

IBM® DB2® Life Sciences Data Connect



計画、インストールおよび構成のガイド

バージョン 8

IBM® DB2® Life Sciences Data Connect



計画、インストールおよび構成のガイド

バージョン 8

ご注意!

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

本マニュアルに関するご意見やご感想は、次の URL からお送りください。今後の参考にさせていただきます。

<http://www.ibm.com/jp/manuals/main/mail.html>

なお、日本 IBM 発行のマニュアルはインターネット経由でもご購入いただけます。詳しくは

<http://www.ibm.com/jp/manuals/> の「ご注文について」をご覧ください。

(URL は、変更になる場合があります)

原典： GC27-1235-00
IBM® DB2® Life Sciences Data Connect
Planning, Installation, and Configuration Guide
Version 8

発行： 日本アイ・ビー・エム株式会社

担当： ナショナル・ランゲージ・サポート

第1刷 2002.10

この文書では、平成明朝体™W3、平成明朝体™W9、平成角ゴシック体™W3、平成角ゴシック体™W5、および平成角ゴシック体™W7を使用しています。この(書体*)は、(財)日本規格協会と使用契約を締結し使用しているものです。フォントとして無断複製することは禁止されています。

注* 平成明朝体™W3、平成明朝体™W9、平成角ゴシック体™W3、
平成角ゴシック体™W5、平成角ゴシック体™W7

© Copyright International Business Machines Corporation 2001, 2002. All rights reserved.

© Copyright IBM Japan 2002

目次

本書について	v	表構造ファイル・ラッパーに関するラッパーの制限と考慮事項	23
本書の対象読者	v	表構造ファイル・ラッパーに関するファイルの制限と考慮事項	24
バージョン 8 の新機能	v	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	24
オンライン情報	vi	表構造ファイル・ラッパーに関する最適化のヒントと考慮事項	25
規則	vii	表構造ファイル・ラッパーのメッセージ	25
構文図の読み方	vii		
第 1 章 DB2 Life Sciences Data Connect とは ?	1		
DB2 Life Sciences Data Connect	1		
IBM Life Sciences DiscoveryLink	2		
第 2 章 DB2 Life Sciences Data Connect のインストール	5		
DB2 Life Sciences Data Connect のインストール	5		
DB2 Life Science Data Connect をインストールする前に	7		
DB2 Life Science Data Connect を AIX、HP-UX、Linux、および Solaris オペレーティング環境のサーバーにインストールする	7		
DB2 Life Science Data Connect を Windows サーバーにインストールする	9		
DB2 Life Sciences Data Connect をインストールした後	10		
第 3 章 データ・ソースとしての表構造ファイル	13		
表構造ファイルとは ?	13		
表構造ファイルのタイプ	13		
ソート・ファイル	14		
ソートされていないファイル	14		
DB2 Life Science Data Connect が表構造ファイル処理する方法	14		
連合システムへの表構造ファイルの追加	16		
表構造ファイル・ラッパーの登録	16		
表構造ファイル・ラッパーの DB2_DJ_COMM 環境変数の設定	17		
表構造ファイルのサーバーの登録	18		
表構造ファイルのニックネームの登録	19		
		第 4 章 データ・ソースとしての Documentum	31
		Documentum とは ?	31
		連合システムへの Documentum の追加	33
		Documentum クライアント・ライブラリーへのリンク (AIX および Solaris オペレーティング環境のみ)	34
		Documentum のクライアント dmcl.ini ファイルを指す	35
		Documentum ラッパーの登録	36
		Documentum ラッパーの DB2_DJ_COMM 環境変数の設定	37
		Documentum データ・ソースのサーバーの登録	38
		引き数	38
		オプション	39
		ユーザーのマッピング (Documentum ラッパー)	40
		Documentum データ・ソースのニックネームの登録	41
		列オプション	42
		ニックネーム列オプション	42
		ニックネーム・オプション	43
		疑似列とは	44
		CREATE NICKNAME の例	48
		Documentum データ・ソースのカスタム関数の登録	51
		カスタム関数のストリング引き数の規則	51
		照会でのカスタム関数の使用	52
		カスタム関数表	52
		Documentum データ・ソースに対する照会の実行	57

Documentum ラッパー用の CreateNicknameFile ユーティリティとは？	58	引き数	103
CreateNicknameFile ユーティリティのインス トール (Documentum ラッパー)	59	オプション	103
CreateNicknameFile ユーティリティの構成 (Documentum ラッパー)	60	BLAST データ・ソースのニックネームの登 録	104
DM_ID オブジェクト・タイプの Documentum 登録済み表へのマッピング	61	ニックネーム列オプション	105
繰り返し属性の二重の定義 (Documentum ラッ パー)	62	ニックネーム・オプション	106
Documentum ラッパーの制限と考慮事項	63	定義行構文解析	106
Documentum ラッパーのアクセス・コントロ ール	64	固定列	107
Documentum ラッパーのメッセージ	64	CREATE NICKNAME の例	111
第 5 章 データ・ソースとしての Excel	73	BLAST SQL 照会の作成	112
Excel とは？	73	サンプル BLAST 照会	113
Excel ラッパーの前提条件	75	BLAST ラッパーの最適化のヒント	115
連合システムへの Excel の追加	75	BLAST ラッパーのメッセージ	115
Excel ラッパーの登録	75	第 7 章 データ・ソースとしての XML	119
Excel データ・ソースのサーバーの登録	76	XML とは？	119
引き数の定義	76	連合システムへの XML の追加	123
Excel データ・ソースのニックネームの登録 CREATE NICKNAME 構文 (Excel 用)	77	XML ラッパーの登録	123
オプションの定義	78	XML ラッパーの DB2_DJ_COMM 環境変数 の設定	124
Excel データ・ソースに対する照会の実行	79	XML データ・ソースのサーバーの登録	125
サンプル Excel ラッパーのシナリオ	79	XML データ・ソースのニックネームの登録 非ルート・ニックネームの連合ビューの作成 (XML ラッパー)	126 130
Excel ラッパーのラッパーの制限	81	XML データ・ソースに対する照会の実行	132
Excel ファイルの制限	82	XML ラッパーの制限と考慮事項	134
Excel ラッパーのファイル・アクセス・コント ロール・モデル	82	XML ラッパーのメッセージ	134
Excel ラッパーのメッセージ	83	第 8 章 ニックネームのコスト計算オプショ ンの指定	143
第 6 章 データ・ソースとしての BLAST	91	第 9 章 ニックネームの変更	145
BLAST とは？	91	ニックネームの変更	145
連合システムへの BLAST の追加	96	データ・タイプの変更	145
blastall 実行可能ファイルとマトリックス・フ ァイルの正しいバージョンがインストールさ れていることの検証	97	ニックネーム・オプションの変更	146
BLAST デーモンの構成	97	特記事項	147
BLAST デーモンの開始	100	商標	150
BLAST ラッパーの登録	101	参考文献	153
BLAST ラッパーの DB2_DJ_COMM 環境変 数の設定	102	索引	155
BLAST データ・ソースのサーバーの登録	103	IBM と連絡をとる	157
		製品情報	157

本書について

本書は、以下のような内容を含みます。

- DB2 Life Sciences Data Connect の紹介、および DB2 Life Sciences Data Connect が、生命科学のために特に作られた総合的なソフトウェアとサービスのセット、IBM Life Sciences DiscoveryLink オファリングにおいて占める位置
- DB2 Life Sciences Data Connect のインストールの方法
- ラッパーを登録して連合システムにデータ・ソースを加える方法。ラッパーとは、ユーザーやアプリケーションが SQL ステートメントを使用してデータ・ソースと通信することを可能にするモジュールのことです。

テキストに技術的変更がある場合、それは変更箇所の左側に縦線で示されます。

本書の対象読者

本書は、生命科学の研究開発用のデータのための連合データベース環境構築の任に当たる管理者、およびそのような環境においてアプリケーションを開発するアプリケーション・プログラマーを対象として作られています。

バージョン 8 の新機能

DB2 Life Sciences Data Connect バージョン 8 は、新しい機能として以下のものを含みます。

全体として

- ラッパーのライブラリー名が更新されました。
- 非リレーショナルのデータ・ソースのニックネームのコスト計算オプションが追加されました。

非リレーショナルのデータ・ソースの照会プラン作成の拡張

DB2 Life Sciences Data Connect ラッパーは、グローバルな照会プラン作成プロセスを扱うようにプログラムを書き換え、サポートされるデータ・ソースの 1 つに対する照会のために開発されたアクセス・ストラテジーを強化しました。この新しいプラン作成機能は、非リレーショナル・ラッパーに送られる照会のパフォーマンスを向上させます。

XML ラッパー

XML ラッパーが追加されました。これにより XML に対する連合アクセスが可能になります。XML も、DB2 Universal Database バージョン 7 に初めて

取り入れられて以来拡大を続ける非リレーショナル・ラッパーのリストに加わります。他に、BLAST、Documentum、Excel、表構造ファイルなどがあります。

表構造ファイル・ラッパー

- TYPE、VERSION および NODE サーバー・オプションは必要でなくなりました。
- SORTED ニックネーム・オプションが加えられました。

Documentum ラッパー

- ALL_VALUES ニックネーム・オプションが加えられました。
- バージョン 7 の下記のカスタム関数は、ニックネーム疑似列となりました。
 - GET_FILE
 - GET_FILE_DEL
 - GET_RENDITION
 - GET_RENDITION_DEL
 - HITS
 - SCORE
- RENDITION_FORMAT カスタム関数が加えられました。

Excel ラッパー

- Excel97 および Excel2000 の両方のデータ・ソース用に必要なラッパーは、1 つのみです。
- TYPE、VERSION および NODE サーバー・オプションは必要でなくなりました。

オンライン情報

このセクションでは、当製品に関連する Web アドレスと e-mail アドレスを記載しています。

<http://www.ibm.com/software/data/db2/lifesciencesdataconnect/>

DB2 Life Sciences Data Connect product Web サイト

<http://www.ibm.com/solutions/lifesciences/discoverylink.html>

DiscoveryLink Web サイト

<http://www.ibm.com/solutions/lifesciences/>

IBM Life Sciences Web サイト

ls@us.ibm.com

IBM Life Sciences email アドレス

規則

本書では、以下のような書体の規則を使用しています。

太字体

コマンドやグラフィカル・ユーザー・インターフェース (GUI) コントロール (たとえば、フィールド名、フォルダー名、メニュー選択) を示します。

モノスペース体

コーディングやユーザーが入力するテキストの例を示します。

イタリック体

値をもって置きかえる変数を示します。イタリック体は、また資料の表題や強調語も示します。

英大文字体

SQL キーワードやオブジェクト名 (たとえば、表、ビュー、およびサーバー) を示します。

構文図の読み方

本書では、構文は次のように定義された構造を用いて記述されています。

構文図は、左から右、上から下に、線に沿って読みます。

▶— シンボルは、ステートメントの開始を示します。

—▶ シンボルは、構文図が次の行に続くことを示します。

▶— シンボルは、ステートメントが前の行から続いていることを示します。

—▶▶ 記号は、ステートメントの終わりを示します。

必須項目は、水平線 (メインパス) の線上に示されます。

▶▶—STATEMENT—必須項目—▶▶

オプション項目は、メインパスの下に示されます。

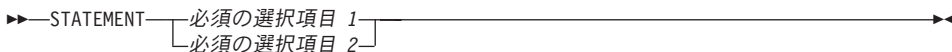
▶▶—STATEMENT—
└ オプション項目 ┘▶▶

オプション項目がメインパスの上に示されている場合、その項目は読みやすくするために示されているのであり、ステートメントの実行には影響を与えません。

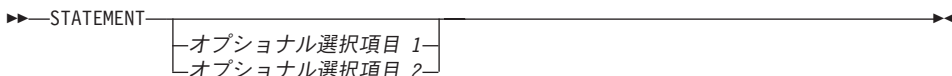
▶▶—STATEMENT—
┌ オプション項目 ┐▶▶

2 つ以上の項目からの選択が可能な場合は、複数の項目が縦に並んでスタックに記載されます。

項目から 1 つを選択しなければならない場合、スタック内の項目の 1 つがメインパスの線上に示されます。



項目から何も選択しなくてもよい場合は、スタック全体がメインパスの下に表示されません。



項目のうちの 1 つがデフォルトの場合、それは、メインパスの上に表示され、残りの選択項目はその下に表示されます。



メインパスの上の左に折り返す矢印は、項目が繰り返されることを示しています。この場合、繰り返し項目は 1 つ以上の空白で区切る必要があります。



繰り返しの矢印がコンマを含む場合は、繰り返し項目をコンマで区切る必要があります。

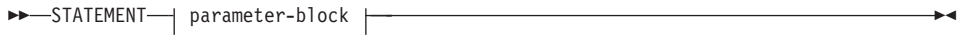


繰り返しの矢印がスタックの上にある場合、スタック項目の中から 1 つ以上選択するか、1 つの選択項目を繰り返すことができることを示しています。

キーワードは大文字で表記されています (たとえば、FROM)。キーワードの記載は、表記どおりでなければなりません。変数は小文字で表記されています (たとえば、column-name)。変数は、構文中でのユーザー指定の名前または値を表します。

句読記号、括弧、算術演算子、または同様の他のシンボルが示されている場合、それらを構文の一部として入力しなければなりません。

単一の変数が、いくつかのパラメーターのセットを表していることがあります。たとえば、下記の図で、変数 `parameter-block` は、**parameter-block** という見だしの下の内容で置きかえられます。



parameter-block:



「黒丸」 (●) の間の隣り合うセグメントは、どのような順序で指定してもかまいません。



上記の図は、`item2` と `item3` は、どちらを先に指定してもかまわないことを示しています。次のいずれも有効です。

```
STATEMENT item1 item2 item3 item4
STATEMENT item1 item3 item2 item4
```


第 1 章 DB2 Life Sciences Data Connect とは ?

この章では、DB2 Life Sciences Data Connect 製品の概要を説明します。この製品は、IBM Life Sciences DiscoveryLink オファリングであり、ライフ・サイエンス・データ照会システムの構築に必要な汎用のステップとなります。

DB2 Life Sciences Data Connect

IBM® DB2® Life Science Data Connect は、さまざまなソースからの遺伝子学、化学、生物学、および他の研究データを DB2 連合システムによって統合することを可能にします。DB2 連合システムとは、DB2 Universal Database™ サーバーと、DB2 Universal Database サーバーがデータを検索する複数のデータ・ソースとから構成される分散コンピューティング・システムのことです。

連合システムを使用することによって、ユーザーまたはアプリケーションは SQL ステートメントを使用して、複数の異種データ・ソース (たとえば、IBM、Oracle、Sybase、および Microsoft などのリレーショナル・データベース、さらには表構造ファイルなどの非リレーショナル・データ・ソース) に存在するデータを照会、検索、結合することができます。図 1 は、DB2 Life Sciences Data Connect を使用して研究データの複数のソースにアクセスする連合システムを示しています。

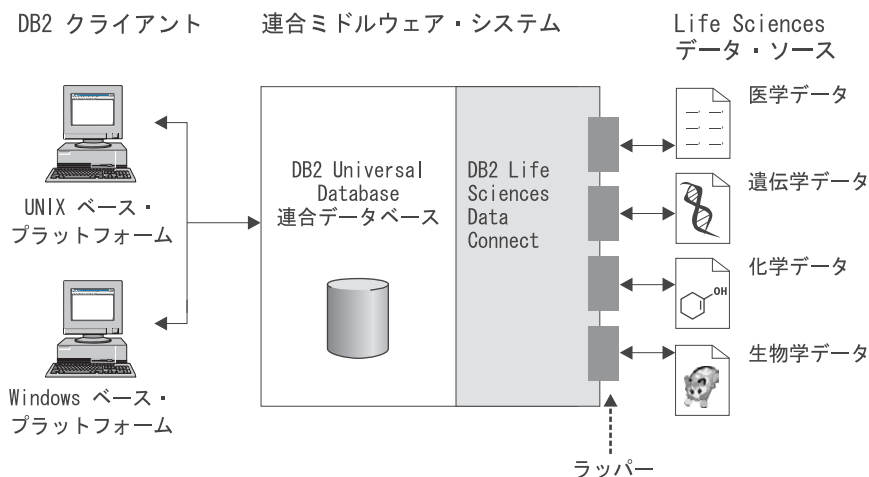


図 1. DB2 Life Sciences Data Connect による Life Sciences データのアクセス

DB2 連合システムには、クライアント、クライアントが照会を発信するデータベース(連合データベースと呼ばれる)、連合データベースがデータ・ソースと通信するためのインターフェースおよびデータ・ソースそのものが含まれます。

連合サーバーがデータ・ソースと通信するメカニズムのことをラッパー と呼びます。ラッパーをインプリメントするために、サーバーはラッパー・モジュール と呼ばれるライブラリーに保管されているルーチンを使用します。サーバーは、これらのルーチンによって、繰り返しデータ・ソースへ接続してそこからデータを検索するような操作を実行できるようにします。

連合システムがセットアップされると、複数のデータ・ソースに保管されている情報を、1 つの大きなデータベースに保管されているかのようにアクセスすることができます。ユーザーおよびアプリケーションは、1 つの連合データベースに照会を発信することによって、複数のデータ・ソースからデータを検索できます。アプリケーションは、他の DB2 データベースと同様に連合データベースを使用して機能します。

連合システムに関して詳しくは、「DB2 SQL リファレンス」を参照してください。

関連概念:

- 2 ページの 『IBM Life Sciences DiscoveryLink』

IBM Life Sciences DiscoveryLink

DiscoveryLink オファリングは、複数の異種データ・ソースに置かれているデータを統合するために、ライフ・サイエンスの研究および開発の要件に合わせて特別に調整したミドルウェア・ソフトウェアとサービスのセットです。

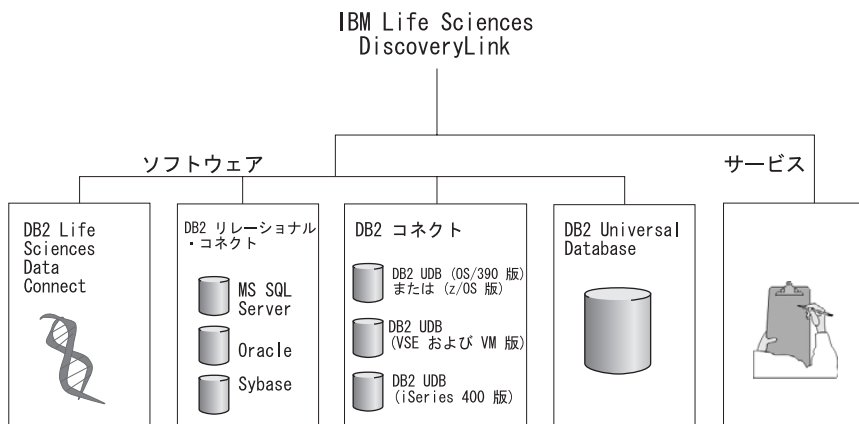


図2. IBM Life Sciences DiscoveryLink

たとえば、DiscoveryLink を利用することによって、1 つの SQL ステートメントを使うだけで、スイスの Oracle データベースに保管されているタンパク質の配列のデータ、日本の Sybase データベースに保管されている化学構造のデータ、そしてユーザー・ローカル・エリア・ネットワークの表構造のフラット・ファイルに保管されている分光データを組み合わせたことができます。データは、1 つの仮想データベースに保管されているように見えます。

ソフトウェアは以下のコンポーネントによって構成されます。

DB2[®] Life Sciences Data Connect

Life Sciences Data にアクセスする

DB2 Relational Connect

Oracle、Sybase、および Microsoft[®] のリレーショナル・データベースにアクセスします。DB2 Relational Connect の詳細については、「[連合システム・ガイド](#)」を参照してください。

DB2 Connect[™]

ホスト・システム上の DB2 データベース・サーバーにアクセスします。DB2 Connect に関して詳しくは、「[DB2 Connect ユーザーズ・ガイド](#)」を参照してください。

DB2 Universal Database[™]

照会を最適化して、複数の異種データ・ソースからの結果を統合します。DB2 Universal Database に関して詳しくは、「[DB2 管理ガイド](#)」を参照してください。

DiscoveryLink ソフトウェアおよびサービスの詳細については、下記の関連リンク・セクションの『[オンライン情報](#)』を参照してください。

関連概念:

- 1 ページの『[DB2 Life Sciences Data Connect](#)』

第 2 章 DB2 Life Sciences Data Connect のインストール

この章では、各ラッパーごとにサポートされるプラットフォーム、Unix ベースと Windows ベースのラッパーのインストールの方法、インストールが完了した後ご使用のシステムに置かれるライブラリーについて詳しく説明します。

DB2 Life Sciences Data Connect のインストール

ライフ・サイエンス・データの照会及び検索のために DB2 Life Sciences Data Connect を使用するには、ラッパーをインストールし、次にそれぞれのラッパーを登録して連合システムにこれを追加する必要があります。

ラッパーは、DB2 Universal Database が要求するもの以上の特別な要件はありません。また、DB2 Universal Database がサポートするすべてのシステム構成で実行されます。

表 1 は、各オペレーティング・システムでの DB2 Life Sciences Data Connect ラッパーを示しています。各ライフ・サイエンス・ラッパーを登録するための説明は、下記の関連リンク・セクションにリストされているトピックの中にあります。

表 1. オペレーティング・システム別の DB2 Life Science Data Connect ラッパー

ラッパー	Windows	AIX	HP-UX	Linux	Solaris オペレーティング環境
表構造ファイル	X	X	X	X	X
Documentum	X	X			X
Excel	X				
BLAST	X	X		X	X
XML	X	X	X	X	X

インストール・プロセスに際して、次のようなユーザーが選択する 3 つのインストール可能なコンポーネントがあります。Scientific、Structured files および Applications です。各インストール可能コンポーネントおよび各コンポーネントに組み込まれているラッパーのリストが 6 ページの表 2 に示されています。

表 2.

インストール可能 コンポーネント名	説明	組み込まれている ラッパー
Scientific	Scientific データ・ソースは、その対象をゲノム、プロテオーム、生物情報学、および化学情報学についての情報を含んでいるライフ・サイエンス産業のみに限定して開発されています。	BLAST
Structured files	Structured file データ・ソースには、定義済みの、繰り返し可能な構造をもつファイルに保管されているライフ・サイエンス・データが含まれています。	表構造ファイル、Excel、XML
Applications	Application データ・ソースは、アプリケーションを使用して、その下にあるライフ・サイエンス・データにアクセスします。ロー・データはいくつかの標準のフォーマットや非標準のフォーマットになっていてもかまいません。	Documentum

手順:

DB2 Life Science Data Connect をインストールするには、次のステップに従ってください。

1. DB2 Life Sciences Data Connect をインストールする前に。
2. DB2 Life Science Data Connect を AIX、HP-UX、Linux、および Solaris オペレーティング環境のサーバーにインストールする
3. DB2 Life Science Data Connect を Windows サーバーにインストールする
4. DB2 Life Science Data Connect をインストールした後。

関連作業:

- 7 ページの 『DB2 Life Science Data Connect をインストールする前に』
- 7 ページの 『DB2 Life Science Data Connect を AIX、HP-UX、Linux、および Solaris オペレーティング環境のサーバーにインストールする』
- 9 ページの 『DB2 Life Science Data Connect を Windows サーバーにインストールする』
- 10 ページの 『DB2 Life Sciences Data Connect をインストールした後』
- 59 ページの 『CreateNicknameFile ユーティリティのインストール (Documentum ラッパー)』

DB2 Life Science Data Connect をインストールする前に

このタスクは、DB2 Life Sciences Data Connect のインストール のためのメインタスクの一部です。

手順:

連合サーバーに DB2 Life Sciences Data Connect をインストールする前に、以下を行う必要があります。

- 連合サーバーに DB2 Universal Database Enterprise Server Edition がインストールされていることを確認します。
- データベースの連合データベース・システム・サポートがオンになっていることを確認します。この設定を確認するには、DB2 コマンド行プロセッサで以下に示すコマンドを実行します。

```
GET DATABASE MANAGER CONFIGURATION
```

このコマンドは、すべてのデータベース・パラメーターとそれぞれの現在の設定を表示します。FEDERATED パラメーターが YES に設定されていることを確認してください。

FEDERATED パラメーターが NO に設定されている場合、DB2 コマンド行プロセッサで以下に示すコマンドを実行します。

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

このタスクの順序における次のタスクは『DB2 Life Science Data Connect を AIX、HP-UX、Linux、および Solaris オペレーティング環境のサーバーにインストールする』です。

関連作業:

- 9 ページの 『DB2 Life Science Data Connect を Windows サーバーにインストールする』
- 10 ページの 『DB2 Life Sciences Data Connect をインストールした後』
- 7 ページの 『DB2 Life Science Data Connect を AIX、HP-UX、Linux、および Solaris オペレーティング環境のサーバーにインストールする』

DB2 Life Science Data Connect を AIX、HP-UX、Linux、および Solaris オペレーティング環境のサーバーにインストールする

このタスクは、DB2 Life Sciences Data Connect のインストール のためのメインタスクの一部です。

前提条件:

下記の関連タスク・セクションの『DB2 Life Science Data Connect をインストールする前に』を参照してください。

手順:

AIX、HP-UX、Linux、および Solaris オペレーティング環境の連合サーバーに DB2 Life Sciences Data Connect をインストールするには、db2setup ユーティリティを使用します。

注: db2setup ユーティリティを使用するときに表示される画面は、連合サーバーにインストールされているソフトウェア製品によって異なります。以下の手順は、DB2 Life Sciences Data Connect がインストールされていない場合を想定しています。

1. root 権限を持つユーザーとしてログインします。
2. DB2 Life Sciences Data Connect CD-ROM を挿入してマウントします。CD-ROM をマウントする方法については、「DB2 for UNIX 概説およびインストール」を参照してください。
3. **cd /cdrom** コマンドを入力することによって、CD-ROM がマウントされているディレクトリに変更します (cdrom は製品 CD-ROM のマウント・ポイントです)。
4. 以下に示すコマンドを入力します。

```
./db2setup
```

「DB2 セットアップ (DB2 Setup)」ウィンドウが表示されます。

5. セットアップ・プログラムが表示するプロンプトの指示に従います。

インストールが完了すると、DB2 Life Sciences Data Connect が他の DB2 製品とともに以下のディレクトリにインストールされます。

- DB2 (AIX 版) サーバーの場合、ディレクトリは /usr/opt/db2_08_01 です。
- DB2 (Solaris 版) サーバーの場合、ディレクトリは /opt/IBM/db2/V8.1 です。
- DB2 (HP-UX 版) サーバーの場合、ディレクトリは /opt/IBM/db2/V8.1 です。
- DB2 (Linux 版) サーバーの場合、ディレクトリは /opt/IBM/db2/V8.1 です。

ここでのタスクの順序では、次のタスクは『DB2 Life Sciences Data Connect を Windows サーバーにインストールする』です。

関連作業:

- 7 ページの 『DB2 Life Science Data Connect をインストールする前に』
- 9 ページの 『DB2 Life Science Data Connect を Windows サーバーにインストールする』
- 10 ページの 『DB2 Life Sciences Data Connect をインストールした後』

DB2 Life Science Data Connect を Windows サーバーにインストールする

このタスクは、DB2 Life Sciences Data Connect のインストール のためのメインタスクの一部です。

前提条件:

下記の関連タスク・セクションの『DB2 Life Science Data Connect をインストールする前に』を参照してください。

手順:

DB2 Life Sciences Data Connect を Windows 連合サーバーにインストールするには、セットアップ・プログラムを使用します。

1. すでに作成したユーザー・アカウントで連合サーバーにログオンし、DB2 Universal Database のインストールを実行します。
2. 実行中のすべてのプログラムを終了して、セットアップ・プログラムが必要なファイルを更新できるようにします。
3. セットアップ・プログラムを起動します。セットアップ・プログラムは、自動でも手動でも起動できます。セットアップ・プログラムが自動的に開始されない場合、またはセットアップを異なる言語で実行したい場合、セットアップ・プログラムを手動で起動します。

- セットアップ・プログラムを自動的に起動するには、DB2 Life Sciences Data Connect CD をドライブに挿入します。自動実行機能によって、セットアップ・プログラムが自動的に開始されます。システムの言語が判別され、その言語でセットアップ・プログラムが起動されます。
- セットアップ・プログラムを手動で起動するには、以下のようになります。
 - a. 「スタート」をクリックして、「ファイル名を指定して実行」をクリックします。
 - b. 「名前」フィールドに、以下のコマンドを入力します。

```
x:%setup /i language
```

ここでは、それぞれ以下のことを意味しています。

x: CD-ROM ドライブです。

language

使用する言語のコードです (たとえば、英語の場合は EN)。

- c. 「OK」をクリックします。

インストール画面が表示されます。

4. 「インストール (Install)」をクリックして、インストール・プロセスを開始します。
5. セットアップ・プログラムが表示するプロンプトの指示に従います。

インストールが完了すると、DB2 Life Sciences Data Connect は他の DB2 製品とともにインストール・ディレクトリーにインストールされます。デフォルトのインストール・ディレクトリーは、C:\Program Files\IBM\SQLLIB です。

ここでのタスクの順序では、次のタスクは『DB2 Life Science Data Connect をインストールした後』です。

関連作業:

- 7 ページの 『DB2 Life Science Data Connect をインストールする前に』
- 10 ページの 『DB2 Life Sciences Data Connect をインストールした後』
- 7 ページの 『DB2 Life Science Data Connect を AIX、HP-UX、Linux、および Solaris オペレーティング環境のサーバーにインストールする』

DB2 Life Sciences Data Connect をインストールした後

このタスクは、DB2 Life Sciences Data Connect のインストール のためのメインタスクの一部です。インストールが完了すると、ラッパー・ライブラリー・ファイルがシステムに保管されます。これらのライブラリーは、ラッパーの登録プロセスの間に使用されます。

手順:

インストールの検証をするためには、インストール・ディレクトリーをチェックして、デフォルトのラッパー・ライブラリーがあるかどうかを確認してください。

各ライブラリーのデフォルト・ファイル名は、サポートされているオペレーティング・システム別に、Windows プラットフォームについては 表 3 に、UNIX プラットフォームについては 11 ページの表 4 にリストされています。

表 3. Windows プラットフォームのデフォルト・ラッパー・ライブラリー名

ラッパー	Windows
表構造ファイル	db2lsfile.dll
Documentum	db2lsdctm.dll
Excel97 / Excel2000	db2lsxls.dll
BLAST	db2lsblast.dll
XML	db2lsxml.dll

