

DABroker

解説・操作書

3020-6-031-G0

前書き

■ 対象製品

P-2463-2354 DABroker 03-24 (適用 OS : Windows2000, Windows XP Professional, Windows XP Professional x64 Edition, Windows Server 2003, Windows Server 2003 x64 Edition, Windows Server 2003 R2, Windows Server 2003 R2 x64 Edition, Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, Windows Server 2016, Windows Sever 2019, Windows Vista, Windows 7, Windows 8, Windows 8.1, Windows 10)

P-2463-2364 DABroker 03-24 (適用 OS : Windows2000, Windows XP Professional, Windows XP Professional x64 Edition, Windows Server 2003, Windows Server 2003 x64 Edition, Windows Server 2003 R2, Windows Server 2003 R2 x64 Edition, Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, Windows Server 2016, Windows Sever 2019, Windows Vista, Windows 7, Windows 8, Windows 8.1, Windows 10)

P-8363-2451 DABroker 03-24 (適用 OS : Red Hat Enterprise Linux Server 7 (64-Bit x86_64), Red Hat Enterprise Linux Server 8 (64-Bit x86_64))

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

HITACHI, DABroker, DBPARTNER, HiRDB, HITSENSER, OpenTP1, TPBroker, VOS3/LS, VOS3/US, VOS3/XS および XDM は、株式会社 日立製作所の商標または登録商標です。

AIX は、世界の多くの国で登録された International Business Machines Corporation の商標です。

Intel は、Intel Corporation またはその子会社の商標です。

Linux は、Linus Torvalds 氏の日本およびその他の国における登録商標または商標です。

Microsoft は、マイクロソフト 企業グループの商標です。

Oracle および Java は、オラクルおよびその関連会社の登録商標です。

Red Hat is a registered trademark of Red Hat, Inc. in the United States and other countries.

Red Hat は、米国およびその他の国における Red Hat, Inc. の登録商標です。

Red Hat Enterprise Linux is a registered trademark of Red Hat, Inc. in the United States and other countries.

Red Hat Enterprise Linux は、米国およびその他の国における Red Hat, Inc.の登録商標です。

SQL Server は、マイクロソフト 企業グループの商標です。

UNIX は、The Open Group の商標です。

Visual Basic は、マイクロソフト 企業グループの商標です。

Visual Studio は、マイクロソフト 企業グループの商標です。

Windows は、マイクロソフト 企業グループの商標です。

Windows Server は、マイクロソフト 企業グループの商標です。

Windows Vista は、マイクロソフト 企業グループの商標です。

その他記載の会社名、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。

■ マイクロソフト製品の表記について

このマニュアルでは、マイクロソフト製品の名称を次のように表記しています。

製品名	表記	
Microsoft SQL Server	SQL Server	
Microsoft Windows XP	Windows XP	Windows
Microsoft Windows Vista Business	Windows Vista	
Microsoft Windows Vista Enterprise		
Microsoft Windows Vista Ultimate		
Microsoft Windows 7 Professional	Windows 7	
Microsoft Windows 7 Enterprise		
Microsoft Windows 7 Ultimate		
Microsoft Windows 8 Pro	Windows 8	
Microsoft Windows 8 Enterprise		
Microsoft Windows 8.1 Pro		
Microsoft Windows 8.1 Enterprise		
Microsoft Windows 10 Pro	Windows 10	
Microsoft Windows 10 Enterprise		
Microsoft Windows Server 2008 Standard	Windows Server 2008	
Microsoft Windows Server 2008 Enterprise		
Microsoft Windows Server 2008 Datacenter		

製品名	表記	
Microsoft Windows Server R2 2008 Standard	Windows Server 2008 R2	
Microsoft Windows Server R2 2008 Enterprise		
Microsoft Windows Server R2 2008 Datacenter		
Microsoft Windows Server 2003, Standard Edition	Windows Server 2003	
Microsoft Windows Server 2003, Enterprise Edition		
Microsoft Windows Server 2003, Datacenter Edition		
Microsoft Windows Server 2003 R2, Standard Edition	Windows Server 2003 R2	
Microsoft Windows Server 2003 R2, Enterprise Edition		
Microsoft Windows Server 2003 R2, Datacenter Edition		
Microsoft Windows 2000 Professional Operating System	Windows 2000	
Microsoft Windows 2000 Server Operating System		
Microsoft Windows 2000 Advanced Server Operating System		
Microsoft Windows 2000 Datacenter Server Operating System		
Microsoft Windows Server 2012 Standard	Windows Server 2012	
Microsoft Windows Server 2012 Datacenter		
Microsoft Windows Server 2012 R2 Standard		
Microsoft Windows Server 2012 R2 Datacenter		
Microsoft Windows Server 2016 Standard	Windows Server 2016	
Microsoft Windows Server 2016 Datacenter		
Microsoft Windows Server 2019 Standard	Windows Server 2019	
Microsoft Windows Server 2019 Datacenter		

■ 発行

2021年12月 3020-6-031-G0

■ 著作権

All Rights Reserved. Copyright (C) 1998, 2021, Hitachi, Ltd.

変更内容

変更内容 (3020-6-031-G0) DABroker 03-24

追加・変更内容	変更箇所
Oracle19c をサポートしました。	1.2.3, 3.5.1(6), 4.5.1(6), 4.9.2(3), 6.4.2, 6.4.3(4), 6.4.3(5), 6.5.2(4)
次に示すメッセージを追加しました。 KFDB027DB-E	7.3

変更内容 (3020-6-031-G0) DABroker 03-23

追加・変更内容	変更箇所
DBPARTNER2 Client, DBPARTNER ODBC Driver から HiRDB の Timestamp 属性を使用できるようにしました。	—
Oracle12c R2 をサポートしました。	3.5.1(6)
HiRDB V10 をサポートしました。	付録 B

変更内容 (3020-6-031-G0) DABroker 03-22

追加・変更内容	変更箇所
Oracle12c をサポートしました。	1.2.3, 3.5.1(6), 3.11, 4.5.1(6), 4.9.2(3), 6.4.2
DABroker セットアップコマンドの指定値「-d」の説明を変更しました。	3.2.3(1)(a)
UNIX 上で DBPARTNER, HITSENSER5, DBPARTNER Library, 又は XDM/RD 外部表を使用する場合の, ネットワーク環境の設定方法の説明を変更しました。	3.3.1
Windows 上で DBPARTNER, HITSENSER5, DBPARTNER Library, 又は XDM/RD 外部表を使用する場合の, ネットワーク環境の設定方法の説明を変更しました。	4.3.1
Windows 上で Database Connection Server を経由してメインフレーム系データベースにアクセスする場合の, ネットワーク環境の設定方法の説明を変更しました。	4.3.2
次に示すメッセージの説明を変更しました。 KFDB001TM-E, KFDB012YM-E	7.3

単なる誤字・脱字などはお断りなく訂正しました。

はじめに

このマニュアルは、次に示すプログラムプロダクトの機能と使い方について説明したものです。

- P-2463-2354 DABroker
- P-2463-2364 DABroker
- P-8363-2451 DABroker

■ 対象読者

このマニュアルは、DABroker を使用してデータベースにアクセスする方を対象としています。

■ マニュアルの構成

このマニュアルは、次に示す章と付録から構成されています。

第 1 章 DABroker の概要 (共通)

DABroker 及び DABroker 関連製品の使用目的及び特長について説明しています。

第 2 章 DABroker の機能 (共通)

DABroker 及び DABroker 関連製品の使用目的を実現する機能について説明しています。

第 3 章 DABroker の環境設定と運用 (UNIX)

UNIX で使用する場合に、DABroker を起動する前に必要な環境設定、及び DABroker の運用方法、運用コマンドについて説明しています。

第 4 章 DABroker の環境設定と運用 (Windows)

Windows で使用する場合に、DABroker を起動する前に必要な環境設定、及び DABroker の運用方法、ユーティリティについて説明しています。

第 5 章 DABroker の運用 (共通)

DABroker の運用方法について説明しています。

第 6 章 DABroker の障害対策 (共通)

DABroker の障害時にユーザが取得する情報について説明しています。

第 7 章 メッセージ (共通)

DABroker が発行するメッセージの要因と対処について説明しています。

付録 A DIAPRISM アクセス時の環境設定方法 (Windows)

HITSENER5 から DIAPRISM へアクセスする場合の、DABroker の環境設定方法について説明しています。

付録 B このマニュアルの参考情報

関連マニュアル、このマニュアルでの表記、KB (キロバイト) などの単位表記について説明しています。

付録 C 用語解説 (共通)

DABroker で使用する用語について説明しています。

■ 読書手順

このマニュアルは、利用目的に合わせて読む章を選択できます。利用目的別にお読みいただくことをお勧めします。



(凡例)

▼ : 必ず読む項目

▽ : 必要に応じて読む項目

■ このマニュアルで使用している記号

このマニュアルで使用している記号について、次に示します。

記号	説明
{ }	この記号で囲まれている複数の項目のうちから一つを選択することを示します。項目が縦に複数行にわたって記述されている場合は、そのうちの1行分を選択します。項目が横に並べられ、記号' 'で区切られている場合は、そのうちの一つを選択します。 (例) {A B C}ではA、B又はCのどれか一つを指定することを示します。
[]	この記号で囲まれている項目は省略してもよいことを意味します。複数の項目が縦又は横に並べて記述されている場合には、すべてを省略するか、記号{ }と同じくどれか一つを選択します。 (例) [A]は「何も指定しない」か、「Aを指定する」ことを示します。 (例) [B C]では「何も指定しない」か、「B又はCを指定する」ことを示します。
...	記述が省略されていることを示します。この記号の直前に示された項目を繰り返し複数個指定できます。 (例) A, B, B, ...では「Aの後ろにBを必要個数指定する」ことを示します。
-	括弧で囲まれた複数項目のうち1項目に対し使用され、括弧内のすべてを省略したときシステムがとる標準値を示します。 (例) [A B C]では「A、B、Cのどれも指定しなかった場合、システムはCが指定されたときと同じ処理をする」ことを示します。 (例) [A={B C D}]では「オペランドAを指定しなかった場合、システムはA=Bを指定されたときと同じ処理をする」ことを示します。
< >	項目を記述するとき従わなければならない構文要素を示します。
△	1個の半角の空白を示します。
△ _n	n個以上の半角の空白を示します。
↓計算式↓	計算式の答えの小数点以下を切り捨てることを示します。
↑計算式↑	計算式の答えの小数点以下を切り上げることを示します。

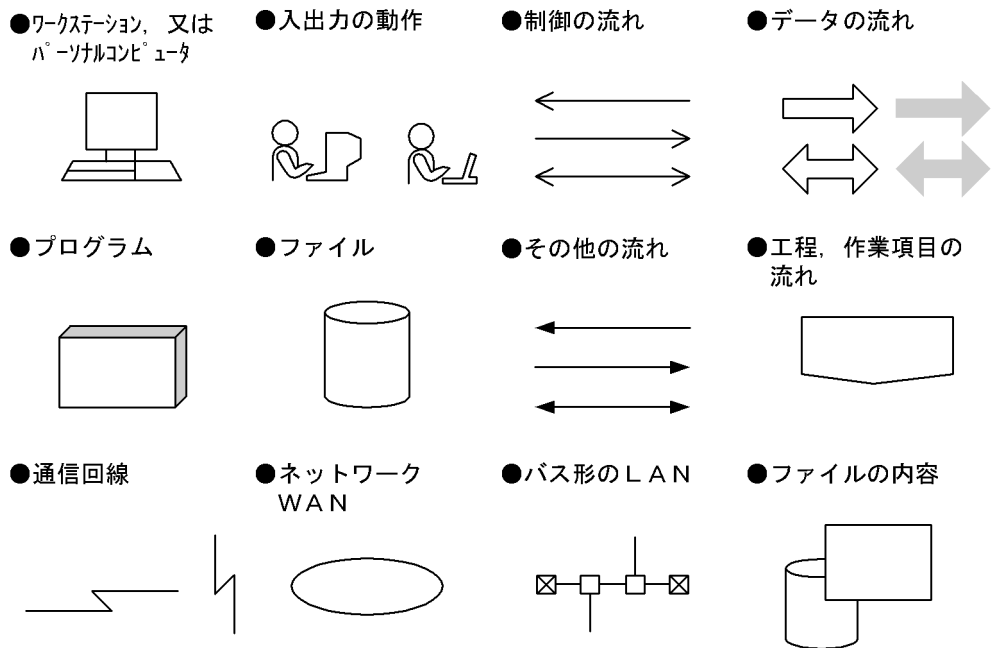
■ このマニュアルで使用する構文要素

このマニュアルで使用する構文要素の種類を、次に示します。

種類	定義
<英字>	A~Z a~z
<英大文字>	A~Z @ # ¥
<カナ文字>	ア~ン ヲ カナ小文字 ` ° ー
<数字>	0~9
<数値>	<数字> . + -
<英数字>	<英字>及び<数字>
<漢字>	2バイトコードの文字列

