



ノンストップデータベース

HiRDB Version 10 XDM/RD E2 接続機能

解説・手引・文法書

3020-6-565-70

前書き

■ 対象製品

●適用 OS : AIX V7.2, AIX 7.3

P-1M62-35A1 HiRDB Server Version 10 10-09

P-1M62-1BA1 HiRDB/Run Time Version 10 10-09

P-1M62-1CA1 HiRDB/Developer's Kit Version 10 10-09

P-1M62-1DA1 HiRDB/Run Time Version 10(64) 10-09

P-1M62-1EA1 HiRDB/Developer's Kit Version 10(64) 10-09

●適用 OS : Red Hat Enterprise Linux 8 (64-bit x86_64), Red Hat Enterprise Linux 9 (64-bit x86_64)

P-8462-35A1 HiRDB Server Version 10 10-09

P-8462-1DA1 HiRDB/Run Time Version 10(64) 10-09

P-8462-1EA1 HiRDB/Developer's Kit Version 10(64) 10-09

●適用 OS : Red Hat Enterprise Linux 8 (64-bit x86_64), Red Hat Enterprise Linux 9 (64-bit x86_64)

P-8362-1BA1 HiRDB/Run Time Version 10 10-09

P-8362-1CA1 HiRDB/Developer's Kit Version 10 10-09

P-8362-3CA1 HiRDB Developer's Suite Version 10 10-09

●適用 OS : Windows Server 2019, Windows Server 2022, Windows 10 Pro (x64), Windows 10 Enterprise (x64), Windows 11

P-2962-91A4 HiRDB Server Version 10 10-09

●適用 OS : Windows Server 2019, Windows Server 2022, Windows 10, Windows 11

P-2662-11A4 HiRDB/Run Time Version 10 10-09

P-2662-12A4 HiRDB/Developer's Kit Version 10 10-09

P-2662-32A4 HiRDB Developer's Suite Version 10 10-09

●適用 OS : Windows Server 2019, Windows Server 2022, Windows 10 Home (x64), Windows 10 Pro (x64), Windows 10 Enterprise (x64), Windows 11

P-2962-11A4 HiRDB/Run Time Version 10(64) 10-09

P-2962-12A4 HiRDB/Developer's Kit Version 10(64) 10-09

これらのプログラムプロダクトのほかにもこのマニュアルをご利用になれる場合があります。詳細は「リリースノート」でご確認ください。

■ 輸出時の注意

本製品を輸出される場合には、外国為替及び外国貿易法の規制並びに米国輸出管理規則など外国の輸出関連法規をご確認の上、必要な手続きをお取りください。

なお、不明な場合は、弊社担当営業にお問い合わせください。

■ 商標類

記載の会社名、製品名などは、それぞれの会社の商標もしくは登録商標です。

■ 発行

2025年1月 3020-6-565-70

■ 著作権

All Rights Reserved. Copyright (C) 2018, 2025, Hitachi, Ltd.

変更内容

変更内容(3020-6-565-70) HiRDB Version 10 10-09

追加・変更内容	変更個所
ODBC ドライバで発生したエラーの情報を ODBC エラーログファイルへ出力する機能をサポートしました。 これによって、エラー発生時のトラブルシュートが容易になります。	表 4-1 , 4.4.2(10)
SQL トレース機能で、SQL トレース情報をメモリ上に退避しエラー発生時にファイル出力する方式をサポートしました。従来の方式と比べて、SQL トレース取得時の性能への影響を低減しています。 これによって、SQL トレースを取得しやすくなり、エラー発生時のトラブルシュートが容易になります。	表 4-1

単なる誤字・脱字などはお断りなく訂正しました。

変更内容(3020-6-565-60) HiRDB Version 10 10-08

追加・変更内容	変更個所
クライアントとサーバ間のネットワーク上で送受信するデータを暗号化する機能をサポートしました。 これによって、パケットスニッフィングなどで悪意のあるユーザが不正にデータを参照することを防止できます。	表 4-1

変更内容(3020-6-565-50) HiRDB Version 10 10-07

追加・変更内容
クライアント-サーバ間の接続で新たな接続方式をサポートしました。これによって、クライアント側ファイアウォールでのポート開放が不要となるほか、NAPT が設定されたネットワーク環境で接続できるようになります。
HiRDB の適用 OS に次の OS を追加しました。
• AIX V7.3 • Red Hat Enterprise Linux 9

変更内容(3020-6-565-40) HiRDB Version 10 10-06

追加・変更内容
HP-UX に関する説明を削除しました。

はじめに

このマニュアルは、次に示す製品の HiRDB の埋込み型 UAP から XDM/RD E2 のデータベースを操作する機能（XDM/RD E2 接続機能）の概要、XDM/RD E2 接続機能を使用する場合の環境設定や SQL の言語仕様などについて説明したものです。なお、ここに記載されていない前提情報については、マニュアル「HiRDB Version 10 解説」を参照してください。

- HiRDB Server Version 10
- HiRDB/Run Time Version 10
- HiRDB/Run Time Version 10(64)
- HiRDB/Developer's Kit Version 10
- HiRDB/Developer's Kit Version 10(64)
- HiRDB Developer's Suite Version 10

■ 対象読者

このマニュアルは、HiRDB クライアントから XDM/RD E2 のデータベースを操作する UAP の作成者、および UAP の実行者を対象としています。なお、UAP の作成者の方は、C 言語のプログラミングまたは COBOL 言語のプログラミングを理解していることを前提としています。また、UAP の実行者の方は、適用 OS の基礎的な知識を持っていることを前提としています。

■ 図中で使用する記号

このマニュアルの図中で使用する記号を次のように定義します。

●WS, PC



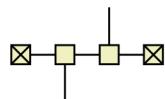
●入出力の動作



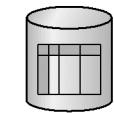
●プログラム



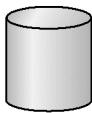
●LAN



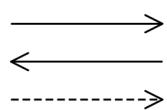
●データベース



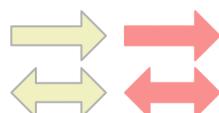
●ファイル



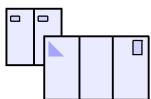
●制御の流れ



●データの流れ



●ホストセンタ



■ 文法で使用する記号

文法の説明に使用する各種の記号について説明します。ここで述べる文法記述記号、属性表示記号、および構文要素記号は、実際には記述しないでください。

文法記述記号

文法の説明で使用する記号です。

文法記述記号	意味
{ }	この記号で囲まれている複数の項目の中から一つを選択することを示します。 (例) PDODBESCAPE= {0 1} これは、PDODBESCAPE オペランドの指定値として 0 と 1 から選択できることを示します。
[]	この記号で囲まれた項目は省略できることを示します。 (例) PDHOST=DB コネクションサーバのホスト名 [, 予備系のホスト名] これは、予備系のホスト名を省略できることを示します。
(ストローク)	記号 { } で囲まれた複数の項目を一つずつの項目に区切ることを示します。 (例) PDPRMTRC= {YES NO} これは、PDPRMTRC オペランドの設定値として YES と NO の項目に区切ることを示します。
__(下線)	指定値を省略したときにシステムが設定する標準値を示します。

文法記述記号	意味
	(例) PDEXWARN= {YES NO} これは、PDEXWARN オペランドを省略したときに PDEXWARN=NO と仮定されることを示します。

属性表示記号

ユーザ指定値の範囲などの説明で使用する記号です。

属性表示記号	意味
~	この記号の後にユーザ指定値の属性を示します。
《 》	ユーザが指定しなかった場合に仮定される値を示します。
< >	ユーザ指定値の構文要素を示します。
(())	ユーザ指定値の指定範囲を示します。

構文要素記号

ユーザ指定値の内容の説明で使用する記号です。

構文要素記号	意味
<英字>	アルファベット (A~Z, a~z) と下線 (_)
<英字記号>	アルファベット (A~Z, a~z) と#, @, ¥
<英数字>	英字と数字 (0~9)
<英数字記号>	英字記号と数字
<符号なし整数>	数字
<識別子>	先頭がアルファベットの英数字列
<文字列>	任意の文字の配列
<記号名称>	先頭が英字記号の英数字記号 UNIX 版の場合、¥は使用できません。
<パス名>*	UNIX 版の場合：/, 英数字, ピリオド (.), #, および@ Windows 版の場合：¥, 英数字, ピリオド (.), 空白, 丸括弧, #, および@

注

すべて半角文字を使用してください。また、英字の大文字と小文字は区別されます。さらに、パス名は使用している OS に依存します。

注※

パス名に空白、または丸括弧を含む場合は、前後を引用符 ("") で囲んでください。

なお、Windows 版の場合、コロン (:) をドライブ名に使用できます。

■ HiRDB のデータベース言語の出典

このマニュアルで記述する HiRDB のデータベース言語仕様は、次に示す規格を基に日立製作所独自の解釈と仕様を追加したものです。原開発者に謝意を表するとともに、仕様の出典を示します。

- JIS X 3005 規格群 データベース言語 SQL
- ISO/IEC 9075 Information technology - Database languages - SQL -

注

JIS : 日本工業規格 (Japanese Industrial Standard)

ISO : 国際標準化機構 (International Organization for Standardization)

IEC : 国際電気標準会議 (International Electrotechnical Commission)

目次

前書き 2
変更内容 4
はじめに 5

1 概要 12

1.1 特長 13
1.1.1 目的 13
1.1.2 適用形態 13
1.2 UAP の開発の流れ 16
1.3 UAP の形式 17
1.3.1 ソースプログラムの記述 17
1.3.2 SQL の記述 17
1.3.3 XDM/RD E2 とのインターフェース 18

2 動作環境 20

2.1 ソフトウェア構成 21
2.1.1稼動 OS 21
2.1.2 プログラムプロダクト構成 21

3 UAP の設計 23

3.1 UAP 中での SQL の基本構成 24
3.1.1 UAP 中での SQL の基本構成の説明 24
3.2 UAP の記述 29
3.2.1 UAP の記述言語 29
3.2.2 インタフェース領域 29
3.3 性能向上、操作性向上に関する UAP の設計 31
3.3.1 配列を使用した FETCH 機能 31
3.3.2 複数接続機能 31
3.3.3 位置付け子機能 34
3.4 他プログラムプロダクトとの関連 35
3.4.1 OpenTP1 との関連 (XA インタフェースを使用する場合) 35
3.4.2 OpenTP1 との関連 (XA インタフェースを使用しない場合) 47
3.4.3 OpenTP1 以外の他プログラムプロダクトとの関連 47

4	クライアントの環境設定 49
4.1	HiRDB クライアントの種類 50
4.2	HiRDB クライアントの環境設定手順 51
4.3	hosts ファイルの設定 52
4.4	クライアント環境定義（環境変数の設定） 53
4.4.1	XDM/RD E2 接続機能使用時の環境変数の差異 53
4.4.2	クライアント環境定義の設定内容 66
5	UAP の作成、実行 83
5.1	UAP の作成 84
5.1.1	SQL のデータ型とデータ記述 84
5.1.2	UAP の形態移行 84
5.2	UAP の実行 86
5.2.1	プリプロセス 86
5.2.2	UAP 実行時の注意事項 88
5.2.3	XDM/RD E2 接続機能を利用するライブラリ 88
5.3	ODBC 対応アプリケーションプログラムからの XDM/RD E2 アクセス 91
5.4	JDBC 対応アプリケーションプログラムからの XDM/RD E2 アクセス 92
5.4.1	Type4 JDBC ドライバを使用した XDM/RD E2 アクセス 92
5.5	UAP の障害対策 94
5.5.1	SQL トレース機能 94
5.5.2	SQL トレース動的取得機能 96
6	SQL 言語仕様 97
6.1	基本項目 98
6.1.1	SQL の記述形式 98
6.1.2	データ型 108
6.2	構成要素の詳細 112
6.2.1	スカラ関数 HEX 112
6.3	SQL 領域 113
6.3.1	SQL 連絡領域 (SQLCA) 113
6.3.2	SQL 記述領域 (SQLDA) 117
6.3.3	列名記述領域 (SQLNAME) 117
6.3.4	型名記述領域 117
6.4	定義系 SQL 118
6.5	操作系 SQL 120
6.5.1	概要 120
6.5.2	データベース操作時の注意事項 121
6.6	制御系 SQL 124

6.7	SQL 制御文	126
6.8	SQL 診断文	127
6.9	埋込み言語文法	128

付録 130

付録 A	SQL のデータ型とデータ記述	131
付録 A.1	SQL のデータ型と C 言語のデータ記述	131
付録 A.2	SQL のデータ型と COBOL 言語のデータ記述	138

索引 148

1

概要

この章では、HiRDB クライアントの XDM/RD E2 接続機能の概要について説明します。

1.1 特長

HiRDB クライアントでの XDM/RD E2 接続機能の目的、および適用形態について説明します。

1.1.1 目的

情報処理システムを構築するとき、PC、WS、メインフレームを適材適所に使用することが求められます。例えば、開発ツールの豊富な PC または WS 上で UAP を開発して、メインフレーム上のデータベースに接続するなどです。

XDM/RD E2 接続機能は、メインフレームの VOS3/FS または VOS3/LS 上で動作する XDM/RD E2 に、PC または WS 上で動作する HiRDB の UAP から直接アクセスする機能です。XDM/RD E2 接続機能を利用することで、メインフレーム（XDM/RD E2）をハイエンドデータベースサーバとして利用できます。

XDM/RD E2 接続機能では、HiRDB クライアントから DB コネクションサーバを経由して XDM/RD E2 に接続します。

XDM/RD E2 接続機能の特長を次に示します。

- XDM/RD E2 のデータベースをアクセスする SQL を含んだ埋込み型 UAP を、PC、WS 上で開発・実行できます。また、HiRDB で構築したシステムで、データベースだけを VOS3 上の XDM/RD E2 へ移行し、UAP はそのまま PC または WS 上に残して、PC、WS 上の UAP から XDM/RD E2 のデータベースをアクセスできます。
- HiRDB クライアントでは、サーバを HiRDB サーバと XDM/RD E2 の 2 種類から選択できます。このとき、クライアント側はサーバの DBMS をほとんど意識しなくてもよいとともに、HiRDB クライアントとしてサポートしている機能（各種言語のプリプロセッサ、ODBC ドライバなど）をそのまま使用できます。
- VOS3 側の SQL 実行空間として、DB コネクションサーバを使用します。

DB コネクションサーバでは、クライアント数の増加に対応して SQL を処理するためのサーバ空間を追加したり、接続するサーバ空間を負荷に応じて自動的に選択したりできるため、クライアント数の増大などのシステムの大規模化に柔軟に対応できます。

1.1.2 適用形態

XDM/RD E2 接続機能を利用したシステムの代表的な適用形態について説明します。

(1) 2階層型システム

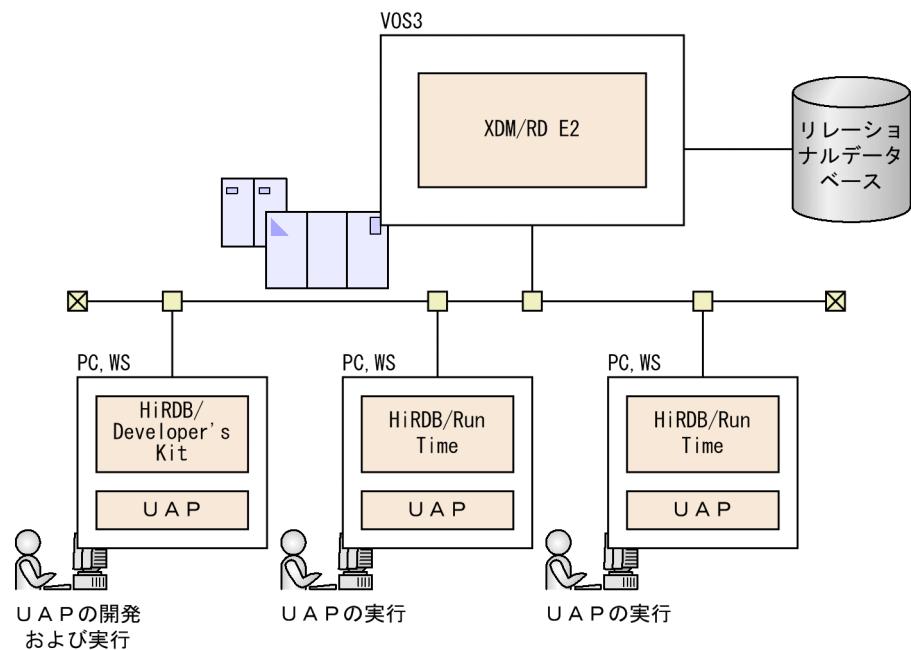
PC 上に XDM/RD E2 をアクセスする SQL を含んだ埋込み型 UAP を配置し、直接 XDM/RD E2 をアクセスする形態です。

従来は、同様のシステム形態にするためには PC 上の UAP で ODBC 関数などを用いていましたが、ODBC 関数を利用した UAP の開発には適用言語の制限などがありました。

XDM/RD E2 接続機能を利用することで、UAP に SQL を直接記述できるため、多様な UAP を作成できます。

2 階層型システムの例を次の図に示します。

図 1-1 2 階層型システムの例



HiRDB サーバには HiRDB クライアントの機能が含まれますので、HiRDB サーバのあるマシンに UAP を置いて、その UAP から XDM/RD E2 にアクセスする形態もできます。

(2) 3 階層型システム

WS 上に OpenTP1 を配置し、トランザクション管理をします。

OpenTP1 の UAP は PC 上の UAP とメッセージを送受信し、かつ XDM/RD E2 をアクセスする SQL を発行します。PC 上の UAP は、メッセージの送受信だけを行い、SQL は発行しません。

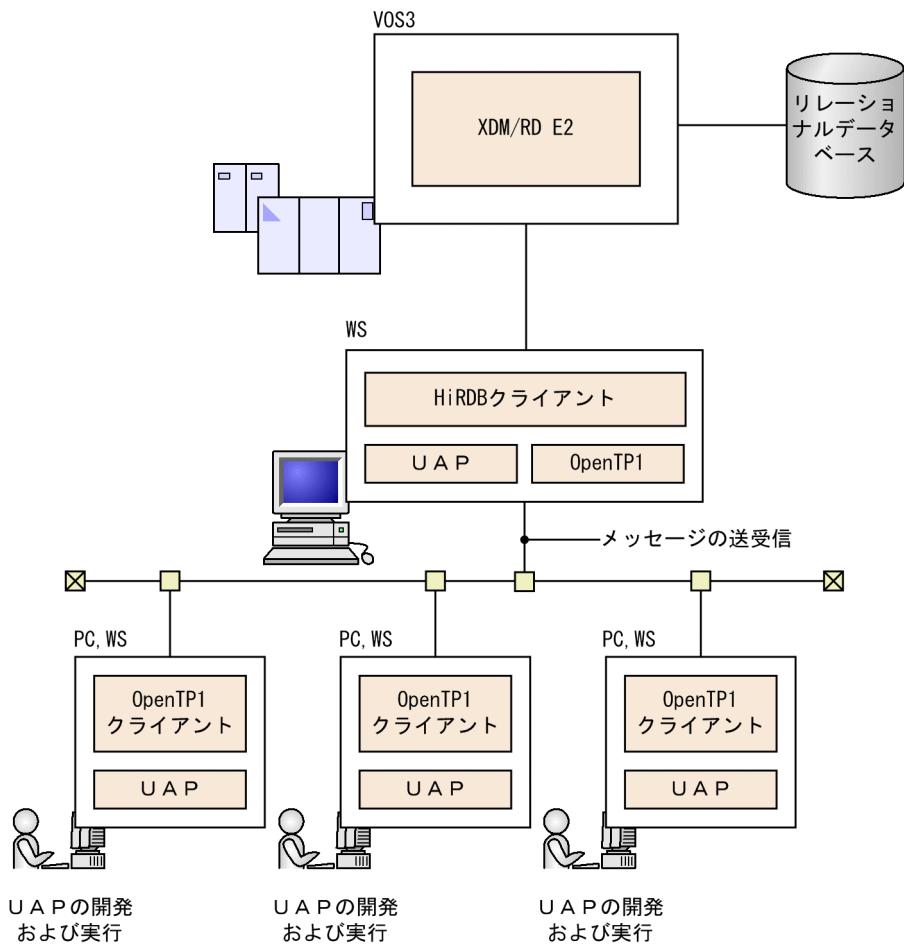
OpenTP1 を使用してトランザクション管理をすることで、2 階層型システムに比べてクライアント数を大幅に増やすことができます。

類似のシステム形態として、WS を使用する代わりに VOS3 上に XDM/DCCM3 を配置する形態 (VOS3 集中型) があります。

VOS3 集中型に比べて、3 階層型システムは、WS とメインフレームに機能および負荷を分散することで、さらに大規模なシステムを構築できます。

3 階層型システムの例を次の図に示します。

図 1-2 3 階層型システムの例

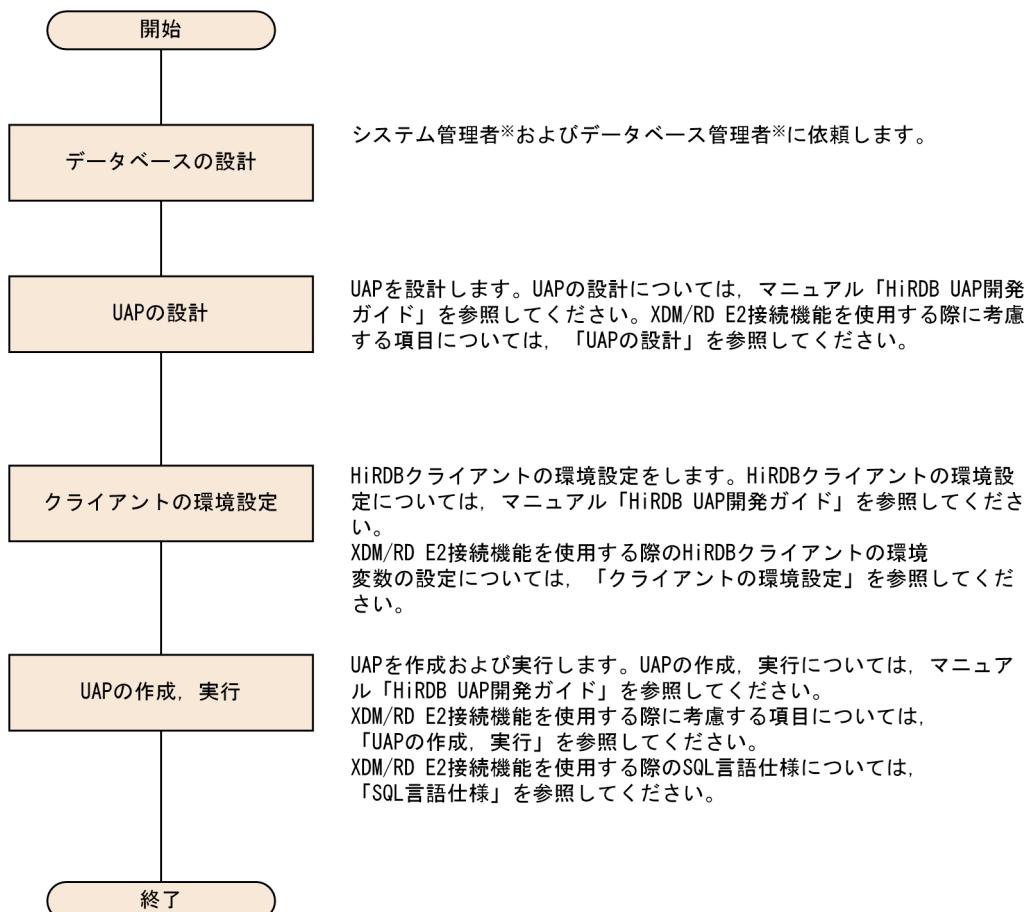


1.2 UAP の開発の流れ

UAP を開発する準備として、業務などで使用しているデータをデータベース化するために業務内容を分析します。分析した結果を基に全体的なデータベースの規模を検討し、UAP の概略を決定します。

UAP の開発作業と、このマニュアルの構成の関係を次の図に示します。

図 1-3 UAP の開発の流れ



注※

作業の内容については、XDM/RD E2 のマニュアルを参照してください。

1.3 UAP の形式

XDM/RD E2 のデータベースを操作する UAP の形式は、埋込み型です。

埋込み型は、高級言語で記述されたソースプログラムの中に SQL を直接記述する方式です。

埋込み型 UAP の特長は、データベース操作（SQL）を含めて一つのプログラムとして記述できるため、プログラムの解析が容易なことです。

1.3.1 ソースプログラムの記述

埋込み型 UAP に使用できる高級言語は、次のとおりです。

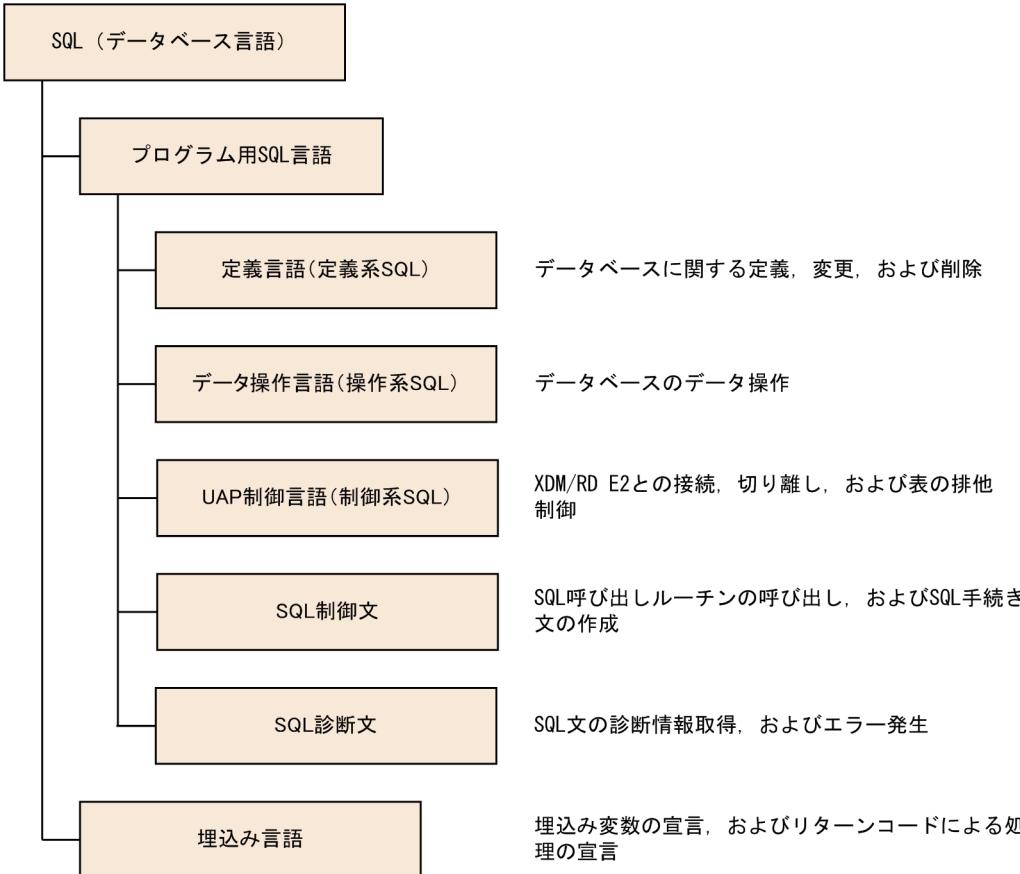
- C 言語
- C++ 言語
- COBOL 言語
- OOCOBOL 言語

1.3.2 SQL の記述

SQL は、データベースの定義から操作、運用制御までの指示を記述するためのデータベース言語です。

HiRDB では、高級言語で記述したソースプログラムに埋め込んで利用できます。SQL の機能体系を次の図に示します。

図 1-4 SQL の機能体系



サーバとして XDM/RD E2 を使用する場合、使用できる SQL の仕様は XDM/RD E2 が規定する仕様に従います。

ただし、埋込み言語については、HiRDB の仕様に従います。

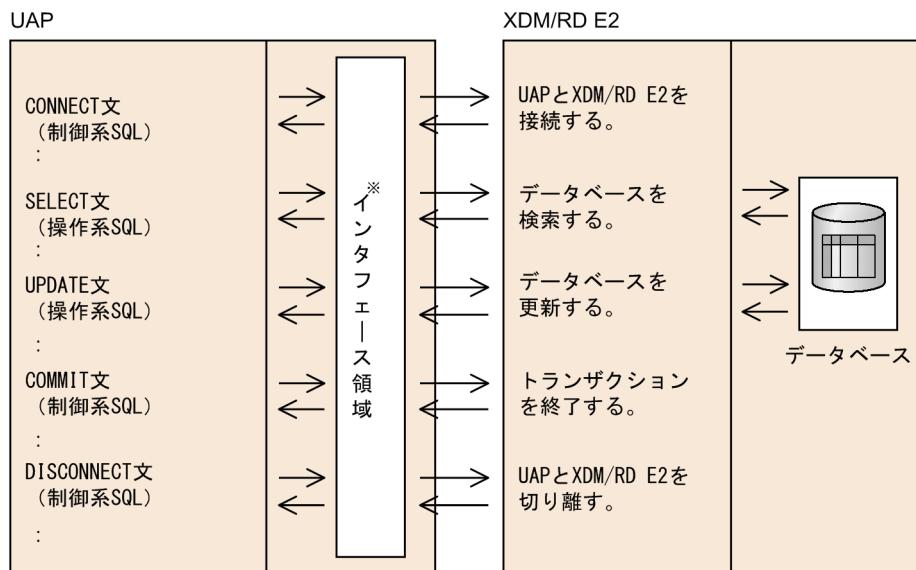
XDM/RD E2 接続機能を使用する場合のプログラム用 SQL 言語の種類と機能概略については、「[SQL 言語仕様](#)」を参照してください。また、XDM/RD E2 接続機能を使用する場合に使用できる埋込み言語については、「[埋込み言語文法](#)」を参照してください。

1.3.3 XDM/RD E2 とのインターフェース

UAP は SQL 文を発行することによって、インターフェース領域を使用して XDM/RD E2 との情報のやり取りをします。

UAP と XDM/RD E2 とのインターフェースを次の図に示します。

図 1-5 UAP と XDM/RD E2 とのインターフェース



注※

インターフェース領域については、「インターフェース領域」を参照してください。

2

動作環境

この章では、XDM/RD E2 接続機能の動作環境について説明します。

2.1 ソフトウェア構成

XDM/RD E2 接続機能を使用するために必要な稼働 OS と、プログラムプロダクト構成について説明します。

2.1.1 稼働 OS

XDM/RD E2 接続機能を利用できるクライアント側 OS を次の表に示します。

表 2-1 XDM/RD E2 接続機能を利用できるクライアント側 OS

クライアント種別	XDM/RD E2 接続機能を利用できるクライアント側 OS
WS	<ul style="list-style-type: none">AIXLinux
PC	<ul style="list-style-type: none">Windows

2.1.2 プログラムプロダクト構成

XDM/RD E2 接続機能を使用する場合に、メインフレームに必要なプログラムプロダクトを次の表に、XDM/RD E2 接続機能を使用するために必要なプログラムプロダクト構成を図「[XDM/RD E2 接続機能を使用するために必要なプログラムプロダクト構成](#)」に示します。

表 2-2 XDM/RD E2 接続機能を使用する場合に、メインフレームに必要なプログラムプロダクト

プログラムプロダクト	バージョン
XDM/RD E2 ^{*1}	06-00 以降
DB コネクションサーバ	03-00 以降
XDM/Library for HiRDB ^{*2}	01-00 以降

注※ 1

機能によって、使用できる XDM/RD E2 のバージョンが違います。XDM/RD E2 のバージョンに依存する機能について、次に示します。

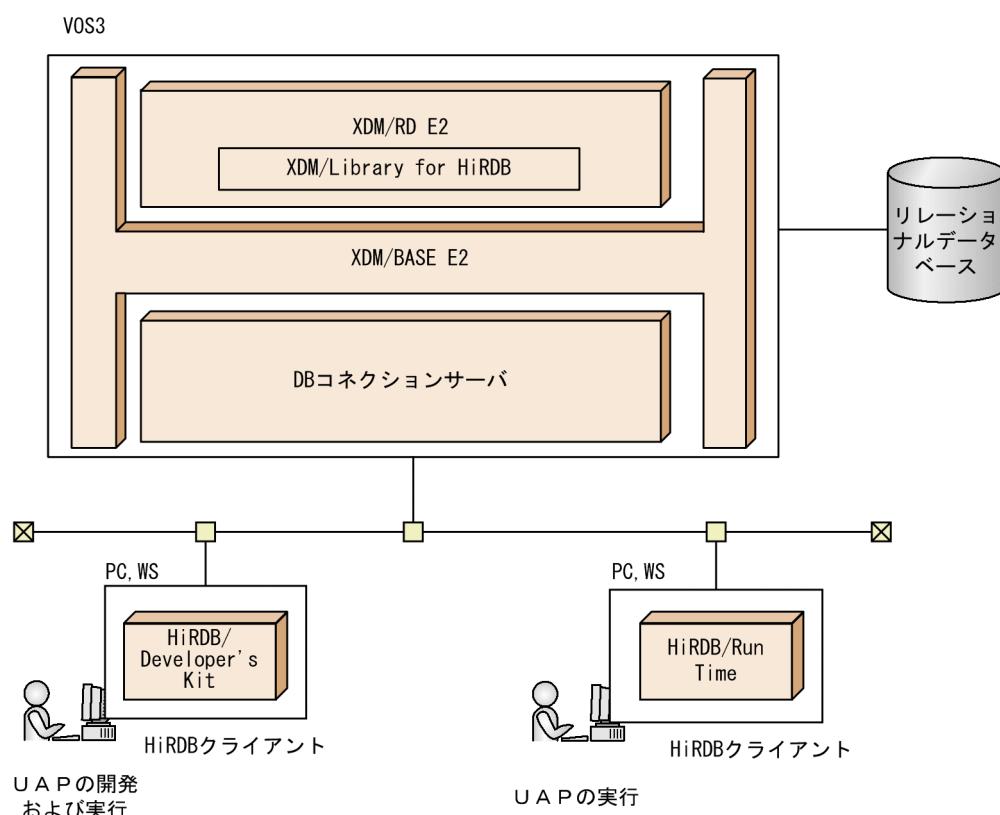
項番	機能	機能が使用できる XDM/RD E2 のバージョン
1	HiRDB クライアントの文字コードが日本語 EUC の場合の、XDM/RD E2 接続	09-01 以降
2	各種のデータ型	表「 HiRDB クライアントからアクセスできる XDM/RD E2 のデータ型 」を参照
3	明示表	明示表 を参照

項目番号	機能	機能が使用できる XDM/RD E2 のバージョン
4	各種の操作系 SQL	表「HiRDB クライアントを使用する場合の操作系 SQL の使用可否」を参照
5	各種の定義系 SQL	表「HiRDB クライアントを使用する場合の定義系 SQL の使用可否」を参照
6	各種の制御系 SQL	表「HiRDB クライアントを使用する場合の制御系 SQL の使用可否」を参照
7	繰返し列	07-00 以降
8	配置を使用した FETCH	08-00 以降
9	X/Open に連携したアプリケーションプログラムインターフェースを使用する	06-03 以降
10	Type4 JDBC ドライバを使用した XDM/RD E2 接続	11-03 以降

注※ 2

使用する機能によっては、より新しいバージョンの製品が必要になることがあります。詳細については、「SQL 言語仕様」を参照してください。

図 2-1 XDM/RD E2 接続機能を使用するために必要なプログラムプロダクト構成



3

UAP の設計

この章では、XDM/RD E2 接続機能を使用する際に、UAP の設計段階で考慮する項目について説明します。

基本的な UAP の設計については、マニュアル「HiRDB UAP 開発ガイド」を参照してください。

3.1 UAP 中での SQL の基本構成

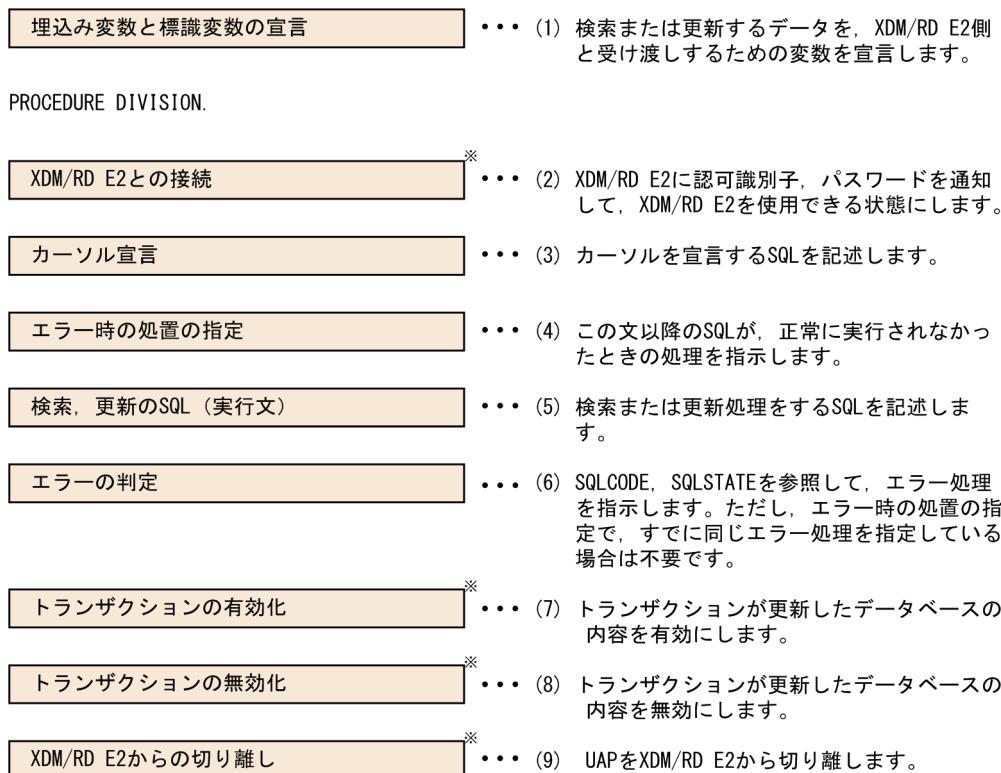
3.1.1 UAP 中での SQL の基本構成の説明

ここでは、UAP 中での SQL の基本構成の中で、特に HiRDB サーバを使用する場合と XDM/RD E2 を使用する場合で差異がある項目について説明します。

UAP 中での SQL の基本構成を次の図に示します。なお、ここでは UAP を COBOL 言語で記述する場合について説明します。

図 3-1 UAP 中での SQL の基本構成

DATA DIVISION.
WORKING-STORAGE SECTION.