11 ページの表 4 は、サポートされている UNIX プラットフォームでのラッパー・ライブラリー名をリストしています。

表 4. UNIX プラットフォームごとのデフォルト・ラッパー・ライブラリー名

ラッパー	AIX	HP-UX	Linux	Solaris オペレーティング環境
表構造ファイル	libdb2lsfile.a	libdb2lsfile.sl	libdb2lsfile.so	libdb2lsfile.so
Documentum	libdb2lsdctm.a			libdb2lsdctm.so
BLAST	libdb2lsblast.a		libdb2lsblast.so	libdb2lsblast.so
XML	libdb2lsxml.a	libdb2lsxml.sl	libdb2lsxml.so	libdb2lsxml.so

注: Windows を除くすべてのプラットフォームにおける Documentum については、これらのライブラリーは Documentum クライアント・ライブラリーにリンク・エディットされた後に作成されます。インストール後にユーザーのシステムに入れられたファイルは、AIX では libdb2lsSTdctmF.a、Solaris オペレーティング環境では libdb2lsSTdctmF.so と名前が付けられます。

ここでの作業は、これで終わりです。

関連作業:

- 7 ページの 『DB2 Life Science Data Connect をインストールする前に』
- 9 ページの 『DB2 Life Science Data Connect を Windows サーバーにインストールする』
- 16 ページの 『連合システムへの表構造ファイルの追加』
- 7 ページの 『DB2 Life Science Data Connect を AIX、HP-UX、Linux、および Solaris オペレーティング環境のサーバーにインストールする』
- 33 ページの 『連合システムへの Documentum の追加』
- 75 ページの 『連合システムへの Excel の追加』
- 96 ページの 『連合システムへの BLAST の追加』
- 123 ページの 『連合システムへの XML の追加』

第 3 章 データ・ソースとしての表構造ファイル

この章では、表構造ファイルとは何か、また表構造ファイルをご使用の連合システムにデータ・ソースとして加える方法などについて解説し、さらに表構造ファイルのラッパーに関連したエラー・メッセージをリストします。

表構造ファイルとは？

表構造ファイルは通常、一連のレコードによって構成されます。各レコードには同じ数のフィールドがあり、任意の区切り文字によって区切られています。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。

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
```

最初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。

関連概念:

- 13 ページの 『表構造ファイルのタイプ』
- 14 ページの 『DB2 Life Science Data Connect が表構造ファイルを処理する方法』
- 31 ページの 『Documentum とは ?』
- 73 ページの 『Excel とは ?』
- 91 ページの 『BLAST とは ?』
- 119 ページの 『XML とは ?』

関連作業:

- 16 ページの 『連合システムへの表構造ファイルの追加』

表構造ファイルのタイプ

表構造ファイルは、ソートされたものでもソートされていないものでもかまいません。

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、ドラッグごとに固有であるため主キーです。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 にはソートされていない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。

```
332,DrugnameB,Manufacturer2
234,DrugnameA,Manufacturer1
333,DrugnameC,Manufacturer2
```

ラッパーは、ソートされていないデータ・ファイルよりもソートされたデータ・ファイルの方が、効率よく検索できます。

関連概念:

- 13 ページの 『表構造ファイルとは ?』
- 14 ページの 『DB2 Life Science Data Connect が表構造ファイル进行处理する方法』

関連作業:

- 16 ページの 『連合システムへの表構造ファイルの追加』

DB2 Life Science Data Connect が表構造ファイル进行处理する方法

DB2 Life Sciences Data Connect は、ラッパーと呼ばれるモジュールを使用することによって、表構造ファイルにあるデータを照会する SQL ステートメント进行处理して、通常のリレーショナル表またはビューにあるデータのように扱うことができます。これにより、表構造ファイルにあるデータを、リレーショナル・データまたは他の表構造ファイルにあるデータと結合することができます。このプロセスは 15 ページの図 3 に図示されています。

DB2 クライアント 連合データベース

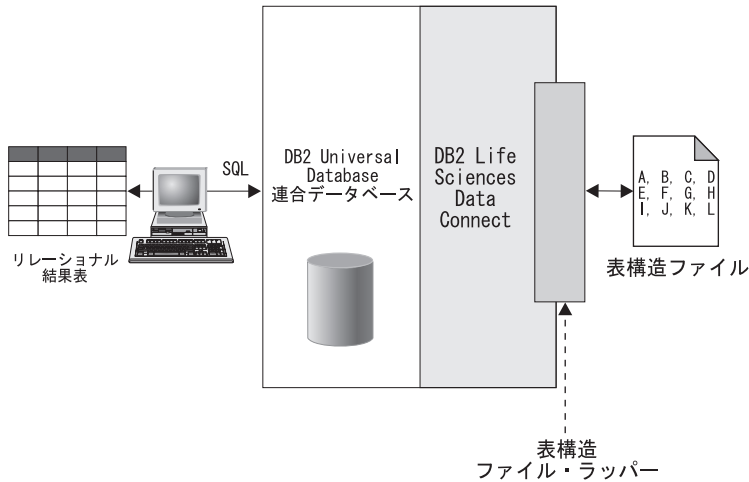


図3. 表構造ファイル・ラッパーがどのように機能するか

たとえば、表構造ファイル、DRUGDATA2.TXT が研究所のコンピューターにあるとします。このデータを照会して、使用する他のデータ・ソースの表との突き合わせを行うことは、単調で面倒な作業となります。

DB2 Life Sciences Data Connect に DRUGDATA2.TXT を登録すると、ファイルはリレーショナル・データ・ソースのようになります。その結果、ファイルを他のリレーショナル・データ・ソースおよび非リレーショナル・データ・ソースとともに照会することができ、データを同時に分析することができるようになります。

たとえば、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	Manufacturer1
332	DrugnameB	Manufacturer2
333	DrugnameC	Manufacturer2

関連概念:

- 13 ページの 『表構造ファイルとは ?』
- 13 ページの 『表構造ファイルのタイプ』

関連作業:

- 16 ページの 『連合システムへの表構造ファイルの追加』

連合システムへの表構造ファイルの追加

手順:

表構造ファイルのデータ・ソースを追加するには、以下を行う必要があります。

1. CREATE WRAPPER コマンドを使用してラッパーを登録します。
2. オプション：照会のパフォーマンスを向上させるために DB2_DJ_COMM 環境変数を設定します。
3. CREATE SERVER コマンドを使用してサーバーを登録します。
4. すべての表構造ファイルに対して CREATE NICKNAME コマンドを使用してニックネームを登録します。

コマンドは、DB2 コマンド行プロセッサから実行できます。

関連作業:

- 16 ページの 『表構造ファイル・ラッパーの登録』
- 17 ページの 『表構造ファイル・ラッパーの DB2_DJ_COMM 環境変数の設定』
- 18 ページの 『表構造ファイルのサーバーの登録』
- 19 ページの 『表構造ファイルのニックネームの登録』
- 33 ページの 『連合システムへの Documentum の追加』
- 75 ページの 『連合システムへの Excel の追加』
- 96 ページの 『連合システムへの BLAST の追加』
- 123 ページの 『連合システムへの XML の追加』

表構造ファイル・ラッパーの登録

このタスクは、連合システムへの表構造ファイルの追加のためのメインタスクの一部です。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、連合サーバーがデータ・ソースとの通信やデータ・ソースからのデータの検索に使用するメカニズムです。ラッパーは、ライブラリー・ファイルとして、システムにインストールされています。

手順:

ラッパーを登録するためには、CREATE WRAPPER ステートメントを使用して、表構造ファイルにアクセスするために使用するラッパーを指定します。

たとえば、AIX にラッパーを登録するには、以下に示すステートメントを実行します。

```
CREATE WRAPPER laboratory_flat_files LIBRARY 'libdb2lsfile.a'  
OPTIONS(DB2_FENCED 'N');
```

この例で、`laboratory_flat_files` はラッパーに付けられた名前です。この名前は、ラッパーが登録されるデータベース内で固有でなければなりません。AIX の表構造ファイル・ラッパーに必要なライブラリー名は `libdb2lsfile.a` です。

ライブラリーはデフォルトで `libdb2lsfile.a` としてインストールされますが、インストール時にカスタマイズすることができます。正しい名前については、システム管理者に確認してください。

サポートされているプラットフォーム別の表構造ファイル・ラッパーのデフォルト・ライブラリー・ファイル名の表については、下記の関連タスク・セクションの『DB2 Life Science Data Connect をインストールした後』を参照してください。CREATE WRAPPER ステートメントに関して詳しくは、「DB2 SQL リファレンス」を参照してください。

ここでのタスクの順序では、次のタスクは『表構造ファイル・ラッパーの DB2_DJ_COMM 環境変数の設定』です。

関連作業:

- 17 ページの 『表構造ファイル・ラッパーの DB2_DJ_COMM 環境変数の設定』
- 36 ページの 『Documentum ラッパーの登録』
- 75 ページの 『Excel ラッパーの登録』
- 101 ページの 『BLAST ラッパーの登録』
- 123 ページの 『XML ラッパーの登録』

表構造ファイル・ラッパーの DB2_DJ_COMM 環境変数の設定

このタスクは、連合システムへの表構造ファイルの追加 のためのメインタスクの一部です。表構造ファイルにアクセスする場合のパフォーマンスを向上させるために、DB2_DJ_COMM 環境変数を設定することができます。この変数は、初期設定時に連合サーバーがラッパーをロードするかどうかを決定します。

手順:

DB2_DJ_COMM 環境変数を設定するには、関連する CREATE WRAPPER ステートメントに指定したラッパーに対応するラッパー・ライブラリーを指定して `db2set` コマンドをサブミットします。

たとえば、次のようにします。

```
db2set DB2_DJ_COMM='libdb2lsfile.a'
```

等号 (=) の両側にスペースを入れないようにしてください。

データベース始動時にラッパー・ライブラリーをロードすることに関連したオーバーヘッドがあります。このオーバーヘッドを回避するには、アクセスしたいライブラリーのみを指定します。

DB2_DJ_COMM 環境変数に関して詳しくは、「DB2 管理ガイド」を参照してください。

ここでのタスクの順序では、次のタスクは『表構造ファイルのサーバーの登録』です。

関連作業:

- 16 ページの 『表構造ファイル・ラッパーの登録』
- 18 ページの 『表構造ファイルのサーバーの登録』
- 37 ページの 『Documentum ラッパーの DB2_DJ_COMM 環境変数の設定』
- 102 ページの 『BLAST ラッパーの DB2_DJ_COMM 環境変数の設定』
- 124 ページの 『XML ラッパーの DB2_DJ_COMM 環境変数の設定』

表構造ファイルのサーバーの登録

このタスクは、連合システムへの表構造ファイルの追加 のためのメインタスクの一部です。ラッパーが登録された後、対応するサーバーを登録する必要があります。

手順:

表構造ファイル・サーバーを連合システムに登録するには、CREATE SERVER ステートメントを使用します。たとえば、次のようにします。

```
CREATE SERVER biochem_lab WRAPPER laboratory_flat_files
```

この例では、biochem_lab は表構造ファイル・サーバーに割り当てられる名前です。名前は、サーバーが登録されるデータベースで固有でなければなりません。

CREATE SERVER ステートメントに関して詳しくは、「DB2 SQL リファレンス」を参照してください。

ここでのタスクの順序では、次のタスクは『表構造ファイルのニックネームの登録』です。

関連作業:

- 17 ページの 『表構造ファイル・ラッパーの DB2_DJ_COMM 環境変数の設定』
- 19 ページの 『表構造ファイルのニックネームの登録』
- 38 ページの 『Documentum データ・ソースのサーバーの登録』

- 76 ページの 『Excel データ・ソースのサーバーの登録』
- 103 ページの 『BLAST データ・ソースのサーバーの登録』
- 125 ページの 『XML データ・ソースのサーバーの登録』

表構造ファイルのニックネームの登録

このタスクは、連合システムへの表構造ファイルの追加のためのメインタスクの一部です。サーバーを登録した後、対応するニックネームを登録する必要があります。ニックネームは、照会の中で表構造ファイル・データ・ソースを参照する場合に使用されません。

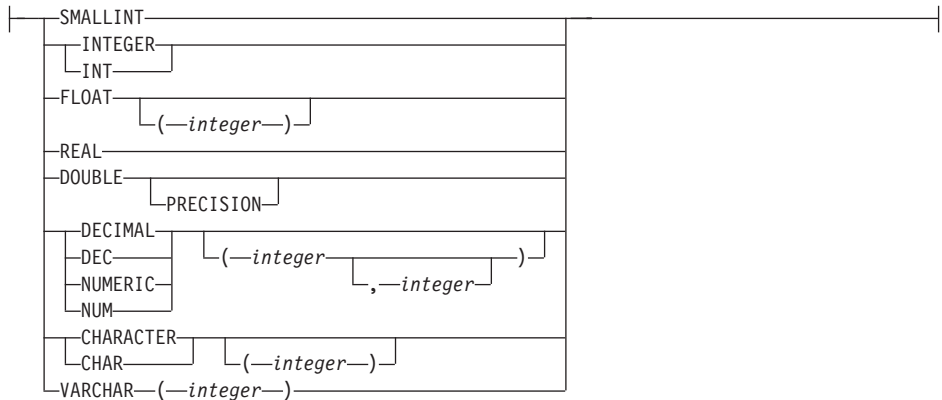
手順:

ニックネームを登録するには、アクセスしたい表構造ファイルごとに CREATE NICKNAME を使用します。

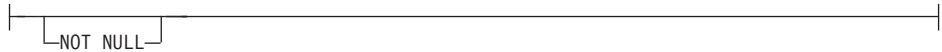
CREATE NICKNAME ステートメントの構文は以下のとおりです。



data-type:



column-option:



注:

- 1 ソートされていないファイルには許可されません。ソートされたファイルについてはオプションです。

nickname

アクセスされる表構造ファイルの固有のニックネーム。これは、登録されるスキーマ内の他のすべてのニックネーム、表、およびビューと区別される必要があります。

column-name

表構造ファイルの各フィールドに指定される固有名。各列名にそのデータ・タイプが続きます。サポートされる列タイプは、CHAR、VARCHAR、SMALLINT、INTEGER、FLOAT、DOUBLE、REAL、および DECIMAL です。

SMALLINT

短整数の場合。

INTEGER または INT

長整数の場合。

FLOAT(integer)

integer の値に応じて、短精度か倍精度の浮動小数点数となります。 *integer* の値は、1 ~ 53 となります。1 ~ 24 の値は短精度を示し、25 ~ 53 の値は倍精度を示します。

REAL 単精度の浮動小数点の場合。

DOUBLE または DOUBLE PRECISION

倍精度の浮動小数点の場合。

FLOAT 倍精度の浮動小数点の場合。

DECIMAL(*precision-integer*, *scale-integer*) または **DEC**(*precision-integer*, *scale-integer*)
10 進数の場合。

最初の整数は数値の精度です。つまり、桁の合計数です。この値は、1 ~ 31 の範囲です。

2 番目の整数は数値の位取りです。つまり、小数点の右側の桁数です。この値は、0 から数値の精度の範囲となります。

精度と位取りが指定されない場合、デフォルト値である 5,0 が使用されます。

NUMERIC および **NUM** という語は、**DECIMAL** および **DEC** の同義語として使用できます。

CHARACTER(*integer*) または **CHAR**(*integer*) または **CHARACTER** または **CHAR**
長さ *integer* の固定長文字ストリング。1 ~ 254 の範囲にすることができます。長さの指定が省略される場合、長さ 1 文字が想定されます。

VARCHAR(*integer*)
最大長 *integer* の可変長文字ストリング。1 ~ 32672 の範囲にすることができます。

NOT NULL

列に NULL 値が入れられないようにします。

server-name

CREATE SERVER ステートメントを使用してすでに登録したサーバーを指定します。 **CREATE SERVER** ステートメントの詳細については、下記の関連リンク・セクションを参照してください。このサーバーは、表構造ファイルにアクセスのに使用されます。

'path' アクセスされる表構造ファイルへの完全修飾パス (単一引用符で囲まれている)。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、パイプや非標準ファイル・タイプにすることはできません。データ・ファイルは DB2 インスタンス所有者が読むことができなければなりません。インスタンス所有者に関して詳しくは、「*DB2 管理ガイド*」を参照してください。

SORTED

データ・ソース・ファイルがソートされているのか、ソートされていないのかを指定します。このオプションは 'Y'、'y'、'n'、または 'N' のいずれも受け入れます。デフォルト値は 'N' です。

注: データ・ソースがソートされていることを指定する場合、
VALIDATE_DATA_FILE を 'Y' に設定することをお勧めします。

'delimiter'

表構造ファイルの列を区切るのに使用される区切り文字 (単一引用符で囲まれ

ている)。単一の桁区切り文字だけが許可されています。列区切り文字が定義されていない場合、列区切り文字はデフォルトのコンマになります。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、列の有効なデータとして保管することができません。たとえば、列の 1 つにコンマが組み込まれたデータが入っている場合、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。

'key-column-name'

ファイルをソートする場合のキーとなる、ファイル内の列の名前 (単一引用符で囲まれる)。このオプションはソート・ファイルでのみ使用してください。

単一系列のキーだけがサポートされます。値は、CREATE NICKNAME ステートメントに定義されている列の名前にする必要があります。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。NOT NULL オプションをニックネーム・ステートメントの中のその定義に追加することによって、キー列が、NULL にすることができないことを指定されることをお勧めします。たとえば、次のようにします。

```
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')
```

注: このオプションは大文字小文字の区別をします。ただし DB2 は、列が二重引用符によって定義されていない限り、列名を大文字に変換します。以下の例は正しく処理されません。なぜなら、empno 列は DB2 によって大文字に変換されますが、empno キー列は小文字でサブミットされるからです。したがって、キーとして指定された列は検出されません。

```
CREATE NICKNAME depart (
empno char(6) NOT NULL)
FOR SERVER DATASTORE
OPTIONS(FILE_PATH'data.txt', SORTED 'Y', KEY_COLUMN 'empno');
```

VALIDATE_DATA_FILE

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。このオプションに有効な値は 'Y' または 'N' だけ (単一引用符で囲まれる) です。このチェックは、登録時に一度だけ行われます。このオプションが指定されない場合、チェックは行われません。

以下の例は、下記の関連リンク・セクションにリストされている『表構造ファイルとは?』で説明されている表構造ファイル、DRUGDATA1.TXT の CREATE NICKNAME ステートメントを示しています。

```
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')
```

CREATE NICKNAME ステートメントについての詳細は、「DB2 SQL リファレンス」を参照してください。ニックネームに関して詳しくは、「DB2 管理ガイド」を参照してください。

ここでの作業は、これで終わりです。

関連作業:

- 10 ページの 『DB2 Life Sciences Data Connect をインストールした後』
- 18 ページの 『表構造ファイルのサーバーの登録』
- 41 ページの 『Documentum データ・ソースのニックネームの登録』
- 77 ページの 『Excel データ・ソースのニックネームの登録』
- 104 ページの 『BLAST データ・ソースのニックネームの登録』
- 126 ページの 『XML データ・ソースのニックネームの登録』
- 143 ページの 『第 8 章 ニックネームのコスト計算オプションの指定』

表構造ファイル・ラッパーに関するラッパーの制限と考慮事項

- パススルー・セッションはラッパーでは使用できません。
- 複数列キーは使用できません。
- ファイルは昇順でのみソートできます。降順でのソートはサポートされていません。
- ラッパーは NOT NULL 制約を強制しませんが、DB2 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、DB2 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。

- DB2 Universal Database Enterprise Server Edition 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。ファイルは、共通のパスを使用してすべてのノードからアクセス可能であれば、DB2 Universal Database のノードに置く必要はありません。

関連資料:

- 24 ページの 『表構造ファイル・ラッパーに関するファイルの制限と考慮事項』
- 63 ページの 『Documentum ラッパーの制限と考慮事項』

- 81 ページの 『Excel ラッパーのラッパーの制限』
- 82 ページの 『Excel ファイルの制限』
- 134 ページの 『XML ラッパーの制限と考慮事項』

表構造ファイル・ラッパーに関するファイルの制限と考慮事項

- ファイルは 1 行につき 1 レコードに限定されます。
- 各レコードには同じ数の区切られた列があります。
- 各レコードは、ラッパーがインストールされているプラットフォームごとの、標準の行終了文字で終了される必要があります。
- 列区切り文字はファイル全体で一貫しています。
- 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、区切り文字に行終了文字を続けることによって表されます。
- 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。
- ソートされるデータ・ソースは、LC_COLLATE 各国語サポート区分の設定によって定義されている現在の地域の照合順序に従って、昇順でソートされます。
- データベースのコード・ページはファイルの文字セットと一致する必要があります。そうでない場合、予期しない結果が生じる場合があります。
- マルチバイト文字の入ったファイルはサポートされません。
- 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 char の後ろに列タイプの scale パラメーターで許可されているよりも多くの桁がある場合、超過したデータは切り捨てられます。
- 行の最大の長さは 32768 です。

関連資料:

- 23 ページの 『表構造ファイル・ラッパーに関するラッパーの制限と考慮事項』
- 63 ページの 『Documentum ラッパーの制限と考慮事項』
- 82 ページの 『Excel ファイルの制限』
- 134 ページの 『XML ラッパーの制限と考慮事項』

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

データベース管理システムは、DB2 インスタンス所有者の権限によっ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。アプリケーションの許可 ID (連合データベースへの接続を確立する ID) は関係ありません。

関連資料:

- 64 ページの 『Documentum ラッパーのアクセス・コントロール』
- 82 ページの 『Excel ラッパーのファイル・アクセス・コントロール・モデル』

表構造ファイル・ラッパーに関する最適化のヒントと考慮事項

- システムは、ソートされていないデータ・ファイルよりもソートされたデータ・ファイルの方が、効率よく検索できます。
- 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することによってパフォーマンスを高めることができます。
- 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。SYSSTAT および SYSCAT ビューの手操作による更新の詳細については「DB2 管理ガイド」を参照してください。

関連資料:

- 115 ページの 『BLAST ラッパーの最適化のヒント』

表構造ファイル・ラッパーのメッセージ

このセクションでは、表構造ファイルのラッパーを処理するときに出されるメッセージをリストして説明します。メッセージの詳細については、「DB2 メッセージ・リファレンス」を参照してください。

表 5. 表構造ファイルのラッパーによって出されるメッセージ

エラー・コード	メッセージ	説明
SQL0405N	数値リテラル "<literal>" は、値が有効な値の範囲外になっているため、無効です。	データ・ファイルの列、または SQL ステートメントの述部値には、そのデータ・タイプに可能な範囲外の値が入っています。データ・ファイルを修正するか、適切なタイプに列を再定義してください。
SQL0408N	値には、その割り当てターゲットのデータ・タイプとの互換性がありません。ターゲット名は "<column_name>" です。	データ・ファイルの列に、データ・タイプに無効な文字が入っています。データ・ファイルを修正するか、適切なタイプに列を再定義してください。

表 5. 表構造ファイルのラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できます。(理由 "Data source path is NULL".)	IBM ソフトウェア・サポート担当者に連絡してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できます。(理由 "Key Column retrieval failure".)	IBM ソフトウェア・サポート担当者に連絡してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できます。(理由 "STAT failed on data source. ERRNO = <error_number>".)	正しいディレクトリー許可を得ていることを確認してください。ファイルが存在していることを確認してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できます。(理由 "No column info found".)	IBM ソフトウェア・サポート担当者に連絡してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できます。(理由 "Unsupported operator".)	IBM ソフトウェア・サポート担当者に連絡してください。

表 5. 表構造ファイルのラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL1816N	ラッパー " <code><wrapper_name></code> " は、連合データベースに定義を試みているデータ・ソース (" <code><type></code> " " <code>''</code> ") の " <code>type</code> " にアクセスするために使用できません。	サーバー・タイプが無効です。CREATE SERVER ステートメントにサーバー・タイプを指定してはなりません。TYPE キーワードおよび値を除去して再実行してください。
SQL1822N	データ・ソース " <code><server_name></code> " から予期しないエラー・コード " <code>ERRNO = <error_number></code> " を受け取りました。関連するテキストおよびトークンは " <code>Unable to read file</code> " です。	エラー番号の値を調べてください。ファイルを DB2 インスタンス所有者が読み取ることができるかどうか確認してください。次に SQL コマンドを再実行してください。
SQL1822N	データ・ソース " <code><server_name></code> " から予期しないエラー・コード " <code>Data Error</code> " を受け取りました。関連するテキストおよびトークンは " <code>Data source is a non-standard file</code> " です。	データ・ソース・ファイルはディレクトリー、ソケット、または FIFO です。標準ファイルだけがデータ・ソースとしてアクセスできます。FILE_PATH オプションを変更して有効なファイルを示すようにし、SQL コマンドを再実行してください。
SQL1822N	データ・ソース " <code><server_name></code> " から予期しないエラー・コード " <code>ERRNO = <error_number></code> " を受け取りました。関連するテキストおよびトークンは " <code>File open error</code> " です。	ラッパーはファイルをオープンできませんでした。エラー番号を調べて、エラーが発生した原因を判別してください。データ・ソースによる問題を解決して、SQL コマンドを再実行してください。
SQL1822N	データ・ソース " <code><server_name></code> " から予期しないエラー・コード " <code>Data Error</code> " を受け取りました。関連するテキストおよびトークンは " <code>Key column missing</code> " です。	データ・ソースから取り出されたレコードはキー・フィールドが欠落しています。キー列を NULL にすることはできません。データを修正するか、ファイルをソートされていないニックネームとして登録してください。

表 5. 表構造ファイルのラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL1822N	データ・ソース " <code><server_name></code> " から予期しないエラー・コード "Data Error" を受け取りました。関連するテキストおよびトークンは "File not sorted" です。	ファイルはキー列でソートされていませんでした。次のいずれかを実行してください。KEY_COLUMN オプションを変更して正しい列を指すようにする、データ・ファイルを再ソートする、またはニックネームをソートされていないニックネームとして登録する。
SQL1822N	データ・ソース " <code><server_name></code> " から予期しないエラー・コード "Data Error" を受け取りました。関連するテキストおよびトークンは "Key exceeds definition size" です。	データ・ソースから読み取られたキー列フィールドが、ラッパー検索ルーチンが正しく機能しなくなる DB2 列定義よりも大きくなっています。データまたはニックネーム定義を修正して、ニックネームを再登録してください。
SQL1822N	データ・ソース " <code><server_name></code> " から予期しないエラー・コード "Data Error" を受け取りました。関連するテキストおよびトークンは "Line in data file exceeds 32k" です。	データ・ファイルの行が、ラッパーによって許可されている最大長を超えています。行の長さを 32768 より大きくすることはできません。データ・ファイルの行の長さを短くしてください。
SQL1823N	サーバー " <code><server_name></code> " からデータ・タイプ " <code><data_type></code> " に存在するデータ・タイプ・マッピングがありません。	ニックネームがサポートされていないデータ・タイプに定義されています。サポートされているデータ・タイプだけを使ってニックネームを再定義してください。
SQL1881N	" <code><option_name></code> " は " <code><object_name></code> " に対して有効な " <code><component></code> " ではありません。	リストされた値は、リストされたオブジェクトに対して有効なオプションではありません。無効なオプションを削除するか変更して、SQL ステートメントを再実行してください。
SQL1882N	"Nickname" オプション "COLUMN_DELIMITER" は、" <code><nickname_name></code> " に対して " <code><delimiter></code> " に設定できません。	列区切り文字が 2 文字以上の長さになっています。1 文字にオプションを再定義してください。次に SQL ステートメントを再実行してください。

表 5. 表構造ファイルのラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL1882N	"Nickname" オプション "KEY_COLUMN" は、"<nickname_name>" に対して "<column_name>" に設定できません。	キー列として選択された列が、このニックネームに定義されていません。このニックネームのソートされた列の 1 つになるよう KEY_COLUMN オプションを変更して、SQL コマンドを再実行してください。
SQL1882N	"Nickname" オプション "VALIDATE_DATA_FILE" は、"<nickname_name>" に対して "<option_value>" に設定できません。	オプションの値が無効です。有効な値は "Y" または "N" です。オプションを変更して、ニックネームを再登録してください。
SQL1883N	"<option_name>" は "<object_name>" に対して必須 "<component>" オプションです。	ラッパーの必須オプションが SQL ステートメントから欠落しています。必須オプションを追加して、SQL ステートメントを再サブミットしてください。
SQL30090N	操作がアプリケーション実行環境で無効です。理由コード = "21"。	パススルー・セッションを試行しました。表構造ファイル・ラッパーはパススルー・セッションをサポートしません。

関連資料:

- 64 ページの 『Documentum ラッパーのメッセージ』
- 83 ページの 『Excel ラッパーのメッセージ』
- 115 ページの 『BLAST ラッパーのメッセージ』
- 134 ページの 『XML ラッパーのメッセージ』

第 4 章 データ・ソースとしての Documentum

この章では、Documentum とは何か、また Documentum データ・ソースをご使用の連合システムに加える方法などについて解説し、さらに Documentum ラッパーに関連したエラー・メッセージをリストします。

Documentum とは ?

Documentum とは、文書の内容や属性 (チェックイン、チェックアウト、ワークフロー、およびバージョン管理など) の管理を行う文書管理ソフトウェアです。Documentum 製品は 3 層であり、リレーショナル・データベースの最上部に構築されたクライアント / サーバー・システムです。

Docbase は、文書の内容、属性、関連、バージョン、レンダリング、フォーマット、ワークフロー、およびセキュリティーを保管する Documentum リポジトリーです。拡張 SQL ダイアレクトである Documentum 照会言語 (DQL) は、Documentum データを照会するために使用されます。Docbase は、Oracle インスタンスまたは DB2[®] データベースに文書内容ファイルを加えたものと同じです。メタデータは、基礎となるリレーショナル・データベース管理システム (RDBMS) に保管され、内容はデータベースにバイナリー・ラージ・オブジェクト (BLOB) として保管されるか、サーバー・システムのファイル・システム内に保管されるファイルとして保管されます。Documentum についての詳細は、Documentum の資料を参照してください。

Documentum のラッパーを使用すると、Documentum データ・ソースを DB2 連合システムに追加できます。Documentum データ・ソースを連合システムに追加することによって、SQL ステートメントを使用して、Documentum Docbase 内のオブジェクトおよび登録済み表をアクセスしたり、照会することができます。その後このデータをネイティブのデータ・ソースの外に移動せずに、連合システム内の他のデータ・ソースと統合できます。Documentum ラッパーは、Documentum サーバーとのインターフェースとして、クライアント・ライブラリーを使用します。Documentum ラッパーは、2 つのタイプの Documentum サーバーへのアクセスを提供します。それらは EDMS 98 (バージョン 3 と呼ばれる) と 4i です。32 ページの図 4 は、Documentum ラッパーがどのように機能するかを示しています。

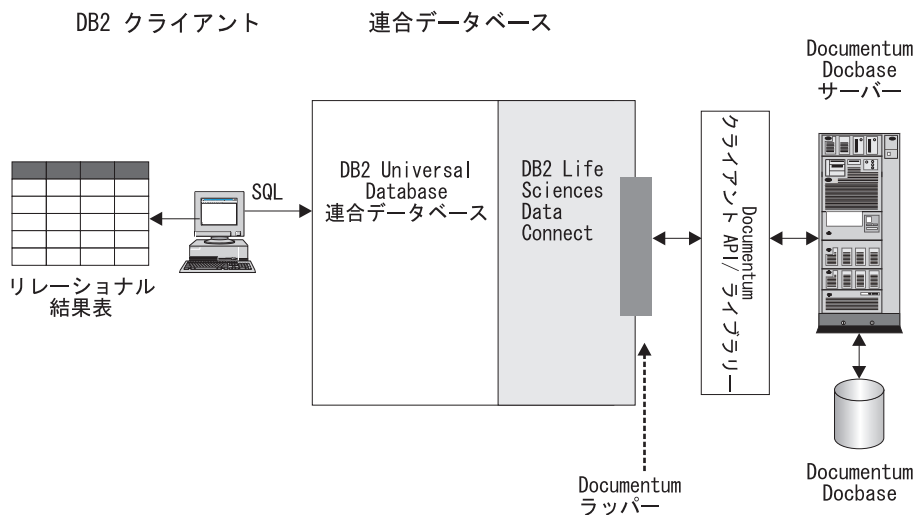


図 4. Documentum ラッパーがどのように機能するか

Documentum ラッパーが登録された後に、Documentum Docbase オブジェクトと登録済み表をリレーショナル表としてマップできます。これは、Docbase 属性を DB2 リレーショナル表の列名にマップすることによって行われます。

たとえば、表 6 は、Documentum Docbase のデフォルトの文書タイプである dm_document を関連するデータとともにリストしています。ユーザーは、この属性サブセットが重要であると判断し、これらの属性を連合データベース・システム内に接続したいと考えています。このデータのサブセットの名前は DrugAB_data としました。

表 6. DrugAB_data

Title (タイトル)	Subject (件名)	Authors (作成者)	Keywords (キーワード)
うさぎに対する Drug A の効果	Drug A	Curran, L.	うさぎ、Drug A
Drug A の毒性結果	Drug A	Abelite, P., McMurtrey, K.	毒性、Drug A
Drug B の相互作用	Drug B	DeNiro, R., Stone, S.	相互作用、drug B
Drug B の化学構造	Drug B	Boyslim, F.	構造、drug B

Documentum ラッパーを登録し終わったら、SQL ステートメントを使用してデータを照会できます。

以下の示す照会では、subject (件名) が Drug A である title (タイトル) と authors (作成者) を表示します。結果の表を 33 ページの表 7 に示します。