■ 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。



目次

前書き	2
変更内容	5
はじめに	6

1	DABroker の概要 (共通)	16
1.1	DABroker とは	17
1.1.1	マルチスレッドによる高性能な処理を実現	17
1.1.2	複数 DBMS での共通インタフェースを提供	17
1.1.3	X/Open DTP モデル XA インタフェースを使用した分散トランザクションを実現	18
1.2	DABroker の使用環境	19
1.2.1	アプリケーション開発用インタフェースが豊富	21
1.2.2	DBPARTNER, HITSENSER5, DBPARTNER Library, XDM/RD 外部表からのアクセスをサポート	22
1.2.3	アクセスできるデータベースの種類が豊富	22
1.2.4	分散トランザクション処理	24
1.3	アプリケーション開発用インタフェースの使用	30
1.3.1	DABroker for C++で開発するアプリケーション	30
1.3.2	DABroker for COBOL で開発するアプリケーション	31
1.3.3	DABroker for Java で開発するアプリケーション	32
1.3.4	DABroker for ORB で開発するアプリケーション	37
1.3.5	DABroker for ODBC で開発するアプリケーション	38
1.3.6	DABroker for .NET Framework で開発するアプリケーション	41
1.4	DBPARTNER, HITSENSER5 などからのアクセス	44
1.4.1	DBPARTNER/Client を利用する場合	45
1.4.2	DBPARTNER/Web を利用する場合	45
1.4.3	DBPARTNER Mobile を利用する場合	46
1.4.4	HITSENSER5 を利用する場合	48
1.4.5	HITSENSER5 Web を利用する場合	49
1.4.6	DBPARTNER Library を利用する場合	49
1.4.7	XDM/RD E2 の外部表を利用する場合	50
2	DABroker の機能 (共通)	51
2.1	マルチスレッドの実行	52
2.1.1	アプリケーションから実行するマルチスレッド	53
2.1.2	DBPARTNER から実行するマルチスレッド	54

- 2.2 データベースアクセス 57
 - 2.2.1 同期アクセス処理 57
 - 2.2.2 非同期アクセス処理 57
 - 2.2.3 DBPARTNER からの予約処理 57
 - 2.2.4 DBPARTNER からのエージェント機能の実行 58
 - 2.2.5 分散トランザクション処理 59
- 2.3 DABroker との接続 60
 - 2.3.1 アプリケーションと DABroker との接続 60
 - 2.3.2 DBPARTNER, HITSENSER5, DBPARTNER Library, 又は XDM/RD 外部表と DABroker との接続 61

3 DABroker の環境設定と運用 (UNIX) 62

- 3.1 環境設定の手順 63
 - 3.1.1 DABroker 関連製品を使用する場合の環境設定 63
 - 3.1.2 DBPARTNER, HITSENSER5, DBPARTNER Library, 又は XDM/RD 外部表を使用する場合の環境設定 64
- 3.2 DABroker のインストールとセットアップ 66
 - 3.2.1 DABroker をインストールするマシン 66
 - 3.2.2 DABroker のインストール 66
 - 3.2.3 DABroker のセットアップ 67
- 3.3 ネットワーク環境の設定 72
 - 3.3.1 DBPARTNER, HITSENSER5, DBPARTNER Library, 又は XDM/RD 外部表を使用する場合 72
 - 3.3.2 Database Connection Server を経由してメインフレーム系データベースにアクセスする場合 72
- 3.4 DABroker の動作環境の設定 74
 - 3.4.1 DABroker 動作環境定義ファイルの作成 74
- 3.5 DABroker の動作環境の設定項目 76
 - 3.5.1 共通項目 76
 - 3.5.2 リモートアクセス設定項目 85
 - 3.5.3 イベントトレースの設定 92
 - 3.5.4 HiRDB の設定 97
 - 3.5.5 ORACLE の設定 98
 - 3.5.6 Database Connection Server 接続設定項目 98
 - 3.5.7 エージェント処理機能の設定 100
 - 3.5.8 その他の設定 107
- 3.6 DABroker 接続先データベースの定義 110
 - 3.6.1 接続先データベース定義ファイルの作成 110
 - 3.6.2 接続先データベース定義ファイルでの設定項目 111
 - 3.6.3 接続先データベース定義ファイルの作成例 116
- 3.7 DABroker の起動と終了 119
 - 3.7.1 DABroker の起動 119

- 3.7.2 DABroker の終了 120
- 3.8 DABroker の処理内容の表示 123
 - 3.8.1 動作状態の表示 123
 - 3.8.2 クライアントとの通信トレースの表示 124
 - 3.8.3 エージェント登録情報の表示 128
 - 3.8.4 エージェント実行情報の表示 129
- 3.9 DABroker コネクションマルチ定義(Database Connection Server 経由の DB アクセス定義) 131
 - 3.9.1 コネクションマルチ定義ファイルの作成 131
 - 3.9.2 コネクションマルチ名の設定 132
 - 3.9.3 最大仮想コネクション数などを設定するときのガイドライン 133
 - 3.9.4 コネクションマルチ定義ファイルでの設定項目 136
 - 3.9.5 コネクションマルチ定義ファイルの作成例 138
- 3.10 コネクションマルチ機能の使用 139
 - 3.10.1 コネクションマルチ機能の運用開始と終了 139
 - 3.10.2 コネクションマルチ運用ユティリティ 140
- 3.11 ORACLE アクセスライブラリ再作成方法 142
 - 3.11.1 HP-UX の場合 142
 - 3.11.2 AIX の場合 154
 - 3.11.3 Solaris の場合 162
 - 3.11.4 Red Hat Linux の場合 170
 - 3.11.5 Oracle9i 以降で追加されたデータ型への対応状況 174

4 DABroker の環境設定と運用 (Windows) 176

- 4.1 環境設定の手順 177
 - 4.1.1 DABroker 関連製品を使用する場合の環境設定 177
 - 4.1.2 DBPARTNER, HITSENSER5, DBPARTNER Library, XDM/RD 外部表を使用する場合の環境設定 178
 - 4.1.3 環境設定方法 179
- 4.2 DABroker のインストール 180
 - 4.2.1 DABroker をインストールするマシン 180
 - 4.2.2 インストール 180
 - 4.2.3 DABroker のファイル 181
- 4.3 ネットワーク環境の設定 182
 - 4.3.1 DBPARTNER, HITSENSER5, DBPARTNER Library, 又は XDM/RD 外部表を使用する場合 182
 - 4.3.2 Database Connection Server を経由してメインフレーム系データベースにアクセスする場合 183
- 4.4 DABroker 環境設定 184
 - 4.4.1 環境設定ユティリティの操作 184
- 4.5 DABroker 環境設定項目 186
 - 4.5.1 共通設定の項目 186
 - 4.5.2 リモートアクセス設定 192

- 4.5.3 エージェント設定 200
- 4.5.4 for C++, for COBOL, for ORB 設定 206
- 4.5.5 Database Connection Server 接続設定 209
- 4.5.6 その他の設定 210
- 4.6 DABroker 接続先データベース定義 213
- 4.6.1 接続先データベース定義ユーティリティの操作 213
- 4.6.2 設定する項目 215
- 4.7 DABroker コネクションマルチ定義(Database Connection Server 経由の DB アクセス定義) 221
- 4.7.1 コネクションマルチ名の設定 221
- 4.7.2 最大仮想コネクション数などを設定するときのガイドライン 222
- 4.7.3 コネクションマルチ定義ユーティリティの操作 226
- 4.7.4 設定する項目 227
- 4.8 DABroker の起動と終了 230
- 4.8.1 起動 230
- 4.8.2 終了 230
- 4.9 DABroker の処理内容の表示 233
- 4.9.1 DABroker 管理ユーティリティの実行 233
- 4.9.2 DABroker トレース出力ユーティリティの実行 235
- 4.9.3 DABroker エージェント管理ユーティリティの実行 244
- 4.10 コネクションマルチ機能の使用 248
- 4.10.1 コネクションマルチ機能の運用開始と終了 248
- 4.10.2 コネクションマルチ管理ユーティリティ 249

5 DABroker の運用 (共通) 252

- 5.1 DABroker を運用する前に 253
- 5.1.1 データベースにアクセスするユーザ ID の設定 253
- 5.1.2 クライアントからの接続先の指定 253
- 5.1.3 エージェント機能を使用するユーザのグループ管理 254
- 5.2 DBPARTNER を使用する場合のスレッドのチューニング 257
- 5.3 DBPARTNER を使用する場合のエージェントの実行権限 258
- 5.3.1 クライアントで指定するユーザ ID・パスワードと DBMS との関係 258
- 5.3.2 エージェントの登録, 削除 258
- 5.3.3 エージェントの実行, 実行結果の削除 258
- 5.3.4 エージェント実行権限の設定 258
- 5.3.5 エージェント実行結果の取得権限の設定 259
- 5.3.6 エージェントの実行権限一覧 259
- 5.4 エージェント機能で使用するファイルの運用 262
- 5.4.1 ファイルのバックアップ 262
- 5.4.2 エージェント登録情報及び実行情報ファイルについて 262

- 5.4.3 ファイルの削除 263
- 5.4.4 エージェント実行名の割り当て規則 264
- 5.4.5 エージェント機能のパラメーター一覧及びファイル容量の見積もり (共通) 265

6 DABroker の障害対策 (共通) 271

- 6.1 障害対策 272
 - 6.1.1 エラー時に DABroker が実行する処理 272
 - 6.1.2 障害情報の取得 272
- 6.2 DABroker でのタイマー監視 278
 - 6.2.1 DBPARTNER と DABroker の接続時のタイマー監視 278
 - 6.2.2 エージェントタイマー監視 278
 - 6.2.3 Database Connection Server 接続時のタイマー監視 278
- 6.3 トレースの取得形式 (共通) 280
 - 6.3.1 イベントトレースの取得形式 (共通) 280
 - 6.3.2 エージェントログの取得形式と取得内容 (共通) 281
 - 6.3.3 XA トレースの取得形式 (共通) 285
 - 6.3.4 デモンの生成・消滅・クリーンアップ実行結果ログ (共通) 287
- 6.4 データベースアクセストレース (UNIX) 289
 - 6.4.1 データベースアクセストレース選択オプション 289
 - 6.4.2 データベースアクセストレースの表示 303
 - 6.4.3 拡張データベースアクセストレース機能 306
- 6.5 データベースアクセストレース (Windows) 314
 - 6.5.1 アクセストレース設定 314
 - 6.5.2 拡張データベースアクセストレース機能 329
- 6.6 エラートレース機能 337
 - 6.6.1 機能概要 337
 - 6.6.2 適用範囲 337
 - 6.6.3 出力形式 337
 - 6.6.4 出力例 341
 - 6.6.5 ファイル名称 342
 - 6.6.6 ファイルの運用方法 342

7 メッセージ (共通) 344

- 7.1 メッセージの出力先 345
- 7.2 メッセージの形式及び注意事項 346
 - 7.2.1 メッセージの形式 346
 - 7.2.2 注意事項 346
 - 7.2.3 Windows を使用している場合の注意事項 347
- 7.3 メッセージ一覧 352

7.4	エラーコードとメッセージの対応	532
7.5	メッセージ中に出力されるコード	534
7.5.1	DABroker 詳細コード	534
7.5.2	要求コード/応答コード	534
7.5.3	システムコールのエラー番号	536
7.5.4	X A 要求時のエラー要因	539

付録 540

付録 A	DIAPRISM アクセス時の環境設定方法 (Windows)	541
付録 A.1	DABroker の環境設定	541
付録 B	このマニュアルの参考情報	542
付録 B.1	関連マニュアル	542
付録 B.2	このマニュアルでの表記	545
付録 B.3	KB (キロバイト) などの単位表記について	548
付録 C	用語解説 (共通)	549

索引 555

1

DABroker の概要（共通）

この章では、DABroker の目的及び特長について説明します。

UNIX を使用する環境と Windows を使用する環境で共通の内容です。なお、DABroker は、「ディーエーブローカー」と呼びます。

1.1 DABroker とは

DABroker は、データベースアクセスを実現するミドルウェアです。複数の異なる DBMS に共通のインタフェースでアクセスできる次のような環境を提供します。

- データベースを活用するサーバアプリケーション開発向けインタフェース
- DBPARTNER, HITSENSER5, DBPARTNER Library, 又は XDM/RD 外部表のデータベースアクセスサーバ

次に、DABroker の特長を示します。

1.1.1 マルチスレッドによる高性能な処理を実現

従来のデータベースアクセスを実現するミドルウェアは、アプリケーションからのデータベースアクセス要求をプロセス単位（シングルスレッド）で実行するため、同時に実行するプロセスの増加に伴い、システムリソースの消費が増大し、処理性能を劣化させていました。

DABroker は、アプリケーションからのデータベースアクセス要求をスレッド単位で実行します（マルチスレッド）。マルチスレッドでは、一つのプロセスで使用するメモリ領域で複数のスレッド用のメモリ領域が確保でき、システムリソースの消費を軽減できます。したがって、従来のシングルスレッドでの実行に比べて、システムリソースの利用効率が向上します。