注 番号は、それぞれ以降の説明の項番と対応しています。

注※ 必要に応じて、「エラー時の処置の指定」または「エラーの判定」によるエラー処理を指定してください。ただし、「エラー時の処置の指定」によって、「トランザクションの無効化」が無限ループにならないように注意してください。

(1) 埋込み変数と標識変数の宣言

SQL で検索したデータを UAP 側で受け取ったり、反対に UAP 側のデータを SQL で表に挿入したりするには、両方の言語間の橋渡しをする変数が必要になります。このために、埋込み変数を使用します。また、ナル値を含むデータを受け渡す必要のあるときには、埋込み変数とあわせて標識変数を使用します。

SQL 中での埋込み変数および標識変数の指定方法については、「[検索、更新の SQL \(実行文\)](#)」を参照してください。

埋込み変数および標識変数の宣言例を次に示します。

```
EXEC SQL
  BEGIN DECLARE SECTION .....1
END-EXEC.
77 XUSERID    PIC X(7).....2
77 XPSWD      PIC X(7).....2
77 XSNAME     PIC X(4).....2
77 XSNAME     PIC N(8).....2
77 XGRYO      PIC S9(9) COMP.....2
77 IGRYO      PIC S9(4) COMP.....3
EXEC SQL
  END DECLARE SECTION .....4
END-EXEC.
```

説明

1. 埋込み変数の宣言の始まりを示します。
2. 埋込み変数を宣言します。SQL と UAP 側でデータを受け渡しするときには、あらかじめ決められた規則に従って記述します。SQL のデータ型とデータ記述については、「[SQL のデータ型とデータ記述](#)」を参照してください。
3. 埋込み変数 XGRYO に対する標識変数を宣言します。なお、BLOB 型の埋込み変数に対する標識変数の場合は、PIC S9(9) COMP.となります。
4. 埋込み変数の宣言の終わりを示します。

ナル値の既定値設定機能を使用している場合、検索結果がナル値のときはナル値の代わりに既定値（数データの場合は 0、文字データの場合は空白）を埋込み変数で受け取ることができます。このとき、既定値とナル値とを区別しなくてよいのであれば、標識変数を使用する必要がなくなります。ナル値の既定値設定機能については、マニュアル「VOS3 データマネジメントシステム XDM E2 系 XDM/RD E2 SQL リファレンス」を参照してください。

(2) XDM/RD E2 との接続

XDM/RD E2 にユーザの認可識別子とパスワードを通知して、UAP が XDM/RD E2 を使用できる状態にします。これを XDM/RD E2 との接続といいます。XDM/RD E2 との接続方法を次に示します。

```
EXEC SQL
  CONNECT :XUSERID IDENTIFIED BY :XPSWD
END-EXEC.
```

説明

埋込み変数 (:XUSERID) に格納された認可識別子および埋込み変数 (:XPSWD) に格納されたパスワードで XDM/RD E2 と接続します。

認可識別子

XDM/RD E2 と接続する場合に有効となる認可識別子を次に示します。

優先順位	認可識別子の指定優先順序
1	CONNECT 文で指定した認可識別子
2	環境変数 PDUSER で指定した値

埋込み型 UAP 中の SQL に、認可識別子を省略した場合に仮定する認可識別子を次に示します。

- 定義系 SQL および動的 SQL

優先順位	認可識別子の指定優先順序
1	CONNECT 文で指定した認可識別子
2	1 を省略した場合は、環境変数 PDUSER で指定した認可識別子

- 定義系 SQL 以外の静的 SQL

優先順位	認可識別子の指定優先順序
1	プリプロセス時の「A」オプションで指定した既定認可識別子
2	1 を省略した場合は、CONNECT 文で指定した認可識別子
3	1, 2 を省略した場合は、環境変数 PDUSER で指定した認可識別子

CALL 文で呼び出すプロジェクト中の SQL に、認可識別子を省略した場合に仮定する認可識別子を次に示します。

- 静的 SQL

プロジェクト所有者（外部プロジェクトとして呼び出す UAP の所有者と同じ）

- 動的 SQL

優先順位	認可識別子の指定優先順序
1	CONNECT 文で指定した認可識別子
2	1 を省略した場合は、環境変数 PDUSER で指定した認可識別子

指定できる認可識別子とパスワードの規則は、XDM/RD E2 の規則に従います。詳細については、マニュアル「VOS3 データマネジメントシステム XDM E2 系 XDM/RD E2 SQL リファレンス」を参照してください。

(3) カーソル宣言

HiRDB サーバを使用する場合と XDM/RD E2 を使用する場合で特に差異はありません。カーソル宣言については、マニュアル「HiRDB UAP 開発ガイド」を参照してください。

(4) エラー時の処置の指定

HiRDB サーバを使用する場合と XDM/RD E2 を使用する場合で特に差異はありません。エラー時の処置の指定については、マニュアル「HiRDB UAP 開発ガイド」を参照してください。

(5) 検索、更新の SQL（実行文）

データを検索、更新、挿入、または削除するための SQL 文を記述します。各 SQL 文の記述方法については、マニュアル「VOS3 データマネジメントシステム XDM E2 系 XDM/RD E2 SQL リファレンス」を参照してください。

ただし、XDM/RD E2 接続機能を使用する場合は、埋込み変数および標識変数の形式として、INDICATOR 句は指定できません。XDM/RD E2 の形式と XDM/RD E2 接続機能を使用する場合の形式を次に示します。

- XDM/RD E2 の形式

```
埋込み変数指定 : : = [GROUPITEM] 埋込み変数名 [標識変数名]  
パラメタ指定 : : = パラメタ名 [標識パラメタ名]
```

```
標識変数名 : : = [INDICATOR] 埋込み変数名  
標識パラメタ名 : : = [INDICATOR] パラメタ名  
埋込み変数名 : : = : [ホスト識別子1.] ホスト識別子2
```

- XDM/RD E2 接続機能を使用する場合の形式

```
埋込み変数指定 : : = [GROUPITEM] 埋込み変数名 [標識変数名]  
パラメタ指定 : : = パラメタ名 [標識パラメタ名]
```

```
標識変数名 : : = 埋込み変数名  
標識パラメタ名 : : = パラメタ名  
埋込み変数名 : : = : [ホスト識別子1.] ホスト識別子2
```

(6) エラーの判定

SQL の実行中にエラーが発生した場合、SQLCODE、SQLSTATE を参照すると、XDM/RD E2 から返されるリターンコードがわかります。そのリターンコードを利用して、その後の処理をどうするか指示します。ただし、「エラー時の処置の指定」で、すでに同じ内容のエラー処理を指定している場合は、ここで指示する必要はありません。

XDM/RD E2 接続機能を使用する場合は、SQLCODE 変数で受け取る SQLCODE は、XDM/RD E2 が設定する SQLCODE または HiRDB クライアントが設定する SQLCODE となります。また、SQLSTATE 変数で受け取る値は、XDM/RD E2 が設定する SQLSTATE または HiRDB クライアントが設定する SQLSTATE となります。

SQLCODE および SQLSTATE については、マニュアル「VOS3 データマネジメントシステム XDM E2 系 XDM/RD E2 SQL リファレンス」を参照してください。

DECLARE CURSOR のような宣言文の直後に、エラーの判定はしないでください。エラーの判定をすると、不正な SQLCODE を参照して、XDM/RD E2 が誤動作する場合があります。

(7) トランザクションの有効化

HiRDB サーバを使用する場合と XDM/RD E2 を使用する場合で特に差異はありません。トランザクションの有効化については、マニュアル「HiRDB UAP 開発ガイド」を参照してください。

(8) トランザクションの無効化

HiRDB サーバを使用する場合と XDM/RD E2 を使用する場合で特に差異はありません。トランザクションの無効化については、マニュアル「HiRDB UAP 開発ガイド」を参照してください。

(9) XDM/RD E2 からの切り離し

トランザクションを正常終了させて、UAP を XDM/RD E2 から切り離します。DISCONNECT 文は、RELEASE 指定の COMMIT 文を実行したときと同じになります。

トランザクションを正常終了させて、UAP を XDM/RD E2 から切り離す SQL 文を次に示します。

```
EXEC SQL
  DISCONNECT
END-EXEC.
```

説明

トランザクションを正常終了させて、UAP を XDM/RD E2 から切り離します。また、トランザクションを取り消したあとに UAP を XDM/RD E2 から切り離す場合には、RELEASE 指定の ROLLBACK 文を実行します。なお、DISCONNECT 文、COMMIT 文 (RELEASE 指定)、ROLLBACK 文 (RELEASE 指定) のどれも実行しないで UAP が終了した場合、ROLLBACK 文 (RELEASE 指定) が自動的に実行され、実行中のトランザクションは無効となるので注意してください。

3.2 UAP の記述

ここでは、UAP を作成するときに考慮する必要がある基本的な内容の中で、特に、HiRDB サーバを使用する場合と XDM/RD E2 を使用する場合で差異がある項目について説明します。

その他の UAP の記述については、マニュアル「HiRDB UAP 開発ガイド」を参照してください。

3.2.1 UAP の記述言語

UAP の形式は、高級言語で記述されたソースプログラム中に直接 SQL を記述する埋込み型です。

XDM/RD E2 接続機能で使用できる UAP の記述言語を次の表に示します。

表 3-1 UAP の記述言語

動作環境	記述言語
AIX, Linux	C 言語（最適化 C） C++ 言語（最適化 C++） COBOL 言語（COBOL85 および COBOL2002） OOCOBOL 言語（OOCOBOL）※
Windows	C 言語（Microsoft Visual C++） C++ 言語（Microsoft Visual C++） COBOL 言語（COBOL85 および COBOL2002） OOCOBOL 言語（OOCOBOL）

注※

OS によっては、コンパイラがサポートされていないため、使用できない場合があります。

なお、埋込み型 UAP は、そのままではコンパイルおよびリンクエージができません。SQL プリプロセサを実行し、ポストソースプログラムに変換してからコンパイルとリンクエージをしてください。プリプロセス、コンパイル、およびリンクエージについては、マニュアル「HiRDB UAP 開発ガイド」を参照してください。

3.2.2 インタフェース領域

インターフェース領域は、XDM/RD E2 と UAP との間で情報をやり取りするために使用します。インターフェース領域の種類と使用目的を次の表に示します。

表 3-2 インタフェース領域の種類と使用目的

領域名	使用目的	言語	
		C	COBOL
SQL 連絡領域	SQL の実行結果の詳細情報を取得する	○※1	○※1
SQL 記述領域	<ul style="list-style-type: none"> UAP 実行時に動的に決定した入力変数の情報をシステムに通知する UAP を動的に実行するために前処理した SQL の検索項目の情報を受け取る 列名記述領域を指定する 	△	△
列名記述領域	UAP を動的に実行するために前処理した SQL の検索項目の情報を受け取る	△	△
型名記述領域	ユーザ定義型のデータ型名を受け取る	△	△
埋込み変数	埋込み型 UAP の SQL 中に指定して値の受け渡しをする	△	△
標識変数	埋込み型 UAP の SQL 中に指定して値の受け渡しをする	△	△
?パラメタ	UAP を動的に実行するために前処理する SQL に対して、UAP から値を渡す	△	△※2

(凡例)

○：必要

△：任意

注※1

SQL プリプロセサを実行すると UAP 中に展開されるので、宣言は不要です。SQL プリプロセサの実行については、マニュアル「HiRDB UAP 開発ガイド」を参照してください。

注※2

?パラメタの代わりに埋込み変数および標識変数を使用します。

XDM/RD E2 接続機能使用時の SQL 連絡領域、SQL 記述領域、列名記述領域、および型名記述領域の詳細については、「[SQL 領域](#)」を参照してください。

また、埋込み変数、標識領域、および?パラメタの詳細については、マニュアル「HiRDB SQL リファレンス」を参照してください。

3.3 性能向上、操作性向上に関する UAP の設計

ここでは、性能向上と操作性向上に関して UAP 設計時に考慮する点の中で、特に HiRDB サーバを使用する場合と XDM/RD E2 を使用する場合で差異がある項目について説明します。

その他の性能向上、および操作性向上に関する UAP 設計については、マニュアル「HiRDB UAP 開発ガイド」を参照してください。

3.3.1 配列を使用した FETCH 機能

XDM/RD E2 のバージョンによって、配列を使用した FETCH 機能を使用できる場合とできない場合があります。配列を使用した FETCH 機能を使用できるバージョンを次に示します。

配列を使用した FETCH 機能を使用できるバージョン

XDM/RD E2 08-00 以降

3.3.2 複数接続機能

(1) 機能概要

複数接続機能は、HiRDB クライアントで一つの UAP から、XDM/RD E2 に対して複数の CONNECT を別々に実行できるようにする機能です。

複数接続機能のそれぞれの接続は独立していて、DB コネクションサーバのタスクは接続ごとに割り当てられて別々のトランザクション（別々のアクティビティ）として処理されるので、UAP は複数の SQL 文を同時に実行できます。一つの UAP から複数接続できるので、実行する UAP の数を削減でき、全体としての UAP のメモリ所要量を削減できます。

XDM/RD E2 の最大接続数は、接続ごとに別ユーザとしてカウントされるので、ユーザ数の上限ではなく、同時に CONNECT する接続数の上限となります。

複数接続機能の特長を次に示します。

- 接続ごとに、異なるユーザ ID、パスワードを使用できます。
- 接続ごとに、複数の XDM/RD E2 へ接続できます。環境変数グループを使用する場合は、ほかの HiRDB サーバと同時に接続することもできます。

複数接続機能の処理概要を図「複数接続機能の処理概要（マルチスレッドを使用しない場合）」および図「複数接続機能の処理概要（マルチスレッドを使用する場合）」に示します。

図 3-2 複数接続機能の処理概要（マルチスレッドを使用しない場合）

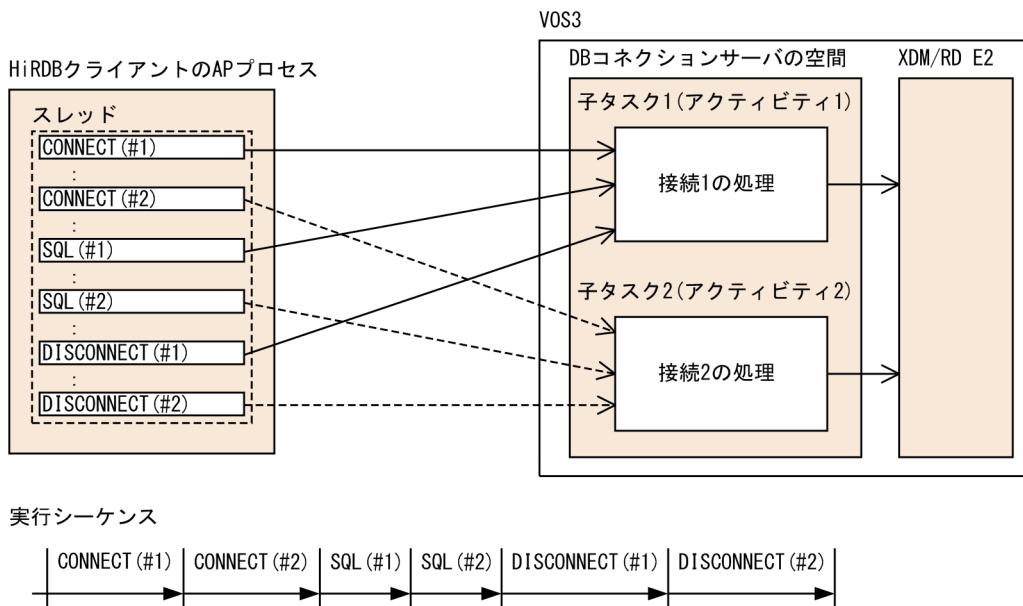
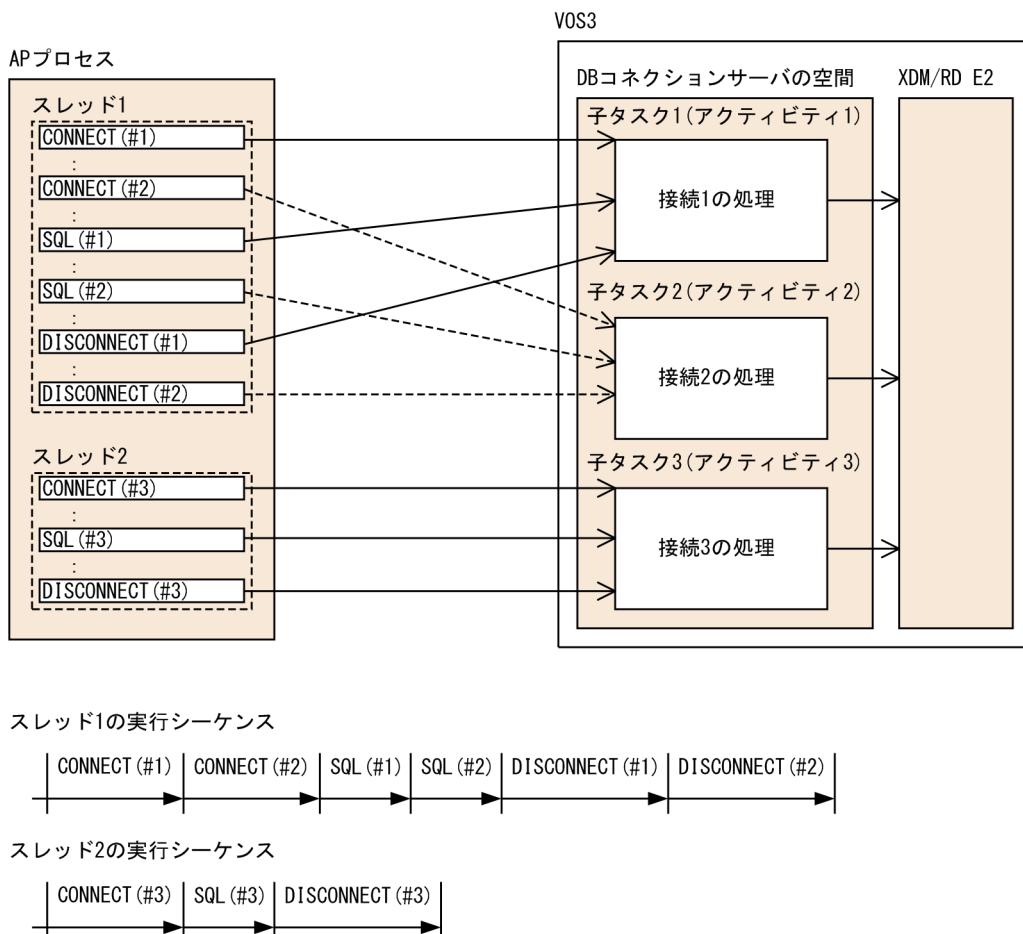


図 3-3 複数接続機能の処理概要（マルチスレッドを使用する場合）



注 それぞれの接続が独立しているので、スレッドごとに同時にSQLを実行できます。

(2) コーディング例

複数接続機能を使用した場合の UAP のコーディング例については、マニュアル「HiRDB UAP 開発ガイド」を参照してください。

(3) 規則

- 複数接続機能を使用する場合は、UAP のコンパイルおよびリンクージ時に専用のライブラリを指定する必要があります。

複数接続機能を使用する場合のコンパイルとリンクージについては、マニュアル「HiRDB UAP 開発ガイド」を参照してください。

- X/Open XA インタフェース環境下での複数接続機能については、マニュアル「HiRDB UAP 開発ガイド」を参照してください。
- 複数接続機能用ライブラリを使用する UAP では、一つの接続を保持したままスレッドを分岐し、そのスレッドが SQL を実行する場合、ほかの SQL を発行するスレッドとの間で処理のシリアル化を必要とします。したがって、同一接続に対する SQL は同時に発行できません。異なる接続に対する SQL は同時に発行できます。
- ALLOCATE CONNECTION HANDLE, FREE CONNECTION HANDLE のエラー情報を参照する場合、SQLCODE, SQLERRM ではなく、リターンコード受け取り変数の値を参照してください。リターンコード受け取り変数については、マニュアル「HiRDB SQL リファレンス」を参照してください。
- 各接続で発行する SQL は、XDM/RD E2 上ではそれぞれ別のアクティビティとして扱われます。

このため、ほかの接続で排他されている資源を別の接続で排他しようとした場合に、永久的な排他待ちになることがありますので注意してください※1。

また、スレッド間で発行する SQL の組み合わせによってデッドロックエラーになる場合がありますので注意してください※2。

注※1 例を次に示します。

- 接続 1 が表 1 を更新
- 同ースレッド内で、接続 2 が表 1 を更新

項番 2.の操作が項番 1.の操作を排他待ちする場合があり、この場合時間監視の制限値まで待ち続けることになります。

注※2 例を次に示します。

- スレッド 1 で表 1 を更新
- スレッド 2 で表 2 を更新
- スレッド 2 で表 1 を更新
- スレッド 1 で表 2 を更新

項番 3.の操作が項番 1.の操作を排他待ちし、項番 4.の操作が項番 2.の操作を排他待ちする場合があり、デッドロックとなります。

- SQL 連絡領域を参照する場合、直接 SQLCA を参照しないで、SQLC で始まる連絡領域名を参照してください。

(4) プラットフォームによる注意事項

複数接続機能を使用する場合、UAP が動作するプラットフォームによって次のことに注意する必要があります。次の注意が守られていない場合、UAP の実行時に不正な動作をする可能性があります。

(a) UNIX の場合

複数接続機能は、マルチスレッド（DCE スレッド、リアルスレッド）およびシングルスレッド対応の UAP で使用できます。したがって、マルチスレッド対応の UAP で複数接続機能を使用する場合、HiRDB での UAP の開発知識のほかに、DCE スレッド、リアルスレッドを用いた UAP の開発知識を必要とします。

(b) Windows の場合

複数接続機能は、マルチスレッド対応 UAP でだけ使用できます。したがって、UAP のコンパイルで使用する C のランタイムライブラリの指定は、マルチスレッド. DLL を選択してください（「コンパイルオプション：コード生成」で、「マルチスレッド. DLL」を指定します）。

3.3.3 位置付け子機能

XDM/RD E2 のバージョンによって、位置付け子機能を使用できる場合とできない場合があります。位置付け子機能を使用できるバージョンを次に示します。

位置付け子機能を使用できるバージョン

XDM/RD E2 11-00 以降

なお、CLOB 型の位置付け子機能を使用できるのは、XDM/RD E2 接続の場合だけです。HiRDB サーバ接続の場合は使用できません。

3.4 他プログラムプロダクトとの関連

ここでは、他プログラムプロダクトが XDM/RD E2 接続機能を使用する場合の関連事項について説明します。

3.4.1 OpenTP1 との関連 (XA インタフェースを使用する場合)

X/Open に準拠した API を使用して、OLTP 環境でリソースマネージャとして XDM/RD E2 を使用できます。

このとき、OLTP システムは XDM/RD E2 を HiRDB サーバとみなして動作します。したがって、HiRDB の関連マニュアルをお読みになる場合は、「HiRDB」を「XDM/RD E2」に置き換えてお読みください。

(1) 必要となるプログラムプロダクトのバージョン

X/Open に準拠した API を使用して、XDM/RD E2 に接続する場合に必要な VOS3 のプログラムプロダクトのバージョンを次に示します。

- XDM/RD E2 : 06-03 以降
- DB コネクションサーバ : 03-02 以降

上記のプログラムプロダクトに関するプログラムプロダクトのバージョンについては、上記プログラムプロダクトのマニュアルを参照してください。

(2) VOS3 のプログラムプロダクトの定義

X/Open に準拠した API を使用して、XDM/RD E2 に接続する場合に必要な VOS3 のプログラムプロダクトの定義を次に示します。詳細については、各プログラムプロダクトのマニュアルを参照してください。

(a) XDM/RD E2

OLTP システムが OpenTP1 の場合、XDM/RD E2 に接続できるユーザ数を一つ多く定義します。

(b) DB コネクションサーバ

- 二相コミット機能を使用する定義。
- OLTP システムが OpenTP1 の場合、DB コネクションサーバに接続できる TCP コネクション数を一つ多く定義します。

(3) UAP 作成上の留意事項

(a) SQL 関連の注意事項

使用できない SQL

X/Open に準拠した API を使用している UAP からは、次に示す SQL は使用できません。

- 定義系 SQL
- PURGE TABLE 文
- COMMIT 文
- CONNECT 文
- DISCONNECT 文
- ROLLBACK 文

暗黙的ロールバック

X/Open に準拠した API を使用している場合、SQL 実行時に XDM/RD E2 で暗黙的ロールバック要因が発生すると、XDM/RD E2 は自動的にロールバックします。

このため、暗黙的ロールバック要因が発生した場合は、XDM/RD E2 以外の資源に対する整合性を保つために、速やかにトランザクションのロールバック※をする必要があります。

暗黙的ロールバック要因が発生すると、XDM/RD E2 はロールバック以外の要求を実行できなくなります。

XDM/RD E2 で暗黙的ロールバック要因が発生したかどうかは、SQL 連絡領域 (SQLCA) の SQLWARN6 を参照して判断してください。暗黙的ロールバック要因が発生している場合、「W」が設定されています。なお、暗黙的ロールバック要因が発生した場合は、SQLCODE は負の値となります。

注※

ここで説明しているロールバックとは、トランザクションマネージャが用意するロールバック関数などの発行を示すものであり、SQL の ROLLBACK 文ではありません。

(b) マルチスレッド用のライブラリに関する注意事項

一つのトランザクションから XDM/RD E2 に対して複数のスレッドを使用して別々に接続できません。マルチスレッド環境での XA 接続は未サポートです。したがって、一つのトランザクションから同時に実行できるスレッドは一つであり、同一トランザクション内で複数のスレッドを使用して SQL 文を同時に実行できません。

(4) HiRDB XA ライブラリ

X/Open XA インタフェースとは、分散トランザクション処理 (DTP : Distributed Transaction Processing) システムのトランザクションマネージャ (TM : Transaction Manager) とリソースマネージャ (RM : Resource Manager) の接続インターフェースを規定した X/Open の標準仕様です。X/Open XA インタフェースを使用すると、リソースマネージャのトランザクション処理をトランザクションマネージャで制御できます。リソースマネージャのトランザクション処理をトランザクションマネージャで制御するには、

リソースマネージャが提供するライブラリとトランザクションマネージャが提供するライブラリを UAP にリンクageします。

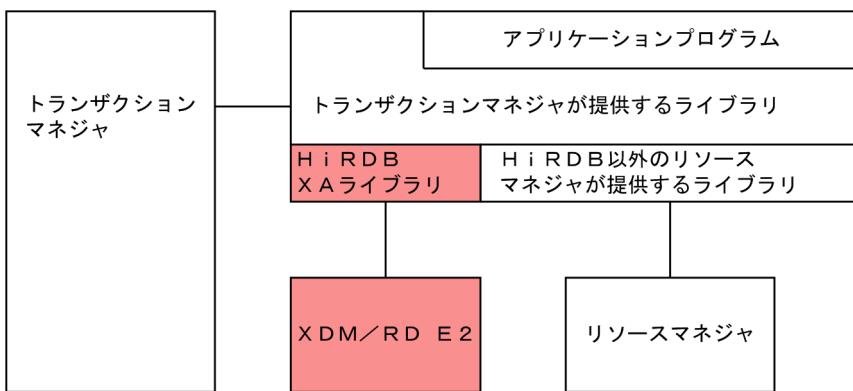
XDM/RD E2 の UAP の処理をトランザクションマネージャで制御するために、HiRDB は HiRDB XA ライブラリを提供しています※。HiRDB XA ライブラリは、X/Open DTP ソフトウェア・アーキテクチャの XA インタフェースの仕様に準拠しています。

注※

XDM/RD E2 に接続する場合も、HiRDB サーバに接続する場合と同じ HiRDB XA ライブラリを使用します。

X/Open DTP モデルでの XDM/RD E2 の位置づけを次の図に示します。

図 3-4 X/Open DTP モデルでの XDM/RD E2 の位置づけ



(5) HiRDB XA ライブラリでサポートする機能

HiRDB XA ライブラリでサポートする機能を次の表に示します。

ただし、XDM/RD E2 をサーバとして使用する場合と HiRDB サーバを使用する場合では、サポートする機能が異なります。

表 3-3 HiRDB XA ライブラリでサポートする機能

機能	説明	サポート有無	
		HiRDB サーバ	XDM/RD E2 サーバ
トランザクションの移行	トランザクションのコミット処理を、UAP が HiRDB にアクセスしたときと異なるサーバプロセスで実行する機能です。ここでいう UAP とは、HiRDB XA ライブラリを使用して HiRDB に接続する UAP のことです。	○	✗*
一相最適化	二相コミット制御を一相に最適化する機能です。	○	○
読み取り専用	プリペア要求で HiRDB のリソースが更新されていない場合、トランザクションマネージャが二相目にコミット要求をしないで最適化する機能です。	○	○

機能	説明	サポート有無	
		HiRDB サーバ	XDM/RD E2 サーバ
動的トランザクションの登録	UAP を実行する直前に、HiRDB が動的にトランザクションを登録する機能です。	○	○
非同期 XA 呼び出し	トランザクションマネージャが非同期に HiRDB XA ライブラリを呼び出す機能です。	×	×
複数接続機能	一つの UAP から XDM/RD E2 に対して複数の CONNECT を別々に実行する機能です。 X/Open XA インタフェース環境下での複数接続機能については、マニュアル「HiRDB UAP 開発ガイド」を参照してください。	○	○

(凡例)