```
SELECT title, authors
FROM drugAB_data
WHERE subject = 'Drug A'
```

表 7. 照会の結果

Title (タイトル)	Authors (作成者)
うさぎに対する Drug A の効果	Curran, L.
Drug A の毒性結果	Abelite, P., McMurtrey, K.

関連概念:

- 13 ページの 『表構造ファイルとは ?』
- 73 ページの 『Excel とは ?』
- 91 ページの 『BLAST とは ?』
- 119 ページの 『XML とは ?』

関連作業:

- 33 ページの 『連合システムへの Documentum の追加』

連合システムへの Documentum の追加

手順:

Documentum データ・ソースを連合サーバーに追加するには、以下のようにします。

1. Documentum クライアント・ライブラリーにリンクします。
2. Documentum のクライアント dmcl.ini ファイルを指します。
3. CREATE WRAPPER ステートメントを使用してラッパーを登録します。
4. オプション：照会のパフォーマンスを向上させるために DB2_DJ_COMM 環境変数を設定します。
5. CREATE SERVER ステートメントを使用してサーバーを登録します。
6. CREATE USER MAPPING ステートメントを使用してユーザーにデータ・ソースに対するアクセス権を付与します。
7. CREATE NICKNAME ステートメントを使用してニックネームを登録します。
8. CREATE FUNCTION ステートメントを使用してカスタム関数を作成します。

ステートメントは、DB2 コマンド行プロセッサから実行できます。登録が完了すると、データ・ソースに対して照会ができます。

関連作業:

- 34 ページの 『Documentum クライアント・ライブラリーへのリンク (AIX および Solaris オペレーティング環境のみ)』

- 35 ページの 『Documentum のクライアント dmcl.ini ファイルを指す』
- 36 ページの 『Documentum ラッパーの登録』
- 37 ページの 『Documentum ラッパーの DB2_DJ_COMM 環境変数の設定』
- 38 ページの 『Documentum データ・ソースのサーバーの登録』
- 40 ページの 『ユーザーのマッピング (Documentum ラッパー)』
- 41 ページの 『Documentum データ・ソースのニックネームの登録』
- 51 ページの 『Documentum データ・ソースのカスタム関数の登録』
- 16 ページの 『連合システムへの表構造ファイルの追加』
- 75 ページの 『連合システムへの Excel の追加』
- 96 ページの 『連合システムへの BLAST の追加』
- 123 ページの 『連合システムへの XML の追加』

Documentum クライアント・ライブラリーへのリンク (AIX および Solaris オペレーティング環境のみ)

このタスクは、連合システムへの *Documentum* の追加 のためのメインタスクの一部です。Documentum データ・ソースにアクセスできるようにするには、DB2 連合システムでクライアント・ライブラリーへのリンク・エディットを行わなければなりません。リンク・エディット・プロセスにより、連合サーバーが通信する各データ・ソースごとに、ラッパー・ライブラリーが作成されます。djxlinkDctm スクリプトを実行する際、Documentum ラッパー・ライブラリーを作成します。

手順:

djxlinkDctm スクリプトを実行するには、以下のようにします。

1. LSDC_DMCL 環境変数を Documentum クライアント・ライブラリーがあるディレクトリーを指すように設定します。

たとえば、次のようにします。

```
export LSDC_DMCL=/usr/documentum/product/3.1.7
```

2. 次のコマンドをルートとして入力します。

```
ksh djxlinkDctm
```

注: djxlinkDctm コマンドを DB2 Universal Database FixPak を適用した後で再実行する必要があります。

ここでのタスクの順序では、次のタスクは『*Documentum* のクライアント dmcl.ini ファイルの位置指定』です。

関連作業:

Documentum のクライアント dmcl.ini ファイルを指す

このタスクは、連合システムへの *Documentum* の追加 のためのメインタスクの一部です。Documentum Docbases へのアクセスは、Documentum クライアントの dmcl.ini ファイルによって制御されます。DB2 インスタンスは Documentum Docbase にアクセスできるようにするために、環境変数を Documentum クライアントの dmcl.ini ファイルに設定する必要があります。

手順:

環境変数を設定するには、以下のように行います。

1. db2dj.ini ファイルを編集して、以下の環境変数の 1 つを設定します。

```
DOCUMENTUM=<path>  
DMCL_CONFIG=<path>/dmcl.ini
```

ここで <path> は、使用したい dmcl.ini ファイルを含んでいる完全修飾ディレクトリーです。

Documentum の dmcl.ini ファイルの場所のデフォルト・パスは、/pkgs/documentum です。両方の行が含まれている場合、DMCL_CONFIG が使用されます。

AIX および Solaris オペレーティング環境では、db2dj.ini ファイルは \$HOME/sql1lib/cfg にあります。

Windows では、db2dj.ini ファイルは x:%sql1lib%cfg にあります (ここで **x:** は sql1lib ディレクトリーのあるドライブを表します)。

注: 36 ページの図 5 に示されているように、DB2 インスタンスにアクセス可能なすべての Docbases の報告先となる docbroker の名前が dmcl.ini ファイルに指定されていることを確認してください。

```
##### DOCUMENTUM CLIENT CONFIGURATION FILE #####
#
# Copyright Documentum 1994.
# Version 3.1 of the Documentum Server.
#
# A generated client init file for the Documentum Server.
#
# The only REQUIRED information in this file is the
# [DOCBROKER_PRIMARY] section and an entry for host.
# The host value should be the name of host on which
# your network wide DocBroker is running

[DOCBROKER_PRIMARY]
host = server16.comp2.big.com
```

図 5. docbroker 名が指定されている dmcl.ini ファイルのサンプル

2. DB2 インスタンスをリサイクルするには、以下のコマンドを出します。

```
db2stop
db2start
```

ここでのタスクの順序では、次のタスクは『Documentum ラッパーの登録』です。

関連作業:

- 17 ページの 『表構造ファイル・ラッパーの DB2_DJ_COMM 環境変数の設定』
- 34 ページの 『Documentum クライアント・ライブラリーへのリンク (AIX および Solaris オペレーティング環境のみ)』
- 37 ページの 『Documentum ラッパーの DB2_DJ_COMM 環境変数の設定』
- 102 ページの 『BLAST ラッパーの DB2_DJ_COMM 環境変数の設定』
- 124 ページの 『XML ラッパーの DB2_DJ_COMM 環境変数の設定』

Documentum ラッパーの登録

このタスクは、*連合システムへの Documentum の追加* のためのメインタスクの一部です。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、連合サーバーがデータ・ソースとの通信やデータ・ソースからのデータの検索に使用するメカニズムです。ラッパーは、ライブラリー・ファイルとして、システムにインストールされています。

手順:

Documentum ラッパーを登録するには、CREATE WRAPPER ステートメントをサブミットします。

たとえば、デフォルト・ライブラリー・ファイル `libdb21sdctm.a` から `Dctm_Wrapper` という AIX 上の Documentum ラッパーを作成するには、次のステートメントをサブミットします。

```
CREATE WRAPPER Dctm_Wrapper LIBRARY 'libdb21sdctm.a'  
OPTIONS(DB2_FENCED 'N');
```

サポートされているプラットフォーム別の Documentum ラッパーのデフォルト・ライブラリー・ファイル名の表については、下記の関連タスク・セクションの『DB2 Life Science Data Connect をインストールした後』を参照してください。CREATE WRAPPER ステートメントに関して詳しくは、「DB2 SQL リファレンス」を参照してください。

ここでのタスクの順序では、次のタスクは『Documentum ラッパーの DB2_DJ_COMM 環境変数の設定』です。

関連作業:

- 10 ページの 『DB2 Life Sciences Data Connect をインストールした後』
- 16 ページの 『表構造ファイル・ラッパーの登録』
- 37 ページの 『Documentum ラッパーの DB2_DJ_COMM 環境変数の設定』
- 75 ページの 『Excel ラッパーの登録』
- 101 ページの 『BLAST ラッパーの登録』
- 123 ページの 『XML ラッパーの登録』

Documentum ラッパーの DB2_DJ_COMM 環境変数の設定

このタスクは、連合システムへの Documentum の追加 のためのメインタスクの一部です。Documentum データ・ソースにアクセスする際のパフォーマンスを向上させるために、DB2_DJ_COMM 環境変数を設定します。この変数は、初期設定時に連合サーバーがラッパーをロードするかどうかを決定します。

手順:

DB2_DJ_COMM 環境変数を設定するには、関連する CREATE WRAPPER ステートメントに指定したラッパーに対応するラッパー・ライブラリーを指定して `db2set` コマンドをサブミットします。

たとえば、次のようにします。

```
db2set DB2_DJ_COMM='libdb21sdctm.a'
```

等号 (=) の両側にスペースを入れないようにしてください。

データベース始動時にラッパー・ライブラリーをロードすることに関連したオーバーヘッドがあります。このオーバーヘッドを回避するには、アクセスしたいライブラリーのみを指定します。

DB2_DJ_COMM 環境変数に関して詳しくは、「DB2 管理ガイド」を参照してください。

ここでのタスクの順序では、次のタスクは『Documentum データ・ソースのサーバーの登録』です。

関連作業:

- 17 ページの 『表構造ファイル・ラッパーの DB2_DJ_COMM 環境変数の設定』
- 36 ページの 『Documentum ラッパーの登録』
- 38 ページの 『Documentum データ・ソースのサーバーの登録』
- 102 ページの 『BLAST ラッパーの DB2_DJ_COMM 環境変数の設定』
- 124 ページの 『XML ラッパーの DB2_DJ_COMM 環境変数の設定』

Documentum データ・ソースのサーバーの登録

このタスクは、連合システムへの *Documentum* の追加 のためのメインタスクの一部です。ラッパーが登録された後、対応するサーバーを登録する必要があります。

手順:

Documentum サーバーを連合システムに登録するには、CREATE SERVER ステートメントを使用します。

たとえば、関連する CREATE WRAPPER ステートメントで作成された Dctm_Wrapper ラッパーに対して Dctm_Server1 というサーバーがあるとします。また、サーバーに AIX 上で実行する Docbase が含まれており、データの保管に Oracle を使用するとします。サーバーを登録するには、以下のステートメントをサブミットします。

```
CREATE SERVER Dctm_Server1
TYPE DCTM
VERSION 3
WRAPPER Dctm_Wrapper
OPTIONS( NODE 'Dctm_Docbase',
OS_TYPE 'AIX',
RDBMS_TYPE 'ORACLE');
```

引き数

TYPE データ・ソースのタイプを指定します。Documentum の場合、タイプは DCTM になります。この引き数は必須です。

VERSION

データ・ソースのバージョンを指定します。EDMS98 の場合、値は '3' になります。4i の場合、値は '4' になります。この引き数は必須です。

WRAPPER

このサーバーと関連したラッパーの名前を指定します。この引き数は必須です。

オプション

CONTENT_DIR

GET_FILE、GET_FILE_DEL、GET_RENDITION、および GET_RENDITION_DEL 疑似列によって検索された内容ファイルを保管するためのローカルにアクセス可能なルート・ディレクトリーの名前を指定します。これは、これらの疑似列を使用できるすべてのユーザーが書込可能でなければなりません。そのデフォルト値は /tmp です。このオプションは任意です。

NODE Documentum Docbase の実際の名前を指定します。このオプションは必須です。

OS_TYPE

Docbase サーバーのオペレーティング・システムを指定します。有効値は AIX、SOLARIS、および WINDOWS です。このオプションは必須です。

RDBMS_TYPE

Docbase で使用される RDBMS を指定します。有効値は DB2、INFORMIX、ORACLE、SQLSERVER または SYBASE です。このオプションは必須です。

TRANSACTIONS

サーバー・トランザクション・モードを指定します。有効な値は以下のとおりです。

- NONE — トランザクションは使用可能になりません。
- QUERY — トランザクションは Dctm_Query メソッドでのみ使用可能です。
- ALL — トランザクションは Dctm_Query メソッドで使用可能です。このリリースでは、ALL と QUERY の機能は同じです。

デフォルトは QUERY です。このオプションは任意です。

CREATE SERVER ステートメントに関して詳しくは、「DB2 SQL リファレンス」を参照してください。

ここでのタスクの順序では、次のタスクは『ユーザーのマッピング (Documentum ラッパー)』です。

関連作業:

- 18 ページの 『表構造ファイルのサーバーの登録』
- 37 ページの 『Documentum ラッパーの DB2_DJ_COMM 環境変数の設定』
- 40 ページの 『ユーザーのマッピング (Documentum ラッパー)』
- 76 ページの 『Excel データ・ソースのサーバーの登録』
- 103 ページの 『BLAST データ・ソースのサーバーの登録』
- 125 ページの 『XML データ・ソースのサーバーの登録』

ユーザーのマッピング (Documentum ラッパー)

このタスクは、*連合システムへの Documentum の追加* のためのメインタスクの一部です。以前に定義したサーバーにユーザーをマップして、データ・ソースへのアクセス権をそれらユーザーに付与する必要があります。

手順:

連合サーバーへユーザーをマップするには、CREATE USER MAPPING ステートメントを使用します。

たとえば、次に示す CREATE USER MAPPING ステートメントは、Dctm_Server1 サーバーでユーザー Chuck をユーザー Charles にマップします。

```
CREATE USER MAPPING FOR Chuck SERVER Dctm_Server1
OPTIONS(REMOTE_AUTHID 'Charles', REMOTE_PASSWORD 'Charles_pw');
```

独自のユーザー・マッピングを定義することもできます。以下の例では、USER は、USER という名前のユーザーではなく、現行のユーザーを意味するキーワードです。

```
CREATE USER MAPPING FOR USER SERVER Dctm_Server1
OPTIONS(REMOTE_AUTHID 'Lisa', REMOTE_PASSWORD 'Lisa_pw')
```

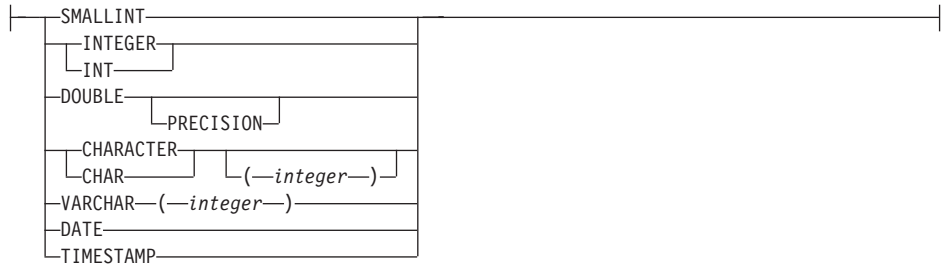
CREATE USER MAPPING ステートメントに関して詳しくは、「*DB2 SQL リファレンス*」を参照してください。

ここでのタスクの順序では、次のタスクは『*Documentum データ・ソースのニックネームの登録*』です。

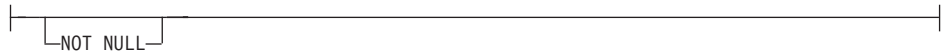
関連作業:

- 38 ページの 『Documentum データ・ソースのサーバーの登録』
- 41 ページの 『Documentum データ・ソースのニックネームの登録』

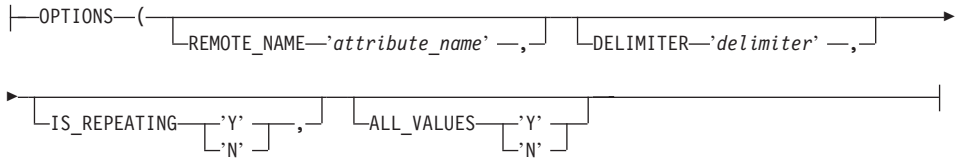
data-type:



column-option:



nickname-column-options:



CREATE NICKNAME に関して詳しくは、「DB2 SQL リファレンス」を参照してください。

列オプション

NOT NULL

TIMESTAMP および DATE と定義されたものを除くすべての単一値の列は、NOT NULL として定義しなければなりません。反復する属性は、ニックネームで NOT NULL として定義してはなりません。

ニックネーム列オプション

ニックネーム列オプションの値は、単一引用符で囲む必要があります。

ALL_VALUES

繰り返し属性のすべての値を、指定された区切り文字で分離して戻すことを指定します。このオプションが指定されていないか、'N' の場合、繰り返し属性の最後の値のみが戻されます。DELIMITER のところの説明のように、ALL_VALUES は IS_REPEATING オプションが 'Y' である VARCHAR 列に対してのみ指定できます (IS_REG_TABLE = 'Y' の場合は無効です)。

DELIMITER

反復する属性の複数值を連結するときに使用する区切り文字を指定します。区

切り文字は、1 つ以上の文字になります。デフォルトの区切り文字はコンマです。このオプションは、IS_REPEATING オプションが 'Y' に設定されているデータ・タイプ VARCHAR のオブジェクトの属性に対してのみ有効です。このオプションは任意です。

IS_REPEATING

列が複数値であることを示します。有効な値は 'Y' および 'N' です。デフォルトは 'N' です。このオプションは任意です。

REMOTE_NAME

対応する Documentum 属性または列の名前を指定します。このオプションは、リモート属性または列名をローカルの DB2 列名にマップします。これはデフォルトの DB2 列名になります。このオプションは任意です。

ニックネーム・オプション

ニックネーム・オプション値は、単一引用符で囲む必要があります。

ALL_VERSIONS

全オブジェクト・バージョンを検索するかどうか指定します。有効な値は 'y'、'Y'、'n'、および 'N' です。デフォルト値の 'N' は、現行オブジェクト・バージョンのみが照会プロセスに組み込まれることを意味します。このオプションは、IS_REG_TABLE = 'Y' の場合は無効になります。このオプションは任意です。

FOLDERS

1 つ以上の論理的に結合され、構文的に正確な Documentum FOLDER 述部を含むストリングを指定します。FOLDER 述部を指定すると、このニックネームによって表現される文書のセットを、指定したフォルダー内のものに制限します。

このオプションを指定するときは、FOLDERS オプションの値全体を単一引用符で囲み、ストリング内では単一引用符の代わりに二重引用符を使用します。

たとえば、次のものを挿入するとします。

```
FOLDER('/Tools',DESCEND) OR FOLDER('/Cars')
```

次の FOLDERS オプションを指定します。

```
FOLDERS 'FOLDER("/Tools",DESCEND) OR FOLDER("/Cars")'
```

このオプションは、IS_REG_TABLE = 'Y' の場合は無効になります。このオプションは任意です。

IS_REG_TABLE

REMOTE_OBJECT オプションによって指定されたオブジェクトが Documentum の登録済みの表かどうかを指定します。有効な値は 'y'、'Y'、'n'、および 'N' です。デフォルト値は 'N' です。このオプションは任意です。

注: ALTER NICKNAME ステートメントによってこのオプションを変更することにより、ニックネームを Documentum オブジェクトから登録済みの表に (またはその逆に) 変更することはできません。そうする代わりに、ニックネームを DROP してもう一度 CREATE する必要があります。

REMOTE_OBJECT

ニックネームに関連した Documentum オブジェクト・タイプの名前を指定します。名前は任意の Documentum オブジェクト・タイプかまたは登録済みの表になります。登録済みの表になる場合、表の所有者の名前を接頭部付ける必要があります。登録済みの表が Docbase 所有者に所属する場合、dm_dbo を所有者の名前に使用できます。このオプションは必須です。

注: 新規オプションの構造がオリジナルのオプションの構造に類似していなければ、ALTER NICKNAME を使用して REMOTE_OBJECT オプションの値を変更するとエラーになります。

疑似列とは

CREATE NICKNAME ステートメントは、また 6 つの疑似列を定義します。これらの列は、オブジェクトの内容およびその他の情報にアクセスするために使用されます。

疑似列およびその定義は 表 8 にリストされています。

表 8. 疑似列名および定義

疑似列名	定義
GET_FILE	VARCHAR (255)
GET_FILE_DEL	VARCHAR (255)
GET_RENDITION	VARCHAR (255)
GET_RENDITION_DEL	VARCHAR (255)
HITS	INTEGER
SCORE	DOUBLE

表 9 は、SELECT 文節の疑似列をリストしたものです。

表 9. SELECT 文節の疑似列

疑似列名	説明
GET_FILE	<p>列の値のほかに、現在行の内容ファイルを検索します。</p> <p>内容ファイルの拡張子は、その Documentum の形式名です。同じ名前のファイルが存在する場合、それは上書きされます。</p> <p>GET_FILE は、オブジェクトの基本形式の取得を試みます。行のその値はオブジェクトの a_content_type です。オブジェクトに内容ファイルがない場合、その値はストリング "no_content" になります。</p> <p>たとえば、次のようにします。</p> <pre>SELECT object_name, DCTM.GET_FILE FROM ...</pre> <p>内容ファイルは、サーバーの CONTENT_DIR オプションで指定したサーバー・ディレクトリーに置かれます。また、ユーザーの DB2 ローカル名を使用して名前が付けられたサブディレクトリーにも置かれます。サブディレクトリーが存在しない場合は作成されます。</p> <p>その拡張子は、文書の形式タイプに対して Docbase で定義された DOS 拡張子になります。たとえば、MS Word 文書の場合は ".doc" です。</p> <p>ストリング "no_content" かまたはファイルの完全修飾名を戻します。</p>
GET_FILE_DEL	<p>GET_FILE_DEL がその照会において以前の行で検索したファイル (存在する場合) を最初に削除する点を除いて、この関数は GET_FILE と同種です。ストリング "no_content" かまたはファイルの完全修飾名を戻します。</p>

表 9. SELECT 文節の疑似列 (続き)

疑似列名	説明
GET_RENDITION	<p data-bbox="592 215 1210 276">列の値のほかに、現在行で、その解釈 (オリジナル文書の異なる形式のコピー) の内容ファイルを検索します。</p> <p data-bbox="592 309 1210 401">内容ファイルの拡張子は、その Documentum の形式名です。同じ名前のファイルが存在する場合、それは上書きされます。</p> <p data-bbox="592 434 1157 526">解釈フォーマットを指定するためには、フォーム <code>DCTM.RENDITION_FORMAT(<format>) = 1</code> の述部を WHERE 文節に指定する必要があります。</p> <p data-bbox="592 558 911 585">たとえば、次のようにします。</p> <pre data-bbox="592 609 1022 687">SELECT object_name, get_rendition FROM ... WHERE DCTM.RENDITION_FORMAT('pdf')=1</pre> <p data-bbox="592 720 1210 881">GET_RENDITION は、オブジェクトの名前つき解釈の取得を試みます。 行の値が、オブジェクトに内容ファイルがない場合の文字列 "no_content"、または解釈が存在しない場合の文字列 "not_found" になる場合を除いて、これはオブジェクトの a_content_type になります。</p> <p data-bbox="592 914 1210 1076">内容ファイルは、サーバーの CONTENT_DIR オプションで指定したサーバー・ディレクトリーに置かれます。また、ユーザーの DB2 ローカル名を使用して名前が付けられたサブディレクトリーにも置かれます。サブディレクトリーが存在しない場合は作成されます。</p> <p data-bbox="592 1109 1210 1201">その拡張子は、文書の形式タイプに対して Docbase で定義された DOS 拡張子になります。たとえば、MS Word 文書の場合は ".doc" です。</p> <p data-bbox="592 1234 1210 1303">文字列 "no_content"、"not found"、またはファイルの完全修飾名を戻します。</p>
GET_RENDITION_DEL	<p data-bbox="592 1321 1210 1482">GET_RENDITION_DEL がその照会において前の行で検索したファイル (存在する場合) を最初に削除する点を除いて、この関数は GET_RENDITION と同じです。文字列 "no_content"、"not found"、またはファイルの完全修飾名を戻します。</p>

表 10 は、検索文節を含む照会においての `SELECT` 文節の疑似列をリストしています。

表 10. 検索文節を含む、照会においての `SELECT` 文節の疑似列

疑似列名	説明
HITS	<p>検索基準が一致した文書内の場所の数を表す整数を戻します。</p> <p>たとえば、次のようにします。</p> <pre>SELECT r_object_id, object_name, hits FROM std_doc WHERE DCTM.SEARCH_WORDS (''workflow'' OR ''flowchart'')=1</pre> <p>戻された各文書ごとに、文書の内容の中で "workflow" および "flowchart" というワードが出現する回数が合計され、HITS 値として戻されます。</p> <p>HITS 疑似列は、文書内に内容ファイルが 1 つしかない場合に適しています。これは、一般的な事例です。この疑似列は、<code>SELECT</code> ステートメントの <code>WHERE</code> 文節の修飾で使用できます。ただし、これは <code>SELECT</code> 文節でも指定する必要があります。</p>

表 10. 検索文節を含む、照会における *SELECT* 文節の疑似列 (続き)

疑似列名	説明
SCORE	<p>文書の検索ヒット・ランキングを戻します。</p> <p>この疑似列は、Documentum の ACCRUE 概念演算子とともに使用します。両方とも、戻された各文書内で指定したワードが検出された数を示す数値を戻します。</p> <p>たとえば、次のようにします。</p> <pre>SELECT object_name, score FROM std_doc ここで、 DCTM.SEARCH_TOPIC('<ACCRUE>("document","management","workflow")')=1 AND SCORE >=75</pre> <p>ステートメントは、内容の中に 2 つか 3 つの指定ワードを持つすべての文書を戻します。文書に 1 つのワードしかない場合、スコア 50 が割り当てられるため、WHERE 文節基準は失敗し、これは戻されません。3 つのうち 2 つのワードが検出される場合、文書にはスコア 75 が割り当てられます。3 つすべてのワードが検出される場合、文書のスコアは 88 になります。</p> <p>SCORE 疑似列は 1 つの内容ファイルを持つ文書に使用されます。これは、一般的な事例です。</p> <p>SCORE は、WHERE に SEARCH_WORDS() または SEARCH_TOPIC() 関数が含まれる場合のみ、SELECT 文節内に存在します。WHERE 文節では、これは ACCRUE 概念演算子とともに使用されます。</p> <p>ACCRUE 概念演算子に関して詳しくは、Documentum の資料を参照してください。</p>

CREATE NICKNAME の例

以下の CREATE NICKNAME ステートメントは、ニックネーム std_doc を定義します。Std_doc は、オブジェクト・タイプ dm_document を持つ Documentum Docbase と関連付けられています。表 11 は、Documentum 属性およびデータ・タイプを DB2 リレーショナル列名およびデータ・タイプにマップしています。これらは、CREATE NICKNAME ステートメントを構成するために使用されます。

表 11. std_doc ニックネームの DB2 列への Documentum 属性のマッピング。

Documentum 属性名	Documentum データ・タイプ	DB2 列名	DB2 データ・タイプ	反復の有無	NULL の可否
object_name	string(255)	object_name	varchar	なし	なし

表 11. *std_doc* ニックネームの DB2 列への *Documentum* 属性のマッピング。(続き)

Documentum 属性名	Documentum データ・タイプ	DB2 列名	DB2 データ・タイプ	反復の 有無	NULL の 可否
r_object_id	ID	object_id	char(16)	なし	なし
r_object_type	string(32)	object_type	varchar	なし	なし
title	string(255)	title	varchar	なし	なし
subject	string(128)	subject	varchar	なし	なし
authors	string(32)	author	varchar	あり	あり
キーワード	string(32)	keyword	varchar	あり	あり
r_creation_date	time	creation_date	timestamp	なし	あり
r_modify_date	time	modified_date	timestamp	なし	あり
a_status	string(16)	status	varchar	なし	なし
a_content_type	string(32)	content_type	varchar	なし	なし
r_content_size	double	content_size	integer	なし	なし
owner_name	string(32)	owner_name	varchar	なし	あり

表 12 は、ニックネームに使用される各 *Documentum* 属性を説明しています。

表 12. *std_doc* ニックネームの *Documentum* 属性の説明

Documentum 属性名	説明
object_name	オブジェクトのユーザー定義名。
r_object_id	このオブジェクトの固有オブジェクト ID。作成時に設定される。
r_object_type	オブジェクトのタイプ。オブジェクトの作成時に設定される。
title	オブジェクトのユーザー定義タイトル。
subject	オブジェクトのユーザー定義サブジェクト。
authors	オブジェクトに関する作成者のユーザー定義リスト
キーワード	オブジェクトに関するユーザー定義キーワードのリスト
r_creation_date	オブジェクトが作成された日時。
r_modify_date	オブジェクトの最終変更日時。
a_status	ルーター・タスクが転送されるときにサーバーによって設定される。値は、ルーター・オブジェクト内の <i>attached_task_status</i> に対して割り当てられた値からとられる。
a_content_type	オブジェクトの内容のファイル形式。
r_content_size	内容のバイト数。複数ページの文書の場合、この属性は、文書と関連した最初の内容のサイズを記録する。

表 12. *std_doc* ニックネームの *Documentum* 属性の説明 (続き)

Documentum 属性名	説明
owner_name	オブジェクトの所有者の名前 (オブジェクトのを作成したユーザー)。

48 ページの表 11 は、以下の CREATE NICKNAME ステートメントに変換されます。