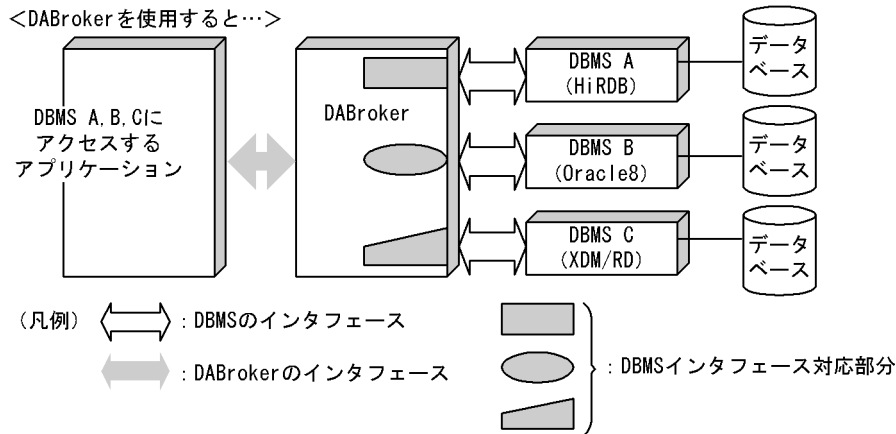
1.1.2 複数 DBMS での共通インタフェースを提供

DBMS が提供するアプリケーション開発用インタフェースは、DBMS ごとに異なります。通常、複数の DBMS にアクセスする場合、それぞれのインタフェースに合わせたアプリケーションを DBMS ごとに開発する必要があります。

DABroker は、複数の異なる DBMS のアプリケーション開発用インタフェースを共通化しています。そのため、DABroker を使用すると、データベースごとのアプリケーション開発用インタフェースの違いを意識することなく、複数のデータベースにアクセスするアプリケーションを開発できます。

ただし、DABroker のアプリケーション開発用インタフェースを使っても、実行する SQL 文の仕様は、各 DBMS に依存します。

図 1-1 DABroker のインタフェースを使用したアプリケーション開発



注 なお、DABrokerからアクセスできるDBMSは、HiRDB、Oracle8、XDM/RDだけではありません。
 DABrokerがサポートするDBMSについては、「1.2.3 アクセスできるデータベースの種類が豊富」を
 参照してください。

1.1.3 X/Open DTP モデル XA インタフェースを使用した分散トランザクションを実現

接続先が次のDBMSの場合、TPBrokerのOTS(Object Transaction Services)機能を利用することで、DABrokerからトランザクションを制御することなく、複数異種のデータベースに対して、X/Open DTPモデルXAインタフェースを使用した分散トランザクションを実現できます。

- HiRDB
- ORACLE

1.2 DABroker の使用環境

DABroker を使用すると、多様な環境でデータベースにアクセスできます。

DABroker を使ったデータベースアクセスには、リモートアクセスとローカルアクセスの 2 種類があります。

リモートアクセスとは、クライアント側でアプリケーションが動作する環境を指します。DBPARTNER, HITSENSER5, DBPARTNER Library, 又は XDM/RD 外部表からのアクセスもリモートアクセスに分類されます。

ローカルアクセスとは、サーバ側でアプリケーションが動作する環境を指します。

DABroker が対応する環境（形態）は次のとおりです。

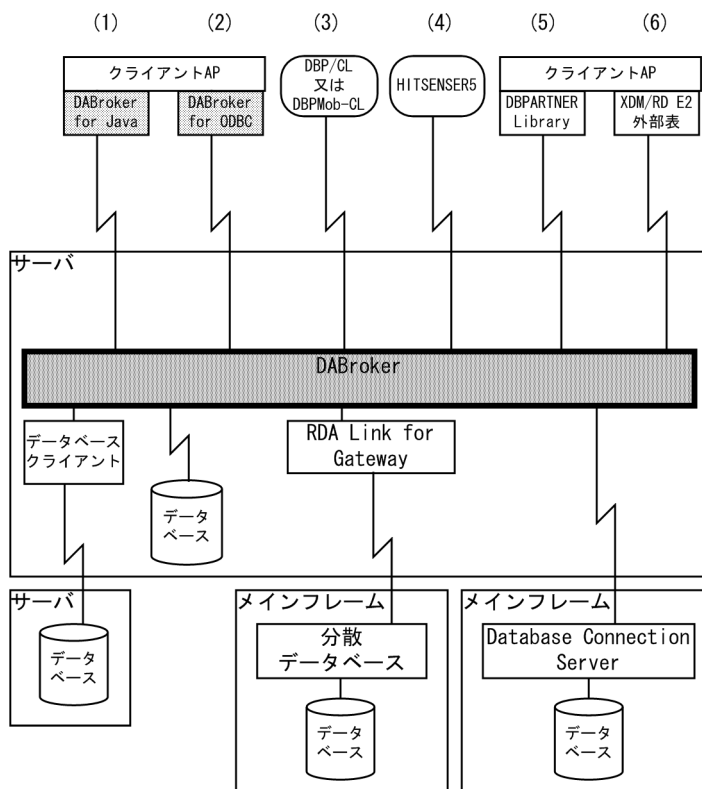
- **クライアント／サーバ環境**

クライアントマシン，サーバマシンの間で TCP/IP などを使用して通信するシステム形態です。クライアント／サーバ環境で DABroker を使用する場合には，図 1-2 で示す(1)～(4)の形態があります。クライアント／サーバ環境は，リモートアクセスに分類されます。

(1)，(2)の形態については「[1.3 アプリケーション開発用インタフェースの使用](#)」を，(3)～(6)の形態については「[1.4 DBPARTNER, HITSENSER5 などからのアクセス](#)」を参照してください。

図 1-2 DABroker の使用環境－1

<クライアント/サーバ環境>



(凡例) : DABroker関連製品

: DBPARTNER関連製品

(表記) DBP/CL : DBPARTNER/Client
DBPMob-CL : DBPARTNER Mobile - Client

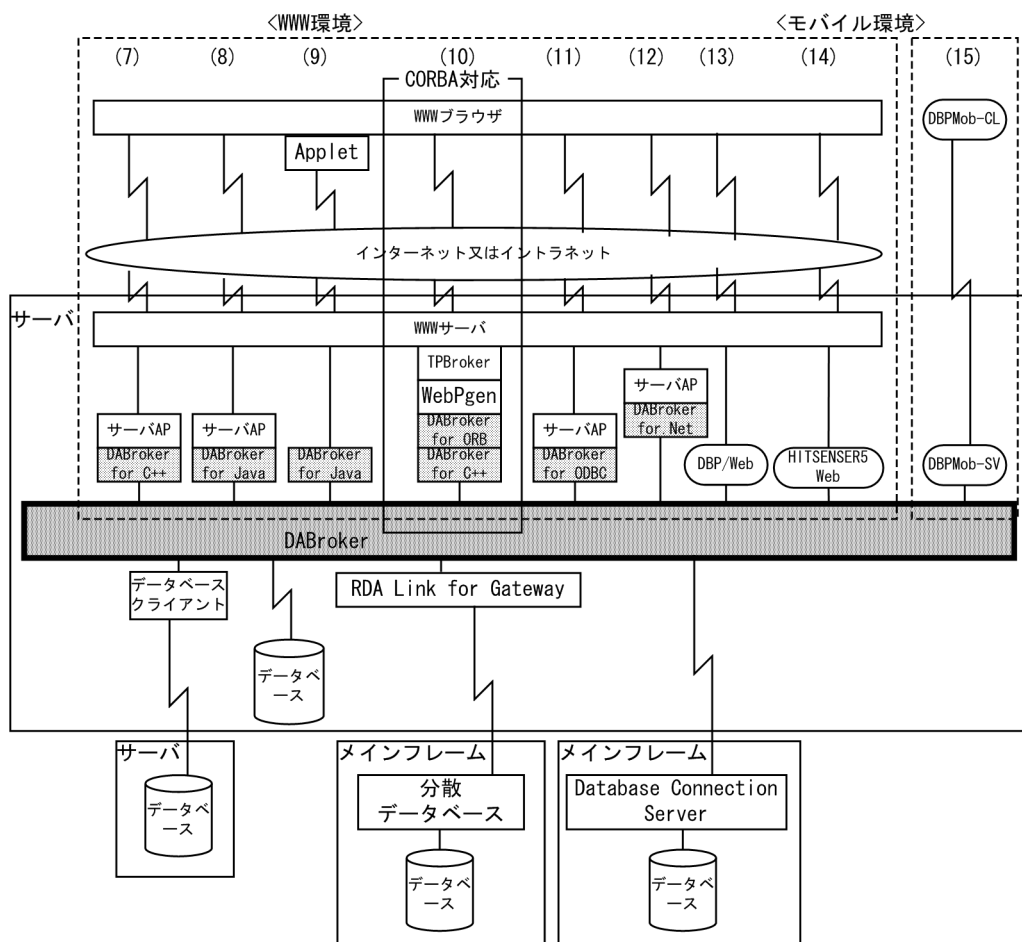
• WWW 環境

インターネット又はイントラネットを利用したシステム形態です。WWW 環境で DABroker を使用する場合には、図 1-3 で示す(7)～(14)の形態があります。WWW 環境は、主にローカルアクセスに分類されます。(13)、(14)の形態だけは、リモートアクセスに分類されます。(7)～(12)の形態については、WWW 環境だけでなく、サーバアプリケーションの内容によって、クライアント/サーバ環境での運用もできます。(7)～(12)の形態については、「1.3 アプリケーション開発用インタフェースの使用」を、(13)、(14)の形態については「1.4 DBPARTNER, HITSENSER5 などからのアクセス」を参照してください。

• モバイル環境

ノート型パーソナルコンピュータなどの携帯端末から公衆回線網を使って、移動先から LAN 内の各種サーバにあるデータを利用する環境をモバイル環境と呼びます。モバイル環境は、リモートアクセスに分類されます。モバイル環境で DABroker を使用する場合には、図 1-3 で示す(15)の形態があります。(15)については、「1.4 DBPARTNER, HITSENSER5 などからのアクセス」を参照してください。

図 1-3 DABroker の使用環境 - 2



- (凡例) DBPXXX : DBPARTNER関連製品
DABroker DABroker for XXX : DABroker関連製品
- (表記) DBP/CL : DBPARTNER/Client
DBPMob-CL : DBPARTNER Mobile - Client
DBPMob-SV : DBPARTNER Mobile - Server
WebPgen : Web Page Generator
サーバAP : サーバアプリケーション

1.2.1 アプリケーション開発用インタフェースが豊富

DABroker 関連製品では、サーバアプリケーションを開発するためのインタフェースを提供します。提供するインタフェースは、多様な開発言語に対応しています。したがって、開発環境に応じた言語を選択できます。DABroker 関連製品で提供するインタフェースは、次のとおりです。

- DABroker for C++
C++クラスライブラリ
- DABroker for COBOL
COBOL インタフェース
- DABroker for Java

JDBC インタフェース (Java)

- DABroker for ODBC
ODBC インタフェース
- DABroker for .NET Framework
.NET Framework データプロバイダ
- DABroker for ORB
CORBA インタフェース

1.2.2 DBPARTNER, HITSENSER5, DBPARTNER Library, XDM/RD 外部表からのアクセスをサポート

DABroker 関連製品で提供する API を使ってアプリケーションを開発しなくても、GUI を提供する DBPARTNER, HITSENSER5, DBPARTNER Library, 又は XDM/RD 外部表からデータベースにアクセスできます。

1.2.3 アクセスできるデータベースの種類が豊富

DABroker は、次の DBMS 及びリモートデータベースアクセス用プログラムに対応するインタフェースを提供します。

- ORACLE
Windows の場合、Oracle9i, Oracle10g, Oracle11g, Oracle12c, 及び Oracle19c にアクセスできます。
HP-UX の場合、Oracle8, Oracle8i, Oracle9i, Oracle10g, 及び Oracle11g にアクセスできます。
Solaris の場合、Oracle8, Oracle8i, Oracle9i, 及び Oracle10g にアクセスできます。
AIX の場合、Oracle8i, Oracle9i, Oracle10g, 及び Oracle11g にアクセスできます。
Linux の場合は、Oracle9i, Oracle10g, Oracle11g, Oracle12c, 及び Oracle19c にアクセスできます。
- HiRDB
- SQL Anywhere
Windows の場合だけアクセスできます。
ODBC2.0 でアクセスします。
- Adaptive Server Anywhere
Windows の場合だけアクセスできます。
ODBC3.0 でアクセスします。
- SQL Server

Windows の場合だけアクセスできます。

ODBC3.0 でアクセスします。

- RDA Link for Gateway
(メインフレーム系データベースへのアクセスを実現するプログラム)
AIX 及び Linux の場合は利用できません。
- DIAPRISM
Windows の場合だけアクセスできます。
ODBC3.0 でアクセスします。
DIAPRISM へアクセスする場合の環境設定方法については「[付録 A DIAPRISM アクセス時の環境設定方法 \(Windows\)](#)」を参照してください。
- Database Connection Server
(メインフレーム系データベースへのアクセスを実現するプログラム)

