザー・アカウントを作成します。 Guest ユーザー・アカウントを使用する場合、Windows ワークグループを指定し、Guest ユーザー・アカウントを作成します。Guest ユーザー・アカウントのパスワードは、DB2 パスワードを使用して指定しなければなりません。

例 Guest ユーザー・アカウントを作成するには、以下のようにします。

- 「管理ツール」を開き、「ローカル セキュリティ ポリシー」 > 「ローカル ポリシー」 > 「ユーザー権利の割り当て」 > 「ネットワーク経由でコンピュータへアクセス」までナビゲートします。
- 「場所を指定してください」フィールドで Windows ワークグループを指定します。
- 「選択するオブジェクト名を入力してください」フィールドで DB2 ユーザー ID を指定します。
- 「名前の確認」をクリックします。
- 「ネットワーク パスワードの入力」パネルで DB2 ユーザー ID およびパスワードを指定します。

3. 「変更」および「読み取り」許可を「許可」に設定することにより、「共用のプロパティ」パネルで共用許可を設定します。

例 E:\\$share%test.xls ネットワーク・ディレクトリーの共用許可を設定するには、以下を行います。

- E:\\$share ディレクトリー・フォルダーを右クリックし、「共有とセキュリティ」をクリックします。
- 「共有」タブで「このフォルダを共有する」を選択し、「アクセス許可」をクリックします。

- ユーザーを選択した後に、「変更」許可と「読み取り」許可の両方で「許可」を選択します。
4. 「読み取りと実行」許可と「読み取り」許可を「許可」に設定することにより、「プロパティ」パネルで NTFS 許可を設定します。

例 E:¥share¥test.xls ファイルの NTFS 許可を設定するには、以下を行います。

- test.xls ファイルを右クリックして、「プロパティ」をクリックする。
- 「セキュリティ」タブで、ユーザーを選択した後に、「読み取りと実行」許可と「読み取り」許可の両方で「許可」を選択します。

次のタスク

ODBC ラッパーを使用して Excel データにアクセスする際の手順を完了します。

IBM InfoSphere Classic Federation Server for z/OS データ・ソースへの ODBC アクセスの構成

ODBC ラッパーは、IBM InfoSphere Classic Federation Server for z/OS データ・ソースへのアクセスのために最適化されています。ODBC ラッパーは、ODBC ドライバーを検出し、自動的にパフォーマンス・オプションを構成します。

始める前に

フェデレーテッド・サーバーで以下の項目の構成が必要になります。

- フェデレーテッド・サーバー
- IBM InfoSphere Classic Federation Server for z/OS クライアント
- フェデレーテッド・データベース
- DB2 プロファイル・レジストリー (db2set) のシステム環境 db2dj.ini ファイルの変数。
 - Linux および UNIX システムの場合、CAC_CONFIG 変数を cac.ini ファイルのディレクトリーに指定する必要があります。例：

```
CAC_CONFIG=/home/db2inst1/cac.ini
```
 - ODBC ドライバー・マネージャー・ライブラリーおよび ODBC ドライバー・ライブラリーのインストール・ディレクトリーによっては、DB2LIBPATH 環境変数が必須の場合もあります。
- 64 ビットの AIX および Solaris オペレーティング・システムでは、32 ビットの IBM InfoSphere Classic Federation Server for z/OS クライアントを使用している場合、DB2_FENCED ラッパー・オプションを Y に設定し、DB2_SOURCE_CLIENT_MODE ラッパー・オプションを 32BIT に設定する必要があります。DB2_SOURCE_CLIENT_MODE オプションがサポートされているのは、AIX および Solaris のみです。

インストール、構成、ODBC 要件については、個々の製品に添付されている資料を参照してください。

このタスクについて

制約事項

ODBC ラッパーは、IBM InfoSphere Classic Federation Server for z/OS データ・ソースに関して、以下のステートメントをサポートしていません。

- CREATE TABLE

IBM InfoSphere Classic Federation Server for z/OS データ・ソースに以下のデータ・タイプを使用することはサポートされていません。

- BLOB
- CLOB
- DBCLOB
- CHAR FOR BIT DATA
- VARCHAR FOR BIT DATA

手順

1. ご使用のオペレーティング・システムに応じて、以下のいずれかの作業をします。
 - Windows の場合、データ・ソースのノード名がシステム DSN として定義されていることを確認します。Microsoft ODBC データ・ソース・アドミニストレータを使用して DSN を定義した場合は、「コントロール パネル」でシステム DSN として登録されていることを確認できます。
 - UNIX の場合、ODBC ドライバーを構成します。
2. クライアント・ライブラリーを適切にロードするため、データ・ソース・クライアント・ライブラリー・ディレクトリーを DB2LIBPATH 環境変数に追加することが必要になる場合もあります。DB2LIBPATH 環境変数を設定して、データ・ソース・クライアント・ライブラリーのインストール・ディレクトリーを指定するには、**db2set** コマンドを実行します。

```
db2set DB2LIBPATH="client_library_directory"
```

ここで、*client_library_directory* はデータ・ソース・クライアント・ライブラリーのディレクトリーです。

64 ビット・バージョンの場合、DB2LIBPATH 環境変数は、必ず 64 ビット lib ディレクトリーに設定してください。

例 IBM InfoSphere Classic Federation Server for z/OS バージョン 8.2 クライアントの場合:

```
db2set DB2LIBPATH="/opt/IBM/DB2IIClassic82/cli/lib"
```

IBM InfoSphere Classic Federation Server for z/OS バージョン 9.1 クライアントの場合:

```
db2set DB2LIBPATH="/opt/IBM/WSCClassic91/cli/lib"
```

IBM InfoSphere Classic Federation Server for z/OS バージョン 9.5 クライアントの場合:

```
db2set DB2LIBPATH="/opt/IBM/WSCClassic95/cli/lib"
```

3. ODBC ラッパーを登録します。
4. ODBC データ・ソースのサーバー定義を登録します。
5. ODBC データ・ソースのユーザー・マッピングを作成します。
6. ODBC データ・ソース・サーバーとの接続をテストします。
7. ODBC データ・ソースの表およびビューのニックネームを登録します。

Netezza データ・ソースへの ODBC アクセスの構成

ODBC ラッパーは、Netezza データ・ソースへのアクセスのために最適化されています。ODBC ラッパーは、ODBC ドライバーを検出し、自動的にパフォーマンス・オプションを構成します。

始める前に

Netezza ODBC ドライバーがフェデレーテッド・サーバー上に構成されている必要があります。Netezza は、AIX、Linux、および Windows オペレーティング・システム用の ODBC ドライバーを提供します。

すべての ODBC データ・ソースに適用する前提条件については、131 ページの『ODBC データ・ソースへのアクセスの構成』を参照してください。

このタスクについて

ENABLE_BULK_INSERT サーバー・オプションを使用すると、Netezza データ・ソースへのバルク挿入を利用できます。

制約事項:

- バルク挿入は、パススルー・セッションではサポートされていません。
- Netezza は、すべてのフェデレーション機能をサポートするわけではありません。Netezza の制約事項の詳細なリストについては、Netezza 製品資料を参照してください。

その他の制約事項は、すべての ODBC データ・ソースに適用されます。131 ページの『ODBC データ・ソースへのアクセスの構成』を参照してください。

手順

1. オペレーティング・システムに応じて、以下の方法のいずれかを使用して、フェデレーテッド・サーバーとフェデレーテッド・データベースを準備します。
 - ODBC (Windows) を介してデータ・ソースにアクセスするためにフェデレーテッド・サーバーを準備します。
 - ODBC (Linux、UNIX) を通してデータ・ソースにアクセスするためにフェデレーテッド・サーバーを準備します。
2. クライアント・ライブラリーを適切にロードするため、データ・ソース・クライアント・ライブラリー・ディレクトリーを DB2LIBPATH 環境変数に追加することが必要になる場合もあります。DB2LIBPATH 環境変数を設定して、データ・ソース・クライアント・ライブラリーのインストール・ディレクトリーを指定するには、**db2set** コマンドを実行します。

```
db2set DB2LIBPATH="client_library_directory"
```

ここで、*client_library_directory* はデータ・ソース・クライアント・ライブラリーのディレクトリーです。

64 ビット・バージョンの場合、DB2LIBPATH 環境変数は必ず 64 ビットの lib ディレクトリーに設定してください。

例 Netezza 64 ビット ODBC ドライバー UNIX の場合:

```
db2set DB2LIBPATH=/<netezza_odbc_install_path>netezza/lib64
```

3. Linux、および UNIX オペレーティング・システムの場合、フェデレーテッド・サーバーに以下の項目を構成する必要があります。

- データ・ソースのクライアント・ライブラリー・ディレクトリーを db2dj.ini の DB2LIBPATH に追加します。

```
DB2LIBPATH=/<netezza_odbc_install_path>netezza/lib64
```

変更内容を有効にするために、DB2 インスタンスをリサイクルする必要があります。

- odbc.ini のシンボリック・リンクを作成します。

```
ln -s $ODBCINI $HOME/.odbc.ini
```

4. ODBC ラッパーを登録します。

5. ODBC データ・ソースのサーバー定義を登録します。

6. ODBC データ・ソースのユーザー・マッピングを作成します。

7. ODBC データ・ソース・サーバーとの接続をテストします。

8. ODBC データ・ソースの表およびビューのニックネームを登録します。

OLE DB データ・ソースへのアクセスの構成

フェデレーテッド・サーバーが OLE DB データ・ソースにアクセスするように構成するには、OLE DB プロバイダーに関する情報をフェデレーテッド・サーバーに提供しなければなりません。

始める前に

- フェデレーションが、フェデレーテッド・サーバーとして機能するサーバー上にインストールされている必要があります。
- フェデレーテッド・サーバーのセットアップを確認します。
- フェデレーテッド・パラメーターをチェックして、フェデレーションが有効になっていることを確認します。

このタスクについて

Microsoft OLE DB は、通常はデータベースに格納されていない情報にアプリケーションからアクセスできるようにするためのコンポーネント・オブジェクト・モデル (COM) インターフェースのセットです。OLE DB コンシューマーとは、システムまたはアプリケーション・コードのうち、OLE DB インターフェースを消費する部分のことです。OLE DB プロバイダーとは、OLE DB インターフェースを公開するソフトウェア・コンポーネントのことです。

DB2 コマンド行で SQL ステートメントを発行することによって、OLE DB データ・ソースに保管されているデータにアクセスするようフェデレーテッド・サーバーを構成することができます。

OLE DB データ・ソースへのアクセスを構成した後、CREATE FUNCTION ステートメントを使用して、ユーザー定義の OLE DB 外部表関数をフェデレーテッド・データベースに登録します。

手順

1. ラッパーに登録します。
2. サーバー定義に登録します。
3. ユーザー・マッピングを作成します。

OLE DB ラッパーの登録

OLE DB データ・ソースにアクセスするためには、ラッパーに登録する必要があります。フェデレーテッド・サーバーは、ラッパーを使用してデータ・ソースとの通信やデータ・ソースからのデータの取得を行います。ラッパーは、ライブラリー・ファイルのセットとしてインプリメントされます。

このタスクについて

OLE DB ラッパーにより、Microsoft OLE DB バージョン 2.0 以上に準拠する OLE DB プロバイダーにアクセスできるようになります。

OLE DB ラッパーは、フェデレーテッド・データベースにおけるユーザー定義の外部表関数の登録を支援する目的でのみ使用されます。他のラッパーとは異なり、OLE DB ラッパーは、データ・ソースに保管されているデータへのアクセスにニックネームを使用しません。

手順

コマンド行から CREATE WRAPPER ステートメントを発行し、OLE DB ラッパーにデフォルト名を指定します。

例:

```
CREATE WRAPPER OLEDB
```

要確認: デフォルト名 OLEDB を使用してラッパーに登録する場合、フェデレーテッド・サーバーは、そのフェデレーテッド・サーバーが実行されているオペレーティング・システムに該当する OLE DB ラッパー・ライブラリーを自動的に使用します。

デフォルトのラッパー名がフェデレーテッド・データベース内の既存のラッパー名と競合する場合、デフォルトのラッパー名を、指定の名前で置き換えることができます。デフォルト名とは別の名前を使用する場合は、CREATE WRAPPER ステートメントに LIBRARY パラメーターを含める必要があります。

たとえば、Windows を使用するフェデレーテッド・サーバー上で oledb_wrapper という名前のラッパーに登録するには、次のステートメントを発行します。

```
CREATE WRAPPER oledb_wrapper  
LIBRARY 'db2oledb.dll'
```

指定するラッパー・ライブラリー・ファイルは、フェデレーテッド・サーバーのオペレーティング・システムによって異なります。

次のタスク

この作業が完了したら、サーバー定義を登録することができます。

OLE DB ラッパー・ライブラリー・ファイル

OLE DB ラッパー・ライブラリー・ファイルは、ラッパーのインストール時にフェデレーテッド・サーバーに追加されます。

OLE DB ラッパーをインストールすると、デフォルトのディレクトリー・パスにライブラリー・ファイルが追加されます。

ラッパー登録時にデフォルトのラッパー名を使用しない場合は、**CREATE WRAPPER** ステートメントに **LIBRARY** パラメーターを含め、ライブラリー・ファイル名を指定する必要があります。

以下の表に、デフォルトのディレクトリー・パスとデフォルトのラッパー・ライブラリー・ファイルの名前がリストされています。

表 31. OLE DB クライアント・ライブラリーの場所とファイル名

オペレーティング・システム	ディレクトリー・パス	ライブラリー・ファイル名
Windows	%DB2PATH%\%bin	db2oledb.dll

OLE DB データ・ソースのサーバー定義の登録

フェデレーテッド・データベース中でアクセスする各 OLE DB データ・ソース・サーバーごとに登録が必要です。

このタスクについて

OLE DB データ・ソースへのサーバー定義の登録は、DB2 コマンド行から実行します。

手順

CREATE SERVER ステートメントを発行します。

例:

```
CREATE SERVER server_definition_name WRAPPER wrapper_name
  OPTIONS (CONNECTSTRING 'keyword=value;keyword=value');
```

CONNECTSTRING 変数は、**CREATE SERVER** ステートメントではオプションですが、OLE DB データ・ソースの場合、このオプションは必須です。

次のタスク

この作業が完了したら、ユーザー・マッピングを作成できます。

CREATE SERVER ステートメント - OLE DB ラッパーの例

OLE DB ラッパーにサーバー定義を登録するには、CREATE SERVER ステートメントを使用します。このトピックでは、必須パラメーターを使用する完全指定の例と、追加のサーバー・オプションを使用する例を提供します。

以下に示すのは、CREATE SERVER ステートメントを発行して、OLE DB ラッパーのサーバー定義を登録する方法の例です。

```
CREATE SERVER Nwind WRAPPER OLEDB
  OPTIONS (CONNECTSTRING 'Provider=Microsoft.Jet.OLEDB.4.0;
    Data Source=c:\msdasdk\bin\oledb\wind.mdb');
```

Nwind OLE DB データ・ソースに割り当てる名前。重複するサーバー定義名は使用できません。

WRAPPER OLEDB

CREATE WRAPPER ステートメントで指定したラッパー名。

```
CONNECTSTRING 'Provider=Microsoft.Jet.OLEDB.4.0; Data
Source=c:\msdasdk\bin\oledb\wind.mdb'
```

OLE DB プロバイダーへの接続に必要な初期化プロパティを提供します。

CONNECTSTRING オプションの値には、一連のキーワードと値の対をセミコロンで区切ったものを入れます。等号 (=) は、各キーワードとその値を区切ります。キーワードは、OLE DB 初期化プロパティ (プロパティ・セット DBPROPSET_DBINT) の記述、またはプロバイダー特有のキーワードです。

CONNECTSTRING オプションの完全な構文とセマンティクスについては、「*Microsoft OLE DB 2.0 Programmer's Reference and Data Access SDK*」(Microsoft Press、1998) を参照してください。

サーバー・オプション

サーバー定義の作成時に、CREATE SERVER ステートメントに追加のサーバー・オプションを指定することができます。サーバー・オプションには、汎用サーバー・オプションと OLE DB に特有のサーバー・オプションを指定できます。

COLLATING_SEQUENCE サーバー・オプションは、データ・ソースがフェデレーテッド・サーバーと同じ照合シーケンスを使用しているかどうかを指定します。**COLLATING_SEQUENCE** サーバー・オプションのデフォルトの設定値は 'N' です。OLE DB データ・ソースで使用されている照合シーケンスがフェデレーテッド・サーバーと同じ場合は、**COLLATING_SEQUENCE** サーバー・オプションを 'Y' に設定します。

以下の例は、**COLLATING_SEQUENCE** サーバー・オプションを含む OLE DB サーバー定義を示します。

```
CREATE SERVER Nwind WRAPPER OLEDB
  OPTIONS (CONNECTSTRING 'Provider=Microsoft.Jet.OLEDB.4.0;
    Data Source=c:\msdasdk\bin\oledb\wind.mdb',
  COLLATING_SEQUENCE 'Y')
```

OLE DB データ・ソースのユーザー・マッピングの作成

OLE DB サーバーにアクセスしようとする、フェデレーテッド・サーバーは、データ・ソースに対して有効なユーザー ID とパスワードを使用することによって、その OLE DB サーバーへの接続を確立します。それぞれのフェデレーテッド許可 ID とパスワード、および対応するデータ・ソースのユーザー ID とパスワードの間の関連付け (ユーザー・マッピング) を定義しなければなりません。

手順

CREATE USER MAPPING ステートメントを発行します。
たとえば、次のようにします。

```
CREATE USER MAPPING FOR local_userid SERVER server_definition_name  
    OPTIONS (REMOTE_AUTHID 'remote_userID', REMOTE_PASSWORD 'remote_password');
```

OLE DB データ・ソースのパスワードかフェデレーテッド・サーバーのパスワードのいずれかの長さが 8 文字未満の場合、OLE DB データ・ソースにアクセスする SQL ステートメントは失敗します。次のエラー・メッセージが表示されます。

```
SQL30082N Attempt to establish connection failed with security reason "15"  
("PROCESSING FAILURE"). SQLSTATE=08001
```

この問題を避けるには、OLE DB データ・ソースのパスワードかフェデレーテッド・サーバーのパスワードを 8 文字以上に変更してください。

CREATE USER MAPPING ステートメント - OLE DB ラッパーの例

CREATE USER MAPPING ステートメントを使用して、フェデレーテッド許可 ID をリモート OLE DB のユーザー ID およびパスワードにマップします。このトピックには、必須パラメーターを使用する完全指定の例と、CREATE USER MAPPING ステートメントで DB2 特殊レジスター USER を使用する方法を示す例が記載されています。

以下に示すのは、フェデレーテッド許可 ID をリモート OLE データ・ソースのユーザー ID およびパスワードにマップする方法の例です。

```
CREATE USER MAPPING FOR laura SERVER Nwind  
    OPTIONS (REMOTE_AUTHID 'lulu', REMOTE_PASSWORD 'raiders')
```

laura OLE DB データ・ソースで定義されたリモート・ユーザー ID およびパスワードにマップするローカル許可 ID を指定します。

SERVER *Nwind*

OLE DB サーバーに関する CREATE SERVER ステートメントで登録したサーバー定義名を指定します。

REMOTE_AUTHID '*lulu*'

lulu のマップ先となる OLE DB データベース・サーバー側のユーザー ID を指定します。CREATE SERVER ステートメントで FOLD_ID サーバー・オプションを「U」または「L」に設定している場合を除き、この値は大文字小文字を区別します。

REMOTE_PASSWORD '*raiders*'

lulu に関連するパスワードを指定します。CREATE SERVER ステートメ

ントで FOLD_PW サーバー・オプションを「U」または「L」に設定している場合を除き、この値は大文字小文字を区別します。

DB2 特殊レジスター USER

DB2 特殊レジスター USER を使用して、CREATE USER MAPPING ステートメントを実行しているユーザーの許可 ID を、REMOTE_AUTHID ユーザー・オプションに指定されているデータ・ソースの許可 ID にマップすることができます。

以下に、特殊レジスター USER を含む CREATE USER MAPPING ステートメントの例を示します。

```
CREATE USER MAPPING FOR USER SERVER Nwind
    OPTIONS (REMOTE_AUTHID 'lulu', REMOTE_PASSWORD 'raiders');
```

Oracle データ・ソースへのアクセスの構成

Oracle データ・ソースにアクセスするようにフェデレーテッド・システムを構成するには、アクセスするデータ・ソースとオブジェクトの情報をフェデレーテッド・システムに提供しなければなりません。

始める前に

- Oracle クライアント・ソフトウェアが、フェデレーテッド・サーバーとして機能するサーバー上にインストールされ、構成されている必要があります。
- フェデレーションが、フェデレーテッド・サーバーとして機能するサーバー上にインストールされている必要があります。
- フェデレーテッド・サーバーのセットアップを確認します。
- フェデレーテッド・パラメーターをチェックして、フェデレーションが有効になっていることを確認します。

手順

1. Oracle 環境変数をセットアップします。
2. Oracle クライアント構成ファイルをセットアップしてテストします。
3. Oracle ラッパーを登録します。
4. Oracle データ・ソースのサーバー定義を登録します。
5. ユーザー・マッピングを作成します。
6. Oracle サーバーとの接続をテストします。
7. ニックネームを登録します。

Oracle 環境変数の設定

Oracle 環境変数は、フェデレーテッド・サーバー上の db2dj.ini ファイルで設定する必要があります。

このタスクについて

制約事項

db2dj.ini ファイルに関する制約事項を確認してください。

db2dj.ini ファイルには、フェデレーテッド・サーバー上にインストールされている Oracle クライアント・ソフトウェアの構成情報が入っています。

Oracle データ・ソース用の環境変数には、必須の変数とオプションの変数とがあります。

Oracle ラッパーをインストールする前に Oracle クライアント・ソフトウェアをインストールした場合、必須の Oracle 環境変数は db2dj.ini ファイルに設定されています。

Oracle ラッパーをインストールする前に Oracle クライアント・ソフトウェアをインストールしていない場合、またはいずれかのオプションの環境変数を設定する場合には、ここで示す手順にしたがって環境変数を設定する必要があります。

手順

1. 次の方法のいずれかを使用します。

方法	説明
環境変数を自動で設定します。	IBM InfoSphere Federation Server インストール・ウィザードを実行します。ウィザードの指示に従ってください。 重要: インストール・ウィザードを実行することによって、必須の環境変数を設定します。オプションの環境変数は手動で設定する必要があります。
ラッパー構成ツールで環境変数を設定します。	ラッパー構成ツールを実行し、必須の環境変数およびオプションの環境変数を設定します。ラッパー構成ツールの実行の指示に従ってください。
環境変数を手動で設定します。	db2dj.ini ファイルを編集します。 db2dj.ini ファイルは、DB2 レジストリー変数 DB2_DJ_INI によって指定されたディレクトリー内にあります。DB2_DJ_INI 変数が設定されていない場合、db2dj.ini ファイルは、オペレーティング・システムに応じて以下のデフォルト・パスのいずれかにあります。 <ul style="list-style-type: none"> Linux および UNIX の場合: <code>instancehome/sqllib/cfg/db2dj.ini</code> <code>instancehome</code> インスタンス所有者のホーム・ディレクトリーです。 Windows の場合: <code>%DB2PATH%\cfg\db2dj.ini</code> <code>%DB2PATH%</code> DB2 データベース・システムがインストールされているディレクトリー (例えば、<code>C:\Program Files\IBM\sqllib</code>) です。 <p>このファイルが存在しない場合、任意のテキスト・エディターを使用して、db2dj.ini という名前のファイルを作成できます。db2dj.ini ファイルには、環境変数の値に完全修飾パスを指定する必要があります。そうしないと、エラーが発生します。</p>

- UNIX が稼働するフェデレーテッド・サーバーの場合は、DB2 インスタンスの `.profile` ファイルに Oracle 環境変数を追加します。例えば、次のようにします。

```
export ORACLE_HOME=oracle_home_directory
export PATH=$ORACLE_HOME/bin:$PATH
```

ここで、`oracle_home_directory` は Oracle クライアント・ソフトウェアがインストールされているディレクトリーです。

- Linux または UNIX が稼働するフェデレーテッド・サーバーの場合、次のように入力して、DB2 インスタンス `.profile` ファイルを実行します。

```
.$HOME/.profile
```

- 環境変数がフェデレーテッド・サーバーで設定されていることを確認するには、以下のコマンドで DB2 インスタンスをリサイクルします。

```
db2stop
db2start
```

次のタスク

この作業が終了したら、Oracle クライアントのセットアップとテストを行えます。

Oracle 環境変数

Oracle データ・ソース用の環境変数には、必須の変数とオプションの変数とがあります。これらの変数は `db2dj.ini` ファイルに設定されます。

以下の環境変数は、Oracle に有効です。

- ORACLE_HOME
- ORACLE_BASE (オプション)
- ORA_NLS (オプション)
- NLS_LANG (オプション)
- TNS_ADMIN (オプション)

変数の説明

ORACLE_HOME

ORACLE_HOME 環境変数に、Oracle クライアント・ソフトウェアがインストールされているディレクトリー・パスを設定します。この環境変数 `ORACLE_HOME=oracle_home_directory` には、完全修飾パスを指定します。たとえば、Oracle ホーム・ディレクトリーが `¥usr¥oracle¥8.1.7` である場合、`db2dj.ini` ファイルの項目は、`ORACLE_HOME=¥usr¥oracle¥8.1.7` となります。

フェデレーテッド・インスタンスの個々のユーザーが ORACLE_HOME 環境変数をローカルに設定していても、フェデレーテッド・インスタンスはその設定を使用しません。フェデレーテッド・インスタンスは、`db2dj.ini` ファイルに設定された ORACLE_HOME の値だけを使用します。

ORACLE_BASE

ORACLE_BASE は、Oracle クライアント・ディレクトリー・ツリーのルートを表します。Oracle クライアント・ソフトウェアをインストールする際

に ORACLE_BASE 環境変数を設定したら、フェデレーテッド・サーバーに ORACLE_BASE 環境変数を設定します。

例:

```
ORACLE_BASE=oracle_root_directory
```

ORA_NLS

Oracle の複数のバージョンがシステム上で稼働している場合、以下を確認します。

- 適切な ORA_NLS 環境変数が設定されている。
- 使用するバージョンに対応する NLS データ・ファイルが使用可能である。

ORA_NLS 環境変数で指定されるディレクトリーに、場所に固有のデータが保存されます。Oracle はバージョンごとに異なる ORA_NLS データ・ディレクトリーを持っています。

表 32. 各バージョンの ORA_NLS 環境変数

Oracle バージョン	環境変数
8.x、9.x	ORA_NLS33
10.x	ORA_NLS10

たとえば、Oracle 8.1 データ・ソースにアクセスする UNIX の稼働するフェデレーテッド・サーバーでは、ORA_NLS33 環境変数の設定は以下のようになります。

```
ORA_NLS33=oracle_home_directory/ocommon/nls/admin/<data>
```

NLS_LANG

NLS_LANG 環境変数は、コード・ページ環境変数です。この変数の設定については、Oracle NLS の資料を参照してください。

TNS_ADMIN

Windows が稼働するフェデレーテッド・サーバーの場合

フェデレーテッド・サーバーの場合、Oracle クライアントは tnsnames.ora ファイルを %ORACLE_HOME%\NETWORK\ADMIN ディレクトリーで探します。ここで %ORACLE_HOME% は db2dj.ini ファイルで定義されます。tnsnames.ora ファイルが %ORACLE_HOME%\NETWORK\ADMIN ディレクトリーにない場合、フェデレーテッド・サーバー上の db2dj.ini ファイルで、TNS_ADMIN 環境変数を設定する必要があります。この環境変数は、db2dj.ini ファイルで、tnsnames.ora ファイルが見つかるパスを設定します。

AIX または Linux が稼働するフェデレーテッド・サーバーの場合

Oracle クライアントは tnsnames.ora ファイルを /etc ディレクトリーで探します。tnsnames.ora ファイルが /etc ディレクトリーにない場合、Oracle クライアントは、\$ORACLE_HOME/network/admin ディレクトリーで tnsnames.ora ファイルを探します。\$ORACLE_HOME は、db2dj.ini ファイルで定義されます。tnsnames.ora ファイルが \$ORACLE_HOME/network/admin ディレクトリーにない場合、フェデレーテッド・サーバー上に TNS_ADMIN

環境変数を設定する必要があります。この環境変数は、db2dj.ini ファイルで、tnsnames.ora ファイルが見つかるパスを設定します。

たとえば、tnsnames.ora ファイルが /home/oracle ディレクトリーにある場合、次のようにこの環境変数を設定します。

```
TNS_ADMIN=/home/oracle
```

Solaris が稼働するフェデレーテッド・サーバーの場合

フェデレーテッド・サーバーの場合、Oracle クライアントは tnsnames.ora ファイルを /var/opt/oracle ディレクトリーで探します。tnsnames.ora ファイルが /var/opt/oracle ディレクトリーにない場合、Oracle クライアントは、\$ORACLE_HOME/network/admin ディレクトリーで tnsnames.ora ファイルを探します。

\$ORACLE_HOME は、db2dj.ini ファイルで定義されます。

tnsnames.ora ファイルが \$ORACLE_HOME/network/admin ディレクトリーにない場合、TNS_ADMIN 環境変数を設定する必要があります。この変数は、db2dj.ini ファイルで、tnsnames.ora ファイルが見つかるパスを設定します。

たとえば、tnsnames.ora ファイルが /home/oracle ディレクトリーにある場合、次のようにこの環境変数を設定します。

```
TNS_ADMIN=/home/oracle
```

Oracle コード・ページ変換:

Oracle ラッパーが Oracle データ・ソースに接続するたびに、ラッパーはその接続のためにどのコード・ページを使用するかを決定します。コード・ページ値を Oracle ラッパーが設定するよう指定したり、あるいは NLS_LANG 環境変数を設定することでコード・ページを指定することができます。

Oracle コード・ページ変換を指定する環境変数は、フェデレーテッド・サーバーの db2dj.ini ファイルで設定されます。

フェデレーテッド・サーバー上の db2dj.ini ファイルで NLS_LANG 環境変数が設定されている場合、ラッパーは db2dj.ini ファイルのコード・ページ値を使用します。

フェデレーテッド・サーバー上の db2dj.ini ファイルで NLS_LANG 環境変数が設定されていない場合、ラッパーはフェデレーテッド・データベースのテリトリーとコード・ページを判断します。ラッパーは NLS_LANG 環境を一致する Oracle ロケールのうち最も近いものに設定します。近く一致するロケールがない場合、NLS_LANG 環境は American_America.US7ASCII に設定されます。

有効なロケールのリストは、Oracle ソフトウェアに付属する資料を参照してください。

中国語コード・ページ GB 18030 の例:

中国語コード・ページ GB 18030 でエンコードされたデータを含む Oracle データ・ソースにアクセスする際、フェデレーテッド・データベースが UTF-8 コード・ページを使用する場合、Oracle ラッパーは、NLS_LANG 環境変数を以下のように設定します。

```
NLS_LANG=Simplified Chinese_China.UTF8
```

この設定は、Oracle 8 クライアントを使用している場合には正しいですが、Oracle バージョン 9i (またはそれ以降) クライアントを使用している場合には、NLS_LANG 環境変数を設定して、Oracle ラッパーのデフォルト設定をオーバーライドする必要があります。Oracle 9i クライアントが GB 18030 データを Unicode に正しく変換できるよう、NLS_LANG 環境変数を Simplified Chinese_China.AL32UTF8 に設定します。

例:

```
NLS_LANG=Simplified Chinese_China.AL32UTF8
```

Oracle クライアント構成ファイルのセットアップおよびテスト

Oracle クライアント構成ファイルは、フェデレーテッド・システムにインストールされているクライアント・ライブラリーを使用して、Oracle データベースに接続するために使用されます。

このタスクについて

クライアント構成ファイルは、各 Oracle データベース・サーバーのロケーション、およびそのデータベース・サーバーへの接続のタイプ (プロトコル) を指定します。

Oracle クライアント構成ファイルのデフォルト名は `tnsnames.ora` です。

クライアント構成ファイルのデフォルトのロケーションは、フェデレーテッド・システムによって使用されるオペレーティング・システムによって異なります。

- Linux および UNIX システムでは、ファイルのデフォルト・ロケーションは、`$ORACLE_HOME/network/admin` です。
- Windows システムでは、ファイルのデフォルト・ロケーションは、`%ORACLE_HOME%\network\admin` です。

手順

1. Oracle クライアント・ソフトウェアに付属してくる Oracle NET8/NET 構成ユーティリティを使用して、`tnsnames.ora` を作成します。

`tnsnames.ora` ファイル内で、SID (または `SERVICE_NAME`) は Oracle インスタンスの名前であり、HOST は Oracle サーバーが存在するホスト名です。

2. `tnsnames.ora` ファイルをデフォルトの検索パス以外のパスに置く場合は、`TNS_ADMIN` 環境変数を設定してファイルのロケーションを指定してください。
 - a. `sqllib/cfg` ディレクトリーにある `db2dj.ini` ファイルを編集し、`TNS_ADMIN` 環境変数を次のように設定します。例:

```
TNS_ADMIN=x:/path/
```
 - b. DB2 インスタンスをリサイクルし、プログラムで環境変数が確実に設定されるようにするため、次のコマンドを発行します。

```
db2stop  
db2start
```

3. クライアント・ソフトウェアが Oracle サーバーに接続できるか確認するために、接続をテストしてください。接続をテストするには、Oracle sqlplus ユーティリティを使用します。

次のタスク

この作業が終了したら、Oracle ラッパーを登録できます。

Oracle ラッパーの登録

Oracle データ・ソースにアクセスするためには、ラッパーを登録する必要があります。フェデレーテッド・サーバーは、ラッパーを使用してデータ・ソースとの通信やデータ・ソースからのデータの取得を行います。ラッパーは、ライブラリー・ファイルのセットとしてインプリメントされます。

手順

Oracle ラッパーを登録するには、以下の方法のいずれかを使用します。

方法	説明
Data Studio の「管理エクスプローラー (Administration Explorer)」を使用します。	「フェデレーテッド・データベース・オブジェクト」フォルダーで、「ラッパー」フォルダーを右クリックし、「ラッパーの作成 (Create Wrapper)」を選択します。
CREATE WRAPPER ステートメントを発行し、Oracle ラッパーにデフォルト名を指定します。	<p>例:</p> <pre>CREATE WRAPPER NET8 ;</pre> <p>要確認: デフォルト名 NET8 を使用してラッパーを登録する場合、フェデレーテッド・サーバーは自動的に、フェデレーテッド・サーバーが稼働するオペレーティング・システムに対応する適切な Oracle ラッパー・ライブラリーを使用します。</p> <p>ラッパーを登録する際にデフォルト・ラッパー名を使用しない場合は、CREATE WRAPPER ステートメントに LIBRARY パラメーターを指定して、デフォルト・ラッパー・ライブラリー・ファイル名を指定する必要があります。</p> <p>たとえば、AIXを使用するフェデレーテッド・サーバー上で、oracle_wrapper という名前でラッパーを登録するには、次のステートメントを発行します。</p> <pre>CREATE WRAPPER oracle_wrapper LIBRARY 'libdb2net8.a'</pre> <p>指定するラッパー・ライブラリー・ファイルは、フェデレーテッド・サーバーのオペレーティング・システムによって異なります。</p>

次のタスク

この作業が完了したら、サーバー定義の登録ができます。

Oracle ラッパー・ライブラリー・ファイル

Oracle ラッパー・ライブラリー・ファイルは、ラッパーのインストール時にフェデレーテッド・サーバーに追加されます。

Oracle ラッパーのインストール時に、デフォルトのディレクトリー・パスに 3 つのライブラリー・ファイルが追加されます。たとえば、フェデレーテッド・サーバーが AIX で稼働している場合、以下のディレクトリー・パスに追加されるラッパー・ライブラリー・ファイルは、libdb2net8.a、libdb2net8F.a、libdb2net8U.a です。デフォルトのラッパー・ライブラリー・ファイルは、libdb2net8.a です。他のラッパー・ライブラリー・ファイルは、Oracle ラッパーによって内部で使用されます。

以下の表に、デフォルトのディレクトリー・パスとデフォルトのラッパー・ライブラリー・ファイルの名前をリストします。

表 33. Oracle ラッパー・ライブラリーの場所とファイル名

オペレーティング・システム	ディレクトリー・パス	ライブラリー・ファイル名
AIX	/usr/opt/install_path/lib32/	libdb2net8.a
	/usr/opt/install_path/lib64/	
Linux	/opt/IBM/db2/install_path/lib32	libdb2net8.so
	/opt/IBM/db2/install_path/lib64	
Solaris	/opt/IBM/db2/install_path/lib32	libdb2net8.so
	/opt/IBM/db2/install_path/lib64	
Windows	%DB2PATH%\bin	db2net8.dll

install_path は、UNIX または Linux 上でフェデレーテッド・サーバーがインストールされているディレクトリー・パスです。

Oracle データ・ソースのサーバー定義の登録

フェデレーテッド・データベース内に、アクセスする Oracle サーバーをそれぞれ登録する必要があります。

手順

1. Oracle tnsnames.ora ファイルでノード名を見つけます。 tnsnames.ora ファイルの例:

```
paris_node =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = somehost)(PORT = 1521)))
    (CONNECT_DATA = (SERVICE_NAME = ora9i.see1)))
```

この例の場合、CREATE SERVER ステートメントで使用するノード値は *paris_node* です。

node_name は CREATE SERVER SQL ステートメントでオプションとなっていますが、Oracle データ・ソースの場合は必要です。

2. Oracle データ・ソースのサーバー定義を登録するには、次の方法のいずれかを使用します。

方法	説明
Data Studio の「管理エクスプローラー (Administration Explorer)」を使用します。	「フェデレーテッド・データベース・オブジェクト」フォルダーで、「リモート・サーバー」フォルダーを右クリックし、「リモート・サーバーの作成 (Create Remote Servers)」を選択します。
CREATE SERVER ステートメントを発行します。	例: <pre>CREATE SERVER server_name TYPE oracle VERSION 8.1.7 WRAPPER net8 OPTIONS (NODE 'node_name')</pre> サーバー定義を作成した後、サーバー・オプションを追加またはドロップするには、ALTER SERVER ステートメントを使用します。

次のタスク

この一連のタスクの次のタスクは、『Oracle データ・ソースのユーザー・マッピングの作成』です。

CREATE SERVER ステートメント - Oracle ラッパーの例

Oracle ラッパーにサーバー定義を登録するには、CREATE SERVER ステートメントを使用します。このトピックでは、必須パラメーターを使用する完全指定の例と、追加のサーバー・オプションを使用する例を提供します。

以下に示すのは、CREATE SERVER ステートメントを発行して、Oracle ラッパーのサーバー定義を登録する方法の例です。

```
CREATE SERVER oraserver TYPE oracle VERSION 8.1.7 WRAPPER wrapper_name
OPTIONS (NODE 'paris_node');
```

oraserver

Oracle データベース・サーバーに割り当てた名前。重複するサーバー定義名は使用できません。

TYPE *oracle*

アクセスを構成するデータ・ソース・サーバーのタイプを指定します。Oracle ラッパーの場合、サーバー・タイプは `oracle` でなければなりません。

VERSION *8.1.7*

アクセスする Oracle データベース・サーバーのバージョン。

WRAPPER *wrapper_name*

CREATE WRAPPER ステートメントで指定したラッパー名。

NODE *'paris_node'*

Oracle データベース・サーバーが存在するノードの名前。ノード名は、tnsnames.ora ファイルから入手します。この値は大文字と小文字を区別します。

ノード名は CREATE SERVER ステートメントではオプションとなっていますが、Oracle データ・ソースの場合は必要です。

サーバー・オプション

サーバー定義の作成時に、CREATE SERVER ステートメントに追加のサーバー・オプションを指定することができます。これらのサーバー・オプションには、汎用のサーバー・オプションと Oracle に特有のサーバー・オプションを指定できます。

フェデレーテッド・サーバーは、Oracle VARCHAR 列にはすべて、末尾空白が含まれていると想定しています。Oracle データベース内のすべての VARCHAR 列が末尾空白を含まないことが確実である場合は、データ・ソースが空白埋め込みなしの VARCHAR 比較セマンティックを使用するということを示す、サーバー・オプションを設定できます。

以下に、VARCHAR_NO_TRAILING_BLANKS サーバー・オプションを使用した Oracle サーバー定義の例を示します。

```
CREATE SERVER oraserver TYPE oracle VERSION 8.1.7 WRAPPER wrapper_name
  OPTIONS (NODE 'paris_node', VARCHAR_NO_TRAILING_BLANKS 'Y');
```

末尾空白を含む列が 1 つもない場合は、VARCHAR_NO_TRAILING_BLANKS サーバー・オプションを使用します。末尾空白を含む VARCHAR 列がいくつかしかない場合は、ALTER NICKNAME ステートメントを使用して、これらの列にオプションを設定できます。

Oracle データ・ソースのユーザー・マッピングの作成

Oracle サーバーにアクセスする場合、フェデレーテッド・サーバーはデータ・ソースにとって有効なユーザー ID とパスワードを使用して、Oracle サーバーへの接続を確立します。

このタスクについて

制約事項

Oracle データ・ソース側のユーザー ID は、Oracle **create user** コマンドで **identified externally** 文節ではなく、**identified by** 文節を使用して作成する必要があります。

手順

Oracle データ・ソースのユーザー・マッピングを作成するには、以下の方法のいずれかを使用します。

方法	説明
Data Studio の「管理エクスプローラー (Administration Explorer)」を使用します。	「フェデレーテッド・データベース・オブジェクト」フォルダーで、「ユーザー・マッピング」フォルダーを右クリックし、「ユーザー・マッピングの作成 (Create User Mapping)」を選択します。

方法	説明
CREATE USER MAPPING ステートメントを発行します。	例: <pre>CREATE USER MAPPING FOR <i>local_userID</i> SERVER <i>server_definition_name</i> OPTIONS (REMOTE_AUTHID '<i>remote_userID</i>', REMOTE_PASSWORD '<i>remote_password</i>');</pre>

次のタスク

この作業が終了したら、Oracle サーバーへの接続をテストします。

CREATE USER MAPPING ステートメント - Oracle ラッパーの例

CREATE USER MAPPING ステートメントを使用して、フェデレーテッド許可 ID をリモートの Oracle ユーザー ID およびパスワードにマップします。このトピックには、必須パラメーターを使用する完全指定の例と、CREATE USER MAPPING ステートメントで DB2 特殊レジスター USER を使用する方法を示す例が含まれます。

以下の例は、フェデレーテッド許可 ID を Oracle ユーザー ID およびパスワードにマップする方法を示しています。

```
CREATE USER MAPPING FOR robert SERVER oraserver
OPTIONS (REMOTE_AUTHID 'rob', REMOTE_PASSWORD 'then4now');
```

robert Oracle サーバーで定義されたりモート・ユーザー ID およびパスワードにマップする許可 ID を指定します。

SERVER *oraserver*

CREATE SERVER ステートメントで登録した Oracle サーバーのサーバー定義名を指定します。

REMOTE_AUTHID '*rob*'

robert のマップ先となるリモート・ユーザー ID を指定します。CREATE SERVER ステートメントで FOLD_ID サーバー・オプションを「U」または「L」に設定している場合を除き、この値は大文字小文字を区別します。

REMOTE_PASSWORD '*then4now*'

rob に関連するリモート・パスワードを指定します。CREATE SERVER ステートメントで FOLD_PW サーバー・オプションを「U」または「L」に設定している場合を除き、この値は大文字小文字を区別します。

DB2 特殊レジスター USER

DB2 特殊レジスター USER を使用して、CREATE USER MAPPING ステートメントを実行しているユーザーの許可 ID を、REMOTE_AUTHID ユーザー・オプションに指定されているデータ・ソースの許可 ID にマップすることができます。

以下に、特殊レジスター USER を含む CREATE USER MAPPING ステートメントの例を示します。

```
CREATE USER MAPPING FOR USER SERVER oraserver
OPTIONS (REMOTE_AUTHID 'rob', REMOTE_PASSWORD 'then4now');
```

パブリック・ユーザー・マッピング

InfoSphere Federation Server バージョン 9.7 フィックスパック 2 以降では、すべてのローカル・データベース・ユーザーが 1 つのリモート・ユーザー ID とパスワードでデータ・ソースにアクセスできるようにするパブリック・ユーザー・マッピングを作成できます。

この例では、ラッパー net8 と Oracle データ・ソース・サーバー oraserver が作成されます。次に、CREATE USER MAPPING ステートメントが PUBLIC と共に指定され、サーバー oraserver に対するパブリック・ユーザー・マッピングが作成され、すべてのローカル・データベース・ユーザーが CREATE USER MAPPING ステートメントの OPTIONS で指定されている許可 ID にマップされます。

例

```
CREATE WRAPPER net8;

CREATE SERVER oraserver
  TYPE oracle VERSION 8.1.7 WRAPPER net8
  OPTIONS (NODE 'paris_node');

CREATE USER MAPPING FOR PUBLIC SERVER oraserver
  OPTIONS (REMOTE_AUTHID 'APP_USER', REMOTE_PASSWORD 'secret');
```

Oracle サーバーへの接続のテスト

フェデレーテッド・サーバーが Oracle データ・ソースにアクセスするために適切に構成されているかを確認するため、Oracle サーバーへの接続をテストします。

このタスクについて

Oracle サーバーへの接続のテストは、定義したサーバー定義とユーザー・マッピングを使用して行うことができます。

手順

パススルー・セッションを開き、Oracle システム表に対して SELECT ステートメントを発行します。SELECT ステートメントがカウントを戻す場合、サーバー定義およびユーザー・マッピングは適切にセットアップされています。

例:

```
SET PASSTHRU remote_server_name
SELECT count(*) FROM sys.all_tables
SET PASSTHRU RESET
```

SELECT ステートメントがエラーを戻す場合には、接続エラーのトラブルシューティングを行う必要があります。

次のタスク

この作業が終了したら、Oracle 表およびビューのニックネームを登録できます。

Oracle データ・ソースの接続に関する問題のトラブルシューティング

Oracle データ・ソースにアクセスするためにフェデレーテッド・サーバーを構成する場合にユーザーが最も多く直面する問題は、接続に関するものです。

症状

フェデレーテッド・サーバーから Oracle データ・ソースに接続できない場合は、TCP/IP hosts ファイルを更新する必要があるかもしれません。

原因

この問題は、TCP/IP hosts ファイルが古くて無効になっていることが原因である可能性があります。

問題の解決方法

tnsnames.ora ファイルの DESCRIPTION セクションにある HOST ごとに、TCP/IP hosts ファイルを更新することが必要になる場合があります。このファイルを更新するかどうかは、ネットワークで TCP/IP がどのように構成されているかによります。ネットワークの一部で、tnsnames.ora ファイルの DESCRIPTION セクションに指定されたリモート・ホスト名をアドレスに変換する必要があります。

ホスト名を認識するネーム・サーバーがネットワークにある場合は、TCP/IP hosts ファイルを更新する必要はありません。ホスト名を認識するネーム・サーバーがネットワークに存在しない場合は、リモート・ホストの TCP/IP hosts ファイルに項目を追加する必要があります。

TCP/IP hosts ファイルのロケーションは、フェデレーテッド・サーバーで稼働するオペレーティング・システムによって異なります。

Linux または UNIX が稼働するフェデレーテッド・サーバーの場合

hosts ファイルは /etc/hosts ディレクトリー内にあります。

Windows が稼働するフェデレーテッド・サーバーの場合

hosts ファイルは、x:%winnt%system32%drivers%etc%hosts ディレクトリー内にあります。

データ・ソース接続エラーのトラブルシューティング

データ・ソース・サーバーへのテスト接続では、いくつかの理由でエラーが戻されることがあります。以下の処置により、エラーが発生した理由を判別することができます。

症状

データ・ソースに接続しようとしたときにエラーが戻されます。

原因

接続の問題には、いくつかの原因が考えられます。

問題の解決方法

データ・ソース接続エラーのトラブルシューティングを行うには、以下の項目で問題がないかどうかをチェックします。

- データ・ソースが使用可能であることを確認します。
- 該当する場合は、データ・ソース・サーバーが着信接続に対して構成されていることを確認します。

- REMOTE_AUTHID と REMOTE_PASSWORD のオプションに関するユーザー・マッピング設定を調べ、データ・ソースとの接続に有効な設定になっているか確認します。必要に応じて、ユーザー・マッピングを変更するか、別のユーザー・マッピングを作成します。
- 該当する場合は、フェデレーテッド・サーバー上のデータ・ソース・クライアント・ソフトウェアがインストールされていて、データ・ソースに接続できるように正しく構成されていることを確認します。
- ODBC データ・ソースの場合、フェデレーテッド・サーバー上の ODBC ドライバーがインストールされていて、ODBC データ・ソース・サーバーに接続できるように正しく構成されていることを確認します。Windows が稼働するフェデレーテッド・サーバーでは、ドライバーをチェックするときに「ODBC データ ソース アドミニストレータ」ツールを使用します。UNIX が稼働するフェデレーテッド・サーバーの場合は、ODBC クライアントのベンダーが提供している資料を参照してください。
- フェデレーテッド・サーバー上で設定されている変数の設定がデータ・ソースについて適切であることを確認します。これらの変数には、システム環境変数、db2dj.ini ファイル、および DB2 Profile Registry (db2set) 変数が含まれます。
- サーバー定義を調べます。必要であれば、サーバー定義をドロップして再作成します。

Oracle 表およびビューのニックネームの登録

登録するそれぞれの Oracle サーバー定義で、アクセスするそれぞれの表またはビューごとにニックネームを登録する必要があります。Oracle サーバーを照会するときには、データ・ソース・オブジェクトの名前ではなく、これらのニックネームを使用します。

始める前に

ニックネームを登録する前に、Oracle データ・ソース側の統計を更新します。フェデレーテッド・データベースは、データ・ソースのカatalog統計に基づいて、照会処理を最適化します。DB2 RUNSTATS コマンドに相当するデータ・ソース・コマンドを使用して、データ・ソースの統計を更新します。

このタスクについて

Oracle Label Security を使用するデータ・ソース・オブジェクトの場合は、ニックネーム・データをキャッシュできません。ALTER NICKNAME ステートメントを使用して、キャッシングのオン/オフを切り替えることができます。

制約事項: シノニムの Oracle シノニムにニックネームを作成することはサポートされておらず、SQL0204 エラー・メッセージがスローされる結果になります。

手順

Oracle 表またはビューのニックネームを登録するには、次の方法のいずれかを使用します。

方法	説明
Data Studio の「管理 エクスプローラー (Administration Explorer)」を使用しま す。	「フェデレーテッド・データベース・オブジェクト」フォルダ で、「ニックネーム」フォルダを右クリックし、「ニックネーム の作成」を選択します。
CREATE NICKNAME ステート メントを発行します。	例: CREATE NICKNAME <i>nickname</i> FOR <i>server_definition_name</i> ." <i>remote_schema</i> ". <i>"remote.table"</i> ;

ニックネームを作成すると、フェデレーテッド・サーバーはニックネームを使用してデータ・ソース・カタログを照会します。この照会は、データベース・ソースの表またはビューへの接続をテストします。接続ができない場合、エラー・メッセージが出ます。

ニックネームを作成する Oracle の表またはビューごとに、このステップを繰り返します。

CREATE NICKNAME ステートメント - Oracle ラッパーの例

CREATE NICKNAME ステートメントを使用して、アクセスする Oracle 表またはビューのニックネームを登録します。必須パラメーターを使用した完全指定の例がこのトピックには含まれています。

以下に示すのは、CREATE NICKNAME ステートメントを使用して、Oracle 表またはビューのニックネームを登録する方法の例です。

```
CREATE NICKNAME PARISINV FOR oraserver."vinnie"."inventory" ;
```

PARISINV

Oracle の表またはビューを識別するために使用する固有なニックネームです。ニックネームは、スキーマとニックネームとで構成されます。ニックネームの登録時にスキーマを省略すると、そのニックネームのスキーマはニックネームを登録するユーザーの許可 ID になります。

*oraserver."**vinnie*".*"inventory"*

リモート・オブジェクトを表す、3 つの部分からなる ID。

- *oraserver* は、CREATE SERVER ステートメントで Oracle データベース・サーバーに割り当てたサーバー定義名です。
- *vinnie* は、表またはビューが属する所有者のユーザー ID です。
- *inventory* は、アクセスするリモートの表またはビューの名前です。

フェデレーテッド・サーバーは、名前を引用符で囲んでいない限り、Oracle スキーマと表の名前を大文字に変換します。

データ・ソースとしてのスクリプトへのアクセスの構成

データ・ソースとしてのスクリプトにアクセスするようにフェデレーテッド・システムを構成するには、スクリプトのカスタム関数、ラッパー、サーバー定義、ニックネームを登録する必要があります。

始める前に

- フェデレーションが、フェデレーテッド・サーバーとして機能するサーバー上にインストールされている必要があります。
- フェデレーテッド・データベースはフェデレーテッド・サーバー上に存在する必要があります。

このタスクについて

DB2 コマンド行で SQL ステートメントを発行することで、スクリプト経由でデータにアクセスするようにフェデレーテッド・サーバーを構成できます。

手順

1. スクリプトを指定するか、または作成します。
2. カスタム関数を登録します。
3. スクリプト・デーモンを構成します。
4. スクリプト・デーモンを開始します。
5. スクリプト・ラッパーを登録します。
6. スクリプトによってアクセスするデータ・ソースのサーバー定義を登録します。
7. データ・ソースのニックネームを登録します。

スクリプト・ラッパーの概要

Perl スクリプトなどのスクリプトを使用すると、データベース内の情報にアクセスしたり、データを生成したりできます。スクリプト・ラッパーを使用することにより、そのデータを他のフェデレーテッド・データ・ソースのデータと統合することが可能になります。

ライフ・サイエンス・データ・バンクなどのデータ・ソースからデータを戻したり、独自にデータを生成したりする既存のスクリプトがあらかじめ用意されていることもあります。スクリプト・ラッパーは、スクリプトをあたかもフェデレーテッド・データ・ソースであるかのように使用できるようにします。そのようにしてスクリプト・ラッパーによりデータへのアクセスが可能になり、データはフェデレーテッド・システムを使用して統合されます。スクリプトから戻される結果は XML でなければなりません。

スクリプト・ラッパーのニックネームには、入出力列を含むことができます。これらのニックネームでは、入力値をスクリプトに渡すために述部で関数テンプレートを使用します。スクリプトからの出力データは階層構造の XML で表現され、主キーおよび外部キーを使用してニックネームにマップできます。

スクリプト・ラッパーは、読み取り専用ラッパーです。スクリプト・ラッパーは、データ・ソースにデータを書き込むことができません。

次の図では、データがスクリプトからスクリプト・ラッパーおよびスクリプト・デーモンを経由してフェデレーテッド・データベースに流れます。そこで、このデータは他のソースからのデータと統合され、フェデレーテッド・クライアントで表示できるようになります。オプションで、フェデレーテッド・システムはプロキシ・サーバーおよびファイアウォール経由でスクリプトにアクセスできます。

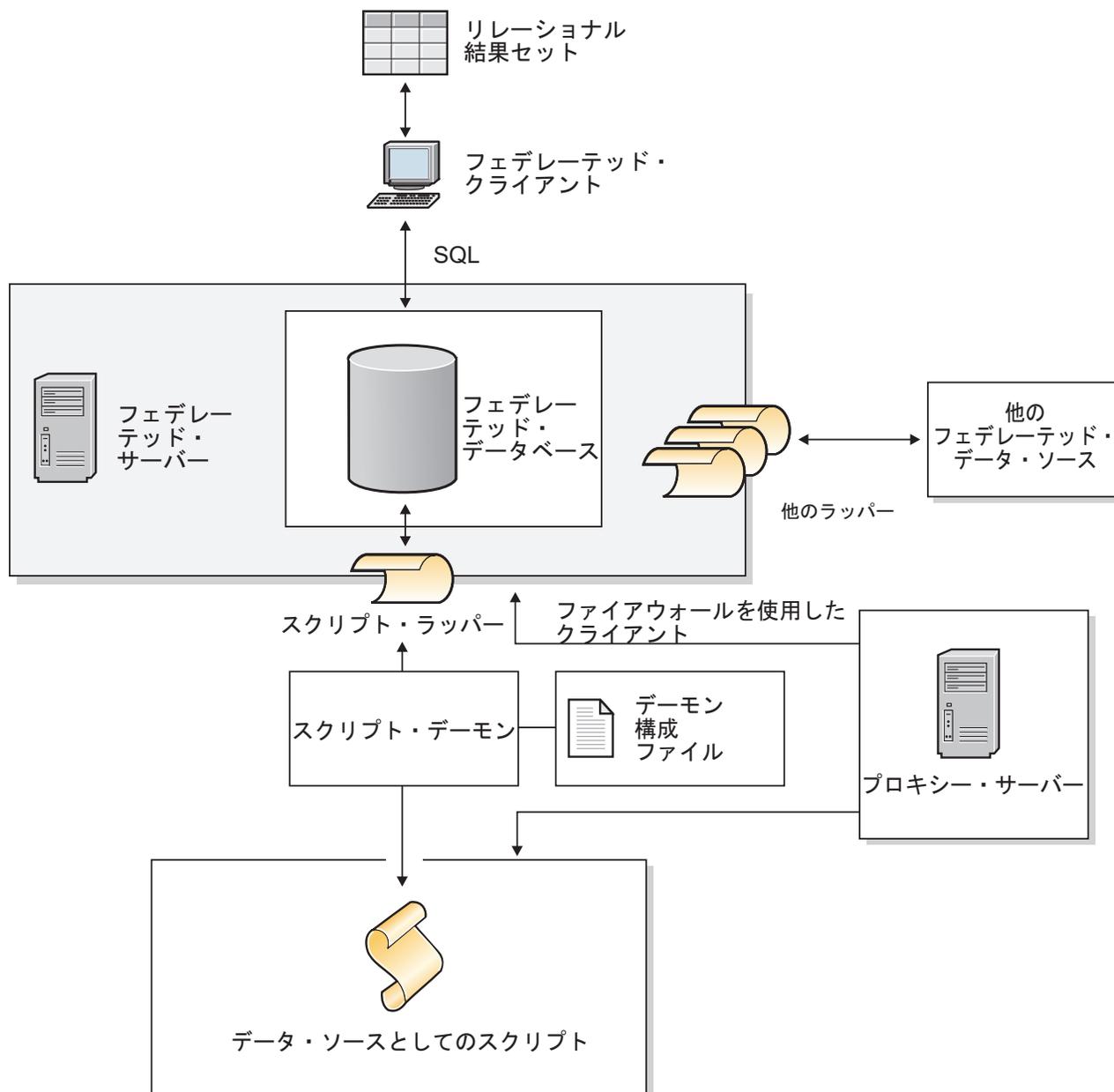


図 10. フェデレーテッド・システム内のスクリプト・ラッパー

スクリプトは、スクリプト・デーモンを含むディレクトリーから呼び出されます。スクリプト・データ・ファイルからデータを取り出そうとして、スクリプトがデータ・ファイルを見つけることができない場合は、スクリプトに相対パスが含まれている可能性があります。スクリプトには絶対パスを使用するようにします。

データ・ソースとしてのスクリプトのフェデレーテッド・システムへの追加

データ・ソースとしてのスクリプトにアクセスするフェデレーテッド・サーバーを構成するには、スクリプトのスクリプト・デーモン、ラッパー、サーバー定義、ニックネームを構成する必要があります。

スクリプトのカスタム関数の登録

スクリプト・ラッパーを登録する前に、スクリプト・カスタム関数 WSSCRIPT.ARGS を登録する必要があります。

このタスクについて

スクリプト・ラッパーがインストールされている各フェデレーテッド・データベース・インスタンスにカスタム関数を登録する必要があります。

スクリプト・ラッパーのカスタム関数は、WSSCRIPT というスキーマ名を使用して登録する必要があります。

スクリプト・ラッパーのカスタム関数を登録するときは、特定のキーワードを含める必要があります。CREATE FUNCTION ステートメントには、AS TEMPLATE、DETERMINISTIC、NO EXTERNAL ACTION キーワードを含めます。

フェデレーテッド環境は 2 つの照会エンジンを使用します。スクリプト・ラッパーの場合、フェデレーテッド・データベース照会エンジンとスクリプト・ラッパー照会エンジンがそれに当たります。SELECT ステートメントの WHERE 文節にスクリプト・ラッパー・カスタム関数を使用して、述部をスクリプト・ラッパー・エンジンに限定することができます。

フェデレーテッド・サーバー上の `sqllib/samples/lifesci/script` ディレクトリーの `create_function_mappings.ddl` ファイルは、カスタム関数のデータ・タイプを指定します。

手順

スクリプト・ラッパーがインストールされている各フェデレーテッド・データベース・インスタンス上で、`create_function_mappings.ddl` ファイルを実行します。

タスクの結果

WSSCRIPT.ARGS 関数の構文例は以下のとおりです。

```
CREATE FUNCTION WSSCRIPT.ARGS (input_column_data_type(), input_column_data_type())  
  RETURNS INTEGER AS TEMPLATE  
  DETERMINISTIC NO EXTERNAL ACTION;
```

スクリプト・ラッパーのカスタム関数のデータ・タイプ:

各カスタム関数には、入力列のデータ・タイプを 2 回指定する必要があります。

WSSCRIPT カスタム関数を、`column_name` 引数のそれぞれの有効なデータ・タイプごとに個別に登録します。WSSCRIPT カスタム関数には、以下のデータ・タイプがあります。

- VARCHAR
- INTEGER
- CLOB
- DOUBLE
- DATE

カスタム関数の 2 つのパラメーターは同じデータ・タイプで、しかも対応する入力列のデータ・タイプでなければなりません。照会述部で使用された場合、最初のパラメーターは、スイッチ入力列の名前になります。2 番目の列は、このスイッチに対応するスクリプトに渡される値になります。

スクリプト・デーモンの構成

スクリプト・ラッパーには、スクリプト・ラッパーからのスクリプト・ジョブ要求を listen するスクリプト・デーモンが必要です。スクリプト・ラッパーを登録する前に、スクリプト・デーモンを構成することが必要です。

始める前に

スクリプト・デーモンには、以下の前提条件があります。

- そのデーモンが一時ファイルを書き込むディレクトリーに対する書き込みアクセス権限を持っている。
- フェデレーテッド・システムから TCP/IP を介してアクセス可能なサーバー上で実行する。このサーバーは、フェデレーテッド・サーバーとして機能するサーバーと同じであっても、別のスクリプト・サーバーであってもかまいません。
- 構成ファイルは必ずスクリプト・デーモンと同じサーバー上になければならない。
-
- スクリプト・ラッパーおよびフェデレーテッド・データベースとは別個に実行する。

手順

1. スクリプト・デーモンの実行可能ファイルが正しいサーバー上にあることを確認します。スクリプト・デーモンの実行可能ファイルを別のサーバーにコピーすることが必要になる場合もあります。

IBM InfoSphere Federation Serverのインストール中に、スクリプト・デーモンの実行可能ファイルはフェデレーテッド・サーバーにインストールされます。ファイルの名前とロケーションは、以下のとおりです。

UNIX db2script_daemon は \$DB2PATH/bin ディレクトリーにインストールされます。\$DB2PATH は、フェデレーテッド・サーバーがインストールされているディレクトリーです。

Windows

db2script_daemon.exe は %DB2PATH%\bin ディレクトリーにインストールされます。%DB2PATH% は、フェデレーテッド・サーバーがインストールされているディレクトリー (通常は C:\SQLLIB\bin) です。

独立したスクリプト・サーバーを使用する場合は、スクリプト・デーモンの実行可能ファイルと構成ファイルをフェデレーテッド・サーバーからスクリプト・サーバーにコピーします。スクリプト・デーモン実行可能ファイルは、ディレクトリー・パスの名前にスペースが含まれないスクリプト・サーバー上の任意のディレクトリーで実行することができます。

2. スクリプト・デーモンの構成ファイルが正しいサーバー上にあることを確認します。

フェデレーテッド・システムのインストール中に、スクリプト・デーモンのサンプル構成ファイルがフェデレーテッド・サーバーにインストールされます。サンプル構成ファイルの名前は `SCRIPT_DAEMON.config` です。ファイルのロケーションは以下のとおりです。

UNIX デーモン構成ファイルは、`$DB2PATH/bin` ディレクトリーにインストールされます。

Windows

デーモン構成ファイルは、`%DB2PATH%\bin` ディレクトリーにインストールされます。

デフォルトでは、デーモンは、デーモンが開始される作業ディレクトリー内で構成ファイルを検索します。構成ファイルは、別のロケーションにコピーすることができます。スクリプト・サーバーを使用する場合、デーモン構成ファイルを、フェデレーテッド・サーバー上のディレクトリーからスクリプト・サーバー上のディレクトリーにコピーします。デーモン構成ファイルは、デーモンがアクセスできるスクリプト・サーバー上の任意のディレクトリーにコピーすることができます。

3. スクリプト・デーモンのサンプル構成ファイルを編集します。
 - a. サンプル・ファイルを再利用できるように、構成ファイルをリネームします。
 - b. 構成ファイルの 1 行目が等号 (=) であることを確認します。等号がないと、デーモンは開始しません。 `DAEMON_PORT` が指定されなかったことを示すエラー・メッセージが出されます。
 - c. 構成ファイルの最終行が改行で終了していることを確認します。

フェデレーテッド・システムに付属のサンプル構成ファイルは、改行文字で終了しています。最終行が改行文字で終わっていない場合、最終行にリストされたデータ・ソースを使用してスクリプト照会を初めて実行しようとする際に、エラー・メッセージが出されます。

- d. ディレクトリー・パスの後または構成ファイルの最後に余分のスペースがないことを確認します。
 - e. 構成ファイルに以下のオプションを指定します。パスを必要とするオプションについては、相対パスを指定できます。相対パスは、デーモン・プロセスが開始されたディレクトリーに相対しています。

DAEMON_PORT=port_number

ラッパーによってサブミットされたスクリプト・ジョブ要求をデーモンが `listen` するネットワーク・ポート。デフォルト値は 4099 です。

MAX_PENDING_REQUESTS=number_of_requests

任意の一時点でデーモンでブロックできるスクリプト・ジョブ要求の最大数です。この数は、並行して実行されているスクリプト・ジョブ数を表すものではありません。一時点でブロックできるジョブ要求数のみです。この値を 5 より大きい数に設定します。スクリプト・デーモンでは、並行して実行できるスクリプト・ジョブ数の制限はありません。

DAEMON_LOGFILE_DIR=*dir*

デーモンがそのログ・ファイルを作成するディレクトリーです。このファイルには、スクリプト・デーモンによって生成された状況およびエラー情報が含まれます。

SCRIPT_OUT_DIR_PATH=*path*

デーモンがスクリプト出力データを保管するための一時ファイルを作成するディレクトリーです。デーモンはこのファイルからデータを読み取り、ネットワーク接続を介してデータをラッパーに戻します。データがラッパーに渡されると、デーモンは一時ファイルをクリーンアップします。

script specification entry=*entry*

スクリプト・ラッパーによる呼び出しが可能なスクリプトの名前とケーションを指定する項目のリスト。項目のフォーマットは以下のようになります。

script_name=fully-qualified_script_path

以下の例は、指定されたオペレーティング・システムに適用されません。

UNIX たとえば、Oracle データ・ソースにアクセスするスクリプトを指定するには、次の行をデーモン構成ファイルに追加します。

```
oracle=/dsk/1/data/oracle
```

Windows

たとえば、Oracle データ・ソースにアクセスするスクリプトを指定するには、次の行をデーモン構成ファイルに追加します。

```
oracle=c:%data%oracle.a
```

例

以下に、4 つのスクリプトの SCRIPT_DAEMON.cfg ファイルの例を示します。

```
=  
DAEMON_PORT=4099  
MAX_PENDING_REQUESTS=10  
DAEMON_LOGFILE_DIR=./  
SCRIPT_OUT_DIR_PATH=./  
fee=/home/user_id/fee  
fie=/home/user_id/fie  
foe=/home/user_id/foe  
fum=/home/user_id/fum
```

スクリプト・デーモンのサンプル構成ファイルは、スクリプト・デーモンの構成の例となります。

スクリプト・デーモンの開始

スクリプトを使用してデータ・ソースにアクセスするためには、スクリプト・デーモンを開始しなければなりません。

始める前に

デーモン構成ファイルで `DAEMON_LOGFILE_DIR` と `SCRIPT_OUT_DIR_PATH` オプションについてリストされているすべてのパスに対する書き込みアクセス権限が必要です。

このタスクについて

スクリプト・デーモンの実行可能ファイルは、スクリプト・デーモンが実行される新規プロセスを開始します。

手順

1. デーモン実行可能ファイルが存在するディレクトリーを開きます。
2. `db2script_daemon` コマンドに適切なオプションを指定して発行し、実行可能ファイルを実行します。

例

以下は、オプションが指定されている例です。

```
db2script_daemon -a action -c config_file -d debug_level -u user_id -p password
```

`db2script_daemon` コマンド - オプションと例:

`db2script_daemon` コマンドは、スクリプト・デーモンを開始します。スクリプト・デーモンは、いくつかのオプションを指定して開始することができます。

`db2script_daemon` コマンドは、UNIX または Windows サーバーで使用できます。構文内でリストされるオプションの一部は、Windows サーバーでのみ使用できません。

`start` アクションで指定されるオプションは、現在のデーモンのインスタンスのみに影響し、`install` アクションを使用して指定される値をオーバーライドします。

オプション - `db2script_daemon` コマンド

`db2script_daemon` コマンドには、以下のオプションがあります。

-a action (Windows のみ)

指定したアクティビティーを実行します。有効なアクションは、`status`、`install`、`start`、`stop`、および `remove` です。

-c config_file

指定した構成ファイルを使用するように、デーモン・サービスに指示します。構成ファイルを指定しなかった場合、デーモンは、デーモン実行可能ファイルがインストールされているディレクトリー内で `SCRIPT_DAEMON.config` ファイルを検索します。このオプションは、`install` および `start` アクションとともに使用できます。

-d [1 | 2 | 3]

デーモン・サービスのデバッグ・レベルを、指定された値に設定します。値 1 を指定するとロギングはオンになり、2 を指定するとすべてのコマンドがトレースされます。3 を指定するとロギングがオンになり、すべてのコマン

ドがトレースされ、XML 出力をキャプチャーしたすべての一時ファイルが保管されます。このオプションは、install および start アクションとともに使用できます。

-u user_id (Windows のみ)

指定したユーザー ID で実行するように、デーモン・サービスを設定します。このオプションは、install アクションとともに使用できます。

-p password (Windows のみ)

指定したユーザー ID のパスワードを指定します。パスワードが有効かつ必要なのは、-u オプションを指定する場合だけです。-u オプションの設定時に -p オプションを指定しなかった場合、パスワードを求めるプロンプトがプログラムによって出されます。このオプションは、install アクションとともに使用できます。

例 - db2script_daemon コマンド

以下の例は、スクリプト・デーモン・オプションの使用法を示したものです。

デーモンの開始

UNIX でデーモンを開始するには、次のコマンドを発行します。

```
db2script_daemon
```

このコマンドは、デーモン構成ファイルが実行可能ファイルと同じディレクトリに存在するという前提で機能します。

Windows でデーモンをインストールして開始するには、次のコマンドを発行します。

```
db2script_daemon -a install
db2script_daemon -a start
```

デーモン構成ファイルを指定します。

デーモン構成ファイルの名前を変更してしまっているか、構成ファイルがデーモン実行可能ファイルと同じディレクトリにない場合、実行可能ファイルの実行時に -c オプションを使用する必要があります。このオプションは、デーモン構成ファイルのディレクトリー・パスと名前を指定します。

この例では、デーモン構成情報が UNIX サーバー上のサブディレクトリー cfg の SCRIPT_D.config というファイルにあります。次のコマンドを発行する。

```
db2script_daemon -c cfg/SCRIPT_D.config
```

デバッグ・レベルを指定します。

デバッグ・レベル 2 でデバッグをオンにしてデーモンを開始する場合は、以下のコマンドを発行します。

```
db2blast_daemon -a install -d 2
db2blast_daemon -a start
```

デーモンの状況をチェックする (Windows)

Windows サーバーでデーモンの状況をチェックするには、次のコマンドを発行します。

```
db2blast_daemon -a status
```

このコマンドは、デーモン構成ファイルが実行可能ファイルと同じディレクトリに存在するという前提で機能します。

デーモンの停止

UNIX 上のデーモンを停止するには、以下のコマンドを発行して、プロセス ID のリストを作成します。

```
ps -ef | grep db2script
```

次に、プロセス ID を使用して以下のコマンドを発行し、デーモンを停止します。

```
kill process_ID
```

このコマンドは、デーモン構成ファイルが実行可能ファイルと同じディレクトリに存在するという前提で機能します。

Windows でデーモンを停止するには、次のコマンドを発行します。

```
db2script_daemon -a stop
```

デーモンの削除

今後スクリプト・ラッパーを使用しないという場合には、スクリプト・デーモンを削除することもできます。

デーモンを削除するには、以下のコマンドを発行します。

```
db2script_daemon -a remove
```

スクリプト・ラッパーの登録

スクリプトを使用してデータ・ソースにアクセスするには、スクリプト・ラッパーを登録する必要があります。スクリプト・ラッパーはライブラリー・ファイルとしてインプリメントされます。

手順

スクリプト・ラッパーの名前とラッパー・ライブラリー・ファイルの名前を指定して、CREATE WRAPPER ステートメントを発行します。

たとえば、AIX オペレーティング・システムを使用するフェデレーテッド・サーバー上で script_wrapper という名前のラッパーを登録するには、次のステートメントを発行します。

```
CREATE WRAPPER script_wrapper LIBRARY 'libdb2lscript.a';
```

指定するラッパー・ライブラリー・ファイルの名前は、フェデレーテッド・サーバーのオペレーティング・システムによって異なります。

スクリプト・ラッパー CREATE WRAPPER ステートメントにはスクリプト・ラッパー固有のオプションはありません。デフォルトでは、ラッパーは unfenced で実行されます。

スクリプト・ラッパー・ライブラリー・ファイル:

スクリプト・ラッパーを登録するには、フェデレーテッド・サーバーのオペレーティング・システム用のスクリプト・ラッパー・ライブラリー・ファイルを指定します。

フェデレーションのインストール時に、スクリプト・ラッパー・ライブラリー・ファイルがデフォルトのディレクトリー・パスに追加されます。

以下の表に、デフォルトのディレクトリー・パスとデフォルトのラッパー・ライブラリー・ファイルの名前がリストされています。

表 34. スクリプト・ラッパー・ライブラリー・ロケーションおよびファイル名

オペレーティング・システム	ディレクトリー・パス	ラッパー・ライブラリー・ファイル名
AIX	<code>/usr/opt/install_path/lib</code>	<code>libdb2lsscript.a</code>
Linux	<code>/opt/IBM/db2/install_path/lib</code>	<code>libdb2lsscript.so</code>
Solaris	<code>/opt/IBM/db2/install_path/lib</code>	<code>libdb2lsscript.so</code>
Windows	<code>%DB2PATH%\bin</code>	<code>db2lsscript.dll</code>

- `install_path` は、UNIX または Linux 上でフェデレーションがインストールされているディレクトリーのパスです。
- `%DB2PATH%` は、Windows 上でフェデレーションがインストールされているディレクトリー・パスを指定するのに使用する環境変数です。デフォルトの Windows ディレクトリー・パスは、`C:\Program Files\IBM\SQLLIB` です。

データ・ソースとしてのスクリプトのサーバー定義の登録

アクセスする各サーバーをフェデレーテッド・データベースに登録する必要があります。

手順

CREATE SERVER ステートメントを発行します。

例:

```
CREATE SERVER script_server WRAPPER script_wrapper
  OPTIONS (NODE 'myserver.example.com', DAEMON_PORT '4099');
```

データ・ソースとしてのスクリプトには、NODE および DAEMON_PORT サーバー・オプションが必要です。

サーバー定義を登録した後、サーバー・オプションを追加またはドロップするには、ALTER SERVER ステートメントを使用します。

CREATE SERVER ステートメント - スクリプト・ラッパーの例:

必須指定のオプションと追加サーバー・オプションの使用法を示す例です。

必須指定のオプションの例

以下に示すのは、CREATE SERVER ステートメントを発行して、スクリプト・ラッパーのサーバー定義を登録する方法の例です。

```
CREATE SERVER server1_scriptn WRAPPER script_wrapper
  OPTIONS (NODE 'big_rs.company.com');
```