```
CREATE NICKNAME std_doc (
  object_name varchar(255) not null,
  object_id char(16) not null OPTIONS(REMOTE_NAME 'r_object_id'),
  object_type varchar(32) not null OPTIONS(REMOTE_NAME 'r_object_type'),
  title varchar(255) not null,
  subject varchar(128) not null,
  author varchar(32) OPTIONS(REMOTE_NAME 'authors', IS_REPEATING 'Y'),
  keyword varchar(32) OPTIONS(REMOTE_NAME 'keywords', IS_REPEATING 'Y'),
  creation_date timestamp OPTIONS(REMOTE_NAME 'r_creation_date'),
  modified_date timestamp OPTIONS(REMOTE_NAME 'r_modify_date'),
  status varchar(16) not null OPTIONS(REMOTE_NAME 'a_status'),
  content_type varchar(32) not null OPTIONS(REMOTE_NAME 'a_content_type'),
  content_size integer not null OPTIONS(REMOTE_NAME 'r_content_size'),
  owner_name varchar(32))
FOR SERVER Dctm_Server2 OPTIONS (REMOTE_OBJECT 'dm_document', IS_REG_TABLE 'N')
```

CREATE NICKNAME ステートメントをサブミットした後、ニックネーム *std_doc* を使用して連合システムを照会することができます。連合システム内で、*std_doc* ニックネームを他のニックネームと結合することもできます。

注: カタログでは、このニックネームの列数は、疑似列を含んでいるため CREATE NICKNAME ステートメントに指定されているものよりも 6 つだけ多くなります。

CreateNicknameFile ユーティリティを使用すると、Documentum タイプを自動的に DB2 タイプにマップしたり、初期 CREATE NICKNAME ステートメントを作成することができます。CreateNicknameFile ユーティリティの詳細については、下記の関連リンク・セクションを参照してください。

ここでのタスクの順序では、次のタスクは『Documentum データ・ソースのカスタム関数の登録』です。

関連作業:

- 19 ページの 『表構造ファイルのニックネームの登録』
- 40 ページの 『ユーザーのマッピング (Documentum ラッパー)』
- 51 ページの 『Documentum データ・ソースのカスタム関数の登録』
- 77 ページの 『Excel データ・ソースのニックネームの登録』
- 104 ページの 『BLAST データ・ソースのニックネームの登録』

- 126 ページの『XML データ・ソースのニックネームの登録』
- 143 ページの『第 8 章 ニックネームのコスト計算オプションの指定』

Documentum データ・ソースのカスタム関数の登録

このタスクは、*Documentum* への追加のためのメインタスクの一部です。いくつかの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する必要があります。これらの関数を使用して、照会内での完全テキストの検索や文書内容の検索などのいくつかの *Documentum* 固有の機能にアクセスすることができます。

述部のカスタム関数は 52 ページの表 13 にリストされています。

DB2 は `BOOLEAN` データ・タイプをサポートしません。そのため、有効な `SQL` ステートメントを作成するには、各カスタム関数の値を明示的にテストする必要があります。ラッパーのインプリメンテーションは、指定したテスト比較演算子に関係なく、`"DCTM.<function><args> = 1"` のセマンティクスのみをサポートします。

注: `TOPIC` 関数への参照は、Verity, Inc. によってサード・パーティーの完全テキスト索引付けシステムの一部として提供される *Documentum* 関数への参照です。

手順:

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。

カスタム関数はすべて、`DCTM` というスキーマ名を使用して登録しなければなりません。各関数の完全修飾名は、`DCTM.<function_name>` です。

以下の例は、`ANY_EQ` カスタム関数を登録します。

```
CREATE FUNCTION DCTM.ANY_EQ (CHAR(), CHAR()) RETURNS INTEGER
AS TEMPLATE DETERMINISTIC NO EXTERNAL ACTION
```

各カスタム関数は、*Documentum* ラッパーをインストールしているそれぞれの DB2 データベースごとに 1 度登録しなければなりません。

カスタム関数の登録するための助けとして、サンプル・ファイル `create_function_mappings.ddl` が `sqllib/samples/lifesci` ディレクトリに提供されています。このファイルには、それぞれのカスタム関数ごとの定義が含まれています。この `ddl` ファイルを実行して、*Documentum* ラッパーをインストールしているそれぞれの DB2 データベースごとにカスタム関数を登録することができます。

カスタム関数のストリング引き数の規則

ストリングとして渡される引き数はすべて、以下の規則に従います。

- 各ストリングは単一引用符で囲みます。

- スtring内の単一引用符は、2 つの単一引用符によって表現されます。

照会でのカスタム関数の使用

以下の例は、照会でのカスタム関数の使用を示しています。

std_doc nickname から、1 つ以上の 'Dave Winters' という名前の作成者を持つ文書のオブジェクト名および作成者を表示するには、次のようにします。

```
SELECT object_name,authors FROM std_doc
WHERE DCTM.ANY_EQ(authors,'Dave Winters')=1
```

std_doc nickname から、1 つ以上の 'Dave Winters' または 'Jon Doe' という名前の作成者を持つ文書のオブジェクト名および作成者を表示するには、次のようにします。

```
SELECT object_name,authors FROM std_doc
WHERE DCTM.ANY_IN(authors,'Dave Winters','Jon Doe')=1
```

オブジェクト名および r_object_id を表示し、作成者の列に 'Dave Win%' というStringを含む文書の std_doc ニックネームから内容ファイルを検索するには、次のようにします。

```
SELECT object_name, r_object_id, get_file FROM std_doc
WHERE DCTM.ANY_LIKE(authors,'Dave Win%')=1
```

カスタム関数表

表 13 は、述部のカスタム関数をリストしたものです。

表 13. 述部のカスタム関数

関数名	説明
ANY_EQ(arg1, arg2)	<p>指定した値と等しい任意の値の反復属性をテストします。 2 つの必須引き数をとります。</p> <p>arg1 反復属性を表す列名を指定します。</p> <p>arg2 比較される値を指定します。</p> <p>たとえば、次のようにします。</p> <pre>... WHERE DCTM.ANY_EQ(authors,'Dave Winters')=1</pre>
ANY_NE(arg1, arg2)	<p>指定した値と等しくない任意の値の反復属性をテストします。 2 つの必須引き数をとります。</p> <p>arg1 反復属性を表す列名を指定します。</p> <p>arg2 比較される値を指定します。</p> <p>たとえば、次のようにします。</p> <pre>... WHERE DCTM.ANY_NE(authors,'Dave Winters')=1</pre>

表 13. 述部のカスタム関数 (続き)

関数名	説明
ANY_LT(arg1, arg2)	<p>指定した値より小さい任意の値の反復属性をテストします。2 つの必須引き数をとります。</p> <p>arg1 反復属性を表す列名を指定します。</p> <p>arg2 比較される値を指定します。</p> <p>たとえば、次のようにします。</p> <pre>... WHERE DCTM.ANY_LT(num_approvers,4)=1</pre>
ANY_GT(arg1, arg2)	<p>指定した値より大きい任意の値の反復属性をテストします。2 つの必須引き数をとります。</p> <p>arg1 反復属性を表す列名を指定します。</p> <p>arg2 比較される値を指定します。</p> <p>たとえば、次のようにします。</p> <pre>... WHERE DCTM.ANY_GT(num_approvers,3)=1</pre>
ANY_LE(arg1, arg2)	<p>指定した値以下の任意の値の反復属性をテストします。2 つの必須引き数をとります。</p> <p>arg1 反復属性を表す列名を指定します。</p> <p>arg2 比較される値を指定します。</p> <p>たとえば、次のようにします。</p> <pre>... WHERE DCTM.ANY_LE(num_approvers,2)=1</pre>
ANY_GE(arg1, arg2)	<p>指定した値以上の任意の値の反復属性をテストします。2 つの必須引き数をとります。</p> <p>arg1 反復属性を表す列名を指定します。</p> <p>arg2 比較される値を指定します。</p> <p>たとえば、次のようにします。</p> <pre>... WHERE DCTM.ANY_GE(num_approvers,1)=1</pre>

表 13. 述部のカスタム関数 (続き)

関数名	説明
ANY_IN(arg1, arg2 - arg11)	<p>指定した値リストの中の任意の 10 の値の反復属性をテストします。同じデータ・タイプの 3 ~ 11 の引き数をとります。</p> <p>arg1 反復属性を表す列名を指定します。</p> <p>arg2-arg11 比較される値のコンマ区切りリストを指定します。</p> <p>たとえば、次のようにします。</p> <pre>... WHERE DCTM.ANY_IN(authors,'Crick','Watson')=1</pre>
ANY_LIKE(arg1, arg2)	<p>指定した値と同種の任意の値の反復リストをテストします。2 つの必須引き数をとります。</p> <p>arg1 反復属性を表す列名を指定します。</p> <p>arg2 単一引用符で囲まれたサブストリングと比較されるパターンを指定します。</p> <p>たとえば、次のようにします。</p> <pre>... WHERE DCTM.ANY_LIKE(authors,'Dave Win%')=1 OR DCTM.ANY_LIKE(keywords,'%¥_%')=1</pre> <p>注: ANY_LIKE() 述部ではエスケープ文節はサポートされません。</p>
ANY_NOT_LIKE(arg1, arg2)	<p>指定した値と異種の任意の値の反復リストをテストします。2 つの必須引き数をとります。</p> <p>arg1 反復属性を表す列名を指定します。</p> <p>arg2 単一引用符で囲まれたサブストリングと比較されるパターンを指定します。</p> <p>たとえば、次のようにします。</p> <pre>... WHERE DCTM.ANY_NOT_LIKE(authors,'Dave Win%')=1 OR DCTM.ANY_NOT_LIKE(keywords,'%¥_%')=1</pre> <p>注: ANY_NOT_LIKE() 述部ではエスケープ文節はサポートされません。</p>

表 13. 述部のカスタム関数 (続き)

関数名	説明
ANY_NULL(arg)	<p>IS NULL の反復属性をテストします。反復属性または単一値 DATE または TIMESTAMP 属性の名前となる 1 つの必須引き数をとります。</p> <p>たとえば、次のようにします。</p> <pre>... WHERE DCTM.ANY_NULL(authors)=1</pre>
ANY_NOT_NULL(arg)	<p>IS NOT NULL の反復属性をテストします。反復属性の名前となる 1 つの必須引き数をとります。</p> <p>たとえば、次のようにします。</p> <pre>... WHERE DCTM.ANY_NOT_NULL(authors)=1</pre>
ANY_SAME_INDEX(arg1 - arg10)	<p>各属性の同じ索引で値の反復属性をテストします。2 ~ 10 の別の ANY_xx() 関数をとります</p> <p>以下の例は、UCD に関連した少なくとも 1 人の Ken という作成者を文書が持つかどうかを調べます。</p> <pre>... WHERE DCTM.ANY_SAME_INDEX(ANY_EQ(author_name,'Ken'), DCTM.ANY_NE(author_affiliation,'UCD'))=1</pre>
CABINET(arg) および CABINET_TREE(arg)	<p>Docbase キャビネットの完全修飾名となる 1 つの必須引き数をとります。</p> <p>たとえば、次のようにします。</p> <pre>... WHERE DCTM.CABINET('/Tools')=1 ... WHERE DCTM.CABINET_TREE('/MyDocs')=1</pre> <p>CABINET および CABINET_TREE の複数のインスタンスを使用して、複数のキャビネットを指定します。</p> <p>たとえば、次のようにします。</p> <pre>... WHERE DCTM.CABINET('/Tools')=1 OR DCTM.CABINET_TREE('/Parts')=1</pre>

表 13. 述部のカスタム関数 (続き)

関数名	説明
FOLDER(arg) および FOLDER_TREE(arg)	<p>Docbase フォルダーまたはキャビネットの完全修飾名となる 1 つの必須引き数をとります。</p> <p>たとえば、次のようにします。</p> <pre>... DCTM.FOLDER('/Tools/Drills')=1 ... DCTM.FOLDER_TREE('/MyDocs/WhitePapers')=1</pre> <p>FOLDER および FOLDER_TREE の複数のインスタンスを使用して、複数のフォルダーを指定します。</p> <p>たとえば、次のようにします。</p> <pre>... DCTM.FOLDER('/Tools/Drills')=1 OR DCTM.FOLDER_TREE('/Animals/Horses')=1</pre>
RENDITION_FORMAT (フォーマット)	<p>GET_RENDITION および GET_RENDITION_DEL 疑似列を処理して、検索する解釈のフォーマットを設定します。フォーマットを指定する単一文字ストリング引き数を取ります。</p> <p>次の例は、PDF フォーマットの文書を検索する例です。</p> <pre>SELECT get_rendition FROM WHERE DCTM.RENDITION_FORMAT('pdf')=1</pre>
USER(1)	<p>値を現行ユーザーの Documentum 作成者 ID と比較します。ダミー引き数 (1) をとります。</p> <p>たとえば、次のようにします。</p> <pre>... WHERE approver = DCTM.USER(1)</pre> <p>注: DB2 作成者 ID に対応する Documentum 作成者 ID を作成するには、CREATE USER MAPPING ステートメントを使用します。ユーザー・マッピングの詳細については、下記の関連リンク・セクションを参照してください。</p>

表 13. 述部のカスタム関数 (続き)

関数名	説明
SEARCH_WORDS(arg)	<p>個々のワードのリストとなる 1 つの必須ストリング引き数をとります。それらのワードは単一引用符で囲まれ、AND、OR、または NOT で区切られ、そして優先制御に対して括弧が使用されます。ワードに空白を含めることはできません。また、ワードは単一引用符で囲まなければなりません。</p> <p>たとえば、次のようにします。</p> <pre>... DCTM.SEARCH_WORDS(''yeast'' AND (''bread'' OR ''cake'') AND NOT ''wedding'')=1</pre>
SEARCH_TOPIC(arg)	<p>Verity TOPIC 照会ステートメントとなる 1 つの必須ストリング引き数をとります。これは、Documentum および Verity に逐語的に渡されます。</p> <p>たとえば、次のようにします。</p> <pre>... WHERE DCTM.SEARCH_TOPIC("quick")=1</pre>

CREATE FUNCTION ステートメントに関して詳しくは、「DB2 SQL リファレンス」を参照してください。

ここでの作業は、これで終わりです。

関連作業:

- 41 ページの 『Documentum データ・ソースのニックネームの登録』

Documentum データ・ソースに対する照会の実行

ラッパーを登録した後、Documentum データ・ソースに対して SQL 照会を実行できます。このセクションでは、いくつかの照会例が紹介されています。

手順:

照会を実行するには、通常の表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。

以下の照会は、'Test Document' という名前の文書の Docbase 文書すべてを表示します。

```
SELECT object_name
FROM std_doc
WHERE object_name='Test Document';
```

以下の照会は、カスタム関数 ANY_EQ を使用して、作成者の 1 人として 'Joe Doe' を持つすべての文書を表示します。

```
SELECT object_name
FROM std_doc
WHERE DCTM.ANY_EQ(author,'Joe Doe')=1
```

以下の照会は、FOLDER_TREE 関数および SEARCH_WORDS 関数を使用して、Approved キャビネット内のテキスト "protein" を含むすべての文書を検索します。

```
SELECT object_name
FROM std_doc
WHERE DCTM.FOLDER_TREE('/Approved')=1
      AND DCTM.SEARCH_WORDS('protein')=1
```

以下の照会は、GET_FILE 疑似列、FOLDER_TREE カスタム関数、および ANY_IN カスタム関数を使用して、DB2 サーバー上のファイル名を検索します。この DB2 サーバーは、リストにあるすべての作成者を持つ、承認済みのキャビネット内のすべての文書の内容が配置されているサーバーです。

```
SELECT object_name, object_id, get_file
FROM std_doc
WHERE DCTM.FOLDER_TREE('/Approved')=1
      AND DCTM.ANY_IN(author, 'Mary Black', 'Joe Carson', 'Peter Miller')=1
```

関連作業:

- 79 ページの 『Excel データ・ソースに対する照会の実行』
- 132 ページの 『XML データ・ソースに対する照会の実行』

Documentum ラッパー用の CreateNicknameFile ユーティリティーとは？

無料でダウンロードできる Docbasic ユーティリティー (CreateNicknameFile) を使用して、Docbase オブジェクトまたは登録されている表の完全な定義の入った ASCII ファイルを作成できます。この出力ファイルを編集して、以下のことを行えます。

- 列および属性ごとにカスタム・ローカル名を定義します。ローカルおよびリモート名は、Docbase で認識される最初の名前です。
- 必要のない列および属性を削除します。事前定義されている Documentum 文書タイプ (dm_document) には、EDMS98 で 59 属性、4i で 76 属性しか含めることができません。これらのほとんどは、低レベルの文書管理およびアプリケーション開発のためのメタデータです。それら必要のない属性を削除することによって、パフォーマンスに影響を与えることなく、SELECT * SQL ステートメントを一層効果的なものにすることができます。

- FOLDERS オプションの値を追加して、特定の Documentum フォルダーへの、このニックネームに対する検索を制限します。
- 必要であれば、DATE から TIMESTAMP へのマッピングを変更します。ユーティリティーは、最も効果的である考えられるため、DQL DATE から DB2® DATE へのマッピングを生成します。
- アプリケーションに応じて、CHAR から VARCHAR への、またその逆のマッピングを変更します。

Docbase にユーティリティーをインストールして、Documentum Windows® グラフィカル・ユーザー・インターフェースから実行する必要があります。ユーティリティーが生成するファイルは、そのユーティリティーがインストールされている Docbase に固有のものであります。

関連作業:

- 59 ページの 『CreateNicknameFile ユーティリティーのインストール (Documentum ラッパー)』
- 60 ページの 『CreateNicknameFile ユーティリティーの構成 (Documentum ラッパー)』
- 61 ページの 『DM_ID オブジェクト・タイプの Documentum 登録済み表へのマッピング』

CreateNicknameFile ユーティリティーのインストール (Documentum ラッパー)

CreateNicknameFile ユーティリティーは、Documentum データ・ソースの CREATE NICKNAME ステートメントを作成する場合の助けとなります。

手順:

ユーティリティーをインストールするには、以下の手順に従います。

1. DB2 Life Science Data Connect 製品の Web サイト (<http://www.ibm.com/software/data/db2/lifesciencesdataconnect/>) のダウンロード・セクションから、CreateNicknameFile ユーティリティーをダウンロードします。
2. EDMS98 Workspace グラフィカル・ユーザー・インターフェースまたは 4i Desktop Client を使用して、CreateNicknameFile.txt という名前のユーティリティーをインポートします。ユーティリティーをプロシージャ・タイプとして、Docbase キャビネットまたはフォルダーにインポートし、自由に名前を指定できます。
3. 新しくインポートした CreateNicknameFile.txt オブジェクトのプロパティ・ダイアログで、「ユーザーによって実行可能 (Can be run by user)」ボックスにチェックマークを付けます。

関連概念:

- 58 ページの『Documentum ラッパー用の CreateNicknameFile ユーティリティとは?』

関連作業:

- 60 ページの『CreateNicknameFile ユーティリティの構成 (Documentum ラッパー)』
- 61 ページの『DM_ID オブジェクト・タイプの Documentum 登録済み表へのマッピング』

CreateNicknameFile ユーティリティの構成 (Documentum ラッパー)

CreateNicknameFile ユーティリティは、Documentum データ・ソースの CREATE NICKNAME ステートメントを作成する場合の助けとなります。

前提条件:

これを構成するには、まず CreateNicknameFile ユーティリティをインストールする必要があります。 CreateNicknameFile ユーティリティのインストールについての詳細は、下記の関連タスク・セクションの『CreateNicknameFile ユーティリティのインストール (Documentum ラッパー)』を参照してください。

手順:

ユーティリティをインストールした後、構成するには以下の手順に従います。

1. ユーティリティのアイコンをダブルクリックして、実行します。
2. Documentum の文書 / オブジェクト・タイプ名を入力します。デフォルトは dm_document です。

注: 登録されている表にニックネームを作成する必要がある場合、名前として dm_registered を指定してください。 dm_registered を指定すると、<owner>.<table_name> の形式で完全修飾表名を入力するように要求されます。表が Docbase の所有者によって所有されている (一般) 場合、所有者名に use dm_dbo を使用します。

ユーティリティは、登録されている表のニックネームの名前の命名規則を想定しています。その規則では、「登録されている表 (registered table)」であることを示すために、表名の接頭部として "rt_" を付けます。この規則を使用したくない場合、ユーティリティが設定するニックネームを変更できます。

3. 作成するニックネームに関連したサーバー名を入力します。
4. ニックネームの名前を入力します。

ニックネームの名前は、注釈的で、DB2 インスタンス内で固有のものである必要があります。複数のサーバーに対して同じ <object_type> が定義される必要があるた

め、ユーティリティーは <server_name>.<object_type> の命名規則を想定します。この規則を使用したくない場合、ユーティリティーが設定するニックネームを変更できます。

5. 出力ファイルの名前を入力します。

デフォルトは C:\Temp\nickname.txt です。出力ファイルを受け取るディレクトリはすでに存在していて、ユーティリティーを実行しているユーザーによって書き込み可能である必要があります。

表示されるプロンプトに答えると、ニックネーム・ファイルが作成され、テキスト・エディターで表示されます。

関連概念:

- 58 ページの 『Documentum ラッパー用の CreateNicknameFile ユーティリティーとは?』

関連作業:

- 41 ページの 『Documentum データ・ソースのニックネームの登録』
- 59 ページの 『CreateNicknameFile ユーティリティーのインストール (Documentum ラッパー)』

DM_ID オブジェクト・タイプの Documentum 登録済み表へのマッピング

CreateNicknameFile ユーティリティーによって作成される列定義は、Documentum ラッパーの要件 (各データ・タイプを対応する DB2 データ・タイプの正しいマッピングも含む) に準拠しています。唯一の例外は、Documentum が登録済み表の DM_ID データ・タイプをサポートしない点です。ユーティリティーは、登録済み表の列が 16 文字長のストリングおよび「_id」で終了する名前のストリングとして定義される場合に、その列がオブジェクト ID を含むために使用されると想定します。DM_ID データ・タイプの場合、ユーティリティーは列を DB2 CHAR(16) データ・タイプにマップします。それ以外のすべての場合、すべての string/varchar 列は DB2 VARCHAR データ・タイプにマップされます。

手順:

正しいデータ・タイプのマッピングを確実なものにするためには、以下のように行います。

1. CreateNicknameFile ユーティリティーによって作成された出力ファイルの中の列データ・タイプ定義を調べます。
2. ユーティリティーが Documentum 列のデータ・タイプを間違った DB2 データ・タイプにマップした場合は、そのファイルを使用してニックネームを DB2 に登録する前に DB2 データ・タイプを変更します。

関連概念:

- 58 ページの 『Documentum ラッパー用の CreateNicknameFile ユーティリティとは?』

関連作業:

- 59 ページの 『CreateNicknameFile ユーティリティのインストール (Documentum ラッパー)』
- 60 ページの 『CreateNicknameFile ユーティリティの構成 (Documentum ラッパー)』

繰り返し属性の二重の定義 (Documentum ラッパー)

ラッパーの照会機能を最大限に活用するには、各属性を DB2 データ・タイプと同じように定義する必要があります。つまり、Documentum 整数を DB2 整数などとして定義する必要があります。しかし、このように定義すると、非 VARCHAR 繰り返し属性について複数の値が戻されなくなります。そのような列については、index[0] の値だけが戻されます。

この制限は、ラッパーが可能な場合いつでも、Docbase オブジェクトにつき結果として 1 つの行しか戻さないために生じます。この制限は、繰り返し属性が選択されている場合だけの問題です。しかし、データ・タイプが VARCHAR ではない、同じリモートの繰り返し属性のための 2 次列を定義できます。

この列名は、すべての値の区切り文字で区切られたリストとして、すべての値を戻すために SELECT リストで使用されます。(各列の DELIMITER オプションは使用される区切り文字を指定します)

複数值の列のローカル名は標準化する必要があります。複数值の列のローカル名は、真のデータ・タイプとして定義される列のローカル名に、接頭部 "m_" を追加することによって標準化できます。

たとえば、データ・タイプ TIMESTAMP で定義されている、approval_dates という名前の Documentum 繰り返し属性のニックネーム列があるとします。この場合、m_approval_dates という名前の 2 次ニックネーム列を作成して、VARCHAR データ・タイプとして定義できます。さらに、SELECT リストで m_approval_dates を使用して、区切り文字で区切られたリストですべての承認日付を戻すことができます。

真のデータ・タイプが VARCHAR である繰り返し属性に関して、二重の定義を使用する必要はありません。

Documentum ラッパーの制限と考慮事項

このセクションには、Documentum ラッパーの使用に関連した制限と考慮事項のリストが含まれています。

- 繰り返し属性の値を戻すことに関する制限：次の項目に関して最後の値だけが戻されます。

- 非 VARCHAR 繰り返し属性
- ALL_VALUES 'N' が指定されているときの VARCHAR 列

この制限に対処するため、繰り返し属性列に二重の定義を作成できます。繰り返し属性に二重の定義を作成する方法の詳細については、下記の関連リンク・セクションを参照してください。

さらに、VARCHAR として定義されている繰り返し属性の複数の値は、区切り文字で区切られた 1 つのストリングとして戻されます。区切り文字は、CREATE NICKNAME ステートメントの DELIMITER ニックネーム・オプション・セットの設定によって異なります。

- パススルー機能はサポートされていません。
- DB2 アプリケーションによって確立される、DB2 データベースへの接続ごとに、Documentum ラッパーは最大で同時に 10 の Documentum セッションをサポートでき、各セッションでは最大で同時に 10 の Documentum 照会を管理できます。1 つの DB2 アプリケーションで、同時に進行する複数の照会を実行できます。照会の存続期間は、DB2 にサブミットされる時に始まり、結果セット上の対応するカーソルが閉じるときに終了します。指定された時間において、その時間に進行中の照会すべてに対して、以下の制限が課されます。

- すべての照会によって参照されるすべてのニックネームは、10 以下の異なった Documentum サーバーに置かれる必要があります。
- 1 つの Documentum サーバーから参照されるニックネームは 10 以下です。

複数の照会で参照される、または 1 つの照会で複数回参照されるニックネームは、参照回数ごとに 1 つとカウントされます。

- Documentum ラッパーは、クライアント・ライブラリーの AIX にバージョン 3.1.7a を使用します。Documentum 4i を使用する場合、Documentum からクライアント・バージョンの古いバージョンを入手する必要があります（まだインストールされていない場合）。
- DB2 はブール・タイプをサポートしていないため、WHERE 文節で使用されるほとんどのカスタム関数 (USER を除く) は "=1" についてチェックする必要があります。これらの関数は、整数を戻すために定義されているためです。

たとえば、次のようにします。