DABroker は、リレーショナルデータベースだけではなく、オブジェクトリレーショナルデータベースやメインフレーム環境のデータベースへのアクセスをサポートします。

アプリケーション開発で使用するインタフェースによって、アクセスできるデータベースが異なります。アクセスできるデータベースについては、次の各マニュアルを参照してください。

- 「DABroker for Java™ Version 2 DABroker Driver for Java™ Technology」
- 「DABroker for COBOL」
- 「DABroker for C++」
- 「DABroker for ORB」
- 「DABroker for ODBC」
- 「DABroker for .Net Framework」

各プログラムによってアクセスできるデータベースには次の制限があります。

- DBPARTNER から SQL Anywhere にはアクセスできません。
- HITSENSER5 以外から DIAPRISM にはアクセスできません。

HITSENSER5 からアクセスできるデータベースについては、HITSENSER5 のマニュアルを参照してください。

(1) リレーショナルデータベース (RDBMS) 及びオブジェクトリレーショナルデータベース (ORDBMS) のサポート

DABroker では、RDBMS (Relational DBMS) の、ORACLE, HiRDB, SQL Anywhere, Adaptive Server Anywhere, 及び SQL Server へのアクセスをサポートします。

(2) メインフレーム系データベースのサポート

DABroker は、メインフレーム系データベースへのアクセスもサポートします。メインフレーム系データベースへのアクセスには、リモートデータベースへのアクセスを実現する RDA Link for Gateway, 又は Database Connection Server を使用します。

(a) RDA Link for Gateway を使ったデータベースアクセス

RDA Link for Gateway でアクセスできるデータベースは、次のとおりです。

- VOS3 XDM/RD
- VOSK SQL/K^{*1}
- VOS1 RDB1 E2^{*2}

注※1

このマニュアルでは、VOSK RDB 編成ファイルを VOSK SQL/K と表記しています。VOSK RDB 編成ファイルを使用する場合は、VOSK SQL/K が前提となります。

注※2

Windows, 及び Solaris の場合、VOS1 RDB1 E2 を使用できます (HP-UX の場合は VOS1 RDB1 E2 を使用できません)。

(b) Database Connection Server を使ったデータベースアクセス

Database Connection Server を使ってアクセスできるデータベースは、次のとおりです。

- VOS3 XDM/RD
- VOS3 XDM/SD
- VOSK SQL/K
- VOS1 PDMII E2
- VOS1 RDB1 E2
- VOSK スプールファイル

使用するアプリケーション開発用インタフェースによって、アクセスできるデータベースが異なります。詳細については、各 DABroker 関連製品、DBPARTNER のマニュアルを参照してください。

1.2.4 分散トランザクション処理

TPBroker の OTS(Object Transaction Services)機能, Java JTS 機能を提供するトランザクションマネージャ(Java TM)又は OpenTP1 for .NET Framework を利用することで、DABroker からトランザクションを制御することなく、複数異種のデータベースに対して、X/Open DTP モデル XA インタフェースを使用した分散トランザクションを実現できます。

(1) OTS を使った分散トランザクション処理

(a) OTS を使った分散トランザクション処理の概要

接続先が次の DBMS の場合、TPBroker の OTS 機能を利用することで分散トランザクションを実現できます。

- HiRDB
- ORACLE

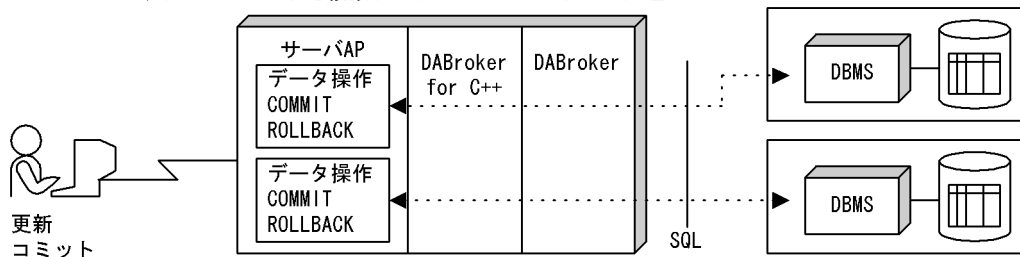
また、TPBroker にトランザクションの制御を任せることで、DABroker のデータベースアクセス処理と、TPBroker と連携する他製品のリソースアクセス処理とを、一つのグローバルトランザクションとして同期制御できるようになります。

複数異種の分散データベースへのアクセスの例を図 1-4 に示します。

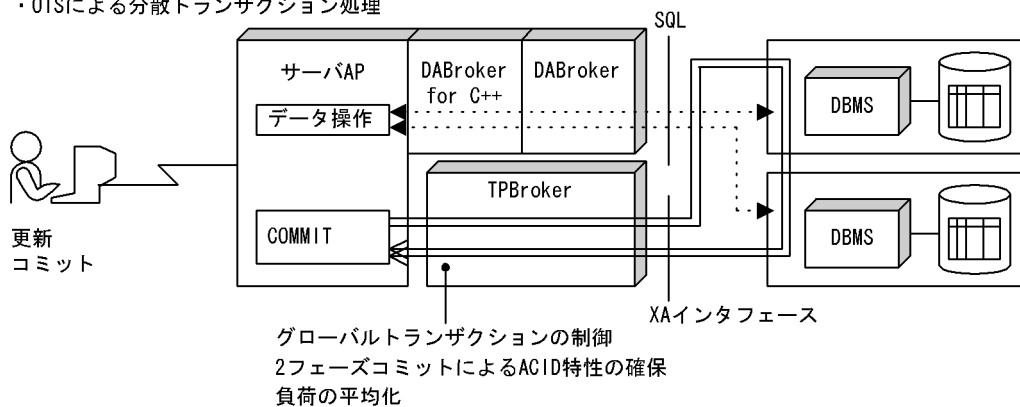
図 1-4 複数異種の分散データベースへのアクセスの例 (OTS を使った分散トランザクション処理)

複数異種の分散データベースへのアクセスの例

- ・サーバアプリケーションによる複数データベースへのアクセス処理



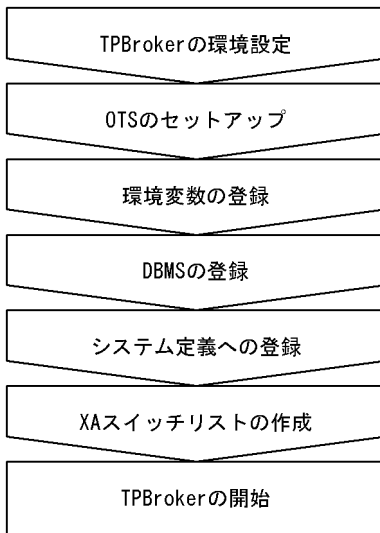
- ・OTSによる分散トランザクション処理



(b) OTS を利用するまでの手順

OTS を使って分散トランザクション処理を実現するには、次の手順で準備しておく必要があります。各手順の詳細については、マニュアル「トランザクショナル分散オブジェクト基盤 TPBroker ユーザーズガイド」を参照してください。OTS を利用するまでの手順を、図 1-5 に示します。

図 1-5 OTS を利用するまでの手順



- TPBroker の環境設定
TPBroker のインストールディレクトリのパスなどを設定します。
- OTS のセットアップ
TPBroker と実行環境をセットアップします。
次のコマンドを使用します。
 - tssetup コマンド
 - admsetup コマンドこれによって、OTS の実行環境と adm デーモンの実行環境が作成されます。
- 環境変数の登録
使用する DBMS ごとに、DBMS のホームディレクトリのパスなどの環境変数を登録します。
環境変数の登録方法については、各 DBMS のマニュアルを参照してください。
- DBMS の登録
TPBroker に DBMS を登録します。
次のコマンドを使用します。
 - tslnkrm コマンド
- システム定義への登録
OTS のシステム定義に DBMS を登録します。
次のコマンドを使用します。
 - tskeycreate コマンド
 - tsdefvalue コマンド
- XA スイッチリストの作成

TPBroker では OTS 機能を実現するために、DBMS の XA インタフェースを利用します。XA インタフェースを利用する場合、使用するデータベース用にリソースマネージャの XA スイッチリストを作成します。

次のコマンドを使用します。

- tsmkobj コマンド
- TPBroker の開始
上記の準備が完了したら、TPBroker を開始します。
次のコマンドを使用します。
 - admstart コマンド

(2) Java JTS を使った分散トランザクション処理

(a) Java JTS を使った分散トランザクション処理の概要

接続先が次の DBMS の場合、Java JTS 機能を提供するトランザクションマネージャ (Java TM) を利用することで分散トランザクションを実現できます。

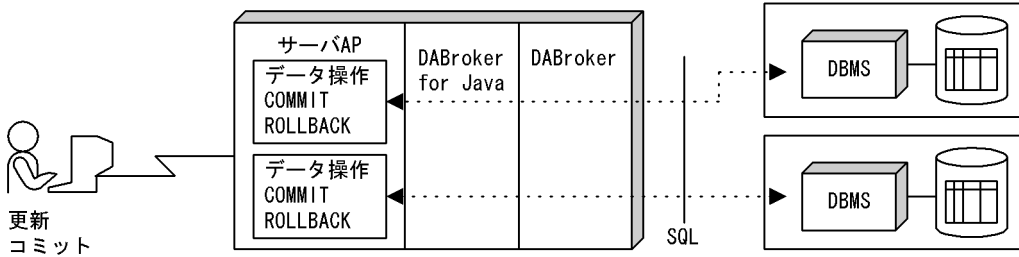
- ORACLE
- HiRDB

また、Java TM にトランザクションの制御を任せることで、DABroker のデータベースアクセス処理と、Java TM と連携する他製品のリソースアクセス処理とを、一つのグローバルトランザクションとして同期制御できるようになります。複数異種の分散データベースへのアクセスの例を図 1-6 に示します。

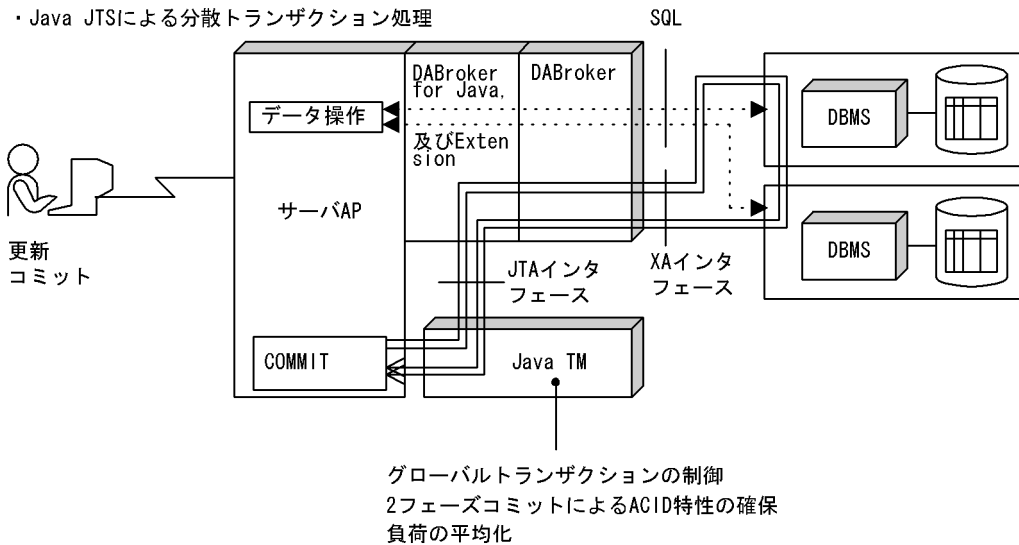
図 1-6 複数異種の分散データベースへのアクセスの例 (Java JTS を使った分散トランザクション処理)

複数異種の分散データベースへのアクセスの例

- ・サーバアプリケーションによる複数データベースへのアクセス処理



- ・Java JTSによる分散トランザクション処理



(b) Java JTS を利用するまでの手順

Java JTS を使って分散トランザクション処理を実現するには、次の手順で準備しておく必要があります。各手順の詳細については、Java TM が提供するドキュメントを参照してください。

- ・ Java TM の環境設定
- ・ DBMS の XA 接続環境設定
- ・ DABroker for Java Version 2 と Extension の JTA 環境設定
- ・ JNDI 環境設定
- ・ DABroker の環境設定
- ・ Java TM の開始