`server1_scriptn`

スクリプト・サーバーに割り当てる名前。重複するサーバー定義名は使用できません。

WRAPPER *script_wrapper*

ラッパー名。

NODE *'big_rs.company.com'*

スクリプト・デーモン・プロセスが実行されるシステムのホスト名。この値は大文字と小文字を区別します。

ノード名は CREATE SERVER ステートメントではオプションとなっていますが、スクリプト・ラッパーの場合は必要です。

任意指定のオプションの例

以下の例には、スクリプト・ラッパーのサーバー定義を登録するときに指定できる追加のサーバー・オプションが示されています。

```
CREATE SERVER server1_scriptn WRAPPER script_wrapper  
  OPTIONS(NODE 'big_rs.company.com', DAEMON_PORT '4088');
```

DAEMON_PORT *'4088'*

デーモンがスクリプト・ジョブ要求を listen するポートの番号を指定します。ポート番号は、デーモン構成ファイルの DAEMON_PORT オプションに指定されているものと同じ番号でなければなりません。デフォルトのポート番号は 4099 です。

スクリプトのニックネームの登録

スクリプトごとに別個のニックネームを登録しなければなりません。これらのニックネームは、スクリプトによってアクセスするデータ・ソースを照会するときに使用します。

このタスクについて

スクリプト・ラッパーは XML データをニックネームに関連付けます。親ニックネームと子ニックネームは、XML 文書のルートおよびネスト・エレメントに対応します。親ニックネームと子ニックネームは、CREATE NICKNAME ステートメントで指定される主キーおよび外部キーによって結び付けられます。各ニックネームは、XML データ・エレメントの識別と各エレメントから列値を抽出する方法の指定とを行う XPath 式によって定義されます。

データ・ソースは CREATE NICKNAME ステートメントによって指定され、DATASOURCE ニックネーム・オプションによってスクリプト名と関連付けられます。スクリプトに渡される入力引数ごとに 1 つの列を作成する必要があります。コマンド行のスイッチの構文を制御するには、入力列オプションを使用します。各スイッチの値は、実行時に ARGS カスタム関数を使用して照会述部を含める必要があります。

コマンド行引数を必要としない簡単なスクリプトには、入力列は必要ありません。

手順

コマンド行から CREATE NICKNAME ステートメントを発行します。

例:

```

CREATE NICKNAME nickname
(
  column_name data_type OPTIONS ('nickname_column_options'),
  column_name data_type OPTIONS ('nickname_column_options'),
  column_name data_type OPTIONS ('nickname_column_options')
)
FOR SERVER server_definition_name
OPTIONS (nickname_options);

```

ニックネームの長さは 128 文字までです。

次のタスク

ニックネームを作成する スクリプトごとに、このステートメントを発行します。

CREATE NICKNAME ステートメント - スクリプト・ラッパーの例

例では、CREATE NICKNAME ステートメントを使用してスクリプト・ラッパーのニックネームを登録する方法を例示します。

以下の例では、*fee* という名前のスクリプトから戻される XML データの親ニックネームを作成します。

```

CREATE NICKNAME customers
(
  argle double OPTIONS(SWITCH '-argle', POSITION 1, DEFAULT 1.0 ),
  argfile CLOB() OPTIONS(SWITCH '-file', INPUT_MODE 'FILE_INPUT', POSITION 2),
  argpos varchar() OPTIONS(SWITCH ' ', POSITION 3),
  id VARCHAR(5) OPTIONS(XPATH './@id'),
  name VARCHAR(16) OPTIONS(XPATH './name'),
  address VARCHAR(30) OPTIONS(XPATH './address/@street'),
  cid VARCHAR(16) FOR BIT DATA NOT NULL OPTIONS(PRIMARY_KEY 'YES'))
FOR SERVER script_server
OPTIONS(DATASOURCE 'fee',
        XPATH '/doc/customer', STREAMING 'YES');

```

以下の例では、*orders* というニックネームを作成します。ニックネーム *orders* は、前の例で作成した *customers* というニックネームの子です。

```

CREATE NICKNAME orders
(
  amount INTEGER OPTIONS(XPATH './amount'),
  date VARCHAR(10) OPTIONS(XPATH './date'),
  oid VARCHAR(16) OPTIONS(PRIMARY_KEY 'YES'),
  cid VARCHAR(16) FOR BIT DATA NOT NULL OPTIONS(FOREIGN_KEY 'CUSTOMERS'))
FOR SERVER script_server
OPTIONS( XPATH './order');

```

スクリプト・ラッパー・ニックネーム・オプション

スクリプトのニックネームを作成するときには、オプションを指定できます。ルート・ニックネームに限って、入力列を含めることができます。

ニックネーム・オプション

以下のリストは、ニックネーム・オプションの説明です。

DATASOURCE

呼び出されるスクリプトの名前です。スクリプトは、スクリプト・デーモン構成ファイル内にリストされている必要があります。親ニックネームのデータ・ソースには、このオプションが必要です。このオプションは、ルート・ニックネームにのみ適用されます。

NAMESPACES

ニックネーム XPath 式の名前空間接頭部の解決にラッパーが使用する、コンマで区切られた名前 = 値のペアのリストです。

TIMEOUT

スクリプト・ラッパーがスクリプト・デーモンからの結果を待つ最大時間 (分単位)。デフォルト値は 60 分です。このオプションは、ルート・ニックネームにのみ適用されます。

VALIDATE

XML データを抽出する前に XML ソース文書の妥当性検査を行うかどうかを指定します。このオプションを YES に設定すると、DB2 データベース・システムは、ソース文書の構造が XML スキーマまたは文書タイプ定義 (DTD) に準拠しているかどうかを検査します。このオプションは、ルート・ニックネーム (XML 文書の最上位にあるエレメントを識別するニックネーム) の列に対してのみ受け入れられます。デフォルト値は NO です。

スクリプト・ラッパーが XML スキーマ・ファイルまたは DTD ファイル (.xsd または .dtd) を見付けることができない場合、XML ソース文書の妥当性検査は行われません。妥当性検査が行われなくても、DB2 データベース・システムはエラー・メッセージを出しません。XML スキーマ・ファイルまたは DTD ファイルが XML ソース文書で指定されるロケーションにあることを確認してください。STREAMING パラメーターを YES に設定する場合、VALIDATE パラメーターを YES に設定しないでください。

STREAMING

XML ソース・ドキュメントをニックネームの XPath 式に一致するノードに対応する論理フラグメントに分けるかどうかを指定します。スクリプト・ラッパーは、XML ソース・データの処理をフラグメントごとに行い、メモリーの使用合計量を減らします。このオプションは、ルート・ニックネーム (XML 文書の最上位にあるエレメントを識別するニックネーム) の列に対してのみ受け入れられます。デフォルトのストリーミング値は NO です。VALIDATE パラメーターを YES に設定する場合、STREAMING パラメーターを YES に設定しないでください。

XPATH

データ・ソース内の個々のタブルを表す XML エレメントを識別する XPath 式です。ラッパーは、子ニックネームの XPATH ニックネーム・オプションを、親ニックネームの XPATH ニックネーム・オプションによって指定されるパスのコンテキストで評価します。この XPath 式は、XPATH ニックネーム列オプションによって識別される列値を評価するためのコンテキストとして使用されません。

ニックネーム列オプション

ニックネーム列オプションは次のように記述されます。

DEFAULT

入力列のデフォルト値。このオプションは、入力列にのみ適用されます。

SQL 照会によって値が提供されない場合、デフォルト値が使用されます。このオプションは必須ではありません。

FOREIGN_KEY

このニックネームが子ニックネームであることを示し、対応する親ニックネームを指定します。

ニックネームには、FOREIGN_KEY 列オプションを 1 つしか指定できません。このオプションの値は、大文字小文字を区別します。FOREIGN_KEY オプションによって指定された列は、ラッパーによって生成されたキーを含みます。列の値は、SELECT 照会では検索できません。XPath オプションを指定してはなりません。列は、親のニックネームと子のニックネームを結合するためだけに使用できます。親ニックネームに別のスキーマ名が付けられている場合、FOREIGN_KEY オプションを指定した CREATE NICKNAME ステートメントは失敗します。FOREIGN_KEY 文節で参照されるニックネームが、対応する CREATE NICKNAME ステートメントのもとで引用符で囲むことによって、小文字または大文字小文字の混合として明示的に定義されていないかぎり、このニックネームを FOREIGN_KEY 文節で参照する場合、ニックネームを大文字で指定する必要があります。

外部キー列は FOR BIT DATA および NOT NULL として指定する必要があります。

INPUT_MODE

列の入力モードを指定します。有効な値は CONFIG または FILE_INPUT です。ラッパーは、指定された値をスクリプト・デーモンに渡します。

CONFIG

値は構成可能なパラメーターとして扱われます。

FILE_INPUT

値を保管するファイルが作成され、ファイル名がコマンド行引数として渡されます。

POSITION

位置パラメーターの整数値です。位置を表す値は 1 から始まります。このオプションは入力列のみに適用されます。

位置の値が整数に設定された場合、入力はコマンド行中のその数値に対応する位置に置かれなければなりません。このオプションが設定されていると、照会実行時にスイッチが適切な位置に挿入されます。POSITION が -1 に設定されている場合は、オプションが最後のコマンド行オプションとして追加されます。たとえば、列値がコマンド行の最後になければならず、かつ SWITCH オプションが指定されていない場合に、POSITION の値を -1 に設定すると、コマンド行の最後に値が追加されます。POSITION 整数値は、ニックネーム中で重複はできません。このオプションは必須ではありません。

PRIMARY_KEY

このニックネームが親ニックネームであることを示します。列データ・タイプは VARCHAR(16) でなければなりません。ニックネームには、PRIMARY_KEY 列オプションを 1 つしか指定できません。有効値は Y のみです。

PRIMARY_KEY として指定された列は、ラッパーによって生成されたキーを含みます。列の値は、SELECT 照会では検索できません。XPath オプションを指定してはなりません。列は、親のニックネームと子のニックネームを結合するためだけに使用できます。主キー列は FOR BIT DATA および NOT NULL として指定する必要があります。

SWITCH

コマンド行でスクリプトのパラメーターを指定する文字ストリング。このオプションは、入力列にのみ適用されます。

コマンド行で、このオプションの値は、WSSCRIPT.ARGS によって提供される列値、またはデフォルト値 (もしあれば) よりも先行します。スイッチの値が空ストリングで、列のデフォルト値が存在する場合、デフォルト値はコマンド行生成時に SWITCH 情報なしで追加されます。デフォルト値が提供されておらず、列の値も SQL 照会によって提供されない場合、この入力列はコマンド行生成時に無視されます。このオプションは、入力列に必要です。

SWITCH_ONLY

コマンド行引数なしでスイッチの使用を可能にします。

SWITCH_ONLY オプションが値 Y とともに指定された場合、有効な入力値は Y か N になります。入力値が Y なら、コマンド行にはスイッチのみが追加されます。入力値が N の場合、コマンド行に値は追加されません。

VALID_VALUES

セミコロンで区切られる、列の有効値のセットです。

XPATH

XML 文書の中でこの列に対応するデータを含む XPath 式を指定します。

CREATE NICKNAME ステートメントがこの XPATH ニックネーム・オプションから XPath 式を適用した後で、スクリプト・ラッパーはその XPath 式を評価します。大文字小文字の区別が不正確であるなどの、XPATH タグ参照が不正確に構成されている列名に対して照会を実行する場合、照会は必要なすべての列に対してその列にある NULL 値を戻します。

スクリプト・ラッパーによる SQL 照会

スクリプト・ラッパーを介して作成される SQL 照会では、スクリプトのパラメーター入力値を受け渡すのにカスタム関数を使用します。

スクリプト・ラッパー経由でスクリプトにパラメーター値を渡す SELECT ステートメントには、スクリプトのパラメーター入力値を受け取るカスタム関数を持つ述部が少なくとも 1 つ含まれている必要があります。

ルート・ニックネーム

たとえば、以下のステートメントでは、myscript という名前のスクリプトのルート・ニックネームが作成されます。

```
CREATE NICKNAME customers (  
  argle double OPTIONS(SWITCH '-argle', POSITION 1, DEFAULT 1.0),  
  argfile CLOB() OPTIONS(SWITCH '-file', INPUT_MODE 'FILE_INPUT', POSITION 2),  
  argpos varchar() OPTIONS(SWITCH ' ', POSITION 3),  
  id varchar(10) OPTIONS(XPATH './@id'),  
  name varchar OPTIONS(XPATH '/name'))  
FOR SERVER script_server  
  OPTIONS(DATASOURCE 'myscript', XPATH 'doc/customer',  
    TIMEOUT '300', VALIDATE 'YES');
```

カスタム関数ステートメントは以下のようになります。

```
CREATE FUNCTION wsscript.args (varchar(), varchar())  
  RETURNS INTEGER AS TEMPLATE  
  DETERMINISTIC NO EXTERNAL ACTION;
```

```

CREATE FUNCTION wsscript.args (date(), date())
  RETURNS INTEGER AS TEMPLATE
  DETERMINISTIC NO EXTERNAL ACTION;

CREATE FUNCTION wsscript.args (integer(), integer())
  RETURNS INTEGER AS TEMPLATE
  DETERMINISTIC NO EXTERNAL ACTION;

CREATE FUNCTION wsscript.args (CLOB(), CLOB())
  RETURNS INTEGER AS TEMPLATE
  DETERMINISTIC NO EXTERNAL ACTION;

CREATE FUNCTION wsscript.args (double(), double)
  RETURNS INTEGER AS TEMPLATE
  DETERMINISTIC NO EXTERNAL ACTION;

```

構成ファイルでは、以下の構成パラメーターを指定します。

```

SCRIPT_OUT_DIR_PATH=C:%temp
myscript=C:%perl%bin%perl myscript.pl -model

```

照会を実行して、表 `t1.bigdata` の内容をデーモンおよびローカル・ファイル `C:%temp%f12345` に送信するには、以下の照会および `wsscript.args` コマンドを発行します。

```

SELECT id, name FROM customers, t1
  WHERE wsscript.args (customers.argfile, t1.bigdata) = 1

```

上の照会の結果、以下のようなコマンド行ができます。

```

C:%perl%bin%perl myscript.pl -model -argle 1.0 -file C:%temp%f12345

```

すべてのパラメーターについてデフォルト値を使用する照会を実行するには、ニックネームに対して述部なしで照会を実行します。出力が過剰になるという問題を防ぐには、ニックネームに `STREAMING` オプションを含めます。

子ニックネーム

スクリプト・ラッパーは、スクリプトからの XML 結果セットを、親子関係を持つニックネームにマップします。子ニックネームからデータを取得するには、子ニックネームを、ルートにいたるまで親ニックネームに結合します。子ニックネームを参照する `SELECT` ステートメントは、主キー列および外部キー列を使用することで、その子ニックネームの親ニックネームと結合する必要があります。

以下の照会は、各カスタマーの各オーダーごとのカスタマー名を表示します。

```

SELECT c.name, o.amount FROM customers c, orders o
  WHERE c.cid=o.cid
  AND wsscript.args (customers.argfile, t1.bigdata) = 1
  AND wsscript.args (customers.argpos, VARCHAR('test1')) = 1
  AND wsscript.args (customers.argle, FLOAT('3.5')) = 1

```

`customers` ニックネームおよび `orders` ニックネームの間の親子関係を示すには、結合 `c.cid=o.cid` ステートメントを指定します。子ニックネームをそれ自身に結合すると、エラー・メッセージが戻されます。

スクリプト・ラッパー・パフォーマンスの最適化

スクリプト・デーモンの位置は、照会のパフォーマンスに影響します。

ネットワーク通信のパフォーマンスを向上するには、スクリプト・デーモン用に別個のスクリプト・サーバーを使用します。フェデレーテッド・サーバーとスクリプト・サーバーを別々のサーバーに置きます。また、スクリプト・デーモンはスクリプト・サーバーに置くようにします。

Sybase データ・ソースへのアクセスの構成

フェデレーテッド・サーバーが Sybase データ・ソースにアクセスするように構成するには、アクセスしたいデータ・ソースとオブジェクトに関する情報をフェデレーテッド・サーバーに提供する必要があります。

始める前に

- Sybase クライアント・ソフトウェアが、フェデレーテッド・サーバーとして機能するサーバー上にインストールされている必要があります。Sybase クライアントを Windows 上にインストールする場合、「フル」または「カスタム」オプションを指定する必要があります。カスタム・オプションを指定する場合は、「**XA Interface Library for ASE Distributed Transaction Manager**」オプションを指定する必要があります。
- IBM InfoSphere Federation Serverが、フェデレーテッド・サーバーとして機能するサーバー上にインストールされている必要があります。
- フェデレーテッド・サーバーのセットアップを確認します。
- フェデレーテッド・パラメーターをチェックして、フェデレーションが有効になっていることを確認します。

手順

1. Sybase 環境変数を設定します。
2. ご使用のオペレーティング・システムに応じて以下の方法のいずれかを使用し、Sybase クライアント構成ファイルをセットアップしてテストします。
 - Sybase クライアント構成ファイルをセットアップしてテストします (Linux、UNIX)。
 - クライアント構成ファイルをセットアップしてテストします (Windows)。
3. ラッパーを登録します。
4. サーバー定義を登録します。
5. ユーザー・マッピングを作成します。
6. Sybase サーバーとの接続をテストします。
7. Sybase の表およびビューのニックネームを登録します。

Adaptive Server Enterprise (ASE) 用の Sybase ラッパー・サポート

Sybase ラッパーは、ASE 12.5 および ASE 12.0 に加え、Sybase Adaptive Server Enterprise (ASE) 15.0 をサポートします。

Sybase ASE 15.0 用にサポートされるクライアント

ASE 15.0 への接続は、Sybase ラッパーを Sybase クライアント、バージョン 12.5.1 以降で使用して行うことができます。

Software Developer Kit (SDK) バージョン 12.5.1 を Sybase クライアントとして使用する場合は、SDK の Electronic Software Distribution (ESD) #12 以降を、Sybase ラッパーと関連付けられているフェデレーテッド・サーバー上にインストールすることをお勧めします。SDK バージョン 15.0 を Sybase クライアントとして使用する場合は、ESD #3 以降をインストールすることをお勧めします。ESD をインストールしない場合、Sybase ラッパーの使用時に予期しないエラーを受け取ることがあります。

Sybase クライアント、バージョン 15.0 のための Sybase ライブラリーの更新

UNIX: Sybase クライアント、バージョン 15.0 を使用する場合、Sybase **lnsyblibs** スクリプトを実行して Sybase ライブラリー名を更新し、サポートされる Sybase ライブラリー・ファイルのバージョン間での整合性を保持することができます。**lnsyblibs** スクリプトは、新しいライブラリー名から古いライブラリー名へのシンボリック・リンクを作成します。これにより、15.0 より前のアプリケーションでも名前変更されたライブラリーを処理できます。

Windows: Sybase クライアント、バージョン 15.0 を使用する場合、**copylibs.bat** ファイルを実行して必要な *.dll ファイルをコピーすることができます。これにより、より前のアプリケーションでも名前変更されたライブラリーを処理できます。

lnsyblibs スクリプトのエラー

現在の **lnsyblibs** スクリプトには問題があります。'lnsyblibs create' を実行すると、次のエラー・メッセージが発生します。

```
"libsyb*.s[o: No such file or directory"
```

この問題の回避策として、スクリプトの 34 行目にある [] (大括弧) 内の | (縦線) を削除できます。Sybase はこの問題を認識しています。詳しくは、Sybase サポートを参照してください。

マージ結合のサポート

Sybase マージ結合はデフォルトでは有効になっていません。Sybase ASE データ・ソースでマージ結合を使用するには、その前に、Sybase サーバーとフェデレーテッド・サーバーの両方でマージ結合を以下のように有効にする必要があります。

1. Sybase サーバーでマージ結合を有効にします。Enabling and disabling merge joins を参照してください。
2. ALTER SERVER ステートメントを実行して、フェデレーテッド・サーバーで以下のようにマージ結合を有効にします。

```
ALTER SERVER sybase_server OPTIONS (ADD DB2_merge_join 'Y');
```

ここで、*sybase_server* は Sybase サーバーの名前です。

サポートされないデータ・タイプ

サポートされないデータ・タイプを含むデータ・ソース・オブジェクトに対してニックネームを作成することはできません。Sybase ラッパーは、ASE バージョン 12.5.1 で導入された以下のデータ・タイプをサポートしていません。

- DATE
- TIME

Sybase ラッパーは、ASE バージョン 15.0 で導入された以下のデータ・タイプをサポートしていません。

- BIGINT
- LONGSYSNAME
- UNITEXT
- UNSIGNED BIGINT
- UNSIGNED INT
- UNSIGNED SMALLINT

Sybase 環境変数の設定

Sybase 環境変数は、フェデレーテッド・サーバー上の db2dj.ini ファイル内で設定する必要があります。

このタスクについて

制約事項

db2dj.ini ファイルに関する制約事項を確認してください。

db2dj.ini ファイルには、フェデレーテッド・サーバーにインストールされている Sybase Open Client ソフトウェアに関する構成情報が含まれています。

Sybase データ・ソースには、必須およびオプションの環境変数があります。

Sybase ラッパーをインストールする前に Sybase Open Client ソフトウェアをインストールした場合、必須の Sybase 環境変数は db2dj.ini ファイルで設定されています。

Sybase ラッパーをインストールする前に Sybase Open Client ソフトウェアをインストールしなかった場合、または、オプションの環境変数のいずれかを設定する場合は、このタスクの手順を使用して環境変数を設定する必要があります。

手順

1. 次の方法のいずれかを使用します。

方法	説明
環境変数を自動で設定します。	IBM InfoSphere Federation Server インストール・ウィザードを実行します。ウィザードの指示に従ってください。 重要: インストール・ウィザードを実行することによって、必須の環境変数を設定します。オプションの環境変数は手動で設定する必要があります。
ラッパー構成ツールで環境変数を設定します。	ラッパー構成ツールを実行し、必須の環境変数とオプションの環境変数を設定します。ラッパー構成ツールの実行の指示に従ってください。

方法	説明
環境変数を手動で設定します。	<p>db2dj.ini ファイルを編集します。</p> <p>db2dj.ini ファイルは、DB2 レジストリー変数 DB2_DJ_INI が指定するファイルにあります。この変数が設定されていない場合のデフォルトは以下のとおりです。</p> <p>Linux または UNIX</p> <pre>instancehome/sqlllib/cfg/db2dj.ini</pre> <p><i>instancehome</i> インスタンス所有者のホーム・ディレクトリーです。</p> <p>Windows</p> <pre>%DB2PATH%\cfg\db2dj.ini</pre> <p>%DB2PATH% DB2 データベース・システムがインストールされているディレクトリー (例えば、C:\Program Files\IBM\sqlllib) です。</p> <p>db2dj.ini ファイルがない場合は、テキスト・エディターで以下の Sybase 環境変数を指定して作成する必要があります。</p> <p>SYBASE</p> <p>Sybase Open Client ソフトウェアがインストールされているディレクトリー・パスです。以下に例を示します。</p> <pre>SYBASE=/sybase</pre> <p>SYBASE_OCS</p> <p>OCS ディレクトリーの SYBASE_OCS 環境変数の名前です。以下に例を示します。</p> <pre>SYBASE_OCS=OCS-12_5</pre> <p>重要: Sybase 環境変数が db2dj.ini ファイルで指定されていない場合、エラーが生じます。</p>

- Linux および UNIX の場合、フェデレーテッド・データベース・インスタンス上にある .profile ファイルを更新する必要があります。

以下のコマンドを実行して、.profile ファイルを、db2dj.ini ファイルに追加した Sybase 環境変数で更新してください。

```
export SYBASE=sybase_home_directory
export SYBASE_OCS=OCS-version_release
export PATH=$SYBASE/bin:$PATH
```

sybase_home_directory

Sybase Open Client ソフトウェアがインストールされているディレクトリーです。

sybase_ocs_directory

OCS ディレクトリーの名前で、これは必須です。

- ホーム・ディレクトリーから、フェデレーテッド・データベース・インスタンス上にある .profile ファイルを以下のように実行し、DB2 データベース・インスタンスを開始します。

```
. .profile
```

- Linux など、オペレーティング・システムによっては、Sybase クライアント・ライブラリーのパスを DB2LIBPATH db2set 変数に追加しなければなりません。以下に例を示します。

```
db2set DB2LIBPATH=/opt/sybase125/OCS-12_5/lib
```

- フェデレーテッド・サーバーに環境変数を設定するには、次のコマンドを使用してフェデレーテッド・データベース・インスタンスを停止してから再開することによりリサイクルします。

```
db2stop  
db2start
```

次のタスク

この作業が終了したら、ラッパーを登録しなければなりません。

Sybase 環境変数

Sybase データ・ソースには、必須およびオプションの環境変数があります。これらの変数は db2dj.ini ファイルに設定されます。

Sybase では以下の環境変数が有効です。

- SYBASE
- SYBASE_OCS
- SYBASE_CHARSET (オプション)

変数の説明

SYBASE

Sybase Open Client ソフトウェアがインストールされているディレクトリー・パスを指定します。この環境変数の完全修飾パスを指定します。

たとえば、Sybase Open Client バージョン 12.5 がディレクトリー・パス D:%djxclient%sybase%V125 にインストールされている場合、以下の SYBASE 環境変数を指定します。

```
SYBASE=D:%djxclient%sybase%V125
```

Sybase Open Client バージョン 12.0 がディレクトリー・パス D:%djxclient%sybase%V12 にインストールされている場合、以下の SYBASE 環境変数を指定します。

```
SYBASE=D:%djxclient%sybase%V12
```

SYBASE_OCS

インストールされている Sybase Open Client ソフトウェアのディレクトリー、バージョン、およびリリースを指定します。この環境変数を指定するときには、完全修飾パスを指定してはなりません。

```
SYBASE_OCS=OCS-version_release
```

たとえば、Sybase Open Client バージョン 12.0 がディレクトリー・パス D:%djxclient%sybase%V12%OCS-12_0 にインストールされている場合、以下の値を SYBASE_OCS 環境変数の値に指定します。

```
SYBASE_OCS=OCS-12_0
```

Sybase Open Client バージョン 12.5 がディレクトリー・パス
D:\djxclient\sybase\125\OCS-12_5 にインストールされている場合、以下の値を SYBASE_OCS 環境変数に指定します。

```
SYBASE_OCS=OCS-12_5
```

SYBASE_CHARSET

使用する文字セットの名前を指定します。SYBASE_CHARSET 環境変数を、フェデレーテッド・サーバーの **CODESET** パラメーターで指定するコード・セットに設定します。有効な文字セット名のリストは \$SYBASE\charsets ディレクトリーにあります。

指定するコード・セットは、**CODESET** パラメーターと SYBASE_CHARSET 環境変数の間で異なります。例えば、**CODESET** パラメーターを UTF-8 (8-bit Unicode Transformation Format) に設定する場合、SYBASE_CHARSET 環境変数で UTF8 を指定します。

```
SYBASE_CHARSET=utf8
```

SYBASE_CHARSET 環境変数を設定しない場合、ラッパーはフェデレーテッド・データベースのコード・ページで指定された文字セットと一致する Sybase 文字セットを使用します。一致する Sybase 文字セットがない場合、ラッパーは iso_1 文字セットを使用します。

Sybase クライアント構成ファイルのセットアップおよびテスト (Windows)

Sybase クライアント構成ファイルは、フェデレーテッド・サーバーにインストールされているクライアント・ライブラリーを使用して、Sybase データベースに接続するために使用されます。

始める前に

Sybase クライアント・ソフトウェアがフェデレーテッド・サーバー上にインストールされている必要があります。

このタスクについて

クライアント構成ファイルは、各 Sybase SQL Server および Adaptive Server Enterprise インスタンスのロケーションと、データベース・サーバーへの接続のタイプ (プロトコル) を指定します。

Sybase に接続するために使用されるフェデレーテッド・サーバー内のそれぞれのインスタンスごとに、クライアント構成ファイルをセットアップする必要があります。

手順

Windows を実行するフェデレーテッド・サーバー上で Sybase クライアント構成ファイルをセットアップしてテストするには、次のようにします。

手順

1. Sybase Open Client ソフトウェアに付属のユーティリティを使用して、クライアント構成ファイルをセットアップします。このユーティリティの使用法の詳細は、Sybase の資料を参照してください。

クライアント構成ファイルは、%SYBASE%\%ini ディレクトリーに作成されます。ファイルの名前は sql.ini です。

2. Sybase Open クライアント・ソフトウェアが Sybase サーバーに接続できるか確認するために、接続をテストしてください。

接続をテストするには、該当の Sybase 照会ユーティリティ (isql など) を使用してください。

たとえば、Sybase Open Client ソフトウェアがディレクトリー・パス D:\%djcclient%\sybase\%V125 にインストールされている場合、コマンド・プロンプトから以下のコマンドを実行できます。

```
cd D:\%djcclient%\sybase\%V125%\OCS-12_5\bin
isql -Ssybnode -Umary
```

あるいは、コマンド・プロンプトから以下のコマンドを実行することもできます。

```
%SYBASE%\%SYBASE_OCS%\bin\isql -Ssybnode -Umary
```

interfaces ファイルへのパスの指定

: デフォルト・ファイルではない interfaces ファイルを使用する場合は、IFILE サーバー・オプションを使用してパスを指定します。Sybase ラッパーは、interfaces ファイルを以下の場所で、以下に指定されている順序で検索します。

- a. IFILE サーバー・オプション
- b. %DB2PATH%\%interfaces
- c. %SYBASE%\%ini%\sql.ini

次のタスク

この作業が完了したら、環境変数を設定することができます。

Sybase クライアント構成ファイルのセットアップおよびテスト (UNIX)

Sybase クライアント構成ファイルは、フェデレーテッド・サーバーにインストールされているクライアント・ライブラリーを使用して、Sybase データベースに接続するために使用されます。

始める前に

Sybase クライアント・ソフトウェアがフェデレーテッド・サーバー上にインストールされている必要があります。

このタスクについて

クライアント構成ファイルは、各 Sybase SQL Server および Adaptive Server Enterprise インスタンスのロケーションと、データベース・サーバーへの接続のタイプ (プロトコル) を指定します。

Sybase に接続するために使用されるフェデレーテッド・サーバー内のそれぞれのインスタンスごとに、クライアント構成ファイルをセットアップする必要があります。

手順

1. Sybase Open Client ソフトウェアに付属のユーティリティーを使用して、クライアント構成ファイルをセットアップします。

クライアント構成ファイルは、`$SYBASE` ディレクトリーに作成されます。ファイルのデフォルト名は `interfaces` です。このユーティリティーの使用法の詳細は、Sybase の資料を参照してください。

2. Sybase Open Client ソフトウェアが Sybase サーバーに接続できるか確認するために、接続をテストしてください。

接続をテストするには、該当の Sybase 照会ユーティリティー (`isql` など) を使用してください。

たとえば、Sybase Open Client ソフトウェアがディレクトリー・パス `/opt/djxclient/sybase/V125` にインストールされている場合、UNIX プロンプトから以下のコマンドを実行できます。

```
cd /opt/djxclient/sybase/V125/OCS-12_5
isql -Ssybnode -Umary
```

あるいは、UNIX プロンプトから以下のコマンドを実行することもできます。

```
$SYBASE/$SYBASE_OCS/bin/isql -Ssybnode -Umary
```

interfaces ファイルへのパスの指定

： デフォルト・ファイルではない `interfaces` ファイルを使用する場合は、`IFILE` サーバー・オプションを使用してパスを指定します。Sybase ラッパーは、`interfaces` ファイルを以下の場所で、以下に指定されている順序で検索します。

- a. `IFILE` サーバー・オプション
- b. `sqllib/interfaces`
- c. `$SYBASE/interfaces`

次のタスク

この作業が完了したら、環境変数を設定することができます。

Sybase ラッパーの登録

Sybase データ・ソースにアクセスするためには、ラッパーを登録する必要があります。フェデレーテッド・サーバーは、ラッパーを使用してデータ・ソースとの通信やデータ・ソースからのデータの取得を行います。ラッパーは、ライブラリー・ファイルのセットとしてインプリメントされます。

手順

CREATE WRAPPER ステートメントをコマンド行から発行し、Sybase ラッパーのデフォルト名を指定します。
たとえば、次のようにします。

```
CREATE WRAPPER CTLIB
```

要確認: デフォルト名 CTLIB を使用してラッパーを登録する場合、フェデレーテッド・サーバーは、フェデレーテッド・サーバーが実行しているオペレーティング・システムに応じた適切な Sybase ラッパー・ライブラリーを自動的に使用します。

デフォルトのラッパー名がフェデレーテッド・データベース内の既存のラッパー名と競合する場合、デフォルトのラッパー名を、指定の名前で置き換えることができます。デフォルト名とは別の名前を使用する場合は、CREATE WRAPPER ステートメントに LIBRARY パラメーターを含める必要があります。

たとえば、AIX を使用するフェデレーテッド・サーバー上で sybase_wrapper という名前を使用してラッパーを登録するには、次のステートメントを実行します。

```
CREATE WRAPPER sybase_wrapper  
LIBRARY 'libdb2ctlib.a';
```

指定するラッパー・ライブラリー・ファイルは、フェデレーテッド・サーバーのオペレーティング・システムによって異なります。

次のタスク

この作業が完了したら、サーバー定義を登録することができます。

Sybase ラッパー・ライブラリー・ファイル

Sybase ラッパー・ライブラリー・ファイルは、ラッパーのインストール時にフェデレーテッド・サーバーに追加されます。

Sybase ラッパーをインストールすると、デフォルトのディレクトリー・パスに 3 つのライブラリー・ファイルが追加されます。たとえば、フェデレーテッド・サーバーが AIX 上で実行されている場合、ディレクトリー・パスには、libdb2ctlib.a、libdb2ctlibF.a、および libdb2ctlibU.a のラッパー・ライブラリー・ファイルが追加されます。他のラッパー・ライブラリー・ファイルは Sybase ラッパーによって内部的に使用されます。

ラッパーの登録時にデフォルトのラッパー名を使用しない場合は、CREATE WRAPPER ステートメントに LIBRARY パラメーターを含め、デフォルトのライブラリー・ファイル名を指定する必要があります。

以下の表に、デフォルトのディレクトリー・パスとデフォルトのラッパー・ライブラリー・ファイルの名前をリストします。

表 35. Sybase ラッパー・ライブラリーのロケーションとファイル名

オペレーティング・システム	ディレクトリー・パス	ライブラリー・ファイル名
AIX	/usr/opt/install_path/lib32/ /usr/opt/install_path/lib64/	libdb2ctlib.a

表 35. Sybase ラッパー・ライブラリーのロケーションとファイル名 (続き)

オペレーティング・システム	ディレクトリー・パス	ライブラリー・ファイル名
Linux	/opt/IBM/db2/install_path/lib32	libdb2ctlib.so
	/opt/IBM/db2/install_path/lib64	
Solaris	/opt/IBM/db2/install_path/lib32	libdb2ctlib.so
	/opt/IBM/db2/install_path/lib64	
Windows	%DB2PATH%\bin	db2ctlib.dll

install_path は、UNIX または Linux 上で IBM InfoSphere Federation Server がインストールされているディレクトリーのパスです。

Sybase データ・ソースのサーバー定義の登録

アクセスするそれぞれの Sybase サーバーをフェデレーテッド・データベース内に登録する必要があります。

手順

1. Sybase interfaces ファイルでノード名を見つけます。以下の例は、UNIX または Windows を実行するフェデレーテッド・サーバーでの interfaces ファイルの項目を示しています。

UNIX:

```
sybase125
query tcp ether anaconda 4100
```

Windows:

```
[sybase125]
query=TCP,anaconda,4100
```

- それぞれの例の最初の行は、たとえば sybase125 などのノード名です。
 - 2 番目の行は、接続のタイプ、ホスト名、およびポート番号をリストしています。この例では、TCP はこれが TCP/IP 接続であることを示し、anaconda がホスト名、4100 がポート番号です。
2. サーバー定義を作成するには、コマンド行から CREATE SERVER ステートメントを発行します。

例:

```
CREATE SERVER server_definition_name TYPE SYBASE
VERSION Sybase_client_version_number WRAPPER wrapper_name
OPTIONS (NODE 'node_name', DBNAME 'database_name');
```

'node_name' および 'database_name' 変数は CREATE SERVER ステートメントではオプションとなっていますが、これらのオプションは Sybase データ・ソースの場合は必要です。

重要: Sybase クライアント構成ファイルをセットアップした際に sql.ini ファイルの名前を interfaces に変更しなかった場合は、サーバー定義の登録時に IFILE サーバー・オプションを含める必要があります。

サーバー定義を登録した後、サーバー・オプションを追加またはドロップするには、ALTER SERVER ステートメントを使用します。

IBM Data Studio では、「SQL および XQuery エディター」でコマンドを入力し、コマンドをデプロイすることが可能です。詳細は、AddTopicRef を参照してください。

次のタスク

この作業を完了したら、ユーザー・マッピングを作成することができます。

CREATE SERVER ステートメント - Sybase ラッパーの例

Sybase ラッパーにサーバー定義を登録するには、CREATE SERVER ステートメントを使用します。このトピックでは、必須パラメーターを使用する完全指定の例と、追加のサーバー・オプションを使用する例を提供します。

次に、CREATE SERVER ステートメントを実行して Sybase ラッパーのサーバー定義を登録する方法の例を示します。

```
CREATE SERVER SYBSERVER TYPE SYBASE VERSION 12.0 WRAPPER CTLIB
OPTIONS (NODE 'sybnode', DBNAME 'sybdb');
```

SYBSERVER

Sybase サーバーに割り当てる名前。重複するサーバー定義名は使用できません。

TYPE SYBASE

アクセスを構成するデータ・ソース・サーバーのタイプを指定します。
CTLIB ラッパーの場合、サーバー・タイプは SYBASE でなければなりません。

VERSION 12.0

フェデレーテッド接続に使用されている Sybase データベース・クライアント・ソフトウェアのバージョン。

WRAPPER CTLIB

CREATE WRAPPER ステートメントで指定したラッパー名。

NODE 'sybnode'

Sybase サーバーが存在するノードの名前。ノード名は、interfaces ファイルにあります。この値は大文字と小文字を区別します。

ノード名は CREATE SERVER ステートメントではオプションとなっていますが、Sybase データ・ソースの場合は必要です。

DBNAME 'sybdb'

アクセスしたい Sybase データベースの名前。この値は大/小文字の区別があります。

データベース名は CREATE SERVER ステートメントではオプションとなっていますが、Sybase データ・ソースの場合は必要です。

サーバー・オプション

サーバー定義の作成時に、CREATE SERVER ステートメントで追加のサーバー・オプションを指定することができます。サーバー・オプションには、一般的なサーバー・オプションと Sybase に特有のサーバー・オプションを指定できます。

DB2_ID_QUOTE_CHAR サーバー・オプション

28 文字より長い列名を持つ表のニックネームを作成する場合、DB2_ID_QUOTE_CHAR サーバー・オプションを指定して、それを単一引用符で囲んだ単一のブランクに設定する必要があります。以下に例を示します。

```
CREATE SERVER SYBSERVER TYPE SYBASE VERSION 12.0 WRAPPER CTLIB
  OPTIONS (NODE 'sybnode', DBNAME 'sybdb',
    DB2_ID_QUOTE_CHAR ' ');
```

IFILE サーバー・オプション

デフォルトの `interfaces` ファイルを使用しない場合、`interfaces` ファイルを作成し、`CREATE SERVER` ステートメントに `IFILE` サーバー・オプションを含める必要があります。デフォルトの `interfaces` ファイルは、Linux および UNIX の場合は `SYBASE/interfaces`、Windows の場合は `%SYBASE%\ini%sql.ini` です。

`IFILE` サーバー・オプションに指定する値は、Sybase Open Client `sql.ini` ファイルの絶対パスおよびファイル名です。

次に、Windows を実行するフェデレーテッド・サーバー上でサーバー定義を登録する場合に、`IFILE` サーバー・オプションを使用する方法の例を示します。

```
CREATE SERVER SYBSERVER TYPE SYBASE VERSION 12.0 WRAPPER CTLIB
  OPTIONS (NODE 'sybnode', DBNAME 'sybdb',
    IFILE 'C:%Sybase%ini%sql.ini');
```

TIMEOUT サーバー・オプション

`TIMEOUT` サーバー・オプションは、ラッパーが Sybase サーバーからの応答を待つ秒数を設定します。`TIMEOUT` オプションを使用して、トランザクションのデッドロックを避けてください。

次に、サーバー定義を登録する場合に、`TIMEOUT` サーバー・オプションを指定する方法の例を示します。

```
CREATE SERVER SYBSERVER TYPE SYBASE VERSION 12.0 WRAPPER CTLIB
  OPTIONS (NODE 'sybnode', DBNAME 'sybdb',
    TIMEOUT '60');
```

その他の Sybase 特有のサーバー・オプションは、以下のとおりです。

- LOGIN_TIMEOUT
- PACKET_SIZE

Sybase データ・ソースのユーザー・マッピングの作成

Sybase サーバーにアクセスする場合、フェデレーテッド・サーバーは、データ・ソースに対して有効なユーザー ID とパスワードを使用して、Sybase サーバーへの接続を確立します。それぞれのフェデレーテッド・サーバーのユーザー ID とパスワード、および対応するデータ・ソースのユーザー ID とパスワードの間の関連付け (ユーザー・マッピング) を定義しなければなりません。

このタスクについて

フェデレーテッド・システムにアクセスする、各ユーザー ID のユーザー・マッピングを作成して、Sybase データ・ソースへの分散要求を送信します。

手順

CREATE USER MAPPING ステートメントを発行します。
たとえば、次のようにします。

```
CREATE USER MAPPING FOR local_userID SERVER server_definition_name  
    OPTIONS (REMOTE_AUTHID 'remote_userID', REMOTE_PASSWORD 'remote_password');
```

次のタスク

このタスクを完了したら、Sybase サーバーへの接続をテストします。

CREATE USER MAPPING ステートメント - Sybase ラッパーの例

CREATE USER MAPPING ステートメントを使用して、フェデレーテッド・サーバーのユーザー ID を Sybase サーバーのユーザー ID およびパスワードにマップしてください。このトピックでは、必須パラメーターを使用する完全指定の例と、CREATE USER MAPPING ステートメントで DB2 特殊レジスター USER を使用する方法を示す例を提供します。

以下に示すのは、フェデレーテッド・サーバーのユーザー ID を Sybase サーバーのユーザー ID およびパスワードにマップする方法を示す例です。

```
CREATE USER MAPPING FOR maria SERVER SYBSERVER  
    OPTIONS (REMOTE_AUTHID 'mary', REMOTE_PASSWORD 'day2night');
```

maria Sybase サーバーで定義されたユーザー ID にマップするローカルのユーザー ID を指定します。

SERVER SYBSERVER

Sybase サーバーの CREATE SERVER ステートメントで登録したサーバー定義名を指定します。

REMOTE_AUTHID '*mary*'

maria をマップする Sybase サーバー側のユーザー ID を指定します。

CREATE SERVER ステートメントで FOLD_ID サーバー・オプションを「U」または「L」に設定している場合を除き、この値は大文字小文字を区別します。

リモート・ユーザー ID は CREATE SERVER ステートメントではオプションとなっていますが、Sybase データ・ソースの場合は必要です。

REMOTE_PASSWORD '*day2night*'

'*mary*' に関連するパスワードを指定します。CREATE SERVER ステートメントで FOLD_PW サーバー・オプションを「U」または「L」に設定している場合を除き、この値は大文字小文字を区別します。

リモート・パスワードは CREATE SERVER ステートメントではオプションとなっていますが、Sybase データ・ソースの場合は必要です。

DB2 特殊レジスター USER

DB2 特殊レジスター USER を使用して、CREATE USER MAPPING ステートメントを実行しているユーザーの許可 ID を、REMOTE_AUTHID ユーザー・オプションに指定されているデータ・ソースの許可 ID にマップすることができます。

以下に、特殊レジスター `USER` を含む `CREATE USER MAPPING` ステートメントの例を示します。

```
CREATE USER MAPPING FOR USER SERVER SYBSERVER
    OPTIONS (REMOTE_AUTHID 'mary', REMOTE_PASSWORD 'day2night');
```

Sybase サーバーへの接続のテスト

Sybase データ・ソース・サーバーへの接続をテストして、Sybase データ・ソースにアクセスするようにフェデレーテッド・サーバーが適切に構成されているかどうかを判別します。

このタスクについて

Sybase サーバーへの接続のテストは、定義したサーバー定義とユーザー・マッピングを使用して行うことができます。

手順

パススルー・セッションを開き、Sybase システム表に対して `SELECT` ステートメントを発行します。 `SELECT` ステートメントがカウントを戻す場合、サーバー定義およびユーザー・マッピングは適切にセットアップされています。

例:

```
SET PASSTHRU server_definition_name
SELECT count(*) FROM dbo.sysobjects
SET PASSTHRU RESET
```

`SELECT` ステートメントがエラーを戻す場合には、接続エラーのトラブルシューティングを行う必要があります。

次のタスク

このタスクを完了したら、Sybase の表およびビューのニックネームを登録できます。

データ・ソース接続エラーのトラブルシューティング

データ・ソース・サーバーへのテスト接続では、いくつかの理由でエラーが戻されることがあります。以下の処置により、エラーが発生した理由を判別することができます。

症状

データ・ソースに接続しようとしたときにエラーが戻されます。

原因

接続の問題には、いくつかの原因が考えられます。

問題の解決方法

データ・ソース接続エラーのトラブルシューティングを行うには、以下の項目で問題がないかどうかをチェックします。

- データ・ソースが使用可能であることを確認します。

- 該当する場合は、データ・ソース・サーバーが着信接続に対して構成されていることを確認します。
- REMOTE_AUTHID と REMOTE_PASSWORD のオプションに関するユーザー・マッピング設定を調べ、データ・ソースとの接続に有効な設定になっているか確認します。必要に応じて、ユーザー・マッピングを変更するか、別のユーザー・マッピングを作成します。
- 該当する場合は、フェデレーテッド・サーバー上のデータ・ソース・クライアント・ソフトウェアがインストールされていて、データ・ソースに接続できるように正しく構成されていることを確認します。
- ODBC データ・ソースの場合、フェデレーテッド・サーバー上の ODBC ドライバーがインストールされていて、ODBC データ・ソース・サーバーに接続できるように正しく構成されていることを確認します。Windows が稼働するフェデレーテッド・サーバーでは、ドライバーをチェックするときに「ODBC データ ソース アドミニストレータ」ツールを使用します。UNIX が稼働するフェデレーテッド・サーバーの場合は、ODBC クライアントのベンダーが提供している資料を参照してください。
- フェデレーテッド・サーバー上で設定されている変数の設定がデータ・ソースについて適切であることを確認します。これらの変数には、システム環境変数、db2dj.ini ファイル、および DB2 Profile Registry (db2set) 変数が含まれます。
- サーバー定義を調べます。必要であれば、サーバー定義をドロップして再作成します。

Sybase 表およびビューのニックネームの登録

登録する各 Sybase サーバー定義で、アクセスする表またはビューごとにニックネームを登録する必要があります。Sybase サーバーを照会するときには、データ・ソース・オブジェクトの名前ではなく、これらのニックネームを使用します。

始める前に

ニックネームを登録する前に、Sybase データ・ソースの統計を更新します。フェデレーテッド・データベースは、データ・ソースのカatalog統計に基づいて、照会処理を最適化します。DB2 RUNSTATS コマンドに相当するデータ・ソース・コマンドを使用して、データ・ソースの統計を更新します。

手順

コマンド行から CREATE NICKNAME ステートメントを発行します。

例:

```
CREATE NICKNAME nickname
FOR server_definition_name."remote_schema"."remote.table" ;
```

ニックネームの長さは 128 文字までです。

ニックネームを作成すると、フェデレーテッド・サーバーはニックネームを使用してデータ・ソース・カatalogを照会します。この照会は、データベース・ソースの表またはビューへの接続をテストします。接続ができない場合、エラー・メッセージが出ます。

次のタスク

ニックネームを作成したい Sybase の表またはビューごとに、このステップを繰り返します。

CREATE NICKNAME ステートメント - Sybase ラッパーの例

CREATE NICKNAME ステートメントを使用して、アクセスしたい Sybase の表またはビューのニックネームを登録します。このトピックには、必須パラメーターを使用する完全指定の例が含まれています。

次に、CREATE NICKNAME ステートメントを使用して Sybase の表またはビューのニックネームを登録する方法の例を示します。

```
CREATE NICKNAME SYBSALES FOR SYBSERVER."vinnie"."europe";
```

SYBSALES

Sybase の表またはビューを識別するために使用する固有なニックネームです。

重要: ニックネームは、2 つの部分 (スキーマとニックネーム) からなる名前です。ニックネームの登録時にスキーマを省略すると、そのニックネームのスキーマはニックネームを登録するユーザーの許可 ID になります。

SYBSERVER."vinnie"."europe"

リモート・オブジェクトを表す、3 つの部分からなる ID。

- *SYBSERVER* は、CREATE SERVER ステートメントで Sybase データベース・サーバーに割り当てたサーバー定義名です。
- *vinnie* は、表またはビューが属する所有者のユーザー ID です。
- *europe* は、アクセスしたいリモートの表またはビューの名前です。

フェデレーテッド・サーバーは、引用符で名前を囲まない限り、Sybase スキーマおよび表の名前を大文字に変換します。

Sybase ラッパー構成のトラブルシューティング

Sybase ラッパー・ライブラリーのロード時の問題

Sybase ラッパーの作成時に、Sybase Open Client ソフトウェアのインストールに関連したエラーが発生して、Sybase ラッパー・ライブラリーのロードができない場合があります。

症状

Sybase ラッパーの作成時に、以下の SQL エラーが出されます。

```
SQL10013N 指定されたライブラリー "db2ctlibF.d11" がロードできませんでした。
```

原因

Sybase XA Interface Library for ASE Distributed Transaction Manager が Sybase Open Client ソフトウェアと共に Windows システムにインストールされていませんでした。

問題の解決方法

Sybase Open Client ソフトウェアを Windows に再インストールし、「フル」または「カスタム」インストール・オプションを選択します。「カスタム」インストール・オプションを選択する場合は、「**XA Interface Library for ASE Distributed Transaction Manager**」オプションを指定します。

欠落している SYBASE 環境変数

db2dj.ini ファイルが正しいディレクトリにないか、欠落している場合は、SQL エラーが発生します。

症状

以下の SQL エラーが出されます。

SQL1822N 予期しないエラー・コード "" をデータ・ソース "server name" から受け取りました。関連したテキストとトークンは "SYBASE variable not set" です。

原因

db2dj.ini ファイルが見つからなかったか、そのファイルに SYBASE 環境変数が含まれていませんでした。db2dj.ini ファイルは、DB2 レジストリー変数 DB2_DJ_INI によって指定されたディレクトリ内にあります。DB2_DJ_INI 変数が設定されていない場合、db2dj.ini ファイルは、オペレーティング・システムに応じて以下のデフォルト・パスのいずれかにあります。

- UNIX の場合: *instancehome*/sqllib/cfg/db2dj.ini

instancehome

インスタンス所有者のホーム・ディレクトリーです。

- Windows の場合: %DB2PATH%\cfg\db2dj.ini

%DB2PATH%

DB2 データベース・システムがインストールされているディレクトリー (たとえば、C:\Program Files\IBM\sqllib) です。

問題の解決方法

必要な Sybase 環境変数を含む db2dj.ini ファイルを作成し、使用しているオペレーティング・システムに応じた正しいディレクトリーにそのファイルを置きます。db2dj.ini ファイルを作成するには、テキスト・エディターを使用できます。

欠落している Sybase ノード名

Sybase クライアント構成ファイルが適切に構成されていない場合、フェデレーテッド・システムは Sybase ノード名を見つけることができない可能性があります。

症状

以下の SQL エラーが出されます。

SQL1097N ノード名がノード・ディレクトリーに見つかりません。

原因

Sybase ノード名が見つかりませんでした。

問題の解決方法

この問題を解決するには、以下のようにします。

- IFILE サーバー・オプションが指定された場合、ノード名が、オプションによって指定されたファイルで宣言されていることを確認します。
- sqllib ディレクトリーに interfaces ファイルが存在する場合、ノード名がそのファイルで宣言されていることを確認します。
- IFILE サーバー・オプションが指定されていない場合で、sqllib ディレクトリーに interfaces ファイルが存在しない場合は、ノード名が %SYBASE%\\$ini¥sql.ini ファイル (Windows の場合) または \$SYBASE/interfaces ファイル (UNIX の場合) で宣言されていることを確認します。

表構造ファイル・データ・ソースへのアクセスの構成

フェデレーテッド・システムを使用すると、表構造ファイルにあるデータと他のソースからの情報を統合させることができます。

このタスクについて

フェデレーテッド・サーバーが表構造ファイルのデータ・ソースにアクセスするように構成するには、アクセスするデータ・ソースとオブジェクトの情報をフェデレーテッド・サーバーに渡す必要があります。フェデレーテッド・サーバーを構成したら、照会を作成し、カスタム関数を使用して、表構造ファイル・データ・ソースにアクセスすることができます。

表構造ファイル - 概要

表構造ファイルとは、情報に関する一連のレコードまたは行から構成される正規の構造を持つファイルのことです。それぞれのレコードには、同一数のフィールドが含まれます。フィールド内のデータは、コンマなどの区切り文字によって区切られています。

以下の例は、DRUGDATA1.TXT というファイルの内容を示しています。このファイルは、3 つのレコードから構成されており、それぞれのレコードにコンマによって区切られている 3 つのフィールドがあります。

```
234,DrugnameA,Manufacturer1  
332,DrugnameB,Manufacturer2  
333,DrugnameC,Manufacturer2
```

1 番目のフィールドは薬品 (drug) の固有の ID 番号です。2 番目のフィールドは薬品 (drug) の名前です。3 番目のフィールドは薬品の製造会社 (manufacturer) の名前です。

フィールド区切り文字には、複数の文字分の長さが可能です。単一引用符は区切り文字としては使用できません。区切り文字は、ファイル全体で一貫性がなければなりません。NULL 値は 2 つの区切り文字が続くことによって表されます。NULL フィールドが行の末尾にある場合、区切り文字に行終止符を続けることによって表されます。列区切り文字は、列の有効なデータとして存在することはできません。

例:

```
234,DrugnameA,Manufacturer1
332,DrugnameB,Manufacturer2
333,DrugnameC,Manufacturer2
356,,Manufacturer1
```

表構造ファイルの属性

表構造ファイル内のレコード (行) は、ソートするか未ソートのままにすることができます。表構造ファイルのラッパーでは、ソートされていないファイルより、ソートされたファイルの方がより効率的にファイルを検索できます。

表構造ファイル内のデータをソートする場合、ソートは、キー列で昇順にする必要があります。ニックネーム用の列を定義するときには、キー列の `SORTED` オプションを `Y` に設定してください。そのように設定しないと、ラッパーは表構造ファイル内のデータを未ソート・データとして処理します。

ソート・ファイル

`DRUGDATA1.TXT` にはソートされたレコードが入っています。このファイルは、最初のフィールドである薬品の固有 ID 番号によってソートされています。このフィールドは、それぞれの薬品ごとに固有であるため、主キーとなっています。ソートされたファイルは必ず昇順でソートされていなければなりません。

```
234,DrugnameA,Manufacturer1
332,DrugnameB,Manufacturer2
333,DrugnameC,Manufacturer2
```

ソートされていないファイル

`DRUGDATA2.TXT` には未ソートのレコードが入っています。ファイルにリストされているレコードの順番に規則性はありません。

```
556,DrugnameB,Manufacturer2
234,DrugnameA,Manufacturer1
721,DrugnameC,Manufacturer2
```

表構造ファイル・ラッパー

表構造ファイル内のデータは、他の表構造ファイルのデータ、リレーショナル・データ、非リレーショナル・データ、または構造化されていないデータと結合することができます。

ラッパーを使用すると、フェデレーテッド・サーバーは、表構造ファイル内のデータを照会する SQL ステートメントを、データが通常のリレーショナル表またはビューに含まれている場合と同じように処理することができます。

以下の図は、フェデレーテッド・サーバーが表構造ファイルを処理する方法を示しています。

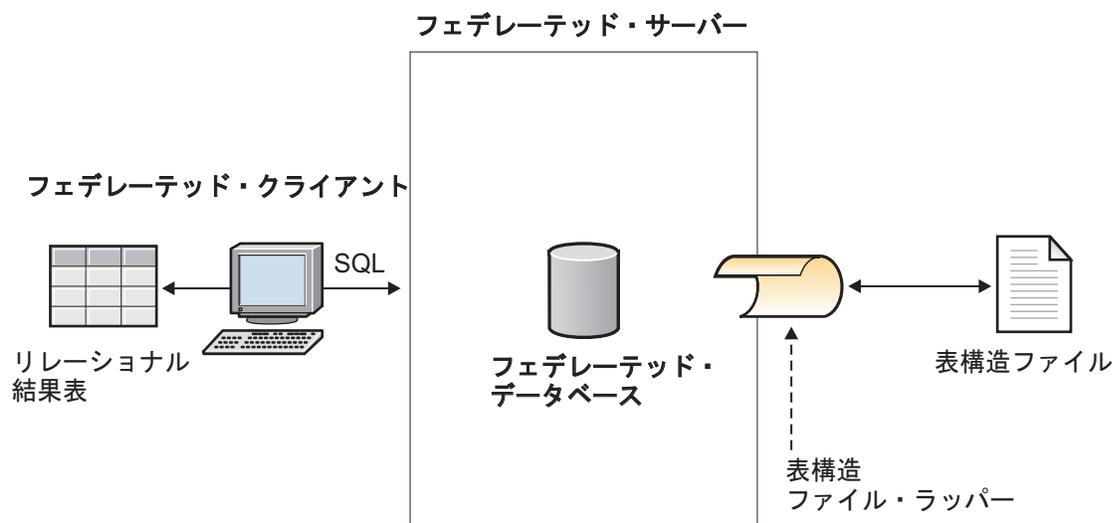


図 11. 表構造ファイル・ラッパーがどのように機能するか

例えば、表構造ファイル DRUGDATA2.TXT が研究所のコンピュータにあるとします。ファイルのデータは次のとおりです。

```
556,DrugnameB,Manufacturer1
234,DrugnameA,Manufacturer2
721,DrugnameC,Manufacturer2
```

使用している他のデータ・ソースの他の表を照会して、このデータと突き合わせることを試みるのは面倒です。

DRUGDATA2.TXT ファイルをフェデレーテッド・サーバーに登録した後、表構造ファイル・ラッパーは、データがリレーショナル表に入れている場合と同じようにこのファイル内のデータにアクセスすることができます。

例えば、以下に示す照会を実行するとします。

```
SELECT * FROM DRUGDATA2 ORDER BY DCODE
```

この照会は以下のような結果を生成します。

DCODE	DRUG	MANUFACTURER
234	DrugnameA	Manufacturer2
556	DrugnameB	Manufacturer1
721	DrugnameC	Manufacturer2

DRUGDATA2.TXT ファイル内のデータを、他のリレーショナルおよび非リレーショナル・データ・ソース内のデータと結合し、それらすべてのデータを一緒に分析することができます。

フェデレーテッド・サーバーへの表構造ファイル・データ・ソースの追加

フェデレーテッド・サーバーが表構造ファイルのデータ・ソースにアクセスするように構成するには、アクセスするデータ・ソースとオブジェクトの情報をフェデレーテッド・サーバーに渡す必要があります。

始める前に

- フェデレーションが、フェデレーテッド・サーバーとして機能するサーバー上にインストールされている必要があります。
- データベースがフェデレーテッド・サーバー上に存在している必要があります。

手順

1. 表構造ファイル・ラッパーを登録します。
2. 表構造サーバー定義を登録します。
3. 表構造ファイルのニックネームを登録します。

表構造ファイル・ラッパーの登録

表構造ファイルのデータ・ソースにアクセスするには、ラッパーを登録する必要があります。ラッパーは、データ・ソースとの通信やデータの取得のためにフェデレーテッド・サーバーによって使用されます。ラッパーは、ライブラリー・ファイルのセットとしてインプリメントされます。

手順

コマンド行から CREATE WRAPPER ステートメントを発行します。

```
CREATE WRAPPER wrapper_name  
LIBRARY library_name;
```

例えば、AIX オペレーティング・システムを使用するフェデレーテッド・サーバー上で `flat_files_wrapper` という名前を使用してラッパーを登録するには、次のステートメントを発行します。

```
CREATE WRAPPER flat_files_wrapper  
LIBRARY 'libdb2lsfile.a';
```

CREATE WRAPPER ステートメントには LIBRARY パラメーターを指定する必要があります。指定するラッパー・ライブラリー・ファイルの名前は、フェデレーテッド・サーバーのオペレーティング・システムによって異なります。CREATE WRAPPER ラッパー・ステートメントで指定する正しいライブラリー名については、『表構造ファイル・ラッパーのライブラリー・ファイル』のリストを参照してください。

表構造ファイル・ラッパー・ライブラリー・ファイル:

表構造ファイル・ラッパー・ライブラリー・ファイルは、IBM InfoSphere Federation Serverのインストール時にフェデレーテッド・サーバーに追加されます。

IBM InfoSphere Federation Server をインストールすると、デフォルトのディレクトリー・パスに 3 つのライブラリー・ファイルが追加されます。例えば、フェデレーテッド・サーバーが AIX 上で実行されている場合、ディレクトリー・パスには、

libdb2lsfile.a、libdb2lsfileF.a、および libdb2lsfileU.a が追加されます。デフォルトのラッパー・ライブラリー・ファイルは libdb2lsfile.a です。他のラッパー・ライブラリー・ファイルは特定のラッパー・オプションで使用されます。

CREATE WRAPPER ステートメントに LIBRARY パラメーターを含め、デフォルトのラッパー・ライブラリー・ファイルの名前を指定する必要があります。

以下の表に、デフォルトのディレクトリー・パスとデフォルトのラッパー・ライブラリー・ファイルの名前をリストします。

表 36. 表構造ファイル・クライアント・ライブラリーのロケーションとファイル名

オペレーティング・システム	ディレクトリー・パス	ラッパー・ライブラリー・ファイル名
AIX	/usr/opt/<install_path>/lib32/ /usr/opt/<install_path>/lib64/	libdb2lsfile.a
Linux	/opt/IBM/db2/<install_path>/lib32 /opt/IBM/db2/<install_path>/lib64	libdb2lsfile.so
Solaris	/opt/IBM/db2/<install_path>/lib32 /opt/IBM/db2/<install_path>/lib64	libdb2lsfile.so
Windows	%DB2PATH%\%bin	db2lsfile.dll

<install_path> は、Linux または UNIX 上で IBM InfoSphere Federation Server がインストールされているディレクトリー・パスです。

%DB2PATH% は、Windows で IBM InfoSphere Federation Server がインストールされているディレクトリーのパスを指定するのに使用する環境変数です。デフォルトの Windows ディレクトリー・パスは C:\Program Files\IBM\SQLLIB です。

表構造ファイルのサーバー定義の登録

フェデレーテッド・オブジェクトの階層では、ニックネームにより識別される表構造ファイルが特定のサーバー定義オブジェクトに関連付けられている必要があります。そのため、表構造ファイルの場合は、サーバー定義を登録する必要があります。

手順

コマンド行から CREATE SERVER ステートメントを発行します。

例:

```
CREATE SERVER server_definition_name
WRAPPER wrapper_name;
```

CREATE SERVER ステートメント - 表構造ファイル・ラッパーの例:

表構造ファイル・ラッパーにサーバー定義を登録するには、CREATE SERVER ステートメントを使用します。この例は、必須パラメーターを示しています。

以下の例は、生化学データを含むテキスト・ファイルについてサーバー定義 biochem_lab を登録する方法を示します。発行する CREATE SERVER ステートメントは次のとおりです。

```
CREATE SERVER biochem_lab WRAPPER flat_files_wrapper;
```

biochem_lab

表構造サーバー定義に割り当てる名前。重複するサーバー定義名は使用できません。

WRAPPER *flat_files_wrapper*

CREATE WRAPPER ステートメントで指定したラッパー名。

表構造ファイルのニックネームの登録

アクセスする表構造ファイルごとに、ニックネームを登録しなければなりません。表構造ファイル・データ・ソースを照会するときには、ファイルの名前ではなく、これらのニックネームを使用します。

このタスクについて

制約事項

- 非数値フィールドがその列タイプに対して長すぎる場合、超過したデータは切り捨てられます。
- ファイルの 10 進数フィールドで、基数文字の後ろに許可されているよりも多くの桁数がある場合、超過したデータは切り捨てられます。例えば数値が 10.123456 の場合、基数文字の後に許可される桁数が 3 桁であることがデータ・タイプで指定されていれば、数値は切り捨てられて 10.123 になります。基数文字は、LC_NUMERIC 各国語サポート区分の RADIXCHAR 項目によって区切られます。列データ・タイプの scale パラメーターは、基数文字の後に許可される桁数を指定します。
- Windows 2003 を実行するフェデレーテッド・サーバーから、共有ドライブ上にある表構造ファイル・データ・ソースにアクセスしようとする、照会が失敗することがあります。これは Windows 2003 の制限です。この問題を回避するには、CREATE NICKNAME ステートメントの FILE_PATH オプションで絶対パスを指定します。
- 1 行のデータの最大長は 10 MB (10485760 バイト) です。
-

表構造ファイルのニックネームを作成すると、ファイルのデータ内の情報がリレーショナル表にマップされます。表構造ファイルのニックネームは、次の 2 とおりの方法のいずれかで作成します。

- ニックネームを作成するときに、FILE_PATH ニックネーム・オプションを使用して表構造ファイルを指定する。
- データ・ソースを照会するときに、DOCUMENT ニックネーム列オプションを使用して表構造ファイルを指定する。このオプションが使用される場合、ニックネームは、スキーマがニックネーム定義に一致する任意の表構造ファイルからのデータを表すために使用できます。

ニックネームに指定する名前の長さは 128 文字までです。

手順

コマンド行から CREATE NICKNAME ステートメントを発行します。

例:

```
CREATE NICKNAME nickname
(
  column_name data_type,
  column_name data_type,
  column_name data_type
)
FOR SERVER server_definition_name
OPTIONS (nickname_options);
```

次のタスク

ニックネームを作成する表構造ファイルごとに、このステップを繰り返します。

CREATE NICKNAME ステートメント - 表構造ファイル・ラッパーの例:

CREATE NICKNAME ステートメントを使用して、アクセスする表構造ファイルのニックネームを登録します。

表構造ファイルのニックネームを登録するときは、FILE_PATH ニックネーム・オプションまたは DOCUMENT ニックネーム列オプションを指定する必要があります。

FILE_PATH ニックネーム・オプションを使ったニックネームの作成

次に、表構造ファイル DRUGDATA1.TXT に関する CREATE NICKNAME ステートメントの例を示します。

```
CREATE NICKNAME DRUGDATA1
(
  Dcode INTEGER NOT NULL,
  Drug CHAR(20),
  Manufacturer CHAR(20)
)
FOR SERVER biochem_lab
OPTIONS(FILE_PATH '/usr/pat/DRUGDATA1.TXT')
```

DRUGDATA1

表構造ファイルを識別するために使用する固有なニックネームです。

ニックネームは、2 つの部分 (スキーマとニックネーム) からなる名前です。ニックネームの登録時にスキーマを省略すると、そのニックネームのスキーマはニックネームを登録するユーザーの許可 ID になります。

Dcode INTEGER NOT NULL

薬品コードを含む表構造ファイル列の名前とデータ・タイプ。

Drug CHAR(20)

薬品の名前を含む表構造ファイル列の名前とデータ・タイプ。

Manufacturer CHAR(20)

薬品製造会社の名前を含む表構造ファイル列の名前とデータ・タイプ。

FOR SERVER biochem_lab

CREATE SERVER ステートメントで表構造ファイル・サーバー定義に割り当てた名前。

FILE_PATH '/usr/pat/DRUGDATA1.TXT'

アクセスするデータを含む表構造ファイルの完全修飾ディレクトリー・パスとファイル名を指定します。パスは、単一引用符で囲む必要があります。

DOCUMENT ニックネーム列オプションを使ったニックネームの作成

DOCUMENT ニックネーム列オプションを使用してニックネームを作成するときは、そのニックネームを使用する照会を実行する際に表構造ファイルの名前が提供されるように指定します。ニックネームを登録するときは、単一の列に対してのみ DOCUMENT ニックネーム列オプションを指定できます。DOCUMENT オプションに関連付けられる列は VARCHAR または CHAR データ・タイプでなければなりません。ニックネームを使用する照会を実行するときは、ファイルの絶対パスを含める必要があります。

以下に、DOCUMENT ニックネーム列オプションを指定する CREATE NICKNAME ステートメントの例を示します。

```
CREATE NICKNAME customers
(
  doc VARCHAR(100) OPTIONS(DOCUMENT 'FILE'),
  name VARCHAR(16),
  address VARCHAR(30),
  id VARCHAR(16)
FOR SERVER biochem_lab
```

customers

ニックネームのためのユニーク ID

ニックネームは、2 つの部分 (スキーマとニックネーム) からなる名前です。ニックネームの登録時にスキーマを省略すると、そのニックネームのスキーマはニックネームを登録するユーザーの許可 ID になります。

doc VARCHAR(100) OPTIONS(DOCUMENT 'FILE')

アクセスしたい表構造ファイルの名前を特定するために使用される列のデータ・タイプと名前。照会を実行するときにファイル名を指定します。

name VARCHAR(16)

顧客の名前を含む表構造ファイル列に対する名前とデータ・タイプ

address VARCHAR(30)

顧客の住所を含む表構造ファイル列に対する名前とデータ・タイプ

id VARCHAR(16)

FOR SERVER biochem_lab

CREATE SERVER ステートメントで表構造ファイル・サーバー定義に割り当てた名前。

FILE_PATH '/usr/pat/DRUGDATA1.TXT'

アクセスするデータを含む表構造ファイルの完全修飾ディレクトリー・パスとファイル名を指定します。CREATE NICKNAME ステートメントでは、FILE_PATH または DOCUMENT のいずれかのニックネーム・オプションを指定する必要があります。パスは、単一引用符で囲む必要があります。

オプション・パラメーターを使ったニックネームの作成

次に、表構造ファイル DRUGDATA1.TXT に関する CREATE NICKNAME ステートメントの例を示します。

```
CREATE NICKNAME DRUGDATA1
(
  Dcode INTEGER NOT NULL,
  Drug CHAR(20),
```

```
Manufacturer CHAR(20)
)
FOR SERVER biochem_lab
OPTIONS(FILE_PATH '/usr/pat/DRUGDATA1.TXT',
        COLUMN_DELIMITER ',',
        SORTED 'Y',
        KEY_COLUMN 'DCODE',
        VALIDATE_DATA_FILE 'Y')
```

DRUGDATA1

表構造ファイルを識別するために使用する固有なニックネームです。

ニックネームは、2 つの部分 (スキーマとニックネーム) からなる名前です。ニックネームの登録時にスキーマを省略すると、そのニックネームのスキーマはニックネームを登録するユーザーの許可 ID になります。

Dcode INTEGER NOT NULL

薬品コードを含む表構造ファイル列の名前とデータ・タイプ。

Drug CHAR(20)

薬品の名前を含む表構造ファイル列の名前とデータ・タイプ。

Manufacturer CHAR(20)

薬品製造会社の名前を含む表構造ファイル列の名前とデータ・タイプ。

FOR SERVER *biochem_lab*

CREATE SERVER ステートメントで表構造ファイル・サーバー定義に割り当てた名前。

FILE_PATH '/usr/pat/DRUGDATA1.TXT'

アクセスするデータを含む表構造ファイルの完全修飾ディレクトリー・パスとファイル名を指定します。CREATE NICKNAME ステートメントでは、FILE_PATH または DOCUMENT のいずれかのニックネーム・オプションを指定する必要があります。パスは、単一引用符で囲む必要があります。

COLUMN_DELIMITER ','

表構造ファイル内のフィールドを区切るのに使用される区切り文字を指定します。区切り文字値は、単一引用符で囲む必要があります。列区切り文字には、複数の文字分の長さが可能です。列区切り文字を指定しない場合、デフォルトの区切り文字はコンマです。単一引用符は区切り文字としては使用できません。列区切り文字は、ファイル全体で一貫性がなければなりません。NULL 値は 2 つの区切り文字が続くことによって表されます。NULL フィールドが行の末尾にある場合、区切り文字に行終止符を続けることによって表されます。列区切り文字は、列の有効なデータとして存在することはできません。例えば、列の 1 つにコンマが組み込まれたデータが含まれている場合、列区切り文字としてコンマを使用することはできません。

SORTED 'Y'

データ・ソース・ファイルがソートされていることを指定します。ソートされるデータ・ソースは、LC_COLLATE 各国語サポート・カテゴリーの設定によって定義されている現在のロケールの照合順序に従って、昇順でソートされなければなりません。データ・ソースがソートされるように指定する場合、VALIDATE_DATA_FILE オプションを 'Y' に設定してください。

SORTED パラメーターのデフォルト値は 'N' です。

KEY_COLUMN 'DCODE'

ファイルをソートするキーとなるファイル内の列の名前。キー列の値は単一

引用符で囲む必要があります。このオプションは、SORTED ニックネーム・オプションを指定する場合にのみ指定します。DOCUMENT ニックネーム列オプションで指定された列は、キー列として指定してはなりません。値は、CREATE NICKNAME ステートメントに定義されている列の名前にしなければなりません。キー列に NULL 値が入ることはできません。単一列のキーのみがサポートされます。複数列のキーは許可されていません。列は必ず昇順でソートされます。ソート・ニックネームに値が指定されていない場合、デフォルトとしてこれはニックネームが付けられたファイルの最初の列となります。

VALIDATE_DATA_FILE 'Y'

キー列が昇順でソートされていることをラッパーがチェックするかどうか、および NULL キーをチェックするかどうかを指定します。このオプションの有効値は Y または N です。DOCUMENT ニックネーム列オプションがファイル・パスに使用されている場合には、このオプションは使用できません。

ニックネームの登録時におけるキー列の指定

次のようにニックネーム・ステートメントで NOT NULL 制約を指定することで、キー列を指定することができます。

```
CREATE NICKNAME tox (tox_id INTEGER NOT NULL, toxicity VARCHAR(100))
FOR SERVER tox_server1
  OPTIONS (FILE_PATH'/tox_data.txt', SORTED 'Y')
```

```
CREATE NICKNAME weights (mol_id INTEGER, wt VARCHAR(100) NOT NULL)
FOR SERVER wt_server
  OPTIONS (FILE_PATH'/wt_data.txt', SORTED 'Y', KEY_COLUMN 'WT')
```

NOT NULL

列に NULL やブランクの値を含められないことを指定します。

ラッパーは NOT NULL 制約を強制しませんが、フェデレーテッド・データベースは強制します。ニックネームを作成して、列に NOT NULL 制約を付け、列に NULL 値の入った行を選択する場合、フェデレーテッド・データベースは NULL 値を NOT NULL 列に割り当てることはできないことを述べる SQL0407N エラーを出します。

この規則の例外はソートされたニックネームの場合です。ソートされたニックネームのキー列を NULL にすることはできません。ソートされたニックネームに関して NULL キー列が検出されると、キー列が欠落しているという SQL1822N エラーが出ます。

大文字と小文字が区別される列名

列が二重引用符付きで定義されていない限り、フェデレーテッド・データベースによって列名が大文字に変更されます。以下の例は、KEY_COLUMN オプションの値が単一引用符で囲まれているので、正しく動作しません。この例では、列名がフェデレーテッド・データベースによって EMPNO に変換されます。結果として、empno を照会で指定したとき、この列はフェデレーテッド・データベースに認識されません。

```
CREATE NICKNAME depart (  
  empno char(6) NOT NULL)  
  FOR SERVER DATASTORE  
  OPTIONS(FILE_PATH'data.txt', SORTED 'Y', KEY_COLUMN 'empno');
```

Windows 2003 フェデレーテッド・サーバー

Windows 2003 を実行するフェデレーテッド・サーバーから、共有ドライブ上にある表構造ファイル・データ・ソースにアクセスしようとする、以下のエラー・メッセージを出して照会が失敗することがあります。

```
SQL1822N データ・ソース "SERVERNAME1" から予期しないエラー・コード "ERRNO = 2" を  
受け取りました。  
関連するテキストおよびトークンは「ファイルの読み取り不可」です。  
SQLSTATE=560BD
```

これは Windows 2003 の制限です。この問題を回避するには、CREATE NICKNAME ステートメントの FILE_PATH オプションで絶対パスを指定します。

次の例は、FILE_PATH オプションで省略パスが指定された CREATE NICKNAME ステートメントを示しています。

```
CREATE NICKNAME nickname  
(  
  COL1 CHAR (10) NOT NULL  
)  
  FOR SERVER servername1  
  OPTIONS (FILE_PATH 'X:#textfile1.txt');
```