```
"... WHERE DCTM.ANY_EQ(authors,'Dave Winters')=1"
```

- DB2 の制限のため、カスタム関数 `USER` は、使用されていない整数の引き数によって定義されます。
- 同じ DB2 インスタンスに対して実行されているすべてのサーバーは、同じ `Documentum dmcl.ini` 構成パラメーターを共有する必要があります。
- 繰り返し属性の `ANY_IN custom` カスタム関数の値の最大値は、1 つのステートメントにつき 10 です。しかし、複数のステートメントについては論理和演算されます。
- `ANY_SAME_INDEX` カスタム関数の場合、繰り返し属性の同じ索引で行われる値のテストの最大数は、10 です。テストは、左から右へと検査される `AND` テストです。
- ラッパーには、特定のコード・ページに固有の機能はありません。

関連資料:

- 23 ページの 『表構造ファイル・ラッパーに関するラッパーの制限と考慮事項』
- 24 ページの 『表構造ファイル・ラッパーに関するファイルの制限と考慮事項』
- 81 ページの 『Excel ラッパーのラッパーの制限』
- 82 ページの 『Excel ファイルの制限』
- 134 ページの 『XML ラッパーの制限と考慮事項』

Documentum ラッパーのアクセス・コントロール

照会は、Docbase でユーザーが許可を得ていることが条件となります。ユーザーが最低でも読み取りアクセス権を持っている文書だけが、照会の結果に含まれます。

関連資料:

- 24 ページの 『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』
- 82 ページの 『Excel ラッパーのファイル・アクセス・コントロール・モデル』

Documentum ラッパーのメッセージ

このセクションでは、Documentum のラッパーを処理する際に出されるメッセージをリストして説明します。メッセージの詳細については、「DB2 メッセージ・リファレンス」を参照してください。

表 14. Documentum のラッパーによって出されるメッセージ

エラー・コード	メッセージ	説明
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由 "sqlno_crule_save_plans [100]:rc (-2144272209) Empty plan list detect".)	DB2 にサブミットされた SQL 照会はラッパーによって処理できませんでした。構文を訂正して再サブミットしてください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由 "dmAPI exec failed: [DM_QUERY_E_BAD_QUAL] error: "The attribute qualifier, A0, for attribute <column_name>, is not a valid qualifier".)	REMOTE_OBJECT ニックネーム・オプションに不正な Documentum タイプまたは登録済みの表が入力されました。正しい Documentum タイプまたは登録済みの表を使用するようにニックネームを変更してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由 "Invalid null column specified".)	内部プログラミング・エラー。IBM ソフトウェア・サポート担当者に連絡してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由 "Nickname specification is empty".)	内部プログラミング・エラー。IBM ソフトウェア・サポート担当者に連絡してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由 "The Output object is empty or incomplete".)	内部プログラミング・エラー。IBM ソフトウェア・サポート担当者に連絡してください。

表 14. Documentum のラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由 "Unexpected number of columns requested".)	内部プログラミング・エラー。IBM ソフトウェア・サポート担当者に連絡してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由 "No column information found".)	内部プログラミング・エラー。IBM ソフトウェア・サポート担当者に連絡してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由 "Unsupported column type requested".)	内部プログラミング・エラー。IBM ソフトウェア・サポート担当者に連絡してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由 "Incorrect Column definition".)	内部プログラミング・エラー。IBM ソフトウェア・サポート担当者に連絡してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由 "Inconsistent type; DB2 request != nickname type".)	内部プログラミング・エラー。IBM ソフトウェア・サポート担当者に連絡してください。

表 14. Documentum のラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由 "Output parameter is not NULL".)	内部プログラミング・エラー。IBM ソフトウェア・サポート担当者に連絡してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由 "Query output variable is not NULL".)	内部プログラミング・エラー。IBM ソフトウェア・サポート担当者に連絡してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由 "Invalid timestamp length".)	内部プログラミング・エラー。IBM ソフトウェア・サポート担当者に連絡してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由 "Inconsistent number of columns".)	内部プログラミング・エラー。IBM ソフトウェア・サポート担当者に連絡してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由 "Could not access data when converting values".)	内部プログラミング・エラー。IBM ソフトウェア・サポート担当者に連絡してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由 "Failed to initialize the DMCL client".)	Documentum クライアントが初期設定できません。システム管理者に連絡してください。

表 14. Documentum のラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由 "Get_User returned NULL".)	内部プログラミング・エラー。IBM ソフトウェア・サポート担当者に連絡してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由 "Get_Local_User returned NULL".)	内部プログラミング・エラー。IBM ソフトウェア・サポート担当者に連絡してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由 "Begin Transaction failed".)	Documentum は begintrans が失敗したことを報告しました。システム管理者に連絡してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由 "Input parameter was not NULL".)	内部プログラミング・エラー。IBM ソフトウェア・サポート担当者に連絡してください。
SQL901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由「Dctm 関数は DCTM.function(...) =1 のようであればなりません」。)	ユーザーは Dctm 関数の述部の RHS として =1 を使用しませんでした。構文を変更して、照会をもう一度実行してください。

表 14. Documentum のラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由 "Invalid column number requested".)	内部プログラミング・エラー。IBM ソフトウェア・サポート担当者に連絡してください。
SQL1881N	"DELIMITER" は "<column-name>" の有効な "COLUMN" オプションではありません。	DELIMITER オプションが列 <column-name> に指定されていましたが、IS_REPEATING オプションが指定されていませんでした。
SQL1882N	"SERVER" オプション "RDBMS_TYPE" は、"<server-name>" の "<option-value>" に設定することはできません。	RDBMS_TYPE サーバー・オプションに指定された値が無効です。次の値のいずれかでなければなりません。: DB2、INFORMIX、ORACLE、SQLSERVER または SYBASE。
SQL1882N	"SERVER" オプション "TRANSACTIONS" は、"<server-name>" の "<option-value>" に設定することはできません。	TRANSACTIONS サーバー・オプションに指定された値が無効です。次の値のいずれかでなければなりません。: NONE、QUERY、PASSTHRU または ALL。
SQL1882N	"NICKNAME" オプション "IS_REG_TABLE" は、"<nickname>" の "<option-value>" に設定することはできません。	IS_REG_TABLE ニックネーム・オプションに指定された値が無効です。次の値のいずれかでなければなりません。: 'Y' または 'N'。
SQL1882N	"NICKNAME" オプション "ALL_VERSIONS" は、"<nickname>" の "<option-value>" に設定することはできません。	ALL_VERSIONS ニックネーム・オプションに指定された値が無効です。次の値のいずれかでなければなりません。: 'Y' または 'N'。
SQL1882N	"SERVER" オプション "OS_TYPE" は、"<server-name>" の "<option-value>" に設定することはできません。	OS_TYPE サーバー・オプションに指定された値が無効です。次の値のいずれかでなければなりません。: AIX、HPUX、SOLARIS または WINDOWS。

表 14. Documentum のラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL1882N	"NICKNAME" オプション "FOLDERS" は、"<nickname>" の "<option-value>" に設定することはできません	FOLDERS ニックネーム・オプションに指定された値が無効です。これは、IS_REG_TABLE が 'Y' の表に指定することはできません。
SQL1882N	"NICKNAME" オプション "VERSIONS" は、"<nickname>" の "<option-value>" に設定することはできません	VERSIONS ニックネーム・オプションに指定された値が無効です。次の値のいずれかでなければなりません。: 'Y' または 'N'。さらに、VERSIONS 'Y' は、IS_REG_TABLE が 'Y' の表に指定することはできません。
SQL30090N	操作がアプリケーション実行環境で無効です。理由コード = "Invalid column name, IS_REG_TABLE, or IS_REPEATING specified in nickname"	ニックネーム・ステートメントを調べ、IS_REG_TABLE、IS_REPEATING、REMOTE_NAME オプションおよび列名が正しく指定されているか確認してください。
SQL30090N	操作がアプリケーション実行環境で無効です。理由コード = "db2dj.ini missing DOCUMENTUM or DMCL_CONFIG env var"	必須の環境変数が設定されていません。それらを db2dj.ini ファイルに設定してください。
SQL30090N	操作がアプリケーション実行環境で無効です。理由コード = "Failed to open log file for debugging"	トラブルシューティング用に使用するログ・ファイルにアクセスできません。システム管理者に連絡してください。
SQL30090N	操作がアプリケーション実行環境で無効です。理由コード = "Only one search condition may be specified"	1 つの照会につき 1 つのカスタム検索機能しか指定できません。
SQL30090N	操作がアプリケーション実行環境で無効です。理由コード = "Failed to create content directory"	DB2 エージェントによって宛先ディレクトリが作成可能であることを確認してください。
SQL30090N	操作がアプリケーション実行環境で無効です。理由コード = "Failed to change permissions on content file"	db2 エージェントによって内容ディレクトリが作成可能であることを確認してください。

関連資料:

- 25 ページの 『表構造ファイル・ラッパーのメッセージ』
- 83 ページの 『Excel ラッパーのメッセージ』
- 115 ページの 『BLAST ラッパーのメッセージ』
- 134 ページの 『XML ラッパーのメッセージ』

第 5 章 データ・ソースとしての Excel

この章では、Excel とは何か、また Excel データ・ソースをご使用の連合システムに加える方法などについて解説し、さらに Excel ラッパーに関連したエラー・メッセージをリストします。

Excel とは ?

Excel スプレッドシートまたはワークブックとは、Microsoft® (MS) 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。DB2 Life Sciences Data Connect は、Excel 97、および Excel 2000 のスプレッドシートをサポートします。図 6 は、Excel ラッパーがスプレッドシートと連合システムを接続する方法を示しています。

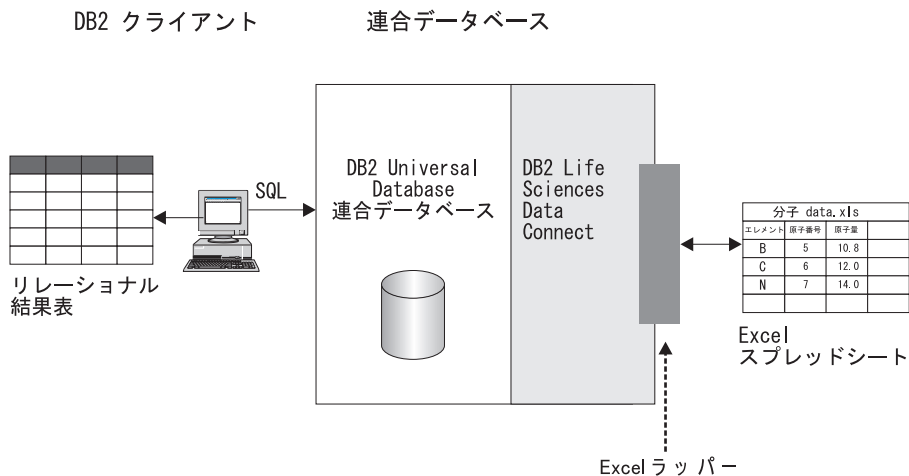


図 6. Excel ラッパーがどのように機能するか

Excel ラッパーは CREATE NICKNAME ステートメントを使用して、Excel スプレッドシート内の列を DB2® 連合システム内の列にマップします。表 15 は、Compound_Master.xls というファイルに保管されるサンプル・スプレッドシート・データを示しています。

表 15. Compound_Master.xls のサンプル・スプレッドシート・データ

	A	B	C	D
1	compound_A	1.23	367	テスト済み

表 15. *Compound_Master.xls* のサンプル・スプレッドシート・データ (続き)

	A	B	C	D
2	compound_G		210	
3	compound_F	0.000425536	174	テスト済み
4	compound_Y	1.00256		テスト済み
5	compound_Q		1024	
6	compound_B	33.5362		
7	compound_S	0.96723	67	テスト済み
8				
9	compound_O	1.2		テスト済み

通常、この情報は標準 SQL コマンドでは使用できません。Excel ラッパーがインストールされて登録されると、この情報が標準リレーショナル・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。たとえば、分子の数が 100 より大きいすべての複合データを知りたい場合、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、表 16 に示されています。

表 16. 照会の結果

COMPOUND_NAME	WEIGHT	MOL_COUNT	WAS_TESTED
compound_A	1.23	367	テスト済み
compound_G		210	
compound_F	0.000425536	174	テスト済み
compound_Q		1024	

関連概念:

- 13 ページの 『表構造ファイルとは ?』
- 31 ページの 『Documentum とは ?』
- 91 ページの 『BLAST とは ?』
- 119 ページの 『XML とは ?』

関連作業:

- 75 ページの 『連合システムへの Excel の追加』

関連資料:

- 75 ページの 『Excel ラッパーの前提条件』

Excel ラッパーの前提条件

Excel データ・ソース・ラッパーの使用に関する前提条件は以下のとおりです。

- MS Excel ラッパーを使用する前に、DB2 Life Science Data Connect がインストールされているサーバーに、Excel アプリケーションをインストールしておく必要があります。

関連作業:

- 75 ページの 『[連合システムへの Excel の追加](#)』
- 75 ページの 『[Excel ラッパーの登録](#)』

連合システムへの Excel の追加

手順:

Excel データ・ソースを連合システムに追加するには、以下のようになります。

1. CREATE WRAPPER ステートメントを使用してラッパーを登録します。
2. CREATE SERVER ステートメントを使用してサーバーを登録します。
3. アクセスしたい Excel スプレッドシートごとに、CREATE NICKNAME ステートメントを使用してニックネームを登録します。

コマンドは、DB2 コマンド行プロセッサから実行できます。

関連作業:

- 75 ページの 『[Excel ラッパーの登録](#)』
- 76 ページの 『[Excel データ・ソースのサーバーの登録](#)』
- 77 ページの 『[Excel データ・ソースのニックネームの登録](#)』
- 16 ページの 『[連合システムへの表構造ファイルの追加](#)』
- 33 ページの 『[連合システムへの Documentum の追加](#)』
- 96 ページの 『[連合システムへの BLAST の追加](#)』
- 123 ページの 『[連合システムへの XML の追加](#)』

Excel ラッパーの登録

このタスクは、[連合システムへの Excel の追加](#) のためのメインタスクの一部です。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、連合サーバーがデータ・ソースとの通信やデータ・ソースからのデータの検索に使用するメカニズムです。ラッパーは、ライブラリー・ファイルとして、システムにインストールされています。

手順:

Excel データ・ソース・ラッパーを登録するには、CREATE WRAPPER ステートメントをサブミットします。

ライブラリー・ファイル db21sx1s.d11 を使用して、Excel_9x_Wrapper という Excel 97 用の Excel ラッパーを作成するには、以下のステートメントをサブミットします。

```
CREATE WRAPPER Excel_9x_Wrapper LIBRARY 'db21sx1s.d11'  
OPTIONS(DB2_FENCED 'N');
```

CREATE WRAPPER ステートメントに関して詳しくは、「DB2 SQL リファレンス」を参照してください。

ここでのタスクの順序では、次のタスクは『Excel データ・ソースのサーバーの登録』です。

関連作業:

- 76 ページの 『Excel データ・ソースのサーバーの登録』

関連資料:

- 75 ページの 『Excel ラッパーの前提条件』

Excel データ・ソースのサーバーの登録

このタスクは、*連合システムへの Excel の追加* のためのメインタスクの一部です。ラッパーが登録された後、対応するサーバーを登録する必要があります。

手順:

Excel サーバーを連合システムに登録するには、CREATE SERVER ステートメントを使用します。

たとえば、CREATE WRAPPER ステートメントを使用して作成した Excel_2000_Wrapper ラッパーのサーバーを登録する biochem_node1 というノード名の biochem_lab というサーバーを作成するには、以下のステートメントをサブミットします。

```
CREATE SERVER biochem_lab WRAPPER Excel_2000_Wrapper;
```

引き数の定義

WRAPPER

関連する CREATE WRAPPER ステートメントに登録したラッパー名を指定します。この引き数は必須です。

CREATE SERVER ステートメントに関して詳しくは、「DB2 SQL リファレンス」を参照してください。

ここでのタスクの順序では、次のタスクは『Excel データ・ソースのニックネームの登録』です。

関連作業:

- 18 ページの 『表構造ファイルのサーバーの登録』
- 38 ページの 『Documentum データ・ソースのサーバーの登録』
- 75 ページの 『Excel ラッパーの登録』
- 77 ページの 『Excel データ・ソースのニックネームの登録』
- 103 ページの 『BLAST データ・ソースのサーバーの登録』
- 125 ページの 『XML データ・ソースのサーバーの登録』

Excel データ・ソースのニックネームの登録

このタスクは、連合システムへの Excel の追加 のためのメインタスクの一部です。サーバーを登録した後、対応するニックネームを登録する必要があります。ニックネームは、照会の中で Excel データ・ソースを参照する場合に使用されます。

手順:

Excel データ・ソースをリレーショナル表にマップするには、CREATE NICKNAME ステートメントを使用してニックネームを作成します。

CREATE NICKNAME 構文 (Excel 用)

```
▶▶ CREATE NICKNAME nickname ( column-name | data-type | column-option | )  
▶▶ FOR SERVER server-name OPTIONS ( (FILE_PATH 'path' ) ) ▶▶
```

data-type:



Excel データ・ソースに対する照会の実行

このセクションでは、ニックネームの例として `Compounds` を使用していくつかのサンプル Excel スプレッドシート照会をリストしています。

手順:

照会を実行するには、通常の表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。

以下の照会は、分子の質量が 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;
```

以下の照会は、複合名または分子の質量が `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;
```

以下の照会は、複合名にストリング `ase` が含まれており、分子の質量が 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;
```

関連作業:

- 57 ページの 『Documentum データ・ソースに対する照会の実行』
- 79 ページの 『サンプル Excel ラッパーのシナリオ』
- 132 ページの 『XML データ・ソースに対する照会の実行』

サンプル Excel ラッパーのシナリオ

このセクションでは、`C:\%Data` ディレクトリーにある Excel 2000 スプレッドシートにアクセスする `Excel_2000` ラッパーのサンプルのインプリメンテーションについて説明します。シナリオはラッパー、サーバー、および 1 つのニックネームを登録しますが、それらはスプレッドシートにアクセスするために使用されます。シナリオに示されるステートメントは、DB2 コマンド行プロセッサを使用して入力されます。ラッパーが登録された後、スプレッドシートに対して照会ができます。

シナリオは、Compound_Master.xls という複合スプレッドシートから始まります。このスプレッドシートは 4 つの列と 9 つの行を持ちます。ファイルの完全修飾パス名は、C:\Data\Compound_Master.xls です。内容は表 17 に示されています。

表 17. サンプル・スプレッドシート Compound_Master.xls

	A	B	C	D
1	compound_A	1.23	367	テスト済み
2	compound_G		210	
3	compound_F	0.000425536	174	テスト済み
4	compound_Y	1.00256		テスト済み
5	compound_Q		1024	
6	compound_B	33.5362		
7	compound_S	0.96723	67	テスト済み
8				
9	compound_O	1.2		テスト済み

手順:

Excel ラッパーを使用して Excel 2000 スプレッドシートにアクセスするには、以下を行います。

1. Excel_2000 ラッパーを登録します。

```
db2 => CREATE WRAPPER Excel_2000 LIBRARY 'db2lsxls.dll'
OPTIONS(DB2_FENCED 'N')
```

2. サーバーを登録します。

```
db2 => CREATE SERVER biochem_lab WRAPPER Excel_2000
```

3. Excel スプレッドシートを参照するニックネームを登録します。

```
db2 => CREATE NICKNAME Compound_Master (compound_name VARCHAR(40),
weight FLOAT, mol_count INTEGER, was_tested VARCHAR(20))
FOR 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 データ・ソースは連合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、サンプル SQL 照会および Excel データ・ソースを使用して得られる結果を示しています。

- サンプル SQL 照会: "Give me all the compound data where mol_count is greater than 100"

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

結果: All fields for rows 1, 2, 3, 5, and 7.

- サンプル SQL 照会: "Give me the compound_name and mol_count for all compounds where the mol_count has not yet been determined.

```
SELECT compound_name, mol_count FROM compound_master  
WHERE mol_count IS NULL
```

結果: Fields compound_name & mol_count of rows 4, 6, and 9 from the spreadsheet.

- サンプル SQL 照会: "Count the number of compounds that have not been tested and the weight is greater than 1."

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

結果: The record count of 1 which represents the single row 6 from the spreadsheet that meets the criteria.

- サンプル SQL 照会: "Give me the compound_name and mol_count for all compounds where the mol_count has been determined and is less than the average mol_count"

```
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)
```

副照会は主照会に平均値 368 を戻し、その後主照会は表 18 を戻します。

表 18. 照会の結果

COMPOUND_NAME	MOL_COUNT
compound_A	367
compound_G	210
compound_F	174
compound_S	67

関連作業:

- 75 ページの 『連合システムへの Excel の追加』
- 79 ページの 『Excel データ・ソースに対する照会の実行』

Excel ラッパーのラッパーの制限

- Excel ラッパーは、DB2 Universal Database Enterprise Server Edition をサポートする Microsoft Windows オペレーティング・システムでのみ使用可能です。
- パススルー・セッションは Excel ラッパーでは使用できません。
- Excel スプレッドシート・データは読み取り専用で、書き込みはできません。

- ラッパーがサポートする DATE データ・タイプの日付範囲は、1970 年 1 月 1 日から 2038 年 1 月 18 日までです。

関連資料:

- 23 ページの 『表構造ファイル・ラッパーに関するラッパーの制限と考慮事項』
- 24 ページの 『表構造ファイル・ラッパーに関するファイルの制限と考慮事項』
- 63 ページの 『Documentum ラッパーの制限と考慮事項』
- 82 ページの 『Excel ファイルの制限』
- 134 ページの 『XML ラッパーの制限と考慮事項』

Excel ファイルの制限

- データ・タイプはそれぞれの列の中で一貫している必要があります、列データ・タイプはニックネームの登録プロセス中に正確に記述しなければなりません。
- Excel ラッパーは、Excel ワークブック内で基本スプレッドシートにのみアクセスできます。
- スプレッドシート内のブランク・セルは、NULL として解釈されます。
- スプレッド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。
- スプレッドシートに、ブランク桁を含めることができます。しかし、これらの桁は、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。

関連資料:

- 81 ページの 『Excel ラッパーのラッパーの制限』

Excel ラッパーのファイル・アクセス・コントロール・モデル

データベース管理システムは、DB2 データベース・サービスの LOG ON AS プロパティの権限によって Excel ファイルにアクセスします。この設定は、DB2 インスタンスの LOG ON プロパティ・ページに表示できます。プロパティ・ページは、Windows NT の「サービス」コントロール・パネルを介してアクセスします。

関連資料:

- 24 ページの 『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』
- 64 ページの 『Documentum ラッパーのアクセス・コントロール』

Excel ラッパーのメッセージ

このセクションでは、Excel のラッパーを処理する際に出されるメッセージをリストして説明します。メッセージの詳細については、「DB2 メッセージ・リファレンス」を参照してください。

表 19. Excel のラッパーによって出されるメッセージ

エラー・コード	メッセージ	説明
SQL1817N	CREATE SERVER ステートメントは、連合データベースに定義したいデータ・ソースの "VERSION" を示していません。	CREATE SERVER ステートメント作成時に VERSION パラメーターが指定されませんでした。SQL ステートメントを変更して再実行してください。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1000.<internal program code>" を受け取りました。関連するテキストおよびトークンは "Memory allocation error" です。	IBM ソフトウェア・サポート担当者に連絡してください。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1001.<internal program code>" を受け取りました。関連するテキストおよびトークンは "Unknown option" です。	DDL ステートメントで指定したオプションはサポートされません。SQL ステートメントを変更して再実行してください。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1002.<internal program code>" を受け取りました。関連するテキストおよびトークンは "Creation of DELTA object failed" です。	内部プログラム・エラーが発生しました。IBM ソフトウェア・サポート担当者に連絡してください。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1100.<internal program code>" を受け取りました。関連するテキストおよびトークンは "Wrapper options are not supported" です。	ラッパー OPTIONS はこのラッパーではサポートされません。SQL ステートメントを変更して再実行してください。

表 19. Excel のラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1200.<internal program code>" を受け取りました。関連するテキストおよびトークンは "<option> is an unsupported Server option" です。	指定したオプションはこのラッパーではサポートされません。SQL ステートメントを変更して再実行してください。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1201.<internal program code>" を受け取りました。関連するテキストおよびトークンは "Error obtaining server name" です。	内部プログラム・エラーが発生しました。IBM ソフトウェア・サポート担当者に連絡してください。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1209.<internal program code>" を受け取りました。関連するテキストおよびトークンは "Error converting VARCHAR data" です。	内部プログラム・エラーが発生しました。IBM ソフトウェア・サポート担当者に連絡してください。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1211.<internal program code>" を受け取りました。関連するテキストおよびトークンは "Error converting INTEGER data" です。	内部プログラム・エラーが発生しました。IBM ソフトウェア・サポート担当者に連絡してください。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1212.<internal program code>" を受け取りました。関連するテキストおよびトークンは "Error converting FLOAT data" です。	内部プログラム・エラーが発生しました。IBM ソフトウェア・サポート担当者に連絡してください。

表 19. Excel のラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1400.<internal program code>" を受け取りました。関連するテキストおよびトークンは "<option> is an unsupported User option" です。	指定したオプションはこのラッパーではサポートされません。SQL ステートメントを変更して再実行してください。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1401.<internal program code>" を受け取りました。関連するテキストおよびトークンは "Creation of USER Delta object failed" です。	内部プログラム・エラーが発生しました。IBM ソフトウェア・サポート担当者に連絡してください。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1500.<internal program code>" を受け取りました。関連するテキストおよびトークンは "<option> is an unsupported Nickname option" です。	指定したオプションはこのラッパーではサポートされません。SQL ステートメントを変更して再実行してください。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1501.<internal program code>" を受け取りました。関連するテキストおよびトークンは "Required option PATH not specified" です。	NICKNAME を登録するためには PATH オプションが必要です。SQL ステートメントを変更して再実行してください。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1502.<internal program code>" を受け取りました。関連するテキストおよびトークンは "Creation of NICKNAME Delta object failed" です。	内部プログラム・エラーが発生しました。IBM ソフトウェア・サポート担当者に連絡してください。

表 19. Excel のラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1503.<internal program code>" を受け取りました。関連するテキストおよびトークンは "Error obtaining Nickname column type" です。	内部プログラム・エラーが発生しました。IBM ソフトウェア・サポート担当者に連絡してください。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1504.<internal program code>" を受け取りました。関連するテキストおよびトークンは "Error obtaining Nickname column type name" です。	内部プログラム・エラーが発生しました。IBM ソフトウェア・サポート担当者に連絡してください。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1505.<internal program code>" を受け取りました。関連するテキストおよびトークンは "received from data source Excel Wrapper" です。	指定した <data type> はこのラッパーではサポートされません。SQL ステートメントを変更して再実行してください。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1506.<internal program code>" を受け取りました。関連するテキストおよびトークンは "Error obtaining Nickname column info" です。	内部プログラム・エラーが発生しました。IBM ソフトウェア・サポート担当者に連絡してください。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1507.<internal program code>" を受け取りました。関連するテキストおよびトークンは "<option> option cannot be dropped" です。	指定したオプションは必須オプションのため、除去できません。

表 19. Excel のラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1508.VANI" を受け取りました。関連するテキストおよびトークンは "Column names cannot be altered" です。	列名の変更は Excel ラッパーによって許可されていません。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1701.<internal program code>" を受け取りました。関連するテキストおよびトークンは "Error parsing SQL" です。	内部プログラム・エラーが発生しました。IBM ソフトウェア・サポート担当者に連絡してください。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1702.<internal program code>" を受け取りました。関連するテキストおよびトークンは "Error accessing NICKNAME object" です。	内部プログラム・エラーが発生しました。IBM ソフトウェア・サポート担当者に連絡してください。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1703.<internal program code>" を受け取りました。関連するテキストおよびトークンは "Error building data storage area" です。	内部プログラム・エラーが発生しました。IBM ソフトウェア・サポート担当者に連絡してください。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1704.<internal program code>" を受け取りました。関連するテキストおよびトークンは "Error linking SQL to Nickname Data" です。	内部プログラム・エラーが発生しました。IBM ソフトウェア・サポート担当者に連絡してください。

表 19. Excel のラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1705.<internal program code>" を受け取りました。関連するテキストおよびトークンは "Excel application startup failed" です。	Excel アプリケーションの始動が失敗しました。Excel がシステムにインストールされていて、正しいバージョンのラッパーが登録されていることを確認してください。Windows NT の「サービス」コントロール・パネルで、DB2 インスタンスの LOG ON AS プロパティを調べてください。この権限を使用して Excel アプリケーションにアクセスします。このユーザーに適切な権限があることを確認するか、またはこのプロパティを許可アカウントに変更した後、DB2 を再始動して SQL 照会を再実行してください。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1706.<internal program code>" を受け取りました。関連するテキストおよびトークンは "Error opening source spreadsheet" です。	SQL 照会のニックネームによって参照されるスプレッドシートをオープンするときに問題が発生しました。CREATE NICKNAME ステートメント時に指定した PATH にファイルが登録時に存在するようにしてください。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1707.<internal program code>" を受け取りました。関連するテキストおよびトークンは "Error accessing DL output storage area" です。	内部プログラム・エラーが発生しました。IBM ソフトウェア・サポート担当者に連絡してください。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1708.<internal program code>" を受け取りました。関連するテキストおよびトークンは "Excel application end failed" です。	内部プログラム・エラーが発生しました。何度照会を繰り返してもこのエラーが続く場合は、IBM ソフトウェア・サポート担当者に連絡してください。

表 19. Excel のラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1711.<internal program code>" を受け取りました。関連するテキストおよびトークンは "Error during fetch, possible data/col type mismatch" です。	SQL 照会の際に取り出されたデータは、ニックネームの登録時に指定したデータ・タイプと異なるデータ・タイプでした。ソース・スプレッドシートのデータを変更するか、またはニックネームに登録したデータ・タイプを変更してください。これによって問題が解決しない場合は、IBM ソフトウェア・サポート担当者に連絡してください。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1900.<internal program code>" を受け取りました。関連するテキストおよびトークンは "Memory allocation error" です。	内部プログラム・エラーが発生しました。IBM ソフトウェア・サポート担当者に連絡してください。

関連資料:

- 25 ページの 『表構造ファイル・ラッパーのメッセージ』
- 64 ページの 『Documentum ラッパーのメッセージ』
- 115 ページの 『BLAST ラッパーのメッセージ』
- 134 ページの 『XML ラッパーのメッセージ』

第 6 章 データ・ソースとしての BLAST

この章では、BLAST とは何か、また BLAST データ・ソースをご使用の連合システムに加える方法などについて解説し、さらに BLAST ラッパーに関連したエラー・メッセージをリストします。

BLAST とは ?

BLAST (Basic Local Alignment Search Tool) は、National Center for Biotechnology Information (NCBI) によってメンテナンスされているユーティリティです。BLAST は、ヌクレオチドまたはアミノ酸のシーケンス・データベースをスキャンして「ヒット」を探すために使用されます。1 つの BLAST ヒットには 1 つまたは複数の高スコア・セグメント・ペア (HSP) が含まれます。HSP は、配列が局所的に最大で、類似性スコアが一定のしきい値を超える、シーケンス・フラグメントのペアです。NCBI は、実行可能ファイル blastall を提供しています。これは GenBank や SWISS-PROT などの BLAST 可能なデータ・ソースで BLAST 検索を実行するために使用されます。

BLAST ラッパーは、BLAST 検索の 5 つのタイプすべて、すなわち BLASTn、BLASTp、BLASTx、tBLASTn、および tBLASTx をサポートします。これらの内容は表 20 に示されています。

表 20. BLAST ラッパーによってサポートされている BLAST 検索タイプ

BLAST 検索タイプ	説明
BLASTn	BLAST 検索の 1 つのタイプ。ヌクレオチド・シーケンスをヌクレオチド・シーケンス・データベースの内容と比較して、オリジナルのシーケンスの領域と同族の領域をもつシーケンス検出します。
BLASTp	BLAST 検索の 1 つのタイプ。アミノ酸シーケンスをアミノ酸シーケンス・データベースの内容と比較して、オリジナルのシーケンスのリージョンと同族の領域をもつシーケンス検出します。
BLASTx	BLAST 検索の 1 つのタイプ。ヌクレオチド・シーケンスをアミノ酸シーケンス・データベースの内容と比較して、オリジナルのシーケンスの領域と同族の領域をもつシーケンス検出します。照会シーケンスは 6 つの読み取りフレームすべてに変換されます。結果の各シーケンスは、シーケンス・データベースを検索するために使用されます。

表 20. BLAST ラッパーによってサポートされている BLAST 検索タイプ (続き)

BLAST 検索タイプ	説明
tBLASTn	BLAST 検索の 1 つのタイプ。アミノ酸シーケンスをヌクレオチド・シーケンス・データベースの内容と比較して、オリジナルのシーケンスの領域と同族の領域をもつシーケンス検出します。シーケンス・データベースの中のシーケンスは 6 つの読み取りフレームすべてで変換されます。結果のシーケンスは、照会シーケンスの領域と同族の領域を検出するために検索されます。
tBLASTx	BLAST 検索の 1 つのタイプ。ヌクレオチド・シーケンスをヌクレオチド・シーケンス・データベースの内容と比較して、オリジナルのシーケンスの領域と同族の領域をもつシーケンス検出します。tBLASTx 検索では照会シーケンスとシーケンス・データベースの両方とも 6 つの読み取りフレームすべてで変換されます。結果のシーケンスは、同族の領域を検出するために比較されます。

図 7 は BLAST がユーザーの連合システムでどう機能するかを示しています。

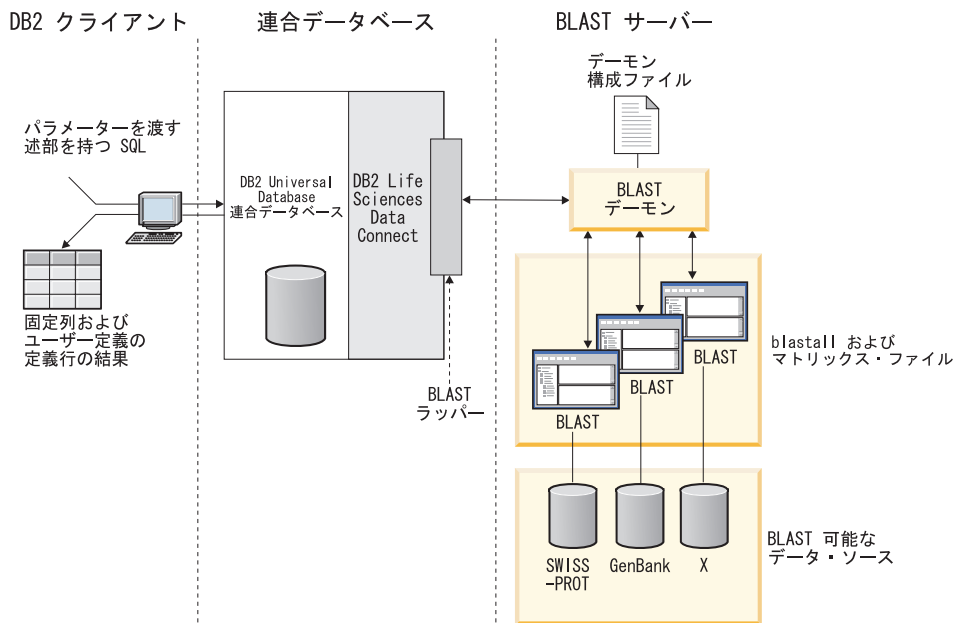


図 7. BLAST ラッパーがどのように機能するか

クライアント・サイドで、ユーザーまたはアプリケーションは、標準の BLAST オプションにマップする BLAST 特有のパラメーターを渡す述部を持つ SQL ステートメントをサブミットします。入力述部を持つ SQL ステートメントは、BLAST ラッパーがインストールされたユーザーの DB2[®] Universal Database 連合データベース・システムへ送信されます。

BLAST ラッパーは、照会を BLAST アプリケーションが認識できるフォーマットに変換し、その変換された照会をユーザーの BLAST サーバーへ送信します。このサーバーは連合システムのマシンとは別のマシンであってもかまいません。特別なデーモン・プログラムがユーザーの BLAST サーバーで実行されます。このデーモンは、デーモン構成ファイルからの情報を使用して、連合システムからの照会要求を受信し、それを BLAST アプリケーションへ送信します。次に BLAST アプリケーションは、BLAST 可能なデータ・ソースに対して通常の方法で実行されます。

結果は BLAST へ戻され、次にデーモンへ戻されます。デーモンは検索されたデータを BLAST ラッパーへ戻します。ラッパーはデータをリレーショナル表フォーマットへ変換し、この表をユーザーまたはアプリケーションへ戻します。戻されたデータは以下の 2 つの部分で構成されています。

- BLAST ユーザーが見なれている一連の標準の固定列、および
- ユーザー構成の定義行情報。

以下の例は、BLAST 可能なデータ・ソースからリレーショナル情報を抽出する方法を示しています。データは、未加工の fasta ファイル・フォーマットから BLAST 可能なデータ・セットへ、そしてユーザーの連合システムの中の他のデータ・ソースと結合可能なリレーショナル表へと移動します。

94 ページの図 8 は fasta ファイルのサンプルで、4 つの定義行およびヌクレオチド・シーケンス・レコードが含まれています。