- : HiRDB XA ライブラリでサポートしています。
- × : HiRDB XA ライブラリでサポートしていません。

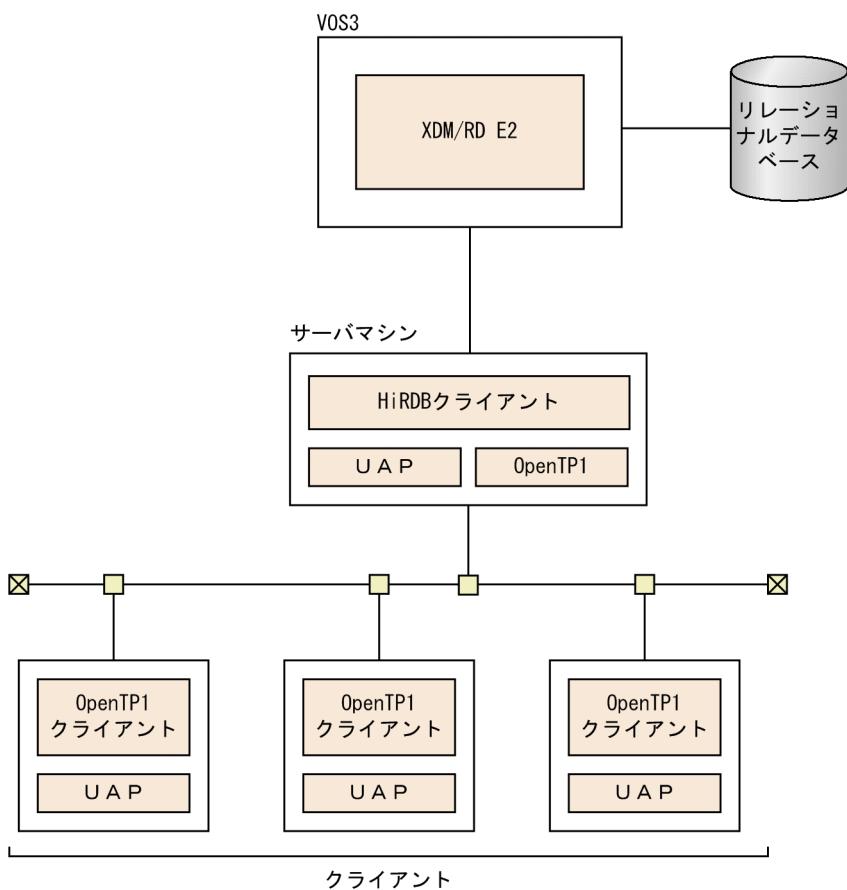
注※

トランザクションの移行は使用できません。このため、HiRDB クライアント環境変数の PDXAMODE オペランドには 0 を指定するか、PDXAMODE オペランドの指定を省略してください。

(6) システムの構成例

OLTP システムを使用したシステムの構成例を次の図に示します。

図 3-5 OLTP システムを使用したシステムの構成例



(7) トランザクションの移行

トランザクションの移行は、XDM/RD E2 に接続する場合には使用できません。

このため、HiRDB クライアント環境変数の PDXAMODE オペランドには 0 を指定するか、PDXAMODE オペランドの指定を省略してください。

(a) トランザクションマネージャが OpenTP1 の場合

XDM/RD E2 に接続する場合は、トランザクションの移行が使用できないため、OpenTP1 のコミット最適化およびプリペア最適化は使用できません。

ただし、OpenTP1 システムの状態によっては、トランザクションの移行を使用しなくても、コミット最適化およびプリペア最適化を使用する指定（OpenTP1 システム定義のトランザクションサービス定義の trnstring オペランドに-d オプションを省略）ができます。

OpenTP1 システム定義のトランザクションサービス定義の `trnstring` オペランドの指定と OpenTP1 システムの状態の組み合わせ可否を次に示します。

条件			組み合わせ可否
trnstring オペランドの指定	OpenTP1 システムの状態		
-d オプションを省略	同一 OpenTP1 システム内では一つの AP だけでグローバルトランザクションを構成している場合		可能
	同一 OpenTP1 システム内では複数の AP でグローバルトランザクションを構成している場合	一つの AP だけが HiRDB XA ライブラリとリンクしているとき	可能
		複数の AP が HiRDB XA ライブラリとリンクしているとき	不可能*
-d オプションを指定	該当しない		可能

注

-d オプションは、TP1/Server Base のバージョンが 03-03 以降のときに指定できます。

注※

同一 OpenTP1 システム内では複数の AP でグローバルトランザクションを構成しており、かつ複数の AP が HiRDB XA ライブラリとリンクしている場合は、trnstring オペランドに必ず-d オプションを指定してください。

trnstring オペランドについては、マニュアル「OpenTP1 システム定義」を参照してください。

コミット最適化およびプリペア最適化については、マニュアル「OpenTP1 プログラム作成の手引」を参照してください。

(8) トランザクションマネジャへの登録

OLTP と連携するには、XDM/RD E2 をトランザクションマネジャに登録する必要があります。

XDM/RD E2 をトランザクションマネジャに登録するには、OpenTP1 の trnlkrm コマンドを使用します。

XDM/RD E2 をトランザクションマネジャに登録するときには、次に示すどちらかの方法を選択してください。

- 動的登録方法
- 静的登録方法

なお、一つのトランザクションマネジャに対して、動的登録方法と静的登録方法を混在して使用できません。

(a) 動的登録方法とは

XDM/RD E2 をトランザクションマネジャに動的登録すると、トランザクション内で最初の SQL 文を発行したときに、UAP がトランザクションマネジャの制御下に入ります。UAP が XDM/RD E2 を含む複数のリソースマネジャをアクセスする場合、または UAP が XDM/RD E2 をアクセスするとは限らない場合などに、トランザクションマネジャからの XDM/RD E2 に対するトランザクション制御のオーバヘッドを削減できます。

(b) 静的登録方法とは

XDM/RD E2 をトランザクションマネージャに静的登録すると、UAP が SQL 文の発行に関係なく、トランザクションの開始時に常にトランザクションマネージャの制御下に入ります。

トランザクションマネージャが OpenTP1 の場合、UAP と XDM/RD E2 とのコネクションが切断されたとき（ユニットの異常終了またはサーバプロセスの異常終了などのとき）に、OpenTP1 にはトランザクション開始時に再接続をする機能があるため、UAP の再起動が不要になります。

(c) 動的登録方法と静的登録方法の違い

動的登録方法と静的登録方法の違いを次の表に示します。

表 3-4 動的登録方法と静的登録方法の違い

差異のポイント	動的登録方法	静的登録方法
トランザクション開始時	管理しない	<ul style="list-style-type: none">コネクション確立中かどうかの確認トランザクションマネージャ制御下でのトランザクションの管理を開始
トランザクション内で最初の SQL 発行時	<ul style="list-style-type: none">トランザクションマネージャの制御下での管理を開始XDM/RD E2 のトランザクション開始SQL 文処理	<ul style="list-style-type: none">XDM/RD E2 のトランザクション開始SQL 文処理
トランザクション処理中のトランザクションマネージャと XDM/RD E2 間の通信回数	SQL 文数 + コミット処理通信回数	SQL 文数 + コミット処理通信回数 + 1 (トランザクション開始処理用の通信分)
トランザクションマネージャと XDM/RD E2 間のコネクションが、途中で切断したときの再接続方法※1	次回のトランザクション開始時に自動的に再接続※2	次回のトランザクション開始時に自動的に再接続※3

注※1

トランザクションマネージャが OpenTP1 のときに限ります。

注※2

OpenTP1 の trn_rm_open_close_scope オペランドに transaction を指定している場合に自動的に再接続します。指定していない場合は、UAP をいったん終了させ、再起動する必要があります。

ただし、UAP が tx_open を再発行すれば、UAP を再起動しなくてもコネクションを再接続できます。詳細については、「[トランザクションマネージャと XDM/RD E2 間のコネクションが切断されたときの再接続方法](#)」を参照してください。

注※3

UAP が tx_open を再発行すれば、UAP を再起動しなくてもコネクションを再接続できます。詳細については、「[トランザクションマネージャと XDM/RD E2 間のコネクションが切断されたときの再接続方法](#)」を参照してください。

(9) トランザクションマネージャに登録する情報

XDM/RD E2 をリソースマネージャとしてトランザクションマネージャに登録する方法については、トランザクションマネージャのマニュアルを参照してください。このとき、次に示す情報をトランザクションマネージャに指定します。

(a) RM スイッチ名

動的登録方法にするか静的登録方法にするかは、RM スイッチ名の指定で決まります。XDM/RD E2 の RM スイッチ名 (xa_switch_t 構造体名) を次に示します。

- 動的登録方法の場合 : pdtxa_switch
- 静的登録方法の場合 : pdtxa_switch_y

(b) RM 名

RM スイッチ (xa_switch_t 構造体) で定義されているリソースマネージャ名は、HiRDB_DB_SERVER です。

(c) オープン文字列

トランザクションマネージャが xa_open でリソースマネージャをオープンするときに使用するオープン文字列は、複数接続機能を使用する場合に指定してください。複数接続機能を使用しない場合はオープン文字列を指定する必要はありません。

複数接続機能を使用する場合は、トランザクションマネージャに複数の XDM/RD E2 を登録し、各 XDM/RD E2 に対してオープン文字列を指定します。オープン文字列には次に示す項目を指定します。

- 接続先で有効にする環境変数を設定したファイルの絶対パス名 (OS が UNIX の場合)
- 接続先で有効にする環境変数をレジストリ登録したときに指定した環境変数グループ名 (OS が Windows の場合)
- 環境変数グループ識別子

OS が UNIX の場合は、"環境変数グループ識別子+環境変数設定ファイル名" または "環境変数グループ識別子*環境変数設定ファイル名" の形式で指定します。

OS が Windows の場合は、"環境変数グループ識別子+環境変数グループ名" または "環境変数グループ識別子*環境変数グループ名" の形式で指定します。

これ以外の形式で指定した場合は、オープン文字列が無視されます。

また、環境変数グループ識別子は 4 バイト固定、オープン文字列は全体で 257 バイト以下で指定してください。

トランザクションマネージャが OpenTP1 の場合のオープン文字列の登録例を次に示します。

トランザクションマネージャがOpenTP1の場合のオープン文字列の登録例

OpenTP1のトランザクションサービス定義のtrnstringオペランドでオープン文字列を登録します。ここでは二つのXDM/RD E2をOpenTP1に登録します。登録条件は次のとおりとします。

- OSがUNIXの場合

リソースマネージャ	環境変数グループ識別子	環境変数設定ファイル名
XDMRD1	XRD1	/usr/conf/XDMRD/XDMRD11.ini /usr/conf/XDMRD/XDMRD12.ini
XDMRD2	XRD2	/usr/conf/XDMRD/XDMRD21.ini /usr/conf/XDMRD/XDMRD22.ini

- OSがWindowsの場合

リソースマネージャ	環境変数グループ識別子	環境変数グループ名
XDMRD1	XRD1	xdmrdrd11 xdmrdrd12
XDMRD2	XRD2	xdmrdrd21 xdmrdrd22

オープン文字列の登録例を次に示します。

- OSがUNIXの場合

```
trnstring -n HiRDB_DB_SERVER -i X1 -o "XRD1*/usr/conf/XDMRD/XDMRD11.ini" -0  
"XRD1+/usr/conf/XDMRD/XDMRD12.ini"  
trnstring -n HiRDB_DB_SERVER -i X2 -o "XRD2*/usr/conf/XDMRD/XDMRD21.ini" -0  
"XRD2+/usr/conf/XDMRD/XDMRD22.ini"
```

- OSがWindowsの場合

```
trnstring -n HiRDB_DB_SERVER -i X1 -o "XRD1*xdmrdrd11" -0 "XRD1+xdmrdrd12"  
trnstring -n HiRDB_DB_SERVER -i X2 -o "XRD2*xdmrdrd21" -0 "XRD2+xdmrdrd22"
```

説明

-n: リソースマネージャ名を指定します。

-i: リソースマネージャ拡張子を指定します。

-o: トランザクションサービス用xa_open関数用文字列を指定します。

OpenTP1のトランザクションサービスプロセスが使用するオープン文字列を指定します。OSがUNIXの場合は"環境変数グループ識別子*環境変数設定ファイル名", OSがWindowsの場合は"環境変数グループ識別子*環境変数グループ名"の形式で指定します。

-O: ユーザサーバ用xa_open関数用文字列を指定します。

ユーザサーバプロセスが使用するオープン文字列を指定します。OSがUNIXの場合は"環境変数グループ識別子+環境変数設定ファイル名", OSがWindowsの場合は"環境変数グループ識別子+環境変数グループ名"の形式で指定します。

- oと-Oには同じ環境変数グループ識別子を指定してください。

- -o と-O に指定するファイルレジストリで設定する環境変数は同じ内容にしてください。

備考

OpenTP1 のユーザサービス定義の tmrmid オペランドで、ユーザサービスから接続する XDM/RD E2 を選択します。XDMRD1 と XDMRD2 に接続する例を次に示します。

```
trnrmid -n HiRDB_DB_SERVER -i X1,X2
```

(d) クローズ文字列

トランザクションマネージャが xa_close でリソースマネージャをクローズするときに使用するクローズ文字列は指定不要です。

(e) RM 関連オブジェクト名

RM 関連オブジェクト名には、次に示すライブラリ名を指定します。

(i) OS が UNIX の場合

ライブラリの種類		ライブラリ名
共用ライブラリ	シングルスレッド用	libzclty.sl (libzcltx.sl)
	シングルスレッド用 (複数接続機能対応)	libzcltys.sl (libzcltxs.sl)
アーカイブ ライブラリ	シングルスレッド用	libcltya.a (libcltxa.a)

注 1

動的登録する場合は、() で囲まれているライブラリ名を指定してもかまいません。

注 2

ライブラリには共用ライブラリとアーカイブルライブラリの両方があります。用途に応じてどちらかを選択してください。ただし、次に示す利点があるため、共用ライブラリを指定することをお勧めします。

- UAP のオブジェクトサイズが小さくなります。
- HiRDB のバージョンアップ時に再リンクエージが不要になります。

注 3

共用ライブラリのサフィックスは、プラットフォームによって異なります。Linux の場合は.so, AIX の場合は.a となります。

(ii) OS が Windows の場合

ライブラリの種類	ライブラリ名
シングルスレッド用	pdcltx32.lib
シングルスレッド用 (複数接続機能対応)	pdcltxs.lib

(f) クライアント環境定義

トランザクションマネージャに XDM/RD E2 のトランザクション処理を制御させるためには、HiRDB のクライアント環境定義をトランザクションマネージャの定義に設定する必要があります。OLTP 環境下でのクライアント環境定義の設定方法については、マニュアル「HiRDB UAP 開発ガイド」を参照してください。

トランザクションマネージャが OpenTP1 の場合、クライアント環境定義のファイル名を次に示す OpenTP1 のシステム定義に、putenv 形式又は dcputenv 形式で定義する必要があります。

- HiRDB クライアント環境定義
- システム環境定義
- ユーザサービスデフォルト定義
- ユーザサービス定義
- トランザクションサービス定義

システム環境定義、ユーザサービスデフォルト定義、ユーザサービス定義、およびトランザクションサービス定義については、マニュアル「OpenTP1 システム定義」を参照してください。

なお、複数の OpenTP1 と接続する場合は、HiRDB クライアント環境定義 HiRDB_PDTMID、または PDTMID を必ず指定してください。

(10) トランザクションマネージャへの登録例

XDM/RD E2 を OpenTP1 に登録するには、OpenTP1 の trnlnkrm コマンドを使用します。trnlnkrm コマンドの指定例を次に示します。

動的登録方法の場合

- OS が UNIX の場合

```
trnlnkrm -a HiRDB_DB_SERVER -s pdtxa_switch -o /HiRDB/client/lib/libzcltx.sl
```

- OS が Windows の場合

```
trnlnkrm -a HiRDB_DB_SERVER -s pdtxa_switch  
-o C:\win32app\hitachi\hirdb_s\client\lib\pdcltx32.lib
```

説明

-a : RM 名を指定します。

-s : RM スイッチ名を指定します。RM スイッチ名は、登録方法（動的登録方法または静的登録方法）によって異なります。

-o : RM 関連オブジェクト名（共用ライブラリのファイル名）を指定します。

静的登録方法の場合

- OS が UNIX の場合

```
trnlnkrm -a HiRDB_DB_SERVER -s pdtxa_switch_y -o /HiRDB/client/lib/libzclty.sl
```

- OS が Windows の場合

```
trnlnkrm -a HiRDB_DB_SERVER -s pdtxa_switch_y
-o C:\win32app\hitachi\hirdb_s\client\lib\pdcltx32.lib
```

説明

- a : RM 名を指定します。
- s : RM スイッチ名を指定します。RM スイッチ名は、登録方法（動的登録方法または静的登録方法）によって異なります。
- o : RM 関連オブジェクト名（共用ライブラリのファイル名）を指定します。

(11) トランザクションマネジャへの登録の変更

トランザクションマネジャへの登録を変更する場合には、次に示す手順に従ってトランザクションマネジャに XDM/RD E2 を登録し直してください。

1. OpenTP1 の trnlnkrm コマンドで、トランザクションマネジャに XDM/RD E2 を登録し直します。
2. OpenTP1 の trnmkobj コマンドで、トランザクション制御用オブジェクトファイルを再作成します。
3. 2.で再作成したトランザクション制御用オブジェクトファイルおよび「[トランザクションマネジャに登録する情報](#)」に示した情報を基に、HiRDB の XA ライブラリとリンクしていたすべての UAP を再リンクageしてください。再リンクageをしないと、UAP の動作は保証されません。

(12) トランザクションマネジャと XDM/RD E2 間のコネクションが切断されたときの再接続方法

(a) UAP で対処する方法

コネクションが切断された場合、実行中の UAP を再起動してください。再起動すると、自動的にコネクションが再接続されます。

UAP を再起動たくない場合は、コネクションが切断されたことを示すエラーが UAP に返ったときに、tx_open 関数を再発行してください。tx_open 関数を再発行することで、UAP を終了しなくても、サービスを続行できます。tx_open 関数を再発行するときのコーディング例を次に示します。

コーディング例

```
int connection = 1;
void service(char *in_data, long *in_len, char *out_data, long *out_len) {
    if (connection == 0) {
        tx_close();
        tx_open();                                .....コネクション切断時のtx_open再発行処理
    }
    tx_begin();
    EXEC SQL INSERT INTO ....;                  .....SQL文発行
    if (SQLCODE == 0) {
        tx_commit();
        *out_data = "OK";
    }
}
```

```

    } else {
        tx_rollback();
        *out_data = "NG";
        if (SQLCODE == -563 || SQLCODE == -722) {
            connection = 0;           .....コネクション切断を記憶
        }
    }
}

```

(b) OpenTP1 の機能を使用する方法

動的登録方法の場合は、OpenTP1 のトランザクションサービス定義またはユーザサービス定義の tm_rm_open_close_scope オペランドに transaction を指定してください。transaction を指定することで、OpenTP1 はトランザクションの開始または終了で XDM/RD E2 とのコネクションを確立または切断します。したがって、途中でコネクションが切断されても、次のトランザクションの開始時にコネクションが再接続されます。

静的登録方法の場合は、トランザクションの開始時に XDM/RD E2 とのコネクションが確立されているかどうかをトランザクションマネージャが確認します。コネクションが切断されている場合は、自動的に再接続されて、トランザクションを開始します。

(13) TP1/Resource Manager Monitor の機能を使用した XDM/RD E2 の監視

XDM/RD E2 を使用する場合、TP1/Resource Manager Monitor の機能を使用した XDM/RD E2 の監視はできません。

3.4.2 OpenTP1 との関連 (XA インタフェースを使用しない場合)

XA インタフェースを使用しない場合、特に制限事項はありません。

なお、HiRDB サーバを使用する場合と同様に、トランザクションの同期点制御は SQL の COMMIT 文を使用してください。OpenTP1 の CBLDCTRN ('C-COMMIT') 関数などの同期点制御関数では XDM/RD E2 の同期点制御はできません。

3.4.3 OpenTP1 以外の他プログラムプロダクトとの関連

HiRDB クライアントを経由して HiRDB サーバに接続できる他プログラムプロダクトが、XDM/RD E2 接続機能を利用できるかどうかは、各プログラムプロダクトが発行する SQL の内容や、内部的に発行する特殊インターフェースの使用有無などによって異なります。

したがって、他プログラムプロダクトが XDM/RD E2 接続機能を使用できるかどうかについては、各プログラムプロダクトの仕様を確認してください。

なお、HiRDB のディクショナリ表を検索するプログラムプロダクトについては、XDM/RD E2 で HiRDB 形式ディクショナリビュー表を作成していなければ、XDM/RD E2 接続機能を使用することはできません。HiRDB 形式ディクショナリビュー表については、マニュアル「VOS3 データマネジメントシステム XDM E2 系 XDM/RD E2 使用の手引－運用編－」を参照してください。

4

クライアントの環境設定

この章では、XDM/RD E2 接続機能を使用する際のクライアントの環境設定について説明します。

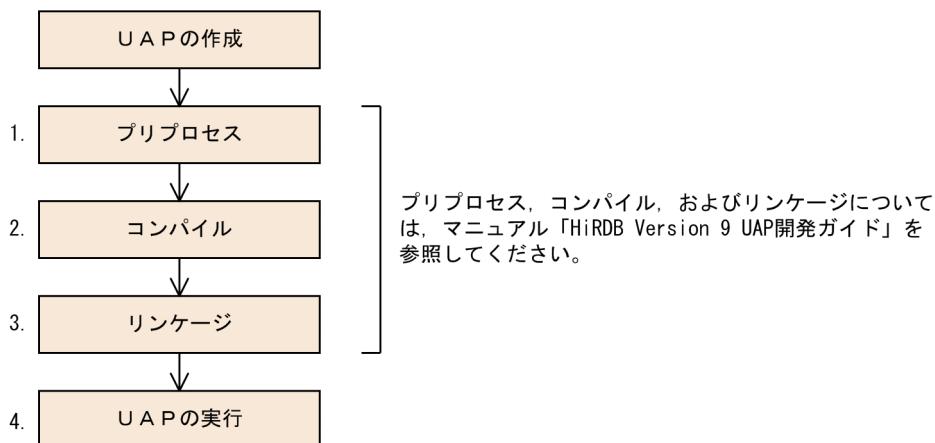
その他のクライアントの環境設定については、マニュアル「HiRDB UAP 開発ガイド」を参照してください。

4.1 HiRDB クライアントの種類

PC, WS とも、HiRDB クライアントには次の 2 種類のプログラムプロダクトがあります。この二つのプログラムプロダクトを、HiRDB クライアントといいます。

- HiRDB/Developer's Kit
- HiRDB/Run Time

HiRDB クライアントの種類によって、UAP の作成から実行までの作業のうち、実行できる作業が異なります。UAP の作成から実行までの作業の流れを次に示します。

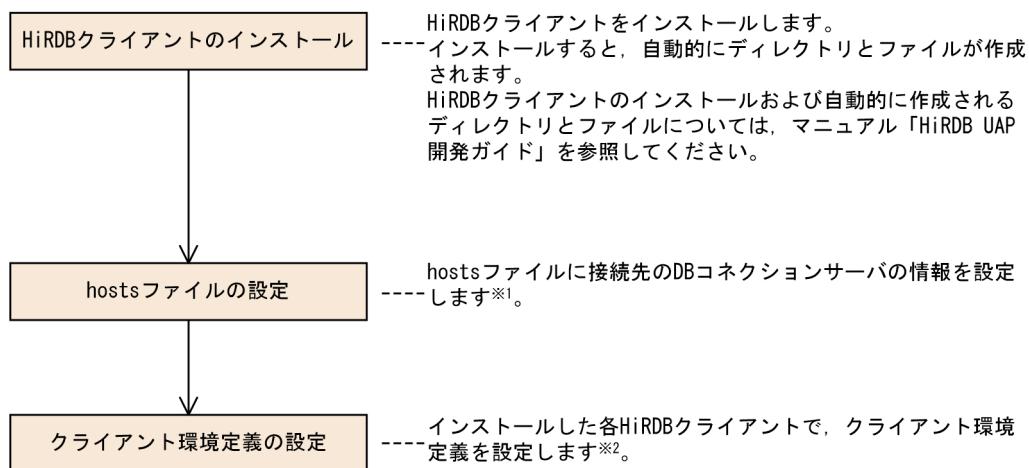


HiRDB クライアントの種類によって実行できる作業は次のとおりです。

- HiRDB/Developer's Kit
項番 1.～項番 4.の作業ができます。
- HiRDB/Run Time
項番 4.の作業ができます。HiRDB クライアントの提供物は HiRDB サーバにも含まれています。したがって、項番 1.～項番 3.は HiRDB サーバの機能で実行してください。

4.2 HiRDB クライアントの環境設定手順

クライアントの環境設定手順を次に示します。



注※1

詳細は、「hosts ファイルの設定」を参照してください。

注※2

クライアント環境定義については、「クライアント環境定義（環境変数の設定）」を参照してください。

4.3 hosts ファイルの設定

HiRDB クライアントから XDM/RD E2 に接続する場合は、クライアントマシンの hosts ファイルに、次に示す情報を指定します。

- 接続する DB コネクションサーバの IP アドレス
- 接続する DB コネクションサーバのホスト名

hosts ファイルとは、Windows では各 TCP/IP が規定する hosts ファイル、UNIX では/etc/hosts ファイルのことです。hosts ファイルにホスト名を指定したあと、ping コマンドを使用して指定した内容が正しいか確認することをお勧めします。

4.4 クライアント環境定義（環境変数の設定）

XDM/RD E2 接続機能を使用するとき、サーバとして HiRDB を使用するか XDM/RD E2 を使用するかによって生じる環境変数の差異と、XDM/RD E2 で設定する環境変数の詳細について説明します。

4.4.1 XDM/RD E2 接続機能使用時の環境変数の差異

XDM/RD E2 接続機能使用時の環境変数の差異を次の表に示します。

なお、各環境変数の詳細は「[クライアント環境定義の設定内容](#)」を参照してください。

表 4-1 XDM/RD E2 接続機能使用時の環境変数の差異

環境変数	内容	Type4 JDBC ドライバ		差異の概要
		使用	不使用	
PDHOST	HiRDB サーバのホスト名 [, 予備系 HiRDB サーバのホスト名]	○	○	XDM/RD E2 をサーバとする場合は、DB コネクションサーバのホスト名または IP アドレスを指定します。
PDNAMEPORT	HiRDB サーバのポート番号	○	○	XDM/RD E2 をサーバとする場合は、DB コネクションサーバのサービスケジュール番号を指定します。
PDFESHOST	フロントエンドサーバのホスト名 [, 予備系フロントエンドサーバのホスト名]	×	×	—
PDSERVICEGRP	サーバ名	×	×	—
PDSRVTYPE	サーバの種別	△	△	XDM/RD E2 をサーバとする場合は、VOS3 を指定します。
PDSERVICEPORT	高速接続用のポート番号 [, 予備系の高速接続用のポート番号]	×	×	—
PDFESGRP	高速接続をする場合の接続するフロントエンドサーバグループ名称	×	×	—
PDCLTRCVPORT	クライアント受信ポート番号指定	×	×	—
PDCLTRCVADDR	クライアント受信 IP アドレス指定	×	×	—

環境変数	内容	Type4 JDBC ドライバ		差異の概要
		使用	不使用	
PDCONTYPE	HiRDB 接続時の接続方式	×	×	—
PDTMID	OLTP 識別子	×	◎	—
PDXAMODE	トランザクションの移行	×	△	XDM/RD E2 をサーバとする場合は、1 は指定できません。
PDTXACANUM	マルチスレッド対応 XA インタフェースを使用した UAP 当たりのトランザクション最大同時実行数	×	×	—
PDXARCVWTIME	トランザクション回復不可時の待ち合わせ時間	×	△	XDM/RD E2 をサーバとする場合は、このオペランドは指定できません。
PDXATRCFILEMODE	X/Open に従った API を使用した接続形態での、各種トレースファイル名の形式指定	×	◎	—
PDXAAUTORECONNECT	TP1/EE と連携する場合に、自動再接続をするかどうかを指定する	×	×	—
HiRDB_PDHOST	X/Open に準拠した API 使用時のデフォルトの PDHOST	×	○	用語の差異。
HiRDB_PDNAMEPOR	X/Open に準拠した API 使用時のデフォルトの PDNAMEPORT	×	○	用語の差異。
HiRDB_PDTMID	X/Open に準拠した API 使用時のデフォルトの PDTMID	×	○	用語の差異。
HiRDB_PDXAMODE	X/Open に準拠した API 使用時のデフォルトの PDXAMODE	×	△	XDM/RD E2 をサーバとする場合は、1 は指定できません。
PDUSER	ユーザ名//パスワード	○	○	XDM/RD E2 をサーバとする場合は、指定できるユーザ名とパスワードの規則は XDM/RD E2 の規則に従います。
PDCLTAPNAME	実行する UAP の識別名称	○	○	用語の差異。
PDCLTLANG	文字コード種別	×	△	XDM/RD E2 をサーバとする場合は、SJIS と UJIS だけ指定できます。

4. クライアントの環境設定

環境変数	内容	Type4 JDBC ドライバ		差異の概要
		使用	不使用	
PDLANG	UAP 実行時の文字コード種別	×	×	—
PDBLOG	UAP 実行時のログの取得方式	×	×	XDM/RD E2 をサーバとする場合に UAP 実行時のログの取得方式を設定するには、DB コネクションサーバのサーバ空間起動制御文の JOURNAL 句でジャーナル取得方法の設定をしてください。
PDEXWARN	拡張警告事象返却の要否	○	○	用語の差異。 SQL コードの差異。
PDSUBSTRLEN	1 文字を表現する最大バイト数を指定	×	×	—
PDCLTCNVMODE	サーバクライアント間での文字コード変換指定	×	×	XDM/RD E2 をサーバとする場合は、XDM/RD E2 が文字コードを判定して自動的に文字コード変換を実行します。
PDCLTGAIJIDLL	ユーザ定義外字変換 DLL ファイル	×	×	XDM/RD E2 をサーバとする場合に外字変換をするには、XDM/RD E2 の RD 環境定義文の CONVERT OPTION 句で外字変換マッピング定義の指定をしてください。
PDCLTGAIJIFUNC	ユーザ定義外字変換関数名	×	×	—
PDCLTCNVUOCLIB	UOC のライブラリファイル名	×	×	—
PDCLTCNVUOCFUNC	UOC の文字コード変換関数名	×	×	—
PDCLTCNVBYTERATIO	UOC による文字コード変換時の、1 文字当たりのバイト数の比率	×	×	—
PDCLTGRP	クライアントグループ名称	×	×	—
PDAUTORECONNECT	自動再接続機能の有無	○	○	—
PDRCCOUNT	自動再接続機能での CONNECT リトライ回数	◎	◎	—
PDRCINTERVAL	自動再接続機能での CONNECT リトライ間隔	◎	◎	—

4. クライアントの環境設定

環境変数	内容	Type4 JDBC ドライバ		差異の概要
		使用	不使用	
PDRCTIMING	自動再接続機能での再接続契機を指定する	◎	◎	—
PDAUTHTYPE	HiRDB クライアントがサーバへ接続するときの認証方式を指定する	×	×	—
PDUAPENVFILE	UAP 環境定義ファイル名称	×	×	—
PDDBUFLRU	UAP がアクセスしたページをグローバルバッファにキャッシュするときの処理に、LRU 方式を適用するかどうかを指定する	×	×	—
PDHATRNQUEUEING	トランザクションキューイング機能の各クライアントの使用有無を指定する	×	×	—
PDCLTBINDLOOPBACKADDR	HiRDB サーバとの通信で使用する受信ポートの生成時、ループバックアドレスで bind() するかどうかを指定する	×	×	—
PDEXTDECHECK	外部 10 進項目の入力データのチェック有無を指定する	◎	◎	—
PDDEFAULTOPTION	クライアント環境定義およびプリプロセスオプションについて、省略時の動作を指定する	◎	◎	—
PDIPCFILEDIR	サーバ接続時に通信情報ファイルを格納するディレクトリ	×	×	—
PDSOCKETCRYPTO	HiRDB クライアントと HiRDB サーバ間の通信を暗号化するかどうかを指定する	×	×	—
PDASTHOST	HiRDB Control Manager - Agent のホスト名	×	×	—
PDASTPORT	HiRDB Control Manager - Agent のポート番号	×	×	—
PDSYSTEMID	HiRDB サーバのシステム ID	×	×	—
PDASTUSER	コマンド実行ユーザ名／パスワード	×	×	—

4. クライアントの環境設定

環境変数	内容	Type4 JDBC ドライバ		差異の概要
		使用	不使用	
PDCMDWAITTIME	コマンド実行時のクライアントの最大待ち時間	×	×	—
PDCMDTRACE	コマンドトレースファイルのサイズ	×	×	—
PDIPC	サーバとクライアントが同一ホストに存在する場合の、プロセス間通信の方法の指定	×	×	—
PDSENDMEMSIZE	クライアント側データ送信用メモリサイズ	×	×	—
PDRECVMEMSIZE	クライアント側データ受信用メモリサイズ	×	×	—
PDCWAITTIME	クライアントの最大待ち時間	○	○	用語の差異。
PDSWAITTIME	サーバの最大待ち時間	×	×	個別には該当する機能はありません。DB コネクションサーバのコントロール空間起動制御文の SVINTERVAL 句で最大サーバ処理間隔監視時間を設定することで代用できます。
PDCWAITTIMEWRNPNT	SQL 実行時間警告出力契機とするクライアントの最大待ち時間に対する比率	×	×	—
PDKALVL	パケット送信機能の使用有無	×	×	—
PDKATIME	パケットの送信間隔	×	×	—
PDTIMEDOUTRETRY	CONNECT システムコールで ETIMEDOUT エラーが発生した場合の CONNECT リトライ回数	×	×	—
PDCONREFRCOUNT	CONNECT システムコールで ECONNREFUSED エラーが発生した場合の CONNECT リトライ回数	×	×	—
PDCONREFRINTERVAL	CONNECT システムコールで ECONNREFUSED エラーが発生した場合の CONNECT リトライ間隔	×	×	—
PDNBLOCKWAITTIME	ノンロックモードでのコネクション確立監視時間	×	△	Windows 版 HiRDB クライアントを使用する場合は指定できません。