ここで、X:¥ は、リモート・マシンへマップするドライブです。省略パスを指定しているため、このニックネームを使用する照会は失敗する可能性があります。

Windows 2003 を実行するフェデレーテッド・サーバーの場合は、CREATE NICKNAME ステートメントの FILE_PATH オプションで絶対パスを指定してください。

例:

```
CREATE NICKNAME nickname  
(  
  COL1 CHAR (10) NOT NULL  
)  
  FOR SERVER servername1  
  OPTIONS (FILE_PATH '¥¥host.svl.ibm.com¥D¥¥textfile1.txt');
```

表構造ファイル・ラッパーのファイル・アクセス・コントロール・モデル

ラッパーは、フェデレーテッド・データベースのインスタンス所有者の許可情報を使用して表構造ファイルにアクセスします。ラッパーは、このユーザー ID またはグループ ID によって読み取ることができるファイルにだけアクセスできます。フェデレーテッド・データベースへの接続を確立する許可 ID は、表構造ファイルにアクセスするために使用されることはありません。

フェデレーテッド・サーバーでは、ニックネームが作成されている表構造ファイルはすべて、各ノードから同じパス名を使用してアクセス可能でなければなりません。

ん。ファイルは、すべてのノードから共通のパスを使用してアクセス可能であれば、フェデレーテッド・データベースのノードに置く必要はありません。

表構造ファイルにアクセスするには、ラッパーはセキュリティー上の目的でユーザー ID が必要です。表構造ファイル・ラッパーは、フェデレーテッド・データベース・サービスに関連付けられたユーザー ID を使用します。フェデレーテッド・データベース・サービスの名前は、データベース・インスタンスの名前によって異なります。例えば、データベース・インスタンスの名前が DB2 である場合、サービス名は DB2 - DB2 となります。フェデレーテッド・データベース・サービスに関連付けられたユーザー ID を判別するには、Windows で「コントロール パネル」を使用してサービスを表示します。サービス名をダブルクリックして、「ログオン」プロパティ・ページを表示します。

リモート・ドライブ上にある表構造ファイル

アクセスする表構造ファイルは、ローカル・ドライブまたはマップされたドライブに存在する必要があります。

Windows ドメインが構成されているネットワーク

フェデレーテッド・データベース・サービスのログオン・アカウントは、表構造ファイルがあるマップされたドライブ上の共有フォルダーにアクセスできるドメインのアカウントでなければなりません。

Windows ドメインが構成されていないネットワーク

フェデレーテッド・データベース・サービスのログオン・アカウントには、そのフォルダーを共有するコンピューター上の正当なユーザーと同じユーザー名およびパスワードがなければなりません。そのユーザーは、少なくとも読み取りアクセスの付与された共有フォルダーの許可リストに含まれていなければなりません。

表構造ファイル・ラッパーの照会パフォーマンス最適化のためのガイドライン

ソートされたファイルの使用、およびニックネームの統計の作成により、表構造ファイル照会のパフォーマンスを改善できます。

照会パフォーマンスを改善するには、以下のヒントを実行します。

- ファイル内のデータをソートします。フェデレーテッド・サーバーでは、ソートされていないファイルより、ソートされたファイルの方がより効率的に検索できます。
- ソートされたファイルの場合、照会をサブミットするときに、キー列の値または範囲を指定してください。
- 表構造ファイルのニックネームの統計は、SYSSTAT または SYSCAT のビューを更新することによって手操作で更新する必要があります。ニックネーム統計情報の更新機能を使用して、表構造ファイルのニックネームについて統計を更新してください。

Teradata データ・ソースへのアクセスの構成

Teradata データ・ソースにアクセスするようにフェデレーテッド・サーバーを構成するには、アクセスするデータ・ソースとオブジェクトに関する情報をフェデレーテッド・サーバーに提供しなければなりません。

始める前に

- Teradata クライアント・ソフトウェアが、フェデレーテッド・サーバーとして機能するサーバー上にインストールされ、構成されている必要があります。
- フェデレーションが、フェデレーテッド・サーバーとして機能するサーバー上にインストールされている必要があります。
- フェデレーテッド・サーバーのセットアップを確認します。
- フェデレーテッド・パラメーターをチェックして、フェデレーションが有効になっていることを確認します。

手順

1. Teradata サーバーへの接続をテストします。
2. Teradata ライブラリーがランタイム・リンクに対応していることを検査します (AIX)。
3. Teradata ラッパーの環境変数を設定します。
4. ラッパーを登録します。
5. サーバー定義を登録します。
6. ユーザー・マッピングを作成します。
7. Teradata サーバーへの接続をテストします。
8. Teradata の表およびビューのニックネームを登録します。

Teradata サーバーへの接続のテスト

Teradata サーバーへの接続をテストして、Teradata クライアント・ソフトウェアがフェデレーテッド・サーバー上で適切にセットアップされていることを検査します。

始める前に

Basic Teradata Query (BTEQ) ユーティリティーおよび Teradata Data Connector Application Program Interface (PIOM) が、フェデレーテッド・サーバーにインストールされていることを確認します。BTEQ ユーティリティーと Teradata Data Connector Application Program Interface は、Teradata クライアント・ソフトウェアのインストール時にフェデレーテッド・サーバーにインストールされます。

このタスクについて

BTEQ ユーティリティーを使用して SQL 照会をサブミットし、フェデレーテッド・サーバーが Teradata サーバーに接続できることを検査します。BTEQ ユーティリティーの詳細については、Teradata 資料を参照してください。

手順

1. BTEQ ユーティリティー・セッションを開始し、Teradata サーバーにログオンします。
2. SQL コマンドを発行して、Teradata サーバーに正常に接続できることを検査します。

例:

```
select count(*) from dbc.tables;
```

正常に接続した場合には、照会の出力が表示されます。

接続が不成功の場合は、エラーが出されます。Teradata クライアント・ソフトウェアをチェックして、フェデレーテッド・サーバー上に適切にインストールされて構成されていることをチェックしてください。

3. Teradata サーバーからログオフし、BTEQ ユーティリティー・セッションを終了します。

次のタスク

この作業を完了したら、Teradata ライブラリーがランタイム・リンクに対応していることを検査します。

Teradata ライブラリーがランタイム・リンクに対応しているかどうかの検査 (AIX)

Teradata データ・ソースを AIX 上のフェデレーテッド・サーバーに追加する場合、ラッパーまたはサーバーを登録する前に、ランタイム・リンクに対応しているかどうかを検査する必要があります。

手順

1. libcliv2.so ファイルが存在するディレクトリーに移動します。

libcliv2.so ファイルは、Teradata クライアント・ソフトウェアでインストールされます。デフォルトでは、/usr/lib ディレクトリーにインストールされます。

2. ランタイム・リンクが使用可能になっているかどうか検査するには、コマンド・プロンプトから次の UNIX コマンドを発行します。

```
dump -H libcliv2.so | grep libtli.a
```

3. 戻されるファイル名をチェックします。libtli.a というファイル名が戻される場合、Teradata ライブラリーはランタイム・リンクに対応しています。

libtli.a というファイル名が戻されない場合、コマンド・ウィンドウを開いて次の UNIX コマンドを発行し、Teradata ライブラリーについてランタイム・リンクを使用可能にします。

```
rtl_enable libcliv2.so -F libtli.a
mv libcliv2.so libcliv2.so.old
mv libcliv2.so.new libcliv2.so
chmod a+r libcliv2.so
```

次のタスク

この作業が完了したら、環境変数を設定することができます。

Teradata 環境変数の設定

フェデレーテッド・サーバー上の db2dj.ini ファイルで、Teradata 環境変数を設定する必要があります。

このタスクについて

制約事項

db2dj.ini ファイルに関する制約事項を確認してください。

db2dj.ini ファイルには、フェデレーテッド・サーバーにインストールされている Teradata クライアント・ソフトウェアに関する構成情報が入っています。

Teradata データ・ソースについて、必須の環境変数とオプションの環境変数があります。

Teradata ラッパーをインストールする前に Teradata クライアント・ソフトウェアをインストールした場合、必要な Teradata 環境変数は db2dj.ini ファイルで設定されています。

Teradata ラッパーをインストールする前に Teradata クライアント・ソフトウェアをインストールしなかった場合や、いずれかのオプションの環境変数を設定する場合は、この作業のステップを使用して環境変数を設定する必要があります。

手順

1. 次の方法のいずれかを使用します。

方法	ステップ
自動的に環境変数を設定します。	IBM InfoSphere Federation Server インストール・ウィザードを実行します。ウィザードの指示に従ってください。 重要: インストール・ウィザードを実行することによって、必須の環境変数を設定します。オプションの環境変数は手動で設定する必要があります。
ラッパー構成ツールを使用して環境変数を設定します。	ラッパー構成ツールを実行して、必須の環境変数とオプションの環境変数を設定します。1 ページの『ラッパー構成ツールの実行』の説明に従ってください。

方法	ステップ
手動で環境変数を設定します。	<p>db2dj.ini ファイルを編集します。</p> <p>db2dj.ini ファイルは、DB2 レジストリー変数 DB2_DJ_INI によって指定されたディレクトリー内にあります。DB2_DJ_INI 変数が設定されていない場合、db2dj.ini ファイルは、オペレーティング・システムに応じて以下のデフォルト・パスのいずれかにあります。</p> <ul style="list-style-type: none"> • UNIX の場合: <i>instancehome</i>/sql1lib/cfg/db2dj.ini <i>instancehome</i> インスタンス所有者のホーム・ディレクトリーです。 • Windows の場合: %DB2PATH%\%cfg%\db2dj.ini %DB2PATH% DB2 データベース・システムがインストールされているディレクトリー (例えば、C:\Program Files\IBM\sql1lib) です。 <p>このファイルが存在しない場合は、任意のテキスト・エディターを使用して、db2dj.ini という名前のファイルを作成することができます。db2dj.ini ファイルには、環境変数の値に完全修飾パスを指定する必要があります。そうしないと、エラーが発生します。次に例を示します。</p> <p>COPLIB=/usr/lib</p>

2. Teradata コード・ページ変換環境変数を (必要に応じて) 設定します。
3. フェデレーテッド・サーバーに環境変数を設定するには、次のコマンドを使用してフェデレーテッド・データベース・インスタンスをリサイクルします。

```
db2stop
db2start
```

次のタスク

この作業が完了したら、ラッパーを登録することができます。

Teradata 環境変数

Teradata データ・ソースについて、必須の環境変数とオプションの環境変数があります。これらの変数は db2dj.ini ファイルで設定されます。

Teradata では、以下の環境変数が有効です。

- COPLIB
- COPERR
- TERADATA_CHARSET (オプション)
- NETRACE (オプション)
- COPANOMLOG (オプション)

変数の説明

COPLIB

CLI システム・パラメーター・ブロック・ファイル clispb.dat のフェデレ

ーテッド・サーバー上のディレクトリー・パスを指定します。COPLIB 変数の絶対パスを指定する必要があります。clispb.dat ファイルの Teradata CLI インストール・ディレクトリー /usr/lib です。このファイルのディレクトリー・パスを変更する場合は、この変数を必ず更新してください。

```
COPLIB=/usr/lib
```

64 ビット・クライアントの場合、djaxlinkTeradata.sh を実行する必要があるが、ライブラリーがデフォルト・ロケーションにない場合、TERADATA_LIB_DIR 環境変数を 64 ビット・クライアント・ライブラリーのディレクトリー・ロケーションに設定する必要があります。オペレーティング・システムごとのライブラリーのデフォルト・ロケーションは以下のとおりです。

- AIX: /usr/lib/lib_64/libcliv2.so
- HP-UX:
 - PA-RISC 64 ビット: /usr/lib/pa20_64/libcliv2.sl
 - Itanium プロセッサー・ファミリー (IPF) 64 ビット: /usr/lib/hpux64/libcliv2.so
- Linux: /usr/lib64/libcliv2.so
- Solaris-SPARC: /usr/lib/sparcv9/libcliv2.so

clispb.dat とエラー・メッセージ・カタログ errmsg.cat は通常、両方とも同じディレクトリーにあります。

COPERR

フェデレーテッド・サーバーにおける errmsg.cat ファイルのディレクトリー・パスを指定します。COPERR 変数の完全修飾パスを指定します。

例:

```
COPERR=/usr/lib
```

TERADATA_CHARSET

Teradata データ・ソースで使用するコード・ページ文字セットを指定します。

フェデレーテッド・サーバーが Teradata データ・ソースに接続するたび、Teradata ラッパーは、その接続に使用するコード・ページ文字セットを判別します。Teradata ラッパーにコード・ページ文字セットを設定させるか、または TERADATA_CHARSET 環境変数を設定してコード・ページを指定することができます。

TERADATA_CHARSET 環境変数がフェデレーテッド・サーバー上の db2dj.ini ファイルに設定されている場合、ラッパーは db2dj.ini ファイルのコード・ページ文字セットを使用します。TERADATA_CHARSET 環境変数の値に対して妥当検査は行われませんが、この環境変数が有効な値に設定されないと、Teradata データ・ソースはエラーを戻します。

TERADATA_CHARSET 環境変数がフェデレーテッド・サーバー上の db2dj.ini ファイルで設定されていない場合、ラッパーはデータベースのコード・ページに基づいてクライアント文字セットを検出します。

UNIX が稼働するフェデレーテッド・サーバーの場合は、TERADATA_CHARSET 環境変数で以下の値が有効です。

- HANGULKSC5601_2R4
- KanjiEUC_0U
- LATIN1_0A
- LATIN9_0A
- LATIN1252_0A
- SCHGB2312_1T0
- TCHBIG5_1R0
- UTF8
- ASCII

Windows が稼働するフェデレーテッド・サーバーの場合は、TERADATA_CHARSET 環境変数で以下の値が有効です。

- HANGULKSC5601_2R4
- KanjiSJIS_0S
- LATIN1_0A
- LATIN1252_0A
- SCHGB2312_1T0
- TCHBIG5_1R0
- UTF8
- ASCII

NETRACE

オプション。Teradata クライアント・ソフトウェアのトレース機能を使用可能にします。この変数はデバッグにのみ必要です。

COPANOMLOG

オプション。Teradata クライアント・ソフトウェアのロギング機能を使用可能にします。この変数はデバッグにのみ必要です。

Teradata サーバー上の文字セットの検査

Teradata サーバー上で適切な文字セットが指定されていないと、接続エラーが発生することがあります。使用する文字セットが Teradata サーバーにインストールされていることを検査してください。

このタスクについて

手順

使用する文字セットが Teradata サーバーにインストールされていることを検査するには、以下のようにします。

手順

1. BTEQ ユーティリティー、または他の有効なログオン・ユーティリティーを使用して、Teradata サーバーにログオンします。
2. 以下のステートメントを発行して、dbc.chartranslations 表を表示します。


```
select * from dbc.chartranslations;
```

- 戻される表の 3 列目 InstallFlag の値をチェックします。3 列目の値 'Y' は、Teradata サーバーで文字セットがインストールされて使用されていることを示します。

以下の表を使用して、適切な文字セットがインストールされているかどうかを判別してください。

表 37. Teradata 用の文字セット

2 バイト文字セット	1 バイト文字セット	Teradata 文字セット	言語	IBM DB2 コード・セット
941	897	"KanjiSJIS_0S"	日本語	IBM-943
1362	1126	"HANGULKSC5601_2R4"	ハングル	1363
1385	1114	"SCHGB2312_1T0"	中国語 (簡体字)	GBk
380	1115	"SCHGB2312_1T0"	中国語 (簡体字)	IBM-1381
947	1114	"TCHBIG5_1R0"	中国語 (繁体字)	big5
1200	1208	"UTF8"	Unicode	UTF-8
0	819	"Latin1_0A"	英語 (Latin 1)	ISO8859-1
0	1252	"Latin1252_0A"	英語 (Windows Latin)	ISO8859-1/15
0	819	"ASCII"	英語 (ASCII)	ISO8859-1

- 必要な文字セットがインストールされていない場合、Teradata ラッパーを使用するためにその文字セットをインストールします。
 - ASCII は Teradata サーバーで使用可能になっていますが、dbc.chartranslations 表にはカタログされません。InstallFlag について戻される値がすべて 'N' である場合は、ASCII が Teradata サーバー上で有効な唯一の文字セットとなり、TERADATA_CHARSET 環境変数は db2dj.ini ファイルで ASCII に設定されていなければなりません。
 - 使用する文字セットが dbc.chartranslations 表にリストされているが、InstallFlag 値が 'N' に設定されている場合は、以下のステートメントを発行して、InstallFlag を 'Y' に変更します。


```
update dbc.chartranslations
set installflag='Y' where CharSetName= 'character_set_name';
```
 - 使用する文字セットが dbc.chartranslations 表にリストされていない場合は、Teradata お客様サポートに連絡してください。
- Teradata サーバーを再始動して、文字セットのリストを更新します。Teradata コマンド・ウィンドウに次のように入力してください。 `tpareset -f reason_for_restart`

Teradata データ・ソースの文字セットのトラブルシューティング:

Teradata データ・ソース用に TERADATA_CHARSET 環境変数を設定するとき、適切な文字セットが指定されていないとエラーが発生することがあります。

症状

Teradata データ・ソースについて適切な文字セットが指定されていないと、以下のエラーが発生します。

SQL 1822N 予期しないエラー・コード "227"
をデータ・ソース "<string>" から受け取りました。
関連したテキストとトークンは
"MTDP: EM_CHARNAME(227) : invalid character set name specif" です。
SQLSTATE=560BD

原因

TERADATA_CHARSET 環境変数で指定されている文字セットが誤っています。

問題の解決方法

Teradata サーバーと db2dj.ini ファイルで適切な文字セットがインストールされ、指定されていることを検査します。

Teradata ラッパーの登録

Teradata データ・ソースにアクセスするためには、ラッパーを登録する必要があります。フェデレーテッド・サーバーは、ラッパーを使用してデータ・ソースとの通信やデータ・ソースからのデータの取得を行います。ラッパーは、ライブラリー・ファイルのセットとしてインプリメントされます。

手順

コマンド行から CREATE WRAPPER ステートメントを発行し、Teradata ラッパーのデフォルト名を指定します。

例:

```
CREATE WRAPPER TERADATA;
```

要確認: デフォルト名 TERADATA を使用してラッパーを登録する場合、フェデレーテッド・サーバーは、フェデレーテッド・サーバーが実行されているオペレーティング・システムに適した Teradata ラッパー・ライブラリーを自動的に使用します。

デフォルトのラッパー名がフェデレーテッド・データベース内の既存のラッパー名と競合する場合、デフォルトのラッパー名を、指定の名前で置き換えることができます。デフォルト名とは別の名前を使用する場合は、CREATE WRAPPER ステートメントに LIBRARY パラメーターを含める必要があります。

例えば、AIX を使用するフェデレーテッド・サーバーで、tera_wrapper という名前のラッパーを登録するには、以下のステートメントを発行します。

```
CREATE WRAPPER tera_wrapper  
  LIBRARY 'libdb2teradata.a'  
  OPTIONS (DB2_FENCED 'Y');
```

指定するラッパー・ライブラリー・ファイルは、フェデレーテッド・サーバーのオペレーティング・システムによって異なります。

AIX および Solaris における必須オプション

注: Teradata ラッパーは 32 ビット・クライアントしかサポートしていないため、AIX と Solaris では、CREATE WRAPPER ステートメントの DB2_FENCED ラッパー・オプションが必須です。

例:

```
CREATE WRAPPER TERADATA
  OPTIONS (DB2_FENCED 'Y')
```

次のタスク

この作業が完了したら、サーバー定義を登録することができます。

Teradata ラッパー・ライブラリー・ファイル

Teradata ラッパー・ライブラリー・ファイルは、ラッパー のインストール時にフェデレーテッド・サーバーに追加されます。

Teradata ラッパー をインストールすると、ディレクトリー・パスに 3 つのライブラリー・ファイルが追加されます。例えば、フェデレーテッド・サーバーが AIX 上で実行されている場合、ディレクトリー・パスに追加されるラッパー・ライブラリー・ファイルは `libdb2teradata.a`、`libdb2teradataF.a`、および `libdb2teradataU.a` です。デフォルトのラッパー・ライブラリー・ファイルは `libdb2teradata.a` です。他のラッパー・ライブラリー・ファイルは Teradata ラッパーの内部で使用されます。

ラッパーの登録時にデフォルトのラッパー名を使用しない場合は、`CREATE WRAPPER` ステートメントに `LIBRARY` パラメーターを含め、デフォルトのラッパー・ライブラリー・ファイルの名前を指定する必要があります。

以下の表に、デフォルトのディレクトリー・パスとデフォルトのラッパー・ライブラリー・ファイルの名前をリストします。

表 38. Teradata ラッパー・ライブラリーのロケーションとファイル名

オペレーティング・システム	ディレクトリー・パス	ライブラリー・ファイル名
AIX	<code>/usr/opt/install_path/lib32/</code> <code>/usr/opt/install_path/lib64/</code>	<code>libdb2teradata.a</code>
Solaris	<code>/opt/IBM/db2/install_path/lib32</code> <code>/opt/IBM/db2/install_path/lib64</code>	<code>libdb2teradata.so</code>
Windows	<code>%DB2PATH%\%bin</code>	<code>db2teradata.dll</code>

`install_path` は、UNIX 上でフェデレーテッド・サーバーがインストールされているディレクトリー・パスです。

Teradata データ・ソースのサーバー定義の登録

フェデレーテッド・データベースでアクセスする各 Teradata サーバーを登録する必要があります。

手順

1. `hosts` ファイルを見つけます。
 - AIX が稼働するフェデレーテッド・サーバーの場合、`hosts` ファイルは `/etc/hosts` ディレクトリーにあります。
 - Windows が稼働するフェデレーテッド・サーバーの場合、`hosts` ファイルは `%WINDIR%\system32\drivers\etc\hosts` ディレクトリーにあります。
2. `hosts` ファイル内で、Teradata ノード項目のサーバー名を見つけます。

hosts ファイル内でサーバー名は、*dbcnameCOPn* のようにサフィックス付きでと共にリストされています。

サーバー名 *dbcname* は、CREATE SERVER ステートメントの *node_name* 変数として指定します。サーバー名 *dbcname* を見つけるには、以下の Teradata ノード項目の構文を使用してください。

構文

```
a.b.c.d host_name dbcnameCOPn aliasnameCOPn
```

パラメーター

a.b.c.d

Teradata サーバーの IP アドレス。

host_name

オプション。Teradata サーバーのホスト名。

dbcnameCOPn

サーバー名および Teradata Communication Processor (COP) グループ内のアプリケーション・プロセッサの番号。複数の Teradata サーバーと単一のデータベース・インスタンスを含むシステムでは、サーバーは COP グループと呼ばれます。COP グループ内の各サーバーは、1 から始まる固有のシーケンス番号を割り当てられています。例えば、Teradata サーバーが 1 つだけ存在する場合は、サーバー名とアプリケーション・プロセッサの番号は *dbcnameCOP1* になります。

dbcname

Teradata サーバーの名前。この名前の最大長は 8 文字です。COP グループ内のサーバーについては、サーバーの *dbcname* は同じで、*COPn* 値が順番に付きます。

n アプリケーション・プロセッサの番号。

alias_nameCOPn

オプション。*dbcname* の代替名および COP グループ内のアプリケーション・プロセッサの番号。アプリケーション・プロセッサの番号は、サーバー名 *dbcnameCOPn* について存在するのと同じ番号です。

例 このサンプル hosts ファイルでは、*nodexyz* がサーバー名で、CREATE SERVER ステートメントの *node_name* 変数としてこれを指定する必要があります。

```
127.0.0.1      localhost
9.22.5.77     nodexyz      nodexyzCOP1    # teradata server
9.66.111.133  rtplib05.data.xxx.com aap
9.66.111.161  rtpscm11.data.xxx.com aaprwr
9.66.111.161  rtpscm11.data.xxx.com accessm
```

3. サーバーを作成するために、コマンド行から CREATE SERVER ステートメントを発行します。

例:

```
CREATE SERVER server_definition_name
  TYPE TERADATA
  VERSION version_number
  WRAPPER wrapper_name
  OPTIONS (NODE 'node_name');
```

node_name 変数の指定は CREATE SERVER ステートメントでオプションですが、このオプションは Teradata データ・ソースの場合は必須です。

CREATE SERVER ステートメント - Teradata ラッパーの例

Teradata ラッパーにサーバー定義を登録するには、CREATE SERVER ステートメントを使用します。必須パラメーターが含まれる完全な例、およびその他のサーバー・オプションが含まれる例が、このトピックに組み込まれています。

次に、CREATE SERVER ステートメントを実行して CREATE SERVER ラッパーのサーバー定義を登録する方法の例を示します。

```
CREATE SERVER tera_server TYPE TERADATA
  VERSION 2.5 WRAPPER my_wrapper
  OPTIONS (NODE 'tera_node');
```

tera_server

Teradata データベース・サーバーに割り当てる名前。重複するサーバー定義名は使用できません。

TYPE TERADATA

アクセスを構成するデータ・ソース・サーバーのタイプを指定します。

Teradata ラッパーの場合、サーバー・タイプは TERADATA でなければなりません。

VERSION 2.5

これは、アクセスしたい Teradata データベース・サーバーのバージョンです。

WRAPPER TERADATA

CREATE WRAPPER ステートメントで指定したラッパー名。

NODE '*tera_node*'

Teradata データベース・サーバーが存在するノードの名前。ノード名は hosts ファイルから入手します。この値は大文字と小文字を区別します。

ノード名は CREATE SERVER ステートメントではオプションとなりますが、CREATE SERVER データ・ソースの場合は必要です。

サーバー・オプション

サーバー定義の作成時に、CREATE SERVER ステートメントに追加のサーバー・オプションを指定することができます。サーバー・オプションには、汎用のサーバー・オプションと Teradata に特有のサーバー・オプションを指定できます。

CPU_RATIO と IO_RATIO のサーバー・オプションは、照会最適化ライブラリに Teradata サーバーに関する統計情報を提供しています。フェデレーテッド・サーバーの CPU リソースが Teradata サーバーの CPU リソースの 2 倍の能力を持つことを指定するには、CPU_RATIO サーバー・オプションの値を 2.0 に設定します。

フェデレーテッド・サーバーの入出力装置が Teradata サーバーの入出力装置に比べて 3 倍の速度でデータを処理することを指定するには、IO_RATIO サーバー・オプションを 3.0 に設定します。

次の例は、これらのオプションを指定した Teradata サーバー定義です。

```
CREATE SERVER tera_server TYPE TERADATA
  VERSION 2.5 WRAPPER my_wrapper
  OPTIONS (NODE 'tera_node', CPU_RATIO '2.0', IO_RATIO '3.0');
```

Teradata データ・ソースのユーザー・マッピングの作成

Teradata サーバーにアクセスする場合、フェデレーテッド・サーバーは、データ・ソースに対して有効なユーザー ID とパスワードを使用して、Teradata サーバーへの接続を確立します。

このタスクについて

フェデレーテッド・システムにアクセスする、各ユーザー ID のユーザー・マッピングを作成して、Teradata データ・ソースに分散要求を送信します。

手順

CREATE USER MAPPING ステートメントを発行します。

例えば、次のようにします。

```
CREATE USER MAPPING FOR local_userID SERVER server_definition_name
  OPTIONS (REMOTE_AUTHID 'remote_userID', REMOTE_PASSWORD 'remote_password');
```

REMOTE_AUTHID および REMOTE_PASSWORD 変数は CREATE USER MAPPING ステートメントではオプションとして指定されますが、Teradata データ・ソースにアクセスするには必須です。

次のタスク

この作業を完了したら、フェデレーテッド・サーバーから Teradata サーバーへの接続をテストします。

CREATE USER MAPPING ステートメント - Teradata ラッパーの例

CREATE USER MAPPING ステートメントを使用して、フェデレーテッド・サーバーの許可 ID をリモートの Teradata ユーザー ID およびパスワードにマップします。このトピックには、必須パラメーターを使用する完全指定の例と、CREATE USER MAPPING ステートメントで DB2 特殊レジスター USER を使用する方法を示す例が含まれます。

以下の例は、ローカルの許可 ID をリモートの Teradata ユーザー ID およびパスワードにマップする方法を示しています。

```
CREATE USER MAPPING FOR MICHAEL SERVER tera_server
  OPTIONS (REMOTE_AUTHID 'mike', REMOTE_PASSWORD 'passxyz123');
```

MICHAEL

Teradata サーバーで定義されたリモート・ユーザー ID およびパスワードにマップするローカルの許可 ID を指定します。

SERVER *tera_server*

CREATE SERVER ステートメントで Teradata サーバーについて登録したサーバー定義名を指定します。

REMOTE_AUTHID *'mike'*

MICHAEL をマップする、リモートの Teradata ユーザー ID を指定します。CREATE SERVER ステートメントで FOLD_ID サーバー・オプションを「U」または「L」に設定している場合を除き、この値は大文字小文字を区別します。

リモート・ユーザー ID は CREATE SERVER ステートメントではオプションとして指定されていますが、Teradata データ・ソースには必須です。

REMOTE_PASSWORD *'passxyz123'*

'mike' に関連するリモートの Teradata パスワードを指定します。CREATE SERVER ステートメントで FOLD_PW サーバー・オプションを「U」または「L」に設定している場合を除き、この値は大文字小文字を区別します。

リモート・パスワードは CREATE SERVER ステートメントではオプションとして指定されていますが、Teradata データ・ソースには必須です。

DB2 特殊レジスター USER

DB2 特殊レジスター USER を使用して、CREATE USER MAPPING ステートメントを実行しているユーザーの許可 ID を、REMOTE_AUTHID ユーザー・オプションに指定されているデータ・ソースの許可 ID にマップすることができます。

以下に、特殊レジスター USER を含む CREATE USER MAPPING ステートメントの例を示します。

```
CREATE USER MAPPING FOR USER SERVER tera_server
  OPTIONS (REMOTE_AUTHID 'mike', REMOTE_PASSWORD 'passxyz123');
```

Teradata サーバーへの接続のテスト

Teradata データ・ソース・サーバーへの接続をテストして、フェデレーテッド・サーバーが Teradata データ・ソースにアクセスするように適切に構成されているかどうかを判別します。

このタスクについて

Teradata サーバーへの接続のテストは、定義したサーバー定義とユーザー・マッピングを使用して行うことができます。

手順

パススルー・セッションを開き、Teradata システム表に対して SELECT ステートメントを発行します。SELECT ステートメントがカウントを戻す場合、サーバー定義およびユーザー・マッピングは適切にセットアップされています。

例:

```
SET PASSTHRU server_definition_name
SELECT count(*) FROM dbc.tables
SET PASSTHRU RESET
```

SELECT ステートメントがエラーを戻す場合には、接続エラーのトラブルシューティングを行う必要があります。

次のタスク

この作業を完了したら、Teradata の表およびビューのニックネームを登録することができます。

データ・ソース接続エラーのトラブルシューティング

データ・ソース・サーバーへのテスト接続では、いくつかの理由でエラーが戻されることがあります。以下の処置により、エラーが発生した理由を判別することができます。

症状

データ・ソースに接続しようとしたときにエラーが戻されます。

原因

接続の問題には、いくつかの原因が考えられます。

問題の解決方法

データ・ソース接続エラーのトラブルシューティングを行うには、以下の項目で問題がないかどうかをチェックします。

- データ・ソースが使用可能であることを確認します。
- 該当する場合は、データ・ソース・サーバーが着信接続に対して構成されていることを確認します。
- REMOTE_AUTHID と REMOTE_PASSWORD のオプションに関するユーザー・マッピング設定を調べ、データ・ソースとの接続に有効な設定になっているか確認します。必要に応じて、ユーザー・マッピングを変更するか、別のユーザー・マッピングを作成します。
- 該当する場合は、フェデレーテッド・サーバー上のデータ・ソース・クライアント・ソフトウェアがインストールされていて、データ・ソースに接続できるように正しく構成されていることを確認します。
- ODBC データ・ソースの場合、フェデレーテッド・サーバー上の ODBC ドライバーがインストールされていて、ODBC データ・ソース・サーバーに接続できるように正しく構成されていることを確認します。Windows が稼働するフェデレーテッド・サーバーでは、ドライバーをチェックするときに「ODBC データ ソース アドミニストレータ」ツールを使用します。UNIX が稼働するフェデレーテッド・サーバーの場合は、ODBC クライアントのベンダーが提供している資料を参照してください。
- フェデレーテッド・サーバー上で設定されている変数の設定がデータ・ソースについて適切であることを確認します。これらの変数には、システム環境変数、db2dj.ini ファイル、および DB2 Profile Registry (db2set) 変数が含まれます。
- サーバー定義を調べます。必要であれば、サーバー定義をドロップして再作成します。

Teradata 表およびビューのニックネームの登録

登録するそれぞれの Teradata サーバー定義で、アクセスしたいそれぞれの表またはビューごとにニックネームを登録する必要があります。Teradata サーバーを照会する時には、データ・ソース・オブジェクトの名前ではなく、これらのニックネームを使用します。

始める前に

フェデレーテッド・データベースは、データ・ソースのカatalog統計に基づいて、照会処理を最適化します。フェデレーテッド・データベースが Teradata 表の完全な統計を持つようにするために、ニックネームを登録する前に **COLLECT STATISTICS** コマンドを使用します。

Teradata サーバーから、**COLLECT STATISTICS** Teradata コマンドを使用して、表内の 1 つ以上の列または索引の統計を収集します。

CREATE NICKNAME ステートメントを使ってニックネームを登録する場合、フェデレーテッド・データベースは Teradata システム・カatalogから統計を読み取り、ニックネームのローカル統計を更新します。

このタスクについて

Teradata ビューに対してニックネームを登録すると、フェデレーテッド・データベースはビューのすべての列を (Teradata ビューの列が NULL 値を許容しない場合でも) NULL 可能として認識します。この制限には次善策がありません。

手順

コマンド行から **CREATE NICKNAME** ステートメントを発行します。

例:

```
CREATE NICKNAME nickname  
FOR server_definition_name.remote_schema.remote_table;
```

ニックネームの長さは 128 文字までです。

ニックネームを作成すると、フェデレーテッド・サーバーはニックネームを使用してデータ・ソース・カatalogを照会します。この照会は、データベース・ソースの表またはビューへの接続をテストします。接続ができない場合、エラー・メッセージが出ます。

次のタスク

ニックネームを作成したい Teradata の表またはビューごとに、このステップを繰り返します。

フェデレーテッド・サーバーでの Teradata ニックネーム

フェデレーテッド・サーバーから Teradata データ・ソースを照会するときは、照会でニックネームを使用して、アクセスする Teradata の表およびビューを識別します。

Teradata の表またはビューのニックネームを作成すると、フェデレーテッド・サーバーはその表またはビューに関連した Teradata サーバーに接続します。フェデレー

テッド・サーバーはニックネームを使用して、Teradata サーバーへの接続を検査します。フェデレーテッド・データベースはデータ・ソース側にその表またはビューが存在するか検査し、次に Teradata サーバー上のカタログから、その Teradata の表またはビューに関する統計データを収集しようとしています。ニックネームで示されるオブジェクトに関して収集された統計は、フェデレーテッド・サーバー上のグローバル・カタログに保管されます。

フェデレーテッド・サーバーは、照会処理を最適化するのに、ニックネームで示されるオブジェクトについて収集した統計に依存します。照会オプティマイザーは Teradata カタログ情報の一部またはすべてを使用することがあるため、ニックネームを作成する前に、Teradata サーバー側で統計データを更新してください。Teradata サーバー側で統計データを更新するには、DB2 の RUNSTATS コマンドと等価のコマンドまたはユーティリティを使用してください。

更新可能な Teradata ビューを参照するニックネームに対する INSERT、UPDATE、または DELETE ステートメントは、その SQL ステートメントを Teradata データ・ソースに対して完全にプッシュダウンできるのでない限りサブミットできません。

CREATE NICKNAME ステートメント - Teradata ラッパーの例

CREATE NICKNAME ステートメントを使用して、アクセスする Teradata の表またはビューのニックネームを登録します。このトピックには、必須パラメーターを使用する完全指定の例が含まれています。

この例は、Teradata サーバー上で Teradata の表またはビューのニックネームを作成する方法を示します。

```
CREATE NICKNAME TERASALES FOR tera_server.vinnie.europe ;
```

TERASALES

Teradata の表またはビューを識別するために使用する固有なニックネームです。

重要: ニックネームは、2 つの部分 (スキーマとニックネーム) からなる名前です。ニックネームの登録時にスキーマを省略すると、そのニックネームのスキーマはニックネームを登録するユーザーの許可 ID になります。これはフェデレーテッド・サーバーの許可 ID であり、リモート Teradata データ・ソースの許可 ID ではありません。

tera_server.vinnie.europe

リモート・オブジェクトを表す、3 つの部分からなる ID。

- *tera_server* は、CREATE SERVER ステートメントで Teradata データベース・サーバーに割り当てたサーバー定義名です。
- *vinnie* は、表またはビューが属する所有者のユーザー ID です。この値は大文字と小文字を区別します。
- *europe* は、アクセスしたいリモートの表またはビューの名前です。

Teradata データ・ソース構成のトラブルシューティング

libcliv2.so (AIX) のランタイム・リンクの使用可能化

djxlinkTeradata.sh ファイルを実行して libcliv2.so と呼ばれる Teradata 共有ライブラリーにリンクした場合、CREATE NICKNAME ステートメントを実行するとエラー・メッセージを受け取ることがあります。

症状

以下は、受け取るエラー・メッセージの例です。

```
DB21034E コマンドが、有効なコマンド行プロセッサ・コマンドでないため、
SQL ステートメントとして処理されました。
SQL 処理中に、そのコマンドが返されました。
SQL30081N 通信エラーが検出されました。
使用されている通信プロトコル: "TCP/IP"。使用されている通信 API: "SOCKETS"。
エラーが検出されたロケーション: "9.112.26.28"。
エラーを検出した通信関数: "recv"。
プロトコル固有のエラー・コード: "*", "*", "0"。
SQLSTATE=08001
```

原因

OsCa11 関数によってフェデレーテッド・サーバーが停止されました。

問題の解決方法

エラー・メッセージを受け取る場合、/sql1lib/db2dump ディレクトリーにトラップ・ファイルがないかどうかをチェックしてください。トラップ・ファイルは、文字 t で始まり 000 という接尾部で終わる名前が付いています。次に例を示します。

```
t123456.000
```

トラップ・ファイルのトレース情報で、OsCa11 関数がフェデレーテッド・サーバーを停止させたことを示す OsCa11 関数参照がないかどうかをチェックしてください。

以下に、トラップ・ファイル内で見つかる可能性のある OsCa11 関数参照を含むトレース情報の例を示します。

```
*** Start stack traceback ***
```

```
0x239690E0 OsCa11 + 0x28C
0x23973FB0 mtdpassn + 0x8A4
0x239795A4 mtdp + 0x208
0x2395A928 MTDP10 + 0x28C
0x239609C4 CLICON + 0xD50
0x23962350 DBCHCL + 0xC4
```

いずれかのトラップ・ファイルに OsCa11 関数参照が見つかる場合には、libcliv2.so 共有ライブラリーについてランタイム・リンクを使用可能にする必要があります。

ランタイム・リンクを使用可能にするには、次の UNIX コマンドを発行します。

```
cd /usr/lib
rtl_enable libcliv2.so -F libtli.a
mv libcliv2.so libcliv2.so.old
mv libcliv2.so.new libcliv2.so
chmod a+r libcliv2.so
```

Teradata データ・ソースの文字セットのトラブルシューティング

Teradata データ・ソース用に TERADATA_CHARSET 環境変数を設定するとき、適切な文字セットが指定されていないとエラーが発生することがあります。

症状

Teradata データ・ソースについて適切な文字セットが指定されていないと、以下のエラーが発生します。

```
SQL 1822N  予期しないエラー・コード "227"
をデータ・ソース "<string>" から受け取りました。
関連したテキストとトークンは
"MTDP: EM_CHARNAME(227) : invalid character set name specif" です。
SQLSTATE=560BD
```

原因

TERADATA_CHARSET 環境変数で指定されている文字セットが誤っています。

問題の解決方法

Teradata サーバーと db2dj.ini ファイルで適切な文字セットがインストールされ、指定されていることを検査します。

ニックネームに対する UPDATE または DELETE 操作エラーのトラブルシューティング

デフォルトでは、Teradata データ・ソース表で行が固有に識別されることはありません。 Teradata 表またはビューに関連するニックネームを更新または削除しようとすると、SQL エラー・メッセージを受け取る場合があります。

症状

Teradata 表またはビューに関連するニックネームを更新または削除しようとすると、SQL30090N、RC="21" エラーが戻されます。

原因

この問題は、Teradata ソース表で行が一意的に定義されていないことに起因します。

問題の解決方法

この問題を解決するには、以下のようにします。

1. ニックネームをドロップし、再作成します。
2. 更新または削除する Teradata 表で 1 つ以上のユニーク索引を作成し、操作を再び試行してください。

Teradata アクセス・ロギングを使用した操作

Teradata 製品では、Teradata が 1 つ以上のデータベースでさまざまなユーザーの特定のセキュリティ特権をチェックする際に、ログ項目を生成するアクセス・ロギング機能が備えられています。アクセス・ロギングではかなりの量の重要なセキュリティ情報が提供されますが、この機能によりプロセッサの使用量が著しく増加し、システムのパフォーマンスが低下する可能性があります。

システムのパフォーマンスを改善することが必要な場合には、アクセス・ロギングに対して定義したチェック特権規則を評価してください。それから、END LOGGING ステートメントを定義して、不必要な規則を終了します。

最高のパフォーマンスを得るには、すべてのアクセス・ロギングをオフにします。Teradata DBC.AccLogRules マクロをドロップしてから、トラステッド・パラレル・アプリケーション (TPA) を強制リセットして、アクセス・ロギングを完全に停止します。

詳細については、Teradata 資料を参照してください。

Web サービス・データ・ソースへのアクセスの構成

フェデレーテッド・システムが Web サービス・データ・ソースにアクセスするように構成するには、Web サービス記述言語 (WSDL) 文書などの、アクセスしたいデータ・ソースとオブジェクトの情報をサーバーに提供する必要があります。

始める前に

- フェデレーションが、フェデレーテッド・サーバーとして機能するサーバー上にインストールされている必要があります。
- データベースがフェデレーテッド・サーバー上に存在している必要があります。

手順

1. Web サービス・ラッパーを登録します。
2. Web サービス・データ・ソースのサーバー定義を登録します。
3. ユーザー・マッピングを登録して、HTTP 認証のためのセキュリティを有効にします (オプション)。
4. Web サービス・データ・ソースのニックネームの登録
5. Web サービス・ニックネームのフェデレーテッド・ビューを作成します。

Web サービスおよび Web サービス・ラッパー

Web サービス・プロバイダーは、Web サービス記述言語 (WSDL) 文書によって記述されています。Web サービス・ラッパーを使用して Web サービス・プロバイダーにアクセスできます。

240 ページの図 12 のダイアグラムは、Web サービスのアーキテクチャーを示しています。

1. Web サービス・プロバイダーはサービスをインプリメントし、WSDL 情報を UDDI などのサービス・ブローカーへ公開します。

2. サービス・コンシューマーはサービス・ブローカーを使用して、Web サービス・プロバイダーを検索することができます。
3. サービス・コンシューマーが Web サービス・プロバイダーを検出する場合、そのサービス・コンシューマーはサービス・プロバイダーにバインドして、Web サービスを使用できるようにします。
4. コンシューマーは、リクエスターとプロバイダー間で SOAP (Simple Object Access Protocol) メッセージを交換することによりサービスを呼び出します。

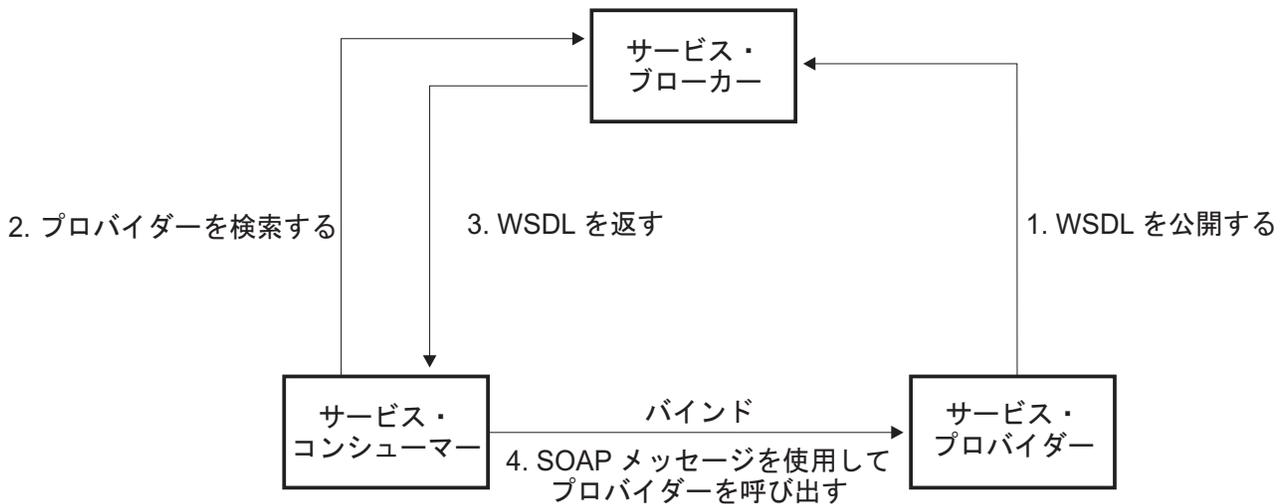


図 12. Web サービス: サービス指向アーキテクチャ

SOAP 仕様は XML に基づいたメッセージのレイアウトを定義します。SOAP メッセージは SOAP エンベロープに含まれています。エンベロープはオプションの SOAP ヘッダーと必須の SOAP 本体から成っています。SOAP ヘッダーには暗号化情報あるいは認証情報といったメッセージに関する情報が含まれます。SOAP 本体にメッセージが含まれます。SOAP 仕様は、SOAP エンコードと呼ばれるプログラム言語バインディングのデフォルト・エンコードも定義します。

WSDL 文書と Web サービス

Web サービスへのキーは WSDL 文書です。WSDL 文書は、Web サービスのメッセージ送受信に関して記述する XML 文書です。メッセージは、通常は XML スキーマであるタイプ・システムを使用して記述されます。Web サービス操作はメッセージ交換パターンを 1 つ以上のメッセージと関連付けます。メッセージ交換パターンは、送信または受信されたメッセージのシーケンスとカーディナリティー、またメッセージの論理的な送信先または送信元を識別します。インターフェースは、トランスポートまたはワイヤー・フォーマットへのコミットメントなしに操作をグループ化します。WSDL バインディングは 1 つ以上のインターフェースに対して、トランスポートおよびワイヤー・フォーマット詳細を指定します。エンドポイントはネットワーク・アドレスをバインディングと関連付けます。サービスは、共通のインターフェースをインプリメントするエンドポイントをグループ化します。メッセージは、文書指向の情報または、リモート・プロシージャ・コール (RPC) としても知られるプロセス指向の情報を含むことができます。WSDL 文書は、1 つ以上の Web サービスを含むことが可能です。

242 ページの図 13 の例は株価見積価格を供給する簡単なサービスの WSDL 定義を示します。Web サービスは、GetLastTradePrice という名前の単一の操作をサポートします。このサービスは、HTTP を通して SOAP 1.1 プロトコルでアクセスできます。要求は、ストリング・データ・タイプであるチッカー・シンボルを入力として読み取り、浮動データ・タイプである価格を戻します。ストリングおよび浮動データ・タイプは XML スキーマ規格の事前定義タイプです。Web サービスもデータ・タイプを定義し、ユーザー定義のデータ・タイプを使用することができます。事前定義およびユーザー定義の XML データ・タイプはニックネームの列にマップします。完全な例と WSDL 仕様は、W3C Web サイトにあります。

```

<?xml version="1.0"?>
<definitions name="StockQuote"
...

<types>
  <schema targetNamespace="http://example.com/stockquote.xsd"
    xmlns="http://www.w3.org/2000/10/XMLSchema">
    <element name="TradePriceRequest">
      <complexType>
        <all>
          <element name="tickerSymbol" type="string"/>
        </all>
      </complexType>
    </element>
    <element name="TradePrice">
      <complexType>
        <all>
          <element name="price" type="float"/>
        </all>
      </complexType>
    </element>
  </schema>
</types>

<message name="GetLastTradePriceInput">
...
</message>

  <portType name="StockQuotePortType">
    <operation name="GetLastTradePrice">
      <input message="tns:GetLastTradePriceInput"/>
      <output message="tns:GetLastTradePriceOutput"/>
    </operation>
  </portType>

  <binding name="StockQuoteSoapBinding"
    type="tns:StockQuotePortType">
    <soap:binding style="document"
      transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="GetLastTradePrice">
      <soap:operation soapAction="http://example.com/GetLastTradePrice"/>
      <input>
        <soap:body use="literal"/>
      </input>
      <output>
        <soap:body use="literal"/>
      </output>
    </operation>
  </binding>

  <service name="StockQuoteService">
    <documentation>My first service</documentation>
    <port name="StockQuotePort" binding="tns:StockQuoteBinding">
      <soap:address location="http://example.com/stockquote"/>
    </port>
  </service>
</definitions>

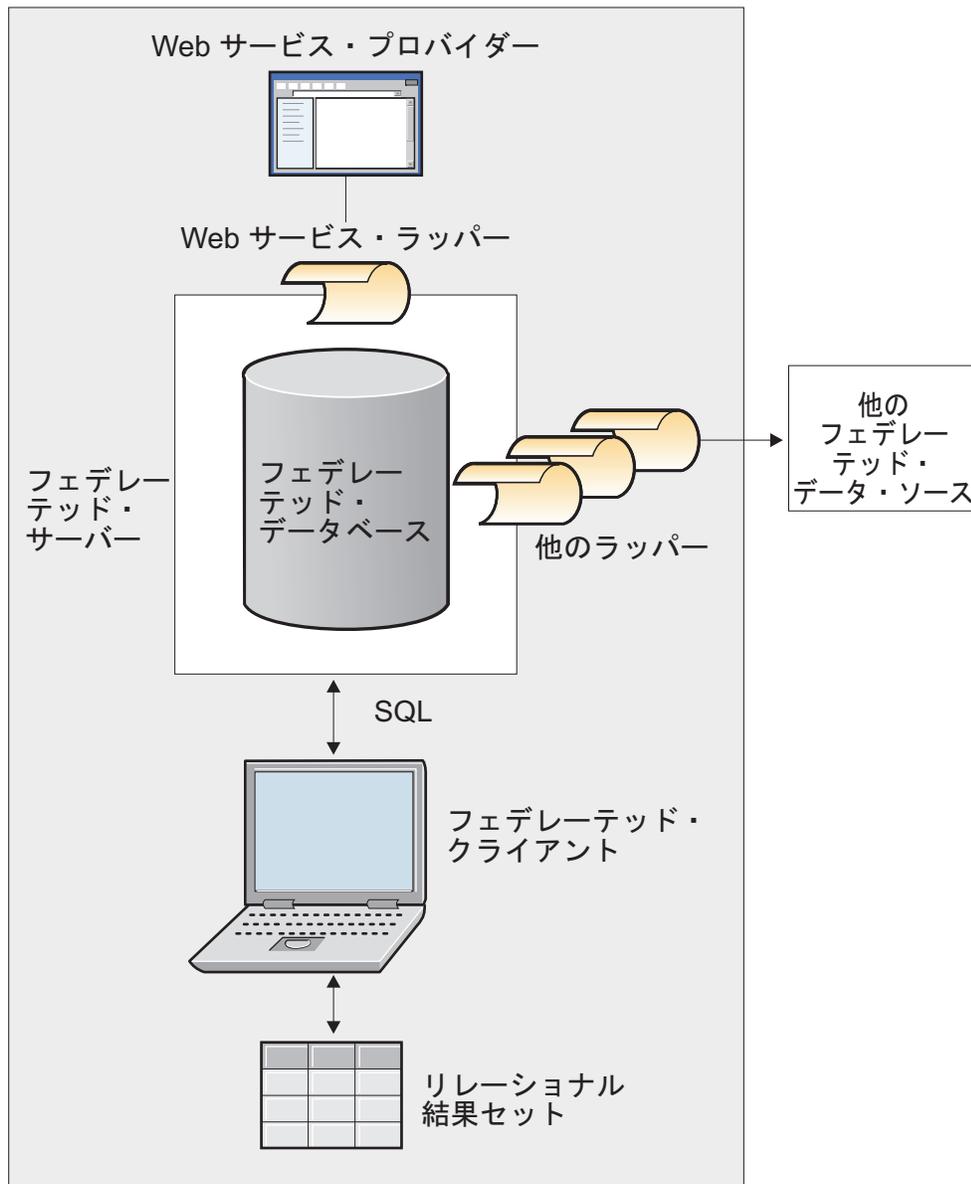
```

図 13. WSDL 文書の例

WSDL 文書、Web サービス・ラッパー、およびニックネーム

Web サービス・ラッパーは、HTTP トランスポートと SOAP バインディングのあるポート・タイプの操作を使用します。操作中の入力メッセージおよび関連したタイプまたはエレメントがニックネーム中の列になります。操作中の出力メッセージはニックネーム階層へ抽出されます。WSDL 文書中で、それぞれの操作ごとに別々のニックネームの階層を作成できます。

Web サービス・ラッパーを使用することにより、フェデレーテッド・システム関数を使用して、Web サービスからのデータを他のフェデレーテッド・データ・ソース上のデータと結合させることができます。



例 244 ページの図 14 では、操作名 GETTEMP のポート・タイプを含む WSDL 文書を使用します。この Web サービスでは、入力データとして郵便番号を入力し、

その郵便番号の地域の気温を受信します。

```
<?xml version="1.0"?>
<definitions name="TemperatureService" targetNamespace=http://www.xmethods.net/
  sd/TemperatureService.wsdl"
  xmlns:tns="http://www.xmethods.net/sd/TemperatureService.wsdl"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns="http://schemas.xmlsoap.org/wsdl/">
  <message name="getTempRequest">
    <part name="zipcode" type="xsd:string"/>
  </message>
  <message name="getTempResponse">
    <part name="return" type="xsd:float"/>
  </message>
  <portType name="TemperaturePortType">
    <operation name="getTemp">
      <input message="tns:getTempRequest"/>
      <output message="tns:getTempResponse"/>
    </operation>
  </portType>
  <binding name="TemperatureBinding" type="tns:TemperaturePortType">
    <soap:binding style="rpc"
      transport="http://schemas.xmlsoap.org/soap/http" />
    <operation name="getTemp">
      <soap:operation soapAction="" />
      <input>
        <soap:body use="encoded" namespace="urn:xmethods-Temperature"
          encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
      </input>
      <output>
        <soap:body use="encoded" namespace="urn:xmethods-Temperature"
          encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
      </output>
    </operation>
  </binding>
  <service name="TemperatureService">
    <documentation>
      Returns current temperature in a given U.S. zipcode
    </documentation>
    <port name="TemperaturePort" binding="tns:TemperatureBinding">
      <soap:address
        location="http://services.xmethods.net:80/soap/servlet/rpcrouter" />
    </port>
  </service>
</definitions>
```

図 14. GETTEMP Web サービス

入力値は、ニックネームの `zipcode` 列で記述されています。出力値は、ニックネームの `return` 列で記述されています。WSDL 文書では、これらの列はメッセージ・エレメントの中で識別されます。メッセージ・エレメントは、Web サービス・プロバイダーと Web サービス消費者との間で送信されるデータの論理定義を表します。メッセージ・エレメント中の情報にさらに説明が必要である場合、WSDL 文書にもタイプ・エレメントが含まれます。タイプ・エレメントは、XML スキーマ仕様に基づいた事前定義タイプ、またはユーザーによって定義されたタイプを意味します。

例 245 ページの図 15 では、GETTEMP Web サービス WSDL 文書から作成されるニックネームを示します。`zipcode` 列は、ニックネーム `TEMPLATE` 構文があるた

め必須入力列です。

```
CREATE NICKNAME GETTEMP (
  ZIPCODE VARCHAR (48) OPTIONS(TEMPLATE '&column'),
  RETURN VARCHAR (48) OPTIONS(XPATH './return/text()')
)
FOR SERVER "EHPWSSERV"
  OPTIONS(URL 'http://services.xmethods.net:80/soap/servlet/rpcrouter',
    SOAPACTION ' ',
    TEMPLATE '<soapenv:Envelope>
      <soapenv:Body>
        <ns2:getTemp>
          <zipcode>&zipcode[1,1]</zipcode>
        </ns2:getTemp>
      </soapenv:Body>
    </soapenv:Envelope>',
    XPATH '/soapenv:Envelope/soapenv:Body/*',
    NAMESPACES ' ns1="http://www.xmethods.net/sd/TemperatureService.wsdl",
      ns2="urn:xmethods-Temperature" ,
      soapenv="http://schemas.xmlsoap.org/soap/envelope/"');
```

図 15. GETTEMP ニックネーム

Web サービス・ラッパーのニックネーム・オプション URL および SOAPACTION は、ニックネーム作成時に指定したエンドポイントまたはアドレスをオーバーライドする機能を提供します。照会に URLCOLUMN または SOAPACTIONCOLUMN を使用可能にした列を使用する場合、同じニックネームにダイナミック・アドレスを使用できます。ニックネーム作成時に、ニックネーム・オプション URL および SOAPACTION を定義し、列オプションで URLCOLUMN および SOAPACTIONCOLUMN を使用可能にした場合、Web サービス・ラッパーの遅延バインディング機能を使用していることとなります。SOAPACTION ニックネーム・オプションの値が HTTP ヘッダー中の属性となります。URL ニックネーム・オプションの値は、要求の送信先である HTTP URL です。

URL および SOAPACTION ニックネーム・オプションは、ダイナミック・ニックネーム関連を提供します。これらのダイナミック・アドレスは、いくつかの会社が 1 つの Web サービス・ポート・タイプをインプリメントする場合に役立ちます。Web サービス・ラッパーには、WSDL 文書間の違いが異なる URL および SOAPACTIONS のみであることが必要です。遅延バインディング機能を使用して、異なる会社が使用する異なるサービス・エンドポイントに対して同じニックネームを作成、使用することができます。URL および SOAPACTION 値は、WSDL 文書から取られています。

以下の例では、URLCOLUMN および SOAPACTIONCOLUMN 列オプションを使用する方法を示します。

```

CREATE NICKNAME GetPartQuote(
  partnumber INTEGER OPTIONS (TEMPLATE '&column'),
  price FLOAT OPTIONS (XPATH './price'),
  urlcol VARCHAR(100) OPTIONS (URLCOLUMN 'Y'),
  soapactioncol VARCHAR(100) OPTIONS (SOAPACTIONCOLUMN 'Y'),
FOR SERVER myServer
OPTIONS (
  ...
  SOAPACTION 'http://example.com/GetPartPrice' ,
  URL 'http://mycompany.com:9080/GetPartPrice',
  ...
)

```

図 16. *GetPartQuote* ニックネーム

以下の例では、URLCOLUMN 列オプションおよび SOAPACTIONCOLUMN 列オプションが使用可能で定義された列、URLCOL および SOAPACTIONCOL を使用します。

```

SELECT * FROM supplier_endpoints p,
  GetPartQuote q
WHERE partnumber=1234 AND
  p.url=q.urlcol AND
  p.soapaction=q.soapactioncol;

```

SQL アプリケーションは、ニックネーム作成時に特定のエンドポイントを定義する代わりに、使用するエンドポイントの選択を照会が実行されるまで延期することができます。

Web サービス・ラッパーは、大量の WSDL 文書データをフラグメントに分離して、使用されるメモリの合計を縮小することができます。Web サービス・ニックネームを作成する場合、**STREAMING** オプションを指定してください。Web サービス・ラッパーは、その結果得られる XML データのストリームを処理し、それから照会フラグメントによって要求されている情報を抽出します。Web サービス・ラッパーは、フラグメントを一度に 1 つずつ解析します。**STREAMING** オプションを使用するのは、大規模な XML 文書を解析する場合だけにしてください。

Web サービス・ラッパーの登録

Web サービス・データ・ソースにアクセスするには、ラッパーを登録する必要があります。ラッパーを介して、フェデレーテッド・サーバーはデータ・ソースとの通信やデータ検索を行います。ラッパーは、ライブラリー・ファイルのセットとしてインプリメントされます。

始める前に

CREATE WRAPPER ラッパー・ステートメントで指定する正しい名前については、『Web サービス・ラッパー・ライブラリー・ファイル』のリストを参照してください。

このタスクについて

指定するラッパー・ライブラリー・ファイルの名前は、フェデレーテッド・サーバーのオペレーティング・システムによって異なります。

手順

ラッパーの名前とラッパー・ライブラリー・ファイルの名前を指定して、CREATE WRAPPER ステートメントを発行します。例えば、Windows を使用するフェデレーテッド・サーバーで websr_wrapper という名前のラッパーを登録するには、次のステートメントを発行します。

```
CREATE WRAPPER websr_wrapper LIBRARY 'db2ws.dll';
```

Web サービス・ラッパー・ライブラリー・ファイル

Web サービス・ラッパー・ライブラリー・ファイルは、フェデレーテッド・サーバーのインストール時にそのフェデレーテッド・サーバーに追加されます。

IBM InfoSphere Federation Server をインストールすると、デフォルトのディレクトリー・パスにライブラリー・ファイルが追加されます。例えば、フェデレーテッド・サーバーが AIX 上で実行されている場合、ディレクトリー・パスには、libdb2ws.a、libdb2wsF.a、および libdb2wsU.a が追加されます。デフォルトのラッパー・ライブラリー・ファイルは libdb2ws.a です。他のラッパー・ライブラリー・ファイルは特定のラッパー・オプションで使用されます。

CREATE WRAPPER ステートメントに LIBRARY パラメーターを含め、デフォルトのラッパー・ライブラリー・ファイルの名前を指定する必要があります。

以下の表に、デフォルトのディレクトリー・パスとデフォルトのラッパー・ライブラリー・ファイルの名前をリストします。

表 39. CREATE WRAPPER のライブラリー・ロケーションとファイル名

オペレーティング・システム	ディレクトリー・パス	ラッパー・ライブラリー・ファイル
AIX	/usr/opt/<install_path>/lib32/ /usr/opt/<install_path>/lib64/	libdb2ws.a
Linux	/opt/IBM/db2/<install_path>/lib32 /opt/IBM/db2/<install_path>/lib64	libsb2ws.so
Solaris	/opt/IBM/db2/<install_path>/lib32 /opt/IBM/db2/<install_path>/lib64	libdb2ws.so
Windows	%DB2PATH%\bin	db2ws.dll

- <install_path> は、Linux または UNIX 上でフェデレーテッド・サーバーがインストールされているディレクトリー・パスです。

%DB2PATH% は、Windows 上でフェデレーテッド・サーバーがインストールされているディレクトリー・パスを指定するのに使用する環境変数です。デフォルトの Windows ディレクトリー・パスは、C:\Program Files\IBM\SQLLIB です。

Web サービス・データ・ソースのサーバー定義の登録

アクセスする Web サービスごとに、Web サービス・サーバー定義を登録する必要があります。

手順

Web サービス・ラッパーのサーバー定義をフェデレーテッド・システムに登録するには、コマンド行から `CREATE SERVER` ステートメントを発行します。

例えば、Windows で `ws_server` という名前の Web サービス・サーバー定義を登録するには、次のステートメントを発行します。

```
CREATE SERVER ws_server WRAPPER websr_wrapper;
```

`CREATE SERVER` ステートメントには、オプションのタイムアウトおよびプロキシ・サーバー・パラメーターを設定できます。

CREATE SERVER ステートメント - Web サービス・ラッパーの例

`CREATE SERVER` ステートメントは、タイムアウト設定およびプロキシ・サーバー設定を使用して Web サービス・ラッパーのサーバー定義を登録するのに使用します。

Web サービス文書のアクセスにプロキシ・サーバーを使用しない場合でも、サーバー定義を登録する必要があります。フェデレーテッド・オブジェクトの階層では、Web サービスのファイルを特定のサーバー定義オブジェクトと関連づける必要があります。サーバー定義を登録するために発行するステートメントは以下のとおりです。

```
CREATE SERVER my_server WRAPPER my_wrapper  
  OPTIONS (TIMEOUT '60');
```

my_server

Web サービス・サーバー定義に割り当てる名前。重複するサーバー定義名は使用できません。

WRAPPER my_wrapper

`CREATE WRAPPER` ステートメントで指定したラッパー名。

TIMEOUT '60'

フェデレーテッド・サーバーがネットワーク転送と結果の計算を待つ時間(分単位)を指定します。

プロキシ・サーバーが使用される場合のサーバー定義

以下の条件のすべてが真である場合は、`CREATE SERVER` ステートメントでプロキシ・サーバー・オプションを使用する必要があります。

- URI を使用してデータを検索したい。
- URI はプロキシを介してファイアウォールの反対側にあるデータを検索する。
- ファイアウォールまたはプロキシは、HTTP または SOCKS である。

指定するオプションは、アクセスしたいプロキシ・サーバーのタイプによって異なります。

使用するプロキシのタイプ、およびプロキシ・オプションに指定する必要のある設定については、ネットワーク管理者に問い合わせてください。

HTTP プロキシ・サーバーのサーバー定義の登録

サーバー定義を登録して、HTTP プロキシ・サーバーを指定するには、以下のステートメントを使用します。

```
CREATE SERVER ws_server_http
  WRAPPER ws_wrapper
  OPTIONS (PROXY_TYPE 'HTTP', PROXY_SERVER_NAME 'proxy_http',
          PROXY_SERVER_PORT '8080');
```

ws_server_http

Web サービス・サーバー定義に割り当てる名前。重複するサーバー定義名は使用できません。

WRAPPER *ws_wrapper*

CREATE WRAPPER ステートメントで指定したラッパー名。

PROXY_TYPE '*HTTP*'

ファイアウォールで保護された状態でインターネットにアクセスする際に使用する、プロキシ・タイプを指定します。

PROXY_SERVER_NAME '*proxy_http*'

プロキシ・サーバー名または IP アドレスを指定します。

PROXY_SERVER_PORT '*8080*'

プロキシ・サーバーのポート番号を指定します。

認証情報のある SOCKS プロキシ・サーバーのサーバー定義の登録

サーバー定義を登録して、認証情報のある SOCKS プロキシ・サーバーを指定するには、以下のステートメントを発行します。

```
CREATE SERVER ws_server_socks
  WRAPPER ws_wrapper
  OPTIONS (PROXY_TYPE 'SOCKS', PROXY_SERVER_NAME 'proxy_socks',
          PROXY_SERVER_PORT '1081', PROXY_AUTHID 'Sampson',
          PROXY_PASSWORD 'sailing4me');
```

PROXY_AUTHID '*Sampson*'

プロキシ・サーバーのユーザー名を指定します。

PROXY_PASSWORD '*sailing4me*'

ユーザー名 *Sampson* に関連したプロキシ・サーバーのパスワードを指定します。

Web サービス・ラッパーを使用したセキュリティの使用可能化

Web サービス・ラッパーは、CREATE USER MAPPING ステートメントを使用して、HTTP 認証をサポートします。

このタスクについて

Web サービス・ラッパーは、SOAP メッセージのトランスポート・プロトコルとして HTTPS をサポートします。Web サービス・プロバイダーが生成する WSDL 文書には、URL に `https://` が含まれています。そのため、HTTP 要求または HTTP 応答中の SOAP メッセージは暗号化されます。

Web サービスがトランスポート・プロトコルとして HTTPS を使用する場合、SSL_VERIFY_SERVER_CERTIFICATE ラッパーまたはサーバー・オプションを使用

することによって、識別のためにサーバーが送信する SSL 証明書を検証するようにラッパーを構成できます。Web サービス・ラッパーは、自己署名証明書で Web サービスを呼び出すことができます。

ユーザー・マッピングは任意です。ユーザー・マッピングを指定しない場合、Web サービス・プロバイダーが認証情報を期待する際にエラーが出る可能性があります。サーバーの中には、サービスへのアクセスを制約するために認証を使用するものがあります。認証が必要であるかは、WSDL 文書中の情報からは明らかではありません。

手順

CREATE USER MAPPING ステートメントを発行します。

例

例えば、以下に示す CREATE USER MAPPING ステートメントでは、S1 サーバー上の Web サービス・ニックネームがアクセスされると、HTTP 要求が SYSTEM というユーザー ID、MANAGER というパスワードで送信されます。

```
CREATE USER MAPPING
  FOR RSPALTEN SERVER S1
  OPTIONS ( REMOTE_AUTHID 'SYSTEM', REMOTE_PASSWORD 'MANAGER'
          PROXY_AUTHID 'ID' PROXY_PASSWORD 'PWD'
          SSL_CLIENT_CERTIFICATE_LABEL 'LABEL');
```

Web サービス・データ・ソースのニックネームの登録

登録する Web サービス・サーバー定義ごとに、アクセスする各データ・ソースに対応するニックネームを登録する必要があります。Web サービス・データ・ソースを照会するときには、データ・ソースの名前ではなく、これらのニックネームを使用します。

始める前に

通信したい Web サービスを説明する有効な WSDL 文書へアクセスできる必要があります。

このタスクについて

制約事項

- 要求/応答の操作のみがサポートされています。
- HTTP トラnsポートの SOAP バインディングのみがサポートされるバインディングです。
- SOAPACTIONCOLUMN、URLCOLUMN、PRIMARY_KEY または FOREIGN_KEY オプションの特殊な列以外は、それぞれの列に TEMPLATE オプションまたは XPATH オプションを使用しなければなりません。

Web サービス記述言語 (WSDL) 文書の中で定義されているそれぞれの Web サービス操作ごとに、1 つのニックネーム階層を作成します。親ニックネームは、SOAP エンベロープを識別します。子ニックネームは、SOAP 本体の要素およびオプションの SOAP ヘッダーを識別します。

手順

コマンド行から CREATE NICKNAME ステートメントを発行します。例えば、GETTEMP という名前の Web サービスのために Windows でニックネームを登録するには、以下のステートメントを発行します。

```
CREATE NICKNAME GETTEMP (
  ZIPCODE VARCHAR (48) OPTIONS(TEMPLATE '&column'),
  RETURN VARCHAR (48) OPTIONS(XPATH './return/text()')
)
FOR SERVER "EHPWSSERV"
  OPTIONS(URL 'http://services.xmethods.net:80/soap/servlet/rpcrouter',
    SOAPACTION ' ',
    TEMPLATE '<soapenv:Envelope>
      <soapenv:Body>
        <ns2:getTemp>
          <zipcode>&zipcode[1,1]</zipcode>
        </ns2:getTemp>
      </soapenv:Body>
    </soapenv:Envelope>',
    XPATH '/soapenv:Envelope/soapenv:Body/*',
    NAMESPACES ' ns1="http://www.xmethods.net/sd/TemperatureService.wsdl",
      ns2="urn:xmethods-Temperature" ,
      soapenv="http://schemas.xmlsoap.org/soap/envelope/"');
```

CREATE NICKNAME ステートメント - Web サービス・ラッパーの例

Web サービスへのアクセスのためにニックネームを作成する場合、Web サービス操作の入力メッセージ中のそれぞれの値ごとに入力列を、また Web サービス操作の出力メッセージ中のそれぞれの値ごとに出力列を作成します。その入出力列定義は、ニックネーム列オプション定義でコントロールします。

TEMPLATE 列オプションは、列が入力列であることを指定します。 XPATH 列オプションは、列が出力列であることを指定します。 TEMPLATE ニックネーム・オプションに大括弧で囲まれた表記 ([1,1]) が含まれる場合、列は必須入力列です。

NAMESPACES ニックネーム・オプションは、入出力 XML 文書のエレメント用のネーム・スペースを解決するためにフェデレーテッド・システムが使用する、名前と値のペアをコンマで区切ったリストです。NAMESPACES はメッセージ要求に使用され、TEMPLATE ニックネーム・オプションの接頭部が定義されます。

NAMESPACES ニックネーム・オプションは、WSDL または XML スキーマで定義されたネーム・スペース URI がある XPath 式の接頭部を解決します。XPath 式は、Web サービスから戻される XML 文書で適用されます。

例 1: 必須入力列

以下の例では、getQuote という名前の Web サービスのニックネームを示します。この Web サービスは株式チャッカー・シンボルを読み取り、取引価格を戻します。

```
CREATE NICKNAME "stockquote.stockquoteport_getquote_nn" (
  symbol VARCHAR (48) OPTIONS(TEMPLATE '&column'),
  result VARCHAR (48) OPTIONS(XPATH './Result/text()')
)
FOR SERVER "xmethods_server" OPTIONS(
  URL 'http://66.28.98.121:9090/soap' ,
  SOAPACTION 'urn:xmethods-delayed-quotes#getQuote' ,
  TEMPLATE '<soapenv:Envelope>
    <soapenv:Body>
      <ns2:getQuote>
```

```

        <symbol>&symbol [1,1]</symbol>
    </ns2:getQuote>
</soapenv:Body>
</soapenv:Envelope>',
XPATH '/soapenv:Envelope/soapenv:Body/*' ,
NAMESPACES 'ns2="urn:xmethods-delayed-quotes" ,
            ns1="http://www.example.com/wsdl/
            net.xmethods.services.stockquote.StockQuote/" ,
            soapenv="http://schemas.xmlsoap.org/soap/envelope/" ');

```

ニックネーム `TEMPLATE` オプションは、列 `SYMBOL` に `[1,1]` 指定が含まれるため、この列を必須入力列として指定します。ニックネーム `TEMPLATE` オプションでは、全部の `SOAP` エンベロープが `Web` サービスに指定されています。`getQuote` 入力値は、`SOAP` エンベロープおよびボディ・エレメントに含まれています。`XPATH` ニックネーム・オプションには、`SOAP` エンベロープおよびボディ・タグを通して取引価格を検索するための情報が含まれています。

以下の照会のように、`stockquote.stockquoteport_getquote_nn` ニックネームを使用して `Web` サービスにアクセスします。

```

SELECT * FROM "stockquote.stockquoteport_getquote_nn"
WHERE symbol='IBM';

```

`symbol` が必須入力列であるため、このステートメントでは述部 `symbol='IBM'` を使用する必要があります。入力列では、等価述部のみが有効な述部です。等価述部のそれぞれが、入力メッセージ中の値を設定します。入力列がオプションである場合には、その列上に等価述部は必要ではありません。入力列が必須である場合、等価述部で照会を発行する必要があります。等価式に `IBM` などのリテラル値、あるいは、中間結合表またはニックネームからの値を使用することができます。

例 2: 繰り返しエレメントおよび子ニックネーム

以下の例では、`getZooReport` という名前の動物園のレポートを作成する `Web` サービスを使用します。入力値は、動物園 `ID` です。出力値は、以下のスキーマで表されるレポートです。

```

<wsdl:definitions name="Name"
  targetNamespace="http://myzoo.com"
  ...
<wsdl:types>
  <xsd:schema elementFormDefault="qualified" targetNamespace="http://myzoo.com"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <xsd:element name="Animal">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element ref="tns:Name"/>
          <xsd:element ref="tns:Species"/>
          <xsd:element ref="tns:Lot"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="AnimalCareList">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element maxOccurs="unbounded" minOccurs="1" ref="tns:Animal"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="Count" type="xsd:string"/>
    <xsd:element name="LastModified" type="xsd:string"/>
    <xsd:element name="Lot" type="xsd:string"/>
    <xsd:element name="Name" type="xsd:string"/>
    <xsd:element name="NumberOfCages" type="xsd:string"/>
    <xsd:element name="Species" type="xsd:string"/>
    <xsd:element name="Zoo">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element ref="tns:ZooName"/>
          <xsd:element ref="tns:Count"/>
          <xsd:element ref="tns:LastModified"/>
          <xsd:element maxOccurs="unbounded" minOccurs="0" ref="tns:Zookeeper"/>
        </xsd:sequence>
        <xsd:attribute name="id" type="xsd:string" use="optional"/>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="ZooName" type="xsd:string"/>
    <xsd:element name="Zookeeper">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element ref="tns:Name"/>
          <xsd:element ref="tns:NumberOfCages"/>
          <xsd:element ref="tns:AnimalCareList"/>
        </xsd:sequence>
        <xsd:attribute name="id" type="xsd:string" use="optional"/>
      </xsd:complexType>
    </xsd:element>
  </xsd:schema>
</wsdl:types>
  ...