```

SELECT Unique_ID, Experiment_Number, Organism_Number, HSP_Info, Score
FROM nucleol
WHERE BlastSeq = 'ACATTCTTATAGAGTATTGCTACTCTCCAGGATAGAGTCATCTCT
GGTCTCCAGAGCCACCGCTGGCTACAAGTTGGTGGTGGCGGAGGCTGTGATTGAGAGATTTG
CACCAATACAGAACTCACCTACAGGAGTGAGCGGGTGGTAGAAGGAGCCCGAGATATGAAA
ACCTTGTTC AAGACCCATTGTCACCGGGG';

```

照会の結果は、BLAST ラッパーによって 表 21 に示されているリレーショナル表フォーマットに変換されます。

表 21. BLAST は、ユーザーの連合システムに組み込まれているときは、結果をリレーショナル表フォームで戻します。

固有の ID	実験番号	生体番号	HSP_INFO	スコア
PMON5744	4986	7	一致 = 57/201 (28%)、ポジティブ = 57/201 (28%)、ギ ャップ = 0/201 (0%)	+1.13487000000000E+002
PMON5426	8924	15	一致 = 35/201 (17%)、ポジティブ = 35/201 (17%)、ギ ャップ = 0/201 (0%)	+6.98754000000000E+001
PMON5498	6512	13	一致 = 26/201 (13%)、ポジティブ = 26/201 (13%)、ギ ャップ = 0/201 (0%)	+5.20342000000000E+001

データは完全にリレーショナル・フォームになっており、研究所で使用される他のデータ・ソースからのデータと結合できます。いくつかのデータ・ソースの結果を結合することによって、ユーザーの連合システムのインプリメンテーション以前は容易に、または効率的に発見できなかった洞察が可能になります。

関連概念:

- 13 ページの 『表構造ファイルとは ?』
- 31 ページの 『Documentum とは ?』
- 73 ページの 『Excel とは ?』
- 119 ページの 『XML とは ?』

関連作業:

- 96 ページの 『連合システムへの BLAST の追加』

連合システムへの BLAST の追加

手順:

BLAST データ・ソースを連合サーバーに追加するには、以下のようにします。

1. `blastall` 実行可能ファイルおよびマトリックス・ファイルの正しいバージョンがインストールされていることを検証します。
2. BLAST デーモンを構成します。
3. BLAST デーモンを開始します。
4. `CREATE WRAPPER` ステートメントを使用してラッパーを登録します。
5. オプション: 照会のパフォーマンスを向上させるために `DB2_DJ_COMM` 環境変数を設定します。
6. `CREATE SERVER` ステートメントを使用してサーバーを登録します。
7. `CREATE NICKNAME` ステートメントを使用してニックネームを登録します。

ステートメントは、DB2 コマンド行プロセッサから実行できます。BLAST ラッパーが連合システムに追加されたら、BLAST データ・ソースに対して照会ができます。

関連作業:

- 97 ページの 『blastall 実行可能ファイルとマトリックス・ファイルの正しいバージョンがインストールされていることの検証』
- 97 ページの 『BLAST デーモンの構成』
- 100 ページの 『BLAST デーモンの開始』
- 101 ページの 『BLAST ラッパーの登録』
- 102 ページの 『BLAST ラッパーの DB2_DJ_COMM 環境変数の設定』
- 103 ページの 『BLAST データ・ソースのサーバーの登録』
- 104 ページの 『BLAST データ・ソースのニックネームの登録』
- 16 ページの 『連合システムへの表構造ファイルの追加』
- 33 ページの 『連合システムへの Documentum の追加』
- 75 ページの 『連合システムへの Excel の追加』
- 123 ページの 『連合システムへの XML の追加』

blastall 実行可能ファイルとマトリックス・ファイルの正しいバージョンがインストールされていることの検証

このタスクは、連合システムへの *BLAST* の追加 のためのメインタスクの一部です。blastall 実行可能ファイル、および BLOSUM62、BLOSUM80、PAM30、PAM70 マトリックス・ファイルの最新バージョンが、使用する *BLAST* サーバー・マシンにインストール済みであることを確認します。インストールされていない場合は、バイナリー・ファイルとマトリックス・ファイルをインストールする必要があります。マトリックス・ファイルは、blastall 実行可能ファイルと同じディレクトリーになければなりません。

手順:

blastall 実行可能ファイルとマトリックス・ファイルのバージョン・レベルのチェックは、次のように行います。

1. コマンド行から *BLAST* 検索を実行し、出力ファイルに現れるバージョン番号に注目します。
2. blastall 実行可能ファイルとマトリックス・ファイルの最新バージョンをお持ちで内場合は、NCBI Web サイト <ftp://ftp.ncbi.nih.gov/blast/executables> からファイルをダウンロードします。

ここでのタスクの順序では、次のタスクは『*BLAST* デーモンの構成』です。

関連作業:

- 97 ページの 『*BLAST* デーモンの構成』

BLAST デーモンの構成

このタスクは、連合システムへの *BLAST* の追加 のためのメインタスクの一部です。*BLAST* ラッパーは、DB2 Universal Database 連合システムから TCP/IP でアクセス可能な UNIX ベースのマシン上で、*BLAST* デーモンが稼動していることを必要とします。デーモンは、ラッパーと DB2 Universal Database とは別に実行され、ラッパーからの *BLAST* ジョブ要求を listen します。デーモン実行可能ファイル db2blast_daemon は、*BLAST* サーバー・マシンのどのディレクトリーにあってもかまいません。

DB2 Universal Database インストールの際、デーモン実行可能ファイルは、連合サーバーがインストールされるマシンで、AIX の場合は /usr/opt/db2_08_01/bin ディレクトリーに、その他の UNIX プラットフォームの場合は /opt/IBM/db2/V8.1/bin ディレクトリーに入れます。もし、ご使用の環境で、*BLAST* が別のマシンで実行されている場合は、そのマシン上の選択した場所にデーモンをコピーする必要があります。

BLAST デーモンは以下の権限を持っている必要があります。

- BLAST 検索の実行のため、blastall バイナリー・ファイルに対する実行アクセス権限。
- 一時ファイルに書き込むことができるディレクトリーに対する書き込みアクセス権限。
- BLAST 検索を実行できる対象となる、少なくとも 1 つの BLAST 可能データ・ソースに対する読み取りアクセス権限。blastall 実行可能ファイルは、formatdb プログラムによって生成されたデータ・ファイルと BLAST 索引ファイルの両方に対する読み取りアクセス権限をもっている必要があります。

BLAST デーモンは構成ファイルを必要とします。BLAST_DAEMON.config という名前のサンプル・デーモン構成ファイルが、ディレクトリー DB2PATH/samples/lifesci にあります。ここで、DB2PATH は DB2 Universal Database がインストールされているディレクトリーです。BLAST_DAEMON.config は、そのファイルのデフォルトの名前です。

構成ファイルをデーモンにアクセス可能な任意の場所へコピーします。必要なら名前を変更します。そして、データ・ソースを処理するために編集します。デフォルトにより、blast_daemon は、自分が開始された作業ディレクトリーの中で自分の構成情報を探します。

手順:

デーモンを構成するには、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

これは、ラッパーによってサブミットされた BLAST ジョブ要求をデーモンが listen するネットワーク・ポートです。

MAX_PENDING_REQUESTS

これは、任意の一時点でデーモンでブロックできる BLAST 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 BLAST 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だけです。これを 5 より大きい数に設定することをお勧めします。BLAST デーモンでは、並行して実行できる BLAST ジョブ数の制限はありません。

DAEMON_LOGFILE_DIR

これは、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、BLAST デーモンが生成する有用な状況情報とエラー情報が含まれます。

Q_SEQ_DIR_PATH

これは、デーモンによって一時照会シーケンス・データ・ファイルが作成されるディレクトリーです。この一時ファイルは、BLAST ジョブが完了するとクリーンアップされます。

BLAST_OUT_DIR_PATH

これは、デーモンが BLAST 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です。データはこのファイルから読み取られ、ネットワーク接続経由でラッパーへ戻されます。この時点でデーモンは一時ファイルをクリーンアップします。

BLASTALL_PATH

これは、デーモンを実行しているマシン上の BLAST 実行可能ファイルの完全修飾名です。

データベース指定項目

BLAST 可能データ・ソースの場所を指定します。デーモンが正しく機能するためには、データ・ソースのニックネームを作成するときに構成ファイルの CREATE NICKNAME ステートメントの DATASOURCE オプションで使用する各項目名を指定する必要があります。CREATE NICKNAME ステートメントの詳細については、下記の関連タスク・セクションの『BLAST データ・ソースのニックネームの登録』を参照してください。

構成ファイルには、少なくとも 1 つのデータベース指定項目が、次のフォームで含まれている必要があります。

```
entry_name = path to BLAST-able_data_source
```

たとえば、GenBank BLAST 可能データ・ソースを指定するには、デーモン構成ファイルに次に示す行を追加します。

```
genbank=/dsk/1/nuc1_data/genbank
```

データベース指定項目に示されるパスには、以下のファイルが含まれていなければなりません。

- オリジナルの、fasta フォーマットされたデータ
- 3 つの索引ファイル
 - ヌクレオチド・データ・ソースの場合、索引ファイルは以下の拡張子をもっています。
 - .nhr
 - .nin
 - .nsq
 - アミノ酸データ・ソースの場合、索引ファイルは以下の拡張子をもっています。
 - .phr
 - .pin
 - .psq

データベース指定項目は、オリジナルの Fasta フォーマットされたデータを含んでいるファイルのファイル名を示していなければなりません。3つの索引ファイルは、オリジナルの Fasta フォーマットされたデータを含んでいるファイルと同じルート・ファイル名をもっていなければなりません。

構成ファイルは改行文字で終わらなければなりません。

例:

以下の例は、必須オプションおよび GenBank と SWISS-PROT についての BLAST 可能データ・ソース指定のついたサンプル構成ファイルの内容を示しています。

```
DAEMON_PORT=4007
MAX_PENDING_REQUESTS=10
DAEMON_LOGFILE_DIR=./
Q_SEQ_DIR_PATH=./
BLAST_OUT_DIR_PATH=./
BLASTALL_PATH=./blastall
genbank=/dsk/1/nuc1_data/genbank
swissprot=/dsk/1/prot_data/swissprot
```

ここでのタスクの順序では、次のタスクは『BLAST デーモンの開始』です。

関連作業:

- 97 ページの 『blastall 実行可能ファイルとマトリックス・ファイルの正しいバージョンがインストールされていることの検証』
- 100 ページの 『BLAST デーモンの開始』
- 104 ページの 『BLAST データ・ソースのニックネームの登録』

BLAST デーモンの開始

このタスクは、**連合システムへの BLAST の追加** のためのメインタスクの一部です。BLAST データ・ソースへアクセスできるためには、BLAST デーモンが実行されていなければなりません。

前提条件:

BLAST デーモンを開始するには、構成ファイルの中の DAEMON_LOGFILE_DIR、BLAST_OUT_DIR_PATH、および Q_SEQ_DIR_PATH 項目の下に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をもっている必要があります。

手順:

デーモン・インストール・ディレクトリーにあり、デーモン構成ファイルの名前を変更せず、また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ーにある場合、BLAST デーモンを開始するには、コマンド行に以下のコマンドを入力します。

db2blast_daemon

実行可能ファイルは、BLAST デーモンが実行される新規プロセスを開始します。

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、デーモン構成ファイルが配置されているディレクトリーにない場合、BLAST デーモンを開始するには、ラッパー・デーモン・コマンドで `-c` オプションを使用して、デーモン実行可能ファイルが新しい名前または新しい場所を指すようにする必要があります。

たとえば、以下のコマンドは、ラッパー・デーモンがその構成情報をサブディレクトリー `cfg` 中の `BLAST_D.config` というファイルの中で探すようにさせます。

```
db2blast_daemon -c cfg/BLAST_D.config
```

ここでのタスクの順序では、次のタスクは『BLAST ラッパーの登録』です。

関連作業:

- 97 ページの 『BLAST デーモンの構成』
- 101 ページの 『BLAST ラッパーの登録』

BLAST ラッパーの登録

このタスクは、連合システムへの *BLAST* の追加 のためのメインタスクの一部です。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、連合サーバーがデータ・ソースとの通信やデータ・ソースからのデータの検索に使用するメカニズムです。ラッパーは、ライブラリー・ファイルとして、システムにインストールされています。

手順:

BLAST ラッパーを登録するには、`CREATE WRAPPER` ステートメントをサブミットします。

たとえば、デフォルト・ライブラリー・ファイル `libdb2lsblast.a` から `my_blast` という AIX 上の BLAST ラッパーを作成するには、次のステートメントをサブミットします。

```
CREATE WRAPPER my_blast LIBRARY 'libdb2lsblast.a'  
OPTIONS(DB2_FENCED 'N');
```

サポートされているプラットフォーム別の BLAST ラッパーのデフォルト・ライブラリー・ファイル名の表については、下記の関連タスク・セクションの『DB2 Life Science Data Connect をインストールした後』を参照してください。 `CREATE WRAPPER` ステートメントに関して詳しくは、「DB2 SQL リファレンス」を参照してください。

ここでのタスクの順序では、次のタスクは『*BLAST* ラッパーの *DB2_DJ_COMM* 環境変数の設定』です。

関連作業:

- 16 ページの 『表構造ファイル・ラッパーの登録』
- 36 ページの 『Documentum ラッパーの登録』
- 75 ページの 『Excel ラッパーの登録』
- 100 ページの 『*BLAST* デーモンの開始』
- 102 ページの 『*BLAST* ラッパーの *DB2_DJ_COMM* 環境変数の設定』
- 123 ページの 『XML ラッパーの登録』

BLAST ラッパーの *DB2_DJ_COMM* 環境変数の設定

このタスクは、連合システムへの *BLAST* の追加 のためのメインタスクの一部です。*BLAST* データ・ソースにアクセスする場合のパフォーマンスを向上させるために、*DB2_DJ_COMM* 環境変数を設定します。この変数は、初期設定時に連合サーバーがラッパーをロードするかどうかを決定します。

手順:

DB2_DJ_COMM 環境変数を設定するには、関連する `CREATE WRAPPER` ステートメントに指定したラッパーに対応するラッパー・ライブラリーを指定して `db2set` コマンドをサブミットします。

たとえば、次のようにします。

```
db2set DB2_DJ_COMM='libdb2lsblast.a'
```

等号 (=) の両側にスペースを入れないようにしてください。

データベース始動時にラッパー・ライブラリーをロードすることに関連したオーバーヘッドがあります。このオーバーヘッドを回避するには、アクセスしたいライブラリーのみを指定します。

DB2_DJ_COMM 環境変数に関して詳しくは、「*DB2* 管理ガイド」を参照してください。

ここでのタスクの順序では、次のタスクは『*BLAST* データ・ソースのサーバーの登録』です。

関連作業:

- 17 ページの 『表構造ファイル・ラッパーの *DB2_DJ_COMM* 環境変数の設定』
- 37 ページの 『Documentum ラッパーの *DB2_DJ_COMM* 環境変数の設定』

- 101 ページの 『BLAST ラッパーの登録』
- 103 ページの 『BLAST データ・ソースのサーバーの登録』
- 124 ページの 『XML ラッパーの DB2_DJ_COMM 環境変数の設定』

BLAST データ・ソースのサーバーの登録

このタスクは、連合システムへの *BLAST* の追加 のためのメインタスクの一部です。ラッパーが登録された後、対応するサーバーを登録する必要があります。

手順:

BLAST サーバーを連合システムに登録するには、`CREATE SERVER` ステートメントを使用します。

ご使用の環境において、BLAST 可能ファイルとデーモンがインストールされている各マシンについて、その BLAST 実行可能ファイルとデーモンのインスタンスを使用して実行したい BLAST 検索のタイプごとに 1 つのサーバーを登録する必要があります。

たとえば、BLASTn 検索に使用される `CREATE WRAPPER` ステートメントを使用して作成した `my_blast` ラッパーの `blast_server1` というサーバーを登録するには、以下のステートメントをサブミットします。

```
CREATE SERVER blast_server1
TYPE blastn
  VERSION 2.1.2
  WRAPPER my_blast
  OPTIONS (NODE 'big_rs.company.com', PORT '4007')
```

引き数

TYPE 所定のサーバーを使用して実行される BLAST 検索のタイプを決定します。この引き数は必須です。これは、`blastn`、`blastp`、`blastx`、`tblastn`、`tblastx` のうちのいずれかの値に設定されなければなりません。

VERSION

使用しようとするサーバーのバージョンを指定します。これは、実行している `blastall` のバージョンに設定する必要があります。この引き数は必須です。

WRAPPER

`CREATE WRAPPER` ステートメントを使用して登録したラッパー名を指定します。この引き数は必須です。

オプション

NODE BLAST デーモン・プロセスを実行中のシステムのホスト名を指定します。このオプションは必須です。

PORT BLAST ジョブ要求をデーモンが `listen` するネットポート番号を指定します。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。デフォルトは 4007 です。このオプションは任意です。

CREATE SERVER ステートメントに関して詳しくは、「*DB2 SQL* リファレンス」を参照してください。

ここでのタスクの順序では、次のタスクは『*BLAST* データ・ソースのニックネームの登録』です。

関連作業:

- 18 ページの 『表構造ファイルのサーバーの登録』
- 38 ページの 『Documentum データ・ソースのサーバーの登録』
- 76 ページの 『Excel データ・ソースのサーバーの登録』
- 102 ページの 『BLAST ラッパーの DB2_DJ_COMM 環境変数の設定』
- 104 ページの 『BLAST データ・ソースのニックネームの登録』
- 125 ページの 『XML データ・ソースのサーバーの登録』

BLAST データ・ソースのニックネームの登録

このタスクは、*連合システムへの BLAST の追加* のためのメインタスクの一部です。サーバーを登録した後、対応するニックネームを登録する必要があります。ニックネームは、照会の中で *BLAST* データ・ソースを参照する場合に使用されます。

手順:

BLAST 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。 . *BLAST* 検索の各タイプは別々のサーバーで取り扱われるので、所定の *BLAST* 可能データ・ソースに対して実行したい *BLAST* 検索の各タイプごとに別々のニックネームを定義する必要があります。

ニックネームは、データ・ソースの定義行部分の列情報を指定します。他のすべての列は固定されています。定義行構文解析の詳細については、106 ページの『定義行構文解析』を参照してください。固定列の詳細については、107 ページの『固定列』を参照してください。

BLAST の `CREATE NICKNAME` ステートメントの構文は以下のとおりです。

```
→ CREATE NICKNAME—nickname—(—column-name— | column-information | —)→
```

```

▶-FOR SERVER—server-name—OPTIONS—(—DATASOURCE—'data_source_name' —,—————▶
▶-TIMEOUT—'timeout_duration' —)—————▶

```

column-information:

```

|—| data-type |—| column-option |—| nickname-column-options |—————|

```

data-type:

```

|—————|
|—| INTEGER |—————|
|—| INT |—————|
|—| FLOAT |—————|
|—| DOUBLE |—————|
|—| VARCHAR—(integer) |—————|
|—| PRECISION |—————|
|—| (integer) |—————|

```

column-option:

```

|—————|
|—| NOT NULL |—————|

```

nickname-column-options:

```

|—| OPTIONS—(—INDEX—'index_number' —, —DELIMITER—'delimiter' —————▶
|—| DEFAULT—'new_default_value' —)—————▶

```

CREATE NICKNAME に関して詳しくは、「DB2 SQL リファレンス」を参照してください。

ニックネーム列オプション

ニックネーム列オプションの値は、単一引用符で囲む必要があります。

INDEX このオプションが定義行列のグループの中で表示される列の序数。このオプションは必須です。定義行構文解析の詳細については、106 ページの『定義行構文解析』を参照してください。

DELIMITER

このオプションが表示される列の定義情報の終点を決定するために使用するべき区切り文字。このオプションの値に複数の文字がある場合は、それらの文字のうちのいずれかが最初に現れる位置がこのフィールドの情報の終端を表します。デフォルトは行の終わりです。このオプションは必須です。ただし、そ

の列に定義行の残りを入れたい場合に指定した最後の列を除きます。定義行構文解析の詳細については、『定義行構文解析』を参照してください。

DEFAULT

以下の入力固定列の新規デフォルト値を指定します。

- E_value
- QueryStrands
- GapAlign
- NMisMatchPenalty
- NMatchReward
- Matrix
- FilterSequence
- NumberOfAlignments
- GapCost
- ExtendedGapCost
- WordSize
- ThresholdEx

この新しい値は事前設定のデフォルト値をオーバーライドします。新しいデフォルト値は、与えられた列に示された値と同じタイプのものでなければなりません。入力固定列の詳細については 107 ページの『入力固定列』を参照してください。このオプションは任意です。

ニックネーム・オプション

ニックネーム・オプション値は、単一引用符で囲む必要があります。

DATASOURCE

BLAST 検索実行の対象のデータ・ソース名。ここで使用される正確なストリングが BLAST デーモンの構成ファイルに存在している必要があります。構成ファイルの詳細については、下記の関連タスク・セクションの『BLAST デーモンの構成』を参照してください。このオプションは必須です。

TIMEOUT

BLAST ラッパーがデーモンからの結果を待機する最大時間 (分表示)。デフォルトは 60 です。このオプションは任意です。

定義行構文解析

定義行 (define と呼ばれます) は、BLAST 可能データ・ソースの各シーケンスに対するキーのようなもので、各 BLAST ヒットの一部として戻されます。

定義行情報をユーザーの結果表に入れたい場合は、CREATE NICKNAME ステートメントに定義行列を指定する必要があります。各列の指定は、INDEX オプションを指定する必要があります。各列に対して DELIMITER オプションを指定する必要があります。ただし、その列に定義行の残りを入れたい場合に指定した最後の列を除きます。

INDEX および DELIMITER オプションの詳細については、105 ページの『ニックネーム列オプション』を参照してください。

定義行フィールドは、integer、float、double、または varchar のタイプのものでなければなりません。

注: BLAST ヒットの Accession 番号フィールドの中にデータが検出された場合、これらのデータは、その BLAST ヒットの Definition フィールドの中のデータの前に挿入されます。Accession 番号データとそれに続く Definition フィールド・データを含んでいる結果の定義行は、ラッパーによって構文解析されます。

CREATE NICKNAME ステートメントに定義行列を指定する方法の例については、111 ページの『CREATE NICKNAME の例』を参照してください。

固定列

CREATE NICKNAME ステートメントは、自動的に固定列を作成します。固定列は、CREATE NICKNAME ステートメントの中には表れません。しかし、ニックネーム定義の一部であり、SQL 照会で参照できます。固定列には入力と出力の 2 つのタイプがあります。

入力固定列

入力固定列は、SQL 照会のパラメーター受け渡し述部として使用されます。それら固定列は、標準の BLAST スイッチを BLAST に渡します。次に、これらのスイッチを使用して、指定されたデータ・ソースに対して BLAST が実行されます。入力固定列は、照会選択リストの中で参照することもできます。そして、結果表の一部として戻すことができます。入力固定列は、表 22 にリストされています。

表 22. 入力固定列

名前	データ・タイプ	説明
BlastSeq	varchar(32000)	照会シーケンスを BLAST ラッパーに渡します。
E_Value	double	入力および出力パラメーターの両方。入力パラメーターとしては、この列は、blastall から戻されるべき期待値の上限を BLAST ラッパーに示します。

表 22. 入力固定列 (続き)

名前	データ・タイプ	説明
QueryStrands	integer	BLASTn 検索を実行するときと比較されるべき成分を指定します。1 の値は、トップの成分を使用すべきことを示します。2 は、ボトム成分を示し、そして 3 は両方の成分が比較されるべきことを示します。
GapAlign	char(1)	BLAST 出力でギャップのある配列が許されるかどうかを、ラッパーに指示します。
Matrix	varchar(50)	アミノ酸のペアの間での類似性の程度を決定するために、blastall によって使用される置換マトリックスを決定します。アミノ酸をアミノ酸と比較する BLAST 検索タイプだけが、この述部を使用します。
NMismatchPenalty	integer	同族領域の中のヌクレオチドのペアの 1 つが一致しない場合に、配列のスコアから blastall が差し引く値を指定します。ヌクレオチドをヌクレオチドと比較する BLAST 検索タイプだけが、この述部を使用します。
NMatchReward	integer	同族領域の中の一致するヌクレオチドのペアのそれぞれに対して、配列のスコアに blastall が加える値を指定します。ヌクレオチドをヌクレオチドと比較する BLAST 検索タイプだけが、この述部を使用します。
FilterSequence	char(1)	生物学的に関心を引かないセグメントを、照会シーケンスから除去するためのフィルター操作を実行するかどうかを、blastall に指示します。検索タイプが BLASTn の場合、使用するフィルターは DUST です。それ以外の場合、フィルター操作は SEG によって実行されます。
NumberOfAlignments	integer	BLAST 出力に含める HSP 配列の数を指定します。
GapCost	integer	配列の長さを伸ばすことを可能にするために、照会シーケンスまたはヒット・シーケンスのどちらかにギャップを導入する必要がある場合に、配列のスコアから blastall が差し引く値を指定します。

表 22. 入力固定列 (続き)

名前	データ・タイプ	説明
ExtendedGapCost	integer	配列の長さを伸ばすことを可能にするために、照会シーケンスまたはヒット・シーケンスのどちらかにすでに導入されているギャップを、ヌクレオチドまたはアミノ酸の 1 つ分だけ拡張する必要がある場合に、配列のスコアから blastall が差し引く値を指定します。
WordSize	integer	blastall が最初にデータベースで検索する初期ヒットの長さを blastall に示します。
ThresholdEx	integer	BLAST が、そのしきい値より下へはそれ以上ヒットを拡張することをしないというスコアのしきい値を示します。

各入力固定列に対するサポートされている BLAST 検索タイプとスイッチについては、表 23 にリストされています。

表 23. 入力固定列によってサポートされている BLAST 検索タイプおよびスイッチ

名前	BLAST 検索タイプ	BLAST スイッチ	必須 ?	デフォルト
BlastSeq	n, p, x, tn, tx	-l	Y	N/A
E_Value	n, p, x, tn, tx	-e	N	10
QueryStrands	n	S	N	3
GapAlign	n, p, x, tn, tx	-g	N	T
Matrix	p, x, tn, tx	-n	N	BLOSUM62
NMismatchPenalty	n	-q	N	-3
NMatchReward	n	-r	N	1
FilterSequence	n, p, x, tn, tx	-F	N	T
NumberOfAlignments	n, p, x, tn, tx	-b	N	250
GapCost	n, p, x, tn, tx	-G	N	11
ExtendedGapCost	n, p, x, tn, tx	-E	N	1
WordSize (Blastn の場合、7 より小さい値は無効)	n, p, x, tn, tx	-W	N	11 -BLASTn 3 -BLASTp
ThresholdEx	n, p, x, tn, tx	-f	N	0

出力固定列

出力固定列は、照会結果表の中に戻され、述部として使用できます。出力固定列は 表 24 にリストされています。

表 24. 出力固定列

名前	データ・タイプ	説明
Score	double	BLAST 結果に報告されている HSP の計算スコア。
E_value	double	入力および出力パラメーターの両方。出力パラメーターとして、この列は BLAST 結果に報告されている HSP の計算スコアを提供します。
Length	integer	BLAST 結果に報告されるヒット・シーケンスの長さ。
HSP_Info	varchar(100)	BLAST によって報告される、与えられた HSP の情報ストリング。このストリングには、照会シーケンスとヒット・シーケンスの間で一致した、ヌクレオチドまたはアミノ酸の数についての情報が含まれています。
HSP_Q_Start	integer	照会シーケンス上の最初の同族ヌクレオチドまたはアミノ酸の数字で表した位置。
HSP_Q_End	integer	照会シーケンス上の最後の同族ヌクレオチドまたはアミノ酸の数字で表した位置。
HSP_Q_Seq	varchar(32000)	HSP_Q_Start で始まり HSP_Q_End で終わる照会シーケンスのセグメント。
HSP_H_Start	integer	ヒット・シーケンス上の最初の同族ヌクレオチドまたはアミノ酸の数字で表した位置。
HSP_H_End	integer	ヒット・シーケンス上の最後の同族ヌクレオチドまたはアミノ酸の数字で表した位置。
HSP_H_Seq	varchar(32000)	HSP_H_Start で始まり HSP_H_End で終わるヒット・シーケンスのセグメント。

表 24. 出力固定列 (続き)

名前	データ・タイプ	説明
HSP_Midline	varchar(32000)	照会シーケンスおよびヒット・シーケンスの同族領域における各位置でのアミノ酸間またはヌクレオチド間での同族関係の程度を示す BLAST によるストリング出力。

CREATE NICKNAME の例

以下の CREATE NICKNAME ステートメントは、ニックネーム genbank を定義します。

ここでは、BLAST 結果の定義フィールドに、以下の情報が入っているものと想定しています。