4. クライアントの環境設定

環境変数	内容	Type4 JDBC ドライバ		差異の概要
		使用	不使用	
PDCONNECTWAITTIME	HiRDB サーバ接続時の HiRDB クライアントの最大待ち時間	×	×	—
PDCLTPATH	トレースファイル格納ディレクトリ	◎	◎	—
PDSQLTRACE	SQL トレースファイルのサイズ	◎	◎	—
PDUAPERLOG	クライアントエラーログファイルのサイズ	×	◎	—
PDERRSKIPCODE	クライアントエラーログトレースに出力を抑止する SQL コードの設定	×	◎	—
PDPRMTRC	SQL トレース情報へのパラメタ情報の出力有無	◎	◎	—
PDPRMTRCSIZE	SQL トレース情報にパラメタ情報を出力する場合のパラメタ情報の最大データ長	◎	◎	—
PDTRCMODE	トラブルシュート情報出力の設定	×	◎	—
PDUAPREPLVL	UAP 統計レポートの出力情報レベル	△	△	XDM/RD E2 をサーバとする場合は、SQL トレース情報だけが出力の対象となります。
PDREPPATH	統計レポートの格納ディレクトリ	○	○	XDM/RD E2 をサーバとする場合は、SQL トレース情報だけが出力の対象となります。
PDTRCPATH	動的 SQL トレースファイルの格納ディレクトリ	◎	◎	—
PDSQLTRCOPENMODE	SQL トレースファイルのオープン設定	◎	◎	—
PDSQLTEXTSIZE	SQL トレースに出力する SQL 文のサイズ	◎	◎	—
PDSQLEXECTIME	SQL トレースに出力する SQL 実行時間の出力有無	×	◎	—
PDRCTRACE	UAP の再接続トレースを出力するファイルのサイズ	◎	×	—

4. クライアントの環境設定

環境変数	内容	Type4 JDBC ドライバ		差異の概要
		使用	不使用	
PDWRTLNPATH	WRITE LINE 文の値式の値を出力するファイルの格納先ディレクトリ	◎	◎	XDM/RD E2 のバージョンが 11-00 以降の場合に有効となります。
PDWRTLNFILSZ	WRITE LINE 文の値式の値を出力するファイルの最大サイズ	○	○	XDM/RD E2 のバージョンが 11-00 以降の場合に有効となります。
PDWRTLNCOMSZ	クライアントから実行要求した SQL で出力される WRITE LINE 文の値式の値の合計長	◎	◎	XDM/RD E2 のバージョンが 11-00 以降の場合に有効となります。
PDUAPEXERLOGUSE	拡張 SQL エラー情報出力機能を使用するかどうかを指定する	×	×	—
PDUAPEXERLOGPRMSZ	パラメタ情報の最大データ長	×	×	—
PDDNDPTRACE	ADO.NET 2.0 に対応した HiRDB データプロバイダ for .NET Framework で出力するメソッドトレースのファイルサイズ	×	×	—
PDSQLTRCFMT	SQL トレースの出力形式を指定する	◎	◎	—
PDSQLTRCTYPE	SQL トレース情報の出力タイミングを指定する	◎	◎	—
PDJDBSQLTRCMEMNUM	Type4 JDBC ドライバで、メモリ上に退避する SQL トレース情報数を指定する	◎	×	—
PDSQLTRCBUFFSIZE	SQL トレース情報の退避領域のサイズを指定する	×	◎	—
PDVWOPTMODE	アクセスパス情報取得モード	×	×	—
PDTAAPINFPATH	HiRDB SQL Tuning Advisor 用アクセスパス情報ファイルの出力先を指定	×	×	—
PDTAAPINFMODE	HiRDB SQL Tuning Advisor 用アクセスパス情報ファイルのファイル名称形式を指定	×	×	—
PDTAAPINFSIZE	HiRDB SQL Tuning Advisor 用アクセスパス情報ファイルの最大サイズを指定	×	×	—

4. クライアントの環境設定

環境変数	内容	Type4 JDBC ドライバ		差異の概要
		使用	不使用	
PDSTJTRNOUT	UAP に関する統計情報のトランザクションごとの出力有無	×	×	—
PDLOCKLIMIT	ユーザ当たりの最大排他資源要求数	×	×	XDM/RD E2 をサーバとする場合に最大排他資源要求数を設定するには、DB コネクションサーバのサーバ空間起動制御文の ENQNUMLIMIT 句で排他資源管理テーブル使用上限値を設定してください。
PDDLKPrio	デッドロックプライオリティ値	×	×	XDM/RD E2 をサーバとする場合にデッドロックプライオリティ値を設定するには、DB コネクションサーバのコントロール空間起動制御文、またはサーバ空間起動制御文の DEAD LOCK PRIORITY 句でデッドロックプライオリティ値を設定してください。
PDLOCKSKIP	無排他条件判定の使用有無	×	×	—
PDFORUPDATEEXLOCK	FOR UPDATE 句の指定時に WITH EXCLUSIVE LOCK を仮定するかどうかを指定する	×	×	—
PDISLLVL	データ保証レベル	×	×	XDM/RD E2 をサーバとする場合にデータ保証レベルを設定するには、DB コネクションサーバのコントロール空間の DBMS オプション制御文、またはサーバ空間起動制御文の ISOLATION LEVEL 句でデータ保証レベルを設定してください。
PDSQLOPTLVL	SQL 最適化オプション	×	×	XDM/RD E2 をサーバとする場合に SQL 最適化オプションを設定するには、DB コネクションサーバのコントロール空間の DBMS オプション制御文、またはサーバ空間起動制御文の OPTIMIZE LEVEL 句で最適化レベルを設定してください。

4. クライアントの環境設定

環境変数	内容	Type4 JDBC ドライバ		差異の概要
		使用	不使用	
PDADDITIONALOPTLVL	SQL 拡張最適化オプション指定	×	×	—
PDHASHTBLSIZE	ハッシュジョイン、副問合せのハッシュ実行適用時のハッシュ表サイズ	×	×	—
PDDFLNVAL	ナル値の既定値設定	×	×	XDM/RD E2 をサーバとする場合にナル値の規定値設定を設定するには、XDM/RD E2 の RD 環境定義文の DEFAULT NULL VALUE OPTION 句でナル値の既定値を設定してください。
PDAGGR	グループ分けのときに発生するグループ数	×	×	—
PDCMMTBFDL	定義系 SQL 実行前の自動コミット要否	×	×	XDM/RD E2 はトランザクション中でも定義系 SQL を実行できます。ただし、定義系 SQL 実行前ではなく、定義系 SQL 実行後のコミットとなります。
PDPRPCRCLS	PREPARE 文実行時のカーソルの自動クローズ機能の有無	◎	◎	XDM/RD E2 のバージョンが 11-00 以降の場合に有効となります。
PDAUTOCONNECT	オートコネクト機能の有無	×	◎	—
PDDLDEAPRPEXE	定義変更のための前処理結果無効化を許可するかどうかを指定する	×	×	—
PDDLDEAPRP	ホールダブルカーソルで使用している表の定義変更を許可するかどうかを指定する	×	×	—
PDLCKWAITTIME	排他要求が待ち状態になってから、解除されるまでを監視する最大時間を指定する	×	×	—
PDCURSORLVL	カーソルの自動オープン・クローズ指定	×	◎	Type4 JDBC ドライバを使用する場合、指定値は常に 2 となります。
PDDELRSVWDFILE	SQL 予約語削除ファイル名	×	×	—
PDCALCMDWAITTIME	CALL COMMAND 文によってコマンドまたはユティ	×	×	—

4. クライアントの環境設定

環境変数	内容	Type4 JDBC ドライバ		差異の概要
		使用	不使用	
	リティを開始してから終了するまでの、HiRDB クライアントの最大待ち時間を指定する			
PDSTANDARDSQLSTATE	SQLSTATE の値を詳細に出力するかどうかを指定する	×	×	—
PDBLK	ロック転送の行数	○	○	用語の差異。
PDBINARYBLKF	定義長が 32,001 以上である BINARY 型選択式を問合せ指定に含む検索で、ロック転送を適用可能にするかどうかを指定する	×	×	—
PDBLKBUFSIZE	ロック転送用クライアント・サーバ間の通信バッファサイズ	×	×	—
PDBINDRETRYCOUNT	UNIX ドメインでの bind システムコールで EADDRINUSE が返却された場合のリトライ回数を指定する	×	×	—
PDBINDRETRYINTERVAL	UNIX ドメインでの bind システムコールで EADDRINUSE が返却された場合のリトライ間隔を指定する	×	×	—
PDDBACCS	インナレプリカグループのアクセス世代番号指定	×	×	—
PDDBORGUAP	更新可能なオンライン再編成を使用する UAP を実行可能にするかどうかを指定する	×	×	—
PDSPACELEVEL	全角空白変換レベル	×	×	XDM/RD E2 をサーバとする場合に空白変換方法を指定するときは、XDM/RD E2 の RD 環境定義文の KEIS CODE SPACE LEVEL 句でデータの変換レベルを指定してください。
PDCLTRDNODE	RD ノード名称	□	□	—

4. クライアントの環境設定

環境変数	内容	Type4 JDBC ドライバ		差異の概要
		使用	不使用	
PDTPI1SERVICE	XDM/RD E2 に OpenTP1 のサービス名称を通知するかどうかを指定する	×	□	—
PDRDCLTCODE	XDM/RD E2 接続機能でのクライアント側の文字コード	×	□	—
PDCNSTRNTNAME	制約名定義の指定位置	×	×	—
PDTMPTBLRDAREA	一時表および一時インデックスを格納する RD エリアの候補を指定する	×	×	—
PDBESCONHOLD	バックエンドサーバ接続保持機能の使用有無	×	×	—
PDBESCONHTI	バックエンドサーバ接続保持有効トランザクション間隔	×	×	—
PDODBSTATCACHE	SQLColumns, SQLStatistics 関数のキャッシュ	×	◎	—
PDODBESCAPE	パターン文字列に対しての ESCAPE 文字の指定有無	×	○	文字コードの差異。 オプション指定方法の差異。
PDGDATAOPT	SQLGetData 関数でデータ取得済みの列に、再度繰り返してデータを取得できるようにする	×	◎	—
PDODBLOCATOR	DB アクセス部品を使用して、 BLOB 型、または BINARY 型の列を検索する場合に、位置付け子機能を使用してデータを分割取得するかどうかを指定する	×	◎	—
PDODBSPLITSIZE	PDODBLOCATOR 指定時の DB アクセス部品での分割取得のサイズ	×	◎	—
PDODBCWRNSKIP	ODBC, OLE DB 接続時のワーニングを回避するかどうかを指定する	×	◎	—
PDJETCOMPATIBLE	ODBC3.5 ドライバ使用時の Microsoft Access 互換モード指定	◎	◎	—

4. クライアントの環境設定

環境変数	内容	Type4 JDBC ドライバ	差異の概要	
		使用	不使用	
PDODBGINFOSUPPRESS	ODBC 関数の SQLGetInfoW()関数使用時に発生するエラーを回避するために指定する	×	◎	—
PDODBSTANDARDARGSIZE	64 ビットモードで動作する ODBC3.5 ドライバについて、一部の ODBC 関数の引数に指定されたポインタが指すバッファを 4 バイト整数として扱うか、8 バイト整数として扱うかを指定する	×	◎	—
PDODBSTANDARDSQLSTATE	ODBC3.5 ドライバから ODBC 規格に準拠した SQLSTATE を返却するかどうかを指定する	×	×	—
PDODBSTANDARDDESCCOL	ODBC3.5 ドライバでの SQLDescribeCol 関数の実行時に、ODBC 規格に準拠した値を返却するかどうかを指定する	×	◎	—
PDODBSTANDARDTYPEINFO	ODBC3.5 ドライバでの SQLGetTypeInfo 関数実行時に、ODBC 規格に準拠した値を返却するかどうかを指定する	×	◎	—
PDPLGIXMK	プラグインインデクス遅延一括作成指定	×	×	—
PDPLUGINNSUB*	詳細については、各プラグインのマニュアルを参照のこと	×	×	—
PDPLGPFSZ	遅延一括作成用インデクス情報ファイルの初期容量	×	×	—
PDPLGPFSZEXP	遅延一括作成用インデクス情報ファイルの増分値	×	×	—
PDHSICOPTIONS	HSIC を使用する際、クライアント・サーバ間の電文に設定する項目を指定する	×	×	—
PDJDBFILEDIR	Type4 JDBC ドライバでの Exception トレースログのログファイルおよび不正電文トレースファイルの出力先	◎	×	—

4. クライアントの環境設定

環境変数	内容	Type4 JDBC ドライバ		差異の概要
		使用	不使用	
PDJDBFILEOUTNUM	Type4 JDBC ドライバでの Exception トレースログのログファイルへの出力数	◎	×	—
PDJDBONMEMNUM	Type4 JDBC ドライバでの Exception トレースログのメモリ内取得情報数	◎	×	—
PDJDBTRACELEVEL	Type4 JDBC ドライバでの Exception トレースログのトレース取得レベル	◎	×	—
PDJDBFILESIZE	Type4 JDBC ドライバでの Exception トレースログのログファイルの最大サイズ	◎	×	—
PDDNDPCOMPATIBLE	ADO.NET 2.0 に対応した HiRDB データプロバイダ for .NET Framework のメソッドおよびプロパティが .NET Framework の規格に準拠した動作とするか、HiRDB 独自の動作とするかを指定する。	×	×	HiRDB データプロバイダ for .NET Framework は XDM/RD E2 接続機能未サポート。
PDODBERRTRC	ODBC3.5 ドライバでのエラーログの出力有無を指定します。	×	◎	—
PDODBERRTRCSIZE	ODBC3.5 ドライバでのエラーログの最大ファイルサイズをバイト単位で指定します。	×	◎	—

(凡例)

◎ :

サーバとして HiRDB を使用する場合と XDM/RD E2 を使用する場合で、仕様の差はありません。

○ :

サーバとして HiRDB を使用する場合と XDM/RD E2 を使用する場合で、機能的な仕様の差はありません。ただし、用語や製品名などに差異があります。

△ :

サーバとして HiRDB を使用する場合と XDM/RD E2 を使用する場合で、指定できる値が異なります。

× :

HiRDB 接続専用の環境変数です。サーバとして XDM/RD E2 を使用する場合は指定しても無視されますが、環境変数の文法チェックは行われます。

□ :

XDM/RD E2 接続専用の環境変数です。サーバとして HiRDB を使用する場合は指定しても無視されますが、環境変数の文法チェックは行われます。

- :

特にありません。

注※

プラグイン用に設定する環境変数です。この環境変数の設定内容については、クライアントライブラリではチェックされません。また、SQL トレースにも情報は出力されません。

4.4.2 クライアント環境定義の設定内容

ここでは、XDM/RD E2 接続機能使用時の環境変数（表「[XDM/RD E2 接続機能使用時の環境変数の差異](#)」の○、○、△、および□の項目）の設定内容について説明します。

(1) システム構成に関する環境変数

システム構成に関する環境変数には、XDM/RD E2 と接続するときに必要な情報を指定します。

PDHOST=DB コネクションサーバのホスト名 [、予備系のホスト名]

～<識別子>

接続する DB コネクションサーバのホスト名を指定します。

また、ホスト名は IP アドレスで指定できます。

IP アドレスは、バイトごとにピリオド (.) で区切られた 10 進数で次の形式で指定します。

IP アドレスの形式：NNN.OOO.PPP.QQQ

通常は最初に指定したホスト名に対して接続します。最初に指定したホスト名に対して接続できなかつた場合に、予備系のホスト名に対して接続します。

PDNAMEPORT=DB コネクションサーバのポート番号

～<符号なし整数>((5001～65535)) 《20000》

接続する DB コネクションサーバのポート番号を指定します。ポート番号には、DB コネクションサーバのコントロール空間起動制御文およびサーバ空間起動制御文の SCHEDULENO オペランドで指定するサーバスケジュール番号を指定します。なお、SCHEDULENO オペランドを省略した場合はサーバスケジュール番号（ポート番号）に 40179 が仮定されています。

PDSRVTYPE=VOS3

サーバの種別を指定します。

XDM/RD E2 をサーバとする場合は、PDSRVTYPE に VOS3 を必ず指定してください。

(2) OLTP 下の X/Open に準拠した API を使用するクライアントの環境変数

PDXAMODE= {0 | 1}

OLTP システムと連携する場合に、トランザクションの移行機能を使用するかどうかを指定します。

0 :

トランザクションの移行機能を使用しません。

1 :

トランザクションの移行機能を使用します。

XDM/RD E2 をサーバとする場合は、1 は指定できません。

PDXARCVWTIME=トランザクション回復不可時の待ち合わせ時間

~<符号なし整数>((0~270)) 《2》 (単位：秒)

XDM/RD E2 をサーバとする場合、このオペランドは指定できません。

PDXATRCFILEMODE= {LUMP | SEPARATE}

X/Open に従った API を使用した接続形態での、各種トレースファイル名の形式を指定します。X/Open に従った API を使用した接続形態以外の場合、PDXATRCFILEMODE の指定は無視されます。

LUMP :

各種トレースファイル名に実行プロセス ID を付けないで出力します。

UAP が非常駐で何回も実行されて、プロセス ID がそのつど変わる場合には、LUMP を指定することをお勧めします。LUMP を指定した場合、非常駐の UAP を実行するたびに各種トレースファイルが増えて、OS や他プログラムの動作が不安定になることを防げます。

なお、LUMP を指定した場合、トレース情報の出力先が限定されるため、トレース出力サイズを大きくする必要があります。また、トレース出力時に別プロセスの出力と競合するため、処理時間が長くなることがあります。

SEPARATE :

各種トレースファイル名に実行プロセス ID を付けて出力します。

UAP が常駐している場合は、SEPARATE を指定することをお勧めします。

HiRDB_PDHOST=DB コネクションサーバのホスト名 [, 予備系のホスト名]

~<識別子>

このオペランドに指定した値が、環境変数 PDHOST の設定値に置き換わります。指定方法については、PDHOST を参照してください。

HiRDB_PDNAMESPACE=DB コネクションサーバのポート番号

~<符号なし整数>((5001~65535)) 《20000》

このオペランドに指定した値が、環境変数 PDNAMEPORT の設定値に置き換わります。指定方法については、PDNAMEPORT を参照してください。

HiRDB_PDTMID=OLTP 識別子

~<識別子>((4 文字))

複数の OLTP から X/Open に準拠した API を使用して一つの XDM/RD E2 システムをアクセスする場合、それぞれの OLTP にユニークな識別子を指定してください。この環境変数に指定した値が、PDTMID の設定値に置き換わります。

なお、この環境変数の指定で次に示す条件のどちらかに該当する場合、どの OLTP からのトランザクションであるかが識別されないため、OLTP 内でシステムダウンやトランザクション異常が発生すると、トランザクション決着の同期が合わなくなります。

- 複数の OLTP からアクセスする運用形態で、この環境変数を省略し、PDTMID の指定も省略した場合
- 複数の OLTP からアクセスする運用形態で、OLTP ごとに指定する識別子がユニークでない場合

HiRDB_PDXAMODE= {0 | 1}

このオペランドに指定した値が、環境変数 PDXAMODE の設定値に置き換わります。指定方法については、PDXAMODE を参照してください。

(3) ユーザ実行環境に関する環境変数

PDUSER=ユーザ名 [/パスワード]

～《パスワードなしのカレントユーザ名》

XDM/RD E2 と接続するときの認可識別子(ユーザ名)とパスワードを指定します。ただし、CONNECT 文で認可識別子(ユーザ名)を指定した場合は、CONNECT 文で指定した値が優先されます。

パスワードの指定が必要ない場合は、パスワードを省略できます。

ユーザ名、パスワードに指定した文字列は、大文字と小文字が区別されます。

Windows 環境の場合、この環境変数は省略できません。ただし、パスワードは省略できます。

OpenTP1 下の UAP をクライアントとする運用形態の場合、「ユーザ名/パスワード」の形式で指定します。

PDCLTAPNAME=実行する UAP の識別名称

～<文字列>((30 文字)) 《unknown》

XDM/RD E2 に対してアクセスする UAP の識別情報 (UAP 識別子) を指定します。これは、どの UAP を実行しているのかを認識するための名称です。

XDM/RD E2 のオン中 UAP トレースで出力される情報や、SQL トレースファイルの情報などの UAP の名称に、この環境変数で指定した名称が表示されます。

PDCLTLANG= {SJIS | UJIS}

プリプロセサが処理する、UAP の記述に使われている文字コード種別を指定します。

XDM/RD E2 をサーバとする場合は、XDM/RD E2 のバージョンが 09-01 以降の時に、SJIS (シフト JIS 漢字) と UJIS (日本語 EUC) が使用できます。それ以外のバージョンの時は、SJIS しか使用できません。

使用できる値以外を設定した場合は、エラーとなります。

Linux は、日本語 EUC しか使えないため、サーバとして XDM/RD E2 を使用する場合は、XDM/RD E2 のバージョンは 09-01 以降にする必要があります。

UAP のプリプロセス時、文字コード種別は次のように判断されます。

PDCLTLANG	クライアントの OS		
	AIX	Linux	Windows
SJIS	Ja_JP	エラー	ja_JP. SJIS
UJIS	ja_JP	ja_JP. eucJP	エラー
設定なし*	Ja_JP	ja_JP. eucJP	ja_JP. SJIS

注※

プリプロセス時、環境変数 LANG に文字コードを設定している場合は、その文字コードが仮定されま
す（ただし、クライアントが Windows の場合を除きます）。

PDLANG= {UTF-8 | SJIS | CHINESE}

UAP 実行時の文字コード種別を指定します。また、HiRDB クライアントが Linux 版で SJIS を使用す
る場合、SJIS を指定します。省略した場合、環境変数 LANG の指定値が仮定されます。

UTF-8 :

文字コード種別に Unicode を指定します。

SJIS :

文字コード種別にシフト JIS 漢字コードを指定します。

CHINESE :

文字コード種別に EUC 中国語漢字コードを指定します。

PDEXWARN= {YES | NO}

XDM/RD E2 からの拡張警告事象の返却を許可するかしないかを指定します。

拡張警告事象とは、+100, +110 以外の正の値の SQLCODE が返却されることを示します。

この環境変数に YES を指定した場合、SQLCODE が 0, +100, または +110 以外をすべてエラー
として処理している UAP（ストアドプロシジャーを含む）は、エラーの判定方法を変更する必要があり
ます。SQL のエラーの判定と処置については、マニュアル「HiRDB UAP 開発ガイド」を参照してく
ださい。

YES :

拡張警告事象の返却を許可します。

NO :

拡張警告事象の返却を許可しません。XDM/RD E2 で +100, +110 以外の正の SQLCODE を検
知した場合、UAP には SQLCODE=0 で返却されます。

このオペランドを省略すると、NO が仮定されます。

したがって、このような UAP の作りでない場合は、PDEXWARN に YES を指定することをお勧めし
ます。

PDAUTORECONNECT= {YES | NO}

自動再接続機能を使用するかどうかを指定します。

再接続する場合の動作は、次に示すクライアント環境定義で設定できます。

- CONNECT リトライ回数 : PDRCCOUNT
- CONNECT リトライ間隔 : PDRCINTERVAL
- 再接続契機 : PDRCTIMING

CONNECT 以外の SQL で自動再接続機能が動作している間も PDCWAITTIME の時間監視に入ります。PDCWAITTIME の値を超えた場合、自動再接続処理を打ち切ります。

自動再接続が失敗した場合は、自動再接続が動作する要因となったエラーをアプリケーションへ返します。

XDM/RD E2 のバージョンが 10-02 以前の場合、XDM/RD E2 接続機能を使用すると、SQL の CONNECT 文のときにだけ自動再接続機能が有効となります。XDM/RD E2 のバージョンが 11-00 以降の場合、SQL の CONNECT 文以外でも自動再接続機能が有効になります。ただし、自動再接続機能が適用できないことがあります。詳細については、マニュアル「VOSS データマネジメントシステム XDM E2 系 XDM/RD E2 使用の手引－運用編－」の「HiRDB クライアント接続機能に関する留意事項」を参照してください。

X/Open に準拠した API を使用してアクセスしているアプリケーションでは指定を無視し、常に NO を仮定します。

YES :

自動再接続機能を使用します。

サーバプロセスダウン、系切り替え、ネットワーク障害などの要因で HiRDB サーバとの接続が切れた場合に、自動的に再接続できます。

オンライン中のシステム構成変更、プログラムメンテナンス機能を使用する場合、YES を設定することで、アプリケーションへエラーを返すことなく処理を続行できます。

NO :

自動再接続機能を使用しません。

PDRCCOUNT=自動再接続機能での CONNECT リトライ回数

~ 〈符号なし整数〉 ((1~200)) 《5》

自動再接続機能で再接続する場合の CONNECT のリトライ回数を指定します。

PDAUTORECONNECT に YES を設定したときだけ有効となります。

PDRCINTERVAL=自動再接続機能での CONNECT リトライ間隔

~ 〈符号なし整数〉 ((1~600)) 《5》 (単位 : 秒)

自動再接続機能で再接続する場合の CONNECT のリトライ間隔を秒単位で指定します。

PDAUTORECONNECT に YES を設定したときだけ有効となります。

PDRCTIMING= {ALL | 再接続契機 [, 再接続契機 [, 再接続契機]]}

自動再接続機能で再接続する場合の再接続契機を指定します。

この環境定義の指定方法については、マニュアル「HiRDB UAP 開発ガイド」を参照してください。

ALL :

すべての契機で自動再接続を適用します。

再接続契機 :

次のどれか 1 つ、 または複数を組み合わせて指定してください。

- CNCT

CONNECT 文を実行した場合で接続障害を検知したときに、 自動再接続を適用します。

- NOTRN

トランザクション未発生で実行した SQL が接続障害を検知した場合に、 自動再接続を適用します。

- TRN

トランザクション発生中に実行した SQL が接続障害でエラーとなった後、 CONNECT 文を実行しないで SQL を実行した場合に、 自動再接続を適用します。

PDDEFAULTOPTION= {RECOM | V0904}

クライアント環境定義およびプリプロセスオプションについて、 省略時の動作を指定します。通常は、 この環境変数を指定する必要はありません。互換モードを適用したい場合に、 指定してください。

バージョンによって、 省略値が異なるクライアント環境定義およびプリプロセスのオプションについては、 マニュアル「HiRDB UAP 開発ガイド」の「バージョン、 リビジョンによるクライアント環境定義及びプリプロセスオプションの変更点」を参照してください。

RECOM :

推奨モードを適用します。

V0904 :

0904 互換モードを適用します。

(4) システム監視に関する環境変数

PDCWAITTIME=クライアントの最大待ち時間

~<符号なし整数>((0~65535)) 《0》 (単位 : 秒)

HiRDB クライアントから XDM/RD E2 へ要求をしてから応答があるまでの HiRDB クライアントの最大待ち時間を、 秒単位で指定します。

0 を指定した場合、 HiRDB クライアントは XDM/RD E2 からの応答があるまで待ち続けます。

留意点

最大待ち時間を経過しても XDM/RD E2 から応答がない場合は、 UAP にエラーリターンします。このとき XDM/RD E2 との接続は切り離され、 トランザクション処理中の場合はロールバックされます。

(5) HiRDB の通信処理に関する環境変数

PDNBLOCKWAITTIME=ノンブロックモードでのコネクション確立監視時間

～<符号なし整数>((0～120)) 《0》(単位:秒)

XDM/RD E2 と HiRDB クライアント間でコネクション接続完了を監視する場合、ノンブロックモード時のコネクション確立監視時間を指定します。

XDM/RD E2 をサーバとするときは、UNIX 版 HiRDB クライアントを使用時だけ指定できます。

Windows 版 HiRDB クライアント使用時は、指定しても無効となります。

この環境変数に 1 以上を指定すると、XDM/RD E2 と HiRDB クライアント間の通信をノンブロック通信にして、connect()システムコールの終了を監視します。これをノンブロックモードといいます。0 を指定したときは、OS のタイムアウト時間までコネクション接続完了を待ちます。これをブロックモードといいます。

この環境変数の適用基準および指定値の見積もり方法については、マニュアル「HiRDB UAP 開発ガイド」を参照してください。

(6) トラブルシュートに関する環境変数

PDCLTPATH=トレースファイル格納ディレクトリ

～<パス名> 《カレントディレクトリのパス名》

HiRDB クライアントが作成する SQL トレースファイルおよびクライアントエラーログファイルの格納先ディレクトリを指定します。

PDSQLTRACE=SQL トレースファイルのサイズ

～<符号なし整数>((0, 4096～2000000000)) (単位:バイト)

UAP の SQL トレースを出力する SQL トレースファイルのサイズを、バイト単位で指定します。

0 を指定した場合はファイルの最大サイズとなり、最大サイズを超えると SQL トレースは出力されません。4,096～2,000,000,000 を指定した場合は指定値のサイズとなり、指定値のサイズを超えると出力先が切り替わります。省略した場合は、SQL トレースを出力しません。

SQL トレース機能については、「[SQL トレース機能](#)」を参照してください。

ほかの環境変数との関係

SQL トレースは、PDCLTPATH で指定したディレクトリに出力されます。PDCLTPATH の指定がない場合、UAP を起動したときのカレントディレクトリ（OpenTP1 から起動される UAP の場合、OS が UNIX のときは\$DCDIR/tmp/home/サーバ名 xx のディレクトリ、OS が Windows のときは%DCDIR%\tmp\home\サーバ名 xx のディレクトリ）の下に出力されます。

見積もり方法

SQL トレースファイルのサイズは、取得したい SQL 文の数から求めてください。取得したい SQL 文の、それぞれの「1 行 (80 バイト) + SQL 文のサイズ」を求めて、すべてを合計した値を指定値の目安としてください。

PDUAPERLOG=クライアントエラーログファイルのサイズ

～<符号なし整数>((0, 4096～2000000000)) 《4096》(単位: バイト)

UAP のエラーログを出力するクライアントエラーログファイルのサイズを、 バイト単位で指定します。0 を指定した場合はファイルの最大サイズとなり、 最大サイズを超えるとクライアントエラーログは出力されません。4,096～2,000,000,000 を指定した場合は指定値のサイズとなり、 指定値のサイズを超えると出力先が切り替わります。

省略時のファイルサイズは 4,096 となります。PDUAPEXERLOGUSE に YES を指定している場合、 またはシステム定義の pd_uap_exerre_log_use オペランドに YES を指定している場合、 ファイルサイズは 65,536 となります。

クライアントエラーログ機能については、 マニュアル「HiRDB UAP 開発ガイド」を参照してください。

ほかの環境変数との関係

クライアントエラーログは、 PDCLTPATH で指定したディレクトリに出力されます。PDCLTPATH の指定がない場合、 UAP を起動したときのカレントディレクトリ (OpenTP1 から起動される UAP の場合、 OS が UNIX のときは\$DCDIR/tmp/home/サーバ名 xx のディレクトリ、 OS が Windows のときは%DCDIR%\tmp\home\サーバ名 xx のディレクトリ) の下に出力されます。

PDERRSKIPCODE=SQLCODE [, SQLCODE] ...