```

図 17. *getZooReport* Web サービス

以下に示す DDL ステートメントは、スキーマを含む WSDL に基づいています。

```

CREATE NICKNAME zooport_getzooreport_nn (
  zooid VARCHAR (48) OPTIONS(TEMPLATE '&column'),
  zoo_id VARCHAR (48) OPTIONS(XPATH './ns1:Zoo/@ns1:id'),
  report_zooname VARCHAR (48) OPTIONS(XPATH './ns1:Zoo/ns1:ZooName/text()'),
  report_count VARCHAR (48) OPTIONS(XPATH './ns1:Zoo/ns1:Count/text()'),
  report_lastmodified VARCHAR (48)
    OPTIONS(XPATH './ns1:Zoo/ns1:LastModified/text()'),
  zooport_getzooreport_pkey VARCHAR (16) FOR BIT DATA NOT NULL
    OPTIONS(PRIMARY_KEY 'YES'))
FOR SERVER "zooserver" OPTIONS(
  URL 'http://localhost:9080/MaelstromTest/services/ZooPort' ,
  SOAPACTION 'http://myzoo.com/getZooReport' ,
  TEMPLATE '<soapenv:Envelope>
    <soapenv:Body>
      <zooId>&zooId[1,1]</zooId>
    </soapenv:Body>
  </soapenv:Envelope>' ,
  XPATH '/soapenv:Envelope/soapenv:Body' ,
  NAMESPACES ' soapenv="http://schemas.xmlsoap.org/soap/envelope/" ,
              ns1="http://myzoo.com " ');

```

図 18. 親ニックネーム - zooport_getzooreport_nn

```

CREATE NICKNAME zooport_getzooreport_report_nn (
  zooport_getzooreport_fkey VARCHAR (16)
    FOR BIT DATA NOT NULL
    OPTIONS(FOREIGN_KEY 'ZOOPORT_GETZOOREPORT_NN'),
  zookeeper_id VARCHAR (48) OPTIONS(XPATH './ns1:Zookeeper/@ns1:id'),
  report_name VARCHAR (48) OPTIONS(XPATH './ns1:Zookeeper/ns1:Name/text()'),
  report_numberofcages VARCHAR (48)
    OPTIONS(XPATH './ns1:Zookeeper/ns1:NumberOfCages/text()'),
  zooport_getzooreport_pkey VARCHAR (16)
    FOR BIT DATA NOT NULL OPTIONS(PRIMARY_KEY 'YES'))
FOR SERVER "zooserver" OPTIONS(
  XPATH './ns1:Zoo' ,
  NAMESPACES ' soapenv="http://schemas.xmlsoap.org/soap/envelope/" ,
              ns1="http://myzoo.com" ');

```

図 19. ニックネーム zooport_getzooreport_nn の子

```

CREATE NICKNAME zooport_getzooreport_report_report_nn (
  zooport_getzooreport_fkey VARCHAR (16) FOR BIT DATA NOT NULL
    OPTIONS(FOREIGN_KEY 'zooport_getzooreport_report_nn'),
  report_name VARCHAR (48)
    OPTIONS(XPATH './ns1:Animal/ns1:Name/text()'),
  report_species VARCHAR (48)
    OPTIONS(XPATH './ns1:Animal/ns1:Species/text()'),
  report_lot VARCHAR (48) OPTIONS(XPATH './ns1:Animal/ns1:Lot/text()'))
FOR SERVER "zooserver" OPTIONS(
  XPATH './ns1:Zookeeper/ns1:AnimalCareList' ,
  NAMESPACES ' soapenv="http://schemas.xmlsoap.org/soap/envelope/" ,
              ns1="http://myzoo.com" ');

```

図 20. zooport_getzooreport_report_nn の子

スキーマには、シーケンス・エレメントと呼ばれる、繰り返しエレメントが含まれます。254 ページの図 18、254 ページの図 19、および 254 ページの図 20で示されているように、これらの繰り返しエレメントが親ニックネームの子ニックネームになります。例えば、zooname、count、lastmodified、および zookeeper はすべて zoo のエレメントです。エレメント zoo には、0 またはそれ以上の zookeeper エレメントが含まれます。ルート・ニックネーム zoo には列、zooname、count、および lastmodified が含まれます。子ニックネーム zookeeper が作成され、zookeeper の繰り返しエレメントを説明します。zookeeper 列の 3 番目のエレメント animalcarelist も 0 またはそれ以上のエレメントを含むため、子ニックネーム zooport_getzooreport_report_report_nn になります。次の図では、ニックネーム階層を示します。

ルート・ニックネーム: zooport_getzooreport_nn

Zoo (親):

- ZooName
- Count
- LastModified
- ZooKeeper (0 またはそれ以上の ZooKeeper エレメントがある)

子ニックネーム: zooport_getzooreport_report_nn

ZooKeeper エレメント

- Name
- NumberOfCages
- AnimalCareList (0 またはそれ以上の Animal エレメントがある)

子ニックネーム: zooport_getzooreport_report_report_nn

Animal

- Name
- Species
- Lot

図 21. 親 -> 子 -> ニックネーム階層

以下のステートメントは、動物園レポートの Web サービスにアクセスするためニックネームに発行する照会の標準的なものです。このステートメントを発行する際には、特定の ID に基づき、また子ニックネームの動物園レポートの主キーと外部キーがどこで一致するかに基づいて動物園レポートからの情報を検索します。

```
SELECT * FROM zooport_getzooreport_nn ,
        zooport_getzooreport_report_nn zk ,
        zooport_getzooreport_report_report_nn a
WHERE zooid='1'AND zooport_getzooreport_pkey=zk.zooport_getzooreport_fkey
and zk.zooport_getzooreport_pkey=a.zooport_getzooreport_fkey;
```

例 3: 遅延バインディング

以下の例では、遅延バインディング・オプションを使用する方法を示します。ニックネーム・オプション URL および SOAPACTION を定義し、ニックネーム作成時に列オプション URLCOLUMN および SOAPACTIONCOLUMN を使用可能にした場合、遅延バインディング機能を使用していることになります。

以下の例は、ある会社のためにすべてのサプライヤーがインプリメントする部品の見積価格を提供する Web サービスのものです。ここに、URLCOLUMN および SOAPACTIONCOLUMN 定義を含む CREATE NICKNAME ステートメントがあります。

```
CREATE NICKNAME GetPartQuote(  
  partnumber INTEGER OPTIONS (TEMPLATE '&column'),  
  price FLOAT OPTIONS (XPATH './price'),  
  urlcol VARCHAR(100) OPTIONS (URLCOLUMN 'Y'),  
  soapactioncol VARCHAR(100) OPTIONS (SOAPACTIONCOLUMN 'Y'),  
  FOR SERVER myServer  
  OPTIONS (  
    ...  
    SOAPACTION 'http://example.com/GetPartPrice' ,  
    URL 'http://mycompany.com:9080/GetPartPrice',  
    ...  
  )
```

単一の照会ですべてのサプライヤーからの見積価格を得るには、SOAPACTION および URL 列オプションにそれぞれのサプライヤーが使用する値が必要です。照会は次のようになります。

```
SELECT * FROM supplier_endpoints p,  
  GetPartQuote q  
WHERE partnumber=1234 AND  
  p.url=q.urlcol AND  
  p.soapaction=q.soapactioncol;
```

ローカル表 `supplier_endpoints` には、Web サービスを呼び出すのに使用するすべての URL および SOAP が含まれています。ORDER BY price 文節を含めて、この部品が最も安価なサプライヤーを判別することができます。

例 4: ESCAPE_INPUT 列オプション

照会に、入力値として XML フラグメントを含めることができます。ニックネームを登録する際に、列オプション `ESCAPE_INPUT=N` を含めてください。このオプションは不等号括弧 (< や >) などの特殊文字を入力値の XML フラグメント中に維持します。

SOAP メッセージの一部として XML を送信することが必要である繰り返し入力値がスキーマに含まれている場合、`ESCAPE_INPUT` 列オプションを使用して、正しい XML で出力メッセージを作成することができます。

例えば、動物園 Web サービスには新しい動物園の飼育係とその飼育係と関連した動物を追加する操作が含まれています。この例のスキーマでは、`AnimalCareList` は複数の動物を持つことができます。

```
CREATE NICKNAME add_zookeeper(  
  zookeeper_id VARCHAR(48) OPTIONS(TEMPLATE '...'),  
  name VARCHAR(48) OPTIONS(TEMPLATE '...'),  
  numberofcages VARCHAR(48) OPTIONS(TEMPLATE '...'),  
  animals VARCHAR(3000) OPTIONS( TEMPLATE '...' , ESCAPE_INPUT 'N')  
  ...
```

新しい動物園の飼育係を 2 匹の動物と共に追加するには、以下のような照会を発行します。

```
SELECT * FROM add_zookeeper  
  WHERE zookeeper_ID='37' AND  
  name='Amit Kapoor' AND
```

```

numberofcages='3' AND
animals='<AnimalCareList xmlns="http://myzoo.com">
  <Animal>
    <Name>Larry</Name>
    <Species>Gorilla</Species>
    <Lot>7</Lot>
  </Animal>
  <Animal>
    <Name>Bill</Name>
    <Species>Chimpanzee</Species>
    <Lot>8H</Lot>
  </Animal>
</AnimalCareList>';

```

add_zookeeper ニックネームは、Web サービスの状態を変更または情報を更新することができる Web サービス操作です。非リレーショナル・ラッパーを更新することはできませんが、この例の SELECT ステートメントは新しい動物園の飼育係を追加して動物園の情報を更新します。

xsd:anyType などのエレメントを使用するスキーマには、ESCAPE_INPUT 列オプションを使用することもできます。この場合、エレメントのタイプは知られていません。入力に対して任意の XML フラグメントを指定できるよう、そのエレメントには ESCAPE_INPUT 列オプションを入力列に使用できます。

SOAP ヘッダー・アクセス

Web サービス・ラッパーを使用して、SOAP メッセージに埋め込まれている Simple Object Access Protocol (SOAP) ヘッダー値にアクセスします。

SOAP メッセージ (要求または応答のいずれか) は、エンベロープ (SOAP メッセージの残りの部分によって使用されるネーム・スペースを含む)、オプションのヘッダー、および本体から構成されます。SOAP ヘッダーには通常、SOAP メッセージに関するメタデータが含まれ、他の情報がオプションに含まれることもあります。SOAP メッセージは、メッセージまたはその副次作用を処理する一連の中継サービスを通じてにより、ソースから宛先へと伝わります。例えば、メッセージは、ネットワーク障害の場合でも保証された呼び出しをクライアントに提供するトランザクション・サービスや、エンタープライズ・ポータルにある、認証情報を提供するセキュリティー・サービスを通じてことがあります。これらの中継者では、多くの場合、SOAP ヘッダーに含まれるメタデータが処理されます。

SOAP ヘッダーには多くの場合、オプションの情報またはユーザー・インターフェースを発展させるサポートが含まれます。例えば、ある銀行は、顧客が ATM カードを使用して当座預金口座から現金を引き出すことを許可しています。ただし、一部の顧客は、複数の口座を持っています。顧客が複数の口座を持っている場合、銀行は、引き出しが行われる口座を識別できなければなりません。

この例を使用すると、銀行口座を指定するヘッダー・エレメントは以下のようになります。

```
<xsd:element name="AccountID" type="xsd:int"/>
```

以下は、このヘッダー・エレメントを使用する SOAP メッセージです。この事例では、顧客は口座 1 から \$1,000 を引き出すことにしました。これには、50 セントの取引手数料がかかります。

```

<SOAP-ENV:Envelope>
  <SOAP-ENV:Header>
    <tns:AccountID>1</tns:AccountID>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <tns:Action>withdrawal</tns:Action>
    <tns:Amount>1000</tns:Amount>
    <tns:Fee>0.50</tns:Fee>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

SOAP ヘッダーには、以下の要件があります。

- soapenv:Header は、soapenv:Envelope の下のオプション・エレメントです。
- SOAP ヘッダーを使用する場合、それが最初のエレメントでなければなりません。
- soapenv:Header のすべての子は、ネーム・スペースで修飾されている必要があります。
- soapenv:Header は、actor、mustUnderstand、および encodingStyle という属性を持つことができます。

SOAP ヘッダーにアクセスするニックネーム階層は、ルート・ニックネームが SOAP エンベロープを指している必要があります。ヘッダーおよび本体のエレメントはルート・ニックネームの子ニックネームです。ニックネームは、CREATE NICKNAME ステートメントを発行して作成できます。その後、SOAP ヘッダーに格納されている情報にアクセスする照会を作成できます。

例

これらの CREATE NICKNAME ステートメントは、ルート、本体、およびヘッダーのニックネームを作成する方法を示しています。

```

CREATE NICKNAME STOCKSERVICE_GETLASTSELLPRICE_ROOT_NN
  (NN_PKEY VARCHAR(16)NOT NULL OPTIONS(PRIMARY_KEY 'YES'))
FOR SERVER "WSSERVER_HDR"
OPTIONS
  (URL 'http://wswrap.svl.ibm.com:9081/StockPrice/services/StockService' ,
  SOAPACTION ' ' , TEMPLATE
  '<soapenv:Envelope>&STOCKSERVICE_GETLASTSELLPRICE_HEADER_NN[1,1]
  &STOCKSERVICE_GETLASTSELLPRICE_BODY_NN[1,1]</soapenv:Envelope>', XPATH
  '/soapenv:Envelope',
  NAMESPACES 'soapenv="http://schemas.xmlsoap.org/soap/envelope/",
  ns1="http://soapheader.ibm.com"');

CREATE NICKNAME STOCKSERVICE_GETLASTSELLPRICE_HEADER_NN
  (NN_PKEY VARCHAR(16)NOT NULL OPTIONS(PRIMARY_KEY 'YES'),
  NN_FKEY VARCHAR (16) NOT NULL OPTIONS(FOREIGN_KEY
  'STOCKSERVICE_GETLASTSELLPRICE_ROOT_NN'),
  QUOTE_TIMESTAMP VARCHAR(30)OPTIONS(TEMPLATE '&column'))
FOR SERVER "WSSERVER_HDR"
OPTIONS
  (XPATH './soapenv:Header',TEMPLATE '<soapenv:Header>
  <ns1:quote_timestamp>&quote_timestamp[1,1]
  </ns1:quote_timestamp></soapenv:Header>',
  NAMESPACES 'ns1="http://soapheader.ibm.com",soapenv=
  "http://schemas.xmlsoap.org/soap/envelope/" ');

CREATE NICKNAME STOCKSERVICE_GETLASTSELLPRICE_BODY_NN
  (NN_PKEY VARCHAR(16)NOT NULL
  OPTIONS(PRIMARY_KEY 'YES'), NN_FKEY VARCHAR 16)NOT NULL
  OPTIONS(FOREIGN_KEY 'STOCKSERVICE_GETLASTSELLPRICE_ROOT_NN'),

```

```

GETLASTSELLPRICE_TICKER VARCHAR (48)
OPTIONS(TEMPLATE '&column'), GETLASTSELLPRICEReturn VARCHAR (48)
OPTIONS(XPATH
'./ns1:getLastSellPriceResponse/ns1:getLastSellPriceReturn/text()')
FOR SERVER "WSSERVER_HDR"
OPTIONS
(TEMPLATE '<soapenv:Body><ns1:getLastSellPrice>
<ns1:ticker>&getLastSellPrice_ticker[1,1]
</ns1:ticker></ns1:getLastSellPrice></soapenv:Body>',
XPATH './soapenv:Body',
NAMESPACES ' soapenv="http://schemas.xmlsoap.org/soap/envelope/" ,
ns1="http://soapheader.ibm.com" ');

```

以下の照会は、特定の日時における IBM 株式の価格を戻します。

```

SELECT * from
STOCKSERVICE_GETLASTSELLPRICE_ROOT_NN as a,
STOCKSERVICE_GETLASTSELLPRICE_HEADER_NN as b,
STOCKSERVICE_GETLASTSELLPRICE_BODY_NN as c
where
a.nn_pkey = b.nn_fkey and
a.nn_pkey = c.nn_fkey and
b.QUOTE_TIMESTAMP='2007-05-09T18:30:20.000Z'and
c.GETLASTSELLPRICE_TICKER='IBM';

```

ニックネームと XPath 式

Web サービス・ラッパーは、XML データのツリー構造に対応するニックネームを使用します。各ニックネームは、出力値を表す XPath 式によって定義されます。

ニックネームは、XML 文書データのツリー構造に対応しています。親ニックネームと子ニックネームは、データ・ツリー構造のルート構造とネスト・エレメントに対応します。それら親ニックネームと子ニックネームは、CREATE NICKNAME ステートメントで指定される主キーおよび外部キーによって結び付けられます。

各ニックネームは、出力値を表す XPath 式によって定義されます。Web サービス・ラッパーは、XPath 式を使用することによって、XML 文書のデータと、リレーショナル表の行との間の対応を確立します。それらの XPath 式は、XML 文書内の値を特定し、それらの値が各行の列にどう対応するかを決定します。ラッパーは XML データの読み取りのみを行い、文書の更新は行いません。XPath オプションには、SOAP エンベロープおよび SOAP ボディ・タグを通して SOAP メッセージを検索するための情報が含まれています。getQuote メッセージは、SOAP エンベロープおよびボディ・エレメントに含まれています。

NICKNAME オプションの XPath 式は、出力エレメントの繰り返しタグを指します。XPath 式によって、ニックネームの行の数および行の種類が決まります。列オプションの XPath 式は NICKNAME XPath 式に対して相対的です。列オプションの XPath は行に含まれる値を識別します。子ニックネームの NICKNAME オプションの XPath は、親ニックネームの NICKNAME オプションの XPath 式に対して相対的です。

ニックネームを作成する際、そのニックネームと XML 文書との間の関連付けを指定するための Web サービス記述言語 (WSDL) 文書内のオプションを選択します。

ニックネームおよび列レベルでの TEMPLATE オプション

ニックネームを作成する場合、Web サービス・ラッパーはニックネーム・レベルおよび列レベルのテンプレート・フラグメントを必要とします。これは、CREATE NICKNAME ステートメントの TEMPLATE オプションです。

Web サービス・ラッパーは、Web サービス環境によって必要とされる XML 文書を作成します。ラッパーはこの情報を、照会計画段階および照会実行段階で使用します。

Web サービス・ラッパーの TEMPLATE オプション:

Web サービス・ラッパーには、ニックネーム作成時に CREATE NICKNAME ステートメントに TEMPLATE オプションが必要です。

Web サービス・ラッパー

Web サービス・ラッパーの場合、属性が必須かオプションかは、WSDL 文書内の定義、および列の導出方法によって異なります。列は、エレメントまたはエレメントの属性のいずれかから導出できます。

- 列がエレメントから導出されている場合、列がオプションかどうかは minOccurs 値によって判別されます。
- minOccurs の値が 0 である場合、列はオプションです。
- minOccurs の値が 1 である場合、列は必須です。
- 列がエレメントの属性から導出されている場合、列がオプションかどうかは属性に使用されている値により判別されます。
- 属性に値 use=optional が含まれている場合、列はオプションです。
- 属性に値 use=required が含まれている場合、列は必須です。

次の例は、列に関連付けられているスキーマ定義内の属性を示しています。

```
<xsd:complexType>
  <xsd:sequence>
    <xsd:element ref="tns:ZooName"/>
    <xsd:element ref="tns:Count"/>
    <xsd:element ref="tns:LastModified"/>
    <xsd:element maxOccurs="unbounded" minOccurs="0" ref="tns:Zookeeper"/>
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:string" use="optional"/>
</xsd:complexType>
```

Web サービス・ニックネームのフェデレーテッド・ビューの作成

Web サービス・ニックネームのフェデレーテッド・ビューを作成して、Web サービス・ニックネーム階層の各部を結合する照会が正常に実行されるようにします。

このタスクについて

Web サービス文書を説明するニックネームの階層のフェデレーテッド・ビューを定義できます。フェデレーテッド・ビューにより、Web サービス・ニックネーム階層の断片を結合する照会が適切に実行されるようになります。

Web サービス・ラッパーは、すべてのニックネームを含めるために子ニックネームから元の親ニックネームへの結合を必要とします。親ニックネームがステートメン

トに含められない場合、子ニックネームの SELECT ステートメントだけが失敗します。必要とされるすべてのニックネームを照会に含める良い方法は、フェデレーテッド・ビューを定義することです。

手順

1. そのようなニックネームを結合する場合は、Web サービスでの操作に関連したすべてのニックネームを含むビューを定義します。
2. ビューの WHERE 文節では、PRIMARY_KEY および FOREIGN_KEY 列オプションによって関連するすべての列に対して、結合述部を使用します。

例

以下の例では、主キーは、ニックネーム zooport_getzooreport_report_nn 中の列 ooport_getzooreport_pk にあります。外部キーは、ニックネーム zooport_getzooreport_report_report_nn 中の列 ooport_getzooreport_fkey にあります。

```
CREATE VIEW zooreport
(zoooid, zooname, number_of_zookeeper,
lastmodified, zookeeper_id, zookeeper_name,
cages, animal_name, animal_species, animal_lot)
AS ( SELECT zoooid, report_zooname,
report_count, report_lastmodified,
zookeeper_id, zk.report_name, report_numbercages,
a.report_name, report_species,
report_lot
FROM zooport_getzooreport_nn ,
zooport_getzooreport_report_nn as zk,
zooport_getzooreport_report_report_nn as a
WHERE zk.ooport_getzooreport_pkey=a.ooport_getzooreport_fkey
AND zooport_getzooreport_pkey=zk.ooport_getzooreport_fkey);
```

以下の SELECT ステートメントで、すべてのニックネームからの情報を得ることができます。

```
SELECT * FROM zooreport WHERE zoooid='1';
```

Web サービス・ラッパーの照会制約事項

照会の述部および結合の使用には、提供されている例で示されているとおり、Web サービスで制限があります。

等価述部

入力列で有効な述部は、等価述部だけです。出力列の場合、任意の述部が有効です。

以下の例ではエラーが返され、その列でその述部がサポートされていないことを示すメッセージが出されます。この例では、列 zipcode が入力列です。

```
SELECT return FROM gettemp WHERE zipcode<'95141'
```

以下の例では、入力列で等価述部が使用される有効な照会が示されています。顧客のニックネームが、顧客 ID を含むローカル表に結合されます。照会では、出力専用列である Sales 列に追加の述部が含まれています。

```

SELECT a.name, a.address
  FROM customers a, local_table b
 WHERE
   a.customer_id=b.custid AND
   a.Sales > 300000;

```

必須の入力列の述部

参照するニックネーム階層の SQL 照会で、必須のすべての入力列の等価述部の値を提供する必要があります。ラッパーは、この制約事項に違反するすべての照会で SQLCODE -901 を返します。

IN または OR 述部

Web サービス・ラッパーでは、入力列で IN または OR 述部は許可されません。

以下の例は、無効な照会を示しています。顧客のニックネームには、必須の入力列である customer_id があります。

```

SELECT * FROM customers
 WHERE customer_id IN (12345, 67890, 11223);
SELECT * FROM customers
 WHERE customer_id IN (SELECT custid FROM local_table); )

```

オプションの入力列を結合します。

以下の例は、オプションの入力列の結合およびホスト変数の使用に関する制約事項を示しています。ローカル表またはニックネームからオプションの入力列を結合することはできません。WSDL でオプションとして入力ニックネーム列が生成され、ユーザーが結合でその列を使用する必要がある場合、DDL ステートメントを編集して、その列を必須の入力列に変更しなければなりません。

この例では、オプションの入力列として shipping_method とともに、order という Web サービス・ラッパーのニックネームが作成されます。以下のステートメントは、述部でリテラルが使用されているため有効な照会です。

```

SELECT * FROM order
 WHERE part="hammer" AND shipping_method="FEDEX";

```

ただし、パーツおよび配送方法を定義する orderparts というローカル表を照会に含める場合、その表にオプションである shipping_method という列が含まれていると、そのステートメントは無効です。

```

SELECT * FROM
 order o, orderparts op
 WHERE
  o.part="hammer" AND
  o.shipping_method=op.shipping_method

```

オプションの入力列の述部を含む照会でホスト変数を使用する場合、以下のステートメントも無効です。

```

SELECT * FROM
 order o
 WHERE
  o.part="hammer" AND
  o.shipping_method=:ashipping_method

```

結果を有効なものとするには、結合された入力列が Web サービス・ラッパーでの必須の列でなければなりません。

外部結合

親ニックネームからの主キーと子ニックネーム列からの外部キーを使用するニックネーム間の外部結合は、サポートされていません。

XML 文書の親エレメントに子エレメントが含まれていない場合、および親ニックネームと子ニックネームの間の内部結合を使用する場合、そのエレメントでは返されません。例えば、ある顧客に関して SAP システムで銀行詳細 (bankdetail) 情報がない場合、その顧客の the sap_bapi_customer_getdetail2_sap_customerbankdetail_NN ニックネームでは返されません。

以下の CREATE NICKNAME ステートメントでは、照会の例で使用される列が定義されます。

```
CREATE NICKNAME sap_bapi_customer_getdetail2_NN(
...
NAME VARCHAR(35)
  OPTIONS(XPATH './ns3:sap_customeraddress/
              ns1:sap_customeraddress/ns1:NAME/text()'),
...
NN_PKEY VARCHAR(16) OPTIONS(PRIMARY_KEY 'YES'),
COMPANYCODE VARCHAR(4) OPTIONS(XPATH './ns3:COMPANYCODE/text()',
  TEMPLATE '<ns3:COMPANYCODE>&column</ns3:COMPANYCODE>'),
CUSTOMERNO VARCHAR(10) OPTIONS(XPATH './ns3:CUSTOMERNO/text()',
  TEMPLATE '<ns3:CUSTOMERNO>&column</ns3:CUSTOMERNO>'),
...
FOR SERVER sap_server
OPTIONS(XPATH '/ns3:sap_bapi_customer_getdetail2',
  TEMPLATE '<ns3:sap_bapi_customer_getdetail2>
           &sap_bapi_customer_getdetail2_sap_customerbankdetail_NN[0,1]
           &COMPANYCODE[0,1]
           &CUSTOMERNO[1,1]
           </ns3:sap_bapi_customer_getdetail2>',
...

```

図 22. getdetail2 ニックネームからの抜粋

```

CREATE NICKNAME sap_bapi_customer_getdetail2_sap_customerbankdetail_NN(
  CUSTOMER VARCHAR(10) OPTIONS(XPATH './ns2:CUSTOMER/text()',
    TEMPLATE '<ns2:CUSTOMER>&column</ns2:CUSTOMER>'),
  BANK_KEY VARCHAR(15) OPTIONS(XPATH './ns2:BANK_KEY/text()',
    TEMPLATE '<ns2:BANK_KEY>&column</ns2:BANK_KEY>'),
  BANK_ACCT VARCHAR(18) OPTIONS(XPATH './ns2:BANK_ACCT/text()',
    TEMPLATE '<ns2:BANK_ACCT>&column</ns2:BANK_ACCT>'),
  CTRL_KEY VARCHAR(2) OPTIONS(XPATH './ns2:CTRL_KEY/text()',
    TEMPLATE '<ns2:CTRL_KEY>&column</ns2:CTRL_KEY>'),
  BANK_REF VARCHAR(20) OPTIONS(XPATH './ns2:BANK_REF/text()',
    TEMPLATE '<ns2:BANK_REF>&column</ns2:BANK_REF>'),
  NN_FKEY VARCHAR(16) OPTIONS(FOREIGN_KEY 'SAP_BAPI_CUSTOMER_GETDETAIL2_NN'))
FOR SERVER sap_server
OPTIONS(XPATH './ns3:sap_customerbankdetail/ns2:sap_customerbankdetail',
  TEMPLATE '<ns3:sap_customerbankdetail>
  <ns2:sap_customerbankdetail>
    &CUSTOMER[0,1]
    &BANK_KEY[0,1]
    &BANK_ACCT[0,1]
    &CTRL_KEY[0,1]
    &BANK_REF[0,1]
  </ns2:sap_customerbankdetail>
</ns3:sap_customerbankdetail>',
  ...

```

図 23. 顧客の銀行詳細ニックネームからの抜粋

以下の例では 2 つのニックネームの間に内部結合があるため、照会で行は返されません。

```

SELECT a.name, b.bank_key
FROM sap_bapi_customer_getdetail2_NN a,
     sap_bapi_customer_getdetail2_sap_customerbankdetail_NN b
WHERE a.customerno='1234567890'
AND a.NN_PKEY=b.NN_FKEY;

```

Web サービス・ラッパーのニックネーム定義に必須の入力列が含まれている場合、そのニックネームと他のローカル表または他のニックネームの間での左外部結合はサポートされません。

Web サービス・データ・ソース - 照会の例

Web サービス・ラッパーに関する SQL 照会の例を示します。

例 1: マテリアライズ照会表の使用

マテリアライズ照会表は、照会の結果をローカルにキャッシュに入れて、照会のパフォーマンスを向上させるために使用します。 Web サービス・データ・ソースからのニックネームを使用して、マテリアライズ照会表を作成することができます。照会によっては、マテリアライズ照会表が基本表にアクセスしないで照会に回答できるかを、データベースは自動的に判別できます。以下のプロシーチャーは、マテリアライズ照会表の作成およびデータ設定の方法を示します。

1. ローカルまたは基本表を、次のようにして作成します。

```
CREATE TABLE mystocks(ticker VARCHAR(10));
```

キャッシュに入れるすべての値の維持に、ローカル表を使用できます。

2. キャッシュに入れるすべての値を表に挿入します。

```
INSERT INTO mystocks VALUES('IBM');
INSERT INTO mystocks VALUES('MSFT');
...
```

3. Web サービス・ニックネームを次のようにして作成します。

```
CREATE NICKNAME stockquote_nn (
    ticker VARCHAR(40) OPTIONS (TEMPLATE '&column'),
    price VARCHAR(16) OPTIONS (XPATH './Result/text()')
)
FOR SERVER stock_server
OPTIONS (TEMPLATE '<ticker>&column</ticker>'
        XPATH './Result/text()' );
```

4. ニックネームおよびローカル表から成るビューを、次のようにして作成します。

```
CREATE VIEW stock_quote_view (ticker, price)
AS (
    SELECT nn.ticker, nn.price
    FROM stockquote_nn nn, mystocks s
    WHERE nn.ticker=s.ticker
);
```

5. マテリアライズ照会表を次のようにして作成します。

```
CREATE TABLE stockquote_MQT (ticker, ticker2, price)
as (SELECT nn.ticker,s.ticker as ticker2, nn.price
FROM stockquote_nn nn, mystocks s
WHERE nn.ticker=s.ticker )
DATA INITIALLY DEFERRED REFRESH DEFERRED;
```

マテリアライズ照会表の出力リストの結合述部にある VARCHAR 列 (nn.ticker および s.ticker) のすべてを含め、マテリアライズ照会表がフェデレーテッド・データベースに使用される機会を最大化します。

マテリアライズ照会表の最新表示を据え置くには、REFRESH DEFERRED キーワードを指定します。REFRESH DEFERRED キーワードで指定されたマテリアライズ照会表は、基礎となる基本表への変更を反映しません。文節 DATA INITIALLY DEFERRED を使用して、データが CREATE TABLE ステートメントの一部として表に挿入されないようにします。

6. REFRESH TABLE ステートメントを発行して表にデータを追加します。表中のデータは、REFRESH TABLE ステートメントの発行時の照会の結果をスナップショットとして反映します。以下の例では、stockquote_MQT 表にデータを追加し、特殊レジスターの値に現在の最新表示経過時間を設定します。

```
REFRESH TABLE stockquote_MQT;
```

```
SET CURRENT REFRESH AGE any;
```

実行先がマテリアライズ照会表のデータである照会は、実行先が基本表のデータである照会よりも高速です。マテリアライズ照会表を使用したい場合、以下のようにニックネームではなくビューを参照します。

```
SELECT * FROM stock_quote_view
WHERE ticker='IBM';
```

キャッシュに入れられていない値を選択する照会を発行した場合、0 行が戻されません。

例 2: 主キーと外部キーを使用した結合の実行

PRIMARY_KEY および FOREIGN_KEY 列は、親ニックネームと子ニックネームの間の関係を定義します。各親ニックネームは、主キー列オプションをもっていなければなりません。ある親ニックネームの子は、親ニックネームの主キー列を参照する外部キー列オプションによって定義します。1つのニックネームに対して複数の子は可能ですが、親については1つのみ可能です。

これらの列にはバイナリー・データのみが含まれるため、列は FOR BIT DATA NOT NULL キーワードで定義されています。ニックネームを作成する時に、PRIMARY_KEY および FOREIGN_KEY 列を FOR BIT DATA NOT NULL として明示的に定義することもできます。

以下の例では、Web サービス・ラッパーが PRIMARY_KEY および FOREIGN_KEY 列を使用して親ニックネームと子ニックネームを関連付ける方法を示します。

```
CREATE NICKNAME zooport_getzooreport_nn (
  zooid VARCHAR (48) OPTIONS(TEMPLATE '&column'),
  zoo_id VARCHAR (48) OPTIONS(XPATH './ns1:Zoo/@id'),
  report_zoo_zooname VARCHAR (48)
    OPTIONS(XPATH './ns1:Zoo/ns1:ZooName/text()'),
  report_zoo_count VARCHAR (48)
    OPTIONS(XPATH './ns1:Zoo/ns1:Count/text()'),
  report_zoo_lastmodified VARCHAR (48)
    OPTIONS(XPATH './ns1:Zoo/ns1:LastModified/text()'),
  nn_pk VARCHAR (16) NOT NULL OPTIONS(PRIMARY_KEY 'YES'),
  url VARCHAR (256) OPTIONS(URLCOLUMN 'Y'),
  soapaction VARCHAR (256) OPTIONS(SOAPACTIONCOLUMN 'Y')
) FOR SERVER "mytestsrvr"
OPTIONS(
  URL 'http://localhost:9080/MaelstromTest/services/ZooPort',
  SOAPACTION 'http://myzoo.com/getZooReport',
  TEMPLATE '<soapenv:Envelope>
    <soapenv:Body>
      <zooId>&zooId[1,1]</zooId>
    </soapenv:Body>
  </soapenv:Envelope>',
  XPATH '/soapenv:Envelope/soapenv:Body',
  NAMESPACE ' soapenv="http://schemas.xmlsoap.org/soap/envelope/",
  ns1="http://myzoo.com" ');

CREATE NICKNAME zooport_getzooreport_report_zookeeper_nn (
  nn_fk VARCHAR (16) NOT NULL
    OPTIONS(FOREIGN_KEY 'ZOOPORT_GETZOOREPORT_NN'),
  zookeeper_id VARCHAR (48) OPTIONS(XPATH './@id'),
  report_zookeeper_name VARCHAR (48) OPTIONS(XPATH './ns1:Name/text()'),
  zookeeper_numbercages VARCHAR (48)
    OPTIONS(XPATH './ns1:NumberOfCages/text()'),
  nn_pk VARCHAR (16) NOT NULL OPTIONS(PRIMARY_KEY 'YES')
)
FOR SERVER "MYTESTSRVR" OPTIONS(
  XPATH './ns1:Zoo/ns1:Zookeeper',
  NAMESPACE ' soapenv="http://schemas.xmlsoap.org/soap/envelope/",
  ns1="http://myzoo.com" ');
```

ニックネーム zooport_getzooreport_report_zookeeper_nn の外部キー nn_fk は、外部キー・オプションの親ニックネーム zooport_getzooreport_nn を指します。指定された主および外部キー・ニックネーム列は、WSDL 文書の中のデータには対応しません。というのは、これらのニックネーム列には、ラッパーによって生成されたキーが含まれるからです。これらのキーは、照会内でのみ固有である親と子ニックネーム

ム間の関係を識別します。子ニックネームが入力列を含む場合、親ニックネーム・オプション・テンプレートは、そのニックネーム・オプションによって、テンプレート構造内で、その子ニックネームを参照します。

以下の SQL ステートメントは親ニックネームと子ニックネームを結合します。

```
SELECT *
FROM   zooport_getzooreport_nn a,
       zooport_getzooreport_report_zookeeper_nn z,

WHERE  a.nn_pk = z.nn_fk
AND    a.zooid = 100
;
```

以下の記述では、照会実行時に Web サービス・ラッパーが TEMPLATE および XPATH ニックネームと列オプションを使用する方法を説明します。これは特定のインプリメンテーションの例として意図されたものではありません。

主キーと外部キー列を結合すると、Web サービス・ラッパーが Web サービス・プロバイダーへメッセージを送信し、Web サービス・プロバイダーからは行のセットが戻されます。ラッパーは、列オプション・テンプレート (ZOOID VARCHAR (48) OPTIONS(TEMPLATE &column')) への参照の照会からの入力列の値 (a.zooid = 100)、それからニックネーム・テンプレート・オプション (<zooid>&zooid[1,1]</zooid>) 中のすべての列参照を置き換えることにより、親ニックネームのメッセージを生成します。ニックネーム・テンプレート・オプションには列参照または子ニックネーム参照が含まれることがあります。その後メッセージは Web サービスに送信されます。

ラッパーは、Web サービスが戻す文書にニックネーム・オプション XPATH を適用することにより、ニックネームの行を生成します。ニックネーム・オプション XPATH が複数の XML フラグメントを戻す場合には、ニックネームには複数の行が含まれます。列 XPATH オプションが、行を表す結果の XML フラグメントに適用され、列値を得ます。ニックネームに 1 つ以上の間接的な親がある場合、親ニックネーム XPATH 式のすべてが、階層のトップから、このニックネームにニックネーム・オプション XPATH および列オプション XPATH が適用される手前まで、順番に適用されます。

XML データ・ソースへのアクセスの構成

フェデレーテッド・システムを使用すると、XML データ・ソースにあるデータと他のソースからの情報を統合させることができます。

このタスクについて

フェデレーテッド・サーバーが XML データ・ソースにアクセスするように構成するには、アクセスするデータ・ソースとオブジェクトの情報をフェデレーテッド・サーバーに渡す必要があります。フェデレーテッド・サーバーを構成したら、照会を作成して XML データ・ソースにアクセスできます。

XML ラッパー

Extensible Markup Language (XML) は、構造化された文書およびデータ用の汎用フォーマットです。XML では、タグを使用して文書内でデータを構造化します。

XML ファイルは .xml ファイル拡張子を使用します。HTML と同様、XML では不
等号括弧 (< >) で囲んだ語をタグとして使用します。タグにより、文書内のデータ
が構造化されます。

以下の文書は、XML 文書のサンプルです。

```
<?xml version="1.0" encoding=UTF-8"?>
<doc>
  <customer id='123'>
    <name>...</name>
    <address>...</address>
    ...
    <order>
      <amount>...</amount>
      <date>...</date>
      <item quant='12'>
        <name>...</name>
      </item>
      <item quant='4'>...</item>
      ...
    </order>
    <order>...</order>
    ...
    <payment>
      <number>...</number>
      <date>...</date>
    </payment>
    <payment>>...</payment>
    ...
  </customer>
  <customer id='124'>...</customer>
</doc>
```

図 24. サンプル XML 文書

XML ラッパーがどのように機能するか

XML ラッパーにより、次のような種類のデータを照会するために SQL を使用できる
ようになります。

- 1 個のファイルとして保管されている複数の外部 XML 文書
- 1 つのディレクトリー・パス内にある複数の XML ファイル
- URI (Uniform Reference Identifier) で参照されるリモート XML ファイル
- リレーショナル列に保管された XML 文書

次の図は、XML ラッパーがフェデレーテッド・システムと連携する方法を示してい
ます。

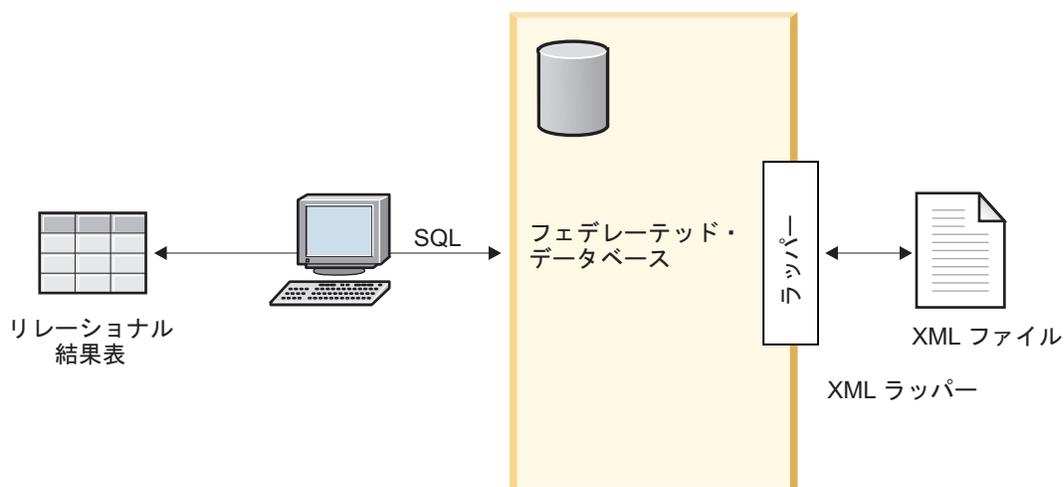


図 25. XML ラッパーがどのように機能するか

XML ラッパーにより、XML データを外部データ・ソースから、ニックネームのセットで構成されるリレーショナル・スキーマへマッピングが可能になります。XML 文書の構造は、ネストされていて繰り返すエレメントが、外部キーをもっている別の表としてモデルとなっているリレーショナル・スキーマと論理的には同等です。

XML 文書に対応するニックネームは、親ニックネームと子ニックネームの関係が、エレメントのネスト関係としてマッピングされるツリー構造に編成されています。

ルート・ニックネームは、ニックネーム階層の最上位にあるニックネームです。非ルート・ニックネームは、ニックネーム階層内の親を持つニックネームです。XML 文書内で最上位のエレメントではないルート・ニックネームも存在することがあります。

ネスト・エレメントが繰り返されている場合や、またはそれぞれ性質の異なるものが複雑な構造になっている場合には、それらネスト・エレメントのそれぞれに別個のニックネームを付けることができます。

子および親ニックネームは、ラッパーによって生成された主および外部キーによって接続されます。

XPath 式は、XML 文書をニックネームのセットで構成されているリレーショナル・スキーマにマップするために使用されます。XPath は、XML ファイルの一部 (XML 文書ツリー内のノードおよび属性のグループなど) を識別するためのアドレッシング・メカニズムです。基本 XPath 構文はファイル・システム・アドレッシングに似ています。

各ニックネームは、個々のタプルを表す XML エレメントを識別する 1 つの XPath 式、および各エレメントから列値を取り出す方法を指定する XPath 式のセットによって定義されます。

XML 文書マッピングの例

次の例は、以下の仕組みを示しています。

- サンプル XML 文書はニックネームのセットにマップされます。
- 親と子の関係は主キーと外部キーを使用して確立されます。
- XPath 式を使用して文書の各エレメント内の個別のタプルと列を定義します。
- 文書をユーザーのフェデレーテッド・システムに登録した後、XML 文書で照会を実行できます。

サンプル XML 文書には、customer エレメントのセットが含まれています。その各エレメントには、複数の order エレメントおよび payment エレメントが入れられます。

order エレメントは、複数の item エレメントを囲んでいます。

エレメント間の関係は次の図に示されています。

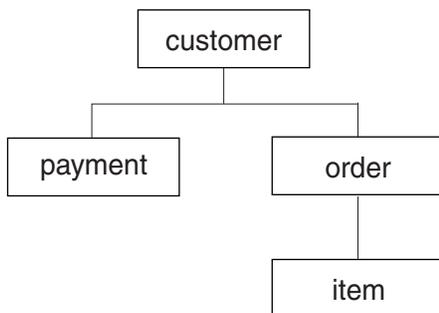


図 26. サンプル XML 文書のツリー構造

この構造に基づいて、CREATE NICKNAME ステートメントを使うことにより、XML 文書を、次の 4 つのニックネームが含まれるリレーショナル・スキーマにマッピングできます。

- customers
- orders
- payments
- items

ニックネーム間の関係は、主キーおよび外部キーを指定する特別なニックネーム列オプションを使用して、各ニックネームを親または子ニックネームとして指定することによって定義されます。各親ニックネームは、主キー列オプションによって指定された特別な列をもっていなければなりません。ある親ニックネームの子は、親ニックネームの主キー列を参照する外部キー列オプションによって定義します。指定された主および外部ニックネーム列は、XML 文書の中のデータには対応しません。というのは、これらのニックネーム列には、ラッパーによって生成されたキーが含まれるからです。1 つのニックネームに対して複数の子は可能ですが、親については 1 つのみ可能です。ただし、ルート・ニックネームに親はありません。

サンプル XML 文書の場合、customers ニックネームに主キーが定義されており、orders、payments、および items ニックネームには、それぞれその親ニックネームを

指す外部キーが定義されています。 orders と payments のニックネームの外部キーは customers ニックネームを指し、 items ニックネームの外部キーは orders ニックネームを指しています。

個々のタプルを表す XML エlementを識別するには、1 つの XPath 式を作成します。この例の場合、すべての customer Elementを参照するには '/doc/customer' XPath 式を使用し、すべての order Elementを参照するには './order' XPath 式を使用します。XPath 式 './order' に含まれるピリオドは、各 order Element のタプルが、それに対応する customer Elementの各タプル内にネストされていることを示しています。

各Elementから列値を取り出す方法を指定するために、XPath 式のセットを作成します。この例の場合、customer Elementの id 属性 (この時点では、ニックネームの中に定義されている列) を参照するには './@id' XPath 式を使用します。customer Elementの name Elementを参照するには './name' XPath 式を使用し、customer Elementの address Elementを参照するには './address/@street' XPath 式を使用します。

CREATE NICKNAME ステートメントを使用して XML 文書をニックネームのセットにマップした後に、主および外部キーを使用することによって、各ニックネームを親または子として定義します。文書の各Element内で個々のタプルおよび列を定義する XPath 式を、これらの主キーと外部キーに指定します。その後、その XML 文書に対して SQL 照会を実行できます。

フェデレーテッド・システムへの XML の追加

フェデレーテッド・サーバーが XML データ・ソースにアクセスするように構成するには、アクセスするデータ・ソースとオブジェクトの情報をフェデレーテッド・サーバーに渡す必要があります。

始める前に

- フェデレーションが、フェデレーテッド・サーバーとして機能するサーバー上にインストールされている必要があります。
- データベースがフェデレーテッド・サーバー上に存在している必要があります。

手順

1. XML ラッパーを登録します。
2. XML サーバー定義を登録します。
3. XML データ・ソースのニックネームを登録します。
4. 非ルート・ニックネームのフェデレーテッド・ビューを作成します。

XML ラッパーの登録

XML データ・ソースにアクセスするためには、ラッパーを登録する必要があります。

このタスクについて

ラッパーは、データ・ソースとの通信やデータの取得のためにフェデレーテッド・サーバーによって使用されます。ラッパーは、ライブラリー・ファイルのセットとしてインプリメントされます。

プロキシー・サーバーと鍵ストアを使用して XML ファイルにアクセスする場合、ラッパーやサーバー定義を登録するときにオプションとしてプロキシー・サーバー情報を指定できます。サーバー定義を登録するときにプロキシー・サーバー情報を指定する場合、これらの設定はラッパーを登録したときに指定したプロキシー・サーバー設定をオーバーライドします。

手順

コマンド行から CREATE WRAPPER ステートメントを発行します。

```
CREATE WRAPPER wrapper_name
LIBRARY library_name;
```

プロキシー・サーバーや鍵ストアを使用して XML ファイルにアクセスする場合、XML ラッパーまたはサーバー定義を登録するときにいくつかのオプションを指定する必要があります。XML ラッパーを登録するときにプロキシー・サーバー情報を指定するには、次のステートメントを発行します。

```
CREATE WRAPPER wrapper_name
LIBRARY library_name
OPTIONS (PROXY_TYPE 'type',
PROXY_SERVER_NAME 'server_name',
PROXY_SERVER_PORT 'port_number');
```

CREATE WRAPPER ステートメントには LIBRARY パラメーターを指定する必要があります。指定するラッパー・ライブラリー・ファイルの名前は、フェデレーテッド・サーバーのオペレーティング・システムによって異なります。CREATE WRAPPER ラッパー・ステートメントで指定する正しいライブラリー名については、『XML ラッパー・ライブラリー・ファイル』のリストを参照してください。プロキシー・サーバーや鍵ストアを使用して XML ファイルにアクセスする場合、XML ラッパーまたはサーバー定義を登録するときにいくつかのオプションを指定する必要があります。

XML ラッパー・ライブラリー・ファイル:

XML ラッパー・ライブラリー・ファイルは、フェデレーテッド・サーバーのインストール時にフェデレーテッド・サーバーに追加されます。

IBM InfoSphere Federation Server をインストールすると、デフォルトのディレクトリー・パスに 3 つのライブラリー・ファイルが追加されます。例えば、フェデレーテッド・サーバーが AIX 上で実行されている場合、ディレクトリー・パスには、libdb21xml.a、libdb21xmlf.a、および libdb21xmlu.a が追加されます。デフォルトのラッパー・ライブラリー・ファイルは libdb21xml.a です。他のラッパー・ライブラリー・ファイルは特定のラッパー・オプションで使用されます。

CREATE WRAPPER ステートメントに LIBRARY パラメーターを含め、デフォルトのラッパー・ライブラリー・ファイルの名前を指定する必要があります。

以下の表に、デフォルトのディレクトリー・パスとデフォルトのラッパー・ライブラリー・ファイルの名前をリストします。

表 40. XML ラッパー・ライブラリーのロケーションとファイル名

オペレーティング・システム	ディレクトリー・パス	ラッパー・ライブラリー・ファイル名
AIX	/usr/opt/<install_path>/lib32/ /usr/opt/<install_path>/lib64/	libdb2lsxml.a
Linux	/opt/IBM/db2/<install_path>/lib32 /opt/IBM/db2/<install_path>/lib64	libdb2lsxml.so
Solaris	/opt/IBM/db2/<install_path>/lib32 /opt/IBM/db2/<install_path>/lib64	libdb2lsxml.so
Windows	%DB2PATH%¥bin	db2lsxml.dll

<install_path> は、Linux または UNIX 上でフェデレーテッド・サーバーがインストールされているディレクトリー・パスです。

%DB2PATH% は、Windows 上でフェデレーテッド・サーバーがインストールされているディレクトリー・パスを指定するのに使用する環境変数です。デフォルトの Windows ディレクトリー・パスは、C:¥Program Files¥IBM¥SQLLIB です。

CREATE WRAPPER ステートメント - XML ラッパーの例:

CREATE WRAPPER ステートメントを使用して XML ラッパーを登録します。この例は、プロキシ・サーバーを使用する場合と使用しない場合に、XML 文書にアクセスするのに必要なパラメーターを示しています。

ラッパーの登録

プロキシ・サーバーを使用しないで XML 文書にアクセスする場合は、ラッパーを登録するために以下のステートメントを発行します。

```
CREATE WRAPPER xml_wrapper LIBRARY 'libdb2lsxml.a';
```

xml_wrapper

XML ラッパーに割り当てる名前。重複するラッパー名は使用できません。

LIBRARY 'libdb2lsxml.a'

AIX オペレーティング・システムを使用するフェデレーテッド・サーバー用のラッパー・ライブラリー・ファイルの名前。

HTTP プロキシ・サーバーのラッパーの登録

ラッパーを登録して、HTTP プロキシ・サーバーを指定するには、以下のステートメントを使用します。

```
CREATE WRAPPER xml_wrapper LIBRARY 'libdb2lsxml.a'
  OPTIONS (PROXY_TYPE 'HTTP', PROXY_SERVER_NAME 'proxy_http',
    PROXY_SERVER_PORT '8080');
```

PROXY_TYPE 'HTTP'

ファイアウォールで保護された状態でインターネットにアクセスする際に使用する、プロキシ・タイプを指定します。有効な値は 'NONE'、'HTTP' または 'SOCKS' です。

PROXY_SERVER_NAME *'proxy_http'*

プロキシ・サーバー名または IP アドレスを指定します。PROXY_TYPE サーバー・オプションの値が 'HTTP' または 'SOCKS' の場合、このオプションは必須です。

PROXY_SERVER_PORT *'8080'*

プロキシ・サーバーのポート番号を指定します。PROXY_TYPE サーバー・オプションの値が 'HTTP' または 'SOCKS' の場合、このオプションは必須です。

SOCKS プロキシ・サーバーのラッパーの登録

ラッパーを登録して、認証情報のない SOCKS プロキシ・サーバーを指定するには、以下のステートメントを使用します。

```
CREATE WRAPPER xml_wrapper LIBRARY 'libdb2lxml.so'  
  OPTIONS (PROXY_TYPE 'SOCKS', PROXY_SERVER_NAME 'proxy_socks',  
          PROXY_SERVER_PORT '1081');
```

LIBRARY *'libdb2lxml.so'*

Linux および Solaris オペレーティング・システムを使用するフェデレーテッド・サーバーの、ラッパー・ライブラリー・ファイルの名前を指定します。

PROXY_TYPE *'SOCKS'*

ファイアウォールで保護された状態でインターネットにアクセスする際に使用する、プロキシ・タイプを指定します。有効な値は 'NONE'、'HTTP' または 'SOCKS' です。

PROXY_SERVER_NAME *'proxy_socks'*

プロキシ・サーバー名または IP アドレスを指定します。PROXY_TYPE サーバー・オプションの値が 'HTTP' または 'SOCKS' の場合、このオプションは必須です。

PROXY_SERVER_PORT *'1081'*

プロキシ・サーバーのポート番号を指定します。PROXY_TYPE サーバー・オプションの値が 'HTTP' または 'SOCKS' の場合、このオプションは必須です。

XML データ・ソースのサーバー定義の登録

XML 文書を検索するためには、サーバー定義を登録する必要があります。登録するサーバー定義は、XML 文書にアクセスするためにプロキシ・サーバーを使用するかによって異なります。

このタスクについて

プロキシと URI (Uniform Resource Identifier) を使用してファイアウォールで保護された XML 文書を検索するには、CREATE SERVER ステートメントにプロキシ・サーバー・オプションを含める必要があります。

フェデレーテッド・オブジェクトの階層では XML ファイルが特定のサーバー定義オブジェクトに関連付けられている必要があるため、プロキシ・サーバーを使用しない場合でもサーバー定義を登録する必要があります。

手順

コマンド行から `CREATE SERVER` ステートメントを発行します。
プロキシ・サーバーを使用しないで XML 文書にアクセスする場合は、以下のステートメントを発行します。

```
CREATE SERVER server_definition_name  
WRAPPER wrapper_name;
```

プロキシ・サーバーを使用して XML 文書にアクセスする場合は、以下のステートメントを発行します。

```
CREATE SERVER server_definition_name  
WRAPPER wrapper_name  
OPTIONS (PROXY_TYPE 'type',  
PROXY_SERVER_NAME 'server_name',  
PROXY_SERVER_PORT 'port_number');
```

プロキシ・サーバーを使用して XML 文書にアクセスする場合、使用するプロトコルがプロキシ・サーバー用の許可 ID とパスワードの指定を必要とすることがあります。サーバー定義を登録するときに、オプションとして認証情報を指定します。

CREATE SERVER ステートメント - XML ラッパーの例:

XML ラッパーにサーバー定義を登録するには、`CREATE SERVER` ステートメントを使用します。この例は、プロキシ・サーバーを使用する場合と使用しない場合に、XML 文書にアクセスするのに必要なパラメーターを示しています。

サーバー定義の登録

プロキシ・サーバーを使用しないで XML 文書にアクセスする場合でも、サーバー定義を登録する必要があります。フェデレーテッド・オブジェクトの階層では、XML ファイルが特定のサーバー定義オブジェクトに関連付けられていることが必要です。ユーザーがサーバー定義を登録するには、次のステートメントを発行します。

```
CREATE SERVER xml_server WRAPPER xml_wrapper;
```

xml_server

XML サーバー定義に割り当てる名前。重複するサーバー定義名は使用できません。

WRAPPER *xml_wrapper*

CREATE WRAPPER ステートメントで指定したラッパー名。

プロキシ・サーバーが使用される場合のサーバー定義

以下の条件のすべてが真である場合は、`CREATE SERVER` ステートメントでプロキシ・サーバー・オプションを使用する必要があります。

- URI を使用してデータを検索したい。
- 使用する URI はプロキシを介してファイアウォールの反対側にあるデータを検索する。
- 使用されるファイアウォールまたはプロキシは、HTTP または SOCKS である。

指定するオプションは、アクセスしたいプロキシ・サーバーのタイプによって異なります。

使用するプロキシのタイプ、およびプロキシ・オプションに指定する必要がある設定については、ネットワーク管理者に問い合わせてください。

HTTP プロキシ・サーバーのサーバー定義の登録

サーバー定義を登録して、HTTP プロキシ・サーバーを指定するには、以下のステートメントを使用します。

```
CREATE SERVER xml_server_http
  WRAPPER xml_wrapper
  OPTIONS (PROXY_TYPE 'HTTP', PROXY_SERVER_NAME 'proxy_http',
          PROXY_SERVER_PORT '8080');
```

xml_server_http

XML サーバー定義に割り当てる名前。重複するサーバー定義名は使用できません。

WRAPPER *xml_wrapper*

CREATE WRAPPER ステートメントで指定したラッパー名。

PROXY_TYPE 'HTTP'

ファイアウォールで保護された状態でインターネットにアクセスする際に使用する、プロキシ・タイプを指定します。有効な値は 'NONE'、'HTTP' または 'SOCKS' です。

PROXY_SERVER_NAME 'proxy_http'

プロキシ・サーバー名または IP アドレスを指定します。PROXY_TYPE サーバー・オプションの値が 'HTTP' または 'SOCKS' の場合、このオプションは必須です。

PROXY_SERVER_PORT '8080'

プロキシ・サーバーのポート番号を指定します。PROXY_TYPE サーバー・オプションの値が 'HTTP' または 'SOCKS' の場合、このオプションは必須です。

SOCKS プロキシ・サーバーのサーバー定義の登録

サーバー定義を登録して、認証情報の必要ない SOCKS プロキシ・サーバーを指定するには、以下のステートメントを使用します。

```
CREATE SERVER xml_server_socks
  WRAPPER xml_wrapper
  OPTIONS (PROXY_TYPE 'SOCKS', PROXY_SERVER_NAME 'proxy_socks',
          PROXY_SERVER_PORT '1081');
```

xml_server_socks

XML サーバー定義に割り当てる名前。重複するサーバー定義名は使用できません。

WRAPPER *xml_wrapper*

CREATE WRAPPER ステートメントで指定したラッパー名。

PROXY_TYPE 'SOCKS'

ファイアウォールで保護された状態でインターネットにアクセスする際に使用する、プロキシー・タイプを指定します。有効な値は 'NONE'、'HTTP' または 'SOCKS' です。

PROXY_SERVER_NAME 'proxy_socks'

プロキシー・サーバー名または IP アドレスを指定します。PROXY_TYPE サーバー・オプションの値が 'HTTP' または 'SOCKS' の場合、このオプションは必須です。

PROXY_SERVER_PORT '1081'

プロキシー・サーバーのポート番号を指定します。PROXY_TYPE サーバー・オプションの値が 'HTTP' または 'SOCKS' の場合、このオプションは必須です。

認証情報のある SOCKS プロキシー・サーバーのサーバー定義の登録

サーバー定義を登録して、認証情報のある SOCKS プロキシー・サーバーを指定するには、以下のステートメントを使用します。

```
CREATE SERVER xml_server_socks
  WRAPPER xml_wrapper
  OPTIONS (PROXY_TYPE 'SOCKS', PROXY_SERVER_NAME 'proxy_socks',
          PROXY_SERVER_PORT '1081', PROXY_AUTHID 'Martin',
          PROXY_PASSWORD 'not4me');
```

PROXY_AUTHID 'Martin'

プロキシー・サーバーのユーザー名を指定します。PROXY_TYPE サーバー・オプションの値が 'SOCKS' の場合、このサーバー・オプションは必須です。

PROXY_PASSWORD 'not4me'

ユーザー名 'Martin' に関連したプロキシー・サーバーのパスワードを指定します。PROXY_TYPE サーバー・オプションの値が 'SOCKS' の場合、このサーバー・オプションは必須です。

プロキシー・サーバー応答の制限時間の指定

サーバー定義の登録時に、プロキシー・サーバーに必要なサーバー・オプションの他に SOCKET_TIMEOUT サーバー・オプションを指定することもできます。

SOCKET_TIMEOUT サーバー・オプションは、フェデレーテッド・サーバーがプロキシー・サーバーからの結果を待つ最大時間を分単位で指定します。

SOCKET_TIMEOUT サーバー・オプションを指定しない場合は時間制限なしとなり、フェデレーテッド・サーバーはプロキシー・サーバーからの結果を無制限に待つこととなります。

サーバー定義を登録して、フェデレーテッド・サーバーがプロキシー・サーバーからの応答を待つ時間を指定するには、以下のステートメントを使用します。

```
CREATE SERVER xml_server_http
  WRAPPER xml_wrapper
  OPTIONS (PROXY_TYPE 'HTTP', PROXY_SERVER_NAME 'proxy_http',
          PROXY_SERVER_PORT '8080', SOCKET_TIMEOUT '12');
```

SOCKET_TIMEOUT 12

フェデレーテッド・サーバーがプロキシ・サーバーからの応答を 12 分待つように指定します。

プロキシ・サーバーを使用した XML ファイルへのアクセス

ネットワークでプロキシ・サーバーを使用する場合、XML データ・ソース用のラッパーやサーバー定義を登録するときに、プロキシ・サーバーに関する情報を指定する必要があります。

指定するオプションは、アクセスするプロキシ・サーバーのタイプと Secure Socket Layer (SSL) プロトコルまたは Transport Layer Security (TLS) プロトコルを使用するかによって異なります。

次のとおり、ラッパーやサーバー定義を登録するときにプロキシと SSL オプションを指定できます。

- ラッパーを登録するときにこれらのオプションを指定する場合、そのラッパーに関連したニックネームは、ラッパーに設定されたオプションを使用します。
- サーバー定義を登録するときにこれらのオプションを指定する場合、そのサーバー定義に関連したニックネームは、サーバー定義に設定されたオプションを使用します。
- ラッパーとサーバー定義を登録するときにこれらのオプションに異なる値を指定する場合、サーバー定義に設定された値がラッパーに設定された値よりも優先されます。

XML ラッパーには検証機能がありますが、プロキシ・サーバーと共に使用するときにはいくらかの制限事項が生じることがあります。この制限事項が生じる条件は、以下のとおりです。

- さまざまなプロキシ・オプションを設定したプロキシ機能を、サーバー・レベルで使用している。
- XML インスタンス文書に、ファイアウォールの外側に存在する外部 XML スキーマへの参照が含まれる。

これらの条件のいずれかが適合する場合、XML スキーマのロケーションをファイアウォールの内側に変更してみてください。XML スキーマ・ロケーションを変更する場合、XML インスタンス文書を XML スキーマの新しいロケーションに合わせて更新する必要があります。

プロキシ・サーバー・オプション

以下の表には、それぞれのタイプのプロキシ・サーバーに指定する必要があるオプションがリストされています。

表 41. プロキシ・サーバーで必要なオプション

プロキシ・サーバーのタイプ	必要なラッパーまたはサーバー・オプション
認証のない HTTP または SOCKS	PROXY_TYPE PROXY_SERVER_NAME PROXY_SERVER_PORT

表 41. プロキシ・サーバーに必要なオプション (続き)

プロキシ・サーバーのタイプ	必要なラッパーまたはサーバー・オプション
認証のある HTTP または SOCKS	PROXY_TYPE PROXY_SERVER_NAME PROXY_SERVER_PORT PROXY_AUTHID PROXY_PASSWORD

SSL サーバー・オプション

以下の表には、SSL プロトコルを使用するときに指定する必要のあるオプションがリストされています。

表 42. SSL プロトコルに必要なオプション

フェデレーテッド・オブジェクト	必要なオプション
ラッパー	SSL_KEYSTORE_FILE SSL_KEYSTORE_PASSWORD SSL_VERIFY_SERVER_CERTIFICATE
サーバー定義	SSL_KEYSTORE_FILE SSL_KEYSTORE_PASSWORD SSL_VERIFY_SERVER_CERTIFICATE SSL_CLIENT_CERTIFICATE_LABEL

XML データ・ソースのニックネーム

アクセスする各 XML 文書のニックネームを登録する必要があります。XML データ・ソースを照会するときは、文書名ではなくこれらのニックネームを使用します。

ニックネームを登録する前に、以下の点について理解する必要があります。

ニックネームと XML 文書の間データ関連:

ニックネームは、XML 文書データのツリー構造に対応しています。親ニックネームと子ニックネームは、データ・ツリー構造のルート構造とネスト・エレメントに対応します。それら親ニックネームと子ニックネームは、CREATE NICKNAME ステートメントで指定される主キーおよび外部キーによって結び付けられます。

各ニックネームは、以下の機能を実行する XPath 式によって定義されます。

- 個々のタプルを表す XML エレメントを識別する。
- 各エレメントから列値を取り出す方法を指定する。

XML ラッパーは、XPath 式を使用することによって、XML 文書のデータと、リレーショナル表の行との間の対応を確立します。それらの XPath 式は、XML 文書内の値を特定し、それらの値が各行の列にどう対応するかを決定します。XML ラッパーは、XML 文書データを読むだけです。XML ラッパーは、そのデータを更新しません。

ニックネームを作成する際、そのニックネームと XML 文書との間の関連付けを指定するためのオプションを選択します。ニックネームは、固定した方法で XML 文書に関連付けられるか、または指定するソース名によって XML 文書に関連付けられます。

固定した関連付けの場合、ニックネームは特定の XML 文書のデータを表します。それらの XML 文書には、次のものが含まれます。

1 つのローカル・ファイル

XML 文書として 1 つの XML ファイルを指定します。

1 つのディレクトリー・パスに含まれる複数のローカル・ファイル

複数の XML ファイルの存在するディレクトリー・パスを指定します。ニックネームには、そのディレクトリー・パス内の XML ファイルから XML 文書データが提供されます。それらの XML ファイルの構成は、すべて同じでなければなりません。ニックネームの構成とは違う構成の XML ファイルがそのディレクトリーに含まれていると、XML ラッパーがその XML データ・ファイルを処理した時点で NULL 値が戻されます。ディレクトリーは、フェデレーテッド・サーバーのローカル・ディレクトリーか、または共有ファイル・システムからアクセス可能なディレクトリーでなければなりません。

XML ラッパーがディレクトリーをスキャンする際には、拡張子が .xml のファイルのみが採用および構文解析されます。その他のファイル (拡張子が .txt や .xsd のファイル、あるいは拡張のないファイルなど) は、XML ラッパーによりすべて無視されます。

ファイルの固定データを指定するには、CREATE NICKNAME ステートメントの FILE_PATH オプションを使用します。ディレクトリーからの固定データを指定するには、DIRECTORY_PATH オプションを使用します。

照会実行中にソース・データを指定する場合には、スキーマがニックネーム定義に一致する任意の XML 文書ソースからのデータを表すニックネームを使用できます。それらの XML 文書には、次のものが含まれます。

URI (Uniform Reference Identifier)

URI によって参照されるリモート XML ファイルからの XML 文書データが、ニックネームに提供されます。この文書ソースを指定するには DOCUMENT 'URI' ニックネーム列オプションを使用します。

リレーショナル列

XML 文書への入力としてリレーショナル表、ビュー、またはニックネームの列が使用されます。この文書ソースを指定するには DOCUMENT 'COLUMN' ニックネーム列オプションを使用します。

ファイル

照会実行時に、XML データが含まれる単一のファイルが入力として提供されます。この文書ソースを指定するには DOCUMENT 'FILE' ニックネーム列オプションを使用します。

ディレクトリー

照会実行時に、指定したディレクトリー・パスに含まれる複数の XML フ

ファイルからデータが提供されます。この文書ソースを指定するには DOCUMENT 'DIRECTORY' ニックネーム列オプションを使用します。

照会時にソース・データが提供されることを指定するには、DOCUMENT 列オプションを指定します。DOCUMENT 列に URI、COLUMN、FILE、または DIRECTORY のいずれかを指定することにより、XML 文書ソースの種類を指定します。

DOCUMENT 列オプションと共に、FILE_PATH オプションまたは DIRECTORY_PATH オプションを指定することはできません。

XML 文書データをフラグメントに分割するには、STREAMING オプションを使用します。STREAMING オプションは、固定形式のデータ、または照会実行時に指定したソース名からのデータに対して使用できます。XML ラッパーは、その結果得られる XML データのストリームを処理し、照会フラグメントによって要求されている情報を抽出します。XML ラッパーは、フラグメントを一度に 1 つずつ解析します。フラグメントが一度に 1 つずつ解析されることにより合計メモリーは少なくなります。照会全体の実行にかかる処理時間はサーバーのメモリー容量に応じて増加します。したがって、STREAMING オプションを使用するのは、50 MB 以上の大規模な XML 文書を解析する場合だけにしてください。

さらに、ニックネーム・オプション値を選択すれば、大量の XML データまたは複数のネスト・エレメントを含むデータを取り出す照会を最適化するのに役立ちます。それらのオプションには、次のものが含まれます。

- INSTANCE_PARSE_TIME
- XPATH_EVAL_TIME
- NEXT_TIME

これらのオプションの値をいろいろと設定してみるにより、XML 照会をテストし、最適化することができます。それらのオプション値によって、エレメントを検出して XML 文書の行に含まれるデータを解析するのに必要な処理時間が制御されます。

XML ラッパーのコスト・モデル機能:

XML ラッパーには、XML ソース文書に対応するニックネームに対する照会を最適化するためのコスト・モデル機能が含まれています。

CREATE NICKNAME ステートメントを使ってニックネームを作成する際には、コスト・モデル機能をサポートするために、次のニックネーム・オプションを指定できます。

- INSTANCE_PARSE_TIME
- XPATH_EVAL_TIME

これらのニックネーム・オプションには、デフォルト値を使用できます。あるいは、これらのニックネーム・オプションに値を指定することによって、作成するルート・ニックネームおよびルート以外のニックネームに対する照会を最適化することもできます。

INSTANCE_PARSE_TIME ニックネーム・オプションは、ルート・ニックネームの 1 つの行生成ルート・エレメントを読み取って構文解析するのに必要な時間 (ミリ秒) です。構文解析時間には、その中に含まれていて行を生成することになるすべての非ルート・エレメントも含まれます。例えば、ルート・ニックネームが customers の場合、各 customer の orders、payments、items に対応するエレメントはすべて非ルート・エレメントです。XML ラッパーは、それら行生成ルート・エレメントおよび行生成非ルート・エレメントを表す構造をメモリー中にビルドします。

XPATH_EVAL_TIME ニックネーム・オプションは、ニックネームの行に対応するデータを探索するための XPath 式を評価するのに必要な時間 (ミリ秒) です。評価される XPath 式には、実際の行を探索するための XPath 式に加えて、それらの行の中に含まれる列値を探索する XPath 式も含まれます。

XML データ・ソースのネーム・スペース:

ネーム・スペースの一部である属性やエレメントを識別するには、NAMESPACES ニックネーム列オプションを使用します。

NAMESPACES ニックネーム・オプションは、ニックネームを登録するときに指定できます。NAMESPACES ニックネーム列オプションの値は、名前と値の対がコマンドで区切られたリストです。XML ラッパーは、名前と値の対を使用して、ニックネーム XPath 式にあるネーム・スペースの接頭部を解決します。XPath 式で使われる接頭部は、XPath プロセッサで処理されます。

次の例では、XML 文書に 3 つの製品用の名前、コード、説明情報が含まれます。XML 文書では 2 つのネーム・スペース http://www.one.com と http://www.two.com を宣言し、1 つのデフォルト・ネーム・スペース http://www.default.com があります。product エレメントは ns1 ネーム・スペースに関連付けられています。product エレメントは name および code 属性と desc エレメントを含みます。name 属性はネーム・スペースと関連付けられていません。code 属性は ns2 ネーム・スペースに関連付けられています。desc エレメントは default ネーム・スペースに関連付けられています。

```
<?xml version="1.0" encoding="UTF-8"?>
<doc xmlns:ns1="http://www.one.com" xmlns:ns2="http://www.two.com"
  xmlns="http://www.default.com">
<ns1:product name="Computer" ns2:code="ABC123"
  <desc>"The Computer product description"</desc>
<ns1:product name="Keyboard" ns2:code="EFG456"
  <desc>"The Keyboard product description"</desc>
<ns1:product name="Mouse" ns2:code="HIJ789"
  <desc>"The Mouse product description"</desc>
</ns1:product>
</doc>
```

次の表は、XML 文書の各エレメントおよび属性に関連付けられたネーム・スペースを示しています。

表 43. XML 文書内のエレメント、属性、ネーム・スペース

エレメントまたは属性	XML 文書内のネーム・スペース	
	ネーム・スペース	注
product: XML 文書内のエレメント	ns1="http://www.one.com"	なし

表 43. XML 文書内のエレメント、属性、ネーム・スペース (続き)

エレメントまたは属性	XML 文書内のネーム・スペース	注
name: XML 文書内の product エレメントの属性	なし	name 属性はネーム・スペースと関連付けられていません。
code: XML 文書内の product エレメントの属性	ns2="http://www.two.com"	なし
desc: XML 文書内の product エレメント内のエレメント	"http://www.default.com"	desc エレメントは default ネーム・スペースを使用します。これは接頭部を含みません。

XML 文書のニックネームを登録するとき、XML 文書内のエレメントと属性に対応した 3 つの列を定義します。NAMESPACES ニックネーム・オプションにネーム・スペース情報を指定します。次に例を示します。

```
CREATE NICKNAME products
  (name VARCHAR(16) OPTIONS (XPATH '@name'),
   code VARCHAR(16) OPTIONS (XPATH '@pre2:code')
   description VARCHAR (256) OPTIONS (XPATH './default:desc'))
FOR SERVER xml_server
OPTIONS (FILE_PATH '/home/mbreining/sql/xml/namespaces.xml',
        XPATH '//pre1:name',
        NAMESPACES 'pre1="http://www.one.com", pre2="http://www.two.com",
                   default="http://www.default.com");
```

次の表は、XML 文書のネーム・スペースが、CREATE NICKNAME ステートメントで指定された値にどのように対応するかを示しています。

表 44. XML 文書のネーム・スペースと、CREATE NICKNAME ステートメントからの対応する値

XML 文書内のネーム・スペース	CREATE NICKNAME ステートメントの列名	XPATH 列オプションの値	NAMESPACES ニックネーム・オプションの値
ネーム・スペースは ns1="http://www.one.com" です。	なし	なし	値は pre1="http://www.one.com" です。値はユーザーが定義する接頭部 (pre1) で、ネーム・スペースの固有 ID ("http://www.one.com") です。
なし。属性はネーム・スペースと関連付けられていません。	名前	値は @name です。値は XML 文書内の属性 (name) です。	なし
ネーム・スペースは ns2="http://www.two.com" です。	コード	値は @pre2:code です。値はユーザーが定義する接頭部 (pre2) で、XML 文書内の属性 (code) です。	値は pre2="http://www.two.com" です。値はユーザーが定義する接頭部 (pre2) で、ネーム・スペースの固有 ID ("http://www.two.com") です。

表 44. XML 文書のネーム・スペースと、CREATE NICKNAME ステートメントからの対応する値 (続き)

XML 文書内のネーム・スペース	CREATE NICKNAME ステートメントの列名	XPATH 列オプションの値	NAMESPACES ニックネーム・オプションの値
ネーム・スペースは "http://www.default.com" です。デフォルト・ネーム・スペースは接頭部を含みません。	説明	値は ./default:desc です。値はユーザーが定義する接頭部 (default) で、XML 文書内のエレメント (desc) です。	値は default="http://www.default.com" です。値はユーザーが定義する接頭部 (default) で、ネーム・スペースの固有 ID ("http://www.default.com") です。

NAMESPACES ニックネーム・オプションは 256 文字を超えるストリングをサポートするためにバック記述子を使用します。

XML ネーム・スペースの詳細情報については、W3C Web サイト上のネーム・スペースの説明 を参照してください。

XML データ・ソースのニックネームの登録:

登録するそれぞれの XML サーバー定義で、アクセスするそれぞれの XML 文書にニックネームを登録する必要があります。XML データ・ソースを照会するときには、XML 文書名ではなくこれらのニックネームを使用します。

このタスクについて

制約事項

Windows 2003 が動作しているフェデレーテッド・サーバーから共有ドライブ上の XML データ・ソースにアクセスする場合、照会が失敗することがあります。これは Windows 2003 の制限です。この問題を回避するには、CREATE NICKNAME ステートメントの FILE_PATH オプションで絶対パスを指定します。

XML データ・ソースのツリー構造に対応するニックネームを作成する必要があります。親ニックネームは、ツリーのルート構造に対応します。子ニックネームは、親ニックネームのエレメント内でネストされているエレメントに対応します。

手順

コマンド行から CREATE NICKNAME ステートメントを発行します。
ニックネームの長さは 128 文字までです。

CREATE NICKNAME ステートメント - XML ラッパーの例:

XML 文書のニックネームを登録するには、CREATE NICKNAME ステートメントを使用します。親と子のニックネームを作成する方法を示す、完全な例と特定の列オプションの例があります。

推奨: 照会内に XPATH 列およびニックネーム・オプションを指定するときには、自己または下位演算子 // を使用しないでください。自己または下位演算子は XPath 演算子です。自己または下位演算子を使用すると、フェデレーテッド・サーバーのパフォーマンスが低下する可能性があります。

完全な例

以下に示すのは、サンプル XML ファイルを使用することによって XML データ・ソースのニックネームを作成する方法の例です。