```
>276342 15:8924 PMON5426
```

ここでは、それぞれ以下のことを意味しています。

276342 BLAST 結果の Accession フィールド。

15:8924 PMON5426

有機体番号、それに続く実験番号そしてその次に固有 ID が含まれている、BLAST 結果の中の定義フィールド。

この情報により、以下のニックネームが作成されます。

```
CREATE NICKNAME genbank (
  acc_num integer OPTIONS(INDEX '1', DELIMITER ' '),
  org_num integer OPTIONS(INDEX '2', DELIMITER ':'),
  exp_num integer OPTIONS(INDEX '3', DELIMITER ' '),
  u_id varchar(10) OPTIONS(INDEX '4'))
FOR SERVER blast_server1
OPTIONS(DATASOURCE 'genbank', TIMEOUT '300');
```

列 acc_num には 276342、列 org_num には 15、列 exp_num には 8924、そして列 u_id には PMON5426 が、それぞれ入ることになります。

CREATE NICKNAME ステートメントをサブミットした後、ニックネーム genbank を使用して連合システムで照会することができます。連合システムで、genbank ニックネームを他のニックネームや表と結合することもできます。

ここでの作業は、これで終わりです。

関連作業:

- 19 ページの 『表構造ファイルのニックネームの登録』

- 41 ページの 『Documentum データ・ソースのニックネームの登録』
- 77 ページの 『Excel データ・ソースのニックネームの登録』
- 97 ページの 『BLAST デーモンの構成』
- 103 ページの 『BLAST データ・ソースのサーバーの登録』
- 126 ページの 『XML データ・ソースのニックネームの登録』
- 143 ページの 『第 8 章 ニックネームのコスト計算オプションの指定』

BLAST SQL 照会の作成

BLAST データ・ソースの SQL には、標準の BLAST スイッチを `blastall` 実行可能ファイルへ渡すために使用される特別な入力述部のみが含まれていなければなりません。

制約事項:

有効であるためには、BLAST ラッパーへ渡されるすべての照会は、少なくとも `BlastSeq` 入力述部を含んでいなければなりません。他のすべての述部はオプションです。

手順:

BLAST 照会を作成するには、SQL ステートメントの `WHERE` 文節に入力述部を使用します。

以下の例は、3 つの入力述部 `BlastSeq`、`GapCost`、および `NMismatchPenalty` を示しています。

```
Select * from blast b where
BlastSeq = 'GTCCAGCC...' AND
GapCost = -10 AND
NMismatchPenalty = -4;
```

各入力述部のサポートされているデータ・タイプ、説明、BLAST スイッチ、および検索タイプのリストについては、下記の関連タスク・セクションの『BLAST データ・ソースのニックネームの登録』を参照してください。

関連作業:

- 113 ページの 『サンプル BLAST 照会』

サンプル BLAST 照会

BLAST データ・ソースに対して照会がどのように作成されるかを示すために、いくつかのサンプルの BLAST 照会を以下に示します。

手順:

照会を実行するために、ガイドとして以下の例を使用します。

これらの照会において、各ニックネームに使用される名前は、BLAST 検索のタイプとデータ・ソースを示しています。サンプル照会ごとに登録ステートメントをリストする必要をなくするために、そのようにしています。また、照会の中には他の仮想のデータ・ソースを使用しているものがあります。これは、これらのサンプルが、他のデータ・ソースと結合されたときのラッパーの動作を示すことができるようにするためです。

照会 1

```
select *
from blastn_genbank
where BlastSeq =
'caaccctccagccgagttgtcaatggcgaggaagctgttccccac';
```

この SQL ステートメントが実行されると、ラッパーは、指示されているシーケンスを使用して、GenBank の BLASTn 検索を実行します。ラッパーは、入力パラメーター列および BLAST 結果列の両方を含むすべての選択可能な列を戻します。

照会 2

```
select *
from blastn_genbank
where BlastSeq =
'caaccctccagccgagttgtcaatggcgaggaagctgttccccac'
and GapCost = 8 and NmisMatchPenalty = -4;
```

この SQL ステートメントが実行されると、ラッパーは、指示されているシーケンスを使用して、GenBank の BLASTn 検索を実行します。さらに、ラッパーは指示されている 2 つのパラメーターをデーモンへ渡します。そして、これらのパラメーターは blastall コマンド行へ渡されます。ラッパーは、入力パラメーター列および BLAST 結果列の両方を含むすべての選択可能な列を戻します。

照会 3

```
select blp.*
from blastp_swissprot blp, protein_db prdb
where prdb.keyword = 'malic enzyme'
and blp.BlastSeq = prdb.sequence;
```

この SQL ステートメントが実行されると、ラッパーは、仮想タンパク質シーケンス・データベースから戻されるシーケンスの数によって、SWISS-PROT のゼロまたは 1 回以上の BLASTp 検索を実行します。このステートメントは、DB2 によって 2 つの別々の照会に分割されます。1 つの BLASTp 検索は、仮想タンパク質データベースから戻される各行に対して実行されます。ラッパーは、入力パラメーター列および BLAST 結果列の両方を含むすべての選択可能な列を戻します。

照会 4

```
select Score, E_Value, HSP_Info, HSP_Q_Seq, HSP_H_Seq, HSP_Midline
from blastx_swissprot
where BlastSeq = 'gagttgtcaatggcgagg'
and GapCost = 8;
```

この SQL ステートメントが実行されると、ラッパーは、指示されているシーケンスを使用して SWISS-PROT の BLASTx 検索を実行します。この場合、blastall は、6 つのすべての読み取りフレームの中の入力シーケンスを変換し、6 つの新しく作成されたタンパク質シーケンスのそれぞれを使用して、同族関係検索を実行します。結果の HSP には、アミノ酸 - アミノ酸の配列が含まれます。ヌクレオチド - ヌクレオチドの配列ではありません。提供されたパラメーターはデーモンへ渡され、次にコマンド行経由で blastall へ渡されます。ラッパーは、照会の中で特定の要求されている列だけを戻します。

照会 5

```
select tblx.Score, tblx.E_Value, tblx.HSP_Info tblx.HSP_Q_Seq,
HSP_H_Seq, HSP_Midline
from tblastx_genbank tblx, gen_exp_database gedb
where tblx.BlastSeq = gedb.sequence
and gedb.organism = 'interesting organism'
and GapCost = 8
and FilterSequence = 'F';
```

この SQL ステートメントが実行されると、ラッパーは、仮想遺伝子表示データベースから戻されるシーケンスの数によって、GenBank のゼロまたは 1 回以上の tBLASTx 検索を実行します。このステートメントは、DB2 によって 2 つの別々の照会に分割されます。1 つの tBLASTx 検索は、仮想遺伝子表示データベースから戻される各行に対して実行されます。この場合、blastall は、入力シーケンスおよび 6 つのすべての読み取りフレームの中の GenBank のすべてのシーケンスを変換し、6 つの新しく作成された照会タンパク質シーケンスのそれぞれと、新しく作成されたすべてのデータベースタンパク質シーケンスを使用して同族関係検索を実行します。結果の HSP には、アミノ酸 - アミノ酸の配列が含まれます。ヌクレオチド - ヌクレオチドの配列ではありません。提供されたパラメーターはデーモンへ渡され、次にコマンド行経由で blastall へ渡されます。ラッパーは、照会の中で特定の要求されている列だけを戻します。

関連作業:

- 57 ページの 『Documentum データ・ソースに対する照会の実行』

- 79 ページの 『Excel データ・ソースに対する照会の実行』
- 132 ページの 『XML データ・ソースに対する照会の実行』

BLAST ラッパーの最適化のヒント

同一サーバー上でラッパーとデーモンの両方を実行すると、ネットワーク通信のボトルネックの可能性を除去することができます。

関連資料:

- 25 ページの 『表構造ファイル・ラッパーに関する最適化のヒントと考慮事項』

BLAST ラッパーのメッセージ

このセクションでは、BLAST のラッパーを処理するときに出されるメッセージをリストして説明します。メッセージの詳細については、「DB2 メッセージ・リファレンス」を参照してください。

表 25. BLAST のラッパーによって出されるメッセージ

エラー・コード	メッセージ	説明
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できます。(理由 "sqlno_crule_save_plans [100]:rc (-2144272209) Empty plan list detect".)	DB2 にサブミットされた SQL 照会はラッパーによって処理できませんでした。構文を訂正して再サブミットしてください。
SQL1816N	ラッパー "BLAST_WRAPPER" は、連合データベースに定義しようとしているデータ・ソース (" <code><server type></code> " "") の "type" にアクセスするためには使用できません。	CREATE SERVER ステートメントが無効な TYPE を使用していました。タイプは、サポートされている BLAST タイプの中のいずれか 1 つでなければなりません。
SQL1817N	CREATE SERVER ステートメントは、連合データベースに定義したいデータ・ソースの "version" を示していません。	CREATE SERVER ステートメントがバージョンを指定していませんでした。

表 25. *BLAST* のラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL1822N	データ・ソース "Blast Wrapper" から予期しないエラー・コード "Unspecified Error" を受け取りました。関連するテキストおよびトークンは "Unable to connect to daemon" です。	blast ラッパーはデーモンに接続できませんでした。デーモンが実行されていなかった可能性があります。構成が間違っている可能性があります。デーモンが実行されているマシンに到達できなかった可能性があります。
SQL1822N	データ・ソース "Blast Wrapper" から予期しないエラー・コード "Unspecified Error" を受け取りました。関連するテキストおよびトークンは "Blast daemon timeout expired" です。	CREATE NICKNAME ステートメントに指定されているタイムアウトが経過する前に、デーモンから何の結果も受け取りませんでした。タイムアウトを長くするか、デーモンに問題があるかどうかをチェックしてください。
SQL1822N	データ・ソース "Blast Wrapper" から予期しないエラー・コード "Unspecified Error" を受け取りました。関連するテキストおよびトークンは "Blast Daemon Failed" です。	デーモンが通信を停止したか、戻された結果が正しくフォーマットされていませんでした。
SQL1822N	データ・ソース "Blast Wrapper" から予期しないエラー・コード "Unspecified Error" を受け取りました。関連するテキストおよびトークンは "Unknown error from the blast daemon" です。	blast ラッパーが、認識できないエラー・コードをデーモンから受け取りました。デーモンのバージョンがラッパーのバージョンと互換性がなかった可能性があります。
SQL1822N	データ・ソース "Blast Wrapper" から予期しないエラー・コード "Unspecified Error" を受け取りました。関連するテキストおよびトークンは "Column rename not allowed" です。	列の中の 1 つの名前を変更しようとして、ALTER NICKNAME ステートメントが出されました。列の名前変更は許可されていません。

表 25. BLAST のラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL1822N	データ・ソース "Blast Wrapper" から予期しないエラー・コード "Unspecified Error" を受け取りました。関連するテキストおよびトークンは "XML parser error" です。	Xerces パーサーが無効な状態にあるか、または例外をスローしました。
SQL1823N	サーバー "<server name>" からのデータ・タイプ "<data type name>" のデータ・タイプ・マッピングがありません。	指定したデータ・タイプはこの列ではサポートされません。
SQL1881N	"DEFAULT" は "<column-name>" の有効な "COLUMN" オプションではありません。	DEFAULT オプションが、それをサポートしない列で使用されていました。出力専用の列および定義行の列はデフォルト値をもちません。
SQL1882N	"COLUMN" オプション "DEFAULT" は、"<column-name>" の "<option-value>" に設定することはできません。	DEFAULT オプションに指定された値が、列に対して非互換のタイプの値であるか、間違っってフォーマットされています。

関連資料:

- 25 ページの 『表構造ファイル・ラッパーのメッセージ』
- 64 ページの 『Documentum ラッパーのメッセージ』
- 83 ページの 『Excel ラッパーのメッセージ』
- 134 ページの 『XML ラッパーのメッセージ』

第 7 章 データ・ソースとしての XML

この章では、XML とは何か、また XML データ・ソースをご使用の連合システムに加える方法などについて解説し、さらに XML ラッパーに関連したエラー・メッセージをリストします。

XML とは ?

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 ファイルは `xml` のファイル拡張子をもっています。HTML のように、XML は文書の中のデータを構造化するために、タグ ('<' および '>') の括弧に入れた語) を使用します。サンプル XML 文書が 図 9 に示されています。

```
<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 9. サンプル XML 文書

XML ラッパーにより、ファイルに保管されている外部 XML 文書を照会するために SQL が使用できるようになります。図 10 は、XML が連合システムを処理する方法を示しています。

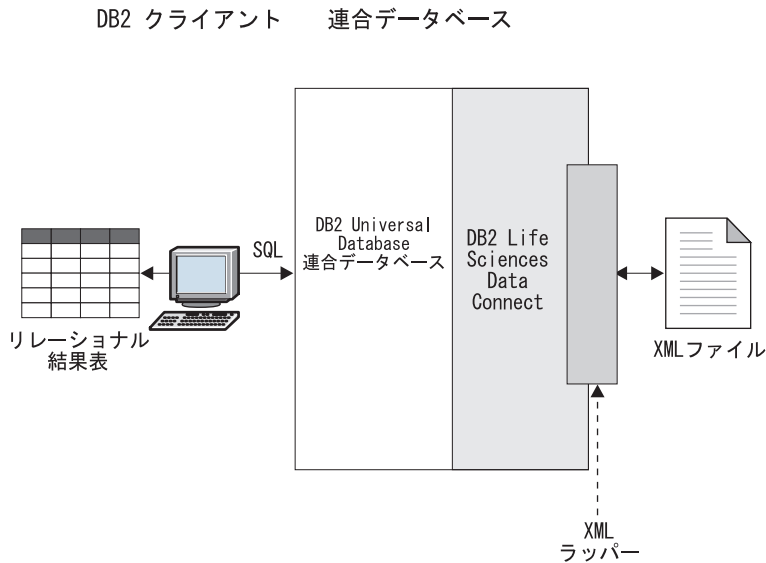


図 10. XML ラッパーがどのように機能するか

XML ラッパーは、外部データ・ソースからの XML データ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・スキーマへのマッピングを可能にします。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、親ニックネームに対応するエレメントにネストされているエレメントを子ニックネームがモデルとするツリーに編成されます。

ネストされたエレメントが別のニックネームとしてモデルになる必要がある場合、基本的には次の 2 つのケースがあります。

- 繰り返しエレメント
- 明白な識別およびリッチ構造をもつエレメント

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 / 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 ファイルの部分 (たとえ

ば、XML 文書ツリー内のノードおよび属性のグループ) 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lementを識別する 1 つの XPath 式、および各Element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。

例:

次の例は 119 ページの図 9 に示されているサンプル XML 文書が、ニックネームのセットにマップされる方法、主および外部キーを使用して親子関係がモデル化される方法、文書の各Element内の個々のタプルおよび列を定義するために XPath 式を使用する方法、ならびに XML 文書がいったん連合システムに登録された後、その XML 文書に対して照会を実行できる方法を示しています。

サンプル XML 文書には、それぞれがいくつかの order および payment Elementを囲んでいる customer Elementのセットが含まれています。

order Elementは、いくつかの item Elementを囲んでいます。

Element間の関係は 図 11 に示されています。

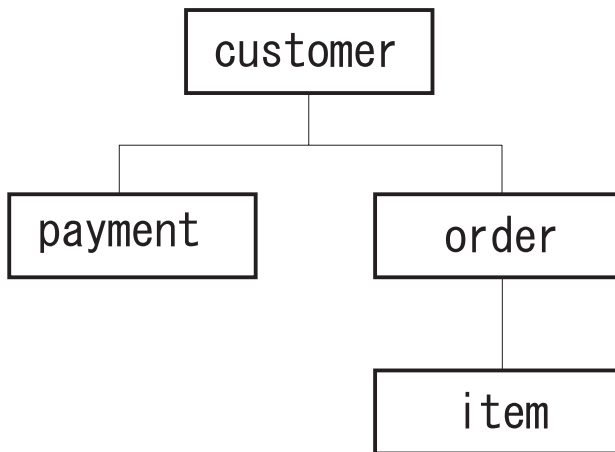


図 11. サンプル XML 文書のツリー構造

この構造から、CREATE NICKNAME ステートメントを使用して、XML 文書を、以下の 4 つのニックネームを使用してリレーショナル・スキーマにマップすることができます。

- customers
- orders

- payments
- items

ニックネーム間の関係は、それぞれ特別な主および外部キー・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。ある親の子は、外部キー列オプションを使用して親ニックネームの主キー列を参照する、特別な列をもつことによって定義されます。指定された主および外部ニックネーム列は、ユーザーの 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。ニックネームは複数の子をもつことができますが、親については、1つだけをもたなければなりません。ルートはその例外で、親をもちません。

サンプル XML 文書については、customers ニックネームには主キーが定義され、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ます。orders および payments ニックネームは customers を指す外部キーをもち、items ニックネームは orders を指す外部キーをもつこととなります。

個々のタプルを表す XML エlementを識別するために、1つの XPath 式が作成されます。この例では、すべての customer エlementは、//customer XPath 式を使用して参照でき、すべての order エlementは、.//order XPath 式を使用して参照できません。

各Elementから列値を取り出す方法を指定するために、XPath 式のセットが作成されます。この例では、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) は、./@id XPath 式を使用して参照できます。customer Elementの name Elementは、.//name XPath 式を使用して参照でき、customer Elementの address Elementは、.//address/@street XPath 式を使用して参照できます。

CREATE NICKNAME ステートメントを使用して XML 文書がいったんニックネームのセットにマップされると (各ニックネームは、文書の各Element内の個々のタプルおよび列を定義する XPath 式を使用して、主および外部キーを使用して親または子として定義されている)、XML 文書に対して SQL 照会を実行できます。

ニックネームの作成方法に関する詳細情報および CREATE NICKNAME ステートメントの構文については、下記の関連タスク・セクションの『XML データ・ソースのニックネームの登録』を参照してください。

関連概念:

- 13 ページの 『表構造ファイルとは ?』
- 31 ページの 『Documentum とは ?』
- 73 ページの 『Excel とは ?』

- 91 ページの 『BLAST とは ?』

関連作業:

- 123 ページの 『連合システムへの XML の追加』
- 126 ページの 『XML データ・ソースのニックネームの登録』

連合システムへの XML の追加

手順:

XML データ・ソースを連合サーバーに追加するには、以下のようにします。

1. CREATE WRAPPER ステートメントを使用してラッパーを登録します。
2. オプション：照会のパフォーマンスを向上させるために DB2_DJ_COMM 環境変数を設定します。
3. CREATE SERVER ステートメントを使用してサーバーを登録します。
4. CREATE NICKNAME ステートメントを使用してニックネームを登録します。
5. 非ルート・ニックネームのビューを作成します。

ステートメントは、DB2 コマンド行プロセッサから実行できます。XML ラッパーが連合システムに追加された後、XML データ・ソースに対して照会を実行できます。

関連作業:

- 123 ページの 『XML ラッパーの登録』
- 124 ページの 『XML ラッパーの DB2_DJ_COMM 環境変数の設定』
- 125 ページの 『XML データ・ソースのサーバーの登録』
- 126 ページの 『XML データ・ソースのニックネームの登録』
- 130 ページの 『非ルート・ニックネームの連合ビューの作成 (XML ラッパー)』
- 16 ページの 『連合システムへの表構造ファイルの追加』
- 33 ページの 『連合システムへの Documentum の追加』
- 75 ページの 『連合システムへの Excel の追加』
- 96 ページの 『連合システムへの BLAST の追加』

XML ラッパーの登録

このタスクは、連合システムへの XML の追加 のためのメインタスクの一部です。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、連合サーバーがデータ・ソースとの通信やデータ・ソースからのデータの検索に使用するメカニズムです。ラッパーは、ライブラリー・ファイルとして、システムにインストールされています。

手順:

XML ラッパーを登録するには、CREATE WRAPPER ステートメントをサブミットします。

たとえば、デフォルト・ライブラリー・ファイル libdb21sxml.a から my_xml という AIX 上の XML ラッパーを作成するには、次のステートメントをサブミットします。

```
CREATE WRAPPER my_xml LIBRARY 'libdb21sxml.a'  
OPTIONS(DB2_FENCED 'N');
```

サポートされているプラットフォーム別の XML ラッパーのデフォルト・ライブラリー・ファイル名の表については、下記の関連タスク・セクションの『DB2 Life Science Data Connect をインストールした後』を参照してください。CREATE WRAPPER ステートメントに関して詳しくは、「DB2 SQL リファレンス」を参照してください。

ここでのタスクの順序では、次のタスクは『XML ラッパーのための DB2_DJ_COMM 環境変数の設定』です。

関連作業:

- 16 ページの 『表構造ファイル・ラッパーの登録』
- 36 ページの 『Documentum ラッパーの登録』
- 75 ページの 『Excel ラッパーの登録』
- 101 ページの 『BLAST ラッパーの登録』
- 124 ページの 『XML ラッパーの DB2_DJ_COMM 環境変数の設定』

XML ラッパーの DB2_DJ_COMM 環境変数の設定

このタスクは、連合システムへの XML の追加 のためのメインタスクの一部です。

XML 文書にアクセスする場合のパフォーマンスを向上させるために、DB2_DJ_COMM 環境変数を設定します。この変数は、初期設定時に連合サーバーがラッパーをロードするかどうかを決定します。

手順:

DB2_DJ_COMM 環境変数を設定するには、関連する CREATE WRAPPER ステートメントに指定したラッパーに対応するラッパー・ライブラリーを指定して db2set コマンドをサブミットします。

```
db2set DB2_DJ_COMM='libdb21sxml.a'
```

等号 (=) の両側にスペースを入れないようにしてください。

データベース始動時にラッパー・ライブラリーをロードすることに関連したオーバーヘッドがあります。このオーバーヘッドを回避するには、アクセスしたいライブラリーのみを指定します。

DB2_DJ_COMM 環境変数に関して詳しくは、「DB2 管理ガイド」を参照してください。

ここでのタスクの順序では、次のタスクは『XML データ・ソースのサーバーの登録』です。

関連作業:

- 17 ページの 『表構造ファイル・ラッパーの DB2_DJ_COMM 環境変数の設定』
- 37 ページの 『Documentum ラッパーの DB2_DJ_COMM 環境変数の設定』
- 102 ページの 『BLAST ラッパーの DB2_DJ_COMM 環境変数の設定』
- 123 ページの 『XML ラッパーの登録』
- 125 ページの 『XML データ・ソースのサーバーの登録』

XML データ・ソースのサーバーの登録

このタスクは、連合システムへの XML の追加 のためのメインタスクの一部です。ラッパーが登録された後、対応するサーバーを登録する必要があります。

手順:

XML サーバーを連合システムに登録するには、CREATE SERVER ステートメントを使用します。

```
CREATE SERVER xml_server WRAPPER my_xml
```

ここでは、それぞれ以下のことを意味しています。

WRAPPER

関連する CREATE WRAPPER ステートメントを使用して登録したラッパー名を指定します。この引き数は必須です。

注: XML ラッパーは TYPE および VERSION キーワードを使用しません。CREATE SERVER ステートメントにこれらのキーワードを使用するとエラーが出されます。

ここでのタスクの順序では、次のタスクは『XML データ・ソースのニックネームの登録』です。

関連作業:

- 18 ページの 『表構造ファイルのサーバーの登録』
- 38 ページの 『Documentum データ・ソースのサーバーの登録』

- 76 ページの 『Excel データ・ソースのサーバーの登録』
- 103 ページの 『BLAST データ・ソースのサーバーの登録』
- 124 ページの 『XML ラッパーの DB2_DJ_COMM 環境変数の設定』
- 126 ページの 『XML データ・ソースのニックネームの登録』

XML データ・ソースのニックネームの登録

このタスクは、連合システムへの XML の追加 のためのメインタスクの一部です。XML データ・ソースのツリー構造をモデルとするニックネームを作成する必要があります。ツリーの親またはルートをモデル化するために、親ニックネームを作成する必要があります。親ニックネームに対応するエレメントにネストされているエレメントをモデル化するために、子ニックネームを作成する必要があります。

親および子ニックネームは、CREATE NICKNAME ステートメントに指定されている主および外部キーによって接続されます。

各ニックネームは、以下のことを行う 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。

ニックネームは、次の 2 通りの方法のいずれかの方法で、XML 文書に関連付けられません。

- 固定された方法で (FILE_PATH ニックネーム・オプションを使用して)。このオプションが使用される場合、ニックネームは特定の XML 文書からのデータを表します。
- 照会時に指定されたファイル名を使用して (DOCUMENT ニックネーム列オプションを使用して)。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 XML 文書からのデータを表すために使用できます。

これらのオプションの詳細は、以下の手順のセクションで説明されています。

手順:

XML データ・ソースをリレーショナル表にマップするには、CREATE NICKNAME ステートメントを使用してニックネームを作成する必要があります。

```

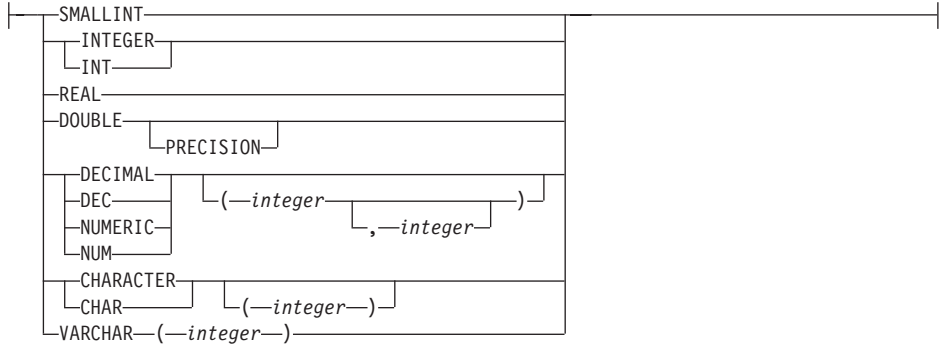
▶—CREATE NICKNAME—nickname—(—column-name—| column-information |—)—————▶
▶—FOR SERVER—server-name—OPTIONS—(—
    |FILE_PATH—'path' —,|
—————▶
▶—XPath—'xpath_expression' —)—————▶

```

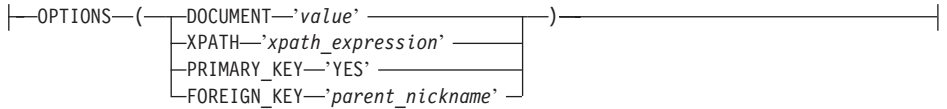
column-information:

| data-type | column-option | nickname-column-options |

data-type:



nickname-column-options:



column-option:



ニックネーム・オプション

FILE_PATH

XML 文書のファイル・パスを指定します。このニックネーム・オプションが指定されている場合、DOCUMENT ニックネーム列オプションは指定してなりません。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) に対してのみ受け入れられます。

XPATH 個々のタプルを表す XML エレメントを識別する 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の中で評価されます。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。

ニックネーム列オプション

DOCUMENT

XML データの種類を指定します。現在、XML ラッパーは FILE のみをサポートしています。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) に対してのみ受け入れられます。

DOCUMENT オプションでは、ニックネーム当たり 1 つの列しか指定できません。DOCUMENT オプションに関連付けられる列は、データ・タイプ VARCHAR 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事は、照会の実行中にこのニックネームに対応する文書が提供されることを意味します。DOCUMENT オプションが「FILE」値をもっている場合、照会の実行中に提供されるものは、文書を含んでいるファイルの名前であることを意味します。以下の CREATE NICKNAME の例は、DOCUMENT 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)   OPTIONS(PRIMARY_KEY 'YES')
FOR SERVER xml_server
OPTIONS(XPATH '/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ますようになります。

```
SELECT * FROM customers WHERE doc='/home/db2user/Customers.xml';
```

XPATH この列に対応するデータを検出できる XML 文書の中に XPath 式を指定します。この XPath 式は、XPath ニックネーム・オプションに指定されている XPath 式を評価した後に適用されます。

PRIMARY_KEY

これは親ニックネームであることを示します。列データ・タイプは常に VARCHAR(16) でなければなりません。ニックネームは、多くても 1 つの PRIMARY_KEY 列オプションしかもてません。'YES' が唯一の正しい値です。このオプションによって指定された列は、ラッパーが生成したキーを保持します。列の値は、SELECT リストでは検索できません。XPATH オプションをこの列に対して指定してはなりません。列は、親と子のニックネームを結合するためにだけ使用できます。