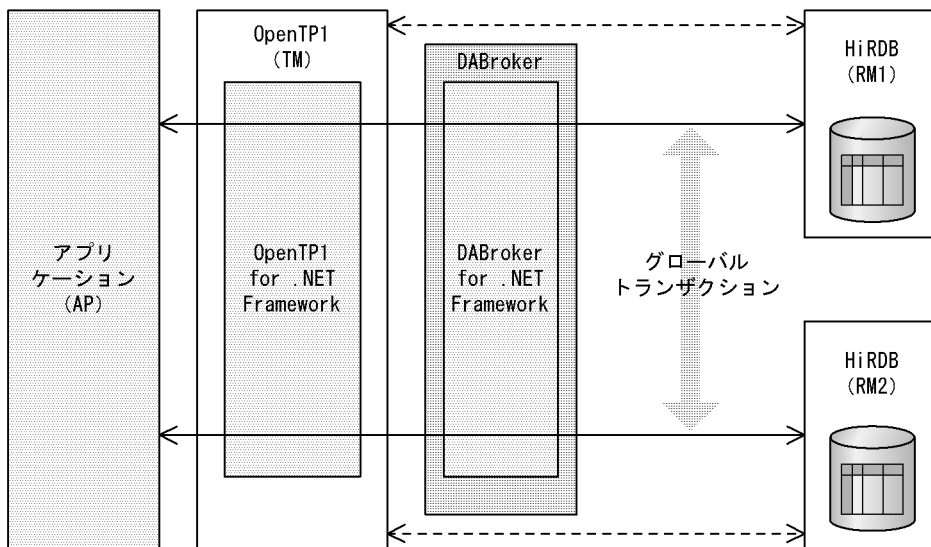
(3) OpenTP1 for .NET Framework を使った分散トランザクション処理

DABroker for .NET Framework は、OpenTP1 for .NET Framework とトランザクション連携をすることができます。OpenTP1 for .NET Framework と連携し、XA インタフェースのグローバルトランザクションに参加することで、OpenTP1 システムに登録された複数の DBMS 間や、DBMS とほかのリソー

スマネージャとの間で、同期を取ったトランザクション制御（グローバルトランザクションの制御）ができます。

OpenTP1 for .NET Framework とのトランザクション連携の概要を図 1-7 に示します。

図 1-7 OpenTP1 for .NET Framework とのトランザクション連携の概要



(凡例)

- ←→ : アプリケーションからのデータベースアクセスを示します。
- ←-→ : XAインタフェースによるトランザクション制御を示します。

OpenTP1 for .NET Framework と連携して、XA インタフェースのグローバルトランザクションに参加する場合、DBMS へのアクセス部分は DABroker for .NET Framework を使用し、トランザクション制御の部分は、OpenTP1 for .NET Framework を使用します。

アプリケーションは OpenTP1 に対してトランザクションの開始や決着を要求します。このとき、DABroker for .NET Framework の DABTransaction クラスは使用しないで、OpenTP1 for .NET Framework が提供するクラスを使用します。グローバルトランザクションの制御対象となっているデータベースに対して検索処理や更新処理をする場合は、DABroker for .NET Framework のクラスを使用します。

1.3 アプリケーション開発用インタフェースの使用

DABroker 関連製品では、データベースにアクセスするアプリケーションを開発するための、インタフェースを提供します。ここでは、DABroker 関連製品で提供する、アプリケーション開発用インタフェースの特長、及び使用形態について説明します。

1.3.1 DABroker for C++で開発するアプリケーション

DABroker for C++では、データの検索、更新、任意の SQL の実行ができる C++クラスライブラリを提供します。DABroker for C++のクラスライブラリを使用して、データベースにアクセスするためのサーバアプリケーションを開発できます。

また、DABroker for C++は、DABroker for ORB で提供する CORBA 対応インタフェースの動作時の前提製品にもなります。

(1) DABroker for C++を使って開発するときの特長

次に、DABroker for C++の特長を示します。

(a) コーディング量を少なくできる高性能なクラスライブラリ

DABroker for C++は、データベースへの接続、SQL の実行、ストアドプロシジャ、トランザクションなどの制御を設定するプロパティ、メソッドを提供します。DABroker for C++のクラスライブラリを使用すると、「埋め込み SQL」などの手続き方法より少ないコーディング量でデータベースアクセス用のサーバアプリケーションを開発できます。

(b) マルチスレッド対応のサーバアプリケーション開発

DABroker では、アプリケーションからのデータベースアクセス要求をマルチスレッドで実行します。DABroker for C++で開発するアプリケーションからの要求も、マルチスレッドで実行します。DABroker for C++で提供するクラスライブラリを使用すると、DABroker でのスレッド制御方法などを意識することなくマルチスレッドで実行するアプリケーションを開発できます。

(c) データベースアクセス処理を同期又は非同期で実行

DABroker for C++で開発するサーバアプリケーションでは、データベースアクセスに同期、又は非同期処理を選択できます。

同期アクセス処理でデータベースアクセスを実行すると、データベースでの処理結果がアプリケーションに通知されるまで、アプリケーションでは、次の処理を実行できません。

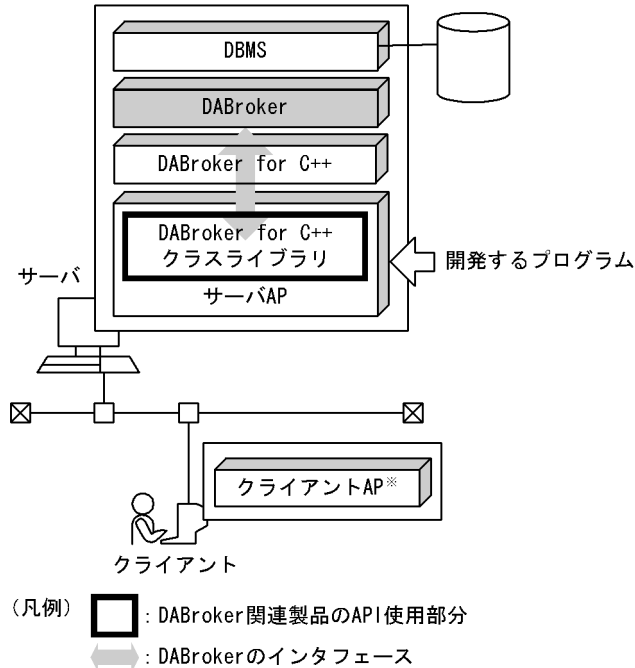
非同期アクセス処理でデータベースアクセスを実行すると、アプリケーションでは、データベースからの処理結果の通知を待たずに次の処理を実行できます。

同期又は非同期アクセス処理については、「[2.2 データベースアクセス](#)」を参照してください。

(2) DABroker for C++の使用形態

DABroker for C++で開発するアプリケーションは、サーバマシンに組み込んで使用します。開発したサーバアプリケーションに対応するクライアントアプリケーションは、必要に応じて開発します。図 1-8 に、システム構成例を示します。

図 1-8 C++で開発した場合のシステム構成例



注※ サーバAPIに対応するために任意で用意するプログラムです。

1.3.2 DABroker for COBOL で開発するアプリケーション

DABroker for COBOL では、データの検索、更新、任意の SQL の実行ができる COBOL インタフェースを提供します。DABroker for COBOL のインタフェースを使用して、データベースにアクセスするためのサーバアプリケーションを開発できます。なお、COBOL インタフェースは、COBOL85 を利用します。

(1) DABroker for COBOL を使って開発するときの特長

次に DABroker for COBOL の特長を示します。

(a) データベースアクセスを実現するインタフェース

DABroker for COBOL は、データベースへの接続、SQL の実行、ストアドプロシジャ、トランザクションなどの制御を設定するインタフェースを提供します。

(b) マルチスレッド対応のサーバアプリケーションの開発

DABroker では、アプリケーションからのデータベースアクセス要求をマルチスレッドで実行します。DABroker for COBOL で開発するアプリケーションからの要求も、マルチスレッドで実行します。

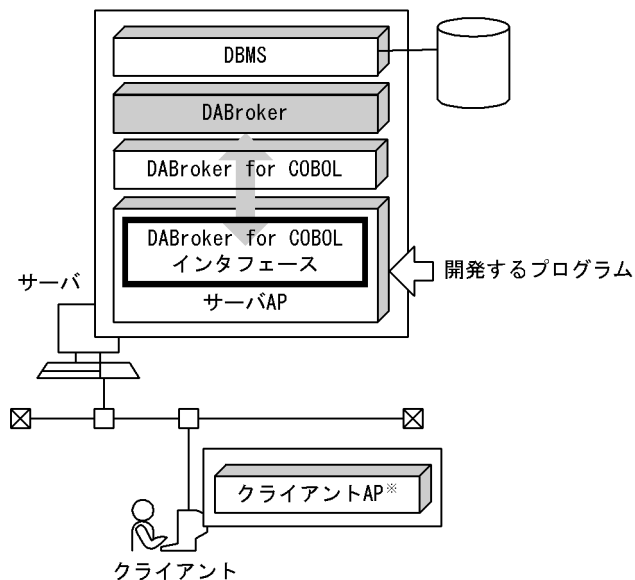
(c) データベースアクセス処理を同期で実行

DABroker for COBOL で開発するサーバアプリケーションは、データベースアクセスを同期処理で実行します。同期アクセス処理でデータベースアクセスを実行すると、データベースでの処理結果がアプリケーションに通知されるまで、アプリケーションでは、次の処理を実行できません。同期アクセス処理については、「2.2 データベースアクセス」を参照してください。

(2) DABroker for COBOL の使用形態

DABroker for COBOL で開発するアプリケーションは、サーバマシンに組み込んで使用します。開発したアプリケーションに対応するクライアントアプリケーションは、必要に応じて開発します。図 1-9 に、システム構成例を示します。

図 1-9 COBOL で開発した場合のシステム構成例



(凡例) □ : DABroker 関連製品のAPI使用部分

⇄ : DABroker のインタフェース

注※ : サーバAPIに対応するために任意で用意するプログラムです。

1.3.3 DABroker for Java で開発するアプリケーション

DABroker for Java では、JDBC インタフェースに対応する JDBC ドライバを提供します。ここで開発するアプリケーションは、JDBC インタフェースの仕様に従って開発します。JDBC とは、Java の標準的なデータベースアクセス用のインタフェースで、JDK1.2 以降 (java.sql パッケージ) で提供されます。

DABroker for Java で開発するアプリケーションは、JDBC で実現できる検索、更新などの処理を要求できます。

DABroker for Java では、OS に依存しない実行環境を提供するなどの Java の特長を活かしたアプリケーションを開発できます。また、Applet を開発した場合、Applet がダウンロードされるクライアントごとに Applet をメンテナンスする必要がないため、Applet 開発者の負担を軽減できます。

Extension^{*}は、JDBC2.0 の拡張機能を提供します。JNDI (Java Naming and Directory Interface)、JTS (Java Transaction Service) や JTA (Java Transaction API) との連携機能を提供します。

注※ DABroker for Java Version 2 の機能です。

(1) DABroker for Java を使って開発するときの特長

次に DABroker for Java の特長を示します。

(a) クライアントアプリケーション及びサーバアプリケーションの開発

DABroker for Java で開発できるアプリケーションには、次の三種類があります。それぞれのアプリケーションは、JDBC インタフェースの仕様に従って開発します。

- Applet

WWW サーバからダウンロードして使用するプログラムです。マスタデータとなるプログラムがサーバにあれば、サーバからのダウンロードで常に最新の Applet を利用できます。ダウンロードするクライアントごとにプログラムをメンテナンスする必要がありません。

- クライアントアプリケーション

クライアントマシンに常駐するプログラムです。WWW ブラウザを使用しないで動作させます。

- サーバアプリケーション

サーバマシンに常駐するプログラムです。サーバアプリケーション、DABroker、JDBC ドライバを同一マシンに組み込んで使用するため、通信処理のオーバーヘッドがなく、クライアントアプリケーションでの動作に比べて処理性能が向上します。

(b) JDBC ドライバの提供

DABroker for Java では、JDBC ドライバを提供します。JDBC ドライバは、JDBC インタフェースに従って開発したアプリケーションからの要求を処理し、データベースへのアクセスを実現します。

(c) すぐに使用できる Applet を提供

データの検索を GUI から実行する Applet を提供します。提供 Applet は、「DB アクセスアプレット」と呼びます。DB アクセスアプレットは、SQL 文を直接入力する GUI を持っているため、SQL 文を入力するとデータの更新なども実行できます。DB アクセスアプレットを使用すると、アプリケーションを開発することなく、Java からデータベースにアクセスできます。

(d) データベースアクセスを同期処理で実行

DABroker for Java で開発するアプリケーションは、データベースアクセスを同期処理で実行します。同期アクセス処理でデータベースアクセスを実行すると、データベースでの処理結果がアプリケーションに通知されるまで、アプリケーションでは、次の処理を実行できません。同期アクセス処理については、「2.2 データベースアクセス」を参照してください。