クライアントエラーログへのメッセージ出力を抑止する SQLCODE を指定します。SQLCODE は最大 10 個指定できます。

例えば、 SQLCODE-901 と-917 を抑止する場合は、 次のように指定します。

```
PDERRSKIPCODE=-901,-917
```

利点

UAP の構造によっては、 SQL の処理で必然的にエラーが発生することがあります。通常の業務でこのようなエラーが頻繁に発生すると、 ファイルシステムを圧迫してしまうおそれがあります。特に、 X/Open に従った API を使用する UAP の場合、 UAP ごとにクライアントエラーログファイルを二つ作成してしまいます。この環境変数を指定すると、 特定のエラーについてはメッセージの出力を抑止できるので、 ファイルシステムへの負荷を軽減できます。

適用基準

次の条件をすべて満たす場合に適用してください。

- UAP の構造上エラーが頻繁に発生する。
- あらかじめ障害の原因が特定できて、 その要因を調査する必要がない。

この環境変数を指定すると、 不慮の障害が発生したときにその原因を調査できなくなります。適用する場合には注意してください。

PDPRMTRC= {YES | NO | IN | OUT | INOUT}

SQL トレースにパラメタ情報、 および検索データ情報を出力するかどうかを指定します。出力内容については、「[SQL トレース機能](#)」を参照してください。

YES :

SQL トレースに入力用パラメタ情報を出力します。YES を指定した場合、検索データ情報と入力パラメタを出力します。

NO :

SQL トレースにパラメタ情報を出力しません。

IN :

SQL トレースに入力用パラメタ情報を出力します。CALL 文の IN パラメタと INOUT パラメタ※も含みます。

OUT :

SQL トレースに出力用パラメタ情報を、および検索データ情報を出力します。CALL 文の OUT パラメタと INOUT パラメタ※も含みます。

INOUT :

SQL トレースに入力用パラメタ情報を、出力用パラメタ情報を、および検索データ情報を出力します。CALL 文の INOUT パラメタ※は 2 回出力します。

注※

CALL 文の INOUT パラメタの情報は、出力データだけになります。

PDPRMTRCSIZE=SQL トレースに出力するパラメタ情報の最大データ長

～<符号なし整数>((4～32008)) 《256》 (単位：バイト)

SQL トレースに出力するパラメタ情報および検索データの最大データ長を指定します。可変長文字列型データ、BLOB 型データ、BINARY 型データの場合は、文字列長領域もデータ長に含みます。

この指定は、PDPRMTRC に NO 以外を指定した場合だけ、有効になります。

この環境変数の指定値を大きくすると、出力する情報量が増えます。そのため、SQL トレースファイルのサイズ (PDSQLTRACE の指定値) も大きくする必要があります。

PDTRCMODE= {ERR | NONE}

SQL トレース以外のトラブルシュート情報 (pderr*.trc の情報) を出力するかどうかを指定します。

ERR :

pderr*.trc の情報を出力します。

NONE :

pderr*.trc の情報を出力しません。

PDUAPREPLVL= {[s] [u] [p] [r] | a}

s または a を指定します。

この環境変数を指定する場合は、PDREPPATH を指定してください。

s : SQL トレース情報を出力します。

u : 指定しても無視されます。

p : 指定しても無視されます。

r : 指定しても無視されます。

a : s を指定した場合と同様です。

s, u, p および r を組み合わせて指定できます。supr と指定した場合は、a と同じ意味になります。u, p, r, up, ur, pr, および upr を指定した場合、SQL トレース情報は出力されません。s および a を指定した場合だけ SQL トレース情報が出力されます。

PDREPPATH=SQL トレースファイルの格納ディレクトリ ~<パス名>

~<パス名>

PDUAPREPLVL を指定する場合に、SQL トレースファイルを出力するディレクトリを指定します。出力ファイル名については、マニュアル「HiRDB UAP 開発ガイド」の「SQL トレース機能」を参照してください。

PDTRCPATH=動的 SQL トレースファイルの格納ディレクトリ

~<パス名>((最大 256 バイト))

HiRDB クライアントが作成する動的 SQL トレースファイルの格納先ディレクトリを指定します。トレース取得コマンド (pdtrcmgr) で動的 SQL トレースファイルを取得する場合には、この環境変数を必ず指定してください。

ここで指定したディレクトリを pdtrcmgr で指定すると、次回の CONNECT から、指定したディレクトリに SQL トレースファイルが作成されます。

pdtrcmgr については、マニュアル「HiRDB UAP 開発ガイド」を参照してください。ただし、XDM/RD E2 をアクセスする場合、pdtrcmgr の-k オプションは指定しても無視されます。SQL トレースについては、マニュアル「HiRDB UAP 開発ガイド」の「SQL トレース機能」を参照してください。

PDSQLTRCOPENMODE= {CNCT | SQL}

SQL トレースファイルのオープンモードを指定します。

CNCT :

CONNECT, DISCONNECT 単位に SQL トレースファイルをオープン、クローズして、トレース情報を出力します。CNCT を指定した場合、PDSQLTRCOPENMODE に SQL を指定するよりオーバヘッドが削減されるため、SQL トレースを出力するための時間が短縮できます。

なお、CNCT を指定した場合、SQL トレースファイルをオープンしたままで情報を書き込むため、正常に DISCONNECT できなかったときには、SQL トレース情報が欠落することがあります。

CNCT を指定する場合は、ほかのクライアント環境変数の指定が必要となります。必要となるクライアント環境変数については、マニュアル「HiRDB UAP 開発ガイド」の「クライアント環境定義の設定内容」を参照してください。

SQL :

オペレーション単位 (SQL 単位) に SQL トレースファイルをオープン、クローズして、トレース情報を出力します。

PDSQLTEXTSIZE=SQL 文のサイズ

~<符号なし整数>((4096~2000000)) 《4096》(単位：バイト)

SQL トレースに出力する SQL 文のサイズを指定します。

アクセスパス取得時に省略した場合、4,096 ではなく、2,000,000 が仮定されます。

PDSQLEXECTIME= {YES | NO}

SQL トレースに SQL 実行時間を出力するかどうかを指定します。

YES :

SQL 実行時間を出力します。

出力される SQL 実行時間の単位はマイクロ秒となります。SQL トレースに出力される値は、実行時間が 24 時間以上のものは正常に出力されません。

NO :

SQL 実行時間を出力しません。

PDWRTLNPATH=WRITE LINE 文の値式の値を出力するファイルの格納先ディレクトリ

～<パス名>((最大 256 バイト))

WRITE LINE 文の値式の値を出力する、ファイルの格納先ディレクトリを指定します。WRITE LINE 文については、マニュアル「HiRDB SQL リファレンス」を参照してください。

この環境変数を省略した場合、PDCLTPATH で指定したディレクトリが仮定されます。

ファイルは、指定したディレクトリ（省略した場合は PDCLTPATH で指定したディレクトリ）に二つ作成されます。作成されるファイルは、X/Open に従がった API (TX_ 関数) の使用有無で異なります。作成されるファイル名を次に示します。

TX_ 関数を使用していない場合：

pdwrtln1.trc, および pdwrtln2.trc

TX_ 関数を使用している場合：

pdwrtlnxxxxx-1.trc, および pdwrtlnxxxxx-2.trc

xxxxx : UAP 実行時のプロセス ID

PDWRTLNFILSZ=WRITE LINE 文の値式の値を出力するファイルの最大サイズ

～<符号なし整数>((0, 4096~2000000000)) (単位：バイト)

WRITE LINE 文の値式の値を出力する、ファイルの最大サイズを指定します。

0 を指定した場合のファイルの最大サイズは、OS で管理できる最大サイズとなります。省略した場合は、WRITE LINE 文の値式の値は出力されません。また、FETCH 文実行時は、WRITE LINE 文の情報は出力されません。

PDWRTLNCOMSZ=クライアントから実行要求した SQL で出力される WRITE LINE 文の値式の値の合計長

～<符号なし整数>((1024~131072)) 《1024》 (単位：バイト)

WRITE LINE 文の値式の値の合計サイズを指定します。

WRITE LINE 文の値式の値の合計サイズが、この環境変数の指定値を超えた場合、超えた分の情報は無視されます。また、この場合、次の行に「**PDWRTLNCOMSZover**」が出力されます。

PDSQLTRCFMT= {1 | 2}

SQL トレースの出力形式を指定します。指定値によって出力ファイル名称が異なります。出力ファイル名称および指定値ごとの出力形式については、マニュアル「HiRDB UAP 開発ガイド」の「SQL トレース機能」を参照してください。

出力形式 2 は、出力形式 1 よりも詳細情報を出力します。このため、通常は指定値を省略するか 2 を指定してください。

1 :

出力形式 1 で出力します。

2 :

出力形式 2 で出力します。

《注意事項》

HiRDB SQL Tuning Advisor 08-03 以前を使用して SQL トレースの解析を行う場合、出力形式 2 には対応していないため 1 を指定してください。

(7) SQL に関する環境変数

PDPRPCRCLS= {YES | NO}

開いているカーソルで使用している SQL 識別子を再度 PREPARE 文で使用する場合、開いているカーソルを自動的にクローズするかどうかを指定します。

この環境変数は、プリプロセス時に-Xe オプションを指定しない場合に有効となります。プリプロセスについては、マニュアル「HiRDB UAP 開発ガイド」を参照してください。

YES : 開いているカーソルを自動的にクローズします。

NO : 開いているカーソルを自動的にクローズしません。

PDAUTOCONNECT= {ON | OFF}

XDM/RD E2 と接続していない状態で SQL 文を実行した場合、自動的に CONNECT するかどうかを指定します。

ON : 自動的に CONNECT してから SQL 文を実行します。

OFF : 自動的に CONNECT しません。このとき SQL 文はエラーとなります。

XDM/RD E2 と接続していない状態で SET SESSION AUTHORIZATION 文を実行した場合、この環境変数の指定に関係なく常にエラーとなります。

UAP を開発する場合は、XDM/RD E2 に正しく CONNECT しているかどうかを判断する必要があるため、この環境変数には OFF を指定することをお勧めします。

(8) ブロック転送機能に関する環境変数

PDBLKF= ブロック転送の行数

~<符号なし整数>((1~4096)) 《1》

XDM/RD E2 から HiRDB クライアントに検索結果を転送するときの、1 回の転送処理で送られる行数を指定します。

オペランドの値を大きく指定すると通信オーバヘッドが減り、検索時間を短縮できますが、その分メモリが余計に必要です。クライアント側に必要なメモリは次に示す計算式で求められます。VOS3 側に必要なメモリはマニュアル「VOS3 データマネジメントシステム XDM E2 系 XDM/RD E2 使用の手

引ーシステム作成・維持編ー」の「HiRDB クライアント接続機能使用時のメモリ所要量」を参照してください。

計算式

$$864 + 16 \times a + (6 \times a + b) \times c \quad (\text{単位: バイト})$$

a : SELECT 句で指定する項目数

b : 1 回の転送処理で出力されるデータ長の総和

c : PDBLKF オペランドの指定値

(9) XDM/RD E2 接続機能専用の環境変数

PDCLTRDNODE=データベース識別子

~<識別子>

接続する XDM/RD E2 のデータベース識別子を指定します。

データベース識別子とは、XDM のサブシステム定義で指定する RD ノード名称を示します。

PDTPI1SERVICE= {YES | NO}

XDM/RD E2 に OpenTP1 のサービス名称を通知するかどうかを指定します。サーバが HiRDB の場合、指定を無視します。

Windows 版 HiRDB クライアントライブラリで cltdll.dll を使用している場合、本機能は使用できません。pdcltm32.dll 等ほかの HiRDB クライアントライブラリと再リンクエージする必要があります。

YES :

OpenTP1 のサービス名称を XDM/RD E2 に通知します。

XDM/RD E2 にサービス名称を通知することで、XDM/RD E2 の統計情報がサービス単位に分析できるようになります。ただし、この場合、XDM/RD E2 09-01 以降が必要となります。

OpenTP1 を使用しない場合、および OpenTP1 のサービスではない場合 (SUP 等) は YES を指定しても XDM/RD E2 にサービス名称は通知されません。

NO :

XDM/RD E2 にサービス名称を通知しません。

PDRDCLTCODE= {SJIS | UTF-8}

この環境変数は、Windows 版クライアントの場合に有効となります。UNIX 版クライアントの場合は指定しても無効となります。

XDM/RD E2 接続機能使用時に、クライアントで使用する文字コード種別を指定します。

SJIS :

シフト JIS 漢字コードを使用します。

UTF-8 :

Unicode (UTF-8) を使用します。UTF-8 を指定する場合、クライアント環境定義 PDCLTCNVMODE には NOUSE を指定するか、または省略してください。

《UTF-8 指定時の規則》

1. 埋込み変数で扱う入出力データ、および? パラメタで扱うデータに UTF-8 を使用できます。
2. UAP で記述する SQL 文には、ASCII コードだけ指定できます。SQL 文中で ASCII コード以外の文字（漢字、半角カタカナ、外字など）を指定する場合、PREPARE 文または EXECUTE IMMEDIATE 文を使用して、埋込み変数で SQL 文を指定してください。
3. XDM/RD E2 から返される、SQL 連絡領域に格納されるエラーメッセージ、列名記述領域に格納される列名、型名記述領域に格納されるデータ型名など、UTF-8 となります。このため、これらの値に ASCII コード以外の文字が含まれている場合、シフト JIS 漢字コードとして出力すると、正しく表示されないことがあります。
4. XDM/RD E2 側で、文字コードを UTF-8 から EBCDIK コードもしくは KEIS コード、または EBCDIK コードもしくは KEIS コードから UTF-8 に変換する場合、データの長さが変化することがあります。このため、埋込み変数の定義長などに注意してください。

(10) ODBC 関数に関する環境変数

PDODBSTATCACHE= {0 | 1}

ODBC 関数の SQLColumns() 関数、SQLStatistics() 関数で、一度発行して取得したカラム情報またはインデクス情報をキャッシュするかどうかを指定します。

0 :

キャッシュしません。

SQLColumns() 関数、SQLStatistics() 関数を呼び出すごとにサーバにアクセスして、カラム情報またはインデクス情報を取得します。

1 :

一度取得したカラム情報およびインデクス情報をキャッシュします。

ただし、サーバと接続中のときはキャッシュのリフレッシュはしません。したがって、接続中にテーブル定義を変更した場合は、実際の定義と異なるカラム情報またはインデクス情報を返すことになるため、一度サーバとの接続を切り離す必要があります。

利点

SQLColumns() 関数、SQLStatistics() 関数が同一のパラメタで呼び出された場合は、キャッシュ上の検索結果を AP に返すことで、サーバとの通信回数を削減できます。

留意点

このオプションの指定が効果的かどうかを判断するには、ODBC のトレースを取得し同一接続中に同じパラメタの SQLColumns() 関数、SQLStatistics() 関数が発行されているかどうかを調査する必要があります。

キャッシュできる行数を次に示します。

SQLColumns() :

約 $60,000 / (50 + \text{表オーナ名長} + \text{表名長} + \text{列名長} + \text{コメント長})$ 行数

SQLStatistics() :

約 60,000/(50 + 表オーナ名長 + 表名長 + インデクス名長 + 列名長)行数文

PDODBESCAPE= {0 | 1}

カタログ系の ODBC 関数(SQLTables(), SQLColumns()など)の検索で、パターン文字に対して ESCAPE 文字('&)を指定するかどうかを指定します。

0 :

パターン文字に対して ESCAPE 文字('&)を指定しません。

1 :

パターン文字に対して ESCAPE 文字('&)を指定します。

各国文字識別子を使用しない環境 (XDM/RD E2 の RD 初期設定ユティリティで KANJINAME=NO を指定するか KANJINAME オペラントを省略) でかつ表名および列名に XDM/RD E2 上のコードで X'50'を含む各国文字 (2 バイトコード) を使用している場合、この指定を 0にしてください。この指定を 1 のまま ODBC 経由でアクセスした場合、特定の表および列が認識されないことがあります。

また、表名などの識別子に下線(_)を使用している場合、この指定を 1にしてください。この指定を 0 で運用した場合、一部の ODBC 対応製品から下線(_)を使用した識別子にアクセスできないことがあります。

留意点

XDM/RD E2 で各国文字識別子を使用しない環境では、表名や列名に 2 バイト文字を使用しないことをお勧めします。

表名や列名に 2 バイト文字を使用する場合は、XDM/RD E2 の RD 初期設定ユティリティまたはデータベース構成変更ユティリティで KANJINAME=YES を指定してください。

PDGDATAOPT= {YES | NO}

ODBC 関数の SQLGetData 関数を使用してデータを取り出す場合、データ取得が完了した列に対して繰り返しデータを取得するときに指定します。

通常、データ取得が完了した列に対して繰り返しデータを取得すると、戻り値として SQL_NO_DATA が返ります。

YES :

SQLGetData 関数でデータ取得が完了した列に対して、繰り返しデータを取得できます。

NO :

SQLGetData 関数でデータ取得が完了した列に対して、繰り返しデータを取得し、戻り値として SQL_NO_DATA が返ります。

適用基準

同じ列に対して複数回データ取得をしたい場合に指定します。例えば、複数回のデータ取得で SQL_SUCCESS を期待する上位 AP などを使用する場合にこの環境変数を指定してください。

PDODBLOCATOR= {YES | NO}

DB アクセス部品を使用して、BLOB 型、または BINARY 型の列を検索する場合に、位置付け子機能を使用してデータを分割取得するかどうかを指定します。DB アクセス部品とは、ODBC ドライバ、OLE DB プロバイダ、および HiRDB データプロバイダ for .NET Framework を示します。

YES :

DB アクセス部品を使用して、BLOB 型、または BINARY 型の列を検索する場合に、位置付け子機能を使用してデータを分割取得します。

NO :

DB アクセス部品を使用して、BLOB 型、または BINARY 型の列を検索する場合に、位置付け子機能を使用しません。

《適用基準》

NO を指定した場合（省略時仮定値の場合）、DB アクセス部品側が列の定義長分のデータ受信領域を確保します。また、HiRDB クライアント側でも列の定義長分のデータ受信領域を必要とします。列の定義長が大きいと、実行時にメモリ不足になったりすることがあるため、メモリ不足になることを回避する場合は YES を指定してください。なお、YES を指定すると、HiRDB サーバとの通信回数が分割取得回数分増えます。

PDODBSPLITSIZE=分割取得サイズ

～<符号なし整数>((4～2097152)) 《100》（単位：キロバイト）

PDODBLOCATOR=YES を指定している場合に、分割取得のサイズを指定します。

《指定値の目安》

実データ長の分布を考慮して、分割取得回数が少なくなり、かつメモリ不足が発生しない値を指定してください。

PDODBCWRNSKIP= {YES | NO}

ODBC、OLE DB 接続時のワーニングを回避するかどうかを指定します。なお、ODBC、OLE DB 接続以外の場合、この環境変数を指定しても無視されます。

YES :

ODBC ドライバ使用時に SQLFetch() の延長で SQLWARN が設定された場合でも、ODBC ドライバで SQLFetch() の戻り値として SQL_SUCCESS を返却します。

OLE DB プロバイダ使用時に SQLWARN が設定された場合でも、エラーオブジェクトの生成をスキップします。

NO :

ODBC ドライバ使用時に SQLFetch() の延長で SQLWARN が設定された場合、ODBC ドライバで SQLFetch() の戻り値として SQL_SUCCESS_WITH_INFO を返却します。

OLE DB プロバイダ使用時に SQLWARN が設定された場合、エラーオブジェクトを生成します。

《適用基準》

ODBC ドライバでは、検索処理で HiRDB の SQL 連絡領域の SQLWARN が設定された場合、SQLFetch() の戻り値として SQL_SUCCESS_WITH_INFO を返却します。しかし、ODBC ドライバを呼び出す上位アプリケーション※によっては、SQL_SUCCESS_WITH_INFO で検索処理を打ち切るものがあります。この環境変数に YES を指定すると、検索処理で SQL 連絡領域の SQLWARN が設定された場合でも、SQLFetch() の戻り値を SQL_SUCCESS とすることで、検索処理が続行できるようになります。

OLE DB プロバイダでは、SQL 実行時に HiRDB の SQL 連絡領域の SQLWARN が設定された場合、エラーオブジェクトを生成します。しかし、OLE DB プロバイダを呼び出す上位アプリケーションによっては、エラーオブジェクトの生成に伴い例外が発生する場合があります。この機能を適用すれば、SQL 実行時に警告が発生した場合でもエラーオブジェクトの生成をスキップすることで、上位アプリケーションによる例外の発生を回避できるようになります。

注※

例えば、ADO.NET を使用して ODBC 経由で HiRDB に接続している場合、
SQL_SUCCESS_WITH_INFO で検索処理が打ち切られることがあります。

PDJETCOMPATIBLE= {YES | NO}

ODBC3.5 ドライバを、ODBC3.5 の規格ではなく Microsoft Access 互換モードで動作させるかどうかを指定します。

YES :

ODBC3.5 ドライバは、Microsoft Access 互換モードで動作します。

NO :

ODBC3.5 ドライバは、ODBC3.5 の規格どおりに動作します。

《適用基準》

Microsoft Access を使用してデータベースにアクセスする場合に指定します。この環境変数を指定しないと、検索結果が "#Delete" と表示されたり、挿入したデータが不正に変換されることがあります。なお、Microsoft 社が提供するほかの製品やインターフェースでこのような現象が発生した場合、この環境変数を指定することで回避することができます。

PDODBERRTRC= {YES | NO}

ODBC3.5 ドライバ内でエラーを検知した場合に、ODBC エラーログを出力するかどうかを指定します。ODBC エラーログの詳細については、マニュアル「HiRDB UAP 開発ガイド」の「ODBC エラーログ」を参照してください。

YES :

ODBC エラーログを出力します。

NO :

ODBC エラーログを出力しません。

《ほかの環境変数との関係》

ODBC エラーログは、PDCLTPATH で指定したディレクトリに出力されます。PDCLTPATH の指定がない場合、UAP を起動したときのカレントディレクトリの下に出力されます。

ODBC エラーログの最大ファイルサイズはクライアント環境定義 ODBCERRTRCSIZE で指定できます。

PDODBERRTRCSIZE= {ODBC エラーログファイルのサイズ}

~<符号なし整数>((65535~2000000000)) 《65535》(単位: バイト)

PDODBERRTRC=YES を指定している場合に ODBC エラーログの最大ファイルサイズを指定します。指定値のサイズを超えると出力先が切り替わります。

5

UAP の作成、実行

この章では、XDM/RD E2 接続機能を使用する際に、UAP の作成、実行段階で考慮する項目について説明します。

基本的な UAP の作成、実行については、マニュアル「HiRDB UAP 開発ガイド」を参照してください。

5.1 UAP の作成

ここでは、UAP の作成に関する仕様の中で、特に HiRDB サーバを使用する場合と XDM/RD E2 を使用する場合で差異がある項目について説明します。

その他の UAP の作成に関する仕様については、マニュアル「HiRDB UAP 開発ガイド」を参照してください。

5.1.1 SQL のデータ型とデータ記述

SQL のデータ型とクライアント UAP の各言語のデータ記述またはデータ型の対応は、HiRDB の仕様に従います。

SQL のデータ型とデータ記述については、マニュアル「HiRDB UAP 開発ガイド」を参照してください。

ただし、XDM/RD E2 接続機能を使用する際の追加項目については、「[SQL のデータ型とデータ記述](#)」を参照してください。

XDM/RD E2 の文字コードを KEIS コードにした場合、データに 2 バイトコードを使用するとデータの長さが変化しますので、変化することを考慮して埋込み変数の長さを定義してください。

例えば、列データの定義が MCHAR(20)である場合に、埋込み変数の定義を 20 バイトにして 2 バイトコードを含むデータの挿入をしようとするとき、文字コード変換後に 20 バイトを超えるので、実行時にエラーになります。

1 バイトコードと 2 バイトコードが混在する場合、文字コード変換によって最大 7/3 倍になるため、埋込み変数(入力変数)の長さは 8 バイト($20 \text{ バイト} \times 3 \div 7 = 8$)で定義しなければなりません。

文字コードに関する注意事項の詳細については、「[文字コードに関する注意事項](#)」を参照してください。

文字コード変換規則の詳細については、マニュアル「VOS3 データマネジメントシステム XDM E2 系 XDM/RD E2 使用の手引－運用編－」を参照してください。

なお、文字コード変換による文字長の変化を意識したくない場合は、表定義時に列のデータ型を文字集合シフト JIS を指定することをお勧めします。

文字集合の詳細については、マニュアル「VOS3 データマネジメントシステム XDM E2 系 XDM/RD E2 SQL リファレンス」を参照してください。

5.1.2 UAP の形態移行

UAP の実行形態として、次の 3 種類が考えられます。

1. HiRDB クライアントを利用して XDM/RD E2 にアクセスする UAP

2. HiRDB クライアントを利用して HiRDB サーバにアクセスする UAP
3. HiRDB クライアントを利用しないで、 XDM/RD E2 内で XDM/RD E2 をアクセスする UAP

これらの UAP のソースは完全に互換性があるわけではないため、 UAP の実行形態を移行する場合には、 UAP の修正が必要になることがあります。

また、異なる OS に移行する場合（項番 3.から項番 1.への移行など）は、 SQL に関する仕様だけではなく、言語の差異もある場合があるので注意が必要です。

移行することを前提に UAP を開発する場合には、移行時の修正量を少なくするために、あらかじめ仕様差のない範囲内で UAP を開発するなどの工夫をお勧めします。

SQL の仕様差については、マニュアル「VOS3 データマネジメントシステム XDM E2 系 XDM/RD E2 SQL リファレンス」を参照してください。

5.2 UAP の実行

ここでは、UAP の実行に関する仕様の中で、特に HiRDB サーバを使用する場合と XDM/RD E2 を使用する場合で差異がある項目について説明します。

その他の UAP の実行に関する仕様については、マニュアル「HiRDB UAP 開発ガイド」を参照してください。

5.2.1 プリプロセス

XDM/RD E2 接続用の UAP をプリプロセスする場合は、SQL プリプロセサ起動時に P オプションを指定します。P オプション以外を指定した場合は、HiRDB の SQL 仕様に従って構文チェックが実施されるため、XDM/RD E2 で使用できる SQL がプリプロセスエラーになる場合があります。

SQL プリプロセサ起動時のオプションの指定を次に示します。

オプション	内容	UNIX 環境		Windows 環境	
		C	COBOL	C	COBOL
S	構文チェックだけをして、ポストソースを出力しない場合に指定します。	○	○	○	○
O	出力するポストソースのファイル名称を指定します。	○	○	○	○
h64	64bit モード用のポストソースを生成することを指定します。	—	—	—	—
P	XDM/RD E2 接続用の UAP をプリプロセスする場合に指定します。このオプションを指定した場合、構文チェックはされません。	○	○	—	—
Xc	SQL プリプロセサが生成する文字列を引用符「"」で囲む場合に指定します。このオプションを省略すると、アポストロフィー「'」で囲みます。	—	○	—	○
XA	Windows 版の XA インタフェースを使用した UAP を作成するときに指定します。	—	—	○	○
Xo	UAP から抽出した SQL 文で標準出力する場合に指定します。 1.SQL 文中の埋込み変数は ? パラメタに置換する。 2.一行検索の INTO 句を削除する。 3.SQL 文中の語句間の空白が 2 文字以上の場合は、空白 1 文字に置換する。 4.複数行に分割して記述されている SQL は 1 行にまとめる。 5.実行時にサーバに送られる SQL だけ出力する。実行されない SQL 文 (WHENEVER 文や BEGIN DECLARE SECTION 文など) は出力しない。 6.SQL の末尾にはセミコロン (;) を追加する。	○	○	○	○

オプション	内容	UNIX 環境		Windows 環境	
		C	COBOL	C	COBOL
	7. 埋込み変数の宣言は出力しない。 8. 動的 SQL は SQL がリテラルで指定されている場合だけ出力する。その他の場合は出力しない。 9. OPEN 文は形式 1 のカーソルの場合だけ、問合せ式を出力する。 10. ポストソースは生成しない。				
A	SQL で認可識別子を省略したときに仮定する認可識別子を変更する場合に指定します。	○	○	○	○
XD	DLL を作成する場合に指定します。	—	—	—	○
Xe {y/n}	PREPARE 時のカーソルの自動 CLOSE 機能の有無を指定します。	○*	○*	○*	○*
Xv	E2 オプションを指定したときに、VARCHAR 型、BINARY 型に対応する構造体を、通常の構造体として解析することを指定します。	○	○	○	○
E {1/2/3} [” オプション 文字列”]	UAP 中で使用されているプリプロセサ宣言文（定数と #ifdef など）の有効化と埋込み SQL 宣言節不要化を指定するオプション。 -E1：プリプロセサ宣言文の有効化を指定する。 -E2：埋込み SQL 宣言節の不要化を指定する。 -E3：-E1 と-E2 を同時に指定する。 オプション文字列： インクルードするファイルを検索するディレクトリパスを、C コンパイラに指定する-I オプション形式で指定する。-D オプションなどの任意のオプションを同時に指定できる（COBOL では指定しても無効）。オプション文字列には複数のオプションを記述する場合は、空白の代わりにセミコロン (；) で区切る。 このオプションを指定するとプリプロセサは内部的に C コンパイラを呼び出すので、PATH 環境変数にコンパイラへのパスを設定しておくことが必要である。	○	○	○	○
Xp	Windows 版プリプロセサで厳密な SQL の構文チェックを行う。	—	—	—	—
Xr	SQL 文の実行時に COBOL の RETURN-CODE 特殊レジスタを更新しないポストソースを生成します。 UAP が、呼び出し元プログラムに返すリターンコードを RETURN-CODE 特殊レジスタに設定してから、呼び出し元プログラムに制御を戻すまでの間に SQL 文を実行する場合、このオプションを指定してください。	—	○	—	○

(凡例)

○：XDM/RD E2 接続機能時に指定できます。

－：該当しません。

注※

XDM/RD E2 のバージョンが 11-00 以降の場合に有効となります。

5.2.2 UAP 実行時の注意事項

UAP 実行時の注意事項を次に示します。

- HiRDB クライアントを用いてコンパイルした UAP の SQL は、すべて動的実行となります。このため、システム全体の性能を向上させるために XDM/RD E2 の SQL プール機能を使用することをお勧めします。
- HiRDB クライアントを用いたプリプロセス時には、最低限の SQL 構文チェックしか実施しません。このため、UAP 実行時にエラーになる場合があります。
- HiRDB クライアントを用いたプリプロセス時は、資源（表や列）の存在チェックを実施しません。このため、UAP 実行時にエラーになる場合があります。
- HiRDB クライアントを用いたプリプロセス時は、XDM のコンパイルオプション（プリプロセスオプション）は使用できません。
- HiRDB サーバを使用する場合と XDM/RD E2 を使用する場合とで、SQL などに仕様の差がなければ、UAP 実行時の環境変数（PDSRVTYPE, PDHOST, PDNAMEPORT など）を変更するだけで、再プリプロセスや再リンクageすることなく、アクセスするサーバを切り替えることができます。
ただし、UNIX 上の UAP で、XDM/RD E2 接続機能サポート以前のバージョンのアーカイブライブラリを使用してリンクageした UAP は、実行時にエラーになります。この場合は、XDM/RD E2 接続機能サポート以降のバージョンのアーカイブライブラリを使用して再リンクageする必要があります。

5.2.3 XDM/RD E2 接続機能を利用するライブラリ

(1) UNIX の場合

クライアント側の OS が UNIX の場合で、XDM/RD E2 接続機能を利用するライブラリを次の表に示します。各ライブラリの用途については、マニュアル「HiRDB UAP 開発ガイド」を参照してください。

表 5-1 XDM/RD E2 接続機能をサポートするライブラリ (UNIX)

名称	ディレクトリ	ファイル名	プラットフォーム	
			AIX	Linux
アーカイブ ファイル	/HiRDB/client/lib/	libclt.a	○	○
		libclt64.a	▲	×
		libcltxa.a	○	○

名称	ディレクトリ	ファイル名	プラットフォーム	
			AIX	Linux
		libcltya.a	○	○
		libcltm.a	×	×
		libcltk.a	○	○
		libcltk64.a	▲	×
		libclts.a	○	○
共用ライブラリ※	<u>/HiRDB/client/lib/</u>	libzclt.sl	○	○
		libzclt64.sl	▲	×
		libzcltx.sl	○	○
		libzclty.sl	○	○
		libzcltm.sl	×	×
		libzcltk.sl	○	○
		libzcltk64.sl	▲	×
		libzpdodbc.sl	×	×
		libsqlauxf.sl	▲	▲
		libsqlauxf64.sl	▲	×
		libzcltxk.sl	×	×
		libzcltyk.sl	×	×
		libzclts.sl	○	○
		libzcltxs.sl	○	○
		libzcltyk.sl	○	○

(凡例)

○ : XDM/RD E2 に接続できます。

▲ : HiRDB サーバには接続できますが、XDM/RD E2 には接続できません。

× : HiRDB サーバにも XDM/RD E2 にも接続できません。

注

下線で示す部分は、HiRDB のインストールディレクトリを示します。

注※

共用ライブラリのサフィックスは、プラットフォームによって異なります。Linux の場合は.so、AIX の場合は.a となります。

(2) Windows の場合

クライアント側の OS が Windows の場合で、XDM/RD E2 接続機能を利用できるライブラリを次の表に示します。各ライブラリの用途については、マニュアル「HiRDB UAP 開発ガイド」を参照してください。

表 5-2 XDM/RD E2 接続機能をサポートするライブラリ (Windows)

名称	ディレクトリ	ファイル名	使用可否
アーカイブファイル	xxxx\\$LIB	CLTDLL.LIB	○
		PDCLTM32.LIB	○
		PDCLTM50.LIB	○
		PDCLTX32.LIB	○
		PDCLTXM.LIB	▲
		PDSQLAUXF.LIB	▲
		PDCLTXS.LIB	○

(凡例)

○ : XDM/RD E2 に接続できます。

▲ : HiRDB サーバには接続できますが、XDM/RD E2 には接続できません。

注

xxxx は HiRDB のインストールディレクトリ名を示します。ディレクトリ名はインストール時に指定できます。

5.3 ODBC 対応アプリケーションプログラムからの XDM/RD E2 アクセス

HiRDB の ODBC ドライバを使用する場合は、XDM/RD E2 で HiRDB 形式ディクショナリビュー表を作成する必要があります。

HiRDB 形式ディクショナリビュー表については、マニュアル「VOS3 データマネジメントシステム XDM E2 系 XDM/RD E2 使用の手引－運用編－」を参照してください。

HiRDB の ODBC ドライバから XDM/RD E2 の TIME 型の列にアクセスする際、入出力パラメタまたは検索結果に小数秒精度が含まれる場合のデータ処理について次に示します。

- 入力パラメタに小数秒精度が含まれる場合

小数秒精度を切り捨てた形式のデータを渡します。XDM/RD E2 では、受け取ったデータを列定義の小数秒精度のけた数分 0 で置き換え、その値をデータとして扱います。

- 出力パラメタおよび検索結果に小数秒精度が含まれる場合

小数秒精度を切り捨てた形式のデータを UAP に返却します。

5.4 JDBC 対応アプリケーションプログラムからの XDM/RD E2 アクセス

HiRDB の JDBC ドライバを使用する場合は、XDM/RD E2 で HiRDB 形式ディクショナリビュー表を作成する必要があります。

HiRDB 形式ディクショナリビュー表については、マニュアル「VOS3 データマネジメントシステム XDM E2 系 XDM/RD E2 使用の手引－運用編－」を参照してください。

HiRDB の JDBC ドライバから XDM/RD E2 の TIME 型の列にアクセスする際、入出力パラメタまたは検索結果に小数秒精度が含まれる場合のデータ処理について次に示します。

- 入力パラメタに小数秒精度が含まれる場合

小数秒精度を切り捨てた形式のデータを渡します。XDM/RD E2 では、受け取ったデータを列定義の小数秒精度のけた数分 0 で置き換え、その値をデータとして扱います。

- 出力パラメタおよび検索結果に小数秒精度が含まれる場合

小数秒精度を切り捨てた形式のデータを UAP に返却します。

5.4.1 Type4 JDBC ドライバを使用した XDM/RD E2 アクセス

Type4 JDBC ドライバを使用して XDM/RD E2 にアクセスする場合について説明します。

(1) 前提バージョン

Type4 JDBC ドライバを使用してアクセスできる XDM/RD E2 のバージョンは、11-03 以降です。

(2) クライアント環境変数

クライアント環境変数の差異については、「[XDM/RD E2 接続機能使用時の環境変数の差異](#)」を参照してください。

(3) 位置付け子 (locator) 機能

Type4 JDBC ドライバを使用して XDM/RD E2 にアクセスする場合、位置付け子機能は使用できません。次に示すインターフェースの指定値に LOCATOR が指定された場合、REAL が指定されたものとして動作します。

- DriverManager クラスの getConnection メソッドに指定するユーザプロパティ LONGVARBINARY_ACCESS
- DataSource クラスの setLONGVARBINARY_Access メソッド
- ConnectionPoolDataSource クラスの setLONGVARBINARY_Access メソッド
- XADataSource クラスの setLONGVARBINARY_Access メソッド