```
<?xml version="1.0" encoding="UTF-8"?>
<doc>
  <customer id='123'>
    <name>...</name>
    <address>...</address>
    ...
    <order>
      <amount>...</amount>
      <date>...</date>
      <item quant='12'>
        <name>...</name>
      </item>
      <item quant='4'>...</item>
      ...
    </order>
    <order>...</order>
    ...
    <payment>
      <number>...</number>
      <date>...</date>
    </payment>
    <payment>...</payment>
    ...
  </customer>
  <customer id='124'>...</customer>
</doc>
```

図 27. サンプル XML ファイル

親ニックネーム

最初のステップは、親ニックネーム `customers` の作成です。ニックネームを作成するには、次のステートメントを発行します。

```
CREATE NICKNAME customers(
  id      VARCHAR(5)   OPTIONS(XPATH './@id')
  name    VARCHAR(16)  OPTIONS(XPATH './name'),
  address VARCHAR(30)  OPTIONS(XPATH './address/@street'),
  cid     VARCHAR(16)  FOR BIT DATA NOT NULL OPTIONS(PRIMARY_KEY 'YES'))
FOR SERVER xml_server
OPTIONS(DIRECTORY_PATH '/home/db2user',
       XPATH '/doc/customer', STREAMING 'YES');
```

このステートメントは、指定されたディレクトリー・パス `/home/db2user` のもとにある複数の XML ファイルに関する `customers` ニックネームを作成します。

`STREAMING` ニックネーム・オプションは、XML ソース・データがノード (この例では、カスタマー・レコード) ごとに分けられて処理されるということを示します。`STREAMING` ニックネーム・オプションが使用されている場合は、ラッパーは XML 文書全体をメモリーに保管することはしません。その代わりに、XML ラッパーは文書を順次個別に構文解析できる複数のセクションに分割します。

`STREAMING` ニックネーム・オプションは、サイズの大きい XML 文書にのみ使用

してください。このオプションを使用すると、照会のパフォーマンスは影響を受けます。

子ニックネーム

次のステップは、orders、payments、および items の子ニックネームを作成することです。

以下のステートメントを発行して orders 子ニックネームを作成します。

```
CREATE NICKNAME orders(  
  amount INTEGER      OPTIONS(XPATH './amount'),  
  date   VARCHAR(10)  OPTIONS(XPATH './date'),  
  oid    VARCHAR(16)  OPTIONS(PRIMARY_KEY 'YES'),  
  cid    VARCHAR(16)  FOR BIT DATA NOT NULL OPTIONS(FOREIGN_KEY 'CUSTOMERS'))  
FOR SERVER xml_server  
OPTIONS( XPATH './order');
```

以下のステートメントを発行して payments 子ニックネームを作成します。

```
CREATE NICKNAME payments(  
  number INTEGER      OPTIONS(XPATH './number'),  
  date   VARCHAR(10)  OPTIONS(XPATH './date'),  
  cid    VARCHAR(16)  FOR BIT DATA NOT NULL OPTIONS(FOREIGN_KEY 'CUSTOMERS'))  
FOR SERVER xml_server  
OPTIONS( XPATH './payment');
```

以下のステートメントを発行して items 子ニックネームを作成します。

```
CREATE NICKNAME items(  
  name      VARCHAR(20)  OPTIONS(XPATH './name'),  
  quantity  INTEGER      OPTIONS(XPATH './@quant'),  
  oid       VARCHAR(16)  FOR BIT DATA NOT NULL OPTIONS(FOREIGN_KEY 'ORDERS'))  
FOR SERVER xml_server  
OPTIONS( XPATH './item');
```

ニックネーム列オプションの例

以下の例は、ニックネームを作成するときに DOCUMENT ニックネーム列オプションを組み込む方法を示したものです。この例には、それらのオプションを照会内で使用する方法も示されています。

DOCUMENT 'FILE' の例

以下の例は、ニックネームを作成するときに DOCUMENT ニックネーム列オプションを組み込む方法を示したものです。この例には、それらのオプションを照会内で使用する方法も示されています。

以下の CREATE NICKNAME の例は、DOCUMENT 'FILE' ニックネーム列オプションの使用例を示しています。

```
CREATE NICKNAME customers(  
  doc      VARCHAR(100)  OPTIONS(DOCUMENT 'FILE'),  
  name     VARCHAR(16)   OPTIONS(XPATH './name'),  
  address  VARCHAR(30)   OPTIONS(XPATH './address/@street'),  
  cid      VARCHAR(16)   FOR BIT DATA NOT NULL OPTIONS(PRIMARY_KEY 'YES'))  
FOR SERVER xml_server  
OPTIONS(XPATH '/doc/customer');
```

以下の照会は (WHERE 文節で XML 文書のロケーションを指定している) customers ニックネームで実行することができるようになります。

```
SELECT * FROM customers WHERE doc = '/home/db2user/Customers.xml';
```

DOCUMENT 'DIRECTORY' の例

以下の CREATE NICKNAME の例は、DOCUMENT 'DIRECTORY' ニックネーム列オプションの使用例を示しています。

```
CREATE NICKNAME customers(  
  doc      VARCHAR(100)  OPTIONS(DOCUMENT 'DIRECTORY'),  
  name     VARCHAR(16)   OPTIONS(XPATH './name'),  
  address  VARCHAR(30)   OPTIONS(XPATH './address/@street'),  
  cid      VARCHAR(16)   FOR BIT DATA NOT NULL OPTIONS(PRIMARY_KEY 'YES'))  
FOR SERVER xml_server  
OPTIONS(XPATH '/doc/customer');
```

このとき、以下の照会を customers ニックネームに対して実行できます。

```
SELECT name FROM customers WHERE doc = '/home/data/xml';
```

この照会は、WHERE 文節で指定されるディレクトリー・パス /home/data/xml のもとで見付かった XML 文書をリトリブします。

DOCUMENT 'URI' の例

以下の CREATE NICKNAME の例は、DOCUMENT 'URI' ニックネーム列オプションの使用例を示しています。

```
CREATE NICKNAME customers(  
  doc      VARCHAR(100)  OPTIONS(DOCUMENT 'URI'),  
  name     VARCHAR(16)   OPTIONS(XPATH './name'),  
  address  VARCHAR(30)   OPTIONS(XPATH './address/@street'),  
  cid      VARCHAR(16)   FOR BIT DATA NOT NULL OPTIONS(PRIMARY_KEY 'YES'))  
FOR SERVER xml_server  
OPTIONS(XPATH '/doc/customer');
```

このとき、以下の照会を customers ニックネームに対して実行し、XML データをリモート・ロケーションから検索できます。

```
SELECT * FROM customers WHERE doc = 'http://www.lg-mv.org/foo.xml';
```

DOCUMENT 'COLUMN' の例

以下の CREATE NICKNAME の例は、DOCUMENT 'COLUMN' ニックネーム列オプションの使用例を示しています。

```
CREATE NICKNAME emp(  
  doc      VARCHAR(500)  OPTIONS(DOCUMENT 'COLUMN')  
  fname    VARCHAR(16)   OPTIONS(XPATH '@first'),  
  lname    VARCHAR(16)   OPTIONS(XPATH '@last'))  
FOR SERVER xml_server  
OPTIONS(XPATH '/doc/name');
```

このとき、以下の照会の 1 つを emp ニックネームに対して実行し、XML データを検索できます。

```
SELECT * FROM emp WHERE doc = '<?xml version="1.0" encoding="UTF-8"?>  
<doc>  
  <title> employees </title>  
  <name first="David" last="Marston"/>  
  <name first="Donald" last="Leslie"/>  
  <name first="Emily" last="Farmer"/>  
  <name first="Myriam" last="Midy"/>
```

```
<name first="Lee" last="Tran"/>
<name first="Lili" last="Farmer"/>
<name first="Sanjay" last="Kumar"/>
</doc>';
```

または

```
SELECT * FROM emp WHERE doc = (SELECT * FROM xml_tab);
```

xml_tab 表には、XML データが追加される 1 つの列が含まれます。

XML データ・ソースの照会

XML データ・ソースにアクセスするために照会を作成する前に、照会パフォーマンスを最適化するために可能な処置があります。

フェデレーテッド・ビュー

フェデレーテッド・ビューを使用して、XML ニックネーム階層の各部を結合する照会が正しく実行されるようにすることができます。

自己または下位演算子の使用回避

XML ニックネームを作成する際に XPATH 列およびニックネーム・オプションを指定するときには、自己または下位演算子 // を使用しないでください。自己または下位演算子は XPath 演算子であり、自己または下位演算子を使用するとフェデレーテッド・サーバーのパフォーマンスが低下する場合があります。

Windows 2003 フェデレーテッド・サーバー

Windows 2003 が動作しているフェデレーテッド・サーバーから共有ドライブ上の XML データ・ソースにアクセスする場合、次のエラー・メッセージが表示され、照会が失敗することがあります。

```
SQL1822N データ・ソース "XML_SERVER" から予期しないエラー・コード "ERRNO = 2" を
受け取りました。
関連するテキストおよびトークンは「ファイルの読み取り不可」です。
SQLSTATE=560BD
```

これは Windows 2003 の制限です。この問題を回避するには、CREATE NICKNAME ステートメントの FILE_PATH オプションで絶対パスを指定します。

次の例は、FILE_PATH オプションで省略パスが指定された CREATE NICKNAME ステートメントを示しています。

```
CREATE NICKNAME customers
(
  id      VARCHAR(5)  OPTIONS(XPATH './@id'),
  name    VARCHAR(16) OPTIONS(XPATH './name'),
  address VARCHAR(30) OPTIONS(XPATH './address/@street'),
  cid     VARCHAR(16) FOR BIT DATA NOT NULL
  OPTIONS(PRIMARY_KEY 'YES'))
  FOR SERVER xml_server
  OPTIONS(DIRECTORY_PATH '#home#db2user',
  XPATH '/doc/customer', STREAMING 'YES');
```

省略パスを指定しているため、このニックネームを使用する照会は失敗する可能性があります。

Windows 2003 でフェデレーテッド・サーバーを実行している場合、CREATE NICKNAME ステートメントの FILE_PATH オプションに絶対パスを指定してください。

例:

```
CREATE NICKNAME customers
(
  id      VARCHAR(5)  OPTIONS(XPATH './@id'),
  name    VARCHAR(16) OPTIONS(XPATH './name'),
  address VARCHAR(30) OPTIONS(XPATH './address/@street'),
  cid     VARCHAR(16) FOR BIT DATA NOT NULL
          OPTIONS(PRIMARY_KEY 'YES'))
FOR SERVER xml_server
  OPTIONS(DIRECTORY_PATH '¥¥host.svl.ibm.com¥D$¥home¥db2user',
          XPATH '/doc/customer', STREAMING 'YES');
```

XML ラッパー・ニックネームのフェデレーテッド・ビューの作成

XML 文書を説明するニックネームの階層に対するフェデレーテッド・ビューを作成できます。フェデレーテッド・ビューを定義することにより、XML ニックネーム階層の各部を結合する照会は正しく実行するようになります。

このタスクについて

このタスクについて

フェデレーテッド・ビューとは、データ・ソース表ではなくニックネームを参照するフェデレーテッド・データベース内のビューです。

XML ニックネーム階層では、ルート・ニックネームと特別な PRIMARY_KEY および FOREIGN_KEY 列以外の列を結合する照会は、フェデレーテッド・ビューを使用することの影響を受けません。

XML ニックネーム用のフェデレーテッド・ビューを作成するとき、必須の述部とルート・ディレクトリーへの絶対パスをすべて組み込む必要があります。

手順

XML ニックネームのフェデレーテッド・ビューを作成するには、次のようにします。

手順

各非ルート・ニックネーム用のビューを定義するには、CREATE VIEW ステートメントを使用します。このビューは、ルート・ニックネームへのパス上のすべてのニックネームの結合である必要があります。

1. ビューの WHERE 文節に、結合述部として PRIMARY_KEY と FOREIGN_KEY 列を定義する。
2. SELECT リストに、非ルート・ニックネームのすべての列を組み込みます。ただし、FOREIGN_KEY ニックネーム列オプションによって指定された列は除きます。SELECT リストに、PRIMARY_KEY オプションによって指定された親ニックネームの列を組み込みます。

CREATE VIEW ステートメント - XML ラッパーの例

非ルート・ニックネーム用のフェデレーテッド・ビューを作成するには、CREATE VIEW ステートメントを使用します。この例では、ビューを作成するために使用するステートメントが含まれるサンプル XML ファイルを使用して、照会でビューを使用する方法を示します。

XML 文書を説明するニックネームの階層に対するフェデレーテッド・ビューを作成して、XML ニックネーム階層の各部を結合する照会が正しく実行されるようにすることができます。照会でフェデレーテッド・ビューを指定すると、データはリモート・データ・ソースから検索されます。

以下の例は、ルートではないニックネームのフェデレーテッド・ビューを作成して XML ソース文書を記述する方法を示しています。

サンプル XML ファイル

以下の例は、このサンプル XML ファイルを基にしています。

```
<?xml version="1.0" encoding="UTF-8"?>
<doc>
  <customer id='123'>
    <name>...</name>
    <address>...</address>
    ...
  <order>
    <amount>...</amount>
    <date>...</date>
    <item quant='12'>
      <name>...</name>
    </item>
    <item quant='4'>...</item>
    ...
  </order>
  <order>...</order>
  ...
  <payment>
    <number>...</number>
    <date>...</date>
  </payment>
  <payment>...</payment>
  ...
</customer>
<customer id='124'>...</customer>
</doc>
```

図 28. サンプル XML ファイル

非ルート・ニックネームの CREATE VIEW ステートメント

以下の例は、ルートではないニックネーム order のフェデレーテッド・ビューを作成する方法を示しています。

```
CREATE VIEW order_view AS
  SELECT o.amount, o.date, o.oid, c.cid
  FROM customers c, orders o
  WHERE c.cid = o.cid;
```

以下の例は、ルートではないニックネーム `payment` のフェデレーテッド・ビューを作成する方法を示しています。

```
CREATE VIEW payment_view AS
  SELECT p.number, p.date, c.cid
  FROM customers c, payments p
  WHERE c.cid = p.cid;
```

以下の例は、ルートではないニックネーム `item` のフェデレーテッド・ビューを作成する方法を示しています。

```
CREATE VIEW item_view AS
  SELECT i.quantity, i.name, o.oid
  FROM customers c, orders o, items i
  WHERE c.cid = o.cid AND o.oid = i.oid;
```

フェデレーテッド・ビューを使用する照会

ルート・ディレクトリーへの結合パスが `WHERE` 文節に指定されているので、フェデレーテッド・ビューにサブミットされる照会は正しく処理されます。

例えば、次の照会は、カスタマー識別番号と発注日を使用して、注文額と未払いの支払額を戻します。この照会ではニックネームを使用する代わりに、`FROM` 文節でビューを指定します。

```
SELECT o.amount, p.amount
FROM order_view o, payment_view p
WHERE p.date = o.date AND
      p.cid = o.cid;
```

XML コスト・モデル機能の照会の最適化のヒント

XML ラッパーのコスト・モデル機能は、作成するニックネームに対する照会を最適化するのに役立ちます。

コスト・モデル機能では、`CREATE NICKNAME` ステートメントの次のニックネーム・オプションを使用します。

- `INSTANCE_PARSE_TIME`
- `XPATH_EVAL_TIME`

これらのニックネーム・オプションの値は、`CREATE NICKNAME` ステートメントを発行して XML データ・ソースのニックネームを登録する際に指定できます。

コスト・モデル機能は、XML ソース文書の各行データを解析するのに必要な時間を判別するときにそれらのパラメーター値を使用します。パラメーター値は、ニックネーム用 XPath 式を評価するためにも使用されます。

これらのニックネーム・オプションには、デフォルト値を使用できます。しかし、作成するニックネームの大規模なまたは複雑な XML ソース構造に関する照会を最適化する場合には、次の例に示されている方針に従ってください。

大規模照会を最適化する例

XML 文書には、次の 4 つのニックネームを含むリレーショナル・スキーマが含まれています。

- `customers`

- orders
- payments
- items

ルート・ニックネームは customers です。

各ニックネームに対して照会を実行します。実際の環境でよくある XML データのサンプルに対して、各照会を実行します。

例えば、次のようにします。

```
SELECT * from customers;
SELECT * from orders;
SELECT * from payments;
SELECT * from items;
```

db2batch コマンドか、それに相当するコマンドやユーティリティーを使用することによって、各照会の実行に必要な時間 (ミリ秒) を記録します。 **db2batch** コマンドを使うと、照会実行に必要な時間を内容として含む出力ファイルが生成されます。戻されるタプルの数も記録します。

各ニックネームごとに、次の式により **INSTANCE_PARSE_TIME** ニックネーム・オプションと **XPATH_EVAL_TIME** ニックネーム・オプションに最適な値を決定します。

$$\text{INSTANCE_PARSE_TIME} = (75\% \times [\text{SELECT * query の実行時間}]) \div [\text{戻されるタプルの数}]$$

$$\text{XPATH_EVAL_TIME} = (25\% \times [\text{SELECT * query の実行時間}]) \div [\text{戻されるタプルの数}]$$

ルート・ニックネーム (この例では customers です) については、**INSTANCE_PARSE_TIME** ニックネーム・オプションと **XPATH_EVAL_TIME** ニックネーム・オプションに対して、上記の式で計算される値を使用します。

非ルート・ニックネーム (この例では orders、payments、および items です) については、**XPATH_EVAL_TIME** パラメーターについてのみ、上記の式で計算される値を使用します。 **INSTANCE_PARSE_TIME** パラメーターの値は、非ルート・ニックネームには適用されません。

これらの式は、照会を調整するための 1 つの方針として使用できます。これらのニックネーム・オプションに最適な値は、XML ソース文書の複雑さや、使用しているプロセッサの速度に応じて異なります。

XML データ・ソース - 照会の例

XML ニックネームを使用した照会の例

これらの例では customers、orders、items のニックネームを使用します。

XML 文書から特定の値を戻す照会

次の SELECT ステートメントを実行すると、ラッパーはすべての顧客名を戻します。

```
SELECT name FROM customers;
```

特定顧客のすべてのレコードを戻す照会

次の SELECT ステートメントを実行すると、ラッパーは顧客名が Chang のすべてのレコードを戻します。

```
SELECT * FROM customers
  WHERE name='Chang';
```

親ニックネームと子ニックネーム間の結合条件を基にした特定の値を戻す照会

次の SELECT ステートメントを実行すると、ラッパーは、各顧客のそれぞれの発注に対して発注額と顧客名を戻します。customers ニックネームおよび orders ニックネームの間の親子関係を示すには、結合 c.cid=o.cid を指定しなければなりません。

```
SELECT c.name, o.amount FROM customers c, orders o
  WHERE c.cid=o.cid;
```

1 つの親ニックネームと複数の子ニックネーム間の結合条件を指定する方法を示す照会

次の SELECT ステートメントを実行すると、ラッパーは、各顧客のそれぞれの発注とアイテムごとに、顧客のアドレス、発注額、アイテム名を戻します。親子関係を維持するには、これら 2 つの結合を指定する必要があります。

```
SELECT c.address, o.amount, i.name FROM customers c, orders o, items i
  WHERE c.cid=o.cid AND o.oid=i.oid;
```

照会内で XML 文書を指定する方法を示す照会

以下の例は、FILE_PATH ニックネーム・オプションは指定しないが、DOCUMENT ニックネーム列オプションは指定するニックネームを使用して、照会を作成する方法を示しています。

customers ニックネームを作成するために使用される CREATE NICKNAME ステートメントは、以下のとおりです。

```
CREATE NICKNAME customers
(
  doc      VARCHAR(100)  OPTIONS(DOCUMENT 'FILE'),
  name     VARCHAR(16)   OPTIONS(XPATH './name'),
  address  VARCHAR(30)   OPTIONS(XPATH './address/@street'),
  cid      VARCHAR(16)   OPTIONS(PRIMARY_KEY 'YES')
  FOR SERVER xml_server
  OPTIONS(XPATH '/doc/customer');
```

以下の照会は、/home/db2user/Customers.xml のパスをもつ XML ファイル Customers.xml からすべてのデータを選択します。

```
SELECT * FROM customers
  WHERE doc='/home/db2user/Customers.xml';
```

以下の照会は、Customers.xml ファイルから、1000 の金額を超えるオーダーについて、カスタマーの名前とオーダーの日付を選択します。パス /home/db2user/Customers.xml は、Customers.xml ファイルのロケーションを指定します。