(2) DABroker for Java の使用形態

DABroker for Java は、次の環境で動作するアプリケーションを開発できます。

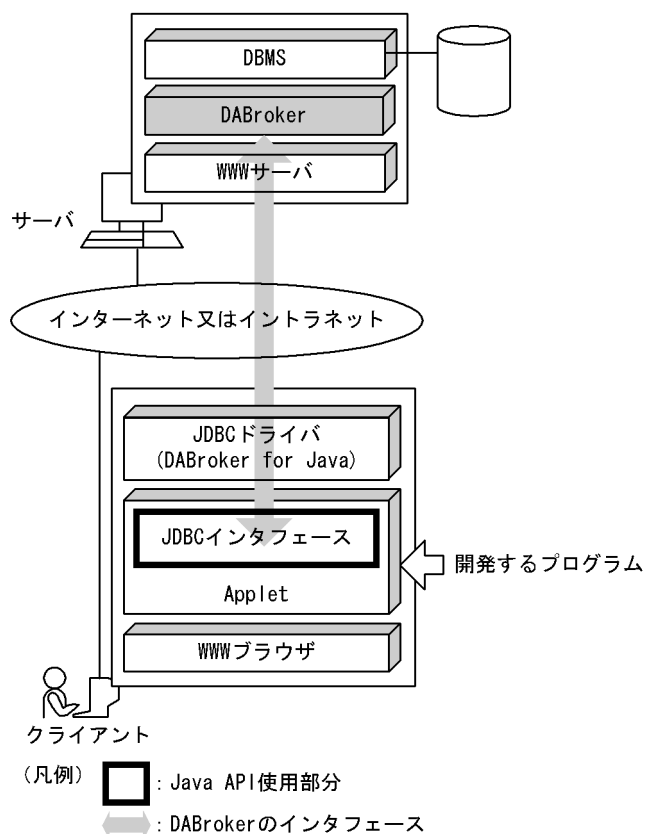
- WWW 環境
- クライアント/サーバ環境

次に、DABroker for Java のそれぞれの環境でのシステム構築例を示します。

(a) WWW 環境で使用する - Applet を使用する

Java Applet を作成し、WWW サーバから Java Applet をダウンロードして使用します。図 1-10 に、システム構成例を示します。

図 1-10 Applet を開発した場合のシステム構成例

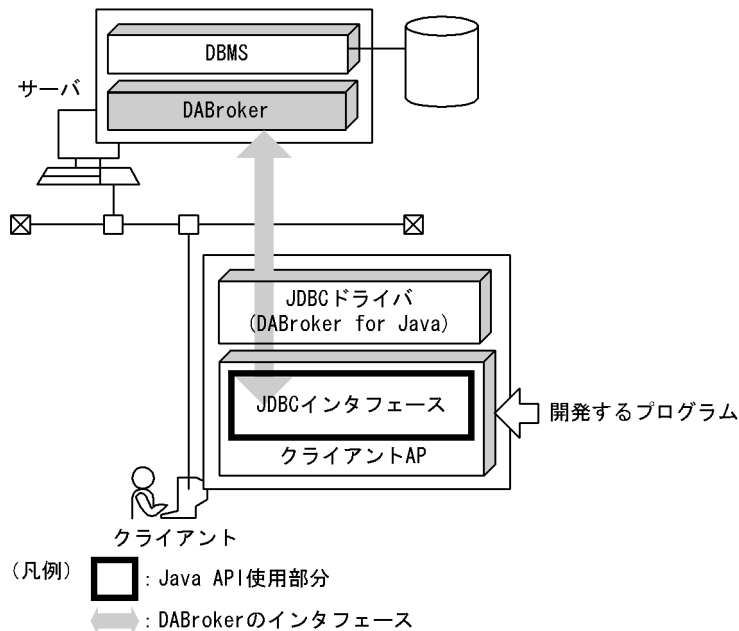


(b) クライアント環境で使用する－クライアントアプリケーションを使用する

クライアント／サーバ環境で使用するアプリケーションです。クライアントマシンに、開発したアプリケーション、及び DABroker for Java で提供する JDBC ドライバを組み込んで使用します。

図 1-11 に、システム構成例を示します。

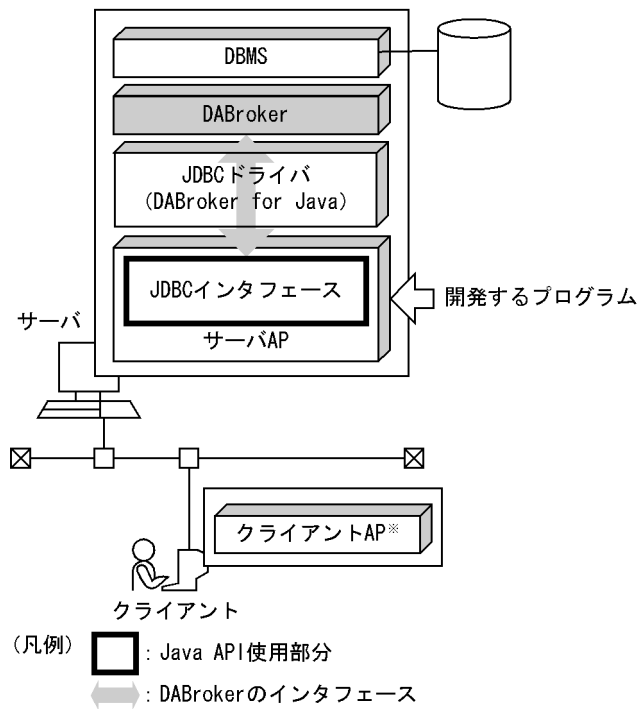
図 1-11 Java でクライアントアプリケーションを開発した場合のシステム構成例



(c) サーバ環境で使用する－サーバアプリケーションを使用する

クライアント／サーバ環境で使用するアプリケーションです。サーバマシンに、開発したアプリケーション及び DABroker for Java で提供する JDBC ドライバを組み込んで使用します。図 1-12 に、システム構成例を示します。

図 1-12 Java でサーバアプリケーションを開発した場合のシステム構成例



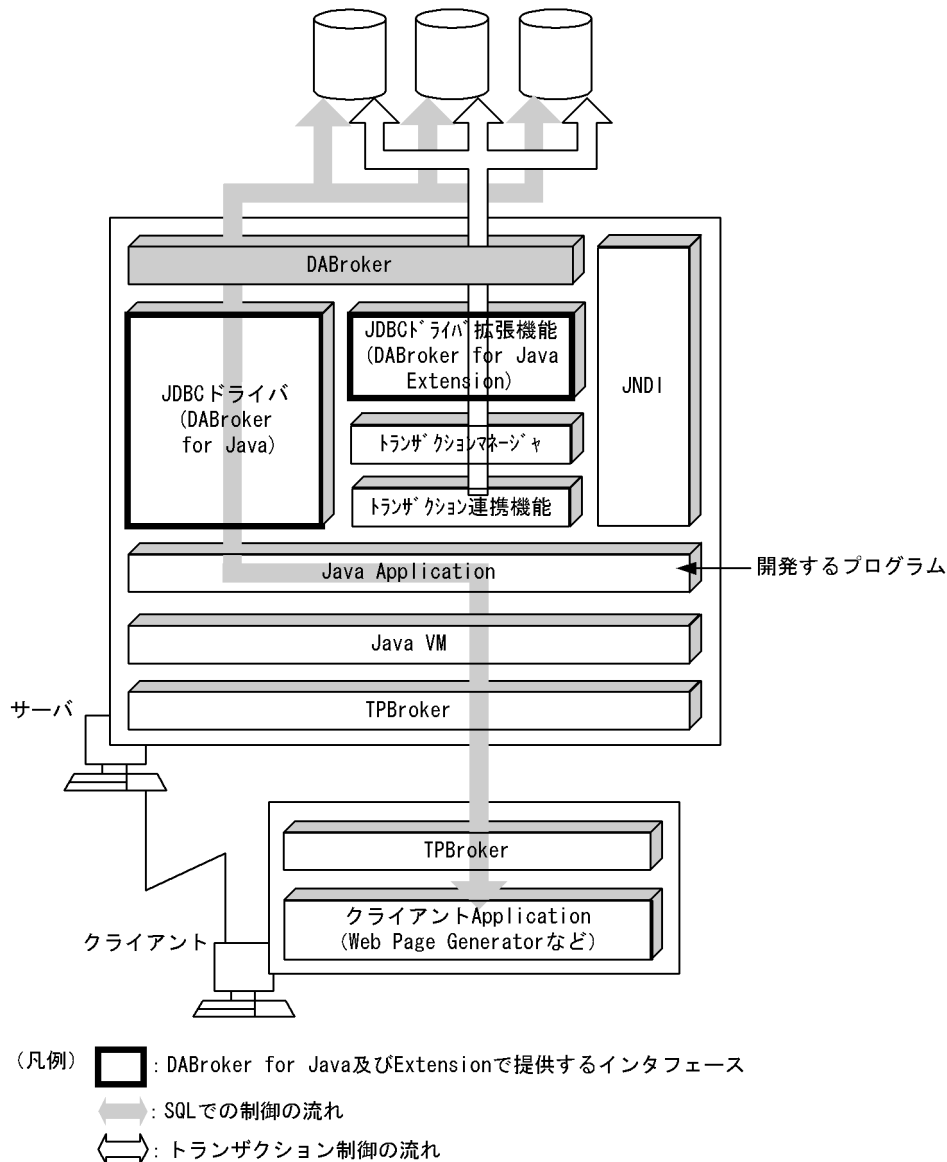
注※ サーバAPIに対応するために任意で用意するプログラムです。

(d) 分散トランザクション環境で使用するサーバアプリケーションを使用する

分散トランザクション環境で使用するアプリケーションです。分散トランザクション機能は、Java TM, Extension 及びトランザクション連携機能に関連するクラスのインタフェース定義を組み込んで使用します。また、使用する前提として、データベースとの接続情報をJNDIに登録しておきます。

Java Application の動作時には、JNDIに登録してあるデータベースとの接続情報を取り込みます。その情報を元にトランザクションマネージャと連携し、Java Application と DABroker を接続します。図 1-13 に、システム構成例を示します。

図 1-13 Java で分散トランザクション機能を利用したサーバアプリケーションを開発した場合のシステム構成例



1.3.4 DABroker for ORB で開発するアプリケーション

DABroker for ORB では、CORBA 対応のデータベースアクセスオブジェクト (Simple Interface) を提供します。

(1) DABroker for ORB の特長

Simple Interface とは、データベースにアクセスするための CORBA オブジェクトです。複雑なデータベースアクセスを 8 種類の関数に統合しています。Simple Interface を Web Page Generator を利用した WWW 環境で使用すると、アプリケーションを開発することなく、HTML 上にデータベース検索結果

を表示したり、HTML からデータベース更新を要求できます。Web Page Generator は、HTML ドキュメントを解釈して HTML で記述されたページを生成するプログラムです。

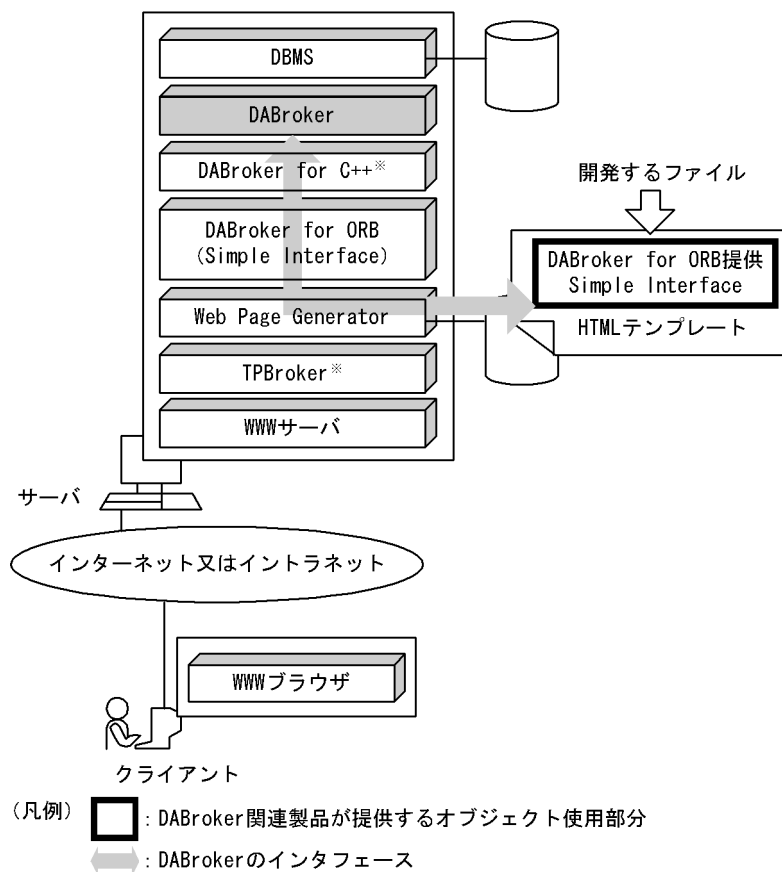
Simple Interface からのデータベースアクセスは、同期処理で実行します。同期アクセス処理でデータベースアクセスを実行すると、データベースでの処理結果がアプリケーションに通知されるまで、アプリケーションでは、次の処理を実行できません。同期アクセス処理については、「2.2 データベースアクセス」を参照してください。