各インターフェースについては、マニュアル「HiRDB UAP 開発ガイド」を参照してください。

(4) 分散トランザクション

分散トランザクション機能を使用した場合、XDM/RD E2 にはアクセスできません。XADataSource クラスの getXAConnection メソッドを使用してアクセスしようとした場合、SQLException が投入されて KFPJ80003-E メッセージが出力されます。分散トランザクション機能については、マニュアル「HiRDB UAP 開発ガイド」を参照してください。

(5) サポートしないドライバ

次に示すドライバを使用した XDM/RD E2 へのアクセスはサポートしていません。

- Type2 JDBC ドライバ
- HiRDB データプロバイダ for .NET Framework
- HiRDB OLEDB データプロバイダ
- SQLJ ランタイムライブラリ

5.5 UAP の障害対策

HiRDB にアクセスする場合と XDM/RD E2 にアクセスする場合の UAP の障害対策の差異について説明します。

UAP の障害対策の詳細については、マニュアル「HiRDB UAP 開発ガイド」を参照してください。

また、DB コネクションサーバの障害については、マニュアル「VOS3 Database Connection Server」を参照してください。

5.5.1 SQL トレース機能

実行した UAP の SQL トレース情報を SQL トレースファイルに取得します。

UAP 実行時に SQL エラーが発生した場合、SQL トレース情報を参照すると、エラーの原因となる SQL 文を特定できます。

HiRDB をアクセスする場合と XDM/RD E2 をアクセスする場合の SQL トレース情報の内容の差異を次の表に示します。

なお、説明中の項番は、マニュアル「HiRDB UAP 開発ガイド」の SQL トレース情報の見方に関する説明の番号に対応しています。

表 5-3 HiRDB をアクセスする場合と XDM/RD E2 をアクセスする場合の SQL トレース情報の内容の差異

項目番号	項目	HiRDB をアクセスする場合と XDM/RD E2 をアクセスする場合の SQL トレース情報の内容の差異
1	UAP 名称	差異なし。
2	認可識別子	差異なし。
3	UAP 開始時刻	差異なし。
4	UAP 実行環境	差異なし。
5	UAP 実行ステータス	CURHOST : 出力されない。 CURPORT : 常に 0。 SRVNAME : JOB 名称。ただし、DB コネクションサーバのバージョンが 05-00 より古い場合は出力されない。 CNCTNO : コネクト通番。ただし、DB コネクションサーバのバージョンが 05-00 より古い場合は 0。 SVRID : サーバユーティリティ番号。ただし、DB コネクションサーバのバージョンが 05-00 より古い場合は 0。 CLTPID : 差異なし。 CLTTID : 差異なし。

項目番	項目	HiRDB をアクセスする場合と XDM/RD E2 をアクセスする場合の SQL トレース情報の内容の差異
6	コネクト通番	コネクト通番を表示する。ただし、DB コネクションサーバのバージョンが 05-00 より古い場合は 0。
7	UAP のプロセス番号	差異なし。
8	UAP のスレッド番号	差異なし。
9	SQL カウンタ	差異なし。
10	オペレーションコード	差異なし。 オペレーションコードの詳細については、マニュアル「HiRDB UAP 開発ガイド」を参照してください。
11	セクション番号	差異なし。
12	SQLCODE	差異なし。
13	SQLWARN	差異なし。
14	SQL 実行要求受付時刻	差異なし。
15	SQL 実行要求終了時刻	差異なし。
16	システムが使用する情報	差異なし。
17	SQL 文	差異なし。
18	ユーザ識別子を変更した場合の情報	差異なし。
19	UAP が動作しているプラットフォーム	差異なし。
20	ライブラリの作成日付	差異なし。
21	パラメタトレース	差異なし。
22	リンクしたライブラリの名称	差異なし。
23	SQL 実行時間	差異なし。
24	SQL トレース出力形式	差異なし。
25	HiRDB クライアント作成バージョン	差異なし。
26	HiRDB.ini ファイルパス	差異なし。
27	サーバとの接続処理で使用したソケット情報	サーバとの接続処理で使用したソケット情報を表示する。 XDM/RD E2 接続時は次の情報を表示する。 日時：クライアント～サーバ間で接続処理をした日時 CLT：クライアント側の IP アドレス、ポート番号 DBS：サーバ側の IP アドレス、ポート番号
28	SQL トレース出力処理時間	差異なし。
29	プリプロセスファイル名	差異なし。
30	プリプロセス時間	差異なし。

5. UAP の作成、実行

項目番	項目	HiRDB をアクセスする場合と XDM/RD E2 をアクセスする場合の SQL トレース情報の内容の差異
31	ルートアプリケーション情報	差異なし。
32	SQL 文長	差異なし。
33	トランザクション識別子	差異なし。
34	拡張文名	XDM/RD E2 では拡張文名未サポートのため表示しない。
35	拡張カーソル名	XDM/RD E2 では拡張カーソル名未サポートのため表示しない。
36	前処理結果保持機能指定情報	差異なし。
37	ホールダブルカーソル指定情報	差異なし。
38	自動再接時にサーバとの接続処理で使用したソケット情報	サーバとの接続処理で使用したソケット情報を表示する。 XDM/RD E2 接続時は次の情報を表示する。 日時：クライアント～サーバ間で接続処理をした日時 CLT：クライアント側の IP アドレス、ポート番号 DBS：サーバ側の IP アドレス、ポート番号

参考

SQL トレースのヘッダ部に表示されるサーバのバージョン番号は、接続している XDM/RD E2 のバージョンに 20 を加算した値が表示されます。例えば、XDM/RD E2 のバージョンが 11-00 の場合は 31-00 となります。出力例を次に示します。

```
** UAP TRACE (CLT:07-02(Nov 17 2004) SVR:31-00) WIN32(WIN_32)**
```

5.5.2 SQL トレース動的取得機能

SQL トレース動的取得機能は使用できますが、XDM/RD E2 をアクセスする場合、トレース取得コマンド (pdtrcmgr) の-k オプションは指定しても無視されます。

6

SQL 言語仕様

この章では、XDM/RD E2 下の UAP から使用する場合の SQL 言語仕様と、HiRDB クライアントから XDM/RD E2 を使用する場合の SQL 言語仕様の差異について説明します。

SQL の言語仕様の詳細については、マニュアル「VOS3 データマネジメントシステム XDM E2 系 XDM/RD E2 SQL リファレンス」を参照してください。

6.1 基本項目

ここでは、XDM/RD E2 で使用できる SQL の記述形式、および HiRDB クライアントからアクセスできる XDM/RD E2 のデータ型について説明します。

6.1.1 SQL の記述形式

(1) 全般規則

サーバとして XDM/RD E2 を使用する場合、使用できる SQL の仕様は XDM/RD E2 が規定する仕様に従います。

XDM/RD E2 の SQL 仕様は、「XDM/RD E2 の SQL 仕様」と「HiRDB プログラムの SQL 仕様」があります。

HiRDB クライアントからのアクセスの場合、通常は「HiRDB プログラムの SQL 仕様」で動作しますが、「XDM/RD E2 の SQL 仕様」で動作したい場合は、XDM/RD E2 の RD 環境定義文の HIRDB PROGRAM 句に SQL SPECIFICATION XDMRD を指定します。

各仕様の違いの詳細については、マニュアル「VOS3 データマネジメントシステム XDM E2 系 XDM/RD E2 SQL リファレンス」を参照してください。また、XDM/RD E2 の RD 環境定義文に関しては、マニュアル「VOS3 データマネジメントシステム XDM E2 系 XDM/RD E2 使用の手引－RD 環境定義文・空間起動制御文・運用コマンド編－」を参照してください。

なお、HiRDB クライアントを使用して XDM/RD E2 をアクセスする形態とは、次の条件を満たす形態をいいます。

- HiRDB クライアントでプリプロセスした埋込み型 UAP
- 会話型 SQL 実行ユーティリティ (pdsql)
- HiRDB の ODBC ドライバを使用した ODBC 接続

HiRDB の ODBC ドライバを使用した ODBC 接続については、「[ODBC 対応アプリケーションプログラムからの XDM/RD E2 アクセス](#)」を参照してください。

- HiRDB の JDBC ドライバを使用した JDBC 接続
- HiRDB の JDBC ドライバを使用した JDBC 接続については、「[JDBC 対応アプリケーションプログラムからの XDM/RD E2 アクセス](#)」を参照してください。
- HiRDB クライアントを経由して XDM/RD E2 に接続するプログラムプロダクト

(2) SQL で使用できる文字

SQL で使用できる文字は、XDM/RD E2 の規則に従います。ただし、クライアント側の文字コードが日本語 EUC の場合、SQL 文中に EUC コードセット 3 の外字 $((8F)_{16}(xxxx)_{16})$ の 3 バイトで表現される文

字コード)を使用することはできません。なお、EUC コードセット 3 の外字を VARCHAR 型などの変数に格納することはできます。

XDM/RD E2 の規則で示されている「EBCDIK コード」および「KEIS コード」は、クライアント側では次のような文字コードに対応します。

XDM/RD E2 側の文字コード	クライアント側の文字コード	
	シフト JIS	日本語 EUC
EBCDIK	JIS8	JIS8※
KEIS	JIS	EUC コードセット 1 または EUC コードセット 3

注※

半角カタカナは EUC コードセット 2 ((8E)₁₆(xx)₁₆ の 2 バイトで表現される文字コード) になります。

SQL 文中に英小文字を指定できます。SQL 文中に指定した英小文字は、XDM/RD E2 の規則に従い、英小文字または英大文字として扱われます。また、HiRDB クライアントを使用して XDM/RD E2 をアクセスする場合、実長が 0 バイトの可変長文字列、可変長混在文字列、または実長が 0 文字の可変長各国文字列を使用できます。

詳細についてはマニュアル「VOS3 データマネジメントシステム XDM E2 系 XDM/RD E2 SQL リファレンス」を参照してください。

(3) 文字コード

XDM/RD E2 接続機能を使用する場合に、クライアント側および XDM/RD E2 側で使用できる文字コードを次の表に示します。

表 6-1 XDM/RD E2 接続機能を使用する場合に、クライアント側および XDM/RD E2 側で使用できる文字コード

実行環境	使用できる文字コード	備考
クライアント側※1	シフト JIS	半角文字は JISX0201 に準拠します。 全角文字は JISX0208 に準拠します。
	日本語 EUC	XDM/RD E2 09-01 以降の場合に使用できます。
	UTF-8※2	XDM/RD E2 10-01 以降の場合に使用できます。
XDM/RD E2 側	EBCDIK/KEIS	表の列の文字集合がシフト JIS の場合は、シフト JIS のままで格納されます。

注※1

JDBC ドライバ使用時は、環境変数 LANG や環境変数 PDLANG の指定値に関係なくシフト JIS となります。

注※2

NCHAR 型、NVARCHAR 型では使用できません。

また、クライアント側 OS が使用できるロケールを次の表に示します。

表 6-2 クライアント側 OS が使用できるロケール

文字コード	ロケール		
	AIX	Linux	Windows
sjis (シフト JIS 漢字)	Ja_JP	—	ja_JP.SJIS
ujis (日本語 EUC) ※1	ja_JP	ja_JP.eucJP, ja_JP※2, ja_JPujis※2	ja_JP.ujis
UTF-8	—	—	UTF-8

(凡例)

— : 使用できません。

注※ 1

XDM/RD E2 09-01 以降の場合だけ使用できます。

注※ 2

ja_JP と ja_JP.ujis は、ja_JP.eucJP の別名とみなされます。

(4) 文字コード変換

HiRDB クライアント側で使用する文字コードと XDM/RD E2 側で使用する文字コードは XDM/RD E2 で変換されます。文字コード変換によって、2 バイトコードを含む文字列は長さが変わることがあるため、注意が必要です。

文字コードに関する注意事項の詳細については、「[文字コードに関する注意事項](#)」を参照してください。

文字コード変換規則の詳細については、マニュアル「VOS3 データマネジメントシステム XDM E2 系 XDM/RD E2 使用の手引－運用編－」を参照してください。

(5) 文字コードに関する注意事項

通常、XDM/RD E2 側で使用できる文字コードは EBCDIK/KEIS であり、クライアント側で使用できる文字コードはシフト JIS、日本語 EUC、および UTF-8 です。

XDM/RD E2 の文字集合機能を使用して文字集合をシフト JIS にすれば、XDM/RD E2 で扱うデータをシフト JIS で扱えますが、文字集合をシフト JIS にしない場合は、文字コードを変換する必要があります。

システムを構築する上で、文字コードを変換するときに注意しなければならない点を次に示します。

(a) データベース中の文字コードを EBCDIK/KEIS にする場合

(i) データ長の変化

KEIS コードは、半角の文字（1 バイトの文字）と全角の文字（2 バイトの文字）を区別するために、シフトコードという制御コードを用います。シフトコードはデータ表示時には表示されませんが、データとして存在するため、見かけの文字長と実際のデータ長が異なります。一方、シフト JIS と日本語 EUC では、シフトコードのような制御コードを持ちません。このため、文字コード変換によってデータの長さが変化する場合があります。

次に例を示します。

なお、ここでは KEIS コードへシフトするシフトコードを[漢]、EBCDIK コードへシフトするシフトコードを[E]と表します。[漢]は 0x0A42、[E]は 0x0A41 という 2 バイトのデータです。

ケース	シフト JIS または日本語 EUC でのデータ	EBCDIK/KEIS でのデータ
半角文字だけの場合 (VARCHAR 型)	'ABCD' (4 バイト)	'ABCD' (4 バイト)
全角文字だけの場合 (NVARCHAR 型)	'あいう' (6 バイト)	'あいう' (6 バイト)
半角文字と全角文字が混在する場合 (MVARCHAR 型)	'A あ B い C' (7 バイト)	'A[漢]あ[E]B[漢]い[E]C' (15 バイト)

半角文字と全角文字が混在するデータの場合、シフト JIS または日本語 EUC のデータを EBCDIK/KEIS に変換すると、最大 7/3 倍になります。

このため、混在文字データを扱う場合のシステムでは次のようにします。

• 表定義時の列の長さ

シフト JIS または日本語 EUC のデータ長の 7/3 倍以上で定義してください。

(例)

UAP で 100 バイトの混在文字データを扱う場合、列の定義は MVARCHAR (100) ではなく、MVARCHAR (234) で定義します。

• UAP で使用する埋込み変数

ある表のデータの追加・更新を XDM/RD E2 接続機能だけを使用して行う場合は、その表にアクセスする SQL の埋込み変数を、その埋込み変数に対応する列の定義長の 3/7 倍の長さで定義してください。

(例 1)

「表定義時の列の長さ」の例の列に対してデータを挿入 (INSERT) する場合の埋込み変数は、MVARCHAR (100) に対応する埋込み変数の定義をしてください。

C 言語の場合は、次のようにになります。

```
struct {
    short 变数名1 ;
    char 变数名2[100] ;
} 構造体名 ;
```

なお、可変長の場合は、実長（変数名 1 で示す長さ）が 100 バイト以下であれば、変数名 2 の長さは 100 バイトを超えていてもかまいません。また、EBCDIK/KEIS に変換しても 234 バイト以下となるデータであれば、実長が 100 バイトを超えていてもかまいません。

しかし、シフト JIS または日本語 EUC で 100 バイトを超えるデータをデータベースに格納すると、検索時に 100 バイトの埋込み変数で受け取れなくなる場合があります。これを防止するため、挿入データに対応する埋込み変数の定義長は、100 バイトにしておくことをお勧めします。

(例 2)

「表定義時の列の長さ」の例の列を検索 (FETCH) する場合の埋込み変数は、MVARCHAR (100) に対応する埋込み変数の定義をしてください。

ただし、これはデータベース中に格納されているすべての EBCDIK/KEIS のデータが、シフト JIS または日本語 EUC に変換しても 100 バイト以下のデータである場合に限ります。

ある表のデータの追加・更新を XDM/RD E2 接続機能以外の方法で行うことがある場合は、その表にアクセスする SQL の埋込み変数の長さは次のとおりにしてください。

対象列	処理	VARCHAR 型変数の長さ	
		シフト JIS の場合	日本語 EUC の場合
MCHAR(n) および VARCHAR(n)	検索	定義長=n	定義長=n×2
	格納	実長≤n×(3/7)*	実長≤n×(3/7)*

注※

追加・更新するデータの EBCDIK/KEIS でのバイト数がわからない場合の見積もりです。バイト数がわかる場合は、EBCDIK/KEIS でのバイト数≤n になるようにしてください。

(例 3)

HiRDB クライアント以外を使用してデータの追加・更新を行うことがある表に VARCHAR (100) の列が存在する場合は、この列を検索 (FETCH) するときの埋込み変数は、3/7 倍にするのではなく、次のようにしてください。

- クライアントの文字コードがシフト JIS の場合

MVARCHAR (100) に対応する埋込み変数の宣言をしてください。これは、既存のデータベース中に 1 バイト文字だけ格納されている場合があるためです。

- クライアントの文字コードが日本語 EUC の場合

MVARCHAR(200)に対応する埋込み変数の宣言をしてください。これは、既存のデータベース中に半角カタカナ文字だけ格納されている場合があるためです。半角カタカナは EBCDIK/KEIS では 1 文字が 1 バイトですが、日本語 EUC では 1 文字が 2 バイトになるので、EBCDIK/KEIS のデータを日本語 EUC に変換すると、最大 2 倍になります。

(ii) 固定長文字に対する考慮

全角文字を含むシフト JIS または日本語 EUC のデータを EBCDIK/KEIS に変換すると、データ長が変化するため、全角文字を扱う場合は、固定長データ (MCHAR) ではなく可変長データ (MVARCHAR) とすることをお勧めします。

固定長データ (MCHAR) とする場合には、次の点に注意してください。

• 文字長の変化

(i) と同様に、挿入 (INSERT) 時などに使用する埋込み変数は、対応するデータの 3/7 倍にします。

代入先のデータ長が代入元より短い場合は、代入先に空白が補われます。

埋込み変数が 3/7 倍以上の場合は、次のような事象が発生します。

(例)

C 言語の埋込み型 UAP であり、表の列の定義が MCHAR (234) の場合で、「INSERT INTO 表 VALUES (:埋込み変数)」としたときの結果を次に示します。

なお、'△'は半角の空白 1 文字を示し、'¥0'は C 言語のナル文字を示します。

埋込み変数の型 (C 言語の場合)	埋込み変数のデータ	結果
char 変数名[235];	'ABCDEFG△△…△¥0' (空白は 228 個)	左記データが正しく格納される。
	'あいうえお△△…△¥0' (空白は 224 個)	SQLCODE=-404 エラー※
char 変数名[101];	'あいうえお△△…△¥0' (空白は 90 個)	左記データが正しく格納される。

注※

'あいうえお△△…△' (空白は 224 個) を EBCDIK/KEIS に変換すると、'[漢]あいうえお[E]△△…△' (空白は 224 個) となり、長さが 238 バイトになります。その結果、挿入するデータ長が列の定義長を超えるため、エラーになります。

• LIKE 述語の比較

XDM/RD E2 の固定長文字列に対する比較述語では、長さが異なるデータを比較する場合は、短い方に空白を補って比較します。しかし、LIKE 述語に関しては空白を補いません。

このため、LIKE 述語でパターン文字 '%' (0 文字以上の任意の文字) を使用しない場合は、比較するときに意図する検索ができない場合があります。

(例)

表の列の型が MCHAR (234) であり、かつ格納されているデータが'ABCDEFG△△…△'と'あいうえお△△…△'である場合で、SELECT 文の探索条件を「WHERE 値式 LIKE :埋込み変数」としたときの結果を次に示します。

なお、'△'は半角の空白 1 文字を示し、'¥0'は C 言語のナル文字を示します。

埋込み変数の型 (C 言語の場合)	埋込み変数のデータ	「WHERE 値式 LIKE :埋込み変数」 の評価
char 変数名[235];	'ABCDEFG△△…△¥0' (空白は 228 個)	真
	'ABCDEFG¥0'	偽※1
	'あいうえお△△…△¥0' (空白は 224 個)	偽
	'あいうえお△△…△¥0' (空白は 220 個)	真※2
char 変数名[101];	'ABCDEFG△△…△¥0' (空白は 93 個)	偽
	'ABCDEFG¥0'	偽
	'あいうえお△△…△¥0' (空白は 90 個)	偽

注※1

UAP が C 言語で記述されている場合、データ中に'¥0'が含まれていると、データ長を'¥0'の一つ前までの長さとみなします。このため、「長さ 7 バイトのデータ'ABCDEFG'」とデータベースの「長さ 234 バイトのデータ'ABCDEFG△△△'」を比較します。

'%'を含まない LIKE 述語は、長さが一致しなければ真とはならないため、この評価は「偽」となります。

注※2

半角文字だけを扱う場合は、埋込み変数に対応する SQL のデータ型のデータ長と同じ長さで埋込み変数を定義して、'%'を含まない LIKE 述語を「真」にできます。

しかし、全角文字が含まれる場合は、EBCDIK/KEIS に変換した後のデータ長を考慮したデータ長としなければなりません。

この例では、[漢]と[E]が付加されて 4 バイト大きくなるため、定義長（234 バイト）よりも 4 バイト少なくなる位置に'¥0'を埋め込みます。

(iii) 空白の扱い

空白には半角の空白と全角の空白があります。XDM/RD E2 には、空白に関する変換規則について次に示すオプションがあります。このオプションは、XDM/RD E2 の RD 環境定義文で指定します。詳細については、マニュアル「VOS3 データマネジメントシステム XDM E2 系 XDM/RD E2 使用の手引－RD 環境定義文・空間起動制御文・運用コマンド編－」を参照してください。

HALFSPACE

XDM/RD E2 の RD 環境定義文の CONVERT OPTION 句に HALFSPACE HALF を指定することをお勧めします。

HALFSPACE HALF を指定すると、半角空白が 2 文字連続した場合の文字コードが次のように変換されます。

変換方向	変換前データ	変換後データ	
		HALFSPACE HALF を指定	HALFSPACE を省略
シフト JIS または日本語 EUC→ EBCDIK/KEIS	半角空白 2 文字 (0x2020)	半角空白 2 文字 (0x4040)	半角空白 2 文字 (0x4040)
EBCDIK/KEIS→ シフト JIS または日本語 EUC	半角空白 2 文字 (0x4040)	半角空白 2 文字 (0x2020)	全角空白 1 文字 (0x8140)

HALFSPACE HALF を指定することで、クライアント側から挿入した半角空白 2 文字を検索する場合、挿入した値と同じ値で検索できます。

(iv) EUC コードセット 3 の外字の扱い

ここでは、クライアントの文字コードが日本語 EUC の場合の、EUC コードセット 3 の外字の扱いについて説明します。

EUC コードセット 3 の外字を含むデータを、NCHAR および NVARCHAR 型の変数に格納すると正しく処理されません。これは、NCHAR および NVARCHAR 型の変数はデータ長をバイト数ではなく文字数で指定し、1 文字が 2 バイトであることを前提にして処理されているためです。EUC コードセット 3 の外字は 1 文字が 3 バイトなので、正しく処理されません。

(例 1)

EUC コードセット 3 の外字コードを含む NCHAR および NVARCHAR 型の列を対象に検索、格納、比較を行う場合は、VARCHAR 型または CHAR 型の変数を使用してください。EUC コードセット 1 の漢字（1 文字が 2 バイト）と EUC コードセット 3 の外字が混在すると、文字数からデータのバイト数が確定しないので、NCHAR 型（固定長）の列が対象の場合でも VARCHAR 型（可変長）の変数を使用することをお勧めします。変数の長さは次のようになります。

対象列	処理	VARCHAR 型変数の長さ
NCHAR(n) および NVARCHAR(n)	検索	定義長(バイト数)= $n \times 3$
	格納	実長 $\leq n \times 2^*$

注※ 格納するデータの文字数がわからない場合の計算です。文字数がわかる場合は、文字数 $\leq n$ にしてください。

(例 2)

DESCRIBE 文が SQL 記述領域に設定したデータ型が NCHAR および NVARCHAR 型の場合に、EUC コードセット 3 の外字を含むデータを変数に格納するときは、SQL 記述領域のデータ型を VARCHAR 型に書き換えてください。

(b) データベース中の文字コードをシフト JIS にする場合

CREATE TABLE 文で表を定義する場合に、文字集合の指定でシフト JIS を選択すると、データベースに格納する文字コードがシフト JIS になります。

データベース中のデータがシフト JIS の場合、文字コードを変換する必要がなくなるため、「(a) データベース中の文字コードを EBCDIK/KEIS にする場合」の注意事項を考慮する必要がありません。

文字集合としてシフト JIS を使用する場合の CREATE TABLE 文の例を次に示します。

```
CREATE TABLE 表名 (列名 CHAR (100) CHARACTER SET SHIFT_JIS)
```

なお、文字集合をシフト JIS にする場合は、次の点に注意してください。

(i)VOS3 上の UAP での文字コード変換

WS や PC で動作する HiRDB クライアントの UAP では文字コードを変換する必要がありませんが、VOS3 上で動作する UAP (EBCDIK/KEIS を使用する UAP) では文字コードを変換する必要があります。

このため、VOS3 上の UAP を作成する上で、文字コードを変換するときに注意する必要があります。

(ii)制限事項

XDM/RD E2 側で文字集合とほかの機能の組み合わせが制限されている場合があります。次に XDM/RD E2 07-00 での制限事項の一部を示します。その他の制限事項については、XDM/RD E2 のマニュアルを参照してください。

- ・ 文字集合シフト JIS の列を分割表の分割条件列に指定できない。
- ・ 文字集合シフト JIS の列をインデクス除外値構成列に指定できない。

(6) SQL の最大長

一つの SQL の最大長は 2,000,000 バイトです。

ただし、最大長は DB コネクションサーバの空間起動制御文などに指定する「MAX SQL LENGTH」の指定によって制限されます。

詳細については、マニュアル「VOS3 データマネジメントシステム XDM E2 系 XDM/RD E2 使用の手引－RD 環境定義文・空間起動制御文・運用コマンド編－」を参照してください。

(7) 名前に指定

名前に使用できる文字を次の表に示します。

表 6-3 名前に使用できる文字

名前の種類	制限文字数	英大文字数字	英小文字	カナ文字	下線	空白	ハイフン	全角文字だけの各国文字の使用可否	全角文字と半角文字の混在可否
認可識別子	RD ※4	RD	RD	RD	RD	RD	RD	RD	RD
パスワード	RD	RD	RD	RD	RD	RD	RD	RD	RD

名前の種類	制限文字数	英大文字数字	英小文字	カナ文字	下線	空白	ハイフン	全角文字だけの各国文字の使用可否	全角文字と半角文字の混在可否
スキーマ識別子	RD ※4	RD	RD	RD	RD	RD	RD	RD	RD
表識別子	RD	RD	RD	RD	RD	RD	RD	RD	RD
列名	RD ※4	RD	RD	RD	RD	RD	RD	RD	RD
相関名	RD ※4	RD	RD	RD	RD	RD	RD	RD	RD
問合せ名	RD	RD	RD	RD	RD	RD	RD	RD	RD
属性名	RD	RD	RD	RD	RD	RD	RD	RD	RD
カーソル識別子	30バイト	○ ※2	○	○	○	×	○ ※3	○	○
SQL文識別子	30バイト	○ ※2	○ ※2	○	○	×	○	○	○
ホスト識別子※1	30文字	○ ※2	○ ※2	○	○	×	○	○	○
SQLパラメタ名	RD	RD	RD	RD	RD	RD	RD	RD	RD
リスト名	RD ※4	RD	RD	RD	RD	RD	RD	RD	RD
文ラベル	RD	RD	RD	RD	RD	RD	RD	RD	RD
インデクス識別子	RD ※4	RD	RD	RD	RD	RD	RD	RD	RD
トリガ識別子	RD	RD	RD	RD	RD	RD	RD	RD	RD
ルーチン識別子	RD	RD	RD	RD	RD	RD	RD	RD	RD
特定化識別子	RD	RD	RD	RD	RD	RD	RD	RD	RD
データ型識別子	RD	RD	RD	RD	RD	RD	RD	RD	RD
制約名	RD ※4	RD	RD	RD	RD	RD	RD	RD	RD
RD エリア名	RD	RD	RD	RD	RD	RD	RD	RD	RD
RD ノード名称	RD	RD	RD	RD	RD	RD	RD	RD	RD
dd 名	RD	RD	RD	RD	RD	RD	RD	RD	RD

6. SQL 言語仕様

(凡例)

RD :

XDM/RD E2 の仕様に従います。詳細については、マニュアル「VOS3 データマネジメントシステム XDM E2 系 XDM/RD E2 SQL リファレンス」を参照してください。

○ :

使用できます。名前の先頭の文字は、半角英大文字、半角英小文字、半角仮名大文字にしてください。ただし、全角文字を使用できる名前の先頭の文字は、全角文字でできます。

× :

使用できません。

注※1

ホスト言語によって制限されます。

注※2

半角英小文字は、半角英大文字として扱われます。ただし、引用符「"」で囲めば、区別して扱えます。

注※3

名前に半角ハイフンを含む場合には、引用符「"」で囲んでください。ただし、半角ハイフンを含むカーネル名は、手続きまたは関数の中に指定する場合だけ引用符「"」で囲んでください。

注※4

全角文字を使用すると、クライアントで使用している文字コード（シフト JIS など）から XDM/RD E2 で使用する文字コード（KEIS コード）に変換する際にデータ長が変わります。このため、制限文字数は変換後のバイト数で計算してください。文字コードに関する注意事項の詳細については、「[文字コードに関する注意事項](#)」を参照してください。

(8) 予約語

HiRDB の予約語に XDM/RD E2 の予約語を加えたものが、SQL の予約語となります。

HiRDB の予約語については、マニュアル「HiRDB UAP 開発ガイド」を参照してください。

XDM/RD E2 の予約語については、マニュアル「VOS3 データマネジメントシステム XDM E2 系 XDM/RD E2 SQL リファレンス」を参照してください。

6.1.2 データ型

HiRDB クライアントからアクセスできる XDM/RD E2 のデータ型を次の表に示します。

表 6-4 HiRDB クライアントからアクセスできる XDM/RD E2 のデータ型

XDM/RD E2 のデータ型	備考	使用可能な XDM/RD E2 のバージョン
INTEGER	—	06-00 以降
SMALLINT	—	06-00 以降
DECIMAL	表定義 (CREATE TABLE 文) などに DECIMAL 型を指定した場合、XDM/RD E2 が内部的に LARGE DECIMAL 型に変換します。 ただし、XDM/RD E2 の RD 環境定義文の HIRDB PROGRAM 句に SQLSPECIFICATION XDMRD を指定した場合は変換しません。	06-00 以降
LARGE DECIMAL および NUMERIC	LARGE DECIMAL 型に対して DESCRIBE 文を発行した場合、SQLCOD に返すデータ型は DECIMAL (X'E4'または X'E5') になります。	06-00 以降
FLOAT および DOUBLE PRECISION	使用できる値の範囲および値の精度は VOS3 の倍精度浮動小数点型 (符号部=0 ビット目, 指数部=1~7 ビット目, 仮数部=8~63 ビット目) と PC または WS の倍精度浮動小数点型 (符号部=0 ビット目, 指数部=1~11 ビット目, 仮数部=12~63 ビット目) の小さい方の値となります。 また、変換の過程で精度が変わったり、アンダーフローおよびオーバーフローが発生したりする可能性があります。	06-00 以降
SMALLFLT および REAL	使用できる値の範囲および値の精度は VOS3 の单精度浮動小数点型 (符号部=0 ビット目, 指数部=1~7 ビット目, 仮数部=8~31 ビット目) と PC または WS の单精度浮動小数点型 (符号部=0 ビット目, 指数部=1~8 ビット目, 仮数部=12~31 ビット目) の小さい方の値となります。 また、変換の過程で精度が変わったり、アンダーフローおよびオーバーフローが発生したりする可能性があります。	06-00 以降
CHAR	—	06-00 以降
VARCHAR	—	06-00 以降
LONG VARCHAR	LONG VARCHAR 型に対して DESCRIBE 文を発行した場合、SQLCOD に返すデータ型は VARCHAR (X'C0'または X'C1') となります。	06-00 以降
NCHAR	—	06-00 以降
NVARCHAR	—	06-00 以降
LONG NVARCHAR	LONG NVARCHAR 型に対して DESCRIBE 文を発行した場合、SQLCOD に返すデータ型は NVARCHAR (X'B0'または X'B1') となります。	06-00 以降
MCHAR	—	06-00 以降
MVARCHAR	—	06-00 以降

XDM/RD E2 のデータ型	備考	使用可能な XDM/RD E2 のバージョン
LONG MVARCHAR	LONG MVARCHAR 型に対して DESCRIBE 文を発行した場合、SQLCOD に返すデータ型は MVARCHAR (X'A0'または X'A1') となります。	06-00 以降
DATE	—	06-00 以降
TIME	—	06-00 以降
	XDM/RD E2 のバージョンが 08-00 以降の場合、小数秒精度の指定ができます。	08-00 以降
TIMESTAMP	埋込み変数は SQL の CHAR 型に対応する属性で定義します。 XDM/RD E2 の TIMESTAMP 型に対して DESCRIBE 文を発行した場合、SQLCOD は X'7C'または X'7D'となるため、HiRDB クライアント側で動的 SQL を実行するときは SQLCOD に X'7C'または X'7D'を指定しないでください。 XDM/RD E2 のバージョンが 08-00 以降の場合、小数秒精度の指定ができます。	07-00 以降 08-00 以降
	XDM/RD E2 のバージョンが 10-01 以降の場合、上記の制限なしで使用できます。	10-01 以降
INTERVAL YEAR TO DAY	—	06-00 以降
INTERVAL HOUR TO SECOND	—	06-00 以降
INTERVAL YEAR TO SECOND	埋込み変数は SQL の DECIMAL 型に対応する属性で定義します。 XDM/RD E2 の INTERVAL YEAR TO SECOND 型に対して DESCRIBE 文を発行した場合、SQLCOD は X' 60' または X' 61' となるため、HiRDB クライアント側で動的 SQL を実行するときは SQLCOD に X' 60' または X' 61' を指定しないでください。	08-00 以降
BLOB	—	06-00 以降
抽象データ型	—	07-00 以降
CLOB	—	11-00 以降