```
SELECT c.name, o.date FROM customers c, orders o
  WHERE c.doc='/home/db2user/Customers.xml' AND o.amount > 1000;
```


第 3 章 フェデレーテッド・フィーチャーのデータ・ソース・サポート

データ・ソースが特定のフェデレーテッド・フィーチャーをサポートするかどうかを知るためには、この表を参照してください。

これらのフィーチャーを使用する前に、特定のラッパーまたはサーバー・オプションを設定するか、あるいは他の操作を行って、その機能を使用可能にする必要がある場合があります。詳しくは、各フィーチャーの特定のトピックを参照してください。

表 45. フィーチャーおよびサポートされるデータ・ソース

フィーチャー	データ・ソース
ニックネームに対して WRITE 操作を指定したアプリケーション・セーブポイント	DB2 for Linux, UNIX, and Windows
非同期最適化	すべてのデータ・ソース
キャッシュ表	DB2 ファミリー Informix Microsoft SQL Server Oracle Sybase
ニックネームへのデータのインポート	DB2 ファミリー Informix Microsoft SQL Server Oracle Sybase Teradata
ネストされた表の式でのエラー許容度	DB2 ファミリー Informix JDBC Microsoft SQL Server ODBC Oracle Sybase Teradata
外部ユーザーのマッピング・リポジトリ	すべてのデータ・ソース

表 45. フィーチャーおよびサポートされるデータ・ソース (続き)

フィーチャー	データ・ソース
フェデレーテッド・ヘルス・インディケーター	DB2 ファミリー Excel Informix JDBC Microsoft SQL Server ODBC Oracle Sybase 表構造ファイル Teradata XML (ルート・ニックネームのみ)
フェデレーテッド・プロシージャ	DB2 ファミリー (トラステッド・モード) Oracle (トラステッド・モード) Microsoft SQL Server (トラステッド・モード) Sybase (fenced モード、UNIX にフェデレーテッド・サーバーをインストール済み) Sybase (fenced モードまたはトラステッド・モード、 Linux または Microsoft Windows にフェデレーテッド・サーバーをインストール済み)
フェデレーテッド・トラステッド・コンテキスト	DB2 for Linux, UNIX, and Windows バージョン 9.5 DB2 for z/OS バージョン 9 Oracle
HTTP プロキシ	Web サービス XML
接続レベルの分離	DB2 ファミリー Informix JDBC Microsoft SQL Server ODBC Oracle Sybase
ラベル・ペースのアクセス制御	DB2 for Linux, UNIX, and Windows バージョン 9.1 および 9.5 Oracle
LOB 読み取りおよび書き込み操作	DB2 for z/OS DB2 for Linux, UNIX, and Windows DB2 for System i® Oracle

表 45. フィーチャーおよびサポートされるデータ・ソース (続き)

フィーチャー	データ・ソース
LOB 読み取り専用操作	BioRS Informix JDBC Microsoft SQL Server ODBC スクリプト Sybase Teradata Web サービス XML
マテリアライズ照会表	すべてのデータ・ソース (固有の制約事項あり)
ニックネーム統計の更新機能	BioRS DB2 ファミリー Excel Informix JDBC Microsoft SQL Server ODBC Oracle Sybase 表構造ファイル Teradata XML (ルート・ニックネームのみ)
パススルー・セッション	DRDA Informix Oracle Microsoft SQL Server Sybase Teradata
リモート XML データ・タイプ	DB2 for Linux, UNIX, and Windows XML ラッパー
SOCKS プロキシ	BioRS スクリプト Web サービス XML
Secure Socket Layer (SSL)	Web サービス XML
ステートメント・レベルの分離	DB2 ファミリー Microsoft SQL Server

表 45. フィーチャーおよびサポートされるデータ・ソース (続き)

フィーチャー	データ・ソース
2 フェーズ・コミット・トランザクション	DB2 for Linux, UNIX, and Windows (トラステッド・モード) DB2 for System i (トラステッド・モード) DB2 for z/OS (トラステッド・モード) Informix (トラステッド・モード) Microsoft SQL Server (トラステッド・モード、 Microsoft Windows にフェデレーテッド・サーバーをインストール済み) Oracle (トラステッド・モード) Sybase (トラステッド・モード、 Microsoft Windows にフェデレーテッド・サーバーをインストール済み) Sybase (fenced モード、UNIX にフェデレーテッド・サーバーをインストール済み)
Unicode サポート	すべてのデータ・ソース

第 4 章 データ・ソース・オプション・リファレンス

各データ・ソースは、特定のラッパー、サーバー、ユーザー・マッピング、ニックネーム、および列のオプションをサポートします。

BioRS オプション・リファレンス

フェデレーテッド・サーバーおよびそのユーザーがデータ・ソースと対話する方法を構成するには、ラッパー、サーバー、ユーザー・マッピング、ニックネーム、および列のオプションを設定および変更します。

ラッパー・オプション

以下の表に、このデータ・ソースに適用されるオプションと、CREATE WRAPPER ステートメントおよび CREATE SERVER ステートメントに指定する必要がある必須指定のオプションをまとめました。

表 46. BioRS のラッパー・オプション

名前	説明
DB2_FENCED	必須。ラッパーを fenced モードとトラステッド・モードのどちらで実行するかを指定します。有効な値は Y と N です。デフォルトは N であり、その場合、ラッパーはトラステッド・モードで実行されます。
DB2_UM_PLUGIN	ユーザー・マッピング・プラグインのインプリメンテーションを指定します。Java で作成されたプラグインの場合、ユーザー・マッピング・リポジトリ・クラスに対応するクラス名として、大/小文字の区別があるストリングを指定します。例えば、 「UserMappingRepositoryLDAP」のように指定します。C で作成されたプラグインの場合、任意の有効な C ライブラリー名を指定します。
DB2_UM_PLUGIN_LANG	ユーザー・マッピング・プラグインの言語を指定します。有効な値は、Java および C です。デフォルトは Java です。
PROXY_SERVER_NAME	プロキシ・サーバーの名前または IP アドレスを指定します。PROXY_TYPE の値が HTTP または SOCKS の場合、このオプションは必須です。有効な IP アドレスは IPv4 (ドット区切り) 形式または IPv6 (コロン区切り) 形式です。IPv6 形式は、IPv6 を構成する場合のみ使用します。

表 46. BioRS のラッパー・オプション (続き)

名前	説明
PROXY_SERVER_PORT	プロキシ・サーバー上のプロキシ・サービスのポートまたはサービス名を指定します。PROXY_TYPE の値が HTTP または SOCKS の場合、このオプションは必須です。有効な値は、10 進数の 1 から 32760 までのポート番号か、サービス名です。
PROXY_TYPE	フェデレーテッド・サーバーがファイアウォールの背後にある場合に、インターネットへのアクセスに使用するプロキシ・タイプを指定します。有効な値は NONE、HTTP、および SOCKS です。デフォルト値は NONE です。このオプションを HTTP または SOCKS に設定した場合は、PROXY_SERVER_NAME と PROXY_SERVER_PORT も指定する必要があります。

サーバー・オプション

表 47. BioRS のサーバー・オプション

名前	説明
CASE_SENSITIVE	BioRS サーバーが大/小文字の区別をして名前を扱うかどうかを指定します。有効な値は Y と N です。デフォルトは N です。この場合、名前は大/小文字の区別をして扱われません。BioRS 製品では、BioRS サーバーに保管されているデータの大/小文字の区別を構成パラメーターで制御します。大/小文字の区別の指定は、CASE_SENSITIVE オプションと構成パラメーターとで同一でなければなりません。サーバー定義を作成した後で CASE_SENSITIVE オプションの値の変更が必要になった場合は、サーバー定義をドロップして、その定義とすべてのニックネームを再度作成する必要があります。
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	照会からの並行非同期要求の最大数を指定します。有効な値は -1 から 64000 です。デフォルトは 1 です。-1 を指定すると、要求の数はフェデレーテッド照会オプティマイザーが決定します。0 は、データ・ソースが、追加の非同期要求に対応できないということを指定します。

表 47. BioRS のサーバー・オプション (続き)

名前	説明
DB2_UM_PLUGIN	ユーザー・マッピング・プラグインのインプリメンテーションを指定します。Java で作成されたプラグインの場合、ユーザー・マッピング・リポジトリ・クラスに対応するクラス名として、大/小文字の区別があるストリングを指定します。例えば、 「UserMappingRepositoryLDAP」のように指定します。C で作成されたプラグインの場合、任意の有効な C ライブラリー名を指定します。
DB2_UM_PLUGIN_LANG	ユーザー・マッピング・プラグインの言語を指定します。有効な値は Java および C です。デフォルトは Java です。
NODE	必須。BioRS 照会ツールが使用可能なシステムの DNS ホスト名または IP アドレスを指定します。有効な IP アドレスは IPv4 (ドット区切り) 形式または IPv6 (コロン区切り) 形式です。IPv6 形式は、IPv6 を構成する場合のみ使用します。デフォルト値は localhost です。
PORT	BioRS サーバーに接続するポートを指定します。有効な値は、数値ポートまたは TCP/IP サービス名です。デフォルトは 5014 です。
PROXY_AUTHID	プロキシ・サーバー認証のユーザー名を指定します。
PROXY_PASSWORD	プロキシ・サーバー認証のパスワードを指定します。
PROXY_SERVER_NAME	プロキシ・サーバーの名前または IP アドレスを指定します。有効な値は、1 から 32760 までの 10 進数のポート番号、またはサービス名です。IPv6 形式は、IPv6 を構成する場合のみ使用します。
PROXY_SERVER_PORT	プロキシ・サーバー上のプロキシ・サービスのポートまたはサービス名を指定します。有効な値は、10 進数の 1 から 32760 までのポート番号か、サービス名です。
PROXY_TYPE	フェデレーテッド・サーバーがファイアウォールの背後にある場合に、インターネットへのアクセスに使用するプロキシ・タイプを指定します。有効な値は NONE、HTTP、および SOCKS です。デフォルト値は NONE です。
TIMEOUT	リモート・サーバーからの応答をフェデレーテッド・サーバーが待つ最大時間 (分単位) を指定します。デフォルトは 10 です。

ユーザー・マッピング・オプション

表 48. BioRS のユーザー・マッピング・オプション

オプション	説明
GUEST	操作実行のために GUEST BioRS 認証 ID を使用することを指定します。有効な値は Y と N です。デフォルトは N です。この場合、GUEST BioRS ID は使用されません。REMOTE_AUTHID オプションおよび REMOTE_PASSWORD オプションを指定した場合、このオプションは無効です。
PROXY_AUTHID	プロキシ・サーバー認証のユーザー名を指定します。
PROXY_PASSWORD	プロキシ・サーバー認証のパスワードを指定します。このパスワードは、フェデレーテッド・データベース・カタログに保管されるときに暗号化されます。
REMOTE_AUTHID	ローカル・ユーザー ID のマップ先となるリモート・ユーザー ID を指定します。このオプションを指定しない場合は、フェデレーテッド・データベースへの接続に使用する ID が使用されます。
REMOTE_PASSWORD	リモート・ユーザー ID のリモート・パスワードを指定します。このオプションを指定しない場合は、フェデレーテッド・データベースへの接続に使用したパスワードが使用されます。

ニックネーム・オプション

表 49. BioRS のニックネーム・オプション

オプション	説明
REMOTE_OBJECT	ニックネームに関連する BioRS データ・バンクの名前を指定します。この名前によって、ニックネームのスキーマと BioRS データ・バンクが決まります。この名前によって、そのニックネームと他のニックネームの関係も指定されます。このオプションの大/小文字の区別をするかどうかは、BioRS サーバーで大/小文字の区別をするかどうか、および CASE_SENSITIVE サーバー・オプションの値によって決まります。ALTER NICKNAME ステートメントを使用して、この名前の変更または削除を行うことはできません。BioRS データ・バンクの名前を変更した場合、ニックネームを削除してから、再びそれを作成しなければなりません。

表 49. BioRS のニックネーム・オプション (続き)

オプション	説明
TIMEOUT	データ・ソース・サーバーからの応答を待つ最大時間 (分単位) を指定します。デフォルトは 10 です。

列オプション

表 50. BioRS の列オプション

オプション	説明
ELEMENT_NAME	BioRS エlement名を指定します。この名前の大/小文字の区別をどうかは、BioRS サーバーで大/小文字の区別をどうか、および CASE_SENSITIVE サーバー・オプションの値によって決まります。BioRS エlement名の指定が必須なのは、Element名が列名と異なる場合だけです。
IS_INDEXED	対応する列が索引付けされ、述部で列が参照可能かどうかを指定します。有効な値は Y と N です。デフォルトは N です。この場合、列は索引付けされません。
REFERENCED_OBJECT	現在の列が参照する BioRS データ・バンクの名前を指定します。この名前の大/小文字の区別をどうかは、BioRS サーバーで大/小文字の区別をどうか、および CASE_SENSITIVE サーバー・オプションの値によって決まります。このオプションは、BioRS データ・タイプが Reference である列にのみ有効です。

DB2 データベース・オプション・リファレンス

フェデレーテッド・サーバーおよびそのユーザーがデータ・ソースと対話する方法を構成するには、ラッパー、サーバー、ユーザー・マッピング、ニックネーム、および列のオプションを設定および変更します。

ラッパー・オプション

次の表は、DB2 データ・ソースに適用されるオプションをリストし、指定する必要がある必須選択のオプションを示しています。

表 51. DB2 データ・ソースのラッパー・オプション

名前	説明
DB2_FENCED	必須。ラッパーを fenced モードとトラステッド・モードのどちらで実行するかを指定します。有効な値は Y と N です。デフォルトは N であり、その場合、ラッパーはトラステッド・モードで実行されます。

表 51. DB2 データ・ソースのラッパー・オプション (続き)

名前	説明
DB2_UM_PLUGIN	ユーザー・マッピング・プラグインのインプリメンテーションを指定します。Java で作成されたプラグインの場合、ユーザー・マッピング・リポジトリ・クラスに対応するクラス名として、大/小文字の区別があるストリングを指定します。例えば、 「UserMappingRepositoryLDAP」のように指定します。C で作成されたプラグインの場合、任意の有効な C ライブラリー名を指定します。
DB2_UM_PLUGIN_LANG	ユーザー・マッピング・プラグインの言語を指定します。有効な値は Java および C です。デフォルトは Java です。

サーバー・オプション

表 52. DB2 データ・ソースのサーバー・オプション

名前	説明
COLLATING_SEQUENCE	データ・ソースがフェデレーテッド・データベースと同じデフォルト照合シーケンスを使用するかどうかを指定します。有効な値は Y、N、および I です。I は大/小文字を区別しないことを指定します。デフォルト値は Y です。フェデレーテッド・サーバーに指定された照合シーケンスは、リモート・データ・ソースの照合シーケンスと一致している必要があります。
COMM_RATE	フェデレーテッド・サーバーとデータ・ソース・サーバーとの間の通信速度を指定します (MB/秒)。有効な値は 0 より大きく、2147483648 より小さい整数です。デフォルトは 2 です。
CPU_RATIO	データ・ソースの CPU がフェデレーテッド・サーバーの CPU と比較してどれほど速いかまたは遅いかを指定します。有効な値は 0 より大きく、 1×10^{23} より小さい値です。デフォルトは 1.0 です。値は、任意の有効な倍精度表記 (例えば 123E10、123、または 1.21E4) で表現できます。1 を設定すると、フェデレーテッド・サーバーとデータ・ソース・サーバーの CPU 速度が同じで、1:1 の比率であるということです。0.5 を設定すると、フェデレーテッド・サーバーの CPU の速度がデータ・ソースの CPU の半分であることを示します。2 を設定すると、フェデレーテッド CPU がデータ・ソース CPU の 2 倍高速であるということです。

表 52. DB2 データ・ソースのサーバー・オプション (続き)

名前	説明
DATE_COMPAT	データベースに date_compat パラメーターを適用するかどうかを指定します。有効な値は Y および N です。デフォルトは N です。このサーバー・オプションは、DB2 Database for Linux, UNIX, and Windows バージョン 9.7 以降でのみ有効です。
DBNAME	必須。初期のリモート DB2 データベース接続で使用する特定のデータベースを指定します。この特定のデータベースとは、CATALOG DATABASE コマンドまたは DB2 構成アシスタントを使用して、フェデレーテッド・サーバーでカタログされた、リモート DB2 データベースのデータベース別名です。
DB2_MAXIMAL_PUSHDOWN	照会オプティマイザーがアクセス・プランの選択に使用する 1 次基準を指定します。有効な値は Y および N です。デフォルトは N であり、その場合、照会オプティマイザーは見積コストが最小になるプランを選択します。Y は、照会オプティマイザーが、照会操作を最も多くデータ・ソースにプッシュダウンするアクセス・プランを選択するということを指定します。
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	照会からの並行非同期要求の最大数を指定します。有効な値は -1 から 64000 です。デフォルトは 1 です。-1 を指定すると、要求の数はフェデレーテッド照会オプティマイザーが決定します。0 を指定すると、データ・ソースは付加的な非同期要求を受け入れることができません。
DB2_TWO_PHASE_COMMIT	フェデレーテッド・サーバーがデータ・ソースに 2 フェーズ・コミット・プロトコルと 1 フェーズ・コミット・プロトコルのどちらで接続するかを指定します。有効な値は Y と N です。デフォルトは N であり、その場合、フェデレーテッド・サーバーは 1 フェーズ・コミット・プロトコルを使用して接続します。Y を指定すると、フェデレーテッド・サーバーは 2 フェーズ・コミット・プロトコルを使用して接続します。

表 52. DB2 データ・ソースのサーバー・オプション (続き)

名前	説明
DB2_UM_PLUGIN	<p>ユーザー・マッピング・プラグインのインプリメンテーションを指定します。Java で作成されたプラグインの場合、ユーザー・マッピング・リポジトリ・クラスに対応するクラス名として、大/小文字の区別があるストリングを指定します。例えば、</p> <p>「UserMappingRepositoryLDAP」のように指定します。C で作成されたプラグインの場合、任意の有効な C ライブラリー名を指定します。</p>
DB2_UM_PLUGIN_LANG	<p>ユーザー・マッピング・プラグインの言語を指定します。有効な値は Java および C です。デフォルトは Java です。</p>
FED_PROXY_USER	<p>インバウンド接続が非トラステッド接続である場合に、すべてのアウトバウンド・トラステッド接続を確立するために使用する許可 ID を指定します。このオプションで ID が指定されているユーザーのユーザー・マッピングには、REMOTE_AUTHID と REMOTE_PASSWORD の両方のオプションが指定されている必要があります。</p> <p>制約事項: このサーバー・オプションは、DB2 Database for Linux, UNIX, and Windows バージョン 9.5 以降、および DB2 for z/OS バージョン 9 以降でのみ有効です。</p>
FOLD_ID	<p>データ・ソースに送信されるユーザー ID で大/小文字のどちらを使用するかを指定します。デフォルト値はありません。フェデレーテッド・サーバーは、ユーザー ID を大文字で送信し、この ID が失敗した場合は、ユーザー ID を小文字で送信します。有効な値は U (大文字)、L (小文字)、および N (NULL) です。NULL に設定すると、パフォーマンスが低下する可能性があるため、NULL 値の使用は避けてください。</p>
FOLD_PW	<p>データ・ソースに送信されるパスワードで大/小文字のどちらを使用するかを指定します。デフォルト値はありません。フェデレーテッド・サーバーは、パスワードを大文字で送信し、このパスワードが失敗した場合は、パスワードを小文字で送信します。有効な値は U (大文字)、L (小文字)、および N (NULL) です。NULL に設定すると、パフォーマンスが低下する可能性があるため、NULL 値の使用は避けてください。</p>

表 52. DB2 データ・ソースのサーバー・オプション (続き)

名前	説明
IO_RATIO	<p>データ・ソース入出力システムがフェデレーテッド・サーバー入出力システムと比較して、どれほど速いかまたは遅いかを指定します。有効な値は 0 より大きく、1×10^{23} より小さい値です。デフォルトは 1.0 です。値は、任意の有効な倍精度表記 (例えば 123E10、123、または 1.21E4) で表現できます。1 を設定すると、フェデレーテッド・サーバーとデータ・ソース・サーバーの入出力速度が同じで、1:1 の比率であるということです。0.5 を設定すると、フェデレーテッド・サーバーの速度がデータ・ソースの速度の半分であることを示します。2 を設定すると、フェデレーテッドの速度がデータ・ソースの速度の 2 倍高速であるということです。</p>
NO_EMPTY_STRING	<p>リモート・データ・ソース・サーバーに、空ストリングを格納できるかどうかを指定します。有効な値は Y と N です。デフォルト値は、ご使用のリモート・データ・ソースによって異なります。リモート Oracle データ・ソースの場合、デフォルトは 1 であり、すべての空ストリング値は NULL 値に変換されます。その他すべてのリモート・データ・ソースの場合、デフォルトは N であり、データ・ソースに空ストリングを格納できます。</p> <p>フェデレーテッド・サーバーが VARCHAR2 互換モードであるが、リモート・データ・ソースが VARCHAR2 互換ではない場合、システム構成でこのオプションを Y に設定すると、システムのパフォーマンスを改善できます。</p>

表 52. DB2 データ・ソースのサーバー・オプション (続き)

名前	説明
NUMBER_COMPAT	<p>データ・ソース・サーバーで NUMBER データ・タイプをサポートするかどうかを指定します。有効な値は Y と N です。デフォルトは N であり、その場合、データ・ソース・サーバーは NUMBER データ・タイプをサポートしません。NUMBER データ・タイプがフェデレーテッド・サーバーではサポートされず、データ・ソース・サーバーではサポートされるシステムの場合、NUMBER_COMPAT オプションを Y に設定する必要があります。これは、データ・ソース・サーバーが DECIMAL データ・タイプの範囲外である DECFLOAT 結果を返すことがあり、SQLSTATE 560BD エラーが発生する可能性があるためです。</p> <p>制約事項: このサーバー・オプションは、DB2 Database for Linux, UNIX, and Windows バージョン 9.7 以降でのみ有効です。</p>
OLD_NAME_GEN	<p>データ・ソースの列名と索引名を、フェデレーテッド・サーバーのニックネーム列名とローカル索引名に変換する方法を指定します。有効な値は Y と N です。デフォルトは N であり、その場合、生成される名前とデータ・ソース中の名前には密接な対応関係があります。Y を指定すると、生成される名前は IBM WebSphere® Federation Server バージョン 9 以前で作成された名前と同じになります。したがって、それらの名前はデータ・ソースの名前と密接に対応しないことがあります。</p>
PUSHDOWN	<p>フェデレーテッド・サーバーが、データ・ソースによる操作の評価を許可するかどうかを指定します。有効な値は Y および N です。デフォルトは Y であり、その場合、データ・ソースは操作を評価します。N は、列名を指定した SELECT のみを含む SQL ステートメントをフェデレーテッド・サーバーが送信するということを指定します。述部 (WHERE= など)、列およびスカラー関数 (MAX や MIN など)、ソート (ORDER BY または GROUP BY など)、および結合は、フェデレーテッド・サーバーがデータ・ソースに送信するどの SQL にも含まれません。</p>

表 52. DB2 データ・ソースのサーバー・オプション (続き)

名前	説明
SAME_DECFLT_ROUNDING	<p>フェデレーテッド・サーバーとデータ・ソース・サーバーの丸めモードが、いずれも同じ DECFLOAT 丸めモード設定値を使用するかどうかを指定します。有効な値は Y と N です。デフォルトは N であり、その場合、フェデレーテッド・サーバーとリモート・サーバーの DECFLOAT 丸めモード設定値は異なります。</p> <p>重要: フェデレーテッド・サーバーとデータ・ソース・サーバーで丸めモードが異なる場合に、このオプションを Y に設定すると、間違った DECFLOAT 丸め結果を受け取ることがあります。</p> <p>同じ DECFLOAT 丸めモード設定値を使用する既存のフェデレーテッド・サーバーおよびデータ・ソース・サーバーを構成する場合は、ALTER SERVER ステートメントを使用します。</p> <p>制約事項: このサーバー・オプションは、DB2 Database for Linux, UNIX, and Windows バージョン 9.5 以降でのみ有効です。</p>
VARCHAR2_COMPAT	<p>リモート・データ・ソースが VARCHAR2 互換であるかどうかを指定します。有効な値は Y と N です。デフォルト値は、リモート・データ・ソースによって異なります。リモート Oracle データ・ソースの場合、デフォルトは Y であり、データ・ソースは VARCHAR2 互換です。その他すべてのリモート・データ・ソースの場合、デフォルトは N であり、データ・ソースは VARCHAR2 互換モードに設定されません。</p> <p>DB2 Database for Linux, UNIX, and Windows、ODBC、または JDBC データ・ソースが VARCHAR2 互換モードで構成されている場合、このサーバー・オプションを Y に設定する必要があります。</p>

ユーザー・マッピング・オプション

表 53. DB2 データ・ソースのユーザー・マッピング・オプション

オプション	説明
FED_PROXY_USER	<p>インバウンド接続が非トラステッド接続である場合に、すべてのアウトバウンド・トラステッド接続を確立するために使用する許可 ID を指定します。このオプションで指定された ID を持つユーザーには、REMOTE_AUTHID と REMOTE_PASSWORD の両方を指定するユーザー・マッピングが必要です。</p> <p>FED_PROXY_USER ユーザー・マッピング・オプションを指定する場合は、FED_PROXY_USER サーバー・オプションも指定する必要があります。</p> <p>制約事項: このサーバー・オプションは、DB2 Database for Linux, UNIX, and Windows バージョン 9.5 以降、および DB2 for z/OS バージョン 9 以降でのみ有効です。</p>
ACCOUNTING_STRING	<p>アカウントリング情報を受け渡す必要がある場合は必須。DRDA 会計情報ストリングを指定します。有効な値は、255 文字以下のストリングです。</p>
REMOTE_AUTHID	<p>ローカル・ユーザー ID のマップ先となるリモート・ユーザー ID を指定します。このオプションを指定しない場合は、フェデレーテッド・データベースへの接続に使用する ID が使用されます。</p>
REMOTE_PASSWORD	<p>リモート・ユーザー ID のリモート・パスワードを指定します。このオプションを指定しない場合は、フェデレーテッド・データベースへの接続に使用したパスワードが使用されます。</p>
USE_TRUSTED_CONTEXT	<p>ユーザー・マッピングがトラステッドかどうかを指定します。有効な値は Y と N です。デフォルトは N であり、その場合ユーザー・マッピングはトラステッドとはならず、非トラステッドのフェデレーテッド・アウトバウンド接続でのみ使用できます。Y が指定されている場合、ユーザー・マッピングはトラステッドであり、トラステッドと非トラステッドのどちらのアウトバウンド・フェデレーテッド接続でも使用できます。</p> <p>制約事項: このサーバー・オプションは、DB2 Database for Linux, UNIX, and Windows バージョン 9.5 以降、および DB2 for z/OS バージョン 9 以降でのみ有効です。</p>

列オプション

表 54. DB2 データ・ソースの列オプション

オプション	説明
NUMERIC_STRING	数値ストリングの処理方法を指定します。デフォルトは N です。データ・ソース・ストリング列に数値ストリングのみが含まれ、ブランクを含め他の文字が含まれていない場合、NUMERIC_STRING オプションを Y に設定します。列に対して NUMERIC_STRING が Y に設定されている場合、照会オペティマイザーは、列のデータをソートする場合に支障となり得るブランクがこの列には含まれないことを認識します。データ・ソースの照合シーケンスが、フェデレーテッド・サーバーが使用する照合シーケンスとは異なる場合に、このオプションを使用してください。このオプションを使用する列は、照合シーケンスが異なるためにリモートでの評価から除かれるということはありません。
NO_EMPTY_STRING	リモート・データ・ソース・サーバーに、空ストリングを格納できるかどうかを指定します。有効な値は Y と N です。デフォルト値は、ご使用のリモート・データ・ソースによって異なります。リモート Oracle データ・ソースの場合、デフォルトは 1 であり、すべての空ストリング値は NULL 値に変換されます。その他すべてのリモート・データ・ソースの場合、デフォルトは N であり、データ・ソースに空ストリングを格納できます。
XML_ROOT	XML シーケンスを参照する XML 列の値に追加する XML ルート要素を指定します。このオプションを指定した場合、XML 列の値が整形形式 XML 文書であることが保証されます。

Excel オプション・リファレンス

フェデレーテッド・サーバーとそのユーザーがどのようにデータ・ソースと対話するかを構成するには、ラッパー、サーバー、およびニックネームの各オプションを設定および変更します。

ラッパー・オプション

次の表は、このデータ・ソースに適用されるオプションをリストし、指定する必要がある必須選択のオプションを示しています。

表 55. Excel のラッパー・オプション

名前	説明
DB2_FENCED	必須。ラッパーを fenced モードとトラステッド・モードのどちらで実行するかを指定します。有効な値は Y と N です。デフォルトは N であり、その場合、ラッパーはトラステッド・モードで実行されます。

サーバー・オプション

表 56. Excel のサーバー・オプション

名前	説明
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	照会からの並行非同期要求の最大数を指定します。有効な値は -1 から 64000 です。デフォルトは 1 です。-1 を指定すると、要求の数はフェデレーテッド照会オプティマイザーが決定します。0 を指定すると、データ・ソースは付加的な非同期要求を受け入れることができません。

ニックネーム・オプション

表 57. Excel のニックネーム・オプション

オプション	説明
FILE_PATH	必須。アクセスする Excel スプレッドシートの完全修飾ディレクトリー・パスおよびファイル名を指定します。
RANGE	使用するセルの範囲を指定します (例: A1:C100)。コロンの前の値は、範囲の左上隅のセルを指定します。コロンの後の値は、範囲の右下隅のセルを指定します。

Informix オプション・リファレンス

フェデレーテッド・サーバーおよびそのユーザーがデータ・ソースと対話する方法を構成するには、ラッパー、サーバー、ユーザー・マッピング、および列のオプションを設定および変更します。

ラッパー・オプション

次の表は、このデータ・ソースに適用されるオプションをリストし、指定する必要がある必須選択のオプションを示しています。

表 58. Informix のラッパー・オプション

名前	説明
DB2_FENCED	必須。ラッパーを fenced モードとトラステッド・モードのどちらで実行するかを指定します。有効な値は Y と N です。デフォルトは N であり、その場合、ラッパーはトラステッド・モードで実行されます。
DB2_UM_PLUGIN	ユーザー・マッピング・プラグインのインプリメンテーションを指定します。Java で作成されたプラグインの場合、ユーザー・マッピング・リポジトリ・クラスに対応するクラス名として、大/小文字の区別があるストリングを指定します。例えば、 「UserMappingRepositoryLDAP」のように指定します。C で作成されたプラグインの場合、任意の有効な C ライブラリー名を指定します。
DB2_UM_PLUGIN_LANG	ユーザー・マッピング・プラグインの言語を指定します。有効な値は、Java および C です。デフォルトは Java です。

サーバー・オプション

表 59. Informix のサーバー・オプション

名前	説明
COLLATING_SEQUENCE	データ・ソースがフェデレーテッド・データベースと同じデフォルト照合シーケンスを使用するかどうかを指定します。有効な値は Y、N、および I です。I を指定すると、大/小文字を区別しません。デフォルトは Y です。フェデレーテッド・サーバーに指定された照合シーケンスは、リモート・データ・ソースの照合シーケンスと一致している必要があります。
COMM_RATE	フェデレーテッド・サーバーとデータ・ソース・サーバーとの間の通信速度を指定します (MB/秒)。有効な値は 0 より大きく、2147483648 より小さい整数です。デフォルトは 2 です。

表 59. Informix のサーバー・オプション (続き)

名前	説明
CPU_RATIO	データ・ソースの CPU がフェデレーテッド・サーバーの CPU と比較して、どれほど速いかまたは遅いかを指定します。有効な値は 0 より大きく、 1×10^{23} より小さい値です。デフォルトは 1.0 です。値は、任意の有効な倍精度表記 (例えば 123E10、123、または 1.21E4) で表現できます。1 を設定すると、フェデレーテッド・サーバーとデータ・ソース・サーバーの CPU 速度が同じで、1:1 の比率であるということです。0.5 を設定すると、フェデレーテッド・サーバーの CPU の速度がデータ・ソースの CPU の半分であることを示します。2 を設定すると、フェデレーテッド CPU がデータ・ソース CPU の 2 倍高速であるということです。
DBNAME	必須。アクセスする Informix データベースの名前を指定します。
DB2_MAXIMAL_PUSHDOWN	照会オプティマイザーがアクセス・プランの選択に使用する 1 次基準を指定します。有効な値は Y と N です。デフォルトは N であり、その場合、照会オプティマイザーは見積コストが最小になるプランを選択します。Y は、照会オプティマイザーが、照会操作を最も多くデータ・ソースにプッシュダウンするアクセス・プランを選択するということを指定します。この基準を満たすアクセス・プランが複数ある場合、最もコストの小さいプランが選択されます。
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	照会からの並行非同期要求の最大数を指定します。有効な値は -1 から 64000 です。デフォルトは、1 です。-1 は、要求数がフェデレーテッド照会オプティマイザーによって決定されるということを指定します。0 を指定すると、データ・ソースは付加的な非同期要求を受け入れることができません。
DB2_TWO_PHASE_COMMIT	フェデレーテッド・サーバーがデータ・ソースに 2 フェーズ・コミット・プロトコルと 1 フェーズ・コミット・プロトコルのどちらで接続するかを指定します。有効な値は Y と N です。デフォルトは N であり、その場合、フェデレーテッド・サーバーは 1 フェーズ・コミット・プロトコルを使用して接続します。Y を指定すると、フェデレーテッド・サーバーは 2 フェーズ・コミット・プロトコルを使用して接続します。

表 59. Informix のサーバー・オプション (続き)

名前	説明
DB2_UM_PLUGIN	ユーザー・マッピング・プラグインのインプリメンテーションを指定します。Java で作成されたプラグインの場合、ユーザー・マッピング・リポジトリ・クラスに対応するクラス名として、大/小文字の区別があるストリングを指定します。例えば、 「UserMappingRepositoryLDAP」のように指定します。C で作成されたプラグインの場合、任意の有効な C ライブラリー名を指定します。
DB2_UM_PLUGIN_LANG	ユーザー・マッピング・プラグインの言語を指定します。有効な値は、Java および C です。デフォルトは Java です。
FOLD_ID	データ・ソースに送信されるユーザー ID で大/小文字のどちらを使用するかを指定します。デフォルト値はありません。フェデレーテッド・サーバーは、ユーザー ID を大文字で送信し、この ID が失敗した場合は、ユーザー ID を小文字で送信します。有効な値は U (大文字)、L (小文字)、および N (NULL) です。NULL に設定すると、パフォーマンスが低下する場合がありますため、NULL 値の使用は避けてください。
FOLD_PW	データ・ソースに送信されるパスワードで大/小文字のどちらを使用するかを指定します。デフォルト値はありません。フェデレーテッド・サーバーは、パスワードを大文字で送信し、このパスワードが失敗した場合は、パスワードを小文字で送信します。有効な値は U (大文字)、L (小文字)、および N (NULL) です。NULL に設定すると、パフォーマンスが低下する場合がありますため、NULL 値の使用は避けてください。
INFORMIX_CLIENT_LOCALE	フェデレーテッド・サーバーとデータ・ソース・サーバー間の接続に使用する CLIENT_LOCALE を指定します。この値は、任意の有効な Informix ロケールです。このオプションを指定しない場合、CLIENT_LOCALE 環境変数は、db2dj.ini ファイルの中で指定されている値に設定されます。db2dj.ini で CLIENT_LOCALE 環境変数が指定されていない場合、INFORMIX_CLIENT_LOCALE は、フェデレーテッド・データベースのコード・ページとテリトリーに最も近い Informix ロケールに設定されます。

表 59. Informix のサーバー・オプション (続き)

名前	説明
INFORMIX_DB_LOCALE	<p>フェデレーテッド・サーバーとデータ・ソース・サーバーとの間の接続に使用する Informix DB_LOCALE を指定します。INFORMIX_DB_LOCALE オプションを指定しない場合、Informix DB_LOCALE 環境変数は、db2dj.ini ファイルで指定された値に設定されます。db2dj.ini ファイルに値が指定されていない場合、Informix DB_LOCALE 環境変数は設定されません。</p>
INFORMIX_LOCK_MODE	<p>Informix データ・ソースに設定するロック・モードを指定します。Informix ラッパーは、Informix データ・ソースに接続した直後に、SET LOCK MODE コマンドを発行します。有効な値は W、N、および数値です。デフォルトは W であり、その場合ラッパーは、ロックの解放を無制限に待機します。N が指定されている場合は待機せず、ただちにエラーが戻されます。待機時間の最大値の秒数を表す数値を使用して指定してください。デッドロックまたはタイムアウトが発生した場合には、ALTER SERVER ステートメントを使用して INFORMIX_LOCK_MODE オプションの値を変更します。例:</p> <pre>ALTER SERVER TYPE informix VERSION 9 WRAPPER informix OPTIONS (ADD informix_lock_mode '60')</pre>
IO_RATIO	<p>データ・ソース入出力システムがフェデレーテッド・サーバー入出力システムと比較して、どれほど速いかまたは遅いかを指定します。有効な値は 0 より大きく、1×10^{23} より小さい値です。デフォルトは 1.0 です。値は、任意の有効な倍精度表記 (例えば 123E10、123、または 1.21E4) で表現できます。1 を設定すると、フェデレーテッド・サーバーとデータ・ソース・サーバーの入出力速度が同じで、1:1 の比率であるということです。0.5 を設定すると、フェデレーテッド・サーバーの速度がデータ・ソースの速度の半分であることを示します。2 を設定すると、フェデレーテッドの速度がデータ・ソースの速度の 2 倍高速であるということです。</p>

表 59. Informix のサーバー・オプション (続き)

名前	説明
IUD_APP_SVPT_ENFORCE	フェデレーテッド・サーバーでアプリケーション・セーブポイント・ステートメントの使用が必須かどうかを指定します。有効な値は Y と N です。デフォルトは Y であり、この場合、データ・ソースでアプリケーション・セーブポイント・ステートメントが必須ではない状態で挿入、更新、または削除の操作中にエラーが発生した場合には、フェデレーテッド・サーバーはトランザクションをロールバックし、SQL エラー・コード SQL1476N が戻されます。デフォルト設定を使用することが推奨されています。
NODE	必須。データ・ソースをそのリレーショナル・データベース管理システムに対してインスタンスとして定義する名前を指定します。
OLD_NAME_GEN	データ・ソースの列名と索引名を、フェデレーテッド・サーバーのニックネーム列名とローカル索引名に変換する方法を指定します。有効な値は Y と N です。デフォルトは N であり、その場合、生成される名前とデータ・ソース中の名前には密接な対応関係があります。Y を指定すると、生成される名前は IBM WebSphere Federation Server バージョン 9 以前で作成された名前と同じになります。したがって、それらの名前はデータ・ソースの名前と密接に対応しないことがあります。
PUSHDOWN	フェデレーテッド・サーバーが、データ・ソースによる操作の評価を許可するかどうかを指定します。有効な値は Y および N です。デフォルトは Y であり、その場合、データ・ソースは操作を評価します。N は、列名を指定した SELECT のみを含む SQL ステートメントをフェデレーテッド・サーバーが送信するということを指定します。述部 (WHERE= など)、列およびスカラー関数 (MAX や MIN など)、ソート (ORDER BY または GROUP BY など)、および結合は、フェデレーテッド・サーバーがデータ・ソースに送信するどの SQL にも含まれません。

ユーザー・マッピング・オプション

表 60. Informix のユーザー・マッピング・オプション

名前	説明
REMOTE_AUTHID	ローカル・ユーザー ID のマップ先となるリモート・ユーザー ID を指定します。このオプションを指定しない場合は、フェデレーテッド・データベースへの接続に使用する ID が使用されます。
REMOTE_PASSWORD	リモート・ユーザー ID のリモート・パスワードを指定します。このオプションを指定しない場合は、フェデレーテッド・データベースへの接続に使用したパスワードが使用されます。

列オプション

表 61. Informix の列オプション

名前	説明
NUMERIC_STRING	数値ストリングの処理方法を指定します。デフォルトは N です。データ・ソース・ストリング列に数値ストリングのみが含まれ、ブランクを含め他の文字が含まれていない場合、NUMERIC_STRING オプションを Y に設定します。列に対して NUMERIC_STRING が Y に設定されている場合、照会オプションマイザーは、列のデータをソートする場合に支障となり得るブランクがこの列には含まれないことを認識します。データ・ソースの照合シーケンスが、フェデレーテッド・サーバーが使用する照合シーケンスとは異なる場合に、このオプションを使用してください。このオプションを使用する列は、照合シーケンスが異なるためにリモートでの評価から除かれるということはありません。

JDBC オプション・リファレンス

フェデレーテッド・サーバーおよびそのユーザーがデータ・ソースと対話する方法を構成するには、ラッパー、サーバー、ユーザー・マッピング、および列のオプションを設定および変更します。

ラッパー・オプション

次の表は、このデータ・ソースに適用されるオプションをリストし、指定する必要がある必須選択のオプションを示しています。

表 62. JDBC のラッパー・オプション

名前	説明
DB2_FENCED	必須。ラッパーを fenced モードとトラステッド・モードのどちらで実行するかを指定します。DB2 サーバーは fenced モードでの JVM のロードのみをサポートしているため、有効な値は Y だけです。デフォルトは Y であり、ラッパーは fenced モードで実行されます。
DB2_UM_PLUGIN	ユーザー・マッピング・プラグインのインプリメンテーションを指定します。Java で作成されたプラグインの場合、ユーザー・マッピング・リポジトリ・クラスに対応するクラス名として、大/小文字の区別があるストリングを指定します。例えば、「UserMappingRepositoryLDAP」のように指定します。C で作成されたプラグインの場合、任意の有効な C ライブラリー名を指定します。
DB2_UM_PLUGIN_LANG	ユーザー・マッピング・プラグインの言語を指定します。有効な値は、Java および C です。デフォルトは Java です。

サーバー・オプション

表 63. JDBC のサーバー・オプション

名前	説明
COLLATING_SEQUENCE	データ・ソースがフェデレーテッド・データベースと同じデフォルト照合シーケンスを使用するかどうかを指定します。有効な値は Y、N、および I です。I は大/小文字を区別しないことを指定します。デフォルト値は Y です。フェデレーテッド・サーバーに指定された照合シーケンスは、リモート・データ・ソースの照合シーケンスと一致している必要があります。
COMM_RATE	フェデレーテッド・サーバーとデータ・ソース・サーバーとの間の通信速度を指定します (MB/秒)。有効な値は 0 より大きく、2147483648 より小さい整数です。デフォルトは 2 です。

表 63. JDBC のサーバー・オプション (続き)

名前	説明
CPU_RATIO	データ・ソースの CPU がフェデレーテッド・サーバーの CPU と比較してどれほど速いかまたは遅いかを指定します。有効な値は 0 より大きく、 1×10^{23} より小さい値です。デフォルトは 1.0 です。値は、任意の有効な倍精度表記 (例えば 123E10、123、または 1.21E4) で表現できます。1 を設定すると、フェデレーテッド・サーバーとデータ・ソース・サーバーの CPU 速度が同じで、1:1 の比率であるということです。0.5 を設定すると、フェデレーテッド・サーバーの CPU の速度がデータ・ソースの CPU の半分であることを示します。2 を設定すると、フェデレーテッド CPU がデータ・ソース CPU の 2 倍高速であるということです。
DATEFORMAT	データ・ソースが使用する日付形式を指定します。日付形式は、'DD'、'MM'、および 'YY' または 'YYYY' を使用して指定します。区切り文字にはスペース、ハイフン、コンマなどを指定できます。例えば 'YYYY-MM-DD' という形式は、日付を 1958-10-01 などとして指定します。値には NULL 値を含めることができます。
DB2_MAXIMAL_PUSHDOWN	照会オプティマイザーがアクセス・プランの選択に使用する 1 次基準を指定します。有効な値は Y および N です。デフォルトは N であり、その場合、照会オプティマイザーは見積コストが最小になるプランを選択します。Y は、照会オプティマイザーが、照会操作を最も多くデータ・ソースにプッシュダウンするアクセス・プランを選択することを指定します。
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	照会からの並行非同期要求の最大数を指定します。有効な値は -1 から 64000 です。デフォルトは 0 です。-1 を指定すると、要求の数はフェデレーテッド照会オプティマイザーが決定します。0 を指定すると、データ・ソースは付加的な非同期要求を受け入れることができません。

表 63. JDBC のサーバー・オプション (続き)

名前	説明
DB2_UM_PLUGIN	<p>ユーザー・マッピング・プラグインのインプリメンテーションを指定します。Java で作成されたプラグインの場合、ユーザー・マッピング・リポジトリ・クラスに対応するクラス名として、大小文字の区別があるストリングを指定します。例えば、</p> <p>「UserMappingRepositoryLDAP」のように指定します。C で作成されたプラグインの場合、任意の有効な C ライブラリー名を指定します。</p>
DB2_UM_PLUGIN_LANG	<p>ユーザー・マッピング・プラグインの言語を指定します。有効な値は、Java および C です。デフォルトは Java です。</p>
DRIVER_CLASS	<p>JDBC ドライバー・ライブラリーを指定します。JDBC ドライバーが JDBC 仕様のバージョン 3.0 以降に準拠している場合、サーバーを複数の JDBC データ・ソースに対して登録できます。JDBC 仕様と、</p> <p>DRIVER_CLASS サーバー・オプションの設定方法については、JDBC ドライバーの資料を参照してください。</p> <p>例 以下の例では、 com.ibm.db2.jcc.DB2Driver JDBC ドライバー・ライブラリーが指定されています。</p> <p>DRIVER_CLASS 'com.ibm.db2.jcc.DB2Driver'</p> <p>重要: このオプションを指定する場合は、URL サーバー・オプションも指定する必要があります。</p>
DRIVER_PACKAGE	<p>JDBC ドライバー・パッケージを指定します。複数のドライバー・クラス・パッケージは、パス分離文字を使用して指定します。Windows オペレーティング・システムではセミコロンを、Linux および Unix オペレーティング・システムではコロンを使用してください。</p> <p>例 以下は、Linux オペレーティング・システムで複数のドライバー・パッケージをコロンを使用して指定する例です。</p> <p>DRIVER_PACKAGE '/path1/file1.jar:/path2/file2.jar'</p>

表 63. JDBC のサーバー・オプション (続き)

名前	説明
FOLD_ID	<p>データ・ソースに送信されるユーザー ID で大/小文字のどちらを使用するかを指定します。デフォルト値はありません。フェデレーテッド・サーバーは、ユーザー ID を大文字で送信し、この ID が失敗した場合は、ユーザー ID を小文字で送信します。有効な値は U (大文字)、L (小文字)、および N (NULL) です。NULL に設定すると、パフォーマンスが低下する場合がありますため、NULL 値の使用は避けてください。</p>
FOLD_PW	<p>データ・ソースに送信されるパスワードで大/小文字のどちらを使用するかを指定します。デフォルト値はありません。フェデレーテッド・サーバーは、パスワードを大文字で送信し、このパスワードが失敗した場合は、パスワードを小文字で送信します。有効な値は U (大文字)、L (小文字)、および N (NULL) です。NULL に設定すると、パフォーマンスが低下する場合がありますため、NULL 値の使用は避けてください。</p>
IO_RATIO	<p>データ・ソース入出力システムがフェデレーテッド・サーバー入出力システムと比較して、どれほど速いかまたは遅いかを指定します。有効な値は 0 より大きく、1×10^{23} より小さい値です。デフォルトは 1.0 です。値は、任意の有効な倍精度表記 (例えば 123E10、123、または 1.21E4) で表現できます。1 を設定すると、フェデレーテッド・サーバーとデータ・ソース・サーバーの入出力速度が同じで、1:1 の比率であるということです。0.5 を設定すると、フェデレーテッド・サーバーの速度がデータ・ソースの速度の半分であることを示します。2 を設定すると、フェデレーテッドの速度がデータ・ソースの速度の 2 倍高速であるということです。</p>

表 63. JDBC のサーバー・オプション (続き)

名前	説明
JDBC_LOG	<p>JDBC ラッパーで、エラー・トレース用のログ・ファイルを作成するかどうかを指定します。有効な値は Y および N です。デフォルトは N であり、ログ・ファイルは作成されません。このサーバー・オプションを Y に設定すると、JDBC ラッパーは JDBC ログ・ファイルを <code>jdbc_wrapper_prod_id.log</code> ファイル (ここで、<code>prod_id</code> は製品 ID) に書き込みます。このログ・ファイルは、DB2 データベース・マネージャー構成パラメーター <code>DIAGPATH</code> によって指定されるディレクトリ内に保管されます。UNIX システムのデフォルト・ディレクトリは、<code>inst_home/sql/lib/db2dump</code> です。</p> <p>推奨: このサーバー・オプションを YES に設定するとシステムのパフォーマンスに影響を与えるため、実動システムではロギングを有効にしないことをお勧めします。</p>
OLD_NAME_GEN	<p>データ・ソースの列名と索引名を、フェデレーテッド・サーバーのニックネーム列名とローカル索引名に変換する方法を指定します。有効な値は Y と N です。デフォルトは N であり、その場合、生成される名前とデータ・ソース中の名前には密接な対応関係があります。Y を指定すると、生成される名前は IBM WebSphere Federation Server バージョン 9 以前で作成された名前と同じになります。したがって、それらの名前はデータ・ソースの名前と密接に対応しないことがあります。</p>
PUSHDOWN	<p>フェデレーテッド・サーバーが、データ・ソースによる操作の評価を許可するかどうかを指定します。有効な値は Y および N です。デフォルトは Y であり、その場合、データ・ソースは操作を評価します。N は、列名を指定した SELECT のみを含む SQL ステートメントをフェデレーテッド・サーバーが送信するということを指定します。述部 (WHERE= など)、列およびスカラー関数 (MAX や MIN など)、ソート (ORDER BY または GROUP BY など)、および結合は、フェデレーテッド・サーバーがデータ・ソースに送信する SQL にも含まれません。</p>

表 63. JDBC のサーバー・オプション (続き)

名前	説明
TIMEFORMAT	<p>データ・ソースが使用する時刻形式を指定します。'hh12'、'hh24'、'mm'、'ss'、'AM'、および 'A.M' を使用して、時刻形式を指定します。例えば、'hh24:mm:22' という形式は、時刻を 16:00:00 などとして指定します。'hh12:mm:ss AM' という形式は、時刻を 8:00:00 AM などとして指定します。値には NULL 値を含めることができます。</p>
TIMESTAMPFORMAT	<p>データ・ソースが使用するタイム・スタンプ形式を指定します。有効な値は、DATEFORMAT オプションおよび TIMEFORMAT オプションで使用される形式です。10 分の 1 秒の場合は 'n'、100 分の 1 秒の場合は 'nn'、1000 分の 1 秒 (ミリ秒) の場合は 'nnn' を指定します。最大でマイクロ秒の 'nnnnnn' まで可能です。例えば、'YYY-MM-DD-hh24:mm:ss.nnnnnn' という形式は、タイム・スタンプを 1994-01-01-24:00:00.000000 などとして指定します。値には NULL 値を含めることができます。</p>
URL	<p>リモート・サーバーの JDBC 接続ストリングを指定します。</p> <p>JDBC 接続ストリングは次の 3 つの部分から構成され、各部の間はいずれもコロンで区切られます。</p> <ul style="list-style-type: none"> • データベース・プロトコル • データベース・タイプ名、または接続ドライバー名 • 別名またはサブネームを使用したデータベース ID <p>例 以下の例では、JDBC 接続ストリングは jdbc:db2://cn.ibm.com:50471/testdb です。</p> <p>URL 'jdbc:db2://cn.ibm.com:50471/testdb'</p> <p>JDBC ドライバーが JDBC 仕様のバージョン 3.0 以降に準拠している場合、サーバーを複数の JDBC データ・ソースに対して登録できます。JDBC 仕様と、URL サーバー・オプションの設定方法については、JDBC ドライバーの資料を参照してください。</p> <p>重要: このオプションを指定する場合は、DRIVER_CLASS サーバー・オプションも指定する必要があります。</p>

表 63. JDBC のサーバー・オプション (続き)

名前	説明
VARCHAR_NO_TRAILING_BLANKS	データ・ソースが、末尾ブランク文字を少なくとも 1 つ含む VARCHAR 列を持つかどうかを指定します。デフォルトは N です。この場合、VARCHAR 列には、末尾ブランク文字が少なくとも 1 つ含まれます。

ユーザー・マッピング・オプション

表 64. JDBC のユーザー・マッピング・オプション

名前	説明
REMOTE_AUTHID	ローカル・ユーザー ID のマップ先となるリモート・ユーザー ID を指定します。このオプションを指定しない場合は、フェデレーテッド・データベースへの接続に使用する ID が使用されます。
REMOTE_PASSWORD	リモート・ユーザー ID のリモート・パスワードを指定します。このオプションを指定しない場合は、フェデレーテッド・データベースへの接続に使用したパスワードが使用されます。

列オプション

表 65. JDBC の列オプション

名前	説明
NUMERIC_STRING	列にブランクを含む数字のストリングが含まれているかどうかを指定します。有効な値は Y と N です。デフォルトは N です。この場合、列には、ブランクを含む数字のストリングが含まれません。後ろに末尾ブランクが続く数字のストリングだけが列に含まれる場合は、Y と指定しないでください。列に対して NUMERIC_STRING を Y に設定すると、列のデータをソートする場合に支障となり得るブランクがこの列には含まれないことを、照会オプティマイザーが認識します。データ・ソースの照合シーケンスが、フェデレーテッド・サーバーが使用する照合シーケンスとは異なる場合に、このオプションを使用してください。このオプションを使用する列は、照合シーケンスが異なるためにリモートでの評価から除かれるということはありません。
VARCHAR_NO_TRAILING_BLANKS	VARCHAR 列に末尾ブランクが少なくとも 1 つあるかどうかを指定します。

Microsoft SQL Server オプション・リファレンス

フェデレーテッド・サーバーおよびそのユーザーがデータ・ソースと対話する方法を構成するには、ラッパー、サーバー、ユーザー・マッピング、および列のオプションを設定および変更します。

ラッパー・オプション

以下の表に、このデータ・ソースに適用されるオプションと、指定する必要がある必須指定のオプションをまとめました。

表 66. Microsoft SQL Server のラッパー・オプション

名前	説明
DB2_FENCED	必須。ラッパーを fenced モードとトラステッド・モードのどちらで実行するかを指定します。有効な値は Y と N です。デフォルトは N であり、その場合、ラッパーはトラステッド・モードで実行されます。
DB2_UM_PLUGIN	ユーザー・マッピング・プラグインのインプリメンテーションを指定します。Java で作成されたプラグインの場合、ユーザー・マッピング・リポジトリ・クラスに対応するクラス名として、大/小文字の区別があるストリングを指定します。例えば、「UserMappingRepositoryLDAP」のように指定します。C で作成されたプラグインの場合、任意の有効な C ライブラリー名を指定します。
DB2_UM_PLUGIN_LANG	ユーザー・マッピング・プラグインの言語を指定します。有効な値は、Java および C です。デフォルトは Java です。

サーバー・オプション

表 67. Microsoft SQL Server のサーバー・オプション

名前	説明
CODEPAGE	データ・ソースのクライアント構成のコード化文字セットに対応するコード・ページを指定します。非 Unicode のフェデレーテッド・データベースを使用する UNIX システムおよび Microsoft Windows システムでは、デフォルトはフェデレーテッド・データベースのコード・ページです。Unicode のフェデレーテッド・データベースを使用する UNIX システムでは、デフォルトは 1208 です。Unicode のフェデレーテッド・データベースを使用する Windows システムでは、デフォルトは 1202 です。

表 67. Microsoft SQL Server のサーバー・オプション (続き)

名前	説明
COLLATING_SEQUENCE	データ・ソースがフェデレーテッド・データベースと同じデフォルト照合シーケンスを使用するかどうかを指定します。有効な値は Y、N、および I です。I は大/小文字を区別しないことを指定します。デフォルト値は Y です。フェデレーテッド・サーバーに指定された照合シーケンスは、リモート・データ・ソースの照合シーケンスと一致している必要があります。
COMM_RATE	データ・ソースの CPU がフェデレーテッド・サーバーの CPU と比較してどれほど速いかまたは遅いかを指定します。有効な値は 0 より大きく、 1×10^{23} より小さい値です。デフォルトは 1.0 です。値は、任意の有効な倍精度表記 (例えば 123E10、123、または 1.21E4) で表現できます。1 を設定すると、フェデレーテッド・サーバーとデータ・ソース・サーバーの CPU 速度が同じで、1:1 の比率であるということです。0.5 を設定すると、フェデレーテッド・サーバーの CPU の速度がデータ・ソースの CPU の半分であることを示します。2 を設定すると、フェデレーテッド CPU がデータ・ソース CPU の 2 倍高速であるということです。
CPU_RATIO	データ・ソースの CPU の実行速度がフェデレーテッド・サーバーの CPU の速度と比較してどれほど速いかまたは遅いかを指定します。有効な値は 0 より大きく、 1×10^{23} より小さい値です。デフォルトは 1.0 です。値は、任意の有効な倍精度表記 (例えば 123E10、123、または 1.21E4) で表現できます。
DBNAME	必須。アクセスしようとするデータベースの別名を指定します。この値は大文字小文字が区別されます。
DB2_MAXIMAL_PUSHDOWN	照会オプティマイザーがアクセス・プランの選択に使用する 1 次基準を指定します。有効な値は Y および N です。デフォルトは N であり、その場合、照会オプティマイザーは見積コストが最小になるプランを選択します。Y は、照会オプティマイザーが、照会操作を最も多くデータ・ソースにプッシュダウンするアクセス・プランを選択することを指定します。この基準を満たすアクセス・プランが複数ある場合、最もコストの小さいプランが選択されます。

表 67. Microsoft SQL Server のサーバー・オプション (続き)

名前	説明
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	照会からの並行非同期要求の最大数を指定します。有効な値は -1 から 64000 です。デフォルトは 1 です。-1 を指定すると、要求の数はフェデレーテッド照会オプティマイザーが決定します。0 を指定すると、データ・ソースは付加的な非同期要求を受け入れることができません。
DB2_TWO_PHASE_COMMIT	フェデレーテッド・サーバーがデータ・ソースに 2 フェーズ・コミット・プロトコルと 1 フェーズ・コミット・プロトコルのどちらで接続するかを指定します。有効な値は Y と N です。デフォルトは N であり、その場合、フェデレーテッド・サーバーは 1 フェーズ・コミット・プロトコルを使用して接続します。Y を指定すると、フェデレーテッド・サーバーは 2 フェーズ・コミット・プロトコルを使用して接続します。 重要: このオプションを Y に設定した場合は、XA_OPEN_STRING_OPTION も指定する必要があります。
DB2_UM_PLUGIN	ユーザー・マッピング・プラグインのインプリメンテーションを指定します。Java で作成されたプラグインの場合、ユーザー・マッピング・リポジトリ・クラスに対応するクラス名として、大/小文字の区別があるストリングを指定します。例えば、 「UserMappingRepositoryLDAP」のように指定します。C で作成されたプラグインの場合、任意の有効な C ライブラリー名を指定します。
DB2_UM_PLUGIN_LANG	ユーザー・マッピング・プラグインの言語を指定します。有効な値は、Java および C です。デフォルトは Java です。
FOLD_ID	データ・ソースに送信されるユーザー ID で大/小文字のどちらを使用するかを指定します。デフォルト値はありません。フェデレーテッド・サーバーは、ユーザー ID を大文字で送信し、この ID が失敗した場合は、ユーザー ID を小文字で送信します。有効な値は U (大文字)、L (小文字)、および N (NULL) です。NULL に設定すると、パフォーマンスが低下する可能性があるため、NULL 値の使用は避けてください。

表 67. Microsoft SQL Server のサーバー・オプション (続き)

名前	説明
FOLD_PW	データ・ソースに送信されるパスワードで大/小文字のどちらを使用するかを指定します。デフォルト値はありません。フェデレーテッド・サーバーは、パスワードを大文字で送信し、このパスワードが失敗した場合は、パスワードを小文字で送信します。有効な値は U (大文字)、L (小文字)、および N (NULL) です。NULL に設定すると、パフォーマンスが低下する場合がありますため、NULL 値の使用は避けてください。
IO_RATIO	データ・ソース入出力システムがフェデレーテッド・サーバー入出力システムと比較して、どれほど速いかまたは遅いかを指定します。有効な値は 0 より大きく、 1×10^{23} より小さい値です。デフォルトは 1.0 です。値は、任意の有効な倍精度表記 (例えば 123E10、123、または 1.21E4) で表現できます。1 を設定すると、フェデレーテッド・サーバーとデータ・ソース・サーバーの入出力速度が同じで、1:1 の比率であるということです。0.5 を設定すると、フェデレーテッド・サーバーの速度がデータ・ソースの速度の半分であることを示します。2 を設定すると、フェデレーテッドの速度がデータ・ソースの速度の 2 倍高速であるということです。
NODE	必須。フェデレーテッド・サーバーが Microsoft Windows を使用している場合、NODE の値は、Microsoft SQL Server 用に指定したシステムの DSN 名になります。フェデレーテッド・サーバーが UNIX または Linux を使用している場合、NODE の値は odbc.ini ファイルで定義されます。この値は大文字小文字が区別されます。
OLD_NAME_GEN	データ・ソースの列名と索引名を、フェデレーテッド・サーバーのニックネーム列名とローカル索引名に変換する方法を指定します。有効な値は Y と N です。デフォルトは N であり、その場合、生成される名前とデータ・ソース中の名前には密接な対応関係があります。Y を指定すると、生成される名前は IBM WebSphere Federation Server バージョン 9 以前で作成された名前と同じになります。したがって、それらの名前はデータ・ソースの名前と密接に対応しないことがあります。

表 67. Microsoft SQL Server のサーバー・オプション (続き)

名前	説明
PASSWORD	パスワードがデータ・ソースに送信されるかどうかを指定します。デフォルト値は Y です。
PUSHDOWN	フェデレーテッド・サーバーが、データ・ソースによる操作の評価を許可するかどうかを指定します。有効な値は Y および N です。デフォルトは Y であり、その場合、データ・ソースは操作を評価します。N は、列名を指定した SELECT のみを含む SQL ステートメントをフェデレーテッド・サーバーが送信するということを指定します。述部 (WHERE= など)、列およびスカラー関数 (MAX や MIN など)、ソート (ORDER BY または GROUP BY など)、および結合は、フェデレーテッド・サーバーがデータ・ソースに送信する SQL にも含まれません。
XA_OPEN_STRING_OPTIONS	DB2_TWO_PHASE_COMMIT が Y に設定されている場合に必須です。Microsoft SQL Server レジストリーのリソース・マネージャー ID を指定します。

ユーザー・マッピング・オプション

表 68. Microsoft SQL Server のユーザー・マッピング・オプション

名前	説明
REMOTE_AUTHID	ローカル・ユーザー ID のマップ先となるリモート・ユーザー ID を指定します。このオプションを指定しない場合は、フェデレーテッド・データベースへの接続に使用する ID が使用されます。
REMOTE_PASSWORD	リモート・ユーザー ID のリモート・パスワードを指定します。このオプションを指定しない場合は、フェデレーテッド・データベースへの接続に使用したパスワードが使用されます。

列オプション

表 69. Microsoft SQL Server の列オプション

名前	説明
BINARY_REP	特定のバイナリー・データ・タイプの列を識別します。 BINARY_REP 列オプションの値を Y に設定すると、フェデレーテッド・サーバーに SQL_BINARY データ・タイプのプッシュダウンを強制することができます。リモート・データ・ソースで照会オプティマイザーがバイナリー形式の列の比較を実行できるようにすると、照会パフォーマンスを改善できます。
NUMERIC_STRING	数値ストリングの処理方法を指定します。デフォルトは N です。データ・ソース・ストリング列に数値ストリングのみが含まれ、空白を含め他の文字が含まれていない場合、NUMERIC_STRING オプションを Y に設定します。列に対して NUMERIC_STRING が Y に設定されている場合、照会オプティマイザーは、列のデータをソートする場合に支障となり得る空白がこの列には含まれないことを認識します。データ・ソースの照合シーケンスが、フェデレーテッド・サーバーが使用する照合シーケンスとは異なる場合に、このオプションを使用してください。このオプションを使用する列は、照合シーケンスが異なるためにリモートでの評価から除かれるということはありません。

ODBC オプション・リファレンス

フェデレーテッド・サーバーおよびそのユーザーがデータ・ソースと対話する方法を構成するには、ラッパー、サーバー、ユーザー・マッピング、および列のオプションを設定および変更します。

ラッパー・オプション

次の表は、このデータ・ソースに適用されるオプションをリストし、指定する必要がある必須選択のオプションを示しています。

表 70. ODBC のラッパー・オプション

名前	説明
DB2_FENCED	<p>必須。ラッパーを fenced モードとトラステッド・モードのどちらで実行するかを指定します。有効な値は Y と N です。デフォルトは N です。この場合、ラッパーはトラステッド・モードで実行されます。</p> <p>重要: UNIX システムでこのオプションを Y に設定した場合は、DB2_SOURCE_CLIENT_MODE ラッパー・オプションも設定する必要があります。</p>
DB2_SOURCE_CLIENT_MODE	<p>データ・ソースのクライアントが 32 ビットであり、フェデレーテッド・サーバー上のデータベース・インスタンスが 64 ビットであることを指定します。有効値は 32 ビットのみです。このオプションは、UNIX のみで有効です。</p> <p>重要: このオプションを設定する場合、DB2_FENCED ラッパー・オプションを Y に設定する必要もあります。</p>
DB2_UM_PLUGIN	<p>ユーザー・マッピング・プラグインのインプリメンテーションを指定します。Java で作成されたプラグインの場合、ユーザー・マッピング・リポジトリ・クラスに対応するクラス名として、大/小文字の区別があるストリングを指定します。例えば、 「UserMappingRepositoryLDAP」のように指定します。C で作成されたプラグインの場合、任意の有効な C ライブラリー名を指定します。</p>
DB2_UM_PLUGIN_LANG	<p>ユーザー・マッピング・プラグインの言語を指定します。有効な値は、Java および C です。デフォルトは Java です。</p>
MODULE	<p>UNIX システム上で稼働するフェデレーテッド・サーバーの場合は必須です。ODBC Driver Manager インプリメンテーションまたは SQL/CLI インプリメンテーションを含むライブラリーの絶対パスを指定します。UNIX の場合、デフォルトはありません。Microsoft Windows システムの場合、デフォルトは odbc32.d11 です。</p>

サーバー・オプション

表 71. ODBC のサーバー・オプション

名前	説明
CODEPAGE	データ・ソースのクライアント構成のコード化文字セットに対応するコード・ページを指定します。非 Unicode のフェデレーテッド・データベースを使用する UNIX システムおよび Windows システムでは、デフォルトはフェデレーテッド・データベースが使用するコード・ページです。Unicode のフェデレーテッド・データベースを使用する UNIX システムでは、デフォルトは 1208 です。Unicode のフェデレーテッド・データベースを使用する Windows システムでは、デフォルトは 1202 です。
COLLATING_SEQUENCE	データ・ソースがフェデレーテッド・データベースと同じデフォルト照合シーケンスを使用するかどうかを指定します。有効な値は Y、N、および I です。I は大/小文字を区別しないことを指定します。デフォルト値は Y です。フェデレーテッド・サーバーに指定された照合シーケンスは、リモート・データ・ソースの照合シーケンスと一致している必要があります。
COMM_RATE	フェデレーテッド・サーバーとデータ・ソース・サーバーとの間の通信速度を指定します (MB/秒)。有効な値は 0 より大きく、2147483648 より小さい整数です。デフォルトは 2 です。
CPU_RATIO	データ・ソースの CPU がフェデレーテッド・サーバーの CPU と比較してどれほど速いかまたは遅いかを指定します。有効な値は 0 より大きく、 1×10^{23} より小さい値です。デフォルトは 1.0 です。値は、任意の有効な倍精度表記 (例えば 123E10、123、または 1.21E4) で表現できます。1 を設定すると、フェデレーテッド・サーバーとデータ・ソース・サーバーの CPU 速度が同じで、1:1 の比率であるということです。0.5 を設定すると、フェデレーテッド・サーバーの CPU の速度がデータ・ソースの CPU の半分であることを示します。2 を設定すると、フェデレーテッド CPU がデータ・ソース CPU の 2 倍高速であるということです。

表 71. ODBC のサーバー・オプション (続き)

名前	説明
DATEFORMAT	<p>データ・ソースが使用する日付形式を指定します。日付形式は、'DD'、'MM'、および 'YY' または 'YYYY' を使用して指定します。区切り文字にはスペース、ハイフン、コンマなどを指定できます。例えば 'YYYY-MM-DD' という形式は、日付を 1958-10-01 などとして指定します。値には NULL 値を含めることができます。</p> <p>値を単一引用符で囲んで指定する場合、その単一引用符を保持する必要があります。単一引用符を保持するには、追加の単一引用符 '' を指定します。これらは二重引用符ではなく、単一引用符を 2 つ重ねたものです。例えば ''YYYY-MM-DD'' のように指定します。</p>
DBNAME	<p>アクセスしようとするデータ・ソース・データベースの名前を指定します。</p>
DB2_AUTHID_QUOTE_CHAR	<p>スキーマ名やユーザー名など authid 名に使用する引用文字を指定します。このオプションを指定しない場合は、二重引用符がデフォルトで使用されます。</p>
DB2_ID_QUOTE_CHAR	<p>カラム名など区切り ID に使用する引用文字を指定します。このオプションを指定しない場合は、二重引用符がデフォルトで使用されます。</p>
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	<p>照会からの並行非同期要求の最大数を指定します。有効な値は -1 から 64000 です。デフォルトは 0 です。-1 を指定すると、要求の数はフェデレーテッド照会オプティマイザーが決定します。0 を指定すると、データ・ソースは付加的な非同期要求を受け入れることができません。</p>
DB2_MAXIMAL_PUSHDOWN	<p>照会オプティマイザーがアクセス・プランの選択に使用する 1 次基準を指定します。有効な値は Y および N です。デフォルトは N であり、その場合、照会オプティマイザーは見積コストが最小になるプランを選択します。Y は、照会オプティマイザーが、照会操作を最も多くデータ・ソースにプッシュダウンするアクセス・プランを選択することを指定します。</p>
DB2_TABLE_QUOTE_CHAR	<p>表名に使用する引用文字を指定します。このオプションを指定しない場合は、二重引用符がデフォルトで使用されます。</p>

表 71. ODBC のサーバー・オプション (続き)

名前	説明
DB2_UM_PLUGIN	ユーザー・マッピング・プラグインのインプリメンテーションを指定します。Java で作成されたプラグインの場合、ユーザー・マッピング・リポジトリ・クラスに対応するクラス名として、大/小文字の区別があるストリングを指定します。例えば、 「UserMappingRepositoryLDAP」のように指定します。C で作成されたプラグインの場合、任意の有効な C ライブラリー名を指定します。
DB2_UM_PLUGIN_LANG	ユーザー・マッピング・プラグインの言語を指定します。有効な値は、Java および C です。デフォルトは Java です。
ENABLE_BULK_INSERT	バルク挿入処理を Netezza データ・ソースに対して有効にするかどうかを指定します。有効な値は Y と N です。デフォルトは N です。
FOLD_ID	データ・ソースに送信されるユーザー ID で大/小文字のどちらを使用するかを指定します。デフォルト値はありません。フェデレーテッド・サーバーは、ユーザー ID を大文字で送信し、この ID が失敗した場合は、ユーザー ID を小文字で送信します。有効な値は U (大文字)、L (小文字)、および N (NULL) です。NULL に設定すると、パフォーマンスが低下する場合がありますため、NULL 値の使用は避けてください。
FOLD_PW	データ・ソースに送信されるパスワードで大/小文字のどちらを使用するかを指定します。デフォルト値はありません。フェデレーテッド・サーバーは、パスワードを大文字で送信し、このパスワードが失敗した場合は、パスワードを小文字で送信します。有効な値は U (大文字)、L (小文字)、および N (NULL) です。NULL に設定すると、パフォーマンスが低下する場合がありますため、NULL 値の使用は避けてください。

表 71. ODBC のサーバー・オプション (続き)

名前	説明
IO_RATIO	<p>データ・ソース入出力システムがフェデレーテッド・サーバー入出力システムと比較して、どれほど速いかまたは遅いかを指定します。有効な値は 0 より大きく、1×10^{23} より小さい値です。デフォルトは 1.0 です。値は、任意の有効な倍精度表記 (例えば 123E10、123、または 1.21E4) で表現できます。1 を設定すると、フェデレーテッド・サーバーとデータ・ソース・サーバーの入出力速度が同じで、1:1 の比率であるということです。0.5 を設定すると、フェデレーテッド・サーバーの速度がデータ・ソースの速度の半分であることを示します。2 を設定すると、フェデレーテッドの速度がデータ・ソースの速度の 2 倍高速であるということです。</p>
NODE	<p>必須。DSN の定義時に、ODBC データ・ソースに割り当てられるノード名またはシステム DSN 名を指定します。この値は大文字小文字が区別されます。</p>
OLD_NAME_GEN	<p>データ・ソースの列名と索引名を、フェデレーテッド・サーバーのニックネーム列名とローカル索引名に変換する方法を指定します。有効な値は Y と N です。デフォルトは N であり、その場合、生成される名前とデータ・ソース中の名前には密接な対応関係があります。Y を指定すると、生成される名前は IBM WebSphere Federation Server バージョン 9 以前で作成された名前と同じになります。したがって、それらの名前はデータ・ソースの名前と密接に対応しないことがあります。</p>
PUSHDOWN	<p>フェデレーテッド・サーバーが、データ・ソースによる操作の評価を許可するかどうかを指定します。有効な値は Y および N です。デフォルトは Y であり、その場合、データ・ソースは操作を評価します。N は、列名を指定した SELECT のみを含む SQL ステートメントをフェデレーテッド・サーバーが送信するということを指定します。述部 (WHERE= など)、列およびスカラー関数 (MAX や MIN など)、ソート (ORDER BY または GROUP BY など)、および結合は、フェデレーテッド・サーバーがデータ・ソースに送信するどの SQL にも含まれません。</p>

表 71. ODBC のサーバー・オプション (続き)

名前	説明
TIMEFORMAT	データ・ソースが使用する時刻形式を指定します。'hh12'、'hh24'、'mm'、'ss'、'AM'、および'A.M'を使用して、時刻形式を指定します。例えば、'hh24:mm:22' という形式は、時刻を 16:00:00 などとして指定します。'hh12:mm:ss AM' という形式は、時刻を 8:00:00 AM などとして指定します。値には NULL 値を含めることができます。
TIMESTAMPFORMAT	データ・ソースが使用するタイム・スタンプ形式を指定します。有効な値は、DATEFORMAT オプションおよび TIMEFORMAT オプションで使用される形式です。10 分の 1 秒の場合は 'n'、100 分の 1 秒の場合は 'nn'、1000 分の 1 秒 (ミリ秒) の場合は 'nnn' を指定します。最大でマイクロ秒の 'nnnnnn' まで可能です。例えば、'YYY-MM-DD-hh24:mm:ss.nnnnnn' という形式は、タイム・スタンプを 1994-01-01-24:00:00.000000 などとして指定します。値には NULL 値を含めることができます。
VARCHAR_NO_TRAILING_BLANKS	データ・ソースが、末尾ブランク文字を少なくとも 1 つ含む VARCHAR 列を持つかどうかを指定します。デフォルトは N です。この場合、VARCHAR 列には、末尾ブランク文字が少なくとも 1 つ含まれます。

ユーザー・マッピング・オプション

表 72. ODBC のユーザー・マッピング・オプション

名前	説明
REMOTE_AUTHID	ローカル・ユーザー ID のマップ先となるリモート・ユーザー ID を指定します。このオプションを指定しない場合は、フェデレーテッド・データベースへの接続に使用する ID が使用されます。
REMOTE_PASSWORD	リモート・ユーザー ID のリモート・パスワードを指定します。このオプションを指定しない場合は、フェデレーテッド・データベースへの接続に使用したパスワードが使用されます。

列オプション

表 73. ODBC の列オプション

名前	説明
NUMERIC_STRING	列に空白を含む数字のストリングが含まれているかどうかを指定します。有効な値は Y と N です。デフォルトは N です。この場合、列には、空白を含む数字のストリングが含まれません。後ろに末尾空白が続く数字のストリングだけが列に含まれる場合は、Y と指定しないでください。列に対して NUMERIC_STRING を Y に設定すると、列のデータをソートする場合に支障となり得る空白がこの列には含まれないことを、照会最適化マイザーが認識します。データ・ソースの照合シーケンスが、フェデレーテッド・サーバーが使用する照合シーケンスとは異なる場合に、このオプションを使用してください。このオプションを使用する列は、照合シーケンスが異なるためにリモートでの評価から除かれるということはありません。
VARCHAR_NO_TRAILING_BLANKS	VARCHAR 列に末尾空白が少なくとも 1 つあるかどうかを指定します。

Oracle オプション・リファレンス

フェデレーテッド・サーバーおよびそのユーザーがデータ・ソースと対話する方法を構成するには、ラッパー、サーバー、ユーザー・マッピング、および列のオプションを設定および変更します。

ラッパー・オプション

次の表は、このデータ・ソースに適用されるオプションをリストし、指定する必要がある必須選択のオプションを示しています。

表 74. Oracle のラッパー・オプション

名前	説明
DB2_FENCED	必須。ラッパーを fenced モードとトラステッド・モードのどちらで実行するかを指定します。有効な値は Y と N です。デフォルトは N であり、その場合、ラッパーはトラステッド・モードで実行されます。

表 74. Oracle のラッパー・オプション (続き)

名前	説明
DB2_UM_PLUGIN	ユーザー・マッピング・プラグインのインプリメンテーションを指定します。Java で作成されたプラグインの場合、ユーザー・マッピング・リポジトリ・クラスに対応するクラス名として、大/小文字の区別があるストリングを指定します。例えば、 「UserMappingRepositoryLDAP」のように指定します。C で作成されたプラグインの場合、任意の有効な C ライブラリー名を指定します。
DB2_UM_PLUGIN_LANG	ユーザー・マッピング・プラグインの言語を指定します。有効な値は、Java および C です。デフォルトは Java です。

サーバー・オプション

表 75. Oracle のサーバー・オプション

名前	説明
COLLATING_SEQUENCE	データ・ソースがフェデレーテッド・データベースと同じデフォルト照合シーケンスを使用するかどうかを指定します。有効な値は Y、N、および I です。I は大/小文字を区別しないことを指定します。デフォルト値は Y です。フェデレーテッド・サーバーに指定された照合シーケンスは、リモート・データ・ソースの照合シーケンスと一致している必要があります。
COMM_RATE	フェデレーテッド・サーバーとデータ・ソース・サーバーとの間の通信速度を指定します (MB/秒)。有効な値は 0 より大きく、2147483648 より小さい整数です。デフォルトは 2 です。
CPU_RATIO	データ・ソースの CPU がフェデレーテッド・サーバーの CPU と比較してどれほど速いかまたは遅いかを指定します。有効な値は 0 より大きく、 1×10^{23} より小さい値です。デフォルトは 1.0 です。値は、任意の有効な倍精度表記 (例えば 123E10、123、または 1.21E4) で表現できます。1 を設定すると、フェデレーテッド・サーバーとデータ・ソース・サーバーの CPU 速度が同じで、1:1 の比率であるということです。0.5 を設定すると、フェデレーテッド・サーバーの CPU の速度がデータ・ソースの CPU の半分であることを示します。2 を設定すると、フェデレーテッド CPU がデータ・ソース CPU の 2 倍高速であるということです。

表 75. Oracle のサーバー・オプション (続き)

名前	説明
DB2_MAXIMAL_PUSHDOWN	照会オプティマイザーがアクセス・プランの選択に使用する 1 次基準を指定します。有効な値は Y および N です。デフォルトは N であり、その場合、照会オプティマイザーは見積コストが最小になるプランを選択します。Y は、照会オプティマイザーが、照会操作を最も多くデータ・ソースにプッシュダウンするアクセス・プランを選択することを指定します。
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	照会からの並行非同期要求の最大数を指定します。有効な値は -1 から 64000 です。デフォルトは 1 です。-1 を指定すると、要求の数はフェデレーテッド照会オプティマイザーが決定します。0 を指定すると、データ・ソースは付加的な非同期要求を受け入れることができません。
DB2_TWO_PHASE_COMMIT	フェデレーテッド・サーバーがデータ・ソースに 2 フェーズ・コミット・プロトコルと 1 フェーズ・コミット・プロトコルのどちらで接続するかを指定します。有効な値は Y と N です。デフォルトは N であり、その場合、フェデレーテッド・サーバーは 1 フェーズ・コミット・プロトコルを使用して接続します。Y を指定すると、フェデレーテッド・サーバーは 2 フェーズ・コミット・プロトコルを使用して接続します。
DB2_UM_PLUGIN	ユーザー・マッピング・プラグインのインプリメンテーションを指定します。Java で作成されたプラグインの場合、ユーザー・マッピング・リポジトリ・クラスに対応するクラス名として、大/小文字の区別があるストリングを指定します。例えば、 「UserMappingRepositoryLDAP」のように指定します。C で作成されたプラグインの場合、任意の有効な C ライブラリー名を指定します。
DB2_UM_PLUGIN_LANG	ユーザー・マッピング・プラグインの言語を指定します。有効な値は、Java および C です。デフォルトは Java です。
FED_PROXY_USER	インバウンド接続が非トラステッド接続である場合に、すべてのアウトバウンド・トラステッド接続を確立するために使用する許可 ID を指定します。 重要: このオプションで指定された ID を持つユーザーには、REMOTE_AUTHID と REMOTE_PASSWORD の両方を指定するユーザー・マッピングが必要です。

表 75. Oracle のサーバー・オプション (続き)

名前	説明
FOLD_ID	データ・ソースに送信されるユーザー ID で大/小文字のどちらを使用するかを指定します。デフォルト値はありません。フェデレーテッド・サーバーは、ユーザー ID を大文字で送信し、この ID が失敗した場合は、ユーザー ID を小文字で送信します。有効な値は U (大文字)、L (小文字)、および N (NULL) です。NULL に設定すると、パフォーマンスが低下する場合がありますため、NULL 値の使用は避けてください。
FOLD_PW	データ・ソースに送信されるパスワードで大/小文字のどちらを使用するかを指定します。デフォルト値はありません。フェデレーテッド・サーバーは、パスワードを大文字で送信し、このパスワードが失敗した場合は、パスワードを小文字で送信します。有効な値は U (大文字)、L (小文字)、および N (NULL) です。NULL に設定すると、パフォーマンスが低下する場合がありますため、NULL 値の使用は避けてください。
IO_RATIO	データ・ソース入出力システムがフェデレーテッド・サーバー入出力システムと比較して、どれほど速いかまたは遅いかを指定します。有効な値は 0 より大きく、 1×10^{23} より小さい値です。デフォルトは 1.0 です。値は、任意の有効な倍精度表記 (例えば 123E10、123、または 1.21E4) で表現できます。1 を設定すると、フェデレーテッド・サーバーとデータ・ソース・サーバーの入出力速度が同じで、1:1 の比率であるということです。0.5 を設定すると、フェデレーテッド・サーバーの速度がデータ・ソースの速度の半分であることを示します。2 を設定すると、フェデレーテッドの速度がデータ・ソースの速度の 2 倍高速であるということです。
NODE	必須。Oracle データベース・サーバーが置かれているノードの名前を指定します。ノードの名前は、tnsnames.ora ファイルから取得します。

表 75. Oracle のサーバー・オプション (続き)

名前	説明
OLD_NAME_GEN	データ・ソースの列名と索引名を、フェデレーテッド・サーバーのニックネーム列名とローカル索引名に変換する方法を指定します。有効な値は Y と N です。デフォルトは N であり、その場合、生成される名前とデータ・ソース中の名前には密接な対応関係があります。Y を指定すると、生成される名前は IBM WebSphere Federation Server バージョン 9 以前で作成された名前と同じになります。したがって、それらの名前はデータ・ソースの名前と密接に対応しないことがあります。
PLAN_HINTS	プラン・ヒントを使用可能にするかどうかを指定します。プラン・ヒントはステートメントの一部であり、データ・ソース・オプティマイザーが照会のパフォーマンスを向上させるために使用する追加情報を提供します。データ・ソース・オプティマイザーはこのプラン・ヒントを使用して、索引を使用するかどうか、およびどの索引を使用するか、またはどの表結合順序を使用するかを決定します。有効な値は Y および N です。デフォルトは N であり、プラン・ヒントは使用可能になりません。Y は、データ・ソースがプラン・ヒントをサポートしている場合に、データ・ソースでプラン・ヒントを使用可能にするということを指定します。
PUSHDOWN	フェデレーテッド・サーバーが、データ・ソースによる操作の評価を許可するかどうかを指定します。有効な値は Y および N です。デフォルトは Y であり、フェデレーテッド・サーバーはデータ・ソースによる操作の評価を許可します。N は、フェデレーテッド・サーバーがリモート・データ・ソースから列を取り出すということを指定します。
VARCHAR_NO_TRAILING_BLANKS	特定のサーバーについて、VARCHAR 列に末尾ブランクが含まれるかどうかを指定します。このオプションを 1 つの列に適用するには、VARCHAR_NO_TRAILING_BLANKS 列オプションを使用します。
XA_OPEN_STRING_OPTIONS	ストリングに付加する追加情報を指定します。例えば、この情報はトレース・ファイルのディレクトリーなどです。

ユーザー・マッピング・オプション

表 76. Oracle のユーザー・マッピング・オプション

名前	説明
FED_PROXY_USER	インバウンド接続が非トラステッド接続である場合に、すべてのアウトバウンド・トラステッド接続を確立するために使用する許可 ID を指定します。 重要: このオプションで指定された ID を持つユーザーには、REMOTE_AUTHID と REMOTE_PASSWORD の両方を指定するユーザー・マッピングが必要です。 FED_PROXY_USER ユーザー・マッピング・オプションを指定する場合は、FED_PROXY_USER サーバー・オプションも指定する必要があります。
REMOTE_AUTHID	ローカル・ユーザー ID のマップ先となるリモート・ユーザー ID を指定します。このオプションを指定しない場合は、フェデレーテッド・データベースへの接続に使用する ID が使用されます。
REMOTE_PASSWORD	リモート・ユーザー ID のリモート・パスワードを指定します。このオプションを指定しない場合は、フェデレーテッド・データベースへの接続に使用したパスワードが使用されます。
USE_TRUSTED_CONTEXT	ユーザー・マッピングがトラステッドかどうかを指定します。有効な値は Y と N です。デフォルトは N です。この場合、ユーザー・マッピングはトラステッドではなく、非トラステッド・アウトバウンド接続でしか使用できません。Y は、ユーザー・マッピングがトラステッドであり、トラステッドおよび非トラステッド両方のアウトバウンド接続で使用できるということを指定します。

列オプション

表 77. Oracle の列オプション

名前	説明
NUMERIC_STRING	列に空白を含む数字のストリングが含まれているかどうかを指定します。有効な値は Y と N です。デフォルトは N です。この場合、列には、空白を含む数字のストリングが含まれません。後ろに末尾空白が続く数字のストリングだけが列に含まれる場合は、Y と指定しないでください。列に対して NUMERIC_STRING を Y に設定すると、列のデータをソートする場合に支障となり得る空白がこの列には含まれないことを、照会オプティマイザーが認識します。データ・ソースの照合シーケンスが、フェデレーテッド・サーバーが使用する照合シーケンスとは異なる場合に、このオプションを使用してください。このオプションを使用する列は、照合シーケンスが異なるためにリモートでの評価から除かれるということはありません。
VARCHAR_NO_TRAILING_BLANKS	列に末尾空白が含まれるかどうかを指定します。

スクリプト・オプション・リファレンス

フェデレーテッド・サーバーとそのユーザーがどのようにデータ・ソースと対話するかを構成するには、ラッパー、サーバー、ユーザー・マッピング、ニックネーム、および列の各オプションを設定および変更します。

ラッパー・オプション

次の表は、このデータ・ソースに適用されるオプションをリストし、指定する必要がある必須選択のオプションを示しています。

表 78. スクリプトのラッパー・オプション

名前	説明
DB2_FENCED	必須。ラッパーを fenced モードとトラステッド・モードのどちらで実行するかを指定します。有効な値は Y と N です。デフォルトは N であり、その場合、ラッパーはトラステッド・モードで実行されます。
PROXY_TYPE	フェデレーテッド・サーバーがファイアウォールの背後にある場合に、インターネットへのアクセスに使用するプロキシー・タイプを指定します。有効な値は NONE および SOCKS です。デフォルト値は NONE です。

表 78. スクリプトのラッパー・オプション (続き)

名前	説明
PROXY_SERVER_NAME	プロキシ・サーバーの名前または IP アドレスを指定します。有効な値は、1 から 32760 までの 10 進数のポート番号、またはサービス名です。IPv6 形式は、IPv6 を構成する場合のみ使用します。
PROXY_SERVER_PORT	プロキシ・サーバー上のプロキシ・サービスのポートまたはサービス名を指定します。有効な値は、10 進数の 1 から 32760 までのポート番号か、サービス名です。

サーバー・オプション

表 79. スクリプトのサーバー・オプション

名前	説明
DAEMON_PORT	SCRIPT デーモンが SCRIPT ジョブ要求を listen するときに使用するポート番号を指定します。デフォルトは 4099 です。このポート番号は、デーモン構成ファイルの DAEMON_PORT オプションと同じでなければなりません。サービス名が SCRIPT デーモン用に構成されている場合は、TCP/IP サービス名を使用します。
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	照会からの並行非同期要求の最大数を指定します。有効な値は -1 から 64000 です。デフォルトは 1 です。-1 を指定すると、要求の数はフェデレーテッド照会オプティマイザーが決定します。0 を指定すると、データ・ソースは付加的な非同期要求を受け入れることができません。
NODE	必須。SCRIPT デーモンが実行されるシステムの DNS ホスト名または IP アドレスを指定します。有効な IP アドレスは IPv4 (ドット区切り) 形式または IPv6 (コロン区切り) 形式です。IPv6 形式は、IPv6 を構成する場合のみ使用します。
PROXY_AUTHID	プロキシ・サーバー認証のユーザー名を指定します。
PROXY_PASSWORD	プロキシ・サーバー認証のパスワードを指定します。
PROXY_SERVER_NAME	プロキシ・サーバーの名前または IP アドレスを指定します。有効な値は、1 から 32760 までの 10 進数のポート番号、またはサービス名です。IPv6 形式は、IPv6 を構成する場合のみ使用します。

表 79. スクリプトのサーバー・オプション (続き)

名前	説明
PROXY_SERVER_PORT	プロキシ・サーバー上のプロキシ・サービスのポートまたはサービス名を指定します。有効な値は、10 進数の 1 から 32760 までのポート番号か、サービス名です。
PROXY_TYPE	フェデレーテッド・サーバーがファイアウォールの背後にある場合に、インターネットへのアクセスに使用するプロキシ・タイプを指定します。有効な値は NONE および SOCKS です。デフォルト値は NONE です。

ユーザー・マッピング・オプション

表 80. スクリプトのユーザー・マッピング・オプション

名前	説明
PROXY_AUTHID	プロキシ・サーバー認証のユーザー名を指定します。
PROXY_PASSWORD	プロキシ・サーバー認証のパスワードを指定します。このパスワードは、フェデレーテッド・データベース・カタログに保管されるときに暗号化されます。

ニックネーム・オプション

表 81. スクリプトのニックネーム・オプション

名前	説明
DATASOURCE	ルート・ニックネームの場合は必須です。呼び出すスクリプトの名前を指定します。このオプションの値として指定するスクリプトは、SCRIPT デーモン構成ファイルでも指定されていなければなりません。 重要: このオプションは、ルート・ニックネームのみで有効です。
NAMESPACES	各列の XPATH および TEMPLATE オプションで使用される、ネーム・スペース接頭部に関連付けられているネーム・スペースを指定します。次の構文を使用します。 <code>NAMESPACES'prefix1="actual_namespace1", prefix2="actual_namespace2"'</code> 複数のネーム・スペースを区切るには、コンマを使用します。例: <code>NAMESPACES='http://www.myweb.com/cust', i='http://www.myweb.com/cust/id', n='http://www.myweb.com/cust/name''</code>

表 81. スクリプトのニックネーム・オプション (続き)

名前	説明
STREAMING	ソース文書を処理するために、論理フラグメントに分ける必要があるかどうかを指定します。このフラグメントは、ニックネームの XPath 式に一致するノードに対応しています。このラッパーは、ソース・データの構文解析と処理をフラグメントごとに行います。このタイプの構文解析では、メモリーの使用量が最小限度で済みます。有効な値は Y と N です。デフォルトは N です。この場合、文書は構文解析されません。このオプションは、ルート・ニックネームのみで有効です。STREAMING と VALIDATE の両方のオプションを Y に設定することは避けてください。
TIMEOUT	デーモンからの結果を待つ最大時間 (分単位) を指定します。デフォルトは 60 です。このオプションは、ルート・ニックネームのみで有効です。
VALIDATE	ソース文書からデータが抽出される前に、その文書が XML スキーマまたは文書タイプ定義 (DTD) に準拠しているかを検証するかどうかを指定します。デフォルトは N です。この場合、検証は行われません。この値を Y に設定する前に、ソース文書が指定する場所にスキーマ・ファイルまたは DTD ファイルを用意します。このオプションは、ルート・ニックネームのみで有効です。STREAMING と VALIDATE の両方のオプションを Y に設定することは避けてください。
XPATH	個々のタプルを表す XML エlementを識別する XPath 式を指定します。子ニックネームの XPATH ニックネーム・オプションは、その親の XPATH ニックネーム・オプションによって指定されるパスのコンテキストで評価されます。この XPath 式は、XPath 列オプションが存在することによって識別される列値を評価するためのコンテキストとして使用されます。

列オプション

表 82. スクリプトの列オプション

名前	説明
DEFAULT	スクリプト入力列のデフォルト値を指定します。SQL 照会で値が指定されない場合は、このデフォルト値が使用されます。

表 82. スクリプトの列オプション (続き)

名前	説明
FOREIGN_KEY	<p>このニックネームが子ニックネームであることを示し、対応する親ニックネームの名前を指定します。ニックネームには、FOREIGN_KEY 列オプションを 1 つしか指定できません。このオプションの値は、大文字小文字を区別します。この列には XPATH オプションを指定しないでください。列は、親のニックネームと子のニックネームを結合するためだけに使用できます。親ニックネームに別のスキーマ名が付けられている場合、FOREIGN_KEY オプションを含む CREATE NICKNAME ステートメントは失敗します。FOREIGN_KEY 文節で参照されるニックネームが CREATE NICKNAME ステートメントで小文字または大/小文字混合として明示的に定義されているのでない限り、このニックネームを FOREIGN_KEY 文節で参照するときは大文字で指定する必要があります。このオプションが列に設定されている場合、その列に他のオプションを設定することはできません。</p>
INPUT_MODE	<p>列の入力モードを指定します。有効な値は、CONFIG および FILE_INPUT です。CONFIG は値を列の入力モードとして処理します。FILE_INPUT は、値を保管するファイルを指定します。指定した値は、ラッパーによって SCRIPT デーモンに渡されます。</p>
POSITION	<p>定位置パラメーターの整数値を指定します。定位置値を整数に設定すると、この入力コマンド行のその位置になければなりません。このオプションを設定すると、照会の実行時に該当位置にスイッチが挿入されます。POSITION を -1 に設定すると、オプションはコマンド行の最後のオプションとして追加されます。POSITION 整数値を同じニックネームの中で 2 回使用することはできません。このオプションは、入力列にのみ適用されます。</p>

表 82. スクリプトの列オプション (続き)

名前	説明
PRIMARY_KEY	1 つ以上の子ニックネームを持つ親ニックネームの場合に必須です。このニックネームを親ニックネームとするということを指定します。列データ・タイプは VARCHAR(16) でなければなりません。ニックネームには、PRIMARY_KEY 列オプションを 1 つしか指定できません。Yes が唯一の有効な値です。この列には XPATH オプションを指定しないでください。列は、親のニックネームと子のニックネームを結合するためだけに使用できます。このオプションが列に設定されている場合、その列に他のオプションを設定することはできません。
SWITCH	コマンド行のスクリプトのフラグを指定します。このオプションの値は、WSSCRIPT.ARGS またはデフォルト値 (存在する場合) によって指定される列の値の前に置かれます。このオプションに値を指定せず、列にデフォルト値が存在する場合には、スイッチ情報なしでデフォルト値が追加されます。このオプションは、入力列の場合に必須です。
SWITCH_ONLY	コマンド行引数なしでのスイッチの使用を可能にします。このオプションは Y に設定します。有効な入力値は Y および N です。入力値が Y の場合、コマンド行にスイッチのみが追加されます。入力値が N の場合、コマンド行に値は追加されません。
VALID_VALUES	列の有効な値のセットを指定します。複数の値を区切るには、セミコロンを使用します。
XPATH	XML 文書の中でこの列に対応するデータを含む Xpath 式を指定します。CREATE NICKNAME ステートメントが XPATH ニックネーム・オプションから XPath 式を適用した後で、ラッパーはその XPath 式を評価します。

Sybase オプション・リファレンス

フェデレーテッド・サーバーおよびそのユーザーがデータ・ソースと対話する方法を構成するには、ラッパー、サーバー、ユーザー・マッピング、および列のオプションを設定および変更します。

ラッパー・オプション

次の表は、このデータ・ソースに適用されるオプションをリストし、指定する必要がある必須選択のオプションを示しています。

表 83. Sybase のラッパー・オプション

名前	説明
DB2_FENCED	必須。ラッパーを fenced モードとトラステッド・モードのどちらで実行するかを指定します。Microsoft Windows では、有効な値は Y と N です。デフォルトは N です。この場合、ラッパーはトラステッド・モードで実行されます。UNIX では、デフォルトで、かつ唯一の有効な値は Y です。ラッパーは、fenced モードで実行しなければなりません。
DB2_UM_PLUGIN	ユーザー・マッピング・プラグインのインプリメンテーションを指定します。Java で作成されたプラグインの場合、ユーザー・マッピング・リポジトリ・クラスに対応するクラス名として、大/小文字の区別があるストリングを指定します。例えば、 「UserMappingRepositoryLDAP」のように指定します。C で作成されたプラグインの場合、任意の有効な C ライブラリー名を指定します。
DB2_UM_PLUGIN_LANG	ユーザー・マッピング・プラグインの言語を指定します。有効な値は Java および C です。デフォルトは Java です。

サーバー・オプション

表 84. Sybase のサーバー・オプション

名前	説明
COLLATING_SEQUENCE	データ・ソースがフェデレーテッド・データベースと同じデフォルト照合シーケンスを使用するかどうかを指定します。有効な値は Y、N、および I です。I は大/小文字を区別しないことを指定します。デフォルト値は Y です。フェデレーテッド・サーバーに指定された照合シーケンスは、リモート・データ・ソースの照合シーケンスと一致している必要があります。
COMM_RATE	フェデレーテッド・サーバーとデータ・ソース・サーバーとの間の通信速度を指定します (MB/秒)。有効な値は 0 より大きく、2147483648 より小さい整数です。デフォルトは 2 です。

表 84. Sybase のサーバー・オプション (続き)

名前	説明
CPU_RATIO	データ・ソースの CPU がフェデレーテッド・サーバーの CPU と比較してどれほど速いかまたは遅いかを指定します。有効な値は 0 より大きく、 1×10^{23} より小さい値です。デフォルトは 1.0 です。値は、任意の有効な倍精度表記 (例えば 123E10、123、または 1.21E4) で表現できます。1 を設定すると、フェデレーテッド・サーバーとデータ・ソース・サーバーの CPU 速度が同じで、1:1 の比率であるということです。0.5 を設定すると、フェデレーテッド・サーバーの CPU の速度がデータ・ソースの CPU の半分であることを示します。2 を設定すると、フェデレーテッド CPU がデータ・ソース CPU の 2 倍高速であるということです。
CONV_EMPTY_STRING	フェデレーテッド・サーバーが、レプリケーション・タスク中に空ストリングをスペースに変換するかどうかを指定します。有効な値は Y と N です。デフォルトは N です。この場合、フェデレーテッド・サーバーは空ストリングを変換しません。空ストリングが保管される非 NULL 可能文字列をデータ・ソースが持っているときは、このオプションを Y に設定してください。
DBNAME	必須。アクセスしようとするデータベースの名前を指定します。データベースの名前は、Sybase サーバーから取得します。
DB2_ID_QUOTE_CHAR	カラム名など区切り ID に使用する引用文字を指定します。このオプションを指定しない場合は、二重引用符がデフォルトで使用されます。
DB2_MAXIMAL_PUSHDOWN	照会オプティマイザーがアクセス・プランの選択に使用する 1 次基準を指定します。有効な値は Y および N です。デフォルトは N であり、その場合、照会オプティマイザーは見積コストが最小になるプランを選択します。Y は、照会オプティマイザーが、照会操作を最も多くデータ・ソースにプッシュダウンするアクセス・プランを選択することを指定します。
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	照会からの並行非同期要求の最大数を指定します。有効な値は -1 から 64000 です。デフォルトは 1 です。-1 を指定すると、要求の数はフェデレーテッド照会オプティマイザーが決定します。0 を指定すると、データ・ソースは付加的な非同期要求を受け入れることができません。

表 84. Sybase のサーバー・オプション (続き)

名前	説明
DB2_TWO_PHASE_COMMIT	フェデレーテッド・サーバーがデータ・ソースに 2 フェーズ・コミット・プロトコルと 1 フェーズ・コミット・プロトコルのどちらで接続するかを指定します。有効な値は Y と N です。デフォルトは N であり、その場合、フェデレーテッド・サーバーは 1 フェーズ・コミット・プロトコルを使用して接続します。Y を指定すると、フェデレーテッド・サーバーは 2 フェーズ・コミット・プロトコルを使用して接続します。
DB2_UM_PLUGIN	ユーザー・マッピング・プラグインのインプリメンテーションを指定します。Java で作成されたプラグインの場合、ユーザー・マッピング・リポジトリ・クラスに対応するクラス名として、大/小文字の区別があるストリングを指定します。例えば、 「UserMappingRepositoryLDAP」のように指定します。C で作成されたプラグインの場合、任意の有効な C ライブラリー名を指定します。
DB2_UM_PLUGIN_LANG	ユーザー・マッピング・プラグインの言語を指定します。有効な値は Java および C です。デフォルトは Java です。
FOLD_ID	データ・ソースに送信されるユーザー ID で大/小文字のどちらを使用するかを指定します。デフォルト値はありません。フェデレーテッド・サーバーは、ユーザー ID を大文字で送信し、この ID が失敗した場合は、ユーザー ID を小文字で送信します。有効な値は U (大文字)、L (小文字)、および N (NULL) です。NULL に設定すると、パフォーマンスが低下する場合がありますため、NULL 値の使用は避けてください。
FOLD_PW	データ・ソースに送信されるパスワードで大/小文字のどちらを使用するかを指定します。デフォルト値はありません。フェデレーテッド・サーバーは、パスワードを大文字で送信し、このパスワードが失敗した場合は、パスワードを小文字で送信します。有効な値は U (大文字)、L (小文字)、および N (NULL) です。NULL に設定すると、パフォーマンスが低下する場合がありますため、NULL 値の使用は避けてください。

表 84. Sybase のサーバー・オプション (続き)

名前	説明
IFILE	<p>デフォルトのインターフェース・ファイルの代わりに使用する Sybase インターフェース・ファイルのパスと名前を指定します。Sybase ラッパーは、インターフェース・ファイルを、次の場所から指定された順序で検索します。Microsoft Windows の場合は、IFILE サーバー・オプション、次に %DB2PATH%\interfaces ディレクトリー、最後に %SYBASE%\ini\sql.ini ディレクトリー。UNIX の場合は、IFILE サーバー・オプション、次に sqllib/interfaces ディレクトリー、最後に \$\$SYBASE/interfaces ディレクトリー。</p>
IO_RATIO	<p>データ・ソース入出力システムがフェデレーテッド・サーバー入出力システムと比較して、どれほど速いかまたは遅いかを指定します。有効な値は 0 より大きく、1×10^{23} より小さい値です。デフォルトは 1.0 です。値は、任意の有効な倍精度表記 (例えば 123E10、123、または 1.21E4) で表現できます。1 を設定すると、フェデレーテッド・サーバーとデータ・ソース・サーバーの入出力速度が同じで、1:1 の比率であるということです。0.5 を設定すると、フェデレーテッド・サーバーの速度がデータ・ソースの速度の半分であることを示します。2 を設定すると、フェデレーテッドの速度がデータ・ソースの速度の 2 倍高速であるということです。</p>
LOGIN_TIMEOUT	<p>フェデレーテッド・サーバーがログイン要求を中止する前に待機する時間 (秒単位) を指定します。デフォルトは 0 です。この場合、フェデレーテッド・サーバーが待機する時間は無制限です。</p>
NODE	<p>必須。Sybase サーバーが置かれているノードの名前を指定します。ノード名は Sybase インターフェース・ファイルにあります。</p>

表 84. Sybase のサーバー・オプション (続き)

名前	説明
OLD_NAME_GEN	データ・ソースの列名と索引名を、フェデレーテッド・サーバーのニックネーム列名とローカル索引名に変換する方法を指定します。有効な値は Y と N です。デフォルトは N であり、その場合、生成される名前とデータ・ソース中の名前には密接な対応関係があります。Y を指定すると、生成される名前は IBM WebSphere Federation Server バージョン 9 以前で作成された名前と同じになります。したがって、それらの名前はデータ・ソースの名前と密接に対応しないことがあります。
PACKET_SIZE	クライアント・ライブラリーが表データ・ストリーム (TDS) パケットの送信に使用するパケット・サイズ (バイト単位) を指定します。Sybase ラッパーが大量のテキスト・データおよびイメージ・データを送信または受信する必要がある場合は、PACKET_SIZE を大きくします。
PLAN_HINTS	プラン・ヒントを使用可能にするかどうかを指定します。プラン・ヒントはステートメントの一部であり、データ・ソース・オプティマイザーが照会のパフォーマンスを向上させるために使用する追加情報を提供します。データ・ソース・オプティマイザーはこのプラン・ヒントを使用して、索引を使用するかどうか、およびどの索引を使用するか、またはどの表結合順序を使用するかを決定します。有効な値は Y および N です。デフォルトは N であり、プラン・ヒントは使用可能になりません。Y は、データ・ソースがプラン・ヒントをサポートしている場合に、データ・ソースでプラン・ヒントを使用可能にするということを指定します。
PUSHDOWN	フェデレーテッド・サーバーが、データ・ソースによる操作の評価を許可するかどうかを指定します。有効な値は Y および N です。デフォルトは Y であり、その場合、データ・ソースは操作を評価します。N は、列名を指定した SELECT のみを含む SQL ステートメントをフェデレーテッド・サーバーが送信するということを指定します。述部 (WHERE= など)、列およびスカラー関数 (MAX や MIN など)、ソート (ORDER BY または GROUP BY など)、および結合は、フェデレーテッド・サーバーがデータ・ソースに送信するどの SQL にも含まれません。

表 84. Sybase のサーバー・オプション (続き)

名前	説明
SERVER_PRINCIPAL_NAME	プリンシパル名がノード名と異なる場合は、Sybase サーバーのプリンシパル名を指定します。デフォルト値はノード名です。
TIMEOUT	リモート・サーバーがコマンドに応答するのをフェデレーテッド・サーバーが待つ最大時間 (秒単位) を指定します。デフォルトは 0 です。これは時間の制限がないということを指定します。
XA_OPEN_STRING_OPTIONS	Sybase DTM XA インターフェースのオープン・ストリングを指定します。これらのストリングは、LRM 名、ユーザー名、およびパスワードに追加されるものです。

ユーザー・マッピング・オプション

表 85. Sybase のユーザー・マッピング・オプション

オプション	説明
ENABLE_KERBEROS_CONNECTION	Kerberos セキュリティー・メカニズムを指定します。デフォルト値 N を指定すると、デフォルトのセキュリティ・メカニズムが使用されます。値 Y を指定すると、Kerberos セキュリティー・メカニズムが有効になります。
REMOTE_AUTHID	ローカル・ユーザー ID のマップ先となるリモート・ユーザー ID を指定します。このオプションを指定しない場合は、フェデレーテッド・データベースへの接続に使用する ID が使用されます。
REMOTE_PASSWORD	リモート・ユーザー ID のリモート・パスワードを指定します。このオプションを指定しない場合は、フェデレーテッド・データベースへの接続に使用したパスワードが使用されます。

列オプション

表 86. Sybase の列オプション

オプション	説明
NUMERIC_STRING	数値ストリングの処理方法を指定します。デフォルトは N です。データ・ソース・ストリング列に数値ストリングのみが含まれ、ブランクを含め他の文字が含まれていない場合、NUMERIC_STRING オプションを Y に設定します。列に対して NUMERIC_STRING が Y に設定されている場合、照会オペティマイザーは、列のデータをソートする場合に支障となり得るブランクがこの列には含まれないことを認識します。データ・ソースの照合シーケンスが、フェデレーテッド・サーバーが使用する照合シーケンスとは異なる場合に、このオプションを使用してください。このオプションを使用する列は、照合シーケンスが異なるためにリモートでの評価から除かれるということはありません。

Teradata オプション・リファレンス

フェデレーテッド・サーバーおよびそのユーザーがデータ・ソースと対話する方法を構成するには、ラッパー、サーバー、ユーザー・マッピング、および列のオプションを設定および変更します。

ラッパー・オプション

次の表は、このデータ・ソースに適用されるオプションをリストし、指定する必要がある必須選択のオプションを示しています。

表 87. Teradata のラッパー・オプション

名前	説明
DB2_FENCED	必須。ラッパーを fenced モードとトラステッド・モードのどちらで実行するかを指定します。Microsoft Windows では、有効な値は Y と N です。デフォルトは N です。この場合、ラッパーはトラステッド・モードで実行されます。UNIX では、デフォルト値は Y です。この場合、ラッパーは、fenced モードで実行しなければなりません。

表 87. Teradata のラッパー・オプション (続き)

名前	説明
DB2_UM_PLUGIN	ユーザー・マッピング・プラグインのインプリメンテーションを指定します。Java で作成されたプラグインの場合、ユーザー・マッピング・リポジトリ・クラスに対応するクラス名として、大/小文字の区別があるストリングを指定します。例えば、 「UserMappingRepositoryLDAP」のように指定します。C で作成されたプラグインの場合、任意の有効な C ライブラリー名を指定します。
DB2_UM_PLUGIN_LANG	ユーザー・マッピング・プラグインの言語を指定します。有効な値は、Java および C です。デフォルトは Java です。

サーバー・オプション

表 88. Teradata のサーバー・オプション

名前	説明
COLLATING_SEQUENCE	データ・ソースがフェデレーテッド・データベースと同じデフォルト照合シーケンスを使用するかどうかを指定します。有効な値は Y、N、および I です。I は大/小文字を区別しないことを指定します。デフォルト値は Y です。フェデレーテッド・サーバーに指定された照合シーケンスは、リモート・データ・ソースの照合シーケンスと一致している必要があります。
COMM_RATE	フェデレーテッド・サーバーとデータ・ソース・サーバーとの間の通信速度を指定します (MB/秒)。有効な値は 0 より大きく、2147483648 より小さい整数です。デフォルトは 2 です。
CPU_RATIO	データ・ソースの CPU がフェデレーテッド・サーバーの CPU と比較してどれほど速いかまたは遅いかを指定します。有効な値は 0 より大きく、 1×10^{23} より小さい値です。デフォルトは 1.0 です。値は、任意の有効な倍精度表記 (例えば 123E10、123、または 1.21E4) で表現できます。1 を設定すると、フェデレーテッド・サーバーとデータ・ソース・サーバーの CPU 速度が同じで、1:1 の比率であるということです。0.5 を設定すると、フェデレーテッド・サーバーの CPU の速度がデータ・ソースの CPU の半分であることを示します。2 を設定すると、フェデレーテッド CPU がデータ・ソース CPU の 2 倍高速であるということです。

表 88. Teradata のサーバー・オプション (続き)

名前	説明
DB2_MAXIMAL_PUSHDOWN	照会オプティマイザーがアクセス・プランの選択に使用する 1 次基準を指定します。有効な値は Y および N です。デフォルトは N であり、その場合、照会オプティマイザーは見積コストが最小になるプランを選択します。Y は、照会オプティマイザーが、照会操作を最も多くデータ・ソースにプッシュダウンするアクセス・プランを選択することを指定します。
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	照会からの並行非同期要求の最大数を指定します。有効な値は -1 から 64000 です。デフォルトは 1 です。-1 を指定すると、要求の数はフェデレーテッド照会オプティマイザーが決定します。0 を指定すると、データ・ソースは付加的な非同期要求を受け入れることができません。
DB2_UM_PLUGIN	ユーザー・マッピング・プラグインのインプリメンテーションを指定します。Java で作成されたプラグインの場合、ユーザー・マッピング・リポジトリ・クラスに対応するクラス名として、大/小文字の区別があるストリングを指定します。例えば、 「UserMappingRepositoryLDAP」のように指定します。C で作成されたプラグインの場合、任意の有効な C ライブラリー名を指定します。
DB2_UM_PLUGIN_LANG	ユーザー・マッピング・プラグインの言語を指定します。有効な値は、Java および C です。デフォルトは Java です。
IO_RATIO	データ・ソース入出力システムがフェデレーテッド・サーバー入出力システムと比較して、どれほど速いかまたは遅いかを指定します。有効な値は 0 より大きく、 1×10^{23} より小さい値です。デフォルトは 1.0 です。値は、任意の有効な倍精度表記 (例えば 123E10、123、または 1.21E4) で表現できます。1 を設定すると、フェデレーテッド・サーバーとデータ・ソース・サーバーの入出力速度が同じで、1:1 の比率であるということです。0.5 を設定すると、フェデレーテッド・サーバーの速度がデータ・ソースの速度の半分であることを示します。2 を設定すると、フェデレーテッドの速度がデータ・ソースの速度の 2 倍高速であるということです。
NODE	必須。Teradata サーバーの別名または IP アドレスを指定します。

表 88. Teradata のサーバー・オプション (続き)

名前	説明
OLD_NAME_GEN	データ・ソースの列名と索引名を、フェデレーテッド・サーバーのニックネーム列名とローカル索引名に変換する方法を指定します。有効な値は Y と N です。デフォルトは N であり、その場合、生成される名前とデータ・ソース中の名前には密接な対応関係があります。Y を指定すると、生成される名前は IBM WebSphere Federation Server バージョン 9 以前で作成された名前と同じになります。したがって、それらの名前はデータ・ソースの名前と密接に対応しないことがあります。
PUSHDOWN	フェデレーテッド・サーバーが、データ・ソースによる操作の評価を許可するかどうかを指定します。有効な値は Y および N です。デフォルトは Y であり、その場合、データ・ソースは操作を評価します。N は、列名を指定した SELECT のみを含む SQL ステートメントをフェデレーテッド・サーバーが送信するという事を指定します。述部 (WHERE= など)、列およびスカラー関数 (MAX や MIN など)、ソート (ORDER BY または GROUP BY など)、および結合は、フェデレーテッド・サーバーがデータ・ソースに送信するどの SQL にも含まれません。

ユーザー・マッピング・オプション

表 89. Teradata のユーザー・マッピング・オプション

名前	説明
REMOTE_AUTHID	ローカル・ユーザー ID のマップ先となるリモート・ユーザー ID を指定します。このオプションを指定しない場合は、フェデレーテッド・データベースへの接続に使用する ID が使用されます。
REMOTE_PASSWORD	リモート・ユーザー ID のリモート・パスワードを指定します。このオプションを指定しない場合は、フェデレーテッド・データベースへの接続に使用したパスワードが使用されます。

列オプション

表 90. Teradata の列オプション

名前	説明
NUMERIC_STRING	数値ストリングの処理方法を指定します。デフォルトは N です。データ・ソース・ストリング列に数値ストリングのみが含まれ、ブランクを含め他の文字が含まれていない場合、NUMERIC_STRING オプションを Y に設定します。列に対して NUMERIC_STRING が Y に設定されている場合、照会オペティマイザーは、列のデータをソートする場合に支障となり得るブランクがこの列には含まれないことを認識します。データ・ソースの照合シーケンスが、フェデレーテッド・サーバーが使用する照合シーケンスとは異なる場合に、このオプションを使用してください。このオプションを使用する列は、照合シーケンスが異なるためにリモートでの評価から除かれるということはありません。

表構造ファイル・オプション・リファレンス

フェデレーテッド・サーバーとそのユーザーがどのようにデータ・ソースと対話するかを構成するには、ラッパー、サーバー、ニックネーム、および列の各オプションを設定および変更します。

次の表は、このデータ・ソースに適用されるオプションをリストし、指定する必要がある必須選択のオプションを示しています。

ラッパー・オプション

表 91. 表構造ファイルのラッパー・オプション

名前	説明
DB2_FENCED	必須。ラッパーを fenced モードとトラステッド・モードのどちらで実行するかを指定します。有効な値は Y と N です。デフォルトは N であり、その場合、ラッパーはトラステッド・モードで実行されます。

サーバー・オプション

表 92. 表構造ファイルのサーバー・オプション

名前	説明
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	照会からの並行非同期要求の最大数を指定します。有効な値は -1 から 64000 です。デフォルトは 1 です。-1 を指定すると、要求の数はフェデレーテッド照会オプティマイザーが決定します。0 を指定すると、データ・ソースは付加的な非同期要求を受け入れることができません。

ニックネーム・オプション

表 93. 表構造ファイルのニックネーム・オプション

名前	説明
COLUMN_DELIMITER	表構造ファイルの列を区切る文字として使用する単一文字を指定します。デフォルトはコンマ (,) です。単一引用符は区切り文字としては使用できません。列区切り文字は、ファイル全体で一貫性がなければなりません。NULL 値は 2 つの区切り文字が続くことによって表されます。NULL フィールドが行の最後にある場合、区切り文字に行の終止符を続けることによって表されます。列区切り文字は、列の有効なデータとして存在することはできません。
CODEPAGE	データ・ソースのファイルのコード・ページを指定します。このオプションは、Unicode を使用するフェデレーテッド・データベースに対してのみ有効です。ソース・データは、指定されたコード・ページから Unicode に変換されます。
FILE_PATH	表構造ファイルまでの絶対パスを指定します。ファイル名は単一引用符で囲みます。データ・ファイルは標準ファイルまたはシンボリック・リンクでなければならず、パイプや別の非標準ファイル・タイプにすることはできません。 重要: FILE_PATH オプションを指定する場合、DOCUMENT 列を指定しないでください。

表 93. 表構造ファイルのニックネーム・オプション (続き)

名前	説明
KEY_COLUMN	<p>ファイルをソートするときの基準となる列の名前を指定します。DOCUMENT 列オプションがある列は、キー列にはできません。単一のキーのみがサポートされています。値は、CREATE NICKNAME ステートメントに定義されている列の名前にしなければなりません。列は必ず昇順でソートされます。NOT NULL オプションをニックネーム・ステートメントの中の列の定義に追加することにより、キー列が NULL 可能ではないということ指定しなければなりません。この値は大文字小文字の区別があります。</p>
SORTED	<p>データ・ソースにあるファイルが昇順にソートされるか否かを指定します。有効な値は Y および N です。デフォルトは N であり、データ・ソースにあるファイルは昇順にソートされません。ソートされるデータ・ソースは、LC_COLLATE 各国語サポート・カテゴリーの設定によって定義されている現在のロケールの照合順序に従って、昇順でソートされなければなりません。データ・ソースがソートされるように指定する場合、VALIDATE_DATA_FILE オプションを Y に設定してください。</p>
VALIDATE_DATA_FILE	<p>ソートされるファイルの場合、このオプションは、キー列が昇順でソートされていることの検証と NULL キーの検査を、ラッパーが行うかどうかを指定します。この検証は、ニックネームが最初に作成されたときに一度だけ行われます。デフォルトは N です。この場合、ソート順は検証されません。このオプションは、SORTED オプションが Y に設定され、DOCUMENT オプションが指定されていない場合にのみ有効です。</p>

列オプション

表 94. 表構造ファイルの列オプション

オプション	説明
DOCUMENT	ニックネームの作成時にはなく、照会の実行時にファイル・パスを指定できるようにします。有効値は FILE のみです。 DOCUMENT オプションで指定できるのは、ニックネームあたり 1 つの列のみです。 DOCUMENT オプションに関連付けられる列は、データ・タイプ VARCHAR または CHAR の列でなければなりません。 FILE_PATH ニックネーム・オプションではなく、DOCUMENT ニックネーム列オプションを使用するということは、照会の実行中に、このニックネームに対応するファイルが提供されることを意味します。DOCUMENT オプションに FILE 値が指定されている場合、照会の実行時に提供される値は、このニックネームのニックネーム定義に一致するスキーマを持つファイルの絶対パスです。

Web サービス・オプション・リファレンス

フェデレーテッド・サーバーおよびそのユーザーがデータ・ソースと対話する方法を構成するには、ラッパー、サーバー、ユーザー・マッピング、ニックネーム、および列のオプションを設定および変更します。

ラッパー・オプション

次の表は、このデータ・ソースに適用されるオプションをリストし、指定する必要がある必須選択のオプションを示しています。

表 95. Web サービスのラッパー・オプション

名前	説明
DB2_FENCED	必須。ラッパーを fenced モードとトラステッド・モードのどちらで実行するかを指定します。有効な値は Y と N です。デフォルトは N であり、その場合、ラッパーはトラステッド・モードで実行されます。
DB2_UM_PLUGIN	ユーザー・マッピング・プラグインのインプリメンテーションを指定します。Java で作成されたプラグインの場合、ユーザー・マッピング・リポジトリ・クラスに対応するクラス名として、大/小文字の区別があるストリングを指定します。例えば、 「UserMappingRepositoryLDAP」のように指定します。C で作成されたプラグインの場合、任意の有効な C ライブラリー名を指定します。

表 95. Web サービスのラッパー・オプション (続き)

名前	説明
DB2_UM_PLUGIN_LANG	ユーザー・マッピング・プラグインの言語を指定します。有効な値は、Java および C です。デフォルトは Java です。
PROXY_TYPE	フェデレーテッド・サーバーがファイアウォールの背後にある場合に、インターネットへのアクセスに使用するプロキシー・タイプを指定します。有効な値は NONE、HTTP、および SOCKS です。デフォルト値は NONE です。
PROXY_SERVER_NAME	プロキシー・サーバーの名前または IP アドレスを指定します。有効な値は、1 から 32760 までの 10 進数のポート番号、またはサービス名です。IPv6 形式は、IPv6 を構成する場合のみ使用します。
PROXY_SERVER_PORT	プロキシー・サーバー上のプロキシー・サービスのポートまたはサービス名を指定します。有効な値は、10 進数の 1 から 32760 までのポート番号か、サービス名です。
SSL_KEYSTORE_FILE	SSL または TSL を使用する通信用の証明書ストレージ・ファイルを指定します。有効な値は、フェデレーテッド・データベース・エージェントまたは fenced モード・プロセスによってアクセス可能な絶対パス名です。デフォルトは <i>install path/cfg/WSWrapperKeystore.kdb</i> です。
SSL_KEYSTORE_PASSWORD	SSL_KEYSTORE_FILE オプションで、ファイルにアクセスするために使用するパスワードを指定します。有効な値はパスワード (フェデレーテッド・データベース・カタログに保管されるときに暗号化される) および <i>file:file_name</i> (<i>file_name</i> は、スタッシュ・ファイルへの絶対パス) です。
SSL_VERIFY_SERVER_CERTIFICATE	SSL 認証の際にサーバー証明書を検証するかどうかを指定します。有効な値は Y と N です。デフォルトは N です。この場合、証明書は検証されません。

サーバー・オプション

表 96. Web サービスのサーバー・オプション

名前	説明
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	照会からの並行非同期要求の最大数を指定します。有効な値は -1 から 64000 です。デフォルトは 1 です。-1 を指定すると、要求の数はフェデレーテッド照会オプティマイザーが決定します。0 を指定すると、データ・ソースは付加的な非同期要求を受け入れることができません。
DB2_UM_PLUGIN	ユーザー・マッピング・プラグインのインプリメンテーションを指定します。Java で作成されたプラグインの場合、ユーザー・マッピング・リポジトリ・クラスに対応するクラス名として、大/小文字の区別があるストリングを指定します。例えば、 「UserMappingRepositoryLDAP」のように指定します。C で作成されたプラグインの場合、任意の有効な C ライブラリー名を指定します。
DB2_UM_PLUGIN_LANG	ユーザー・マッピング・プラグインの言語を指定します。有効な値は、Java および C です。デフォルトは Java です。
PROXY_AUTHID	プロキシ・サーバー認証のユーザー名を指定します。
PROXY_PASSWORD	プロキシ・サーバー認証のパスワードを指定します。
PROXY_SERVER_NAME	プロキシ・サーバーの名前または IP アドレスを指定します。有効な値は、1 から 32760 までの 10 進数のポート番号、またはサービス名です。IPv6 形式は、IPv6 を構成する場合のみ使用します。
PROXY_SERVER_PORT	プロキシ・サーバー上のプロキシ・サービスのポートまたはサービス名を指定します。有効な値は、10 進数の 1 から 32760 までのポート番号か、サービス名です。
PROXY_TYPE	フェデレーテッド・サーバーがファイアウォールの背後にある場合に、インターネットへのアクセスに使用するプロキシ・タイプを指定します。有効な値は NONE、HTTP、および SOCKS です。デフォルト値は NONE です。
SSL_CLIENT_CERTIFICATE_LABEL	SSL 認証の際に送信するクライアント証明書を指定します。値を指定しない場合は、フェデレーテッド・データベースの現在の許可 ID を使用して、証明書を見つけます。

表 96. Web サービスのサーバー・オプション (続き)

名前	説明
SSL_KEYSTORE_FILE	SSL または TSL を使用する通信用の証明書ストレージ・ファイルを指定します。有効な値は、フェデレーテッド・データベース・エージェントまたは fenced モード・プロセスによってアクセス可能な絶対パス名です。デフォルトは <i>install path/cfg/WSWrapperKeystore.kdb</i> です。
SSL_KEYSTORE_PASSWORD	SSL_KEYSTORE_FILE オプションで、ファイルにアクセスするために使用するパスワードを指定します。有効な値はパスワード (フェデレーテッド・データベース・カタログに保管されるときに暗号化される) および <i>file:file_name</i> (<i>file_name</i> は、スタッシュ・ファイルへの絶対パス) です。
SSL_VERIFY_SERVER_CERTIFICATE	SSL 認証の際にサーバー証明書を検証するかどうかを指定します。有効な値は Y と N です。デフォルトは N です。この場合、証明書は検証されません。

ユーザー・マッピング・オプション

表 97. Web サービスのユーザー・マッピング・オプション

名前	説明
PROXY_AUTHID	プロキシ・サーバー認証のユーザー名を指定します。
PROXY_PASSWORD	プロキシ・サーバー認証のパスワードを指定します。このパスワードは、フェデレーテッド・データベース・カタログに保管されるときに暗号化されます。
REMOTE_AUTHID	ローカル・ユーザー ID のマップ先となるリモート・ユーザー ID を指定します。このオプションを指定しない場合は、フェデレーテッド・データベースへの接続に使用する ID が使用されます。
REMOTE_PASSWORD	リモート・ユーザー ID のリモート・パスワードを指定します。このオプションを指定しない場合は、フェデレーテッド・データベースへの接続に使用したパスワードが使用されます。
SSL_CLIENT_CERTIFICATE_LABEL	SSL 認証の際に送信するクライアント証明書を指定します。値を指定しない場合は、フェデレーテッド・データベースの現在の許可 ID を使用して、証明書を見つけます。

ニックネーム・オプション

表 98. Web サービスのニックネーム・オプション

名前	説明
NAMESPACES	<p>各列の XPATH および TEMPLATE オプションで使用される、ネーム・スペース接頭部に関連付けられているネーム・スペースを指定します。次の構文を使用します。</p> <pre>NAMESPACES'prefix1="actual_namespace1", prefix2="actual_namespace2"'</pre> <p>複数のネーム・スペースを区切るには、コンマを使用します。例:</p> <pre>NAMESPACES='http://www.myweb.com/cust', i='http://www.myweb.com/cust/id', n='http://www.myweb.com/cust/name''</pre>
SOAPACTION	<p>ルート・ニックネームの場合は必須です。Web サービス記述言語 (WSDL) 形式からの URI SOAPACTION 属性を指定します。URL には、コロン区切り IPv6 アドレスを、大括弧で囲んで含めることができます (例えば <code>http://[1080:0:0:0:8:800:200C:417A]</code>)。注: このオプションは、非ルート・ニックネームでは無効です。</p>
STREAMING	<p>ソース文書を処理するために、論理フラグメントに分ける必要があるかどうかを指定します。このフラグメントは、ニックネームの XPath 式に一致するノードに対応しています。このラッパーは、ソース・データの構文解析と処理をフラグメントごとに行います。このタイプの構文解析では、メモリーの使用量が最小限度で済みます。有効な値は Y と N です。デフォルトは N です。この場合、文書は構文解析されません。このオプションは、ルート・ニックネームのみで有効です。</p>
TEMPLATE	<p>SOAP 要求を構成するために使用する、ニックネーム・テンプレートのフラグメントを指定します。フラグメントは、指定されたテンプレートの構文に準拠していなければなりません。このオプションは、ルート・ニックネームのみで有効です。</p>
URL	<p>ルート・ニックネームの場合は必須です。Web サービス・エンドポイントの URL を指定します。サポートされているプロトコルは HTTP および HTTPS です。URL には、コロン区切り IPv6 アドレスを、大括弧で囲んで含めることができます (例えば <code>http://[1080:0:0:0:8:800:200C:417A]</code>)。</p>

表 98. Web サービスのニックネーム・オプション (続き)

名前	説明
XML_CODESET	XML データの送受信に使用するエンコード方式を指定します。このオプションは、内部エンコード方式をオーバーライドします。
XPATH	必須。個々のタプルを表す SOAP 応答エレメントを識別する XPath 式を指定します。この XPath 式は、XPATH ニックネーム列オプションによって識別される列値を評価するためのコンテキストとして使用されます。

列オプション

表 99. Web サービスの列オプション

名前	説明
ESCAPE_INPUT	XML 特殊文字が XML 入力値で置換されるかどうかを指定します。このオプションは、繰り返しエレメントを含む XML フラグメントを含めるなど、XML フラグメントを入力として含めるために使用します。有効な値は Y と N です。N がデフォルトです。この場合、XML 入力値は保持されます。列データ・タイプは、VARCHAR または CHAR でなければなりません。ESCAPE_INPUT が Y に設定されている場合は、TEMPLATE 列オプションも指定する必要があります。

表 99. Web サービスの列オプション (続き)

名前	説明
FOREIGN_KEY	<p>このニックネームが子ニックネームであることを示し、対応する親ニックネームの名前を指定します。ニックネームには、FOREIGN_KEY 列オプションを 1 つしか指定できません。このオプションの値は、大文字小文字を区別します。この列には XPATH オプションを指定しないでください。列は、親のニックネームと子のニックネームを結合するためだけに使用できます。親ニックネームに別のスキーマ名が付けられている場合、FOREIGN_KEY オプションを含む CREATE NICKNAME ステートメントは失敗します。FOREIGN_KEY 文節で参照されるニックネームが CREATE NICKNAME ステートメントで小文字または大/小文字混合として明示的に定義されているのでない限り、このニックネームを FOREIGN_KEY 文節で参照するときは大文字で指定する必要があります。</p> <p>注:</p> <ul style="list-style-type: none"> このオプションが列に設定されている場合、その列に他のオプションを設定することはできません。 この列オプションを設定した場合は、後から ALTER NICKNAME ステートメントを使用してこのオプションをドロップすることはできません。代わりに、ニックネームをドロップしてから、この列オプションを付けずにニックネームを再作成する必要があります。

表 99. Web サービスの列オプション (続き)

名前	説明
PRIMARY_KEY	<p>1 つ以上の子ニックネームを持つ親ニックネームの場合に必須です。このニックネームを親ニックネームとするということを指定します。列データ・タイプは VARCHAR(16) でなければなりません。ニックネームには、PRIMARY_KEY 列オプションを 1 つしか指定できません。Yes が唯一の有効な値です。この列には XPATH オプションを指定しないでください。列は、親のニックネームと子のニックネームを結合するためだけに使用できます。</p> <p>注:</p> <ul style="list-style-type: none"> このオプションが列に設定されている場合、その列に他のオプションを設定することはできません。 この列オプションを設定した場合は、後から ALTER NICKNAME ステートメントを使用してこのオプションをドロップすることはできません。代わりに、ニックネームをドロップしてから、この列オプションを付けずにニックネームを再作成する必要があります。
SOAPACTIONCOLUMN	<p>照会を実行するときに Web サービス・エンドポイントの SOAP アクションを動的に指定する列を指定します。このオプションは、ルート・ニックネームのみで有効です。ホスト名が IPv6 (コロン区切り) アドレスの場合、ホスト名を大括弧で囲みます。例: 「http:// [1080:0:0:0:8:800:200C:417A]:99/soap」。</p> <p>このオプションが列に設定されている場合、その列に他のオプションを設定することはできません。</p>
TEMPLATE	<p>XML 入力文書を構成するために使用する、列テンプレートのフラグメントを指定します。フラグメントは、指定されたテンプレートの構文に準拠していなければなりません。</p> <p>注: この列オプションを設定した場合は、後から ALTER NICKNAME ステートメントを使用してこのオプションをドロップすることはできません。代わりに、ニックネームをドロップしてから、この列オプションを付けずにニックネームを再作成する必要があります。</p>

表 99. Web サービスの列オプション (続き)

名前	説明
URLCOLUMN	照会を実行するときに Web サービス・エンドポイントの SOAP アクションを動的に指定する列を指定します。このオプションは、ルート・ニックネームのみで有効です。ホスト名が IPv6 (コロン区切り) アドレスの場合、ホスト名を大括弧で囲みます。例: 「http:// [1080:0:0:0:8:800:200C:417A]:99/soap」。このオプションが列に設定されている場合、その列に他のオプションを設定することはできません。
XPATH	XML 文書の中でこの列に対応するデータを含む Xpath 式を指定します。CREATE NICKNAME ステートメントが XPATH ニックネーム・オプションから XPath 式を適用した後で、ラッパーはその XPath 式を評価します。 注: この列オプションを設定した場合は、後から ALTER NICKNAME ステートメントを使用してこのオプションをドロップすることはできません。代わりに、ニックネームをドロップしてから、この列オプションを付けずにニックネームを再作成する必要があります。

XML オプション・リファレンス

フェデレーテッド・サーバーとそのユーザーがどのようにデータ・ソースと対話するかを構成するには、ラッパー、サーバー、ユーザー・マッピング、ニックネーム、および列の各オプションを設定および変更します。

ラッパー・オプション

次の表は、このデータ・ソースに適用されるオプションをリストし、指定する必要がある必須選択のオプションを示しています。

表 100. XML のラッパー・オプション

名前	説明
DB2_FENCED	必須。ラッパーを fenced モードとトラステッド・モードのどちらで実行するかを指定します。有効な値は Y と N です。N がデフォルトです。この場合、ラッパーはトラステッド・モードで実行されます。

表 100. XML のラッパー・オプション (続き)

名前	説明
PROXY_TYPE	フェデレーテッド・サーバーがファイアウォールの背後にある場合に、インターネットへのアクセスに使用するプロキシー・タイプを指定します。有効な値は NONE、HTTP、および SOCKS です。デフォルト値は NONE です。
PROXY_SERVER_NAME	プロキシー・サーバーの名前または IP アドレスを指定します。有効な値は、1 から 32760 までの 10 進数のポート番号、またはサービス名です。IPv6 形式は、IPv6 を構成する場合のみ使用します。
PROXY_SERVER_PORT	プロキシー・サーバー上のプロキシー・サービスのポートまたはサービス名を指定します。有効な値は、10 進数の 1 から 32760 までのポート番号か、サービス名です。
SSL_KEYSTORE_FILE	SSL または TSL を使用する通信用の証明書ストレージ・ファイルを指定します。有効な値は、フェデレーテッド・データベース・エージェントまたは fenced モード・プロセスによってアクセス可能な絶対パス名です。
SSL_KEYSTORE_PASSWORD	SSL_KEYSTORE_FILE オプションで、ファイルにアクセスするために使用するパスワードを指定します。有効な値はパスワード (フェデレーテッド・データベース・カタログに保管されるときに暗号化される) および <code>file:file_name</code> (<code>file_name</code> は、スタッシュ・ファイルへの絶対パス) です。
SSL_VERIFY_SERVER_CERTIFICATE	SSL 認証の際にサーバー証明書を検証するかどうかを指定します。有効な値は Y と N です。デフォルトは N です。この場合、証明書は検証されません。

サーバー・オプション

表 101. XML のサーバー・オプション

名前	説明
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	照会からの並行非同期要求の最大数を指定します。有効な値は -1 から 64000 です。デフォルトは 1 です。-1 を指定すると、要求の数はフェデレーテッド照会オプティマイザーが決定します。0 は、データ・ソースが、追加の非同期要求に対応できないということ指定します。
PROXY_AUTHID	プロキシー・サーバー認証のユーザー名を指定します。

表 101. XML のサーバー・オプション (続き)

名前	説明
PROXY_PASSWORD	プロキシ・サーバー認証のパスワードを指定します。
PROXY_SERVER_NAME	プロキシ・サーバーの名前または IP アドレスを指定します。有効な値は、1 から 32760 までの 10 進数のポート番号、またはサービス名です。IPv6 形式は、IPv6 を構成する場合のみ使用します。
PROXY_SERVER_PORT	プロキシ・サーバー上のプロキシ・サービスのポートまたはサービス名を指定します。有効な値は、10 進数の 1 から 32760 までのポート番号か、サービス名です。
PROXY_TYPE	フェデレーテッド・サーバーがファイアウォールの背後にある場合に、インターネットへのアクセスに使用するプロキシ・タイプを指定します。有効な値は NONE、HTTP、および SOCKS です。デフォルト値は NONE です。
SOCKET_TIMEOUT	プロキシ・サーバーからの結果をフェデレーテッド・サーバーが待つ最大時間 (分単位) を指定します。有効な値は 0 以上の任意の数字です。デフォルトは 0 です。この場合、サーバーが待機する時間は無制限です。
SSL_CLIENT_CERTIFICATE_LABEL	SSL 認証の際に送信するクライアント証明書を指定します。値を指定しない場合は、フェデレーテッド・データベースの現在の許可 ID を使用して、証明書を見つけます。
SSL_KEYSTORE_FILE	SSL または TSL を使用する通信用の証明書ストレージ・ファイルを指定します。有効な値は、フェデレーテッド・データベース・エージェントまたは fenced モード・プロセスによってアクセス可能な絶対パス名です。
SSL_KEYSTORE_PASSWORD	SSL_KEYSTORE_FILE オプションで、ファイルにアクセスするために使用するパスワードを指定します。有効な値はパスワード (フェデレーテッド・データベース・カタログに保管されるときに暗号化される) および file:file_name (file_name は、スタッシュ・ファイルへの絶対パス) です。
SSL_VERIFY_SERVER_CERTIFICATE	SSL 認証の際にサーバー証明書を検証するかどうかを指定します。デフォルトは N です。この場合、証明書は検証されません。

ユーザー・マッピング・オプション

表 102. XML のユーザー・マッピング・オプション

名前	説明
PROXY_AUTHID	プロキシ・サーバー認証のユーザー名を指定します。
PROXY_PASSWORD	プロキシ・サーバー認証のパスワードを指定します。
SSL_CLIENT_CERTIFICATE_LABEL	SSL 認証の際に送信するクライアント証明書を指定します。値を指定しない場合は、フェデレーテッド・データベースの現在の許可 ID を使用して、証明書を見つけます。

ニックネーム・オプション

表 103. XML のニックネーム・オプション

名前	説明
DIRECTORY_PATH	1 つ以上の XML ファイルを含むディレクトリーのパス名を指定します。このオプションは、複数の XML ソース・ファイルに単一のニックネームを作成するために使用します。XML ラッパーは、指定するディレクトリー内にある、.xml 拡張子を持つファイルだけを使用します。XML ラッパーは、このディレクトリーにある他のすべてのファイルを無視します。このニックネーム・オプションを指定する場合、DOCUMENT 列を指定しないでください。このオプションは、ルート・ニックネームのみで有効です。
FILE_PATH	XML 文書のファイル・パスを指定します。FILE_PATH を指定する場合、DOCUMENT 列を指定しないでください。このオプションは、ルート・ニックネームのみで有効です。
INSTANCE_PARSE_TIME	XML ソース文書の 1 行を構文解析するのに必要な時間 (ミリ秒単位) を指定します。有効な値は、整数または 10 進数値です。デフォルトは 7 です。このオプションは、ルート・ニックネームの列のみで有効です。大規模または複雑な XML ソース構造の照会を最適化するには、INSTANCE_PARSE_TIME、XPATH_EVAL_TIME、および NEXT_TIME の各オプションを変更します。

表 103. XML のニックネーム・オプション (続き)

名前	説明
NAMESPACES	<p>各列の XPATH および TEMPLATE オプションで使用される、ネーム・スペース接頭部に関連付けられているネーム・スペースを指定します。次の構文を使用します。</p> <pre>NAMESPACES'prefix1="actual_namespace1", prefix2="actual_namespace2"'</pre> <p>複数のネーム・スペースを区切るには、コンマを使用します。例:</p> <pre>NAMESPACES='http://www.myweb.com/cust', i='http://www.myweb.com/cust/id', n='http://www.myweb.com/cust/name''</pre>
NEXT_TIME	<p>XPath 式から後続のソース・エレメントを位置指定するために必要な時間 (ミリ秒単位) を指定します。デフォルトは 1 です。このオプションはルート・ニックネームおよび非ルート・ニックネームで有効です。大規模または複雑な XML ソース構造の照会を最適化するには、INSTANCE_PARSE_TIME、XPATH_EVAL_TIME、 および NEXT_TIME の各オプションを変更します。</p>
STREAMING	<p>ソース文書进行处理するために、論理フラグメントに分ける必要があるかどうかを指定します。このフラグメントは、ニックネームの XPath 式に一致するノードに対応しています。このラッパーは、ソース・データの構文解析と処理をフラグメントごとに行います。このタイプの構文解析では、メモリーの使用量が最小限度で済みます。有効な値は Y と N です。デフォルトは N です。この場合、文書は構文解析されません。このオプションは、ルート・ニックネームのみで有効です。</p>
VALIDATE	<p>ソース文書からデータが抽出される前に、その文書が XML スキーマまたは文書タイプ定義 (DTD) に準拠しているかを検証するかどうかを指定します。デフォルトは N です。この場合、検証は行われません。この値を Y に設定する前に、ソース文書が指定する場所にスキーマ・ファイルまたは DTD ファイルを用意します。このオプションは、ルート・ニックネームのみで有効です。STREAMING と VALIDATE の両方のオプションを Y に設定することは避けてください。</p>
XPATH	<p>必須。個々のタプルを表すエレメントを識別する XPath 式を指定します。この XPath 式は、XPATH 列オプションによって識別される列値を評価するためのコンテキストとして使用されます。</p>

表 103. XML のニックネーム・オプション (続き)

名前	説明
XPATH_EVAL_TIME	ニックネームの XPath 式を評価して最初のエレメントを位置指定するために必要な時間 (ミリ秒単位) を指定します。可能な値は、整数または 10 進数値です。デフォルトは 1 です。このオプションはルート・ニックネームおよび非ルート・ニックネームで有効です。大規模または複雑な XML ソース構造の照会を最適化するには、INSTANCE_PARSE_TIME、XPATH_EVAL_TIME、 および NEXT_TIME の各オプションを変更します。

列オプション

表 104. XML の列オプション

名前	説明
DOCUMENT	この列が DOCUMENT 列であることを指定します。DOCUMENT 列の値は、照会の実行時にニックネームに提供される XML ソース・データのタイプを示します。このオプションは、ルート・ニックネーム (XML 文書の最上位にあるエレメントを識別するニックネーム) の列に対してのみ有効です。DOCUMENT オプションで指定できるのは、ニックネームあたり 1 つの列のみです。FILE_PATH または DIRECTORY_PATH ニックネーム・オプションの代わりに DOCUMENT 列オプションを使用する場合、このニックネームに対応する文書が照会の実行時に提供されます。有効な値は FILE、DIRECTORY、URI、および COLUMN です。デフォルトは FILE です。列は、VARCHAR データ・タイプでなければなりません。

表 104. XML の列オプション (続き)

名前	説明
FOREIGN_KEY	<p>このニックネームが子ニックネームであることを示し、対応する親ニックネームの名前を指定します。ニックネームには、FOREIGN_KEY 列オプションを 1 つしか指定できません。このオプションの値は、大文字小文字を区別します。この列には XPATH オプションを指定しないでください。列は、親のニックネームと子のニックネームを結合するためだけに使用できます。親ニックネームに別のスキーマ名が付けられている場合、FOREIGN_KEY オプションを含む CREATE NICKNAME ステートメントは失敗します。FOREIGN_KEY 文節で参照されるニックネームが CREATE NICKNAME ステートメントで小文字または大/小文字混合として明示的に定義されているのでない限り、このニックネームを FOREIGN_KEY 文節で参照するときは大文字で指定する必要があります。</p> <p>注:</p> <ul style="list-style-type: none"> このオプションが列に設定されている場合、その列に他のオプションを設定することはできません。 この列オプションを設定した場合は、後から ALTER NICKNAME ステートメントを使用してこのオプションをドロップすることはできません。代わりに、ニックネームをドロップしてから、この列オプションを付けずにニックネームを再作成する必要があります。

表 104. XML の列オプション (続き)

名前	説明
PRIMARY_KEY	<p>1 つ以上の子ニックネームを持つ親ニックネームの場合に必須です。このニックネームを親ニックネームとするということを指定します。列データ・タイプは VARCHAR(16) でなければなりません。ニックネームには、PRIMARY_KEY 列オプションを 1 つしか指定できません。Yes が唯一の有効な値です。この列には XPATH オプションを指定しないでください。列は、親のニックネームと子のニックネームを結合するためだけに使用できます。</p> <p>注:</p> <ul style="list-style-type: none"> このオプションが列に設定されている場合、その列に他のオプションを設定することはできません。 この列オプションを設定した場合は、後から ALTER NICKNAME ステートメントを使用してこのオプションをドロップすることはできません。代わりに、ニックネームをドロップしてから、この列オプションを付けずにニックネームを再作成する必要があります。
XPATH	<p>XML 文書の中でこの列に対応するデータを含む Xpath 式を指定します。CREATE NICKNAME ステートメントが XPATH ニックネーム・オプションから XPath 式を適用した後で、ラッパーはその XPath 式を評価します。</p> <p>注: この列オプションを設定した場合は、後から ALTER NICKNAME ステートメントを使用してこのオプションをドロップすることはできません。代わりに、ニックネームをドロップしてから、この列オプションを付けずにニックネームを再作成する必要があります。</p>

第 5 章 フェデレーテッド情報を含むグローバル・カタログ表内のビュー

フェデレーテッド・データベース内のカタログ・ビューのほとんどは、他の DB2 for Linux, UNIX, and Windows データベース内のカタログ・ビューと同じです。

いくつかの独自のビューとして、フェデレーテッド・システムに関係のある情報を含むビュー (SYSCAT.WRAPPERS ビューなど) があります。

SYSCAT ビューは、読み取り専用です。SYSCAT スキーマ内のビューに対して更新または挿入操作を発行することはできません。システム・カタログを更新する場合は SYSSTAT ビューを使用することをお勧めします。SYSCAT ビューを参照するアプリケーションは、SYSCAT ではなく更新可能な SYSSTAT ビューを参照するように変更してください。

次の表は、フェデレーテッド情報が含まれている SYSCAT ビューをリストしています。これらのビューは読み取り専用です。

表 105. 通常、フェデレーテッド・システムで使用されるカタログ・ビュー

カタログ・ビュー	説明
SYSCAT.CHECKS	定義したチェック制約の情報が含まれている。
SYSCAT.COLCHECKS	チェック制約によって参照される列が含まれている。
SYSCAT.COLUMNS	ニックネームを作成したデータ・ソース・オブジェクト (表およびビュー) の列情報が含まれている。
SYSCAT.COLOPTIONS	ニックネーム用にセットした列のオプション値の情報が含まれている。
SYSCAT.CONSTDEP	定義した情報制約の従属関係が含まれている。
SYSCAT.DATATYPES	ローカルの組み込み DB2 データ・タイプおよびユーザー定義 DB2 データ・タイプについてのデータ・タイプ情報が含まれている。
SYSCAT.DBAUTH	個々のユーザーおよびグループが保持するデータベース権限が含まれている。
SYSCAT.FUNCMAPOPTIONS	関数マッピング用にセットしたオプション値の情報が含まれている。
SYSCAT.FUNCMAPPINGS	フェデレーテッド・データベースとデータ・ソース・オブジェクト間の関数マッピングが含まれている。
SYSCAT.INDEXCOLUSE	索引に含める列が含まれている。
SYSCAT.INDEXES	データ・ソース・オブジェクトの索引仕様が含まれている。

表 105. 通常、フェデレーテッド・システムで使用されるカタログ・ビュー (続き)

カタログ・ビュー	説明
SYSCAT.INDEXOPTIONS	索引オプションについての情報が含まれている。
SYSCAT.KEYCOLUSE	ユニーク・キー、主キー、または外部キーの制約によって定義されるキーに加わる列が含まれている。
SYSCAT.NICKNAMES	作成したニックネームについての情報が含まれている。
SYSCAT.REFERENCES	定義した参照制約についての情報が含まれている。
SYSCAT.ROUTINES	ローカル DB2 のユーザー定義関数または関数テンプレートが含まれている。関数テンプレートは、データ・ソース関数と対応付けるために使用されます。
SYSCAT.REVTYPEMAPPINGS	このビューは使用されない。すべてのデータ・タイプ・マッピングは、SYSCAT.TYPEMAPPINGS ビューに記録されます。
SYSCAT.ROUTINEOPTIONS	フェデレーテッド・ルーチンのオプション値についての情報が含まれている。
SYSCAT.ROUTINEPARMOPTIONS	フェデレーテッド・ルーチンのパラメーター・オプション値についての情報が含まれている。
SYSCAT.ROUTINEPARMS	SYSCAT.ROUTINES で定義されているルーチンのパラメーターまたは結果が含まれている。
SYSCAT.ROUTINESFEDERATED	定義したフェデレーテッド・ルーチンについての情報が含まれている。
SYSCAT.SERVERS	データ・ソース・サーバー用に作成したサーバー定義が含まれている。
SYSCAT.TABCONST	各行は、タイプ CHECK、UNIQUE、PRIMARY KEY、または FOREIGN KEY の表およびニックネームの制約を表す。
SYSCAT.TABLES	作成したそれぞれのローカル DB2 表、フェデレーテッド・ビュー、およびニックネームについての情報が含まれている。
SYSCAT.TYPEMAPPINGS	フォワード・データ・タイプ・マッピングおよびリバース・データ・タイプ・マッピングが含まれている。マッピングは、データ・ソースのデータ・タイプからローカル DB2 データ・タイプへの対応付け。これらのマッピングは、データ・ソース・オブジェクトでニックネームを作成するときに使用されます。
SYSCAT.USEROPTIONS	フェデレーテッド・データベースとデータ・ソース・サーバー間にユーザー・マッピングを作成した時にセットした、ユーザー権限情報が含まれている。

表 105. 通常、フェデレーテッド・システムで使用されるカタログ・ビュー (続き)

カタログ・ビュー	説明
SYSCAT.VIEWS	作成したローカル・フェデレーテッド・ビューについての情報が含まれている。
SYSCAT.WRAPOPTIONS	ラッパーにセットしたオプション値についての情報が含まれている。
SYSCAT.WRAPPERS	ラッパーを作成したそれぞれのデータ・ソースの、ラッパーおよびライブラリー・ファイルの名前が含まれている。

次の表は、フェデレーテッド情報が含まれている **SYSSTAT** ビューをリストしています。これらのビューは読み取り/書き込みビューであり、そこに含まれている統計を更新することができます。

表 106. 更新可能なフェデレーテッド・グローバル・カタログ・ビュー

カタログ・ビュー	説明
SYSSTAT.COLUMNS	ニックネームを作成したデータ・ソース・オブジェクト (表およびビュー) の列についての統計情報が含まれている。タイプされた表の継承された列については統計は記録されません。
SYSSTAT.INDEXES	データ・ソース・オブジェクトの索引仕様についての統計情報が含まれている。
SYSSTAT.ROUTINES	ユーザー定義関数についての統計情報が含まれている。組み込み関数は含まれない。タイプされた表の継承された列については統計は記録されません。
SYSSTAT.TABLES	基本表についての情報が含まれている。このビューには、ビュー、シノニム、および別名の情報は含まれていません。タイプされた表の場合、表の階層のルート表だけがビューに含まれます。タイプされた表の継承された列については統計は記録されません。

第 6 章 フェデレーテッド・システムの関数マッピング・オプション

フェデレーテッド・サーバーは、DB2 関数とデータ・ソース関数の間のデフォルトのマッピングを提供します。ほとんどのデータ・ソースの場合、ラッパー内にデフォルトの関数マッピングがあります。フェデレーテッド・サーバーが認識しないデータ・ソース関数を使用したり、デフォルトのマッピングを変更したりする場合は、関数マッピングを作成する必要があります。

関数マッピングを作成する時、データ・ソース関数の名前を指定し、マップされる関数を使用可能にする必要があります。その後マップされた関数を使用する時、照会オプティマイザーは、その関数をデータ・ソースで実行する場合のコストとその関数をフェデレーテッド・サーバーで実行する場合のコストを比較します。

表 107. 関数マッピングのオプション

名前	説明
DISABLE	デフォルト関数マッピングを使用可能または使用不可にします。有効な値は Y と N です。デフォルトは N です。
REMOTE_NAME	データ・ソース関数の名前。デフォルトはローカル名です。

第 7 章 SQL ステートメントで有効なサーバーのタイプ

サーバー・タイプは、サーバー定義が表すデータ・ソースの種類を示します。

サーバー・タイプは、ベンダー、目的、およびオペレーティング・システムに応じて変わります。サポートされる値はデータ・ソースによって異なります。

ほとんどのデータ・ソースで、CREATE SERVER ステートメントに有効なサーバー・タイプを指定しなければなりません。

表 108. データ・ソースおよびサーバー・タイプ

データ・ソース	サーバー・タイプ
BioRS	サーバー・タイプは CREATE SERVER ステートメントでは必要ありません。
Excel	サーバー・タイプは CREATE SERVER ステートメントでは必要ありません。
IBM DB2 Universal Database™ for Linux, UNIX, and Windows	DB2/UDB
IBM DB2 Universal Database for System i and AS/400®	DB2/ISERIES
IBM DB2 Universal Database for z/OS	DB2/ZOS
IBM DB2 for VM	DB2/VM
Informix	INFORMIX
JDBC	JDBC (JDBC ドライバー 3.0 以降がサポートする JDBC データ・ソースでは必須。)
Microsoft SQL Server	MSSQLSERVER (DataDirect Connect ODBC 4.2 以降のドライバー、または Microsoft SQL Server ODBC 3.0 以降のドライバーがサポートするデータ・ソースでは必須。)
ODBC	ODBC (ODBC 3.x ドライバーがサポートする ODBC データ・ソースでは必須。)
OLE DB	サーバー・タイプは CREATE SERVER ステートメントでは必要ありません。
Oracle	ORACLE (Oracle NET8 クライアント・ソフトウェアがサポートする Oracle データ・ソースでは必須。)
Sybase (CTLIB)	SYBASE
表構造ファイル	サーバー・タイプは CREATE SERVER ステートメントでは必要ありません。
Teradata	TERADATA
Web サービス	サーバー・タイプは CREATE SERVER ステートメントでは必要ありません。
XML	サーバー・タイプは CREATE SERVER ステートメントでは必要ありません。

第 8 章 データ・タイプ・マッピング

リレーショナル・データ・ソースのデータ・タイプ・マッピングには、順方向タイプ・マッピング、逆方向タイプ・マッピング、および Unicode に固有のタイプ・マッピングがあります。非リレーショナル・データ・ソースは、それぞれ固有のデータ・タイプをサポートします。

デフォルトの順方向データ・タイプ・マッピング

データ・ソースのデータ・タイプとフェデレーテッド・データベースのデータ・タイプとの間のマッピングには、順方向データ・タイプ・マッピングと逆方向データ・タイプ・マッピングの 2 種類があります。順方向タイプ・マッピングとは、リモート・データ・タイプから対応するローカル・タイプへのマッピングのことです。

デフォルトのタイプ・マッピングをオーバーライドすることも、CREATE TYPE MAPPING ステートメントを使用して新規タイプ・マッピングを作成することもできます。

これらのマッピングは、特に注記のない限り、すべてのサポートされるバージョンで有効です。

データ・ソースからフェデレーテッド・データベースへの、すべてのデフォルトの順方向データ・タイプ・マッピングでは、フェデレーテッド・スキーマは SYSIBM です。

以下の表に、フェデレーテッド・データベースのデータ・タイプとデータ・ソースのデータ・タイプとの間の、デフォルトの順方向マッピングを示しています。

DB2 Database for Linux, UNIX, and Windows データ・ソースのデフォルトの順方向データ・タイプ・マッピング

以下の表に、DB2 Database for Linux, UNIX, and Windows データ・ソースのデフォルトの順方向データ・タイプ・マッピングをリストしています。

表 109. DB2 Database for Linux, UNIX, and Windows のデフォルトの順方向データ・タイプ・マッピング (一部の列は示されていません)

リモートの タイプ名	リモートの 長さの 下限	リモートの 長さの 上限	リモートの 位取り の下限	リモートの 位取り の上限	リモートの ビット・デー タ	リモートの データ演算 子	フェデレーテ ッドのタイプ 名	フェデレ ーテッド の長さ	フェデレ ーテッド の位取り	フェデレ ーテッド のビット ・デー タ
BIGINT	-	-	-	-	-	-	BIGINT	-	0	-
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHAR	-	-	-	-	-	-	CHAR	-	0	N
CHAR	-	-	-	-	Y	-	CHAR	-	0	Y
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	-	-	-	-	-	DATE	-	0	-
DATE	-	-	-	-	-	-	TIMESTAMP ¹	-	0	-

表 109. DB2 Database for Linux, UNIX, and Windows のデフォルトの順方向データ・タイプ・マッピング (一部の列は示されていません) (続き)

リモートのタイプ名	リモートの長さの下限	リモートの長さの上限	リモートの位取りの下限	リモートの位取りの上限	リモートのビット・データ	リモートのデータ演算子	フェデレーテッドのタイプ名	フェデレーテッドの長さ	フェデレーテッドの位取り	フェデレーテッドのビット・データ
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
DECFLOAT ₂	-	-	-	-	-	-	DECFLOAT	0	-	-
DOUBLE	-	-	-	-	-	-	DOUBLE	-	-	-
FLOAT	-	-	-	-	-	-	DOUBLE	-	-	-
GRAPHIC	-	-	-	-	-	-	GRAPHIC	0	N	-
INTEGER	-	-	-	-	-	-	INTEGER	0	-	-
LONGVAR	-	-	-	-	N	-	CLOB	-	-	-
LONGVAR	-	-	-	-	Y	-	BLOB	-	-	-
LONGVARG	-	-	-	-	-	-	DBCLOB	-	-	-
REAL	-	-	-	-	-	-	REAL	-	-	-
SMALLINT	-	-	-	-	-	-	SMALLINT	0	-	-
TIME	-	-	-	-	-	-	TIME	0	-	-
TIME-STAMP (p)	-	-	p	p	-	-	TIMESTAMP (p)	p	-	-
VARCHAR	-	-	-	-	-	-	VARCHAR	0	N	-
VARCHAR	-	-	-	-	Y	-	VARCHAR	0	Y	-
VARGRAPH	-	-	-	-	-	-	VARGRAPHIC	0	N	-
VAR GRAPHIC	-	-	-	-	-	-	VARGRAPHIC	0	N	-

注:

1. date_compat 構成パラメーターが ON に設定されている場合、フェデレーテッドのタイプは TIMESTAMP(0) になります。
2. SAME_DECFLT_ROUNDING サーバー・オプションはデフォルトでは N に設定され、SAME_DECFLT_ROUNDING が Y に設定されていないと操作はリモート・データ・ソースにプッシュダウンされません。SAME_DECFLT_ROUNDING サーバー・オプションについては、DB2 データベース・オプションの参照情報を参照してください。

DB2 for System i データ・ソースのデフォルトの順方向データ・タイプ・マッピング

以下の表に、DB2 for System i データ・ソースのデフォルトの順方向データ・タイプ・マッピングをリストしています。

表 110. DB2 for System i のデフォルトの順方向データ・タイプ・マッピング (一部の列は示されていません)

リモートのタイプ名	リモートの長さの下限	リモートの長さの上限	リモートの位取りの下限	リモートの位取りの上限	リモートのビット・データ	リモートのデータ演算子	フェデレーテッドのタイプ名	フェデレーテッドの長さ	フェデレーテッドの位取り	フェデレーテッドのビット・データ
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHAR	1	254	-	-	-	-	CHAR	0	N	-
CHAR	255	32672	-	-	-	-	VARCHAR	0	N	-

表 110. DB2 for System i のデフォルトの順方向データ・タイプ・マッピング (一部の列は示されていません) (続き)

リモートのタイプ名	リモートの長さの下限	リモートの長さの上限	リモートの位の取りの下限	リモートの位の取りの上限	リモートのビット・データ	リモートのデータ演算子	フェデレーテッドのタイプ名	フェデレーテッドの長さ	フェデレーテッドの位取り	フェデレーテッドのビット・データ
CHAR	1	254	-	-	Y	-	CHAR	-	0	Y
CHAR	255	32672	-	-	Y	-	VARCHAR	-	0	Y
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	-	-	-	-	-	DATE	-	0	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
FLOAT	4	-	-	-	-	-	REAL	-	-	-
FLOAT	8	-	-	-	-	-	DOUBLE	-	-	-
GRAPHIC	1	127	-	-	-	-	GRAPHIC	-	0	N
GRAPHIC	128	16336	-	-	-	-	VARGRAPHIC	-	0	N
INTEGER	-	-	-	-	-	-	INTEGER	-	0	-
NUMERIC	-	-	-	-	-	-	DECIMAL	-	-	-
SMALLINT	-	-	-	-	-	-	SMALLINT	-	0	-
TIME	-	-	-	-	-	-	TIME	-	0	-
TIMESTAMP	-	-	-	-	-	-	TIMESTAMP(6)	-	6	-
VARCHAR	1	32672	-	-	-	-	VARCHAR	-	0	N
VARCHAR	1	32672	-	-	Y	-	VARCHAR	-	0	Y
VARG	1	16336	-	-	-	-	VARGRAPHIC	-	0	N
VARGRAPHIC	1	16336	-	-	-	-	VARGRAPHIC	-	0	N

DB2 for VM and VSE データ・ソースのデフォルトの順方向データ・タイプ・マッピング

以下の表に、DB2 for VM and VSE データ・ソースのデフォルトの順方向データ・タイプ・マッピングをリストしています。

表 111. DB2 Server for VM and VSE のデフォルトの順方向データ・タイプ・マッピング (一部の列は示されていません)

リモートのタイプ名	リモートの長さの下限	リモートの長さの上限	リモートの位の取りの下限	リモートの位の取りの上限	リモートのビット・データ	リモートのデータ演算子	フェデレーテッドのタイプ名	フェデレーテッドの長さ	フェデレーテッドの位取り	フェデレーテッドのビット・データ
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHAR	1	254	-	-	-	-	CHAR	-	0	N
CHAR	1	254	-	-	Y	-	CHAR	-	0	Y
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	-	-	-	-	-	DATE	-	0	-
DBAHW	-	-	-	-	-	-	SMALLINT	-	0	-
DBAINT	-	-	-	-	-	-	INTEGER	-	0	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
FLOAT	4	-	-	-	-	-	REAL	-	-	-
FLOAT	8	-	-	-	-	-	DOUBLE	-	-	-

表 111. DB2 Server for VM and VSE のデフォルトの順方向データ・タイプ・マッピング (一部の列は示されていません) (続き)

リモートのタイプ名	リモートの長さの下限	リモートの長さの上限	リモートの位の取りの下限	リモートの位の取りの上限	リモートのビット・データ	リモートのデータ演算子	フェデレーテッドのタイプ名	フェデレーテッドの長さ	フェデレーテッドの位取り	フェデレーテッドのビット・データ
GRAPHIC	1	127	-	-	-	-	GRAPHIC	-	0	N
INTEGER	-	-	-	-	-	-	INTEGER	-	-	-
SMALLINT	-	-	-	-	-	-	SMALLINT	-	-	-
TIME	-	-	-	-	-	-	TIME	-	0	-
TIMESTAMP	-	-	-	-	-	-	TIMESTAMP(6)	-	6	-
VARCHAR	1	32672	-	-	-	-	VARCHAR	-	0	N
VARCHAR	1	32672	-	-	Y	-	VARCHAR	-	0	Y
VARGRAPHIC	1	16336	-	-	-	-	VARGRAPHIC	-	0	N
VARGRAPH	1	16336	-	-	-	-	VARGRAPHIC	-	0	N

DB2 for z/OS データ・ソースのデフォルトの順方向データ・タイプ・マッピング

以下の表に、DB2 for z/OS データ・ソースのデフォルトの順方向データ・タイプ・マッピングをリストしています。

表 112. DB2 for z/OS のデフォルトの順方向データ・タイプ・マッピング (一部の列は示されていません)

リモートのタイプ名	リモートの長さの下限	リモートの長さの上限	リモートの位の取りの下限	リモートの位の取りの上限	リモートのビット・データ	リモートのデータ演算子	フェデレーテッドのタイプ名	フェデレーテッドの長さ	フェデレーテッドの位取り	フェデレーテッドのビット・データ
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHAR	1	254	-	-	-	-	CHAR	-	0	N
CHAR	255	32672	-	-	-	-	VARCHAR	-	0	N
CHAR	1	254	-	-	Y	-	CHAR	-	0	Y
CHAR	255	32672	-	-	Y	-	VARCHAR	-	0	Y
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	-	-	-	-	-	DATE	-	0	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
FLOAT	4	-	-	-	-	-	REAL	-	-	-
FLOAT	8	-	-	-	-	-	DOUBLE	-	-	-
GRAPHIC	1	127	-	-	-	-	GRAPHIC	-	0	N
INTEGER	-	-	-	-	-	-	INTEGER	-	0	-
ROWID	-	-	-	-	Y	-	VARCHAR	40	-	Y
SMALLINT	-	-	-	-	-	-	SMALLINT	-	0	-
TIME	-	-	-	-	-	-	TIME	-	0	-
TIMESTAMP	-	-	-	-	-	-	TIME-STAMP(6)	-	6	-
VARCHAR	1	32672	-	-	-	-	VARCHAR	-	0	N
VARCHAR	1	32672	-	-	Y	-	VARCHAR	-	0	Y
VARG	1	16336	-	-	-	-	VARGRAPHIC	-	0	N
VARGRAPHIC	1	16336	-	-	-	-	VARGRAPHIC	-	0	N

Informix データ・ソースのデフォルトの順方向データ・タイプ・マッピング

以下の表に、Informix データ・ソースのデフォルトの順方向データ・タイプ・マッピングをリストしています。

表 113. Informix のデフォルトの順方向データ・タイプ・マッピング (一部の列は示されていません)

リモートのタイプ名	リモートの長さの下限	リモートの長さの上限	リモートの位の取り下限	リモートの位の取り上限	リモートのビット・データ	リモートのデータ演算子	フェデレーテッドのタイプ名	フェデレーテッドの長さ	フェデレーテッドの位の取り	フェデレーテッドのビット・データ
BLOB	-	-	-	-	-	-	BLOB	2147483647	-	-
BOOLEAN	-	-	-	-	-	-	CHARACTER	1	-	-
BYTE	-	-	-	-	-	-	BLOB	2147483647	-	-
CHAR	1	254	-	-	-	-	CHARACTER	-	-	-
CHAR	255	32672	-	-	-	-	VARCHAR	-	-	-
CLOB	-	-	-	-	-	-	CLOB	2147483647	-	-
DATE	-	-	-	-	-	-	DATE	4	-	-
DATE	-	-	-	-	-	-	TIMESTAMP ¹	-	0	-
DATETIME ²	0	4	0	4	-	-	DATE	4	-	-
DATETIME	6	10	6	10	-	-	TIME	3	-	-
DATETIME	0	4	6	15	-	-	TIMESTAMP(6)	10	6	-
DATETIME	6	10	11	15	-	-	TIMESTAMP(6)	10	6	-
DECIMAL	1	31	0	31	-	-	DECIMAL	-	-	-
DECIMAL	32	130	-	-	-	-	DOUBLE	8	-	-
DECIMAL	1	32	255	255	-	-	DOUBLE	-	-	-
FLOAT	-	-	-	-	-	-	DOUBLE	8	-	-
INTEGER	-	-	-	-	-	-	INTEGER	4	-	-
INTERVAL	-	-	-	-	-	-	VARCHAR	25	-	-
INT8	-	-	-	-	-	-	BIGINT	19	0	-
LVARCHAR	1	32672	-	-	-	-	VARCHAR	-	-	-
MONEY	1	31	0	31	-	-	DECIMAL	-	-	-
MONEY	32	32	-	-	-	-	DOUBLE	8	-	-
NCHAR	1	254	-	-	-	-	CHARACTER	-	-	-
NCHAR	255	32672	-	-	-	-	VARCHAR	-	-	-
NVARCHAR	1	32672	-	-	-	-	VARCHAR	-	-	-
REAL	-	-	-	-	-	-	REAL	4	-	-
SERIAL	-	-	-	-	-	-	INTEGER	4	-	-
SERIAL8	-	-	-	-	-	-	BIGINT	-	-	-
SMALLFLOAT	-	-	-	-	-	-	REAL	4	-	-
SMALLINT	-	-	-	-	-	-	SMALLINT	2	-	-
TEXT	-	-	-	-	-	-	CLOB	2147483647	-	-
VARCHAR	1	32672	-	-	-	-	VARCHAR	-	-	-

表 113. Informix のデフォルトの順方向データ・タイプ・マッピング (一部の列は示されていません) (続き)

リモートのタイプ名	リモートの長さの下限	リモートの長さの上限	リモートの位の取り下限	リモートの位の取り上限	リモートのビット・データ	リモートのデータ演算子	フェデレーテッドのタイプ名	フェデレーテッドの長さ	フェデレーテッドの位の取り	フェデレーテッドのビット・データ
-----------	------------	------------	-------------	-------------	--------------	-------------	---------------	-------------	---------------	------------------

注:

1. date_compat 構成パラメーターが ON に設定されている場合、フェデレーテッドのタイプは TIMESTAMP(0) になります。
2. Informix DATETIME データ・タイプについて、DB2 UNIX and Windows フェデレーテッド・サーバーは Informix 高位修飾子を REMOTE_LENGTH として、Informix 低位修飾子を REMOTE_SCALE として使用します。

Informix 修飾子は、Informix クライアント SDK の datatype.h ファイルで定義された「TU_」定数です。この定数は、以下のようになります。

0 = YEAR	8 = MINUTE	13 = FRACTION(3)
2 = MONTH	10 = SECOND	14 = FRACTION(4)
4 = DAY	11 = FRACTION(1)	15 = FRACTION(5)
6 = HOUR	12 = FRACTION(2)	

JDBC データ・ソースのデフォルトの順方向データ・タイプ・マッピング

以下の表に、JDBC データ・ソースのデフォルトの順方向データ・タイプ・マッピングをリストしています。

表 114. JDBC のデフォルトの順方向データ・タイプ・マッピング

リモートのタイプ名	リモートの長さの下限	リモートの長さの上限	リモートの位の取り下限	リモートの位の取り上限	リモートのビット・データ	リモートのデータ演算子	フェデレーテッドのタイプ名	フェデレーテッドの長さ	フェデレーテッドの位の取り	フェデレーテッドのビット・データ
BIGINT	-	-	-	-	-	-	BIGINT	8	-	-
BINARY	-	254	-	-	-	-	CHAR	-	-	Y
BINARY	255	32672	-	-	-	-	VARCHAR	-	-	Y
BINARY	32673	2147483647	-	-	-	-	BLOB	2147483647	-	-
BIT	-	-	-	-	-	-	SMALLINT	2	-	-
BLOB	-	-	-	-	-	-	BLOB	2147483647	-	-
BOOLEAN-	-	-	-	-	-	-	SMALLINT	2	-	-
CHAR	-	254	-	-	-	-	CHAR	-	-	-
CHAR	255	32672	-	-	-	-	VARCHAR	-	-	-
CHAR	32673	2147483647	-	-	-	-	CLOB	2147483647	-	-
CLOB	-	-	-	-	-	-	CLOB	2147483647	-	-
DATE	-	-	-	-	-	-	DATE	-	-	-
DATE	-	-	-	-	-	-	TIME-STAMP ¹	-	-	-
DECIMAL	1	31	0	31	-	-	DECIMAL	-	-	-
DECIMAL	32	38	0	38	-	-	DOUBLE	8	-	-
DOUBLE	-	-	-	-	-	-	DOUBLE	8	-	-

表 114. JDBC のデフォルトの順方向データ・タイプ・マッピング (続き)

リモートのタイプ名	リモートの長さの下限	リモートの長さの上限	リモートの位の下限	リモートの位の上限	リモートのビット・データ	リモートのデータ演算子	フェデレーテッドのタイプ名	フェデレーテッドの長さ	フェデレーテッドの位取り	フェデレーテッドのビット・データ
FLOAT	-	-	-	-	-	-	FLOAT	4	-	-
INTEGER	-	-	-	-	-	-	INTEGER	4	-	-
LONGVARCHAR	-	32672	-	-	-	-	VARCHAR	-	-	-
LONGVARCHAR	32673	2147483 647	-	-	-	-	CLOB	2147483 647	-	-
LONGVARBINARY	-	32672	-	-	-	-	VARCHAR	-	-	Y
LONGVARBINARY	32673	2147483 647	-	-	-	-	BLOB	2147483 647	-	-
LONGNVARCHAR	-	16336	-	-	-	-	VARGRAPHIC	-	-	-
LONGNVARCHAR ²	16337	1073741 823	-	-	-	-	DBCLOB	-	-	-
NCHAR ²	-	127	-	-	-	-	GRAPHIC	-	-	-
NCHAR ²	128	16336	-	-	-	-	VARGRAPHIC	-	-	-
NCHAR ²	16337	1073741 823	-	-	-	-	DBCLOB	-	-	-
NCLOB ²	-	-	-	-	-	-	DBCLOB	-	-	-
NUMERIC	1	31	0	31	-	-	DECIMAL	-	-	-
NUMERIC	32	38	0	38	-	-	DOUBLE	8	-	-
NVARCHAR ²	-	16336	-	-	-	-	VARGRAPHIC	-	-	-
NVARCHAR ²	16337	1073741 823	-	-	-	-	DBCLOB	-	-	-
REAL	-	-	-	-	-	-	REAL	4	-	-
SMALLINT	-	-	-	-	-	-	SMALLINT	2	-	-
TIME	-	-	-	-	-	-	TIME	3	-	-
TIMESTAMP	-	-	-	-	-	-	TIMESTAMP (6)	10	6	-
TIMESTAMP(p)	-	-	-	-	-	-	TIMESTAMP (6)	10	6	-
TINYINT	-	-	-	-	-	-	SMALLINT	2	-	-
VARBINARY	-	32672	-	-	-	-	VARCHAR	-	-	Y
VARBINARY	32673	2147483 647	-	-	-	-	BLOB	2147483 647	-	-
VARCHAR	-	32672	-	-	-	-	VARCHAR	-	-	-
VARCHAR	32673	2147483 647	-	-	-	-	CLOB	2147483 647	-	-

注:

1. date_compat 構成パラメーターが ON に設定されている場合、フェデレーテッドのタイプは TIMESTAMP(0) になります。
2. JDBC 4.0 ドライバーでのみサポートされるデータ・タイプは、NCHAR、NVARCHAR、LONGVARCHAR、および NCLOB です。

データ・タイプ

DATALINK、OTHER、JAVA_OBJECT、DISTINCT、STRUCT、ARRAY、および REF は JDBC ラッパーでサポートされません。

Microsoft SQL Server データ・ソースのデフォルトの順方向データ・タイプ・マッピング

以下の表に、Microsoft SQL Server データ・ソースのデフォルトの順方向データ・タイプ・マッピングをリストしています。

表 115. Microsoft SQL Server のデフォルトの順方向データ・タイプ・マッピング

リモートのタイプ名	リモートの長さの下限	リモートの長さの上限	リモートの位の取り下限	リモートの位の取り上限	リモートのビット・演算子データ	フェデレーテッドのタイプ名	フェデレーテッドの長さ	フェデレーテッドの位取り	フェデレーテッドのビット・データ
bigint ¹	-	-	-	-	-	BIGINT	-	-	-
binary	1	254	-	-	-	CHARACTER	-	-	Y
binary	255	8000	-	-	-	VARCHAR	-	-	Y
bit	-	-	-	-	-	SMALLINT	2	-	-
char	1	254	-	-	-	CHAR	-	-	N
char	255	8000	-	-	-	VARCHAR	-	-	N
datetime	-	-	-	-	-	TIMESTAMP(6)	10	6	-
decimal	1	31	0	31	-	DECIMAL	-	-	-
decimal	32	38	0	38	-	DOUBLE	-	-	-
float	-	8	-	-	-	DOUBLE	8	-	-
float	-	4	-	-	-	REAL	4	-	-
image	-	-	-	-	-	BLOB	2147483647	-	Y
int	-	-	-	-	-	INTEGER	4	-	-
money	-	-	-	-	-	DECIMAL	19	4	-
nchar	1	127	-	-	-	CHAR	-	-	N
nchar	128	4000	-	-	-	VARCHAR	-	-	N
numeric	1	31	0	31	-	DECIMAL	-	-	-
numeric	32	38	0	38	-	DOUBLE	8	-	-
ntext	-	-	-	-	-	CLOB	2147483647	-	Y
nvarchar	1	4000	-	-	-	VARCHAR	-	-	N
real	-	-	-	-	-	REAL	4	-	-
smallint	-	-	-	-	-	SMALLINT	2	-	-
smalldatetime	-	-	-	-	-	TIMESTAMP(6)	10	6	-
smallmoney	-	-	-	-	-	DECIMAL	10	4	-
SQL_BIGINT	-	-	-	-	-	BIGINT	-	-	-
SQL_BINARY	1	254	-	-	-	CHARACTER	-	-	Y
SQL_BINARY	255	8000	-	-	-	VARCHAR	-	-	Y
SQL_BIT	-	-	-	-	-	SMALLINT	2	-	-
SQL_CHAR	1	254	-	-	-	CHAR	-	-	N
SQL_CHAR	255	8000	-	-	-	VARCHAR	-	-	N
SQL_DATE	-	-	-	-	-	DATE	4	-	-
SQL_DECIMAL	1	31	0	31	-	DECIMAL	-	-	-
SQL_DECIMAL	32	38	0	38	-	DOUBLE	8	-	-
SQL_DOUBLE	-	-	-	-	-	DOUBLE	8	-	-
SQL_FLOAT	-	-	-	-	-	DOUBLE	8	-	-

表 115. Microsoft SQL Server のデフォルトの順方向データ・タイプ・マッピング (続き)

リモートのタイプ名	リモートの長さの下限	リモートの長さの上限	リモートの位の下限	リモートの位の上限	リモートのビット・データ	リモートのデータの演算子	フェデレーテッドのタイプ名	フェデレーテッドの長さ	フェデレーテッドの位取り	フェデレーテッドのビット・データ
SQL_GUID	-	-	-	-	-	-	VARCHAR	-	-	Y
SQL_INTEGER	-	-	-	-	-	-	INTEGER	4	-	-
SQL_LONGVARCHAR	-	-	-	-	-	-	CLOB	2147483647	-	N
SQL_LONGVARBINARY	-	-	-	-	-	-	BLOB	-	-	Y
SQL_NUMERIC	1	31	0	31	-	-	DECIMAL	-	-	-
SQL_NUMERIC	32	38	0	38	-	-	DOUBLE	8	-	-
SQL_REAL	-	-	-	-	-	-	REAL	8	-	-
SQL_SMALLINT	-	-	-	-	-	-	SMALLINT	2	-	-
SQL_TIME	-	-	-	-	-	-	TIME	3	-	-
SQL_TIMESTAMP	-	-	-	-	-	-	TIMESTAMP	10	6	-
SQL_TINYINT	-	-	-	-	-	-	SMALLINT	2	-	-
SQL_VARBINARY	1	8000	-	-	-	-	VARCHAR	-	-	Y
SQL_VARCHAR	1	8000	-	-	-	-	VARCHAR	-	-	N
SQL_WCHAR	1	254	-	-	-	-	CHARACTER	-	-	N
SQL_WCHAR	255	8800	-	-	-	-	VARCHAR	-	-	N
SQL_WLONGVARCHAR	-	1073741823	-	-	-	-	CLOB	2147483647	-	N
SQL_WVARCHAR	1	16336	-	-	-	-	VARCHAR	-	-	N
text	-	-	-	-	-	-	CLOB	-	-	N
timestamp	-	-	-	-	-	-	VARCHAR	8	-	Y
tinyint	-	-	-	-	-	-	SMALLINT	2	-	-
uniqueidentifier	1	4000	-	-	Y	-	VARCHAR	16	-	Y
varbinary	1	8000	-	-	-	-	VARCHAR	-	-	Y
varchar	1	8000	-	-	-	-	VARCHAR	-	-	N

注:

1. このタイプ・マッピングは、Microsoft SQL Server バージョン 2000 でのみ有効です。

ODBC データ・ソースのデフォルトの順方向データ・タイプ・マッピング

以下の表に、ODBC データ・ソースのデフォルトの順方向データ・タイプ・マッピングをリストしています。

表 116. ODBC のデフォルトの順方向データ・タイプ・マッピング (一部の列は示されていません)

リモートのタイプ名	リモートの長さの下限	リモートの長さの上限	リモートの位の下限	リモートの位の上限	リモートのビット・データ	リモートのデータの演算子	フェデレーテッドのタイプ名	フェデレーテッドの長さ	フェデレーテッドの位取り	フェデレーテッドのビット・データ
SQL_BIGINT	-	-	-	-	-	-	BIGINT	8	-	-

表 116. ODBC のデフォルトの順方向データ・タイプ・マッピング (一部の列は示されていません) (続き)

リモートのタイプ名	リモートの長さの下限	リモートの長さの上限	リモートの位の取りの下限	リモートの位の取りの上限	リモートのビット・データ	リモートのデータの演算子	フェデレーテッドのタイプ名	フェデレーテッドの長さ	フェデレーテッドの位の取り	フェデレーテッドのビット・データ
SQL_BINARY	1	254	-	-	-	-	CHARACTER	-	-	Y
SQL_BINARY	255	32672	-	-	-	-	VARCHAR	-	-	Y
SQL_BIT	-	-	-	-	-	-	SMALLINT	2	-	-
SQL_CHAR	1	254	-	-	-	-	CHAR	-	-	N
SQL_CHAR	255	32672	-	-	-	-	VARCHAR	-	-	N
SQL_DATE	-	-	-	-	-	-	DATE	-	-	-
SQL_DATE	-	-	-	-	-	-	TIME-STAMP ¹	-	-	-
SQL_DECIMAL	1	31	0	31	-	-	DECIMAL	-	-	-
SQL_DECIMAL	32	38	0	38	-	-	DOUBLE	8	-	-
SQL_DOUBLE	-	-	-	-	-	-	DOUBLE	8	-	-
SQL_FLOAT	-	8	-	-	-	-	FLOAT	8	-	-
SQL_FLOAT	-	4	-	-	-	-	FLOAT	4	-	-
SQL_INTEGER	-	-	-	-	-	-	INTEGER	4	-	-
SQL_LONGVARCHAR	-	-	-	-	-	-	CLOB	2147483647	-	N
SQL_LONGVARBINARY	-	-	-	-	-	-	BLOB	2147483647	-	Y
SQL_NUMERIC	1	31	0	31	-	-	DECIMAL	-	-	-
SQL_NUMERIC	32	32	0	31	-	-	DOUBLE	8	-	-
SQL_REAL	-	-	-	-	-	-	REAL	4	-	-
SQL_SMALLINT	-	-	-	-	-	-	SMALLINT	2	-	-
SQL_TIMESTAMP	-	-	-	-	-	-	TIMESTAMP (6)	10	6	-
SQL_TIMESTAMP (p)	-	-	-	-	-	-	TIMESTAMP (6)	10	6	-
SQL_TYPE_DATE	-	-	-	-	-	-	DATE	4	-	-
SQL_TYPE_TIME	-	-	-	-	-	-	TIME	3	-	-
SQL_TYPE_TIMESTAMP	-	-	-	-	-	-	TIMESTAMP	10	-	-
SQL_TINYINT	-	-	-	-	-	-	SMALLINT	2	-	-
SQL_VARBINARY	1	32672	-	-	-	-	VARCHAR	-	-	Y
SQL_VARCHAR	1	32672	-	-	-	-	VARCHAR	-	-	N
SQL_WCHAR	1	127	-	-	-	-	CHAR	-	-	N
SQL_WCHAR	128	16336	-	-	-	-	VARCHAR	-	-	N
SQL_WVARCHAR	1	16336	-	-	-	-	VARCHAR	-	-	N
SQL_WLONGVARCHAR	-	1073741823	-	-	-	-	CLOB	2147483647	-	N

注:

1. date_compat 構成パラメーターが ON に設定されている場合、フェデレーテッドのタイプは TIMESTAMP(0) になります。

Oracle NET8 データ・ソースのデフォルトの順方向データ・タイプ・マッピング

以下の表に、Oracle NET8 データ・ソースのデフォルトの順方向データ・タイプ・マッピングをリストしています。

表 117. Oracle NET8 のデフォルトの順方向データ・タイプ・マッピング

リモートのタイプ名	リモートの長さの下限	リモートの長さの上限	リモートの位の取りの下限	リモートの位の取りの上限	リモートのビット・データ	リモートのデータ演算子	フェデレーテッドのタイプ名	フェデレーテッドの長さ	フェデレーテッドの位の取り	フェデレーテッドのビット・データ
BLOB	0	0	0	0	-	\0	BLOB	2147483647	0	Y
CHAR	1	254	0	0	-	\0	CHAR	0	0	N
CHAR	255	2000	0	0	-	\0	VARCHAR	0	0	N
CLOB	0	0	0	0	-	\0	CLOB	2147483647	0	N
DATE	0	0	0	0	-	\0	TIMESTAMP (6)	0	0	N
FLOAT	1	126	0	0	-	\0	DOUBLE	0	0	N
LONG	0	0	0	0	-	\0	CLOB	2147483647	0	N
LONG RAW	0	0	0	0	-	\0	BLOB	2147483647	0	Y
NUMBER	10	18	0	0	-	\0	BIGINT	0	0	N
NUMBER	1	38	-84	127	-	\0	DOUBLE	0	0	N
NUMBER	1	31	0	31	-	>=	DECIMAL	0	0	N
NUMBER	1	4	0	0	-	\0	SMALLINT	0	0	N
NUMBER	5	9	0	0	-	\0	INTEGER	0	0	N
NUMBER	-	10	0	0	-	\0	DECIMAL	0	0	N
RAW	1	2000	0	0	-	\0	VARCHAR	0	0	Y
ROWID	0	0	0	NULL	-	\0	CHAR	18	0	N
TIMESTAMP(p) ¹	-	-	-	-	-	\0	TIMESTAMP (6)	10	6	N
VARCHAR2	1	4000	0	0	-	\0	VARCHAR	0	0	N

注:

1.

- TIMESTAMP(p) は、可変の位取りが 0 から 9 までのタイム・スタンプを表します。Oracle タイム・スタンプの位取りは、デフォルトで TIMESTAMP(6) にマップされます。ユーザー定義のタイプ・マッピングを使用すると、このデフォルト・タイプ・マッピングを変更し、Oracle TIMESTAMP を同じ位取りのフェデレーテッド TIMESTAMP にマップすることができます。
- このタイプ・マッピングは、Oracle 9i 以降のクライアント/サーバー構成だけで有効です。

Sybase データ・ソースのデフォルトの順方向データ・タイプ・マッピング

以下の表に、Sybase データ・ソースのデフォルトの順方向データ・タイプ・マッピングをリストしています。

表 118. Sybase CTLIB のデフォルトの順方向データ・タイプ・マッピング

リモートのタイプ名	リモートの長さの下限	リモートの長さの上限	リモートの位の取りの下限	リモートの位の取りの上限	リモートのビット・データ	リモートのデータ演算子	フェデレーテッドのタイプ名	フェデレーテッドの長さ	フェデレーテッドの位の取り	フェデレーテッドのビット・データ
binary	1	254	-	-	-	-	CHAR	-	-	Y
binary	255	32672	-	-	-	-	VARCHAR	-	-	Y
bit	-	-	-	-	-	-	SMALLINT	-	-	-
char	1	254	-	-	-	-	CHAR	-	-	N
char	255	32672	-	-	-	-	VARCHAR	-	-	N
char null (varchar を参照)										
date	-	-	-	-	-	-	DATE	-	-	-
date	-	-	-	-	-	-	TIMESTAMP ¹	-	-	-
datetime	-	-	-	-	-	-	TIMESTAMP(6)	-	-	-
datetimn	-	-	-	-	-	-	TIMESTAMP	-	-	-
decimal	1	31	0	31	-	-	DECIMAL	-	-	-
decimal	32	38	0	38	-	-	DOUBLE	-	-	-
decimaln	1	31	0	31	-	-	DECIMAL	-	-	-
decimaln	32	38	0	38	-	-	DOUBLE	-	-	-
float	-	4	-	-	-	-	REAL	-	-	-
float	-	8	-	-	-	-	DOUBLE	-	-	-
floatn	-	4	-	-	-	-	REAL	-	-	-
floatn	-	8	-	-	-	-	DOUBLE	-	-	-
image	-	-	-	-	-	-	BLOB	-	-	-
int	-	-	-	-	-	-	INTEGER	-	-	-
intn	-	-	-	-	-	-	INTEGER	-	-	-
money	-	-	-	-	-	-	DECIMAL	19	4	-
moneyn	-	-	-	-	-	-	DECIMAL	19	4	-
nchar	1	254	-	-	-	-	CHAR	-	-	N
nchar	255	32672	-	-	-	-	VARCHAR	-	-	N
nchar null (nvarchar を参照)										
numeric	1	31	0	31	-	-	DECIMAL	-	-	-
numeric	32	38	0	38	-	-	DOUBLE	-	-	-
numericn	1	31	0	31	-	-	DECIMAL	-	-	-
numericn	32	38	0	38	-	-	DOUBLE	-	-	-
nvarchar	1	32672	-	-	-	-	VARCHAR	-	-	N
real	-	-	-	-	-	-	REAL	-	-	-
smalldatetime	-	-	-	-	-	-	TIMESTAMP(6)	-	-	-
smallint	-	-	-	-	-	-	SMALLINT	-	-	-
smallmoney	-	-	-	-	-	-	DECIMAL	10	4	-
sysname	-	-	-	-	-	-	VARCHAR	30	-	N
text	-	-	-	-	-	-	CLOB	-	-	-
time	-	-	-	-	-	-	TIME	-	-	-

表 118. Sybase CTLIB のデフォルトの順方向データ・タイプ・マッピング (続き)

リモートのタイプ名	リモートの長さの下限	リモートの長さの上限	リモートの位の取りの下限	リモートの位の取りの上限	リモートのビット・データ	リモートのデータ演算子	フェデレーテッドのタイプ名	フェデレーテッドの長さ	フェデレーテッドの位の取り	フェデレーテッドのビット・データ
timestamp	-	-	-	-	-	-	VARCHAR	8	-	Y
tinyint	-	-	-	-	-	-	SMALLINT	-	-	-
unichar ²	1	254	-	-	-	-	CHAR	-	-	N
unichar ²	255	32672	-	-	-	-	VARCHAR	-	-	N
unichar null (univarchar を参照)										
univarchar ²	1	32672	-	-	-	-	VARCHAR	-	-	N
varbinary	1	32672	-	-	-	-	VARCHAR	-	-	Y
varchar	1	32672	-	-	-	-	VARCHAR	-	-	N

注:

1. date_compat 構成パラメーターが ON に設定されている場合、フェデレーテッドのタイプは TIMESTAMP(0) になります。
2. 非 Unicode のフェデレーテッド・データベースに関して有効です。

Teradata データ・ソースのデフォルトの順方向データ・タイプ・マッピング

以下の表に、Teradata データ・ソースのデフォルトの順方向データ・タイプ・マッピングをリストしています。

表 119. Teradata のデフォルトの順方向データ・タイプ・マッピング (一部の列は示されていません)

リモートのタイプ名	リモートの長さの下限	リモートの長さの上限	リモートの位の取りの下限	リモートの位の取りの上限	リモートのビット・データ	リモートのデータ演算子	フェデレーテッドのタイプ名	フェデレーテッドの長さ	フェデレーテッドの位の取り	フェデレーテッドのビット・データ
BLOB	1	2097088000	-	-	-	-	BLOB	-	-	-
BYTE	1	254	-	-	-	-	CHAR	-	-	Y
BYTE	255	32672	-	-	-	-	VARCHAR	-	-	Y
BYTE	32673	64000	-	-	-	-	BLOB	-	-	-
BYTEINT	-	-	-	-	-	-	SMALLINT	-	-	-
CHAR	1	254	-	-	-	-	CHARACTER	-	-	-
CHAR	255	32672	-	-	-	-	VARCHAR	-	-	-
CHAR	32673	64000	-	-	-	-	CLOB	-	-	-
CLOB	1	2097088000 (Latin)	-	-	-	-	CLOB	-	-	-
CLOB	1	1048544000 (Unicode)	-	-	-	-	CLOB	-	-	-
DATE	-	-	-	-	-	-	DATE	-	-	-
DATE	-	-	-	-	-	-	TIMESTAMP ¹	-	-	-
DECIMAL	1	18	0	18	-	-	DECIMAL	-	-	-
DOUBLE PRECISION	-	-	-	-	-	-	DOUBLE	-	-	-
FLOAT	-	-	-	-	-	-	DOUBLE	-	-	-

表 119. Teradata のデフォルトの順方向データ・タイプ・マッピング (一部の列は示されていません) (続き)

リモートのタイプ名	リモートの長さの下限	リモートの長さの上限	リモートの位の取りの下限	リモートの位の取りの上限	リモートのビット・データ	リモートのデータ演算子	フェデレーテッドのタイプ名	フェデレーテッドの長さ	フェデレーテッドの位取り	フェデレーテッドのビット・データ
GRAPHIC	1	127	-	-	-	-	GRAPHIC	-	-	-
GRAPHIC	128	16336	-	-	-	-	VARGRAPHIC	-	-	-
GRAPHIC	16337	32000	-	-	-	-	DBCLOB	-	-	-
INTEGER	-	-	-	-	-	-	INTEGER	-	-	-
INTERVAL	-	-	-	-	-	-	CHAR	-	-	-
NUMERIC	1	18	0	18	-	-	DECIMAL	-	-	-
REAL	-	-	-	-	-	-	DOUBLE	-	-	-
SMALLINT	-	-	-	-	-	-	SMALLINT	-	-	-
TIME	0	21	0	21	-	-	TIME	-	-	-
TIME-STAMP(<i>p</i>)	-	-	<i>p</i>	<i>p</i>	-	-	TIMESTAMP(6)	10	6	-
VARBYTE	1	32762	-	-	-	-	VARCHAR	-	-	Y
VARBYTE	32763	64000	-	-	-	-	BLOB	-	-	-
VARCHAR	1	32672	-	-	-	-	VARCHAR	-	-	-
VARCHAR	32673	64000	-	-	-	-	CLOB	-	-	-
VARGRAPHIC	1	16336	-	-	-	-	VARGRAPHIC	-	-	-
VARGRAPHIC	16337	32000	-	-	-	-	DBCLOB	-	-	-

注:

1. date_compat 構成パラメーターが ON に設定されている場合、フェデレーテッドのタイプは TIMESTAMP(0) になります。

順方向データ・タイプ・マッピングのサンプル

順方向タイプ・マッピングのサンプルを使用して、精度を持つ TIMESTAMP データ・タイプのサポートを活用することができます。

Informix データ・ソースの場合、これらのタイプ・マッピングはニックネーム列タイプ、フェデレーテッド・プロシージャー・パラメーター、パススルー、およびフェデレーテッド・プロシージャー結果セットに対して使用されます。

Informix 以外のデータ・ソースの場合、これらのタイプ・マッピングはニックネーム列タイプとフェデレーテッド・プロシージャー・パラメーターに対するマッピングにのみ影響を与えます。パススルーおよびフェデレーテッド・プロシージャー結果セットには影響しません。

順方向データ・タイプ・マッピング - Informix のサンプル

フェデレーテッド・オブジェクトを作成する際に、Informix 用に提供されている順方向タイプ・マッピングのサンプルを使用できます。

フェデレーテッド・オブジェクトを作成する前に、以下のマッピングを作成する必要があります。