(2) DABroker for ORB の使用形態

Simple Interface を、Web Page Generator の HTML テンプレートファイルに埋め込んで使用します。図 1-14 に、システム構成例を示します。

Web Page Generator を利用したシステムでは、WWW サーバ及び CORBA オブジェクト (Simple Interface) を呼び出すための TPBroker が必要です。

図 1-14 Simple Interface を使用した場合のシステム構成例



注※ DABroker for ORBの前提製品です。

1.3.5 DABroker for ODBC で開発するアプリケーション

DABroker for ODBC は、DABroker に対応する ODBC ドライバを提供します。

ODBC ドライバはデータベースアクセスのための API です。ODBC ドライバを使うと、アプリケーションから各種の DBMS に統一したインターフェースでアクセスできます。

DABroker for ODBC を使うと、ODBC インタフェースを利用したサーバアプリケーションを作成できます。また、既存の ODBC 対応のアプリケーションも、DABroker for ODBC を使ってデータベースにアクセスできます。

(1) DABroker for ODBC の特長

DABroker for ODBC の特長を次に示します。

ODBC 3.0 対応

DABroker for ODBC は ODBC 3.0 に対応しています。DABroker for ODBC を使えば、ODBC 3.0 で新しく追加された機能を含んだアプリケーションが作成できます。

マルチスレッド対応

DABroker for ODBC はサーバ上で動作することを前提としているため、マルチスレッドに対応しています。これによって、複数処理でのスループットの向上を実現しました。

(2) DABroker for ODBC の使用形態

DABroker for ODBC は、次の環境で動作するアプリケーションを開発できます。

- WWW 環境
- クライアント／サーバ環境

ODBC インタフェースを利用して開発したサーバアプリケーションは、DABroker for ODBC と同じマシンに組み込んで使用します。DABroker for ODBC と DABroker は、異なるマシンに組み込まれていても構いません。

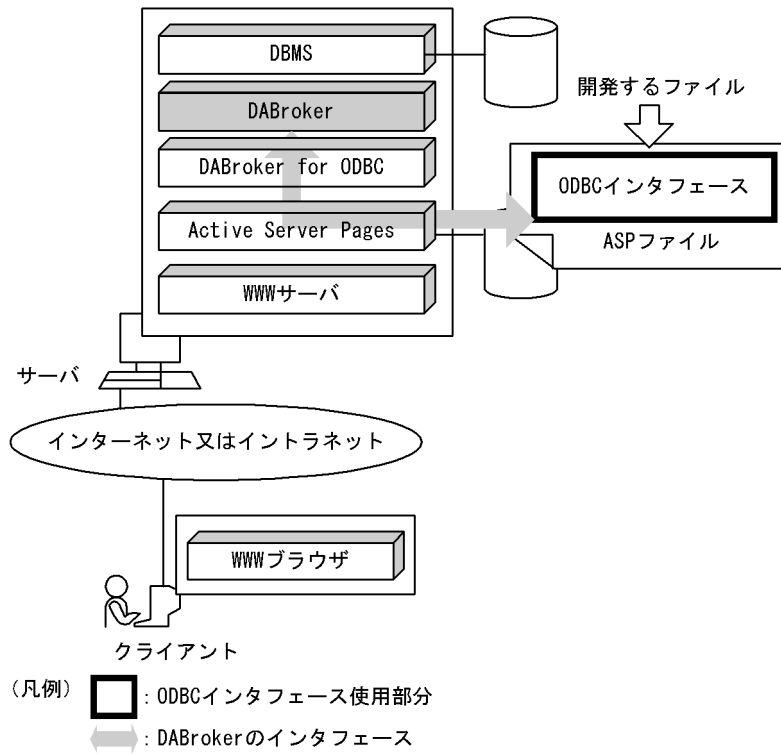
次に、DABroker for ODBC のそれぞれの環境でのシステム構築例を示します。

(a) WWW 環境で使用する－ASP ファイルを使用する

DABroker for ODBC を組み込んだ WWW サーバ上に、ODBC インタフェースを使用した ASP ファイルを作成し、WWW ブラウザから使用します。ASP ファイルとは、Microsoft Internet Information Server の機能である Active Server Pages で使用するファイルです。

図 1-15 にシステム構成例を示します。

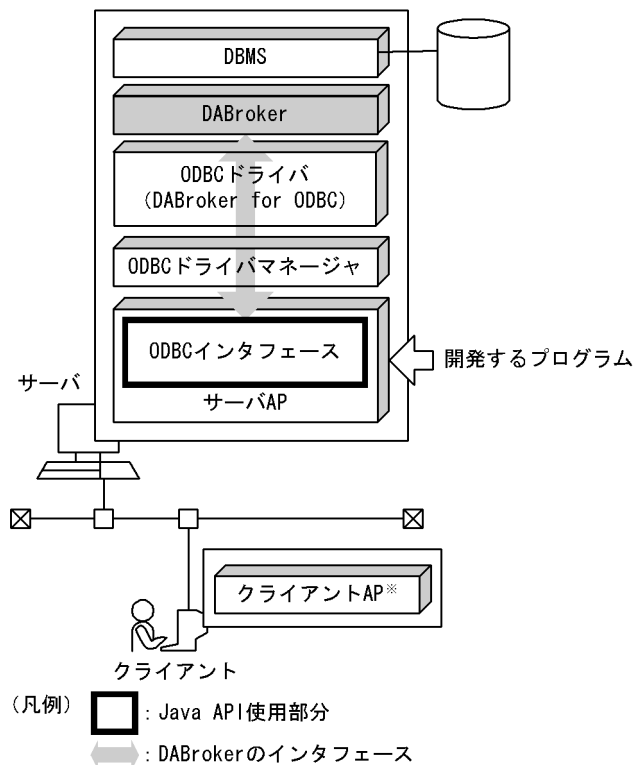
図 1-15 ODBC インタフェースで ASP ファイルを作成した場合のシステム構成例



(b) クライアント／サーバ環境で使用する－サーバアプリケーションを使用する

クライアント／サーバ環境で使用するアプリケーションです。サーバマシンに、開発したアプリケーションと DABroker for ODBC を組み込んで使用します。図 1-16 にシステム構成例を示します。

図 1-16 ODBC インタフェースでサーバアプリケーションを開発した場合のシステム構成例



注※ サーバAPIに対応するために任意で用意するプログラムです。

1.3.6 DABroker for .NET Framework で開発するアプリケーション

DABroker for .NET Framework は、Microsoft の .NET Framework のデータベースアクセス技術である ADO.NET に準拠した .NET Framework データプロバイダです。

DABroker for .NET Framework が提供する、.NET Framework に対応したインタフェースを使用することによって、.NET Framework のアプリケーションから DABroker を利用したデータベースアクセスができます。

DABroker for .NET Framework は、DABroker の持つ柔軟でシームレスなデータベースアクセス技術と、.NET Framework の持つ優れた開発環境を活用して、DABroker を利用したデータベースアクセスアプリケーションの開発を迅速にします。これによって、Windows プラットフォーム上で高いシステム生産性、運用性を持つデータベースアクセス基盤を構築できます。

(1) DABroker for .NET Framework の特徴

DABroker for .NET Framework の特徴を次に示します

.NET Framework アプリケーションからのデータベースアクセス

ASP.NET に代表される Microsoft のアプリケーション開発技術を利用して ASP.NET XML Web サービスや Web アプリケーションなどのアプリケーションから DABroker を経由してデータベースにアクセスできます。

OpenTP1 for .NET Framework との連携によるトランザクション制御

OpenTP1 システムに登録された複数の DBMS 間や、DBMS とほかのリソースマネージャとの間で、同期を取ったトランザクション制御ができます。

アプリケーション開発環境の拡充

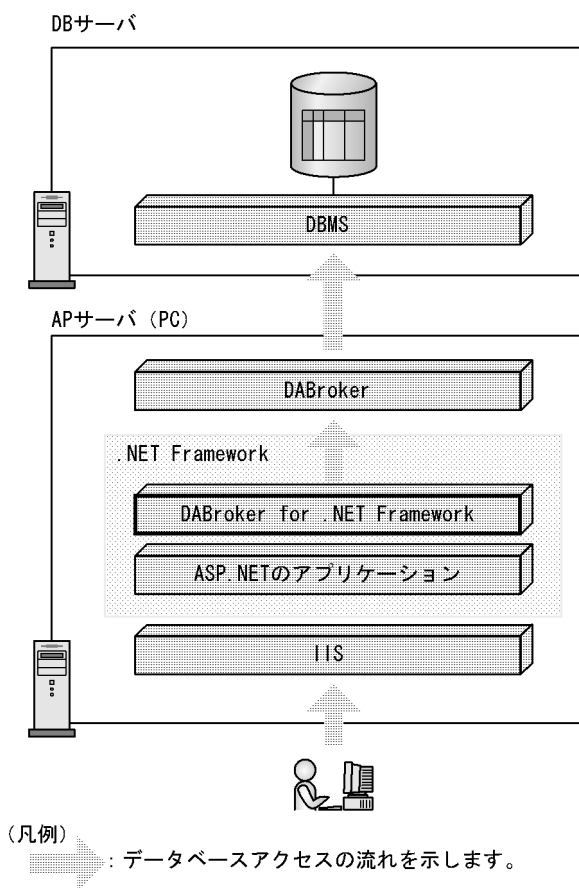
アプリケーションを開発する言語を C#、および Visual Basic .NET から選択できます。また、Visual Studio .NET といった .NET Framework が持つ優れた開発環境やクラスライブラリを活用することによって、アプリケーションの開発を迅速に行うことができます。

(2) DABroker for .NET Framework の使用形態

(a) ASP.NET アプリケーションからの利用

ASP.NET のアプリケーションからデータベースにアクセス場合の構築例を図 1-17 に示します。

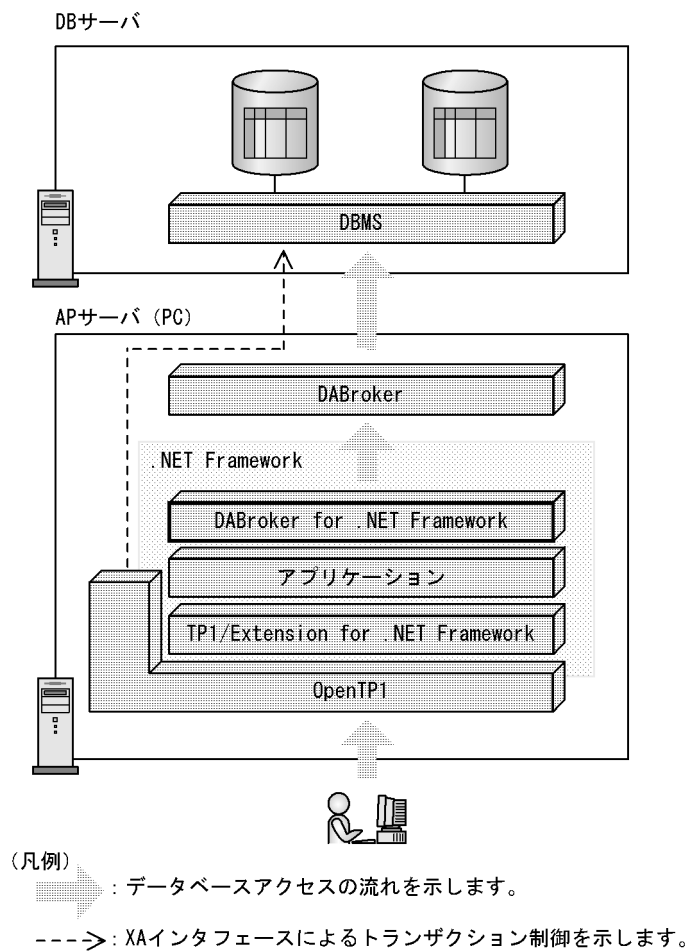
図 1-17 ASP.NET のアプリケーションから DABroker を経由してデータベースにアクセスする場合のシステム構成例



(b) OpenTP1 for .NET Framework と連携する

OpenTP1 for .NET Framework と連携しトランザクション制御を行う場合の構築例を図 1-18 に示します。

図 1-18 OpenTP1 for .NET Framework と連携する場合のシステム構成例



1.4 DBPARTNER, HITSENSER5 などからのアクセス

DABroker 関連製品で提供する API を使ってアプリケーションを開発しなくても、GUI を提供する DBPARTNER, 又は HITSENSER5 からデータベースにアクセスできます。

DBPARTNER には、次の製品があります。

- **DBPARTNER/Client**
クライアント/サーバ環境の形態で使用できます。
- **DBPARTNER/Web**
WWW 環境の形態で使用できます。
- **DBPARTNER Mobile (DBPARTNER Mobile - Server, 及び DBPARTNER Mobile - Client)**
モバイル環境の形態で使用できます。クライアント/サーバ環境でも使用できます。