(凡例)

—：備考はありません。

HiRDB クライアントからアクセスできない XDM/RD E2 のデータ型を次の表に示します。

表 6-5 HiRDB クライアントからアクセスできない XDM/RD E2 のデータ型

XDM/RD E2 のデータ型	備考
BOOLEAN	SQL 変数、SQL パラメタを使用したデータの受け渡しはできません。ただし、定数を使用した BOOLEAN 型の更新はできます。

XDM/RD E2 のデータ型	備考
ROW	-

6.2 構成要素の詳細

ここでは、構成要素の詳細の中で、特に、HiRDB サーバを使用する場合と XDM/RD E2 を使用する場合で差異がある項目について説明します。

その他の構成要素の詳細については、マニュアル「HiRDB UAP 開発ガイド」を参照してください。

6.2.1 スカラ関数 HEX

スカラ関数 HEX は、XDM/RD E2 がシステム内部で扱う値を 16 進文字列表現に変換します。

例えば、「SELECT HEX ('ABC') FROM 表名」とすると、EBCDIK コードの'ABC'を 16 進文字列表現に変換するため、結果は JIS コードの'414243'にはなりません。'C1C2C3'になります。

6.3 SQL 領域

ここでは、HiRDB クライアントを使用する場合、HiRDB クライアントで使用する SQL 領域の形式と、XDM/RD E2 から HiRDB クライアントに返す内容について説明します。

6.3.1 SQL 連絡領域 (SQLCA)

HiRDB クライアントを使用する場合、HiRDB クライアントで使用する SQL 連絡領域 (SQLCA) は XDM/RD E2 の形式ではなく、HiRDB の形式となります。

ただし、設定される内容は、HiRDB サーバを使用する場合とは異なります。

HiRDB の SQL 連絡領域の形式と、XDM/RD E2 から HiRDB クライアントに返す内容を次の表に示します。

なお、XDM/RD E2 から HiRDB クライアントに返す内容は、XDM/RD E2 のバージョンおよびリビジョンの違いによって変化する場合があります。

次に示す内容は XDM/RD E2 06-00 の内容です。XDM/RD E2 のバージョンが 06-01 以降の場合は、マニュアル「VOS3 データマネジメントシステム XDM E2 系 XDM/RD E2 SQL リファレンス」を参照してください。

表 6-6 HiRDB の SQL 連絡領域の形式と、XDM/RD E2 から HiRDB クライアントに返す内容

レベル 番号※1	連絡領域名	データ 型	長さ (バイト)	内容
1	1	SQLCA	—	SQL 連絡領域全体の名前を表します。
	2	SQLCAID	—	SQLCAIDC 領域、SQLCAIDX 領域、SQLCAIDS 領域、および SQLCAIDE 領域の内容を値として持ります。
	3	SQLCAIDC	char	SQL 連絡領域である文字列 ('SQLCA') が設定されます。
	3	SQLCAIDX	char	エラーの発生した XDM/RD E2 の XDM 識別子が設定されます。
	3	SQLCAIDS	char	文字列 ('△') を値として持ります。
	3	SQLCAIDE	char	文字列 ('△') を値として持ります。※2
	2	SQLCABC	long	SQL 連絡領域の長さ (336 バイト) が設定されています。
	2	SQLCODE	long	SQL 実行後に HiRDB クライアントまたは XDM/RD E2 から返されるリターンコードを受け取る領域です。リターンコードには次に示す意味があります。 負： 正常に終了していません。 0： 正常に終了しました。

レベル 番号※1	連絡領域名	データ 型	長さ (バイト)	内容
				<p>正：</p> <p>正常に終了しましたが、メッセージ情報があります。</p> <p>リターンコードに対応するメッセージについては、SQLERRMC 領域に設定されたメッセージが KFPA から始まる場合は、マニュアル「HiRDB メッセージ」、JXZ から始まる場合は、マニュアル「VOS3 データマネジメントシステム XDM E2 系 メッセージ (XDM/RD E2)」を参照してください。</p> <p>なお、リターンコードに対応するメッセージ ID については、表「リターンコードに対応するメッセージ ID」を参照してください。</p>
2	2 SQLERRM	—	256	<p>SQLERRML 領域および SQLERRMC 領域の内容を値として持ちます。なお、SQLCODE 領域に返されるリターンコードの正負によって次に示すように異なります。</p> <ul style="list-style-type: none"> リターンコードが負の場合、誤りの個所や原因を示す文字列を返すときがあります。 リターンコードが正の場合、メッセージ情報を示す文字列を返すときがあります。
3	SQLERRML	short	2	SQLERRMC 領域に返されるメッセージの長さを示します。
3	SQLERRMC	char	254	SQLCODE 領域に返されるリターンコードに対応するメッセージが格納される領域です。メッセージが KFPA から始まる場合は、HiRDB クライアントから返されたメッセージです。設定内容については、マニュアル「HiRDB メッセージ」を参照してください。また、メッセージが JXZ から始まる場合は、XDM/RD E2 から返されたメッセージです。設定内容については、マニュアル「VOS3 データマネジメントシステム XDM E2 系 メッセージ (XDM/RD E2)」を参照してください。
2	SQLERRP	char	8	XDM/RD E2 が使用します。
2	SQLERRD	long	4×6	<p>XDM/RD E2 の内部状態を示す領域で、データ型が long の 6 個の配列です。</p> <p>SQLERRD[0] :</p> <p>0, または SQL 文のエラー位置 (単位 : バイト)</p> <p>SQLERRD[1] :</p> <p>CPU 時間 (単位 : 0.1 ミリ秒) ※3</p> <p>SQLERRD[2] :</p> <p>次のどれかの値が返されます。</p> <ul style="list-style-type: none"> SELECT 文で取り出した行数 UPDATE 文で更新した行数 DELETE 文で削除した行数 INSERT 文で挿入した行数 FETCH 文で取り出した行数 ASSIGN LIST 文で作成した行数 ASSIGN LIST 文で名称変更したリストの行数※4

レベル 番号※1	連絡領域名	データ 型	長さ (バイト)	内容
				<p>SQLERRD[3] : XDM/RD E2 が使用します。</p> <p>SQLERRD[4] : XDM/RD E2 が使用します。</p> <p>SQLERRD[5] : 経過時間 (単位 : 0.1 ミリ秒) ※3</p>
2	SQLWARN0	char	1	SQLWARN1～F の領域のどれかに警告フラグ ('W') が設定された場合に'W'が設定されます。
2	SQLWARN1	char	1	文字データの切り捨てが発生した場合に'W'が設定されます。または、繰返し列の検索でそのデータを受け取る埋込み変数の要素数が少なかったため、要素が切り捨てられた値を受け取った場合に'W'が設定されます。それ以外は空白が設定されます。
2	SQLWARN2	char	1	集合関数の処理でナル値を無視した場合に'W'が設定されます。それ以外は空白が設定されます。
2	SQLWARN3	char	1	検索結果の列の数と、検索結果を受け取る埋込み変数の数が一致しない場合に'W'が設定されます。それ以外は空白が設定されます。
2	SQLWARN4	char	1	WHERE 句がない UPDATE 文または DELETE 文が実行された場合に'W'が設定されます。それ以外は空白が設定されます。
2	SQLWARN5	char	1	SELECT 文を実行した結果、検索結果が 2 行以上あった場合に'W'が設定されます。それ以外は空白が設定されます。
2	SQLWARN6	char	1	暗黙的にトランザクションが取り消された場合に'W'が設定されます。なお、X/Open に準拠した API を使用した UAP の場合は、トランザクションのロールバックが必要なときに'W'が設定されます。'W'が設定されている場合は、必ずトランザクションのロールバックをしてください。
2	SQLWARN7	char	1	UPDATE 文で SET 句または DELETE 句に添字付きの繰返し列を指定したが、更新する行にその要素がないため、更新指定が無視されたものがある場合に'W'が設定されます。それ以外は空白が設定されます。
2	SQLWARN8	char	1	予備
2	SQLWARN9	char	1	予備
2	SQLWARNA	char	1	日付演算の結果、存在しない日付が現れたときに、XDM/RD E2 が自動的にその月の最終日に修正した場合に'W'が設定されます。それ以外は空白が設定されます。
2	SQLWARNB	char	1	SQL 実行時の演算途中でオーバーフローまたは 0 除算が発生したため、演算結果をナル値とした場合に'W'が設定されます。探索条件中でオーバーフローが発生した場合は、条件が真となった行が処理される以前に処理された行で、オーバーフローが発生したことになります。それ以外は空白が設定されます。

6. SQL 言語仕様

レベル 番号※1	連絡領域名	データ 型	長さ (バイト)	内容
2	SQLWARNC	char	1	日付演算の結果で日間隔中の日の部分が、00～99の範囲外となった場合に'W'が設定されます。それ以外は空白が設定されます。
2	SQLWARND	char	1	予備
2	SQLWARNE	char	1	予備
2	SQLWARNF	char	1	予備
2	SQLCASYS	char	16	予備

(凡例)

－：該当しません。

注※1

表のレベル番号は、SQL連絡領域の包含関係を示します。例えば、レベル番号1の連絡領域はレベル番号2の連絡領域で構成されることを示します。

注※2

リモートデータベースアクセスしたサーバ側のデータベース管理システムの種別が設定されます。SQLCAIDEに設定される内容を次に示します。

設定値	サーバ側のデータベース管理システムの種別	リモートデータベースアクセスプロトコル
'D'	XDM/RD E2を示します。	DF-RDA
'△'	ローカルアクセスを示します。	該当しません
上記以外	次のどちらかを示します。 <ul style="list-style-type: none"> 上記以外のデータベース管理システムを示します。 サーバシステムと接続していないため、データベース管理システムの種別を特定できません。 	該当しません

注※3

COMMIT, ROLLBACKの時間は含まれません。

注※4

システム共通定義でpd_list_rowcount_in_renameオペランドにYを指定した場合だけ参照してください。

表6-7 リターンコードに対応するメッセージID

リターンコード	対応するメッセージID
-yyy（例：-125）	KFPA11yyyまたはJXZ1yyy（例：KFPA11125またはJXZ1125）
yyy（例：100）	KFPA12yyyまたはJXZ2yyy（例：KFPA12100またはJXZ2100）
-1yyy（例：-1000）	JXZ9yyy（例：JXZ9000）

6.3.2 SQL 記述領域 (SQLDA)

HiRDB クライアントを使用する場合、HiRDB クライアントで使用する SQL 記述領域 (SQLDA) は XDM/RD E2 の形式ではなく、HiRDB の形式となります。

ただし、HiRDB は CLOB をサポートしていません。CLOB の場合の SQL 記述領域に設定するデータコードとデータの長さを次の表に示します。

表 6-8 CLOB の場合の SQL 記述領域に設定するデータコードとデータの長さ

データコード 10 進数	データコード 16 進数	標識変数	データ型	データの長さ	単位
152	98	無	CLOB(n)	$1 \leq n \leq 2,147,483,647$	バイト
153	99	有			
148	94	無	CLOB 位置付け 子	4	
149	95	有			

HiRDB の SQL 記述領域の詳細については、マニュアル「HiRDB SQL リファレンス」を参照してください。

6.3.3 列名記述領域 (SQLNAME)

HiRDB クライアントを使用する場合、HiRDB クライアントで使用する列名記述領域 (SQLNAME) は XDM/RD E2 の形式ではなく、HiRDB の形式となります。

列名記述領域の詳細については、マニュアル「HiRDB SQL リファレンス」を参照してください。

ただし、設定する値については、マニュアル「VOS3 データマネジメントシステム XDM E2 系 XDM/RD E2 SQL リファレンス」を参照してください。

6.3.4 型名記述領域

HiRDB クライアントを使用する場合、HiRDB クライアントで使用する型名記述領域は XDM/RD E2 の形式ではなく、HiRDB の形式となります。

型名記述領域の詳細については、マニュアル「HiRDB SQL リファレンス」を参照してください。

ただし、SQLTNVAR には、スキーマ識別子が設定されます。スキーマ識別子が 9 バイト以上の場合は、ユーザ定義型の定義情報を受け取ることができません。

6.4 定義系 SQL

定義系 SQL の仕様は、XDM/RD E2 の仕様に従います。各 SQL の詳細仕様は、マニュアル「VOS3 データマネジメントシステム XDM E2 系 XDM/RD E2 SQL リファレンス」を参照してください。

なお、OLTP 下の X/Open に準拠した UAP では、定義系 SQL は使用できません。

HiRDB クライアントを使用する場合の定義系 SQL の使用可否を次の表に示します。

表 6-9 HiRDB クライアントを使用する場合の定義系 SQL の使用可否

項番	SQL	静的 SQL	動的 SQL ^{*1}	使用可能な XDM/RD E2 のバージョン
1	ALTER FUNCTION	×	○	07-00 以降
2	ALTER PROCEDURE	○	○	07-00 以降
3	ALTER PROGRAM	×	○	06-00 以降
4	ALTER SCHEMA	—	—	—
5	ALTER TABLE	○	○	06-00 以降
6	ALTER TRIGGER	×	○	06-02 以降
7	COMMENT	○	○	06-00 以降
8	CREATE FUNCTION	○ ^{*2}	○	07-00 以降
9	CREATE INDEX	○	○	06-00 以降
10	CREATE NONLOCAL TABLE	×	○	06-00 以降
11	CREATE PROCEDURE	○ ^{*2}	○	06-00 以降
12	CREATE PROGRAM	×	○	06-00 以降
13	CREATE SCHEMA	○	○	06-00 以降
14	CREATE TABLE	○	○	06-00 以降
15	CREATE TRIGGER	×	○	06-02 以降
16	CREATE TYPE	○ ^{*2}	○	07-00 以降
17	CREATE VIEW	○	○	06-00 以降
18	DROP FUNCTION	○	○	07-00 以降
19	DROP INDEX	○	○	06-00 以降
20	DROP NONLOCAL TABLE	×	○	06-00 以降
21	DROP PROCEDURE	○	○	06-00 以降
22	DROP PROGRAM	×	○	06-00 以降
23	DROP SCHEMA	○	○	06-00 以降

項番	SQL	静的 SQL	動的 SQL ^{※1}	使用可能な XDM/RD E2 のバージョン
24	DROP TABLE	○	○	06-00 以降
25	DROP TRIGGER	×	○	06-02 以降
26	DROP TYPE	○	○	07-00 以降
27	DROP VIEW	○	○	06-00 以降
28	GRANT RDAREA	○	○	06-00 以降
29	GRANT SCHEMA	○	○	06-00 以降
30	GRANT アクセス権限 (アクセス権限定義)	○	○	06-00 以降
31	GRANT ユーザ定義型権限 (ユーザ定義型権限定義)	×	○	08-00 以降
32	LABEL	×	○	06-00 以降
33	REVOKE RDAREA	○	○	06-00 以降
34	REVOKE SCHEMA	○	○	06-00 以降
35	REVOKE アクセス権限 (アクセス権限定義)	○	○	06-00 以降
36	REVOKE ユーザ定義型権限 (ユーザ定義型権限定義)	×	○	08-00 以降

(凡例)

○：使用できます。

×：使用できません（ただし、プリプロセス時にエラーにならない場合があります）。

-：該当しません。

注※1

動的 SQL とは PREPARE 文または EXECUTE IMMEDIATE 文で前処理する SQL を指します。

注※2

次に示す SQL 文中の SQL パラメタおよび RETURNS 句には、文字集合指定 (CHARACTER SET) はできません。文字集合指定をする場合は、動的 SQL (EXECUTE IMMEDIATE 文) を使用してください。

- CREATE FUNCTION
- CREATE PROCEDURE
- CREATE TYPE

6.5 操作系 SQL

ここでは、HiRDB を使用する場合、操作系 SQL が使用できるかどうか、およびデータベースを操作するときの注意事項について説明します。

6.5.1 概要

操作系 SQL の仕様は、XDM/RD E2 の仕様に従います。各 SQL の詳細仕様は、マニュアル「VOS3 データマネジメントシステム XDM E2 系 XDM/RD E2 SQL リファレンス」を参照してください。

HiRDB クライアントを使用する場合の操作系 SQL の使用可否を次の表に示します。

表 6-10 HiRDB クライアントを使用する場合の操作系 SQL の使用可否

項目番号	SQL	静的 SQL	動的 SQL	使用可能な XDM/RD E2 のバージョン
1	ALLOCATE CURSOR 文	×	×	—
2	ASSIGN LIST 文	×	×	06-00 以降
3	CLOSE 文	○	—	06-00 以降
4	DEALLOCATE PREPARE 文	×	×	—
5	DECLARE CURSOR WITH 句	○	—	06-00 以降
	DECLARE CURSOR WITH RECURSIVE 句	×	—	
	DECLARE CURSOR その他	○	—	
6	動的 DECLARE CURSOR	○	—	06-00 以降
7	DELETE 文	○	○	06-00 以降
8	DESCRIBE 文 INPUT ^{*1}	○	—	06-00 以降
	DESCRIBE 文 OUTPUT	○	—	
9	DESCRIBE TYPE 文	○	—	07-00 以降
10	DROP LIST 文	×	×	—
11	EXECUTE 文	○	—	06-00 以降
12	EXECUTE IMMEDIATE 文	○	—	06-00 以降
13	FETCH 文	○	—	06-00 以降

項目番号	SQL	静的 SQL	動的 SQL	使用可能な XDM/RD E2 のバージョン
14	FREE LOCATOR 文	○	○	11-00 以降
15	HOLD LOCATOR 文	×	○	11-00 以降
16	INSERT 文※2	○	○	06-00 以降
17	OPEN 文	○	—	06-00 以降
18	PREPARE 文	○	○	06-00 以降
19	PURGE TABLE 文※3	○	○	06-00 以降
20	1 行 SELECT 文	○	○	06-00 以降
21	動的 SELECT 文 WITH 句	—	○	06-00 以降
	動的 SELECT 文 WITH RECURSIVE 句	—	○	06-00 以降
	動的 SELECT 文 その他	—	○	06-00 以降
22	1 行 SELECT LIST 文	×	×	—
23	UPDATE 文	○	○	06-00 以降

(凡例)

○：使用できます。

×：使用できません（ただし、プリプロセス時にエラーにならない場合もあります）。

—：該当しません。

注※1

XDM/RD E2 07-00 以降のバージョンでは、SQLNAME（列名記述領域）に CALL 文で指定した？パラメタに対応する SQL パラメタ名が返却されます。

注※2

静的 SQL の場合は、「挿入値」に「値式」は指定できず、「値指定」となります。

注※3

OLTP 下の X/Open に準拠した UAP では、PURGE TABLE 文は使用できません。

6.5.2 データベース操作時の注意事項

(1) 行単位 (ROW 指定) インタフェース

PC または WS では、行単位 (ROW 指定) インタフェースは使用できません。

SQL 構文規則

- 選択式に ROW は指定できません。
- UPDATE 文に「SET ROW = 行更新値」は指定できません。
- INSERT 文に「表識別子 (ROW)」は指定できません。

(2) 繰返し列

繰返し列を使用する場合の SQL の構文規則は、XDM/RD E2 の RD 環境定義文の HIRDB PROGRAM 句に SQL SPECIFICATION XDMRD を指定しているかどうかによって異なります。

SQL の構文規則の詳細については、マニュアル「VOS3 データマネジメントシステム XDM E2 系 XDM/RD E2 SQL リファレンス」を参照してください。

また、XDM/RD E2 のバージョンによって、繰返し列が使用できる場合とできない場合があります。XDM/RD E2 のバージョンと繰返し列の使用可否を次に示します。

XDM/RD E2 のバージョン	繰返し列の使用可否
06-03 以前	×
07-00 以降	○

(凡例)

- ：使用できます。
- ×：使用できません。

注意事項

- 添字指定の構文形式

XDM/RD E2 接続機能を使用する場合は、添字指定の構文形式は角括弧形式となります。
XDM/RD E2 の構文形式と XDM/RD E2 接続機能を使用する場合の構文形式を次に示します。

XDM/RD E2 の構文形式

```
SELECT * FROM T1 WHERE C1 (ANY) =1
```

XDM/RD E2 接続機能を使用する場合の構文形式

```
SELECT * FROM T1 WHERE C1 [ANY] =1
```

- 要素リスト

要素の追加に指定する要素リストの構文形式は、角括弧形式となります。

要素リストの詳細については、マニュアル「VOS3 データマネジメントシステム XDM E2 系 XDM/RD E2 SQL リファレンス」を参照してください。

(3) 配列列

配列列は使用できません。

DESCRIBE 文を発行すると、データ型を示す SQLCOD に 0 (未サポートの意味) を返します。

(4) ユティリティ

HiRDB のユティリティを用いて XDM/RD E2 にアクセスすることはできません。

ただし、「会話型 SQL 実行ユティリティ (pssql)」を使用して SQL を実行させることはできます (pssql コマンドの使用可否については、HiRDB SQL Executer を参照してください。なお、一部のコマンドについては、XDM/RD E2 で HiRDB 形式ディクショナリビュー表を作成する必要があります。HiRDB 形式ディクショナリビュー表については、マニュアル「VOS3 データマネジメントシステム XDM E2 系 XDM/RD E2 使用の手引－運用編－」を参照してください)。

この場合の SQL の仕様は、XDM/RD E2 の SQL 仕様 (HiRDB プログラムの SQL 仕様) に従います。

(5) 明示表

XDM/RD E2 のバージョンや呼び出し形態によって、明示表が使用できる場合とできない場合があります。XDM/RD E2 のバージョンと明示表の使用可否を次に示します。

XDM/RD E2 のバージョン	明示表の使用可否	
	動的 SQL	静的 SQL
07-01 以前	×	×*
08-00 以降	○	×*

(凡例)

○：使用できます。

×：使用できません。

注※

プリプロセス時にエラーにならない場合がありますが、動作は保証されません。

- プリプロセス時にエラーにならない例

次のように WHERE 句に明示表を指定した場合

```
SELECT * FROM T1 WHERE (C1,C2) = (TABLE T2)
```

6.6 制御系 SQL

制御系 SQL の仕様は、XDM/RD E2 の仕様に従います。各 SQL の詳細仕様は、マニュアル「VOS3 データマネジメントシステム XDM E2 系 XDM/RD E2 SQL リファレンス」を参照してください。

HiRDB クライアントを使用する場合の制御系 SQL の使用可否を次の表に示します。

表 6-11 HiRDB クライアントを使用する場合の制御系 SQL の使用可否

項番	SQL	静的 SQL	動的 SQL	使用可能な XDM/RD E2 のバージョン
1	CALL 文※1	○	○	06-00 以降
2	COMMIT 文※2	○	—	06-00 以降
3	CONNECT 文※2,3	○	—	06-00 以降
4	DISCONNECT 文※2	○	—	06-00 以降
5	LOCK 文	○	○	06-00 以降
6	ROLLBACK 文※2,4	○	—	06-00 以降
7	SET SCHEMA 文※5	×	○	06-00 以降

(凡例)

○：使用できます。

×：使用できません（ただし、プリプロセス時にエラーにならない場合もあります）。

—：該当しません。

注※1

HiRDB では CALL 文は制御系 SQL に含まれます。

XDM/RD E2 では CALL 文は SQL 制御文に含まれます。

注※2

OLTP 下の X/Open に準拠した UAP では、次に示す SQL 文は使用できません。

- COMMIT 文
- CONNECT 文
- DISCONNECT 文
- ROLLBACK 文

注※3

CONNECT 文を記述する時に、CONNECT だけを指定して以降のオペランド（埋込み変数など）を省略した場合は、環境変数 PDUSER に指定した認可識別子とパスワードを使用して XDM/RD E2 と接続します。

注※4

ROLLBACK 文に WITH STOPPING は指定できません。

注※5

SET SCHEMA 文の対象となる SQL は動的 SQL であり、静的 SQL は対象になりません。

6.7 SQL 制御文

HiRDB クライアントを使用する場合の SQL 制御文の使用可否を次の表に示します。

表 6-12 HiRDB クライアントを使用する場合の SQL 制御文の使用可否

項目番	SQL	静的 SQL	動的 SQL	使用可能な XDM/RD E2 のバージョン
1	CALL 文※1	○	○	06-00 以降
2	CASE 文	—※2	—※2	08-00 以降
3	IF 文	—※2	—※2	07-00 以降
4	ITERATE 文	—※2	—※2	08-00 以降
5	LEAVE 文	—※2	—※2	07-00 以降
6	LOOP 文	—※2	—※2	08-00 以降
7	REPEAT 文	—※2	—※2	08-00 以降
8	RETURN 文	—※2	—※2	07-00 以降
9	WHILE 文	—※2	—※2	07-00 以降
10	代入文	—※2	—※2	06-02 以降
11	複合文	—※2	—※2	06-02 以降
12	WRITE LINE 文	—※2	—※2	11-00 以降

(凡例)

○：使用できます。

－：該当しません。

注※1

XDM/RD E2 では CALL 文は SQL 制御文に含まれます。

HiRDB では CALL 文は制御系 SQL に含まれます。

注※2

SQL 手続き文中で使用する SQL です。SQL 手続き文中に指定できます。

6.8 SQL 診断文

HiRDB クライアントを使用する場合の SQL 診断文の使用可否を次の表に示します。

表 6-13 HiRDB クライアントを使用する場合の SQL 診断文の使用可否

項目番	SQL	静的 SQL	動的 SQL	使用可能な XDM/RD E2 のバージョン
1	GET DIAGNOSTICS 文	×	—※1	—
2	SIGNAL 文	—※2	—※2	—

(凡例)

×：使用できません。

－：該当しません。

注※1

動的 SQL では使用できない SQL 文のため該当しません。

注※2

SQL 手続き文中で使用する SQL です。SQL 手続き文中に指定できます。

6.9 埋込み言語文法

埋込み言語は、埋込み型 UAP を作成する場合にプログラム用 SQLと一緒に使用して、埋込み変数の宣言およびリターンコードによる処理の宣言をする SQL です。

埋込み型言語については、XDM/RD E2 の仕様ではなく、HiRDB の仕様に従います。ただし、HiRDB で使用できるすべてが使用できるわけではなく、使用できないものもあります。

XDM/RD E2 接続機能を使用する場合の埋込み型言語の使用可否を次の表に示します。

表 6-14 XDM/RD E2 接続機能を使用する場合の埋込み型言語の使用可否

項番	埋込み型言語	使用可否
1	BEGIN DECLARE SECTION (埋込み SQL 開始宣言)	○
2	END DECLARE SECTION (埋込み SQL 終了宣言)	○
3	ALLOCATE CONNECTION HANDLE	○
4	FREE CONNECTION HANDLE	○
5	DECLARE CONNECTION HANDLE SET	○
6	DECLARE CONNECTION HANDLE UNSET	○
7	GET CONNECTION HANDLE	○
8	COPY (登録集原文の引込み)	○
9	GET DIAGNOSTICS (診断情報取得)	×
10	COMMAND EXECUTE	×
11	SQL 先頭子	○
12	SQL 終了子	○
13	WHENEVER (埋込み例外宣言)	○
14	SQLCODE 変数	△
15	SQLSTATE 変数	△
16	PDCNCTHDL 変数	○
17	INSTALL JAR	×
18	REPLACE JAR	×
19	REMOVE JAR	×
20	DECLARE AUDIT INFO SET	×

(凡例)

○：使用できます。

△：使用できますが、仕様に差異があります。

×：使用できません。

- SQLCODE

SQLCODE 変数で受け取る SQLCODE は、XDM/RD E2 が設定する SQLCODE または HiRDB クライアントが設定する SQLCODE となります。SQLCODE 変数の値は、SQL 連絡領域 (SQLCA) の SQLCODE と同じです。詳細については「[SQL 連絡領域 \(SQLCA\)](#)」を参照してください。

XDM/RD E2 が設定する SQLCODE は、XDM/RD E2 の RD 環境定義文の HIRDB PROGRAM 句の指定によって変化します。詳細についてはマニュアル「[VOS3 データマネジメントシステム XDM E2 系 XDM/RD E2 使用の手引－RD 環境定義文・空間起動制御文・運用コマンド編－](#)」を参照してください。

- SQLSTATE

SQLSTATE 変数で受け取る値は、XDM/RD E2 が設定する SQLSTATE または HiRDB クライアントが設定する SQLSTATE となります。

XDM/RD E2 が設定する SQLSTATE は、XDM/RD E2 の RD 環境定義文の HIRDB PROGRAM 句の指定によって変化します。詳細についてはマニュアル「[VOS3 データマネジメントシステム XDM E2 系 XDM/RD E2 使用の手引－RD 環境定義文・空間起動制御文・運用コマンド編－](#)」を参照してください。

各埋込み言語の詳細については、マニュアル「[HiRDB SQL リファレンス](#)」を参照してください。

付録

付録 A SQL のデータ型とデータ記述

ここでは、SQL のデータ型とデータ記述の対応の中で、HiRDB サーバを使用する場合と XDM/RD E2 を使用する場合で差異がある項目について説明します。

HiRDB サーバを使用する場合の SQL のデータ型とデータ記述については、マニュアル「HiRDB UAP 開発ガイド」を参照してください。

付録 A.1 SQL のデータ型と C 言語のデータ記述

SQL のデータ型と C 言語のデータ記述の対応は、マニュアル「HiRDB UAP 開発ガイド」の「SQL のデータ型と C 言語のデータ記述」で示す対応に従います。ただし、XDM/RD E2 接続機能で固有の対応をする場合があるので注意が必要です。

(1) 配列、繰返し列を使用しない場合

SQL のデータ型と C 言語のデータ記述の対応の、XDM/RD E2 接続機能を使用時に固有の追加項目を次の表に示します。

表 A-1 SQL のデータ型と C 言語のデータ記述（追加項目）

SQL のデータ型	C 言語のデータ記述	項目の記述	備考
LARGE DECIMAL [(p[, s])]	SQL TYPE IS DECIMAL(p,s) 変数名;※ ¹	変数	$1 \leq p \leq 29$, $0 \leq s \leq p$
LONG VARCHAR(n)	struct { short 変数名 1; char 変数名 2[n]; } 構造体名;	二つの変数から構成される構造体 変数名 1:文字列長 変数名 2:文字列	$1 \leq n \leq 32,000$
LONG NVARCHAR(n)	struct { short 変数名 1; char 変数名 2[2n]; } 構造体名;	二つの変数から構成される構造体 変数名 1:文字列長 変数名 2:文字列	$1 \leq n \leq 16,000$
LONG MVARCHAR(n)	struct { short 変数名 1; char 変数名 2[n]; } 構造体名;	二つの変数から構成される構造体 変数名 1 : 文字列長 変数名 2 : 文字列	$1 \leq n \leq 32,000$
INTERVAL YEAR TO SECOND(q)	SQL TYPE IS DECIMAL(14+q,q) 変数名;	変数	$q=0, 2, 4, 6$

SQL のデータ型	C 言語のデータ記述	項目の記述	備考
CLOB	SQL TYPE IS CLOB(n{K M G})変数名;※2	変数	単位省略時： 1≤n≤ 2,147,483,647 単位が KB： 1≤n≤2,097,152 単位が MB： 1≤n≤2,048 単位が GB： 1≤n≤2
CLOB 位置付け子	SQL TYPE IS CLOB AS LOCATOR 変数名; ※3	変数	—
標識変数	CLOB	long 変数名;	変数

(凡例)

n : 長さ (バイト)

p : 精度 (全体のけた数)

q : 小数秒精度

s : 位取り (小数点以下のけた数)

– : 備考はありません。

注※ 1

DECIMAL の UAP の記述は、内部的に次のように展開されます。

```
unsigned char 変数名[p/2+1];
```

DECIMAL のデータは 1 バイトで 2 けたの数字を表現します。符号は、右端のバイトの右 4 ビットで表します。そのため、偶数けたの DECIMAL 型の場合、左端の左 4 ビットに 0 を補う必要があります。0 以外の値は設定しないでください。

標準的な符号表現を次に示します。HiRDB での DECIMAL 型の符号については、マニュアル「HiRDB SQL リファレンス」を参照してください。

符号の 16 進数表現	意味
X'C'	正の符号とみなします。正数には 0 を含みます。
X'D'	負の符号とみなします。

記述例

123.4567 の場合 (奇数けた)

```
unsigned char ex1[4]={0x12,0x34,0x56,0x7c} ;
```

-123.456 の場合 (偶数けた)

```
unsigned char ex2[4]={0x01,0x23,0x45,0x6d} ;
```

0の場合（奇数けた）

```
unsigned char ex3[1]={0x0c} ;
```

注※2

CLOB の UAP の記述は、内部的に次のように展開されます。

```
struct{
    long      变数名_reserved;      ... 1
    unsigned long 变数名_length;    ... 2
    char       变数名_data[m];     ... 3
}变数名;
```

1. 変数名_reserved は、使用しません。64 ビットモードの場合は、int 変数名_reserved;となります。
2. 変数名_length は、CLOB の実際の長さ（バイト）になります。64 ビットモードの場合は、
unsigned int 変数名_length となります。
3. 変数名_data[m] は、CLOB のデータ格納領域（m は実際のデータ長）になります。

注※3

CLOB 位置付け子の UAP の記述は、内部的に次のように展開されます。

```
unsigned long 变数名;
```

64 ビットモードの場合は、unsigned long 変数名は unsigned int 変数名;になります。

また、SQL のデータ型と C 言語のデータ記述の変更前と変更後を表「SQL のデータ型と C 言語のデータ記述（変更前）」および表「SQL のデータ型と C 言語のデータ記述（変更後）」に示します。

表 A-2 SQL のデータ型と C 言語のデータ記述（変更前）

SQL のデータ型	C 言語のデータ記述	項目の記述	備考
TIME	char 変数名[9];	変数	－
TIMESTAMP[(q)]	char 変数名[n+1];	変数	q=0 の場合： n=19 q=2 の場合： n=21, 22 q=4 の場合： n=23, 24 q=6 の場合： n=25, 26

（凡例）

n：長さ（バイト）

q：小数秒精度

－：備考はありません。

表 A-3 SQL のデータ型と C 言語のデータ記述 (変更後)