```
CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(0)
  TO SERVER TYPE informix REMOTE TYPE datetime(0,10);
```

```
CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(1)
  TO SERVER TYPE informix REMOTE TYPE datetime(0,11);
```

```

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(2)
  TO SERVER TYPE informix REMOTE TYPE datetime(0,12);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(3)
  TO SERVER TYPE informix REMOTE TYPE datetime(0,13);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(4)
  TO SERVER TYPE informix REMOTE TYPE datetime(0,14);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(5)
  TO SERVER TYPE informix REMOTE TYPE datetime(0,15);

```

順方向データ・タイプ・マッピング - Microsoft SQL Server のサンプル

フェデレーテッド・オブジェクトを作成する際に、Microsoft SQL Server 用に提供されている順方向タイプ・マッピングのサンプルを使用できます。

ニックネームまたはフェデレーテッド・プロシージャを作成する前に、以下のマッピングを作成する必要があります。

```

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(3)
  TO SERVER TYPE mssqlserver REMOTE TYPE "datetime";

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(0)
  TO SERVER TYPE mssqlserver REMOTE TYPE "smalldatetime";

```

順方向データ・タイプ・マッピング - Oracle のサンプル

フェデレーテッド・オブジェクトを作成する際に、Oracle 用に提供されている順方向タイプ・マッピングのサンプルを使用できます。

ニックネームまたはフェデレーテッド・プロシージャを作成する前に、以下のマッピングを作成する必要があります。

```

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(0)
  TO SERVER TYPE oracle REMOTE TYPE timestamp(0);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(1)
  TO SERVER TYPE oracle REMOTE TYPE timestamp(1);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(2)
  TO SERVER TYPE oracle REMOTE TYPE timestamp(2);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(3)
  TO SERVER TYPE oracle REMOTE TYPE timestamp(3);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(4)
  TO SERVER TYPE oracle REMOTE TYPE timestamp(4);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(5)
  TO SERVER TYPE oracle REMOTE TYPE timestamp(5);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(7)
  TO SERVER TYPE oracle REMOTE TYPE timestamp(7);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(8)
  TO SERVER TYPE oracle REMOTE TYPE timestamp(8);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(9)
  TO SERVER TYPE oracle REMOTE TYPE timestamp(9);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(0)
  TO SERVER TYPE oracle REMOTE TYPE date;

```

順方向データ・タイプ・マッピング - Sybase のサンプル

フェデレーテッド・オブジェクトを作成する際に、Sybase 用に提供されている順方向タイプ・マッピングのサンプルを使用できます。

ニックネームまたはフェデレーテッド・プロシージャを作成する前に、以下のマッピングを作成する必要があります。

```
CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(3)
  TO SERVER TYPE sybase REMOTE TYPE datetime);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(0)
  TO SERVER TYPE sybase REMOTE TYPE smalldatetime);
```

順方向データ・タイプ・マッピング - Teradata のサンプル

フェデレーテッド・オブジェクトを作成する際に、Teradata 用に提供されている順方向タイプ・マッピングのサンプルを使用できます。

ニックネームまたはフェデレーテッド・プロシージャを作成する前に、以下のマッピングを作成する必要があります。