FOREIGN_KEY

これは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。ニックネームは、多くても 1 つの FOREIGN_KEY 列オプションしかもてません。このオプションの値は、大文字小文字の区別をします。このオプションによって指定された列は、ラッパーが生成したキーを保持します。列の値は、SELECT リストでは検索できません。XPATH オプションをこの列に対して指定してはなりません。列は、親と子のニックネームを結合するためにだけ使用できます。

ニックネームの例

以下の例は 図 12 のサンプル文書に示されているサンプル XML ファイルを使用して、XML データ・ソースのニックネームを作成する手順を示しています。

```
<doc>
  <customer id='123'>
    <name>...</name>
    <address>...</address>
    ...
    <order>
      <amount>...</amount>
      <date>...</date>
      <item quant='12'>
        <name>...</name>
      </item>
      <item quant='4'>...</item>
      ...
    </order>
  </customer>
  <customer id='124'>...</customer>
</doc>
```

図 12. サンプル XML ファイル

親ニックネーム customers を作成するには、以下のステートメントを指定します。

```
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)  OPTIONS(PRIMARY_KEY 'YES')
  FOR SERVER xml_server
  OPTIONS( XPATH './customer',
          FILE_PATH '/home/db2user/Customers.xml');
```

customer の子 - orders、payments、および items - のニックネームを作成するには、以下の 3 つのニックネーム・ステートメントを指定します。

orders の場合

```
CREATE NICKNAME orders
(
  amount  INTEGER          OPTIONS(XPATH './amount'),
  date    VARCHAR(10)     OPTIONS(XPATH './date'),
```

```

oid      VARCHAR(16)  OPTIONS(PRIMARY_KEY 'YES'),
cid      VARCHAR(16)  OPTIONS(FOREIGN_KEY 'CUSTOMERS'))
FOR SERVER xml_server
OPTIONS( XPATH './order');

```

payments の場合

```

CREATE NICKNAME payments
(
  number  INTEGER      OPTIONS(XPATH './number'),
  date    VARCHAR(10)  OPTIONS(XPATH './date'),
  cid     VARCHAR(16)  OPTIONS(FOREIGN_KEY 'CUSTOMERS'))
FOR SERVER xml_server
OPTIONS( XPATH './payment');

```

items の場合

```

CREATE NICKNAME items
(
  name     VARCHAR(20)  OPTIONS(XPATH './name'),
  quantity INTEGER      OPTIONS(XPATH './@quant'),
  oid      VARCHAR(16)  OPTIONS(FOREIGN_KEY 'ORDERS'))
FOR SERVER xml_server
OPTIONS( XPATH './item');

```

ここでのタスクの順序では、次のタスクは『非ルート・ニックネームの連合ビューの作成 (XML ラッパー)』です。

関連作業:

- 19 ページの 『表構造ファイルのニックネームの登録』
- 41 ページの 『Documentum データ・ソースのニックネームの登録』
- 77 ページの 『Excel データ・ソースのニックネームの登録』
- 104 ページの 『BLAST データ・ソースのニックネームの登録』
- 125 ページの 『XML データ・ソースのサーバーの登録』
- 143 ページの 『第 8 章 ニックネームのコスト計算オプションの指定』

非ルート・ニックネームの連合ビューの作成 (XML ラッパー)

このタスクは、連合システムへの XML の追加 のためのメインタスクの一部です。XML 文書を説明するニックネームの階層に対する連合ビューを定義することをお勧めします。連合ビューを定義することにより、ルートを含まない XML ニックネーム階層の部分と結合する照会、ならびに特別な PRIMARY_KEY および FOREIGN_KEY 列以外の列で結合する照会が、正しく実行されることが保証されます。

手順:

必要なすべての述部およびルートへの絶対パスを含む連合ビューを定義するには、以下のステップを実行します。

1. 各非ルート・ニックネームのビューをルートへのパス上のすべてのニックネームの結合として定義します。
2. WHERE 文節において、PRIMARY_KEY および FOREIGN_KEY 列に対する結合述部を作成します。
3. 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ものは除きます。
4. 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。

ビューの例:

以下の例は、ビューの使用を示しています。この例では 図 13 に示されているサンプル・ファイルのニックネームは、前に customers、orders、payments および items として作成されていたものと想定しています。

```
<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 13. サンプル XML ファイル

非ルート・ニックネーム order、payment および item のビューは、以下のとおりです。

order の場合

```
CREATE FEDERATED VIEW order_view AS
SELECT o.amount, o.date, o.oid, c.cid
FROM customers c, orders o
WHERE c.cid = o.cid;
```

payment の場合

```
CREATE FEDERATED VIEW payment_view AS
SELECT p.amount, p.date, c.cid
FROM customers c, payments p
WHERE c.cid = p.cid;
```

item の場合

```
CREATE FEDERATED VIEW item_view AS
SELECT it.quantity, it.name, o.oid
FROM customers c, orders o, items i
WHERE c.cid = o.cid AND o.oid = i.oid;
```

これらのビューにサブミットされた照会は、ルートへの結合パスが存在するので、正しく処理されます。

たとえば、以下の照会は、カスタマーのオーダーの金額と、同じ日付からの支払いをペアにします。

```
SELECT o.amount, p.amount
FROM order_view o, payment_view p
WHERE p.date = o.date AND
      p.cid = o.cid;
```

ここでの作業は、これで終わりです。

関連作業:

- 126 ページの 『XML データ・ソースのニックネームの登録』

XML データ・ソースに対する照会の実行

このセクションでは、CREATE NICKNAME ステートメントによって作成されたニックネーム customers、orders、および items を使用するいくつかのサンプル照会をリストしています。

手順:

照会を実行するために、ガイドとして以下の例を使用します。

以下の照会は、すべてのカスタマー名を表示します。

```
SELECT name FROM customers;
```

以下の照会は、カスタマー名が 'Smith' のすべてのレコードを表示します。

```
SELECT * FROM customers where name='Smith';
```

以下の照会は、各カスタマーの各オーダーごとの金額と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;
```

オーダーとカスタマー・ニックネームの間の親子関係を示すためには、結合 `c.cid=o.cid` が必要です。

以下の照会は、オーダーごとおよび各カスタマーの品目ごとの、オーダー金額、品目名およびカスタマーの住所を選択します。

```
SELECT c.address, o.amount, i.name FROM customers c, orders o, items i
WHERE c.cid=o.cid AND o.oid=i.oid;
```

親子関係を維持するために、ここでも 2 つの結合が必要です。

以下の 2 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。ニックネーム `customers` を作成するために使用される、対応する `CREATE NICKNAME` ステートメントが、以下に示されています。

```
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)   OPTIONS(PRIMARY_KEY 'YES'))
FOR SERVER xml_server
OPTIONS(XPATH '/customer');
```

以下の照会は、`/home/db2user/Customers.xml` のファイル・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、`/home/db2user/Customers.xml` にある XML ファイル `Customers.xml` 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。

```
SELECT c.name, o.date FROM customers c, orders o
WHERE c.doc='/home/db2user/Customers.xml' AND o.amount > 1000;
```

関連作業:

- 57 ページの 『Documentum データ・ソースに対する照会の実行』
- 79 ページの 『Excel データ・ソースに対する照会の実行』

XML ラッパーの制限と考慮事項

このセクションには、XML ラッパーの使用に関連した制限と考慮事項のリストが含まれています。

- パススルー機能はサポートされていません。
- XML 文書は読み取り専用です。

関連資料:

- 23 ページの『表構造ファイル・ラッパーに関するラッパーの制限と考慮事項』
- 24 ページの『表構造ファイル・ラッパーに関するファイルの制限と考慮事項』
- 63 ページの『Documentum ラッパーの制限と考慮事項』
- 81 ページの『Excel ラッパーのラッパーの制限』
- 82 ページの『Excel ファイルの制限』

XML ラッパーのメッセージ

このセクションでは、XML のラッパーを処理するときに出されるメッセージをリストして説明します。メッセージの詳細については、「DB2 メッセージ・リファレンス」を参照してください。

表 26. XML のラッパーによって出されるメッセージ

エラー・コード	メッセージ	説明
SQL0405N	数値リテラル "<column_name>" は、値が有効な値の範囲外になっているため、無効です。	指定された数値リテラルが許容範囲がありません。 CREATE NICKNAME ステートメントの中の列のデータ・タイプをチェックしてください。
SQL0408N	値が、その割り当てターゲットのデータ・タイプと互換性がありません。ターゲット名は "<column_name>" です。	列に割り当てられる値のデータ・タイプが、割り当てターゲットの宣言されたデータ・タイプと互換性がありません。 CREATE NICKNAME ステートメントの中の列のデータ・タイプをチェックしてください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由 "Error creating wrapper object".)	新しいラッパー・オブジェクトを作成しているときにエラーが発生しました。 DB2 サポートに連絡してください。

表 26. XML のラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由 "Xerces initialization error".)	Xerces 構文解析プログラムの初期化の際に例外が発生しました。DB2 サポートに連絡してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由 "<xalan_error_message>".)	Xalan 関数を呼び出すときにエラーが発生しました。XML 文書をチェックしてください。文書が適切に構造化されている場合、エラー・メッセージの詳細については Xalan 資料を参照してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由 "XalanDOMException: exception code is <exception_code>".)	XalanDOMException 例外が発生しました。Xalan 資料を参照して、例外コードの意味を調べてください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由 "Error getting node value".)	Xalan が無効なノードにアクセスしようとしました。DB2 サポートに連絡してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由 "Error parsing XML document".)	XML 文書を構文解析しているときにエラーが発生しました。XML 文書をチェックしてください。

表 26. XML のラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由 "Error getting root element of XML document".)	XML 文書を構文解析した後に、Xalan がルート・エレメントを検索しようとしたが失敗しました。XML 文書をチェックしてください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由 "Unspecified exception while evaluating XPath expression".)	XPath 式を評価しているときに、Xalan によって未指定の例外が生成されました。XML 文書をチェックし、Xalan 資料を参照してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由 "Unspecified exception while getting node value".)	ノード値を検索しているときに、Xalan によって未指定の例外が生成されました。XML 文書をチェックし、Xalan 資料を参照してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由 "Unspecified exception while building DOM tree".)	XML 文書の DOM ツリーを構築しているときに、Xalan によって未指定の例外が生成されました。XML 文書をチェックし、Xalan 資料を参照してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由 "Memory allocation error".)	メモリー割り振りの際に、エラーが発生しました。

表 26. XML のラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL1822N	データ・ソース「XML ラッパー」から予期しないエラー・コード「<trace_point>」を受け取りました。関連するテキストおよびトークンは "Column data type not supported" です。	ニックネーム列がサポートされていないデータ・タイプをもっています。 CREATE NICKNAME ステートメントをチェックしてください。
SQL1822N	データ・ソース「XML ラッパー」から予期しないエラー・コード「<trace_point>」を受け取りました。関連するテキストおよびトークンは "TYPE clause not supported" です。	CREATE SERVER ステートメントに TYPE 文節が含まれています。この文節は XML ラッパーによってサポートされていません。これを除去してください。
SQL1822N	データ・ソース「XML ラッパー」から予期しないエラー・コード「<trace_point>」を受け取りました。関連するテキストおよびトークンは "VERSION clause not supported" です。	CREATE SERVER ステートメントに VERSION 文節が含まれています。この文節は XML ラッパーによってサポートされていません。これを除去してください。
SQL1822N	データ・ソース「XML ラッパー」から予期しないエラー・コード「<trace_point>」を受け取りました。関連するテキストおよびトークンは "Invalid use of predicate with DOCUMENT column" です。	照会に、誤ったオペランドのある述部が含まれています。照会の中の述部をチェックしてください。
SQL1822N	データ・ソース「XML ラッパー」から予期しないエラー・コード「<trace_point>」を受け取りました。関連するテキストおよびトークンは "Invalid use of predicate with FOREIGN_KEY column" です。	照会に、誤ったオペランドのある述部が含まれています。照会の中の述部をチェックしてください。

表 26. XML のラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL1822N	データ・ソース「XML ラッパー」から予期しないエラー・コード「<trace_point>」を受け取りました。関連するテキストおよびトークンは "Invalid use of predicate with PRIMARY_KEY column" です。	照会に、誤ったオペランドのある述部が含まれています。照会の中の述部をチェックしてください。
SQL1822N	データ・ソース「XML ラッパー」から予期しないエラー・コード「<trace_point>」を受け取りました。関連するテキストおよびトークンは "XPATH and DOCUMENT options not compatible" です。	CREATE NICKNAME ステートメントの指定が正しくありません。その構文をチェックしてください。
SQL1822N	データ・ソース「XML ラッパー」から予期しないエラー・コード「<trace_point>」を受け取りました。関連するテキストおよびトークンは "XPATH and FOREIGN_KEY options not compatible" です。	CREATE NICKNAME ステートメントの指定が正しくありません。その構文をチェックしてください。
SQL1822N	データ・ソース「XML ラッパー」から予期しないエラー・コード「<trace_point>」を受け取りました。関連するテキストおよびトークンは "XPATH and PRIMARY_KEY options not compatible" です。	CREATE NICKNAME ステートメントの指定が正しくありません。その構文をチェックしてください。
SQL1822N	データ・ソース「XML ラッパー」から予期しないエラー・コード「<trace_point>」を受け取りました。関連するテキストおよびトークンは "DOCUMENT and FOREIGN_KEY options not compatible" です。	CREATE NICKNAME ステートメントの指定が正しくありません。その構文をチェックしてください。

表 26. XML のラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL1822N	データ・ソース「XML ラッパー」から予期しないエラー・コード「<trace_point>」を受け取りました。関連するテキストおよびトークンは "DOCUMENT and PRIMARY_KEY options not compatible" です。	CREATE NICKNAME ステートメントの指定が正しくありません。その構文をチェックしてください。
SQL1822N	データ・ソース「XML ラッパー」から予期しないエラー・コード「<trace_point>」を受け取りました。関連するテキストおよびトークンは "FOREIGN_KEY and PRIMARY_KEY options not compatible" です。	CREATE NICKNAME ステートメントの指定が正しくありません。その構文をチェックしてください。
SQL1822N	データ・ソース「XML ラッパー」から予期しないエラー・コード「<trace_point>」を受け取りました。関連するテキストおよびトークンは "Column option missing" です。	CREATE NICKNAME ステートメントの指定が正しくありません。その構文をチェックしてください。
SQL1822N	データ・ソース「XML ラッパー」から予期しないエラー・コード「<trace_point>」を受け取りました。関連するテキストおよびトークンは "DOCUMENT column option not unique" です。	CREATE NICKNAME ステートメントの指定が正しくありません。その構文をチェックしてください。
SQL1822N	データ・ソース「XML ラッパー」から予期しないエラー・コード「<trace_point>」を受け取りました。関連するテキストおよびトークンは "FOREIGN_KEY column option not unique" です。	CREATE NICKNAME ステートメントの指定が正しくありません。その構文をチェックしてください。

表 26. XML のラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL1822N	データ・ソース「XML ラッパー」から予期しないエラー・コード「<trace_point>」を受け取りました。関連するテキストおよびトークンは "PRIMARY_KEY column option not unique" です。	CREATE NICKNAME ステートメントの指定が正しくありません。その構文をチェックしてください。
SQL1822N	データ・ソース「XML ラッパー」から予期しないエラー・コード「<trace_point>」を受け取りました。関連するテキストおよびトークンは "Invalid DOCUMENT option value" です。	CREATE NICKNAME ステートメントに指定されている DOCUMENT オプションの値が無効です。その値になり得るのは「FILE」だけです。CREATE NICKNAME ステートメントをチェックしてください。
SQL1822N	データ・ソース「XML ラッパー」から予期しないエラー・コード「<trace_point>」を受け取りました。関連するテキストおよびトークンは "Invalid PRIMARY_KEY option value" です。	CREATE NICKNAME ステートメントに指定されている PRIMARY_KEY オプションの値が無効です。その値になり得るのは「YES」だけです。CREATE NICKNAME ステートメントをチェックしてください。
SQL1822N	データ・ソース「XML ラッパー」から予期しないエラー・コード「<trace_point>」を受け取りました。関連するテキストおよびトークンは "Invalid FOREIGN_KEY option value" です。	CREATE NICKNAME ステートメントに指定されている FOREIGN_KEY オプションの値が無効です。そのオプション値に一致する親ニックネームは検出されませんでした。CREATE NICKNAME ステートメントをチェックしてください。
SQL1822N	データ・ソース「XML ラッパー」から予期しないエラー・コード「<trace_point>」を受け取りました。関連するテキストおよびトークンは "FILE_PATH and DOCUMENT options not compatible" です。	CREATE NICKNAME ステートメントは、指定されているように、正しくありません。FILE_PATH および DOCUMENT オプションは同時に指定することはできません。CREATE NICKNAME 構文をチェックしてください。

表 26. XML のラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL1881N	"<option_name>" は "<object_name>" に対して有効な "<option_type>" ではありません。	指定されたオプションは存在しないか、この特定のデータ・ソースに対しては無効である可能性があります。CREATE NICKNAME ステートメントをチェックしてください。
SQL1883N	"<option_name>" は "<object_name>" に対して必須 "<option_type>" オプションで	必須の DB2 オプションが指定されていません。CREATE NICKNAME ステートメントをチェックしてください。

関連資料:

- 25 ページの 『表構造ファイル・ラッパーのメッセージ』
- 64 ページの 『Documentum ラッパーのメッセージ』
- 83 ページの 『Excel ラッパーのメッセージ』
- 115 ページの 『BLAST ラッパーのメッセージ』

第 8 章 ニックネームのコスト計算オプションの指定

効率的な実行計画を作成するために、最適化プログラムはいくつかの異なる計画を生成し、それぞれについて必要なリソースを見積もります。最も少ないリソースを必要とする計画が、評価の対象として使用されます。

外部データ・ソースによって実行される照会部分に対する評価時間の見積もりは、ラッパーによって提供されます。この計算で使用される数式は、3 つの基本パラメーターを使用します。これらは、特定のインストールにあわせて変更することができます。

これらのパラメーターは、以下のニックネーム・オプションとして指定されます。

RESET_COST

外部サーバーと接触し、結果を戻してもらうまでの時間を、ミリ秒で指定します。

ADVANCE_COST

各行を入手する時間を、ミリ秒で指定します。

BIND_COST

パラメーターをラッパーから外部ソースへ渡すのに必要な時間を、ミリ秒で指定します。

すべての値は、整数で与えられなければなりません。デフォルト値については、「*DB2 Life Sciences Data Connect* バージョン 8 リリース情報」を参照してください。

手順:

ニックネームのコスト計算オプションを指定するには、次のように行います。

1. インストールを分析して、コスト計算オプションのカスタマイズが、御社における連合照会の処理に利益をもたらすかどうかを判別します。
2. 利益をもたらすのであれば、1 つまたは複数のコスト計算オプションを、ニックネーム・オプションとして、ラッパーの `CREATE NICKNAME` ステートメントに追加します。
3. `CREATE NICKNAME` ステートメントをサブミットします。

`CREATE NICKNAME` ステートメントについての詳細は、「*DB2 SQL* リファレンス」を参照してください。

関連作業:

- 19 ページの 『表構造ファイルのニックネームの登録』
- 41 ページの 『Documentum データ・ソースのニックネームの登録』

- 77 ページの 『Excel データ・ソースのニックネームの登録』
- 104 ページの 『BLAST データ・ソースのニックネームの登録』
- 145 ページの 『ニックネームの変更』
- 126 ページの 『XML データ・ソースのニックネームの登録』

第 9 章 ニックネームの変更

この章では、既に登録済みのニックネームを変更するための ALTER NICKNAME ステートメントの使用法を説明します。

ニックネームの変更

ALTER NICKNAME ステートメントを使用して、データ・ソースまたはビューの連合データベースの表現を変更することができます。

制約事項:

ALTER NICKNAME ステートメントは、DB2 Life Science Data Connect ラッパーのいずれの列名を変更するためにも使用することはできません。

手順:

ニックネーム列値を変更するには、ALTER NICKNAME ステートメントを使用して、以下のことを行う必要があります。

- これらの列のローカル・データ・タイプの変更
- これらの列のオプションの追加、変更、または削除

ALTER NICKNAME ステートメントに関して詳しくは、「*DB2 SQL* リファレンス」を参照してください。

関連作業:

- 145 ページの 『データ・タイプの変更』
- 146 ページの 『ニックネーム・オプションの変更』

データ・タイプの変更

ALTER NICKNAME ステートメントを使用して、列のデータ・タイプを変更することができます。

手順:

列のデータ・タイプを変更するには、ALTER NICKNAME ステートメントを使用します。

たとえば、以下の ALTER NICKNAME ステートメントは、DRUG 列のローカル・データ・タイプを CHAR(30) に変更します。DRUG 列は最初は、CREATE NICKNAME ステートメントを使用して CHAR(20) として定義されていました。ニックネーム DRUGDATA1 は、drugdata1.txt というローカルの表構造ファイルです。

```
ALTER NICKNAME DRUGDATA1
  ALTER COLUMN DRUG
  LOCAL TYPE CHAR(30)
```

関連作業:

- 145 ページの 『ニックネームの変更』
- 146 ページの 『ニックネーム・オプションの変更』

ニックネーム・オプションの変更

ALTER NICKNAME ステートメントを使用して、ニックネーム・オプションを変更することができます。

手順:

ニックネーム・オプションを変更するには、ALTER NICKNAME ステートメントを使用します。

たとえば、以下の ALTER NICKNAME ステートメントは、表構造ファイル drugdata1.txt の完全修飾パスを変更します。パスは最初は、CREATE NICKNAME ステートメントを使用して '/user/pat/drugdata1.txt' として定義されていました。ニックネーム DRUGDATA1 は、drugdata1.txt というローカルの表構造ファイルです。

```
ALTER NICKNAME DRUGDATA1
  OPTIONS (SET FILE_PATH '/usr/kelly/data/drugdata1.txt')
```

関連作業:

- 145 ページの 『ニックネームの変更』
- 145 ページの 『データ・タイプの変更』

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、プログラムまたはサービスの操作性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。実施権の許諾については、下記の宛先に書面にてご照会ください。

〒106-0032 東京都港区六本木 3-2-31

IBM World Trade Asia Corporation

Licensing

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。

本書に含まれる情報には、技術的に不正確なもの、または誤植が含まれる場合があります。これらに対する変更は、定期的に行われます。これらの変更は、資料の改訂版に含まれます。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 Canada Limited
Office of the Lab Director
8200 Warden Avenue
Markham, Ontario
L6G 1C7
CANADA

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が、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。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。したがって 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のめかしたり、保証することはできません。

それぞれの複製物、サンプル・プログラムのすべての部分、またはすべての派生した創作物には、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. _西暦年_. All rights reserved.

商標

以下は、IBM Corporation の商標です。

ACF/VTAM	LAN Distance
AISPO	MVS
AIX	MVS/ESA
AIXwindows	MVS/XA
AnyNet	Net.Data
APPN	NetView
AS/400	OS/390
BookManager	OS/400
C Set++	PowerPC
C/370	pSeries
CICS	QBIC
Database 2	QMF
DataHub	RACF
DataJoiner	RISC System/6000
DataPropagator	RS/6000
DataRefresher	S/370
DB2	SP
DB2 Connect	SQL/400
DB2 Extenders	SQL/DS
DB2 OLAP Server	System/370
DB2 Universal Database	System/390
Distributed Relational Database Architecture	SystemView
DRDA	Tivoli
eServer	VisualAge
Extended Services	VM/ESA
FFST	VSE/ESA
First Failure Support Technology	VTAM
IBM	WebExplorer
IMS	WebSphere
IMS/ESA	WIN-OS/2
iSeries	z/OS
	zSeries

以下は、他社の商標または登録商標です。

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。

"Pentium" "MMX" "ProShare" "LANDesk" および "Action Media" は Intel Corporation の米国およびその他の国における商標。
(Intel の商標は <http://www.intel.com/sites/corporate/tradmarx.htm> で参照できます)。

Java およびすべての Java 関連の商標およびロゴは、Sun Microsystems, Inc. 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。

UNIX は、The Open Group がライセンスしている米国およびその他の国における登録商標です。

他の会社名、製品名およびサービス名などはそれぞれ各社の商標または登録商標です。

参考文献

当参考文献には、DB2 Life Sciences Data Connect ご使用の際に役に立つと思われる DB2 Universal Database に関する文献が掲載されています。

- DB2 コネクト 使用者の手引き (SC88-8521)
- DB2 ユニバーサル・データベース (UNIX 版) 概説およびインストール (GC88-8536)
- DB2 SQL 解説書 (SC88-8540)
- DB2 管理の手引き: 計画 (SC88-8513)
- DB2 管理の手引き: インプリメンテーション (SC88-8511)
- DB2 管理の手引き: パフォーマンス (SC88-8512)
- DB2 メッセージ解説書 第 1 巻 (GC88-8543)
- IBM DB2 Universal Database 連合システム・ガイド (GC88-9170)
- DB2 Life Sciences Data Connect リリース情報 バージョン 8

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。

[サ行]

最適化

表構造ファイル 25

BLAST 115

サンプルの照会 57

BLAST

組み立て 104, 112

サンプル 113

Documentum 57

Excel 79

XML 132

[ナ行]

ニックネーム

オプションの原価計算の指定
143

データ・タイプの変更 145

ニックネーム・オプションの変更
146

変更 145

ニックネーム・オプション 143

[ハ行]

表構造ファイル

概要 13

最適化 25

制限と考慮事項 23, 24

タイプ 13

ファイル・アクセス制御モデル
24

メッセージ 25

例 13

表構造ファイル (続き)

連合システムへの追加

サーバーの登録 18

ニックネームの登録 19

ラッパーの登録 16

DB2 Life Science Data Connect

を使って 14

[マ行]

メッセージ

表構造ファイル・ラッパー 25

BLAST ラッパー 115

Documentum ラッパー 64

Excel ラッパー 83

XML ラッパー 115

[ラ行]

ライフ・サイエンス・データ・ソース 1

ラッパー

定義 1

プラットフォームごとのデフォルト・ライブラリー名 10

ライフ・サイエンス、プラットフォームごと 5

B

BLAST

説明 91

メッセージ 115

連合システムへの追加

サーバーの登録 103

正しい blastall executable がインストールされていることを
検査 97

正しいマトリックス・ファイルがインストールされている
ことを検査 97

BLAST (続き)

連合システムへの追加 (続き)

ニックネームの登録 104

ラッパーの登録 101

BLAST 構成ファイル 97

BLAST デーモンの開始 100

BLAST デーモンの設定と構成
97

CREATE NICKNAME ステートメント 104

CREATE SERVER ステートメント 103

CREATE WRAPPER ステートメント 101

C

CREATE FEDERATED VIEW ステートメント

XML 130

CREATE FUNCTION ステートメント

Documentum 51

CREATE NICKNAME ステートメント

表構造ファイル 19

BLAST 104

Documentum 41

Excel ファイル 77

XML 126

CREATE SERVER ステートメント

表構造ファイル 18

BLAST 103

Documentum 38

Excel ファイル 76

XML 125

CREATE USER MAPPING ステートメント

Documentum 40

CREATE WRAPPER ステートメント
表構造ファイル 16

CREATE WRAPPER ステートメント
(続き)
BLAST 101
Documentum 36
Excel ファイル 75
XML 123
CreateNicknameFile ユーティリティ
一、Documentum 58
インストール 59
構成 60
DM_ID オブジェクト・タイプの
マッピング 61

D

DB2_DJ_COMM 環境変数 17, 37,
102, 124
DiscoveryLink 2
Documentum
繰り返し属性の二重の定義 62
資料へのユーザー・アクセス 64
制限と考慮事項 63
説明 31
メッセージ 64
例 31
連合システムへの追加
カスタム関数の登録 51
サーバーの登録 38
ニックネームの登録 41
ユーザーのマッピング 40
ラッパーの登録 36
CREATE FUNCTION ステ
ートメント 51
CREATE NICKNAME ステ
ートメント 41
CREATE SERVER ステ
ートメント 38
CREATE USER MAPPING ス
テートメント 40
CREATE WRAPPER ステ
ートメント 36
CreateNicknameFile ユーティリ
ティ一 58

Documentum (続き)
連合システムへの追加 (続き)
Documentum クライアント・
ライブラリーへのリンク
(AIX および Solaris オペレ
ーティング環境のみ) 34
Documentum のクライアント
dmcl.ini ファイルを指す 35
CreateNicknameFile ユーティリテ
ィ一 58

E

Excel ファイル
サンプル・ユーザーのシナリオ
79
制限と考慮事項 81, 82
説明 73
ファイル・アクセス制御モデル
82
メッセージ 83
例 73
連合システムへの追加
サーバーの登録 76
ニックネームの登録 77
ラッパーの登録 75
CREATE NICKNAME ステ
ートメント 77
CREATE SERVER ステ
ートメント 76

X

XML (eXtensible Markup Language)
制限と考慮事項 134
説明 119
メッセージ 134
連合システムへの追加 123
サーバーの登録 125
ニックネームの登録 126
非ルート・ニックネームの連
合ビューの作成 130
ラッパーの登録 123
CREATE FEDERATED VIEW
ステートメント 130

XML (eXtensible Markup Language)
(続き)
連合システムへの追加 (続き)
CREATE NICKNAME ステ
ートメント 126
CREATE SERVER ステ
ートメント 125
CREATE WRAPPER ステ
ートメント 123

IBM と連絡をとる

技術上の問題がある場合は、お客様サポートに連絡をとってください。

製品情報

以下の情報は英語で提供されます。内容は英語版製品に関する情報です。

DB2 Universal Database 製品に関する情報は、 www.ibm.com/software/data/db2/udb から入手できます。

このサイトには、技術ライブラリー、資料の注文方法、クライアント・ダウンロード、ニュースグループ、フィックスパック、ニュース、および Web リソースへのリンクに関する最新情報が掲載されています。

米国以外の国で IBM に連絡する方法については、 [IBM Worldwide](http://www.ibm.com/planetwide) ページ (www.ibm.com/planetwide) にアクセスしてください。



部品番号: CT16FJA

Printed in Japan

GC88-9173-00



日本アイ・ビー・エム株式会社
〒106-8711 東京都港区六本木3-2-12

(1P) P/N: CT16FJA



Spine information:



IBM® DB2® Life Sciences
Data Connect

DB2 LSDC 計画、インストールおよび構
成のガイド

バージョン 8