SQL のデータ型	C 言語のデータ記述	項目の記述	備考
TIME[(q)]	char 変数名[n+1] ;	変数	q=0 の場合 : n=8 q=2 の場合 : n=11 q=4 の場合 : n=13 q=6 の場合 : n=15
TIMESTAMP[(q)]	char 変数名[n+1];	変数	q=0 の場合 : n=19 q=2 の場合 : n=22 q=4 の場合 : n=24 q=6 の場合 : n=26

(凡例)

n : 長さ (バイト)

q : 小数秒精度

注

太字の部分が変更点です。

(2) 配列を使用する場合

SQL のデータ型と C 言語のデータ記述の対応の、XDM/RD E2 接続機能を使用時に固有の追加項目（配列を使用する場合）を次の表に示します。

表 A-4 配列を使用する場合の SQL のデータ型と C 言語のデータ記述 (追加項目)

SQL のデータ型	C 言語のデータ記述	項目の記述	備考
LARGE DECIMAL	—	—	—
LONG VARCHAR(n)	struct { short 変数名 1; char 変数名 2[n]; } 構造体名[m];	構造体の配列	1≤n≤32,000
LONG NVARCHAR(n)	struct { short 変数名 1; char 変数名 2[2n]; } 構造体名[m];	構造体の配列	1≤n≤16,000
LONG MVARCHAR(n)	struct {	構造体の配列	1≤n≤32,000

SQL のデータ型	C 言語のデータ記述	項目の記述	備考
	short 変数名 1; char 変数名 2[n]; } 構造体名[m];		
ROW	char 変数名[m][n+1];	構造体の配列	$1 \leq n \leq 30,000$
INTERVAL YEAR TO SECOND(q)	—	—	—
CLOB	—	—	—
CLOB 位置付け子	—	—	—

(凡例)

m : 配列の要素数

n : 長さ (バイト)

q : 小数秒精度

— : 該当しません。

また、SQL のデータ型と C 言語のデータ記述の変更前と変更後を表「配列を使用する場合の SQL のデータ型と C 言語のデータ記述 (変更前)」および表「配列を使用する場合の SQL のデータ型と C 言語のデータ記述 (変更後)」に示します。

表 A-5 配列を使用する場合の SQL のデータ型と C 言語のデータ記述 (変更前)

SQL のデータ型	C 言語のデータ記述	項目の記述	備考
TIME	char 変数名[m][9];	配列	—
TIMESTAMP[(q)]	char 変数名[m][n+1];	配列	q=0 の場合： n=19 q=2 の場合： n=21, 22 p=4 の場合： n=23, 24 q=6 の場合： n=25, 26

(凡例)

m : 配列の要素数

n : 長さ (バイト)

q : 小数秒精度

— : 備考はありません。

表 A-6 配列を使用する場合の SQL のデータ型と C 言語のデータ記述 (変更後)

SQL のデータ型	C 言語のデータ記述	項目の記述	備考
TIME[(q)]	char 変数名[m][n+1];	配列	q=0 の場合 : n=8 q=2 の場合 : n=11 q=4 の場合 : n=13 q=6 の場合 : n=15
TIMESTAMP[(q)]	char 変数名[m][n+1];	配列	q=0 の場合 : n=19 q=2 の場合 : n=22 q=4 の場合 : n=24 q=6 の場合 : n=26

(凡例)

m : 配列の要素数

n : 長さ (バイト)

q : 小数秒精度

注

太字の部分が変更点です。

(3) 繰返し列を使用する場合

SQL のデータ型と C 言語のデータ記述の対応の、 XDM/RD E2 接続機能を使用時に固有の追加項目（繰返し列を使用する場合）を次の表に示します。

表 A-7 繰返し列を使用する場合の SQL のデータ型と C 言語のデータ記述 (追加項目)

SQL のデータ型	C 言語のデータ記述	項目の記述	備考
LARGE DECIMAL [(p[, s])]	PD_MV_DEC(m,p,s) 変数名;	要素と配列から構成される構造体	—
NVARCHAR(n)	PD_MV_NVCHAR(m,n) 変数名;	要素と配列から構成される構造体	$1 \leq n \leq 16,000$
TIMESTAMP	PD_MV_CHAR(m,19) 変数名;	要素と配列から構成される構造体	—
INTERVAL YEAR TO SECOND(q)	PD_MV_DEC(m,14+q,q) 変数名;	要素と配列から構成される構造体	q=0, 2, 4, 6
CLOB	—	—	—

SQL のデータ型	C 言語のデータ記述	項目の記述	備考
CLOB 位置付け子	—	—	—

(凡例)

m : 繰返し列の最大要素数
n : 長さ (バイト)
p : 精度 (全体のけた数)
q : 小数秒精度
s : 位取り (小数点以下のけた数)
– : 該当しません。

また、SQL のデータ型と C 言語のデータ記述の変更前と変更後を表「繰返し列を使用する場合の SQL のデータ型と C 言語のデータ記述 (変更前)」および表「繰返し列を使用する場合の SQL のデータ型と C 言語のデータ記述 (変更後)」に示します。

表 A-8 繰返し列を使用する場合の SQL のデータ型と C 言語のデータ記述 (変更前)

SQL のデータ型	C 言語のデータ記述	項目の記述	備考
VARCHAR(n) /MVARCHAR(n)	PD_MV_VCHAR(m,n) 変数名;	要素と配列から構成される構造体	$1 \leq n \leq 32,000$
NVARCHAR(n)	PD_MV_NVCHAR(m,n) 変数名;		$1 \leq n \leq 16,000$
TIME	PD_MV_CHAR(m,8) 変数名;		—
TIMESTAMP[(q)]	PD_MV_CHAR(m,n) 変数名;		<p>q=0 の場合： n=19</p> <p>q=2 の場合： n=21, 22</p> <p>q=4 の場合： n=23, 24</p> <p>q=6 の場合： n=25, 26</p>

(凡例)

m : 繰返し列の最大要素数
n : 長さ (バイト)
q : 小数秒精度
– : 備考はありません。

表 A-9 繰返し列を使用する場合の SQL のデータ型と C 言語のデータ記述 (変更後)

SQL のデータ型	C 言語のデータ記述	項目の記述	備考
VARCHAR(n) /LONG VARCHAR(n) /MVARCHAR(n) /LONG MVARCHAR(n)	PD_MV_VCHAR(m,n) 変数名;	要素と配列から構成される構造体	$1 \leq n \leq 32,000$
NVARCHAR(n) /LONG NVARCHAR(n)	PD_MV_NVCHAR(m,n) 変数名;		$1 \leq n \leq 16,000$
TIME[(q)]	PD_MV_CHAR(m,n) 変数名;		q=0 の場合 : n=8 q=2 の場合 : n=11 q=4 の場合 : n=13 q=6 の場合 : n=15
TIMESTAMP[(q)]	PD_MV_CHAR(m,n) 変数名;		q=0 の場合 : n=19 q=2 の場合 : n=22 q=4 の場合 : n=24 q=6 の場合 : n=26

(凡例)

m : 繰返し列の最大要素数

n : 長さ (バイト)

q : 小数秒精度

注

太字の部分が変更点です。

付録 A.2 SQL のデータ型と COBOL 言語のデータ記述

SQL のデータ型と COBOL 言語のデータ記述の対応は、マニュアル「HiRDB UAP 開発ガイド」の「SQL のデータ型と COBOL 言語のデータ記述」で示す対応に従います。ただし、XDM/RD E2 接続機能で固有の対応をする場合があるので注意が必要です。

(1) 配列、繰返し列を使用しない場合

SQL のデータ型と COBOL 言語のデータ記述の対応の、XDM/RD E2 接続機能を使用時に固有の追加項目を次の表に示します。

表 A-10 SQL のデータ型と COBOL 言語のデータ記述（追加項目）

SQL のデータ型	COBOL 言語のデータ記述	項目の記述	備考
LARGE DECIMAL [(p[, s])]	L1 基本項目名 PICTURE S9(p-s)[V9(s)] COMPUTATIONAL-3.	基本項目または独立項目	$1 \leq p \leq 18$, $0 \leq s \leq p$ $p=s$ の場合, SV9(s) します。 $s=0$ の場合, [V9(s)] を省略します。
LONG VARCHAR(n)	L2 集団項目名. L3 基本項目名 1 PICTURE S9(4) COMPUTATIONAL. L3 基本項目名 2 PICTURE X(n).	二つの基本項目から構成される集団項目 基本項目名 1 : 文字列長基本項目名 2 : 文字列	$1 \leq n \leq 32,000$
LONG NVARCHAR(n)	L2 集団項目名. L3 基本項目名 1 PICTURE S9(4) COMPUTATIONAL. L3 基本項目名 2 PICTURE N(n).	二つの基本項目から構成される集団項目 基本項目名 1 : 文字列長基本項目名 2 : 文字列	$1 \leq n \leq 16,000$
LONG MVARCHAR(n)	L2 集団項目名. L3 基本項目名 1 PICTURE S9(4) COMPUTATIONAL. L3 基本項目名 2 PICTURE X(n).	二つの基本項目から構成される集団項目 基本項目名 1 : 文字列長 基本項目名 2 : 文字列	$1 \leq n \leq 32,000$
INTERVAL YEAR TO SECOND(q)	L1 基本項目名 PICTURE S9(14)[V9(q)] COMPUTATION-3.	基本項目または独立項目	$q=0, 2, 4, 6$ $q=0$ の場合, [V9(q)] を省略します。 $q>4$ の場合, 精度 > 18 となるため, $q=6$ の場合は精度 = 18 で展開されるので, V9(4) となります。
CLOB(n)	L2 集団項目名※1 [USAGE [IS]] SQL TYPE IS CLOB(n{K M G}).※2	基本項目	単位省略時： $1 \leq n \leq 2,147,483,647$ 単位が KB： $1 \leq n \leq 2,097,152$ 単位が MB： $1 \leq n \leq 2,048$ 単位が GB： $1 \leq n \leq 2$
CLOB 位置付け子	L1 基本項目名	基本項目または独立項目	—

SQL のデータ型		COBOL 言語のデータ記述	項目の記述	備考
		[USAGE [IS]] SQL TYPE IS CLOB AS LOCATOR.*3		
標識 変数	CLOB	L4 基本項目名 PICTURE S9(9) COMPUTATIONAL.	基本項目または独立項目	—

(凡例)

L1 : レベル番号 01~49, または 77

L2 : レベル番号 01~48

L3 : レベル番号 02~49(L2<L3)

L4 : レベル番号 01~49

n : 長さ (バイト)

p : 精度 (全体のけた数)

q : 小数秒精度

s : 位取り (小数点以下のけた数)

– : 備考はありません。

注※ 1

集団項目名は 21 文字以下にしてください。ただし、COBOL2002 の場合は 22 文字以下にしてください。

注※ 2

CLOB の UAP の記述は、内部的に次のように展開されます。

L2 集団項目名.

49 集団項目名-RESERVED PICTURE S(9) USAGE IS BINARY.	… 1
49 集団項目名-LENGTH PICTURE S(9) USAGE IS BINARY.	… 2
49 集団項目名-DATA PIC X(m).	… 3

1. 変数名-RESERVED は、使用しません。

2. 変数名-LENGTH は、CLOB の実際の長さ (バイト) になります。

3. 変数名-DATA は、CLOB のデータ格納領域 (m は実際のデータ長) になります。

注※ 3

CLOB 位置付け子の UAP の記述は、内部的に次のように展開されます。

L1 基本項目名 PICTURE S9(9) COMPUTATIONAL.

また、SQL のデータ型と COBOL 言語のデータ記述の変更前と変更後を表「SQL のデータ型と COBOL 言語のデータ記述 (変更前)」および表「SQL のデータ型と COBOL 言語のデータ記述 (変更後)」に示します。

表 A-11 SQL のデータ型と COBOL 言語のデータ記述 (変更前)

SQL のデータ型	COBOL 言語のデータ記述	項目の記述	備考
TIME	L1 基本項目名 PICTURE X(8).	基本項目または独立項目	—
TIMESTAMP[(q)]	L1 基本項目名 PICTURE X(n).	基本項目または独立項目	q=0 の場合 : n=19 q=2 の場合 : n=21, 22 q=4 の場合 : n=23, 24 q=6 の場合 : n=25, 26

(凡例)

L1 : レベル番号 01~49, または 77

n : 長さ (バイト)

q : 小数秒精度

– : 備考はありません。

表 A-12 SQL のデータ型と COBOL 言語のデータ記述 (変更後)

SQL のデータ型	COBOL 言語のデータ記述	項目の記述	備考
TIME[(q)]	L1 基本項目名 PICTURE X(n).	基本項目または独立項目	q=0 の場合 : n=8 q=2 の場合 : n=11 q=4 の場合 : n=13 q=6 の場合 : n=15
TIMESTAMP[(q)]	L1 基本項目名 PICTURE X(n).	基本項目または独立項目	q=0 の場合 : n=19 q=2 の場合 : n=22 q=4 の場合 : n=24 q=6 の場合 : n=26

(凡例)

L1 : レベル番号 01~49, または 77

n : 長さ (バイト)

q : 小数秒精度

注

太字の部分が変更点です。

(2) 配列を使用する場合

SQL のデータ型と COBOL 言語のデータ記述の対応の、 XDM/RD E2 接続機能を使用時に固有の追加項目 (配列を使用する場合) を次の表に示します。

表 A-13 配列を使用する場合の SQL のデータ型と COBOL 言語のデータ記述 (追加項目)

SQL のデータ型	COBOL 言語のデータ記述	項目の記述	備考
LARGE DECIMAL [(p,s)]	L2 基本項目名 PICTURE S9(p-s)[V9(s)] COMPUTATIONAL-3 OCCURS m TIMES.	OCCURS の指定によって同じデータ構造を繰り返す反復データ項目から構成される集団項目	—
LONG VARCHAR(n)	L2 集団項目名 2 OCCURS m TIMES. L3 基本項目名 1 PICTURE S9(4) COMPUTATIONAL. L3 基本項目名 2 PICTURE X(n).	OCCURS の指定によって同じデータ構造を繰り返す反復データ項目から構成される集団項目	$1 \leq n \leq 32,000$
LONG NVARCHAR(n)	L2 集団項目名 2 OCCURS m TIMES. L3 基本項目名 1 PICTURE S9(4) COMPUTATIONAL. L3 基本項目名 2 PICTURE N(n).	OCCURS の指定によって同じデータ構造を繰り返す反復データ項目から構成される集団項目	$1 \leq n \leq 16,000$
LONG MVARCHAR(n)	L2 集団項目名 2 OCCURS m TIMES. L3 基本項目名 1 PICTURE S9(4) COMPUTATIONAL. L3 基本項目名 2 PICTURE X(n).	OCCURS の指定によって同じデータ構造を繰り返す反復データ項目から構成される集団項目	$1 \leq n \leq 32,000$
INTERVAL YEAR TO SECOND(q)	L2 基本項目名 PICTURE S9(14)[V9(q)] COMPUTATIONAL-3 OCCURS m TIMES.	OCCURS の指定によって同じデータ構造を繰り返す反復データ項目から構成される集団項目	$q=0, 2, 4, 6$ $q=0$ の場合、 [V9(q)] を省略します。 $q > 4$ の場合、 精度 > 18 となるため、 $q=6$ の場合は精度 = 18 で展開されるので、 V9(4) となります。
CLOB	—	—	—
CLOB 位置付け子	—	—	—

(凡例)

L2 : レベル番号 02~49(L2<L3)

L3 : レベル番号 03~49

m : 配列の要素数

n : 長さ (バイト)
 p : 精度 (全体のけた数)
 q : 小数秒精度
 s : 位取り (小数点以下のけた数)
 - : 該当しません。

また、SQL のデータ型と COBOL 言語のデータ記述の変更前と変更後を表「配列を使用する場合の SQL のデータ型と COBOL 言語のデータ記述 (変更前)」および表「配列を使用する場合の SQL のデータ型と COBOL 言語のデータ記述 (変更後)」に示します。

表 A-14 配列を使用する場合の SQL のデータ型と COBOL 言語のデータ記述 (変更前)

SQL のデータ型	COBOL 言語のデータ記述	項目の記述	備考
TIME	L2 基本項目名 PICTURE X(8) OCCURS m TIMES.	OCCURS の指定によって同じデータ構造を繰り返す反復データ項目から構成される集団項目	-
TIMESTAMP[(q)]	L2 基本項目名 PICTURE X(n) OCCURS m TIMES.		q=0 の場合 : n=19 q=2 の場合 : n=21, 22 q=4 の場合 : n=23, 24 q=6 の場合 : n=25, 26

(凡例)

L2 : レベル番号 02~49(L2<L3)

m : 配列の要素数

n : 長さ (バイト)

q : 小数秒精度

- : 備考はありません。

表 A-15 配列を使用する場合の SQL のデータ型と COBOL 言語のデータ記述 (変更後)

SQL のデータ型	COBOL 言語のデータ記述	項目の記述	備考
TIME[(q)]	L2 基本項目名 PICTURE X(n) OCCURS m TIMES.	OCCURS の指定によって同じデータ構造を繰り返す反復データ項目から構成される集団項目	q=0 の場合 : n=8 q=2 の場合 : n=11 q=4 の場合 : n=13 q=6 の場合 : n=15
TIMESTAMP[(q)]	L2 基本項目名 PICTURE X(n) OCCURS m TIMES.		q=0 の場合 : n=19 q=2 の場合 : n=22 q=4 の場合 : n=24 q=6 の場合 : n=26

(凡例)

L2 : レベル番号 02~49(L2<L3)

m : 配列の要素数

n : 長さ (バイト)

q : 小数秒精度

注

太字の部分が変更点です。

(3) 繰返し列を使用する場合

SQL のデータ型と COBOL 言語のデータ記述の対応の、 XDM/RD E2 接続機能を使用時に固有の追加項目（繰返し列を使用する場合）を次の表に示します。

表 A-16 繰返し列を使用する場合の SQL のデータ型と COBOL 言語のデータ記述（追加項目）

SQL のデータ型	COBOL 言語のデータ記述	項目の記述	備考
LARGE DECIMAL [(p[, s])]	L2 集団項目名. L3 基本項目名 1 PICTURE S9(9) COMPUTATIONAL. L3 基本項目名 2 PICTURE S9(p-s)[V9(s)] COMPUTATIONAL-3 OCCURS m TIMES. (P=S のときは SV9(p)とする)	二つの基本項目から構成される集団項目	—
INTERVAL YEAR TO SECOND(q)	L2 集団項目名. L3 基本項目名 1 PICTURE S9(9) COMPUTATIONAL. L3 基本項目名 2 PICTURE S9(14)[V9(q)] COMPUTATIONAL-3 OCCURS m TIMES.	二つの基本項目から構成される集団項目	q=0, 2, 4, 6 q=0 の場合、 [V9(q)] を省略します。 q>4 の場合、 精度>18 となるため、 q=6 の場合は精度=18 で展開されるので、 V9(4) となります。
CLOB	—	—	—
CLOB 位置付け子	—	—	—

(凡例)

L2 : レベル番号 02~49(L2<L3)

L3 : レベル番号 03~49

m : 繰返し列の最大要素数

p : 精度 (全体のけた数)

q : 小数秒精度

s : 位取り (小数点以下のけた数)

－：該当しません。

また、SQL のデータ型と COBOL 言語のデータ記述の変更前と変更後を表「繰返し列を使用する場合の SQL のデータ型と COBOL 言語のデータ記述（変更前）」および表「繰返し列を使用する場合の SQL のデータ型と COBOL 言語のデータ記述（変更後）」に示します。

表 A-17 繰返し列を使用する場合の SQL のデータ型と COBOL 言語のデータ記述（変更前）

SQL のデータ型	COBOL 言語のデータ記述	項目の記述	備考
VARCHAR(n) /MVARCHAR(n)	L2 集団項目名. L3 基本項目名 1 PICTURE S9(9) COMPUTATIONAL. L3 集団項目名 2 OCCURS m TIMES. L4 基本項目名 3 PICTURE S9(4) COMPUTATIONAL. L4 基本項目名 4 PICTURE X(n).	二つの基本項目から構成される集団項目と、一つの基本項目から構成される集団項目	$1 \leq n \leq 32,000$
NVARCHAR(n)	L2 集団項目名. L3 基本項目名 1 PICTURE S9(9) COMPUTATIONAL. L3 集団項目名 2 OCCURS m TIMES. L4 基本項目名 3 PICTURE S9(4) COMPUTATIONAL. L4 基本項目名 4 PICTURE N(n).	二つの基本項目から構成される集団項目と、一つの基本項目から構成される集団項目	$1 \leq n \leq 16,000$
TIME	L2 集団項目名. L3 基本項目名 1 PICTURE S9(9) COMPUTATIONAL. L3 基本項目名 2 PICTURE X(8) OCCURS m TIMES.	二つの基本項目から構成される集団項目	－
TIMESTAMP[(q)]	L2 集団項目名. L3 基本項目名 1 PICTURE S9(9) COMPUTATIONAL. L3 基本項目名 2	二つの基本項目から構成される集団項目	$q=0$ の場合 : $n=19$ $q=2$ の場合 : $n=21, 22$ $q=4$ の場合 : $n=23, 24$ $q=6$ の場合 : $n=25, 26$

SQL のデータ型	COBOL 言語のデータ記述	項目の記述	備考
	PICTURE X(n) OCCURS m TIMES.		

(凡例)

L2 : レベル番号 02~49

L3, L4 : レベル番号 03~49

m : 繰返し列の最大要素数

n : 長さ (バイト)

q : 小数秒精度

- : 備考はありません。

表 A-18 繰返し列を使用する場合の SQL のデータ型と COBOL 言語のデータ記述 (変更後)

SQL のデータ型	COBOL 言語のデータ記述	項目の記述	備考
VARCHAR(n) /LONG VARCHAR(n) /MVARCHAR(n) /LONG MVARCHAR(n)	L2 集団項目名. L3 基本項目名 1 PICTURE S9(9) COMPUTATIONAL. L3 集団項目名 2 OCCURS m TIMES. L4 基本項目名 3 PICTURE S9(4) COMPUTATIONAL. L4 基本項目名 4 PICTURE X(n).	二つの基本項目から構成される集団項目と、一つの基本項目から構成される集団項目	$1 \leq n \leq 32,000$
NVARCHAR(n) /LONG NVARCHAR(n)	L2 集団項目名. L3 基本項目名 1 PICTURE S9(9) COMPUTATIONAL. L3 集団項目名 2 OCCURS m TIMES. L4 基本項目名 3 PICTURE S9(4) COMPUTATIONAL. L4 基本項目名 4 PICTURE N(n).	二つの基本項目から構成される集団項目と、一つの基本項目から構成される集団項目	$1 \leq n \leq 16,000$
TIME[(q)]	L2 集団項目名. L3 基本項目名 1 PICTURE S9(9) COMPUTATIONAL. L3 基本項目名 2	二つの基本項目から構成される集団項目	$q=0$ の場合 : $n=8$ $q=2$ の場合 : $n=11$ $q=4$ の場合 : $n=13$ $q=6$ の場合 : $n=15$

SQL のデータ型	COBOL 言語のデータ記述	項目の記述	備考
	PICTURE X(n) OCCURS m TIMES.		
TIMESTAMP[(q)]	L2 集団項目名. L3 基本項目名 1 PICTURE S9(9) COMPUTATIONAL. L3 基本項目名 2 PICTURE X(n) OCCURS m TIMES.	二つの基本項目から構成される集団項目	q=0 の場合 : n=19 q=2 の場合 : n=22 q=4 の場合 : n=24 q=6 の場合 : n=26

(凡例)

L2 : レベル番号 02~49

L3, L4 : レベル番号 03~49

m : 繰返し列の最大要素数

n : 長さ (バイト)

q : 小数秒精度

注

太字の部分が変更点です。

索引

数字

2階層型システム 13

3階層型システム 14

D

DB コネクションサーバ 21

DTP 36

H

HiRDB_PDHOST 54, 67

HiRDB_PDNAMEPOR 54, 67

HiRDB_PDTMID 54, 67

HiRDB_PDXAMODE 54, 68

HiRDB/Developer's Kit 50

HiRDB/Run Time 50

HiRDB XA ライブラリ 36

HiRDB XA ライブラリでサポートする機能 37

HiRDB クライアントからアクセスできない XDM/RD E2 のデータ型 110

HiRDB クライアントからアクセスできる XDM/RD E2 のデータ型 109

HiRDB クライアントの環境設定手順 51

HiRDB クライアントの種類 50

HiRDB の通信処理に関する環境変数 72

hosts ファイルの設定 52

J

JDBC 対応アプリケーションプログラムからの

XDM/RD E2 アクセス 92

O

ODBC 関数に関する環境変数 79

ODBC 対応アプリケーションプログラムからの
XDM/RD E2 アクセス 91

OLTP 下の X/Open に準拠した API を使用するクライアントの環境変数 67

OpenTP1 との関連 (XA インタフェースを使用しない場合) 47

OpenTP1 との関連 (XA インタフェースを使用する場合) 35

P

PDADDITIONALOPTLVL 61

PDAGGR 61

PDASTHOST 56

PDASTPORT 56

PDASTUSER 56

PDAUTHTYPE 56

PDAUTOCONNECT 61, 77

PDAUTORECONNECT 55, 70

PDBESCONHOLD 63

PDBESCONHTI 63

PDBINARYBLKF 62

PDBINDRETRYCOUNT 62

PDBINDRETRYINTERVAL 62

PDBLKBUFSIZE 62

PDBLK 62, 77

PDCALCMDWAITTIME 61

PDCLTAPNAME 54, 68

PDCLTBINDLOOPBACKADDR 56

PDCLTCNVBYTERATIO 55

PDCLTCNVMODE 55

PDCLTCNVUOCFUNC 55

PDCLTCNVUOCLIB 55

PDCLTGAIJIDLL 55

PDCLTGAIJIFUNC 55

PDCLTGRP 55

PDCLTLANG 54, 68

PDCLTPATH 58, 72

PDCLTRCVADDR 53

PDCLTRCVPORT 53

PDCLTRDNODE 62, 78

PDCMDTRACE 57

PDCMDWAITTIME 57

PDCMMTBFDDL 61

PDCNSTRNTNAME 63
PDCONNECTWAITTIME 58
PDCONREFRCOUNT 57
PDCONREFINTERVAL 57
PDCONTYPE 54
PDCURSORLVL 61
PDCWAITTIME 57, 71
PDCWAITTIMEWRNPNT 57
PDDBACCS 62
PDDBUFLRU 56
PDDBLOG 55
PDDBORGUAP 62
PDDDLDEAPRP 61
PDDDLDEAPRPEXE 61
PDDEFAULTOPTION 56, 71
PDDELRSVWDFILE 61
PDDFLINVAL 61
PDDLKPRIOR 60
PDDNDPCOMPATIBLE 65
PDDNDPTRACE 59
PDERRSKIPCODE 58, 73
PDEXTDECHECK 56
PDEXWARN 55, 69
PDFESGRP 53
PDFESHOST 53
PDFORUPDATEEXLOCK 60
PDGDATAOPT 63, 80
PDHASHTBLSIZE 61
PDHATRNQUEUING 56
PDHOST 53, 66
PDHSICOPTIONS 64
PDIPC 57
PDIPCFILERDIR 56
PDISLLVL 60
PDJDBFILEDIR 64
PDJDBFILEOUTNUM 65
PDJDBFILESIZE 65
PDJDBONMEMNUM 65
PDJDBSQLTRCMEMNUM 59
PDJDBTRACELEVEL 65
PDJETCOMPATIBLE 63, 82
PDKALVL 57
PDKATIME 57
PDLANG 55, 69
PDLOCKWAITTIME 61
PDLOCKLIMIT 60
PDLOCKSKIP 60
PDNAMEPORT 53, 66
PDNBLOCKWAITTIME 57, 72
PDODBCWRNSKIP 63, 81
PDODBERRTRC 65, 82
PDODBERRTRCSIZE 65, 82
PDODBESCAPE 63, 80
PDODBINFOSUPPRESS 64
PDODBLOCATOR 63, 80
PDODBSPLITSIZE 63, 81
PDODBSTANDARDARGSIZE 64
PDODBSTANDARDDESCCOL 64
PDODBSTANDARDGTYPEINFO 64
PDODBSTANDARDSQLSTATE 64
PDODBSTATCACHE 63, 79
PDPLGIXMK 64
PDPLGPFSZ 64
PDPLGPFSZEXP 64
PDPLUGININNSUB 64
PDPRMTRC 58, 73
PDPRMTRCSIZE 58, 74
PDPRPCRCLS 61, 77
PDRCOUNT 55, 70
PDRCINTERVAL 55, 70
PDRCTIMING 56
PDRCTRACE 58
PDRDCLTCODE 63, 78
PDRECMEMSIZE 57
PDREPPATH 58, 75
PDSENDMEMSIZE 57
PDSERVICEGRP 53
PDSERVICEPORT 53

PDSOCKETCRYPTO 56
PDSPACELEVEL 62
PDSQLEXECTIME 58, 76
PDSQLOPTLVL 60
PDSQLTEXTSIZE 58, 75
PDSQLTRACE 58, 72
PDSQLTRCBUFFSIZE 59
PDSQLTRCFMT 59, 76
PDSQLTRCOPENMODE 58, 75
PDSQLTRCTYPE 59
PDSRVTYPE 53, 66
PDSTANDARDSQLSTATE 62
PDSTJTRNOUT 60
PDSUBSTRLEN 55
PDSWAITTIME 57
PDSWATCHTIME 57
PDSYSTEMID 56
PDTAAPINFMODE 59
PDTAAPINFPATH 59
PDTAAPINFSIZE 59
PDTIMEDOUTRETRY 57
PDTMID 54
PDTMPTBLRDAREA 63
PDTP1SERVICE 63, 78
PDTRCMODE 58, 74
PDTRCPATH 58, 75
PDTXACANUM 54
PDUAPENVFILE 56
PDUAPERLOG 58, 73
PDUAPEXERLOGPRMSZ 59
PDUAPEXERLOGUSE 59
PDUAPREPLVL 58, 74
PDUSER 54, 68
PDVWOPTMODE 59
PDWRTLNCOMSZ 59, 76
PDWRTLNFILSZ 59, 76
PDWRTLNPATH 59, 76
PDXAAUTORECONNECT 54
PDXAMODE 54, 67

PDXARCVWTIME 54, 67
PDXATRCFILEMODE 54, 67

R

RM 36
RM 関連オブジェクト名 44
RM スイッチ名 42
RM 名 42

S

SQL 関連の注意事項 36
SQL 記述領域 (SQLDA) 117
SQL 言語仕様 97
SQL 診断文 127
SQL 制御文 126
SQL トレース機能 94
SQL トレース動的取得機能 96
SQL に関する環境変数 77
SQL の記述 17
SQL の記述形式 98
SQL の機能体系 18
SQL のデータ型と COBOL 言語のデータ記述 138
SQL のデータ型と C 言語のデータ記述 131
SQL のデータ型とデータ記述 84
SQL 領域 113
SQL 連絡領域 (SQLCA) 113

T

TM 36
TP1/Resource Manager Monitor の機能を使用した XDM/RD E2 の監視 47
Type4 JDBC ドライバを使用した XDM/RD E2 アクセス 92

U

UAP 実行時の注意事項 88
UAP 中での SQL の基本構成 24
UAP と XDM/RD E2 とのインターフェース 19
UAP の開発の流れ 16
UAP の記述 29

UAP の記述言語 29

UAP の形式 17

UAP の形態移行 84

UAP の作成 84

UAP の作成、実行 83

UAP の実行 86

UAP の障害対策 94

UAP の設計 23

X

X/Open XA インタフェース 36

xa_switch_t 構造体名 42

XDM/Library for HiRDB 21

XDM/RD E2 21

XDM/RD E2 からの切り離し 28

XDM/RD E2 接続機能 13

XDM/RD E2 接続機能使用時の環境変数の差異 53

XDM/RD E2 接続機能専用の環境変数 78

XDM/RD E2 接続機能を利用するライブラリ 88

XDM/RD E2 との接続 25

い

位置付け子機能 34

一相最適化 37

インターフェース 18

インターフェース領域の種類と使用目的 30

う

埋込み型 17, 29

埋込み型 UAP 13

埋込み言語 128

埋込み言語文法 128

埋込み変数と標識変数の宣言 24

え

エラー時の処置の指定 27

エラーの判定 27

お

オープン文字列 42

か

カーソル宣言 26

概要 12

型名記述領域 117

稼働 OS 21

く

クライアント環境定義 45

クライアント環境定義（環境変数の設定） 53

クライアント環境定義の設定内容 66

クライアントの環境設定 49

クローズ文字列 44

け

検索、更新の SQL（実行文） 27

こ

構成要素の詳細 112

し

システム監視に関する環境変数 71

システム構成に関する環境変数 66

す

スカラ関数 HEX 112

せ

制御系 SQL 124

静的登録方法 41

性能向上、操作性向上に関する UAP の設計 31

そ

操作系 SQL 120

ソースプログラムの記述 17

ソフトウェア構成 21

た

他プログラムプロダクトとの関連 35

て

定義系 SQL 118

データ型 108

データベース操作時の注意事項 121

適用形態 13

と

動作環境 20

動的登録方法 40

動的トランザクションの登録 38

特長 13

トラブルシュートに関する環境変数 72

トランザクションの移行 37, 39

トランザクションの無効化 28

トランザクションの有効化 28

トランザクションマネージャ 36

トランザクションマネージャと XDM/RD E2 間のコネクションが切断されたときの再接続方法 46

トランザクションマネージャに登録する情報 42

トランザクションマネージャへの登録 40

トランザクションマネージャへの登録の変更 46

トランザクションマネージャへの登録例 45

は

配列を使用した FETCH 機能 31

ひ

非同期 XA 呼び出し 38

ふ

複数接続機能 31, 38

プリプロセス 86

プログラムプロダクト構成 21

ロック転送 77

ロック転送機能に関する環境変数 77

分散トランザクション処理 36

ま

マルチスレッド用のライブラリに関する注意事項 36

め

明示表 123

ゆ

ユーザ実行環境に関する環境変数 68

よ

読み取り専用 37

り

リソースマネージャ 36

れ

列名記述領域 (SQLNAME) 117