```
CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(0)
  TO SERVER TYPE teradata REMOTE TYPE timestamp(0);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(1)
  TO SERVER TYPE teradata REMOTE TYPE timestamp(1);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(2)
  TO SERVER TYPE teradata REMOTE TYPE timestamp(2);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(3)
  TO SERVER TYPE teradata REMOTE TYPE timestamp(3);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(4)
  TO SERVER TYPE teradata REMOTE TYPE timestamp(4);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(5)
  TO SERVER TYPE teradata REMOTE TYPE timestamp(5);
```

デフォルトの逆方向データ・タイプ・マッピング

ほとんどのデータ・ソースの場合、ラッパー内にデフォルトのタイプ・マッピングがあります。

データ・ソースのデータ・タイプとフェデレーテッド・データベースのデータ・タイプの間には、順方向データ・タイプ・マッピングと逆方向データ・タイプ・マッピングの 2 種類があります。順方向タイプ・マッピングとは、リモート・データ・タイプから対応するローカル・タイプへのマッピングのことです。マッピングのもう 1 つのタイプは、逆方向タイプ・マッピングであり、透過 DDL でリモート表を作成または変更するために使用します。

DB2 ファミリー・データ・ソース用のデフォルトのタイプ・マッピングは、DRDA ラッパー内にあります。Informix 用のデフォルトのタイプ・マッピングは INFORMIX ラッパーにあります。その他のタイプ・マッピングについても同様です。

フェデレーテッド・データベースに対してリモートの表またはビューを定義する場合、その定義には逆方向タイプ・マッピングが含まれます。このマッピングは、各

列のローカルのフェデレーテッド・データベース・データ・タイプから、対応するリモートのデータ・タイプに対して行われます。例えば、ローカル・タイプ REAL が Informix タイプ SMALLFLOAT を指す、デフォルトの逆方向タイプ・マッピングがあります。

フェデレーテッド・データベースでは、LONG VARCHAR、LONG VARGRAPHIC、およびユーザー定義タイプへのマッピングはサポートされません。

CREATE TABLE ステートメントを使用してリモート表を作成する場合、そのリモート表に組み込みたいローカル・データ・タイプを指定します。これらのデフォルトの逆方向タイプ・マッピングは、これらの列に対応するリモート・タイプを割り当てます。例えば、CREATE TABLE ステートメントを使用して、列 C2 を持つ Informix 表を定義するとします。このステートメント内では、C2 のデータ・タイプとして BIGINT を指定します。BIGINT のデフォルトの逆方向タイプ・マッピングは、どのバージョンの Informix で表を作成しているかによって異なります。Informix 表内の C2 のマッピングは、Informix バージョン 8 の場合は DECIMAL に、Informix バージョン 9 の場合は INT8 に対して行われます。

デフォルトの逆方向タイプ・マッピングをオーバーライドすることも、CREATE TYPE MAPPING ステートメントを使用して新規の逆方向タイプ・マッピングを作成することもできます。

以下の表に、フェデレーテッド・データベースのローカル・データ・タイプとリモート・データ・ソースのデータ・タイプとの間の、デフォルトの逆方向マッピングを示しています。

これらのマッピングは、特に注記のない限り、すべてのサポートされるバージョンで有効です。

DB2 Database for Linux, UNIX, and Windows データ・ソースのデフォルトの逆方向データ・タイプ・マッピング

以下の表に、DB2 Database for Linux, UNIX, and Windows データ・ソースのデフォルトの逆方向データ・タイプ・マッピングをリストしています。

表 120. DB2 Database for Linux, UNIX, and Windows のデフォルトの逆方向データ・タイプ・マッピング (一部の列は示されていません)

フェデレーテッドのタイプ名	フェデレーテッドの長さの下限	フェデレーテッドの長さの上限	フェデレーテッドの位取りの下限	フェデレーテッドの位取りの上限	フェデレーテッドのビット・データ	フェデレーテッドのデータ演算子	リモートのタイプ名	リモートの長さ	リモートの位取り	フェデレーテッドのビット・データ
BIGINT	-	8	-	-	-	-	BIGINT	-	-	-
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHARACTER	-	-	-	-	-	-	CHAR	-	-	N
CHARACTER	-	-	-	-	Y	-	CHAR	-	-	Y
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE ¹	-	4	-	-	-	-	DATE	-	-	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-

表 120. DB2 Database for Linux, UNIX, and Windows のデフォルトの逆方向データ・タイプ・マッピング (一部の列は示されていません) (続き)

フェデレーテッドのタイプ名	フェデレーテッドの長さの下限	フェデレーテッドの長さの上限	フェデレーテッドの位取りの下限	フェデレーテッドの位取りの上限	フェデレーテッドのビット・データ	フェデレーテッドのデータ演算子	リモートのタイプ名	リモートの長さ	リモートの位取り	フェデレーテッドのビット・データ
DECFLOAT ²	-	8	-	-	-	-	DECFLOAT	-	0	-
DECFLOAT ²	-	16	-	-	-	-	DECFLOAT	-	0	-
DOUBLE	-	8	-	-	-	-	DOUBLE	-	-	-
FLOAT	-	8	-	-	-	-	DOUBLE	-	-	-
GRAPHIC	-	-	-	-	-	-	GRAPHIC	-	-	N
INTEGER	-	4	-	-	-	-	INTEGER	-	-	-
REAL	-	-	-	-	-	-	REAL	-	-	-
SMALLINT	-	2	-	-	-	-	SMALLINT	-	-	-
TIME	-	3	-	-	-	-	TIME	-	-	-
TIMESTAMP (p)	-	-	p	p	-	-	TIMESTAMP (p)	-	p ³	-
VARCHAR	-	-	-	-	-	-	VARCHAR	-	-	N
VARCHAR	-	-	-	-	Y	-	VARCHAR	-	-	Y
VARGRAPH	-	-	-	-	-	-	VARGRAPHIC	-	-	N
VARGRAPHIC	-	-	-	-	-	-	VARGRAPHIC	-	-	-

注:

1. date_compat パラメーターが OFF に設定されている場合、フェデレーテッドの DATE は TIMESTAMP(0) にマップされます。
2. SAME_DECFLT_ROUNDING サーバー・オプションはデフォルトでは N に設定され、SAME_DECFLT_ROUNDING が Y に設定されていないと操作はリモート・データ・ソースにプッシュダウンされません。SAME_DECFLT_ROUNDING サーバー・オプションについては、DB2 データベース・オプションの参照情報を参照してください。
3. バージョン 9.5 では、TIMESTAMP のリモートの位取りは 6 です。

DB2 for System i データ・ソースのデフォルトの逆方向データ・タイプ・マッピング

以下の表に、DB2 for System i データ・ソースのデフォルトの逆方向データ・タイプ・マッピングをリストしています。

表 121. DB2 for System i のデフォルトの逆方向データ・タイプ・マッピング (一部の列は示されていません)

フェデレーテッドのタイプ名	フェデレーテッドの長さの下限	フェデレーテッドの長さの上限	フェデレーテッドの位取りの下限	フェデレーテッドの位取りの上限	フェデレーテッドのビット・データ	フェデレーテッドのデータ演算子	リモートのタイプ名	リモートの長さ	リモートの位取り	リモートのビット・データ
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHARACTER	-	-	-	-	-	-	CHARACTER	-	-	N
CHARACTER	-	-	-	-	Y	-	CHARACTER	-	-	Y
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	4	-	-	-	-	DATE	-	-	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	NUMERIC	-	-	-

表 121. DB2 for System i のデフォルトの逆方向データ・タイプ・マッピング (一部の列は示されていません) (続き)

フェデレーテッドのタイプ名	フェデレーテッドの長さの下限	フェデレーテッドの長さの上限	フェデレーテッドの位取りの下限	フェデレーテッドの位取りの上限	フェデレーテッドのビット	フェデレーテッドのデータ演算子	リモートのタイプ名	リモートの長さ	リモートの位取り	リモートのビット
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
DOUBLE	-	8	-	-	-	-	FLOAT	-	-	-
GRAPHIC	-	-	-	-	-	-	GRAPHIC	-	-	N
INTEGER	-	4	-	-	-	-	INTEGER	-	-	-
REAL	-	4	-	-	-	-	FLOAT	-	-	-
SMALLINT	-	2	-	-	-	-	SMALLINT	-	-	-
TIME	-	3	-	-	-	-	TIME	-	-	-
TIMESTAMP (p)	-	-	p	p	-	-	TIMESTAMP	-	-	-
VARCHAR	-	-	-	-	-	-	VARCHAR	-	-	N
VARCHAR	-	-	-	-	Y	-	VARCHAR	-	-	Y
VARGRAPHIC	-	-	-	-	-	-	VARG	-	-	N

DB2 for VM and VSE データ・ソースのデフォルトの逆方向データ・タイプ・マッピング

以下の表に、DB2 for VM and VSE データ・ソースのデフォルトの逆方向データ・タイプ・マッピングをリストしています。

表 122. DB2 for VM and VSE のデフォルトの逆方向データ・タイプ・マッピング (一部の列は示されていません)

フェデレーテッドのタイプ名	フェデレーテッドの長さの下限	フェデレーテッドの長さの上限	フェデレーテッドの位取りの下限	フェデレーテッドの位取りの上限	フェデレーテッドのビット	フェデレーテッドのデータ演算子	リモートのタイプ名	リモートの長さ	リモートの位取り	リモートのビット
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHARACTER	-	-	-	-	-	-	CHAR	-	-	-
CHARACTER	-	-	-	-	Y	-	CHAR	-	-	Y
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	4	-	-	-	-	DATE	-	-	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
DOUBLE	-	8	-	-	-	-	FLOAT	-	-	-
GRAPHIC	-	-	-	-	-	-	GRAPHIC	-	-	N
INTEGER	-	4	-	-	-	-	INTEGER	-	-	-
REAL	-	4	-	-	-	-	REAL	-	-	-
SMALLINT	-	2	-	-	-	-	SMALLINT	-	-	-
TIME	-	3	-	-	-	-	TIME	-	-	-
TIMESTAMP (p)	-	-	p	p	-	-	TIMESTAMP	-	-	-
VARCHAR	-	-	-	-	-	-	VARCHAR	-	-	-
VARCHAR	-	-	-	-	Y	-	VARCHAR	-	-	Y
VARGRAPH	-	-	-	-	-	-	VARGRAPH	-	-	N

DB2 for z/OS データ・ソースのデフォルトの逆方向データ・タイプ・マッピング

以下の表に、DB2 for z/OS データ・ソースのデフォルトの逆方向データ・タイプ・マッピングをリストしています。

表 123. DB2 for z/OS のデフォルトの逆方向データ・タイプ・マッピング (一部の列は示されていません)

フェデレーテッドのタイプ名	フェデレーテッドの長さの下限	フェデレーテッドの長さの上限	フェデレーテッドの位取りの下限	フェデレーテッドの位取りの上限	フェデレーテッドのビット・データ	フェデレーテッドのデータ演算子	リモートのタイプ名	リモートの長さ	リモートの位取り	リモートのビット・データ
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHARACTER	-	-	-	-	-	-	CHAR	-	-	N
CHARACTER	-	-	-	-	Y	-	CHAR	-	-	Y
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	4	-	-	-	-	DATE	-	-	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
DOUBLE	-	8	-	-	-	-	DOUBLE	-	-	-
FLOAT	-	8	-	-	-	-	DOUBLE	-	-	-
GRAPHIC	-	-	-	-	-	-	GRAPHIC	-	-	N
INTEGER	-	4	-	-	-	-	INTEGER	-	-	-
REAL	-	4	-	-	-	-	REAL	-	-	-
SMALLINT	-	2	-	-	-	-	SMALLINT	-	-	-
TIME	-	3	-	-	-	-	TIME	-	-	-
TIME-STAMP(p)	-	-	p	p	-	-	TIMESTAMP	-	-	-
VARCHAR	-	-	-	-	-	-	VARCHAR	-	-	N
VARCHAR	-	-	-	-	Y	-	VARCHAR	-	-	Y
VARGRAPHIC	-	-	-	-	-	-	VARGRAPHIC	-	-	N

Informix データ・ソースのデフォルトの逆方向データ・タイプ・マッピング

以下の表に、Informix データ・ソースのデフォルトの逆方向データ・タイプ・マッピングをリストしています。

表 124. Informix のデフォルトの逆方向データ・タイプ・マッピング

フェデレーテッドのタイプ名	フェデレーテッドの長さの下限	フェデレーテッドの長さの上限	フェデレーテッドの位取りの下限	フェデレーテッドの位取りの上限	フェデレーテッドのビット・データ	フェデレーテッドのデータ演算子	リモートのタイプ名	リモートの長さ	リモートの位取り	リモートのビット・データ
BIGINT ¹	-	-	-	-	-	-	DECIMAL	19	-	-
BIGINT ²	-	-	-	-	-	-	INT8	-	-	-
BLOB	1	2147483647	-	-	-	-	BYTE	-	-	-
CHARACTER	-	-	-	-	N	-	CHAR	-	-	-

表 124. Informix のデフォルトの逆方向データ・タイプ・マッピング (続き)

フェデレー テッドのタ イプ名	フェデレ ーテッド の長さの 下限	フェデレ ーテッド の長さの 上限	フェデレ ーテッド の位取り の下限	フェデ レーテ ッドの 位取り の上限	フェデレ ーテッド のビット ト・デー タ	フェデレ ーテッド のデータ 演算子	リモートのタイ プ名	リモートの長 さ	リモートの位 取り	リモートのビ ット・デー タ
CHARAC- TER	-	-	-	-	Y	-	BYTE	-	-	-
CLOB	1	2147483647	-	-	-	-	TEXT	-	-	-
DATE	-	4	-	-	-	-	DATE	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
DOUBLE	-	8	-	-	-	-	FLOAT	-	-	-
INTEGER	-	4	-	-	-	-	INTEGER	-	-	-
REAL	-	4	-	-	-	-	SMALLFLOAT	-	-	-
SMALLINT	-	2	-	-	-	-	SMALLINT	-	-	-
TIME	-	3	-	-	-	-	DATETIME	6	10	-
TIMESTAMP	-	10	-	-	-	-	DATETIME	0	15	-
VARCHAR	1	254	-	-	N	-	VARCHAR	-	-	-
VARCHAR ¹	255	32672	-	-	N	-	TEXT	-	-	-
VARCHAR	-	-	-	-	Y	-	BYTE	-	-	-
VARCHAR ²	255	2048	-	-	N	-	LVARCHAR	-	-	-
VARCHAR ²	2049	32672	-	-	N	-	TEXT	-	-	-

注:

1. このタイプ・マッピングは、バージョン 8 以前の Informix サーバーでのみ有効です。
2. このタイプ・マッピングは、バージョン 9 以降の Informix サーバーでのみ有効です。

Informix DATETIME データ・タイプについて、フェデレーテッド・サーバーは Informix 高位修飾子を REMOTE_LENGTH として、Informix 低位修飾子を REMOTE_SCALE として使用します。

Informix 修飾子は、Informix クライアント SDK の datatime.h ファイルで定義された「TU_」定数です。この定数は、以下のようになります。

0 = YEAR	8 = MINUTE	13 = FRACTION(3)
2 = MONTH	10 = SECOND	14 = FRACTION(4)
4 = DAY	11 = FRACTION(1)	15 = FRACTION(5)
6 = HOUR	12 = FRACTION(2)	

JDBC データ・ソースのデフォルトの逆方向データ・タイプ・マッピング

以下の表に、DB2 JDBC ドライバーのタイプ・マッピングに準拠する JDBC データ・ソースのデフォルトの逆方向データ・タイプ・マッピングをリストしています。

表 125. JDBC のデフォルトの逆方向データ・タイプ・マッピング

フェデレーテッ ドのタイプ名	フェデレ ーテッド の長さの 下限	フェデレ ーテッド の長さの 上限	フェデレ ーテッド の位取り の下限	フェデレ ーテッド の位取り の上限	フェデレ ーテッド のビット ト・デー タ	フェデレ ーテッド のデータ 演算子	リモートの タイプ名	リモートの長 さ	リモートの位 取り	リモートのビ ット・デー タ
BIGINT	-	-	-	-	-	-	BIGINT	-	-	-

表 125. JDBC のデフォルトの逆方向データ・タイプ・マッピング (続き)

フェデレーテッドのタイプ名	フェデレーテッドの長さの下限	フェデレーテッドの長さの上限	フェデレーテッドの位取りの下限	フェデレーテッドの位取りの上限	フェデレーテッドのビット・データ	フェデレーテッドのデータ演算子	リモートのタイプ名	リモートの長さ	リモートの位取り	リモートのビット・データ
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHAR	-	-	-	-	Y	-	BINARY	-	-	-
CHAR	-	-	-	-	-	-	CHAR	-	-	-
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	4	-	-	-	-	DATE	-	-	-
DBCLOB	-	-	-	-	-	-	NCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
DOUBLE	-	8	-	-	-	-	DOUBLE	-	-	-
GRAPHIC	-	-	-	-	-	-	NCHAR	-	-	-
INTEGER	-	-	-	-	-	-	INTEGER	-	-	-
REAL	-	4	-	-	-	-	REAL	-	-	-
SMALLINT	-	-	-	-	-	-	SMALLINT	-	-	-
TIME	-	3	-	-	-	-	TIME	-	-	-
TIMESTAMP	-	-	-	-	-	-	TIME STAMP	-	-	-
TIMESTAMP(p)	-	-	p	p	-	-	TIMESTAMP(p)	-	min(9,p)	-
VARCHAR	-	-	-	-	Y	-	VAR BINARY	-	-	-
VARCHAR	-	-	-	-	N	-	VARCHAR	-	-	-
VARGRAPHIC	-	-	-	-	-	-	NVAR CHAR	-	-	-

Microsoft SQL Server データ・ソースのデフォルトの逆方向データ・タイプ・マッピング

以下の表には、Microsoft SQL Server データ・ソースのデフォルトの逆方向データ・タイプ・マッピングをリストしています。

表 126. Microsoft SQL Server のデフォルトの逆方向データ・タイプ・マッピング (一部の列は示されていません)

フェデレーテッドのタイプ名	フェデレーテッドの長さの下限	フェデレーテッドの長さの上限	フェデレーテッドの位取りの下限	フェデレーテッドの位取りの上限	フェデレーテッドのビット・データ	フェデレーテッドのデータ演算子	リモートのタイプ名	リモートの長さ	リモートの位取り	リモートのビット・データ
BIGINT ¹	-	-	-	-	-	-	bigint	-	-	-
BLOB	-	-	-	-	-	-	image	-	-	-
CHARACTER	-	-	-	-	Y	-	binary	-	-	-
CHARACTER	-	-	-	-	N	-	char	-	-	-
CLOB	-	-	-	-	-	-	text	-	-	-
DATE	-	4	-	-	-	-	datetime	-	-	-
DECIMAL	-	-	-	-	-	-	decimal	-	-	-
DOUBLE	-	8	-	-	-	-	float	-	-	-

表 126. Microsoft SQL Server のデフォルトの逆方向データ・タイプ・マッピング (一部の列は示されていません)
(続き)

フェデレーテッドのタイプ名	フェデレーテッドの長さの下限	フェデレーテッドの長さの上限	フェデレーテッドの位取りの下限	フェデレーテッドの位取りの上限	フェデレーテッドのビット・データ	フェデレーテッドのデータ演算子	リモートのタイプ名	リモートの長さ	リモートの位取り	リモートのビット・データ
INTEGER	-	-	-	-	-	-	int	-	-	-
SMALLINT	-	-	-	-	-	-	smallint	-	-	-
REAL	-	4	-	-	-	-	real	-	-	-
TIME	-	3	-	-	-	-	datetime	-	-	-
TIMESTAMP	-	10	-	-	-	-	datetime	-	-	-
VARCHAR	1	8000	-	-	N	-	varchar	-	-	-
VARCHAR	8001	32672	-	-	N	-	text	-	-	-
VARCHAR	1	8000	-	-	Y	-	varbinary	-	-	-
VARCHAR	8001	32672	-	-	Y	-	image	-	-	-

注:

1. このタイプ・マッピングは、Microsoft SQL Server バージョン 2000 でのみ有効です。

ODBC データ・ソースのデフォルトの逆方向データ・タイプ・マッピング

以下の表に、ODBC データ・ソースのデフォルトの逆方向データ・タイプ・マッピングをリストしています。

表 127. ODBC のデフォルトの逆方向データ・タイプ・マッピング (一部の列は示されていません)

フェデレーテッドのタイプ名	フェデレーテッドの長さの下限	フェデレーテッドの長さの上限	フェデレーテッドの位取りの下限	フェデレーテッドの位取りの上限	フェデレーテッドのビット・データ	フェデレーテッドのデータ演算子	リモートのタイプ名	リモートの長さ	リモートの位取り	リモートのビット・データ
BIGINT	-	-	-	-	-	-	SQL_ BIGINT	-	-	-
BLOB	-	-	-	-	-	-	SQL_ LONG VAR BINARY	-	-	-
CHAR	-	-	-	-	Y	-	SQL_ BINARY	-	-	-
CHAR	-	-	-	-	N	-	SQL_ CHAR	-	-	-
CLOB	-	-	-	-	-	-	SQL_ LONGVAR CHAR	-	-	-
DATE	-	4	-	-	-	-	SQL_ TYPE_ DATE	-	-	-
DBCLOB	-	-	-	-	-	-	SQL_ WLONG VAR CHAR	-	-	-

表 127. ODBC のデフォルトの逆方向データ・タイプ・マッピング (一部の列は示されていません) (続き)

フェデレーテッドのタイプ名	フェデレーテッドの長さの下限	フェデレーテッドの長さの上限	フェデレーテッドの位取りの下限	フェデレーテッドの位取りの上限	フェデレーテッドのビット・データ	フェデレーテッドのデータ演算子	リモートのタイプ名	リモートの長さ	リモートの位取り	リモートのビット・データ
DECIMAL	-	-	-	-	-	-	SQL_DECIMAL	-	-	-
DOUBLE	-	8	-	-	-	-	SQL_DOUBLE	-	-	-
FLOAT	-	-	-	-	-	-	SQL_FLOAT	-	-	-
GRAPHIC	-	-	-	-	-	-	SQL_WCHAR	-	-	-
INTEGER	-	-	-	-	-	-	SQL_INTEGER	-	-	-
REAL	-	4	-	-	-	-	SQL_REAL	-	-	-
SMALLINT	-	-	-	-	-	-	SQL_SMALLINT	-	-	-
TIME	-	3	-	-	-	-	SQL_TYPE_TIME	-	-	-
TIMESTAMP	-	10	-	-	-	-	SQL_TYPE_TIMESTAMP(p)	-	-	-
VARCHAR	-	-	-	-	Y	-	SQL_VAR_BINARY	-	-	-
VARCHAR	-	-	-	-	N	-	SQL_VAR_CHAR	-	-	-
VARGRAPHIC	-	-	-	-	Y	-	SQL_WVAR_CHAR	-	-	-

Oracle NET8 データ・ソースのデフォルトの逆方向データ・タイプ・マッピング

以下の表に、Oracle NET8 データ・ソースのデフォルトの逆方向データ・タイプ・マッピングをリストしています。

表 128. Oracle NET8 のデフォルトの逆方向データ・タイプ・マッピング

フェデレーテッドのタイプ名	フェデレーテッドの長さの下限	フェデレーテッドの長さの上限	フェデレーテッドの位取りの下限	フェデレーテッドの位取りの上限	フェデレーテッドのビット・データ	フェデレーテッドのデータ演算子	リモートのタイプ名	リモートの長さ	リモートの位取り	リモートのビット・データ
BIGINT	0	8	0	0	N	\0	NUMBER	19	0	N
BLOB	0	2147483647	0	0	Y	\0	BLOB	0	0	Y

表 128. Oracle NET8 のデフォルトの逆方向データ・タイプ・マッピング (続き)

フェデレーテッド のタイプ名	フェデレ ーテッド の長さの 下限	フェデレ ーテッド の長さの 上限	フェデレ ーテッド の位取り の下限	フェデレ ーテッド の位取り の上限	フェデレ ーテッド のビット ト・デー タ	フェデレ ーテッド のデータ 演算子	リモートのタ イプ名	リモー トの長 さ	リモー トの位 取り	リモートのビット ト・デー タ
CHARACTER	1	254	0	0	N	\0	CHAR	0	0	N
CHARACTER	1	254	0	0	Y	\0	RAW	0	0	Y
CLOB	0	2147483647	0	0	N	\0	CLOB	0	0	N
DATE ¹	0	4	0	0	N	\0	DATE	0	0	N
DECIMAL	0	0	0	0	N	\0	NUMBER	0	0	N
DECFLOAT	0	8	0	0	N	\0	NUMBER	0	0	N
DECFLOAT	0	16	0	0	N	\0	NUMBER	0	0	N
DOUBLE	0	8	0	0	N	\0	FLOAT	126	0	N
FLOAT	0	8	0	0	N	\0	FLOAT	126	0	N
INTEGER	0	4	0	0	N	\0	NUMBER	10	0	N
REAL	0	4	0	0	N	\0	FLOAT	63	0	N
SMALLINT	0	2	0	0	N	\0	NUMBER	5	0	N
TIME	0	3	0	0	N	\0	DATE	0	0	N
TIMESTAMP ²	0	10	0	0	N	\0	DATE	0	0	N
TIMESTAMP(p) ³	-	-	-	-	N	\0	TIMESTAMP (p)	-	-	N
VARCHAR	1	4000	0	0	N	\0	VARCHAR2	0	0	N
VARCHAR	1	2000	0	0	Y	\0	RAW	0	0	Y

注:

- date_compat パラメーターが OFF に設定されている場合、フェデレーテッドの DATE は Oracle の date にマップされます。date_compat パラメーターが ON に設定されている場合、フェデレーテッドの DATE (TIMESTAMP(0) と等価) は、Oracle の TIMESTAMP(0) にマップされます。
- このタイプ・マッピングは、Oracle バージョン 8 でのみ有効です。
- TIMESTAMP(p) は、Oracle の場合は 0 から 9 まで、フェデレーションの場合は 0 から 12 までの可変の位取りを持つタイム・スタンプを表します。位取りが 0 から 9 までである場合、リモートの Oracle TIMESTAMP の位取りは、フェデレーテッドの TIMESTAMP と同じになります。フェデレーテッドの TIMESTAMP の位取りが 9 より大きい場合、対応する Oracle TIMESTAMP の位取りは、Oracle の最大の位取りである 9 になります。
 - このタイプ・マッピングは、Oracle バージョン 9、10、および 11 でのみ有効です。

Sybase データ・ソースのデフォルトの逆方向データ・タイプ・マッピング

以下の表に、Sybase データ・ソースのデフォルトの逆方向データ・タイプ・マッピングをリストしています。

表 129. Sybase CTLIB のデフォルトの逆方向データ・タイプ・マッピング

フェデレーテッドの タイプ名	フェデレ ーテッド の長さの 下限	フェデレ ーテッド の長さの 上限	フェデレ ーテッド の位取り の下限	フェデ レーテ ッドの 位取り の上限	フェデレ ーテッド のビット ト・デー タ	フェデレ ーテッド のデータ 演算子	リモートの タイプ名	リモート の長さ	リモート の位取り	リモートのビット ト・デー タ
BIGINT	-	-	-	-	-	-	decimal	19	0	-

表 129. Sybase CTLIB のデフォルトの逆方向データ・タイプ・マッピング (続き)

フェデレーテッドの タイプ名	フェデレ ーテッド の長さの 下限	フェデレ ーテッド の長さの 上限	フェデレ ーテッド の位取り の下限	フェデ レーテ ッドの 位取り の上限	フェデレ ーテッド のビット ト・デー タ	フェデレ ーテッド のデータ 演算子	リモート のタイプ 名	リモート の長さ	リモート の位取り	リモート のビット ト・デー タ
BLOB	-	-	-	-	-	-	image	-	-	-
CHARACTER	-	-	-	-	N	-	char	-	-	-
CHARACTER	-	-	-	-	Y	-	binary	-	-	-
CLOB	-	-	-	-	-	-	text	-	-	-
DATE	-	-	-	-	-	-	datetime	-	-	-
DECIMAL	-	-	-	-	-	-	decimal	-	-	-
DOUBLE	-	-	-	-	-	-	float	-	-	-
INTEGER	-	-	-	-	-	-	integer	-	-	-
REAL	-	-	-	-	-	-	real	-	-	-
SMALLINT	-	-	-	-	-	-	smallint	-	-	-
TIME	-	-	-	-	-	-	datetime	-	-	-
TIMESTAMP	-	-	-	-	-	-	datetime	-	-	-
VARCHAR ¹	1	255	-	-	N	-	varchar	-	-	-
VARCHAR ¹	256	32672	-	-	N	-	text	-	-	-
VARCHAR ²	1	16384	-	-	N	-	varchar	-	-	-
VARCHAR ²	16385	32672	-	-	N	-	text	-	-	-
VARCHAR ¹	1	255	-	-	Y	-	varbinary	-	-	-
VARCHAR ¹	256	32672	-	-	Y	-	image	-	-	-
VARCHAR ²	1	16384	-	-	Y	-	varbinary	-	-	-
VARCHAR ²	16385	32672	-	-	Y	-	image	-	-	-

注:

1. このタイプ・マッピングは、Sybase サーバーのバージョン 12.0 以前の CTLIB に関してのみ有効です。
2. このタイプ・マッピングは、Sybase サーバーのバージョン 12.5 以降の CTLIB に関してのみ有効です。

Teradata データ・ソースのデフォルトの逆方向データ・タイプ・マッピング

以下の表に、Teradata データ・ソースのデフォルトの逆方向データ・タイプ・マッピングをリストしています。

表 130. Teradata のデフォルトの逆方向データ・タイプ・マッピング (一部の列は示されていません)

フェデレーテッ ドのタイプ名	フェデレ ーテッド の長さの 下限	フェデレ ーテッド の長さの 上限	フェデレ ーテッド の位取り の下限	フェデレ ーテッド の位取り の上限	フェデレ ーテッド のビット ト・デー タ	フェデレ ーテッド のデータ 演算子	リモートのタ イプ名	リモー トの長 さ	リモー トの位 取り	リモート のビット ト・デー タ
BLOB	1	2097088000	-	-	-	-	BLOB	-	-	-
CHARACTER	-	-	-	-	-	-	CHARACTER	-	-	-
CHARACTER	-	-	-	-	Y	-	BYTE	-	-	-
CLOB	1	2097088000	-	-	-	-	CLOB	-	-	-
DATE	-	-	-	-	-	-	DATE	-	-	-
DBCLOB ¹	1	64000	-	-	-	-	VARGRAPHIC	-	-	-
DECIMAL	1	18	0	18	-	-	DECIMAL	-	-	-

表 130. Teradata のデフォルトの逆方向データ・タイプ・マッピング (一部の列は示されていません) (続き)

フェデレーテッドのタイプ名	フェデレーテッドの長さの下限	フェデレーテッドの長さの上限	フェデレーテッドの位取りの下限	フェデレーテッドの位取りの上限	フェデレーテッドのビット・データ	フェデレーテッドの演算子	リモートのタイプ名	リモートの長さ	リモートの位取り	リモートのビット・データ
DECIMAL	19	31	0	31	-	-	FLOAT	8	-	-
DOUBLE	-	-	-	-	-	-	FLOAT	-	-	-
GRAPHIC	-	-	-	-	-	-	GRAPHIC	-	-	-
INTEGER	-	-	-	-	-	-	INTEGER	-	-	-
REAL	-	-	-	-	-	-	FLOAT	8	-	-
SMALLINT	-	-	-	-	-	-	SMALLINT	-	-	-
TIME	-	-	-	-	-	-	TIME	15	-	-
TIMESTAMP	10	10	6	6	-	-	TIMESTAMP	26	6	-
VARCHAR	-	-	-	-	-	-	VARCHAR	-	-	-
VARCHAR	-	-	-	-	Y	-	VARBYTE	-	-	-
VARGRAPHIC	-	-	-	-	-	-	VARGRAPHIC	-	-	-

注:

1. Teradata VARGRAPHIC データ・タイプには、指定された長さ (1 から 32000) の DBCLOB データ・タイプしか格納できません。

Unicode のデフォルトのデータ・タイプ・マッピング

特定のデータ・ソースは、Unicode データベースに対する順方向データ・タイプ・マッピングと逆方向データ・タイプ・マッピングをサポートしています。

JDBC データ・ソースのデフォルトの順方向データ・タイプ・マッピング (Unicode)

以下の表に、フェデレーテッド・データベースが Unicode データベースである場合の、JDBC データ・ソースのデフォルトの順方向データ・タイプ・マッピングをリストしています。

表 131. JDBC データ・ソースのデフォルトの順方向データ・タイプ・マッピング (Unicode)

UTF-8 データ・タイプ	JDBC データ・タイプ	長さ
CHAR	CHAR	1 から 254 バイト
VARCHAR	VARCHAR	1 から 32672 バイト
CLOB	CLOB	-
GRAPHIC	NCHAR	1 から 127 文字
VARGRAPHIC	NVARCHAR	1 から 16336 文字
DBCLOB	NCLOB	-

JDBC データ・ソースのデフォルトの逆方向データ・タイプ・マッピング (Unicode)

以下の表に、フェデレーテッド・データベースが Unicode データベースである場合の、JDBC データ・ソースのデフォルトの逆方向データ・タイプ・マッピングをリストしています。

表 132. JDBC データ・ソースのデフォルトの逆方向データ・タイプ・マッピング (Unicode)

UTF-8		JDBC
データ・タイプ	長さ	データ・タイプ
CHAR	1 から 254 バイト	CHAR
VARCHAR	1 から 32672 バイト	VARCHAR
CLOB	1 から 2147483647 バイト	CLOB
GRAPHIC	1 から 127 文字	NCHAR
VARGRAPHIC	1 から 16336 文字	NVARCHAR
DBCLOB	1 から 1073741823 文字	NCLOB

Microsoft SQL Server データ・ソースのデフォルトの順方向データ・タイプ・マッピング (Unicode)

以下の表に、フェデレーテッド・データベースが Unicode データベースである場合の、Microsoft SQL Server データ・ソースのデフォルトの順方向データ・タイプ・マッピングをリストしています。

表 133. Microsoft SQL Server データ・ソースのデフォルトの順方向データ・タイプ・マッピング (Unicode)

UTF-8	Microsoft SQL Server	
データ・タイプ	データ・タイプ	長さ
CHAR	CHAR	1 から 254 バイト
VARCHAR	CHAR	255 から 8000 バイト
	VARCHAR	1 から 8000 バイト
CLOB	TEXT	-
GRAPHIC	NCHAR	1 から 127 文字
VARGRAPHIC	NCHAR	128 から 16336 文字
	NVARCHAR	1 から 16336 文字
DBCLOB	NTEXT	-

Microsoft SQL Server データ・ソースのデフォルトの逆方向データ・タイプ・マッピング (Unicode)

以下の表に、フェデレーテッド・データベースが Unicode データベースである場合の、Microsoft SQL Server データ・ソースのデフォルトの逆方向データ・タイプ・マッピングをリストしています。

表 134. Microsoft SQL Server データ・ソースのデフォルトの逆方向データ・タイプ・マッピング (Unicode)

UTF-8		Microsoft SQL Server
データ・タイプ	長さ	データ・タイプ
CHAR	1 から 254 バイト	CHAR
VARCHAR	1 から 32672 バイト	VARCHAR
CLOB	1 から 2 147 483 647 バイト	TEXT
GRAPHIC	1 から 127 文字	NCHAR
VARGRAPHIC	1 から 16336 文字	NVARCHAR
DBCLOB	1 から 1 073 741 823 文字	NTEXT

NET8 データ・ソースのデフォルトの順方向データ・タイプ・マッピング (Unicode)

以下の表に、フェデレーテッド・データベースが Unicode データベースである場合の、NET8 データ・ソースのデフォルトの順方向データ・タイプ・マッピングをリストしています。

表 135. NET8 データ・ソースのデフォルトの順方向データ・タイプ・マッピング (Unicode)

UTF-8	Oracle	
データ・タイプ	データ・タイプ	長さ
CHAR	CHAR	1 から 254 バイト
VARCHAR	CHAR	255 から 2000 バイト
	VARCHAR2	1 から 4000 バイト
DBCLOB	NCLOB	
GRAPHIC	NCHAR	1 から 127 文字
VARGRAPHIC	NCHAR	128 から 1000 文字
	NVARCHAR2	1 から 2000 文字

NET8 データ・ソースのデフォルトの逆方向データ・タイプ・マッピング (Unicode)

以下の表に、フェデレーテッド・データベースが Unicode データベースである場合の、NET8 データ・ソースのデフォルトの逆方向データ・タイプ・マッピングをリストしています。

表 136. NET8 データ・ソースのデフォルトの逆方向データ・タイプ・マッピング (Unicode)

UTF-8	Oracle	
データ・タイプ	長さ	データ・タイプ
CHAR	1 から 254 バイト	CHAR
VARCHAR	1 から 4000 バイト	VARCHAR2
CLOB	1 から 2 147 483 647 バイト	CLOB
GRAPHIC	1 から 127 文字	NCHAR
VARGRAPHIC	1 から 2000 文字	NVARCHAR2

表 136. NET8 データ・ソースのデフォルトの逆方向データ・タイプ・マッピング (Unicode) (続き)

UTF-8		Oracle
データ・タイプ	長さ	データ・タイプ
DBCLOB	1 から 1 073 741 823 文字	NCLOB

ODBC データ・ソースのデフォルトの順方向データ・タイプ・マッピング (Unicode)

以下の表に、フェデレーテッド・データベースが Unicode データベースである場合の、ODBC データ・ソースのデフォルトの順方向データ・タイプ・マッピングをリストしています。

表 137. ODBC データ・ソースのデフォルトの順方向データ・タイプ・マッピング (Unicode)

UTF-8	ODBC	
データ・タイプ	データ・タイプ	長さ
CHAR	SQL_CHAR	1 から 254 バイト
VARCHAR	SQL_CHAR	255 から 32672 バイト
	SQL_VARCHAR	1 から 32672 バイト
CLOB	SQL_LONGVARCHAR	-
GRAPHIC	SQL_WCHAR	1 から 127 文字
VARGRAPHIC	SQL_WCHAR	128 から 16336 文字
	SQL_WVARCHAR	1 から 16336 文字
DBCLOB	SQL_WLONGVARCHAR	-

ODBC データ・ソースのデフォルトの逆方向データ・タイプ・マッピング (Unicode)

以下の表に、フェデレーテッド・データベースが Unicode データベースである場合の、ODBC データ・ソースのデフォルトの逆方向データ・タイプ・マッピングをリストしています。

表 138. ODBC データ・ソースのデフォルトの逆方向データ・タイプ・マッピング (Unicode)

UTF-8	ODBC	
データ・タイプ	長さ	データ・タイプ
CHAR	1 から 254 バイト	SQL_CHAR
VARCHAR	1 から 32672 バイト	SQL_VARCHAR
CLOB	1 から 2 147 483 647 バイト	SQL_LONGVARCHAR
GRAPHIC	1 から 127 文字	SQL_WCHAR
VARGRAPHIC	1 から 16336 文字	SQL_WVARCHAR
DBCLOB	1 から 1 073 741 823 文字	SQL_WLONGVARCHAR

Sybase データ・ソースのデフォルトの順方向データ・タイプ・マッピング (Unicode)

以下の表に、フェデレーテッド・データベースが Unicode データベースである場合の、Sybase CTLIB データ・ソースのデフォルトの順方向データ・タイプ・マッピングをリストしています。

表 139. Sybase CTLIB データ・ソースのデフォルトの順方向データ・タイプ・マッピング (Unicode)

UTF-8	Sybase	
データ・タイプ	データ・タイプ	長さ
CHAR	char	1 から 254 バイト
	nchar	1 から 127 文字
VARCHAR	char	255 から 32672 バイト
	varchar	1 から 32672 バイト
	nchar	128 から 16336 文字
	nvarchar	1 から 16336 文字
CLOB	text	
GRAPHIC	unichar	1 から 127 文字
VARGRAPHIC	unichar	128 から 16336 文字
	univarchar	1 から 16336 文字

Sybase データ・ソースのデフォルトの逆方向データ・タイプ・マッピング (Unicode)

以下の表に、フェデレーテッド・データベースが Unicode データベースである場合の、Sybase CTLIB データ・ソースのデフォルトの逆方向データ・タイプ・マッピングをリストしています。

表 140. Sybase CTLIB データ・ソースのデフォルトの逆方向データ・タイプ・マッピング (Unicode)

UTF-8	Sybase	
データ・タイプ	長さ	データ・タイプ
CHAR	1 から 254 バイト	char
VARCHAR	1 から 32672 バイト	varchar
CLOB	1 から 2 147 483 647 バイト	text
GRAPHIC	1 から 127 文字	unichar
VARGRAPHIC	1 から 16336 文字	univarchar

非リレーショナル・データ・ソースでサポートされるデータ・タイプ

非リレーショナル・データ・ソースのほとんどの場合、そのデータ・ソースへのアクセスに使用するニックネームを作成する際に、データ・タイプなどの列情報を指定する必要があります。

一部の非リレーショナル・ラッパーは、データ・ソースへのアクセスに必要なすべての列を作成します。これらを**固定列**と呼びます。その他のラッパーでは、列のデータ・タイプの一部または全部を、ユーザーが CREATE NICKNAME ステートメントで指定します。

以下のセクションに、ユーザーがデータ・タイプを指定できるラッパーと、そのラッパーでサポートされるデータ・タイプをリストしています。

BioRS ラッパーでサポートされるデータ・タイプ

以下の表に、BioRS ラッパーでサポートされる DB2 データ・タイプをリストしています。

表 141. DB2 データ・タイプにマップする BioRS データ・タイプ

BioRS データ・タイプ	DB2 データ・タイプ
AUTHOR	CHARACTER, CLOB, VARCHAR
DATE	CHARACTER, CLOB, VARCHAR
NUMBER	CHARACTER, CLOB, VARCHAR
REFERENCE	CHARACTER, CLOB, VARCHAR
TEXT	CHARACTER, CLOB, VARCHAR

CLOB データ・タイプで許可される最大長は 5 メガバイトです。

Excel ラッパーでサポートされるデータ・タイプ

以下の表に、Excel ラッパーでサポートされる DB2 データ・タイプをリストしています。

表 142. DB2 データ・タイプにマップする Excel データ・タイプ

Excel データ・タイプ	DB2 データ・タイプ
文字	CHARACTER
日付	DATE
数値	DOUBLE
数値	FLOAT
整数	INTEGER
文字	VARCHAR

Script ラッパーでサポートされるデータ・タイプ

以下の表に、Script ラッパーでサポートされる DB2 データ・タイプをリストしています。

表 143. DB2 データ・タイプにマップする Script データ・タイプ

XML データ・タイプ	DB2 データ・タイプ
文字	BLOB
文字	CHARACTER
文字	CHARACTER FOR BIT DATA

表 143. DB2 データ・タイプにマップする Script データ・タイプ (続き)

XML データ・タイプ	DB2 データ・タイプ
文字	CLOB (最大長は 5 メガバイト)
日付	DATE
数値	DECIMAL
数値	DOUBLE
数値	FLOAT
整数	INTEGER
数値	REAL
整数	SMALLINT
文字	VARCHAR
文字	VARCHAR FOR BIT DATA

表構造ファイル・ラッパーでサポートされるデータ・タイプ

以下の表に、表構造ファイル・ラッパーでサポートされる DB2 データ・タイプをリストしています。

表 144. DB2 データ・タイプにマップする表構造ファイルのデータ・タイプ

表構造ファイルのデータ・タイプ	DB2 データ・タイプ
文字	CHARACTER
文字	CLOB (最大長は 5 メガバイト)
数値	DECIMAL
数値	DOUBLE
数値	FLOAT
整数	INTEGER
数値	REAL
整数	SMALLINT
文字	VARCHAR

Web サービス・ラッパーでサポートされるデータ・タイプ

以下の表に、Web サービス・ラッパーでサポートされる DB2 データ・タイプをリストしています。 Web サービス・ラッパーは、XML データ・タイプを使用しません。

表 145. Web サービス・ラッパーの、DB2 データ・タイプにマップする XML データ・タイプ

XML データ・タイプ	DB2 データ・タイプ
文字	BLOB
文字	CHARACTER
文字	CHARACTER FOR BIT DATA
文字	CLOB (最大長は 5 メガバイト)
日付	DATE

表 145. Web サービス・ラッパーの、DB2 データ・タイプにマップする XML データ・タイプ (続き)

XML データ・タイプ	DB2 データ・タイプ
数値	DECIMAL
数値	DOUBLE
数値	FLOAT
整数	INTEGER
数値	REAL
整数	SMALLINT
文字	VARCHAR
文字	VARCHAR FOR BIT DATA

XML ラッパーでサポートされるデータ・タイプ

以下の表に、XML ラッパーでサポートされる DB2 データ・タイプをリストしています。

表 146. XML ラッパーの、DB2 データ・タイプにマップする XML データ・タイプ

XML データ・タイプ	DB2 データ・タイプ
文字	BLOB
文字	CHARACTER
文字	CHARACTER FOR BIT DATA
文字	CLOB (最大長は 5 メガバイト)
日付	DATE
数値	DECIMAL
数値	DOUBLE
数値	FLOAT
整数	INTEGER
数値	REAL
整数	SMALLINT
文字	VARCHAR
文字	VARCHAR FOR BIT DATA
XML 文書	XML

アクセシビリティ対応資料

資料は XHTML 形式で提供され、これは、ほとんどの Web ブラウザーで表示可能です。

XHTML では、ご使用のブラウザーで設定された表示設定に従って資料を表示できます。スクリーン・リーダーやその他の支援技術も使用できます。

構文図は小数点付き 10 進数の形式で提供されます。この形式は、スクリーン・リーダーを使用してオンライン文書にアクセスした場合にのみ、使用可能です。

特記事項および商標

本書は米国 IBM が提供する製品およびサービスについて作成したものです。

特記事項

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒103-8510
東京都中央区日本橋箱崎町19番21号
日本アイ・ビー・エム株式会社
法務・知的財産
知的財産権ライセンス渉外

以下の保証は、国または地域の法律に沿わない場合は、適用されません。IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003 U.S.A.

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性があります。その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

本書はプランニング目的としてのみ記述されています。記述内容は製品が使用可能になる前に変更になる場合があります。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。

できます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほめかしたり、保証することはできません。これらのサンプル・プログラムは特定物として現存するままの状態を提供されるものであり、いかなる保証も提供されません。IBM は、お客様の当該サンプル・プログラムの使用から生ずるいかなる損害に対しても一切の責任を負いません。

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生的創作物にも、次のように、著作権表示を入れていただく必要があります。

© (お客様の会社名) (西暦年).このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。 © Copyright IBM Corp. _年を入れる_ All rights reserved.

この情報をソフトコピーでご覧になっている場合は、写真やカラーの図表は表示されない場合があります。

商標

IBM、IBM ロゴおよび ibm.com は、世界の多くの国で登録された International Business Machines Corporation の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、www.ibm.com/legal/copytrade.shtml をご覧ください。

以下は、それぞれ各社の商標または登録商標です。

Adobe は、Adobe Systems Incorporated の米国およびその他の国における登録商標です。

IT Infrastructure Library は英国 Office of Government Commerce の一部である the Central Computer and Telecommunications Agency の登録商標です。

Intel、Intel (ロゴ)、Intel Inside、Intel Inside (ロゴ)、Intel Centrino、Intel Centrino (ロゴ)、Celeron、Intel Xeon、Intel SpeedStep、Itanium、Pentium は、Intel Corporation または子会社の米国およびその他の国における商標または登録商標です。

Linux は、Linus Torvalds の米国およびその他の国における登録商標です。

Microsoft、Windows、Windows NT および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。

ITIL は、英国 Office of Government Commerce の登録商標および共同体商標であって、米国特許商標庁にて登録されています。

UNIX は The Open Group の米国およびその他の国における登録商標です。

Cell Broadband Engine, Cell/B.E は、米国およびその他の国における Sony Computer Entertainment, Inc. の商標であり、同社の許諾を受けて使用しています。

Java およびすべての Java 関連の商標およびロゴは Oracle やその関連会社の米国およびその他の国における商標または登録商標です。

索引

日本語, 数字, 英字, 特殊文字の順に配列されています。なお, 濁音と半濁音は清音と同等に扱われています。

[ア行]

アクセシビリティ 421
アプリケーション・セーブポイント
サポートされるデータ・ソース 295
引用符
SQL ステートメントでの使用 27
エラー許容度
サポートされるデータ・ソース 295
エラー・ロギング
Teradata アクセス・ロギングの使用不可 239
大文字
SQL ステートメントでの使用 27

[カ行]

カスタム関数
スクリプト 178
BioRS 50
BioRS, 説明 36
BioRS, 登録 35
カタログ
「グローバル・カタログ」を参照 379
各国語サポート (NLS)
フェデレーテッド・データベース 19
環境変数 13
制約事項 14
設定
データ・ソース・クライアント 13
Oracle 変数 161
Teradata 環境変数 223
データ・ソース 13
必須 13
Informix 93
Informix ラッパー 92
Microsoft SQL Server データ・ソース 121
Microsoft SQL Server 変数の構成 120
Oracle データ・ソース 163
Sybase 環境変数の構成 194
Sybase 環境変数のトラブルシューティング 208
Sybase データ・ソース 196
Teradata データ・ソース 224

関数マッピング
オプション 383
逆方向タイプ・マッピング
デフォルトのマッピング 402
Unicode
JDBC データ・ソース 414
Microsoft SQL Server データ・ソース 415
NET8 データ・ソース 415
ODBC データ・ソース 416
Sybase データ・ソース 417
キャッシュ表
サポートされるデータ・ソース 295
グローバル・カタログ
フェデレーテッド情報を含むビュー 379
コード・セット
フェデレーテッド・データベース 19
コード・ページ
異なる場合のエラー 24
フェデレーテッド・システム 21
フェデレーテッド・データベース 19
変換
Informix 94
Oracle データ・ソース 165
MSSQL, ODBCm, および
CODEPAGE でサポートされる 23
Unicode サポートのための指定
概要 23
例 24

構成
ツール 1
フェデレーテッド・サーバー 30
構成ツール 1
構成ファイル
Informix クライアント
セットアップおよびテスト 90
コマンド
発行 11
コマンド行プロセッサ (CLP)
開始 11
小文字
SQL ステートメントでの使用 27

[サ行]

サーバー
Microsoft SQL Server への接続のテスト 127
Sybase サーバーへの接続のテスト 205

サーバー定義
作成 4
登録 70
Informix データ・ソースの登録
DB2 コマンド行 97
JDBC データ・ソース 109
Microsoft SQL Server データ・ソース
についての登録 123
ODBC データ・ソースのための登録 139
OLE DB データ・ソースの登録 158
Oracle データ・ソースのための登録 168
Sybase データ・ソースに関する登録 201
Teradata データ・ソースについての登録 229
Web サービス 248
サーバー・オプション
スクリプト 344
表構造ファイル 360
BioRS 299
DB2 データベース 303
Excel 311
FOLD_ID および FOLD_PW サーバー・オプションの設定 103
Informix 312
JDBC 318
Microsoft SQL Server 326
ODBC 331
OLE DB の例 159
Oracle 338
Sybase 349
Teradata 356
Web サービス 363
XML 371
サーバー・タイプ
有効なフェデレーテッド・タイプ 385
最適化
スクリプト・データ・ソース 192
サンプルの照会
BioRS データ・ソース 56
指定
値
オプション 27
シナリオ
Excel ファイルへのアクセス 87
順方向タイプ・マッピング
デフォルトのマッピング 387
Unicode
JDBC データ・ソース 413

順方向タイプ・マッピング (続き)
 Unicode (続き)
 Microsoft SQL Server 414
 NET8 データ・ソース 415
 ODBC データ・ソース 416
 Sybase データ・ソース 417

照会
 スクリプト・データ・ソース、データ・ソースへのアクセス 190
 データ・ソースへのアクセス
 Web サービス 264

例
 Web サービス・ラッパー 261
 BioRS データ・ソースへのアクセス 50, 56
 BioRS 等価演算 53
 Excel データ・ソースへのアクセス 87
 XML 照会 292

照会の最適化
 表構造ファイルのためのガイドライン 220
 BioRS 照会のガイドライン 61
 XML 291

照合シーケンス
 フェデレーテッド・データベース 19

商標
 リスト 423

資料
 アクセシビリティ対応の 421
 スクリーン・リーダー 421

スクリプト
 サーバー・オプション 344
 サポートされるフェデレーテッド・フィーチャー 295
 データ・ソースとして構成 176
 ニックネーム・オプション 344
 ユーザー・マッピング・オプション 344
 ラッパー・オプション 344
 列オプション 344

スクリプト・データ・ソース
 サーバー定義 185
 サンプルの照会 190
 ニックネームの登録 186
 ニックネーム・オプション 187
 パフォーマンスの最適化 192
 ラッパー・ライブラリー名 185
 CREATE NICKNAME ステートメントの例 187
 CREATE SERVER ステートメント 185

スクリプト・ラッパー
 概要 176
 カスタム関数 178
 作成 184

ステートメント・レベルの分離
 サポートされるデータ・ソース 295

制約事項
 db2dj.ini ファイル 14

セキュリティ
 Web サービス 249

接続
 テスト 74
 トラブルシューティング 75, 102, 114, 128, 143, 173, 205, 234
 Oracle サーバー接続のテスト 172
 Oracle データ・ソースのトラブルシューティング 173

接続に関する問題
 Microsoft SQL Server データ・ソース 130

接続レベルの分離
 サポートされるデータ・ソース 295

[タ行]

大/小文字の区別
 保持 27

データのインポート
 サポートされるデータ・ソース 295

データベース
 カタログ
 リモート側で 68

データ・ソース
 オプション 299
 構成 1, 67
 構成の概要 25
 サポートされるフィーチャー 295
 接続のトラブルシューティング 75, 102, 114, 128, 143, 173, 205, 234
 登録 68
 メッセージ・ファイル 15
 有効なサーバー・タイプ 385
 Informix サーバー定義の登録
 DB2 コマンド行 97
 Informix サーバーへの接続のテスト 101
 Informix へのアクセスの構成 90
 Informix ラッパーの登録 96

データ・タイプ
 非リレーショナル・データ・ソース 418

データ・タイプ・マッピング
 逆方向 402
 順方向 387

データ・バンク
 説明 33

デーモン
 構成ファイル
 スクリプト例 179
 スクリプトのための開始 182

デーモン (続き)
 スクリプトのための構成 179

デフォルトの逆方向データ・タイプ・マッピング
 DB2 Database for Linux, UNIX, and Windows データ・ソース 403
 DB2 for System i データ・ソース 404
 DB2 for VM and VSE データ・ソース 405
 DB2 for z/OS データ・ソース 406
 Informix データ・ソース 406
 JDBC データ・ソース 407
 Microsoft SQL Server データ・ソース 408
 ODBC データ・ソース 409
 Oracle NET8 データ・ソース 410
 Sybase データ・ソース 411

デフォルトの順方向データ・タイプ・マッピング
 例 400
 DB2 Database for Linux, UNIX, and Windows データ・ソース 387
 DB2 for System i データ・ソース 388
 DB2 for VM and VSE データ・ソース 389
 DB2 for z/OS データ・ソース 390
 Informix データ・ソース 391
 Informix のサンプル 400
 JDBC データ・ソース 392
 Microsoft SQL Server データ・ソース 394
 Microsoft SQL Server のサンプル 401
 ODBC データ・ソース 395
 Oracle NET8 データ・ソース 397
 Oracle のサンプル 401
 Sybase データ・ソース 398
 Sybase のサンプル 402
 Teradata データ・ソース 399, 412
 Teradata のサンプル 402

ドイツ Biomax 社 33

統計
 BioRS データ・ソースの 62

特記事項 423

トラブルシューティング
 接続 75, 102, 114, 128, 143, 173, 205, 234
 データ・ソース・アクセス 12
 フェデレーテッド・サーバー 15
 フェデレーテッド・サーバーのセットアップ 12
 Federation のインストール 15, 17

[ナ行]

ニックネーム
オプション
スクリプト 344
表構造ファイル 360
BioRS 299
Web サービス 363
XML 371
作成 4
DB2 コマンド行の Web サービス・データ・ソース 250
Web サービス 251
スクリプトの登録 186
説明
スクリプト 186
XPath 式 259
データベース技法
Web サービス・ラッパー 261
登録
DB2 表 76
CREATE NICKNAME
例 207
Informix 表、ビュー、およびシノニムの登録 104
JDBC データ・ソース 115
Microsoft SQL Server データ・ソースについての登録 129
ODBC データ・ソースのための登録 144
Oracle データ・ソース 174
Sybase データ・ソース 206
Teradata データ・ソース
登録 235
フェデレーテッド・サーバー 235
Web サービス記述言語 (WSDL) 239
Web サービス・データ・ソース
DB2 コマンド行 250
ニックネームおよび列オプション
Web サービス・ラッパー 239, 260
ニックネーム統計の更新機能
サポートされるデータ・ソース 295
ニックネーム・オプション
DB2 データベース 303
ノード
カタログ 67
DB2 67

[ハ行]

場所
デフォルト 13
パススルー・セッション
サポートされるデータ・ソース 295
パスワード
大/小文字の区別 27

非同期最適化
サポートされるデータ・ソース 295
表構造ファイル
サーバー・オプション 360
サポートされるフェデレーテッド・フィーチャー 295
データ・タイプ、サポートされる 418
ニックネーム・オプション 360
ラッパー・オプション 360
列オプション 360
Unicode サポート 23, 24
表構造ファイル・データ・ソース 209
概要 209
サーバー定義、登録 213
照会の最適化 220
前提条件 212
デフォルト・ディレクトリ・パス 212
ニックネーム、登録 214
ファイル・アクセスの許可 219
ファイル・タイプ属性 210
フェデレーテッド・システムへの追加 212
ラッパー、登録 212
ラッパーの概要 210
ラッパー・ライブラリ名 212
CREATE NICKNAME ステートメント構文 214
CREATE NICKNAME ステートメントの例 215
CREATE SERVER ステートメント構文 213
CREATE SERVER ステートメントの例 213
CREATE WRAPPER ステートメント構文 212
CREATE WRAPPER ステートメントの例 212
非リレーショナル・データ・ソース
サポートされるデータ・タイプ 418
フェデレーション
インストール 1
トラブルシューティング 17
有効化 12
フェデレーテッド・オブジェクト
Data Studio 2, 4
フェデレーテッド・サーバー
構成 1
構成、複数の 8
構成の概要 25
コマンド 11
セットアップ 12
データベース 19
データベースのカタログ 68
データ・ソースの構成 67
トラブルシューティング 12

フェデレーテッド・サーバー (続き)
複数の構成 30
Informix へのアクセスの構成 90
JDBC データ・ソース・サーバーへのアクセスのテスト 114
JDBC 用の構成 107
Microsoft SQL Server データ・ソースへのアクセス 117
Microsoft SQL Server データ・ソースへのアクセスの構成 117
Microsoft SQL Server へのアクセスの構成 116
ODBC データ・ソース・サーバーへのアクセスのテスト 142
ODBC 用の構成 133, 134
Sybase データ・ソースへのアクセスの構成 192
Teradata サーバーへの接続のテスト 221
フェデレーテッド・データベース
コード・セット 19
作成 19
照合シーケンス 19
フェデレーテッド・トラステッド・コンテキスト
サポートされるデータ・ソース 295
フェデレーテッド・ビュー
Web サービス 260
フェデレーテッド・プロシージャ
サポートされるデータ・ソース 295
フェデレーテッド・ヘルス・インディケータ
サポートされるデータ・ソース 295
プラン
Excel データへのアクセス 79
プロキシ・サーバー
BioRS ラッパーの例 38
Web サービス・サーバーの定義の例 248
XML データ・ソースへのアクセス 278

[マ行]

マッピング
XML 文書 259
マテリアライズ照会表
サポートされるデータ・ソース 295
文字セット
Teradata データ・ソース構成のトラブルシューティング 227, 238

[ヤ行]

ユーザー ID
大/小文字の区別 27
ユーザー・マッピング
作成 4, 72
Informix データ・ソース用の作成
100
JDBC データ・ソース 112
Microsoft SQL Server データ・ソース
についての定義 126
ODBC データ・ソースの定義 141
OLE DB データ・ソース 160
Oracle データ・ソースのための定義
170
Sybase データ・ソースに関する作成
203
Teradata データ・ソースについての定
義 232
ユーザー・マッピング・オプション
スクリプト 344
BioRS 299
DB2 データベース 303
Informix 312
JDBC 318
Microsoft SQL Server 326
ODBC 331
Oracle 338
Sybase 349
Teradata 356
Web サービス 363
XML 371
ユーザー・マッピング・リポジトリー
サポートされるデータ・ソース 295

[ラ行]

ライセンス
フェデレーション 18
ポリシー
フェデレーション 18
ライセンス・キー
フェデレーション 18
ライブラリー
スクリプト・ラッパー 185
表構造ファイル・ラッパー 212
BioRS ラッパー 37
Excel ラッパー 83
Teradata ラッパー 229
XML ラッパー 272
ライブラリー・ファイル
ファイル名 15
リンク 15, 17
ラッパー
作成 4
スクリプト 176

ラッパー (続き)
JDBC ラッパーの構成 105
JDBC ラッパーの登録 108
JDBC ラッパー・ライブラリー・ファ
イル 108
Microsoft SQL Server ラッパーの登録
122
Microsoft SQL Server ラッパー・ライ
ブラリー 123
ODBC ラッパーの構成 131, 153, 155
ODBC ラッパーの登録 136
ODBC ラッパー・ライブラリー・ファ
イル 137
OLE DB ラッパーの構成 156
OLE DB ラッパー・ライブラリー・フ
ァイル 158
Oracle ラッパーの登録 167
Oracle ラッパー・ライブラリー・ファ
イル 168
Sybase ラッパーの登録 200
Teradata データ・ソースに関するラン
タイム・リンクの検査 222
Teradata データ・ソースへのアクセス
の構成 221
Teradata ラッパーの登録 228
Web サービス・データ・ソース 239,
246
ラッパー・オプション
スクリプト 344
表構造ファイル 360
BioRS 299
DB2 データベース 303
Excel 311
Informix 312
JDBC 318
Microsoft SQL Server 326
ODBC 331
Oracle 338
Sybase 349
Teradata 356
Web サービス 363
XML 371
ラッパー・ライブラリー 247
スクリプト・ファイル名 185
表構造ファイル名 212
BioRS ファイル名 37
Excel ファイル名 83
Teradata ファイル名 229
XML 272
ラッパー・ライブラリー・ファイル
ファイル名 15
ラベル・ベースのアクセス制御
サポートされるデータ・ソース 295
リンク
ライブラリー・ファイル 15

例

照会
Web サービス 264
スクリプト・データ・ソース、
CREATE NICKNAME 187
スクリプト・データ・ソース、
CREATE SERVER 185
プロキシ・サーバー
BioRS ユーザー・マッピング例
44
BioRS 照会 54, 56
BioRS データ・ソース
プロキシ・サーバー、ユーザー・
マッピングの例 44
CREATE NICKNAME ステートメン
ト、表構造ファイル 215
CREATE NICKNAME ステートメン
ト、BioRS 47
CREATE NICKNAME ステートメン
ト、Excel 85
CREATE SERVER
Informix 98
Teradata 231
CREATE SERVER ステートメント、
表構造ファイル 213
CREATE SERVER ステートメント、
BioRS 40
CREATE SERVER ステートメント、
Excel 84
CREATE SERVER ステートメント、
Web サービス 248
CREATE SERVER ステートメント、
XML 275
CREATE USER MAPPING ステートメ
ント
Informix 100
Sybase データ・ソース 204
CREATE USER MAPPING ステートメ
ント、BioRS 44
CREATE VIEW ステートメント、
XML 290
CREATE WRAPPER ステートメン
ト、表構造ファイル 212
CREATE WRAPPER ステートメン
ト、BioRS 38
CREATE WRAPPER ステートメン
ト、Excel 83
CREATE WRAPPER ステートメン
ト、XML 273
db2script_daemon コマンド 182
Excel の照会 87
Excel ファイル 77, 79
UPDATE ステートメント、BioRS 64
Web サービス・ラッパー 251, 261
XML の例 292
XML 文書マッピング 268

列オプション
スクリプト 344
表構造ファイル 360
BioRS 299
DB2 データベース 303
Informix 312
JDBC 318
Microsoft SQL Server 326
ODBC 331
Oracle 338
Sybase 349
Teradata 356
Web サービス 363
XML 371

[数字]

2 フェーズ・コミット
サポートされるデータ・ソース 295

A

AllText エレメント
BioRS 56

B

BioRS
サポートされるフェデレーテッド・フ
ィーチャー 295
データ・タイプ、サポートされる 418
BioRS データ・ソース
アクセスの構成 33
カーディナリティー統計、確認 63
カーディナリティー統計、ニックネー
ム 63
カスタム関数
概要 50
説明 36
登録 35
サーバー定義、登録 39
サーバー・オプション 299
サンプルの照会 54
照会、結合述部の使用 54
照会、等価演算 53
照会パフォーマンス、最適化 62
説明 33
前提条件 34
デフォルト・ディレクトリー・パス
37
統計
ニックネーム・カーディナリティー
ー、更新 63
列カーディナリティー、更新 64

BioRS データ・ソース (続き)
統計 (続き)
ID 列カーディナリティー、更新
66
統計情報 62
ニックネームの登録 46
ニックネーム・オプション 299
フェデレーテッド・システムへの
BioRS の追加 34
プロキシ・サーバー
サーバー定義の例 40
ラッパーの例 38
BioRS サーバー定義の例 40
ユーザー・マッピング、作成 42
ユーザー・マッピング・オプション
299
ラッパー、登録 37
ラッパーのパフォーマンス、最適化
61
ラッパー・オプション 299
ラッパー・パフォーマンスの最適化
61
ラッパー・ライブラリー名 37
列オプション 299
ワイルドカード、照会での 50, 56
admin_find ユーティリティー・プログ
ラム 63
AllText エレメント 56
CREATE FUNCTION ステートメント
構文 35
CREATE NICKNAME ステートメント
構文 46
CREATE NICKNAME ステートメント
の例 47
CREATE SERVER ステートメント構
文 39
CREATE SERVER ステートメントの
例 40
CREATE USER MAPPING ステートメ
ント構文 42
CREATE WRAPPER ステートメント
構文 37
CREATE WRAPPER ステートメント
の例 38
www_find.cgi ユーティリティー・プロ
グラム 63

C

CODEPAGE オプション 23
CONCENTRATOR パラメーター 12
CREATE FUNCTION ステートメント
BioRS 構文 35
CREATE NICKNAME ステートメント
構文、Excel 85
スクリプト・データ・ソース 187

CREATE NICKNAME ステートメント
(続き)
表構造ファイル構文 214
表構造ファイルの例 215
例、Excel 85
BioRS 構文 46
BioRS の例 47
Informix 104
JDBC データ・ソース 115
Teradata 236
Teradata データ・ソースに関するトラ
ブルシューティング 237
Web サービス・データ・ソース 250
XML の例 284
CREATE NICKNAME の例
Microsoft SQL Server データ・ソース
129
ODBC データ・ソース 144
Oracle データ・ソース 175
CREATE SERVER
JDBC データ・ソース 110
CREATE SERVER ステートメント
構文、Excel 84
スクリプト・データ・ソース 185
表構造ファイル構文 213
表構造ファイルの例 213
例、Excel 84
BioRS 構文 39
BioRS の例 40
DB2 ファミリーのデータ・ソース 71
Informix 98
ODBC の例 139
Sybase データ・ソース 202
Teradata 231
Web サービスの例 248
XML 構文 274
XML の例 275
CREATE SERVER の例
Microsoft SQL Server データ・ソース
125
OLE DB データ・ソース 159
Oracle データ・ソース 169
Sybase データ・ソース 202
CREATE USER MAPPING
JDBC データ・ソース 113
CREATE USER MAPPING ステートメン
ト
BioRS 構文 42
BioRS の例 44
Informix 100
Sybase データ・ソース 204
CREATE USER MAPPING の例
Microsoft SQL Server データ・ソース
126
ODBC データ・ソース 141
OLE DB データ・ソース 160

CREATE USER MAPPING の例 (続き)
Oracle データ・ソース 171
Teradata データ・ソース 232
CREATE VIEW ステートメント
XML の例 290
CREATE WRAPPER ステートメント
表構造ファイル構文 212
表構造ファイルの例 212
BioRS 構文 37
BioRS の例 38
Excel、例 83
JDBC データ・ソース 109
ODBC の例 137
OLE DB データ・ソースの登録 157
XML 構文 272
XML の例 273

D

DAEMON_LOGFILE_DIR
スクリプト・デーモン 179
DAEMON_PORT
スクリプト・デーモン 179
Data Studio 2, 4
管理エクスプローラー 4, 8
SQL および XQuery エディター 9
DB2 Database for Linux, UNIX, and
Windows データ・ソース
デフォルトの逆方向データ・タイプ・
マッピング 403
デフォルトの順方向データ・タイプ・
マッピング 387
DB2 for Linux, UNIX and Windows
デフォルトの逆方向タイプ・マッピ
ング 402
デフォルトの順方向タイプ・マッピ
ング 387
DB2 for System i
デフォルトの逆方向タイプ・マッピ
ング 402
デフォルトの順方向タイプ・マッピ
ング 387
DB2 for System i データ・ソース
デフォルトの逆方向データ・タイプ・
マッピング 404
デフォルトの順方向データ・タイプ・
マッピング 388
DB2 for VM and VSE
デフォルトの逆方向タイプ・マッピ
ング 402
デフォルトの順方向タイプ・マッピ
ング 387
DB2 for VM and VSE データ・ソース
デフォルトの逆方向データ・タイプ・
マッピング 405

DB2 for VM and VSE データ・ソース
(続き)
デフォルトの順方向データ・タイプ・
マッピング 389
DB2 for z/OS and OS/390
デフォルトの逆方向タイプ・マッピ
ング 402
デフォルトの順方向タイプ・マッピ
ング 387
DB2 for z/OS データ・ソース
デフォルトの逆方向データ・タイプ・
マッピング 406
デフォルトの順方向データ・タイプ・
マッピング 390
DB2 データベース
サーバー・オプション 303
サポートされるフェデレーテッド・フ
ィーチャー 295
ニックネーム・オプション 303
ユーザー・マッピング・オプション
303
ラッパー・オプション 303
列オプション 303
DB2 データ・ソース
アクセス権限の構成 4
構成 67
サーバーの登録 70
接続のテスト 74
デフォルト・ディレクトリー・パス
69
登録 68
ニックネーム 76
ニックネームの作成 77
ユーザー・マッピング 72, 73
ライブラリー・ファイル 69
CREATE SERVER ステートメント
71
DB2 特殊レジスター USER
OLE DB データ・ソースの例 160
DB2 ラッパー
登録 68
db2dj.ini ファイル 13
制約事項 14
db2fedsvrcfg コマンド
構文 146
テスト表の作成 148
例 146
db2script_daemon コマンド 182
DB2_DJ_INI レジストリー変数 13
DISABLE 関数マッピング・オプション
有効な設定値 383
DRDA ラッパー
ライブラリー・ファイル名 69

E

Excel
サーバー・オプション 311
サポートされるフェデレーテッド・フ
ィーチャー 295
ニックネーム・オプション 311
ラッパー・オプション 311
Excel ODBC ドライバー
アクセス許可 151
Microsoft Excel ワークブックへのアク
セス 149
Excel データ・ソース 77
サーバー定義、登録 84
サンプルの照会 87
サンプル・ユーザーのシナリオ 87
説明 77
前提条件 82
デフォルト・ディレクトリー・パス
83
ニックネーム、登録 85
ファイル・アクセスの許可 89
ラッパー、登録 83
ラッパー・ライブラリー名 83
CREATE NICKNAME ステートメント
の例 85
CREATE SERVER ステートメントの
例 84
CREATE WRAPPER ステートメント
の例 83
Excel のフェデレーテッド・システム
への追加 82
Excel ファイル
アクセスの準備 79
データ・タイプ、サポートされる 418

F

FEDERATED パラメーター 12
FOLD_ID サーバー・オプション
大/小文字の区別 27
FOLD_PW サーバー・オプション
大/小文字の区別 27

H

HTTP プロキシ
サポートされるデータ・ソース 295

I

Informix
コード・ページ変換 94
サーバー・オプション 312

Informix (続き)
サポートされるフェデレーテッド・フ
ィーチャー 295
デフォルトの逆方向タイプ・マッピ
ング 402
デフォルトの順方向タイプ・マッピ
ング 387
デフォルト・ディレクトリー・パス
96
ユーザー・マッピング・オプション
312
ライブラリー・ファイル 17
ライブラリー・ファイル名 96
ラッパー・オプション 312
列オプション 312
CREATE NICKNAME ステートメン
トの例 104
CREATE SERVER ステートメン
ト 98
CREATE USER MAPPING ステートメ
ント 100
Informix クライアント構成ファイル
セットアップおよびテスト 90
Informix サーバー
接続のテスト 101
Informix シノニム
ニックネームの登録 104
Informix データ・ソース
環境変数 13
デフォルトの逆方向データ・タイプ・
マッピング 406
デフォルトの順方向データ・タイプ・
マッピング 391
ライブラリー・ファイル 15
Informix ビュー
ニックネームの登録 104
Informix 表
ニックネームの登録 104
Informix ラッパー
環境変数 92, 93
登録 96
パフォーマンスの改善 103
ユーザー・マッピングの作成 100

J

JDBC
サーバー・オプション 318
サポートされるフェデレーテッド・フ
ィーチャー 295
ユーザー・マッピング・オプション
318
ラッパー・オプション 318
列オプション 318
JDBC データ・ソース
アクセスの構成 105

JDBC データ・ソース (続き)
アクセスのテスト 114
環境変数 13
サーバー定義の登録 108, 109
デフォルトの逆方向タイプ・マッピ
ング (Unicode) 414
デフォルトの逆方向データ・タイプ・
マッピング 407
デフォルトの順方向タイプ・マッピ
ング (Unicode) 413
デフォルトの順方向データ・タイプ・
マッピング 392
ニックネーム 115
フェデレーテッド・サーバーの構成
107
ユーザー・マッピングの定義 112
ラッパー・ライブラリー・ファイル
108
CREATE NICKNAME の例 115
CREATE SERVER の例 110
CREATE USER MAPPING の例 113
CREATE WRAPPER の例 109

L

LOB
サポートされるデータ・ソース、読み
取りおよび書き込み 295
サポートされるデータ・ソース、読み
取り専用 295

M

Microsoft Excel
ワークブック
アクセス 149
アクセス許可 151
Microsoft SQL Server
サーバー・オプション 326
サポートされるフェデレーテッド・フ
ィーチャー 295
デフォルトの逆方向タイプ・マッピ
ング 402
デフォルトの順方向タイプ・マッピ
ング 387
デフォルトの順方向タイプ・マッピ
ング (Unicode) 414
ユーザー・マッピング・オプション
326
ライブラリー・ファイル 17
ラッパー・オプション 326
列オプション 326
Unicode サポート 23
Microsoft SQL Server データ・ソース
アクセス 117

Microsoft SQL Server データ・ソース (続
き)
アクセスの構成 116
環境変数 13, 121
環境変数の構成 120
構成のトラブルシューティング 130
サーバー定義の登録 123
デフォルトの逆方向タイプ・マッピ
ング (Unicode) 415
デフォルトの逆方向データ・タイプ・
マッピング 408
デフォルトの順方向データ・タイプ・
マッピング 394
登録 122
ニックネーム 129
フェデレーテッド・サーバーの構成
117
ライブラリー・ファイル 15
ラッパー・ライブラリー・ファイル
123
CREATE NICKNAME の例 129
CREATE SERVER の例 125
CREATE USER MAPPING の例 126
Microsoft SQL Server への接続のテス
ト 127

Microsoft SQL データ・ソース
ユーザー・マッピングの定義 126

N

NET8 データ・ソース
デフォルトの逆方向タイプ・マッピ
ング (Unicode) 415
デフォルトの順方向タイプ・マッピ
ング (Unicode) 415

O

ODBC
環境変数 13
サーバー・オプション 331
サポートされるフェデレーテッド・フ
ィーチャー 295
デフォルトの順方向タイプ・マッピ
ング 387
トレース 130
ユーザー・マッピング・オプション
331
ラッパー・オプション 331
列オプション 331
Unicode サポート 23
ODBC チューニング・ユーティリティー
テスト表の作成 148
db2fedsvrcfg コマンド構文 146
db2fedsvrcfg コマンドの例 146

ODBC チューニング・ユーティリティ
(続き)
ODBC ラッパーのチューニング 145

ODBC データ・ソース
アクセス許可 151
アクセスのテスト 142
サーバー定義の登録 139
デフォルトの逆方向タイプ・マッピング (Unicode) 416
デフォルトの逆方向データ・タイプ・マッピング 409
デフォルトの順方向タイプ・マッピング (Unicode) 416
デフォルトの順方向データ・タイプ・マッピング 395
登録 136
ニックネームの登録 144
フェデレーテッド・サーバーの構成 133, 134
ユーザー・マッピングの定義 141
ラッパー・ライブラリー・ファイル 137
CREATE NICKNAME の例 144
CREATE SERVER の例 139
CREATE USER MAPPING の例 141
CREATE WRAPPER の例 137
IBM InfoSphere Classic Federation
Server for z/OS データ・ソースへのアクセスの構成 153, 155
Microsoft Excel ワークブックへのアクセス 149
ODBC チューニング・ユーティリティによるチューニング 145
ODBC チューニング・ユーティリティのためのテスト表 148
ODBC データ・ソースへのアクセスの構成 131

OLE DB データ・ソース
アクセスの構成 156
サーバー定義の登録 158
サーバー・オプション 159
登録 157
ユーザー・マッピングの定義 160
ライブラリー・ファイル名 158
CREATE SERVER の例 159
CREATE USER MAPPING の例 160

Oracle
クライアント構成ファイル 166
サーバー接続のテスト 172
サーバー・オプション 338
サポートされるフェデレーテッド・フィーチャー 295
デフォルトの逆方向タイプ・マッピング 402
デフォルトの順方向タイプ・マッピング 387

Oracle (続き)
ユーザー・マッピング・オプション 338
ライブラリー・ファイル 17
ラッパー・オプション 338
列オプション 338

Oracle NET8 データ・ソース
デフォルトの逆方向データ・タイプ・マッピング 410
デフォルトの順方向データ・タイプ・マッピング 397

Oracle データ・ソース
アクセス 161
アクセス権限の構成 4
環境変数 13, 163
環境変数の設定 161
コード・ページ変換 165
サーバー定義の登録 168
接続に関する問題のトラブルシューティング 173
接続の構成 166
登録 167
ニックネームの登録 174
フェデレーテッド・システムへの追加 161
ユーザー・マッピングの定義 170
ライブラリー・ファイル 15
ラッパー・ライブラリー・ファイル 168
CREATE NICKNAME の例 175
CREATE SERVER の例 169
CREATE USER MAPPING の例 171

R

REMOTE_NAME 関数マッピング・オプション
有効な設定値 383

S

Secure Socket Layer (SSL)
サポートされるデータ・ソース 295

SOAP ヘッダー
フェデレーテッド・アクセス 257

SOCKS プロキシ
サポートされるデータ・ソース 295

SQL10013N
トラブルシューティング 207

SQL1097N
トラブルシューティング 208

SQL1822N
トラブルシューティング 208

Sybase
サーバー・オプション 349

Sybase (続き)

サポートされるフェデレーテッド・フィーチャー 295
デフォルトの逆方向タイプ・マッピング 402
デフォルトの順方向タイプ・マッピング 387
ユーザー・マッピング・オプション 349
ライブラリー・ファイル 17
ラッパー・オプション 349
列オプション 349
djxlinkSybase スクリプト 17

Sybase クライアント構成ファイル
構成およびテスト 197, 198

Sybase データ・ソース
アクセスの構成 192
環境変数 13, 196
環境変数の構成 194
クライアント構成ファイルの構成 197, 198
サーバー定義の登録 201
デフォルトの逆方向タイプ・マッピング (Unicode) 417
デフォルトの逆方向データ・タイプ・マッピング 411
デフォルトの順方向タイプ・マッピング (Unicode) 417
デフォルトの順方向データ・タイプ・マッピング 398
登録 200
ニックネームの登録 206
ユーザー・マッピングの作成 203
ライブラリー・ファイル 15
ラッパー構成のトラブルシューティング 207, 208
ラッパーのデフォルトのディレクトリー・パス 200
ラッパー・ライブラリー・ファイル名 200
CREATE NICKNAME の例 207
CREATE SERVER の例 202
CREATE USER MAPPING ステートメント 204
CREATE USER MAPPING の例 204
Sybase サーバーへの接続のテスト 205

SYSCAT カタログ・ビュー 379
SYSSTAT カタログ・ビュー 379

T

Teradata
サーバー・オプション 356
サポートされるフェデレーテッド・フィーチャー 295

Teradata (続き)

- デフォルトの逆方向タイプ・マッピング 402
- デフォルトの順方向タイプ・マッピング 387
- ユーザー・マッピング・オプション 356
- ライブラリー・ファイル 17
- ラッパー・オプション 356
- ラッパー・ライブラリー名 229
- 列オプション 356
- CREATE NICKNAME ステートメントの例 236
- CREATE SERVER ステートメント 231
- djxlinkTeradata スクリプト 17

Teradata データ・ソース

- アクセスの構成 221
- アクセス・ロギングの使用不可 239
- 環境変数 13, 224
- 環境変数の設定 223
- クライアント構成のテスト 221
- サーバー定義の登録 229
- 接続のテスト 221
- 接続のトラブルシューティング 233
- チューニング 239
- デフォルトの順方向データ・タイプ・マッピング 399, 412
- 登録 228
- ニックネーム 235
- ニックネームの登録 235
- ニックネームのトラブルシューティング 238
- フェデレーテッド・サーバーへの接続のテスト 233
- 文字セットのトラブルシューティング 227, 238
- ユーザー・マッピングの定義 232
- ライブラリー・ファイル 15
- ランタイム・リンクの検査 222
- CREATE NICKNAME ステートメントのトラブルシューティング 237
- CREATE USER MAPPING の例 232

U

Unicode

- コード・ページ 23, 24
- コード・ポイント・サイズ 24
- サポートされるデータ・ソース 295
- フェデレーテッド・システムのサポート 21
- MSSQL および ODBC ラッパー 23

UPDATE ステートメント

- 例
- UPDATE ステートメント、BioRS 63
- 例、BioRS ニックネーム・カーディナリティー 63
- 例、BioRS 列カーディナリティー 64
- 例、BioRS _ID_ 列カーディナリティー 66
- UPDATE または DELETE 操作エラー
- Teradata データ・ソースに関するトラブルシューティング 238

W

Web サービス

- サーバー定義 248
- サーバー・オプション 363
- サポートされるフェデレーテッド・インターチャージャー 295
- サンプルの照会 264
- セキュリティ 249
- データ・タイプ、サポートされる 418
- ニックネーム 239, 251
- ニックネーム・オプション 363
- フェデレーテッド・ビュー 260
- プロキシ・サーバー、サーバー定義の例 248
- ユーザー・マッピング・オプション 363
- ラッパー・オプション 363
- ラッパー・ライブラリー・ファイル 247
- 例 261
- 列オプション 363
- CREATE SERVER ステートメントの例 248
- SOAP ヘッダー・アクセス 257
- Web サービス記述言語 (WSDL)
- ニックネーム 239
- ニックネームへのマッピング 259
- Web サービス・データ・ソース
- ニックネームの登録 (DB2 コマンド行) 250
- Web サービス・ラッパー
- 作成 246
- データ・ソース 239
- WSSCRIPT.ARGS 関数 178

X

XML

- サーバー・オプション 371
- データ・タイプ、サポートされる 418
- ニックネーム・オプション 371

XML (続き)

- ユーザー・マッピング・オプション 371
- ラッパー・オプション 371
- 列オプション 371
- XML スキーマ
- ニックネームへのマッピング 259
- XML データ・ソース 267
- 概要 268
- 概要、ニックネーム 279
- コスト・モデル機能 281
- サーバー定義、登録 274
- 照会、例 292
- 照会の最適化 281
- 照会の最適化、概要 288
- 照会の最適化、コスト・モデル機能 291
- 照会の最適化、フェデレーテッド・ビュー 289
- 前提条件 271
- タブル 279
- デフォルト・ディレクトリー・パス 272
- ニックネーム 281
- ニックネーム、データ関連 279
- ニックネーム、登録 284
- ニックネーム・オプション
- ネーム・スペース 282
- XML 文書の構文解析 279
- ニックネーム・オプション、照会の最適化 281
- ネーム・スペース 282
- フェデレーテッド・システムへの追加 271
- プロキシ・サーバー 278
- プロキシ・サーバー、サーバー定義の例 275
- プロキシ・サーバー、ラッパーの例 273
- ラッパー、登録 272
- ラッパー・ライブラリー名 272
- CREATE NICKNAME ステートメントの例 284
- CREATE SERVER ステートメントの例 275
- CREATE VIEW ステートメントの例 290
- CREATE WRAPPER ステートメント 273
- XPath 式 279
- XML ニックネーム
- フェデレーテッド・ビュー 289
- XPath
- ニックネームでの使用 259



Printed in Japan

SA88-4832-00



日本アイ・ビー・エム株式会社
〒103-8510 東京都中央区日本橋箱崎町19-21

Spine information:

IBM InfoSphere Federation Server

バージョン 10.1

フェデレーテッド・データ・ソースの構成ガイド