HITSENSER5 には、次の製品があります。

- **HITSENSER5 Professional**
クライアント/サーバ環境の形態で使用できます。
- **HITSENSER5 Standard**
クライアント/サーバ環境の形態で使用できます。
- **HITSENSER5 for Cosmicube**
クライアント/サーバ環境の形態で使用できます。
- **HITSENSER5 for RDB**
クライアント/サーバ環境の形態で使用できます。
- **HITSENSER5 Web**
WWW 環境の形態で使用できます。

このほか、次の製品を使用してデータベースにアクセスできます。これらの方法ではデータベースを操作するアプリケーションプログラムを作成します。

- **DBPARTNER Library**
クライアント/サーバ環境の形態で使用できます。
- **XDM/RD E2 の外部表**
クライアント/サーバ環境の形態で使用できます。

DBPARTNER から SQL Anywhere にはアクセスできません。

エージェント機能を使う DBPARTNER (DBPARTNER2 Client, DBPARTNER/Client32, DBPARTNER2 Web, 又は DBPARTNER Mobile) からは、Database Connection Server 経由でメインフレーム系データベースにアクセスできません。

HITSENER5 からアクセスできるデータベースについては、HITSENER5 のマニュアルを参照してください。

XDM/RD E2 の外部表からアクセスできるデータベースについては、XDM/RD E2 のマニュアルを参照してください。

1.4.1 DBPARTNER/Client を利用する場合

データベースアクセスを DBPARTNER/Client で提供する GUI から要求できます。DBPARTNER/Client は、各種データベースに直接アクセスしてデータ処理を要求します。実行できるデータベースの操作は、次のとおりです。

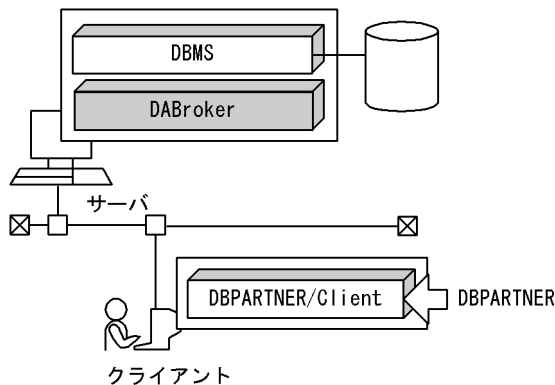
- データの検索（抽出）
- データの更新（追加，変更，削除）
- 表定義

DBPARTNER/Client を使用すると、DBPARTNER/Client のマシン上にある表計算ソフトと連携し、各種アプリケーションと連動できるため、データベースで検索したデータの加工・編集ができます。

DBPARTNER/Client からのデータベースアクセスは、同期アクセス処理、予約処理、又はエージェント機能を使用して実行できます。データベースアクセスの方法については、「2.2 データベースアクセス」を参照してください。

図 1-19 にシステム構成例を示します。

図 1-19 DBPARTNER/Client を使用する場合のシステム構成例



1.4.2 DBPARTNER/Web を利用する場合

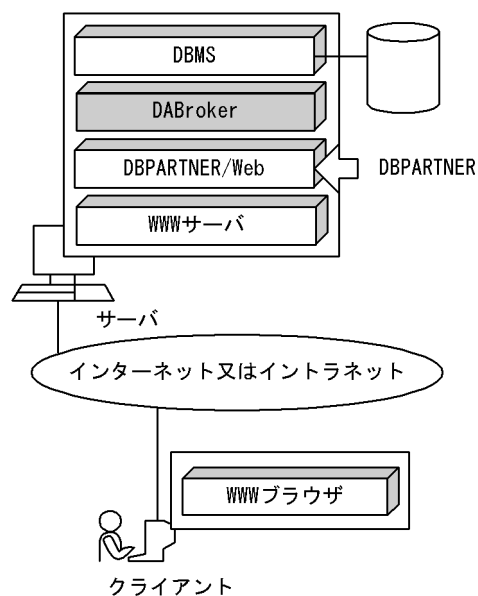
DBPARTNER/Web は、WWW サーバに組み込んで使用します。DBPARTNER/Web を利用する場合は、DBPARTNER/Web が提供する HTML インタフェースを表示した WWW ブラウザからデータベースアクセスを要求します。実行できるデータベースの操作処理を次に示します。

- データの検索（抽出）
- データの更新（追加，変更，削除）

DBPARTNER/Web からのデータベースアクセスは，同期アクセス処理，予約処理で実行できます。また，DBPARTNER2 Web の場合はエージェント機能を使用した実行もできます。データベースアクセスの方法については，「2.2 データベースアクセス」を参照してください。

図 1-20 にシステム構成例を示します。

図 1-20 DBPARTNER/Web を利用する場合のシステム構成例



1.4.3 DBPARTNER Mobile を利用する場合

DBPARTNER Mobile は，データベースアクセスを外出先・移動中に実現するために開発された製品です。外出先・移動中でのアクセス要求には，公衆回線網を利用しますが，通信状態，通信速度，通信コストなどに問題があります。しかし，DBPARTNER Mobile では，Mobile Message Queue と連携して確実な通信を実現し，データの暗号化を処理する Secure Socket と連携して情報の漏洩を防止できます。DBPARTNER Mobile には，DBPARTNER Mobile - Client 及び DBPARTNER Mobile - Server があります。DBPARTNER Mobile - Client は，データベースアクセスを要求し，DBPARTNER Mobile - Server は，要求のメッセージを解析します。DBPARTNER Mobile で実行できるデータベースの操作は次のとおりです。

- データの検索（抽出）
- データの更新（追加）

DBPARTNER Mobile からのデータベースアクセスは，エージェント機能を使用して実行します。エージェントとは，アクセスの実行時間，アクセス条件などのデータベースの定型処理内容をあらかじめ DABroker に登録したものです。エージェントを使用すると，クライアントでデータベースの処理内容を

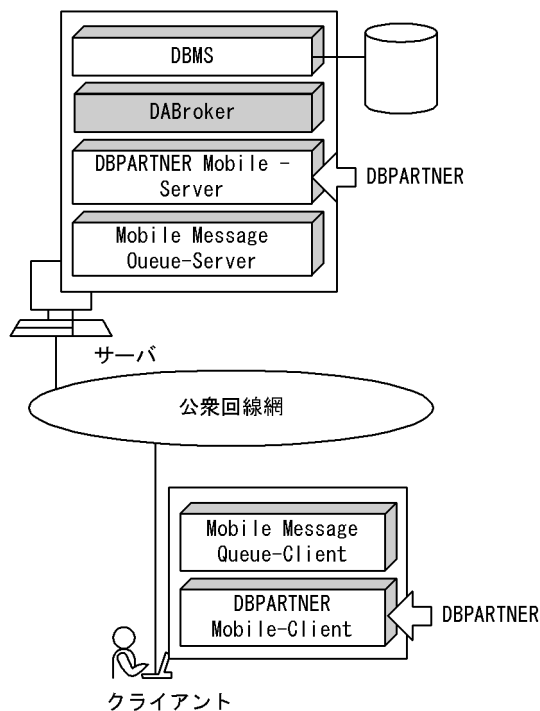
設定することなく、エージェントを実行するだけでエージェントに登録された定型処理を実現します。データベースアクセスの方法については、「2.2 データベースアクセス」を参照してください。

なお、DBPARTNER Mobile は、モバイル環境だけではなく、クライアント／サーバ環境でも使用できます。

(1) モバイル環境で使用する場合

この場合、公衆回線網を使用して確実な通信を実現するために、Mobile Message Queue を使用します。図 1-21 にシステム構成例を示します。

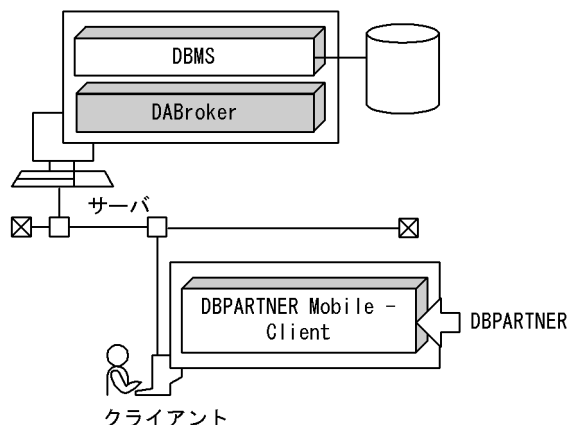
図 1-21 DBPARTNER Mobile を利用する場合のシステム構成例



(2) クライアント／サーバ環境で使用する場合

DBPARTNER Mobile は、クライアント／サーバ環境でも使用できます。この場合は、DBPARTNER Mobile - Client だけを使用します。モバイル環境で使用する場合と同様、データベースへのアクセスをエージェントで実行できます。図 1-22 にシステム構成例を示します。

図 1-22 クライアント／サーバ環境で DBPARTNER Mobile - Client を利用する場合のシステム構成例



1.4.4 HITSENSER5 を利用する場合

データベースアクセスを、HITSENSER5 で提供する GUI から要求できます。HITSENSER5 を使うと、多数の条件を組み合わせてデータベース中のデータを絞り込み、結果を表やグラフに表示できます。

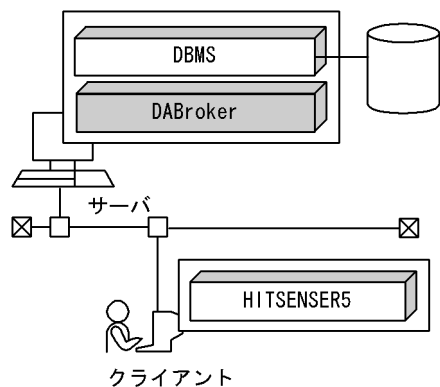
HITSENSER5 では、次の場合にデータベースを参照します。

- 分析データベースを作成する場合
- 分析対象データを作成する場合
- 分析データを取得する場合
- 管理情報をリレーショナルデータベースに登録する場合

HITSENSER5 からのデータベースアクセスは、同期アクセス処理になります。同期アクセス処理については、「[2.2.1 同期アクセス処理](#)」を参照してください。

図 1-23 にシステム構成例を示します。

図 1-23 HITSENSER5 を利用する場合のシステム構成例



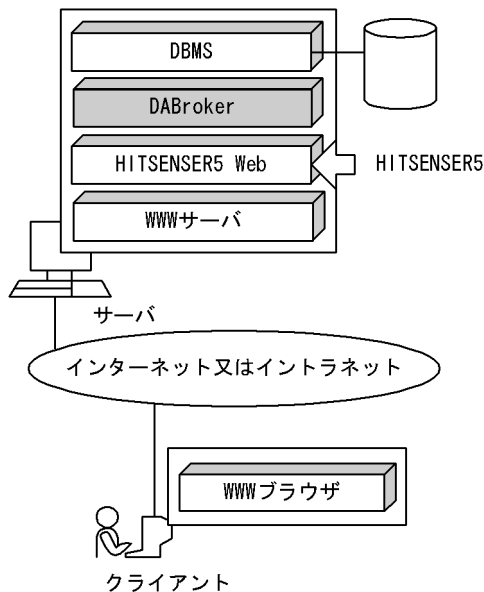
1.4.5 HITSENER5 Web を利用する場合

HITSENER5 Web は、WWW サーバに組み込んで使用します。HITSENER5 Web を使うと、HITSENER5 Web が提供する HTML インタフェースを表示した WWW ブラウザから HITSENER5 の機能を利用できます。データベース中のデータの参照なども、WWW ブラウザから要求します。

HITSENER5 Web からのデータベースアクセスは、同期アクセス処理になります。同期アクセス処理については、「2.2.1 同期アクセス処理」を参照してください。

図 1-24 にシステム構成例を示します。

図 1-24 HITSENER5 Web を利用する場合のシステム構成例



1.4.6 DBPARTNER Library を利用する場合

データベースアクセスを、DBPARTNER Library が提供する API を使用して作成したアプリケーションプログラムから要求できます。

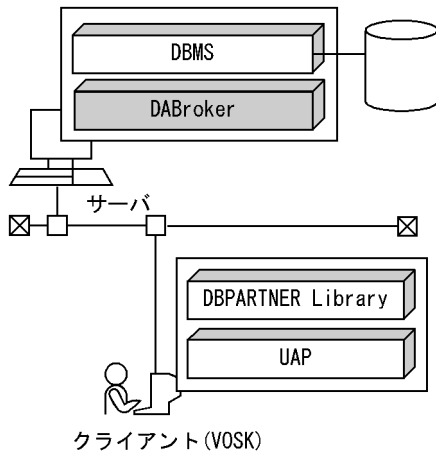
DBPARTNER Library を使うと、VOSK システムから DABroker を経由して、ワークステーションやパーソナルコンピュータ上のデータベースの検索、加工、及び編集ができます。

実行できるデータベース操作は、次のとおりです。

- データの検索
- データの更新（追加・変更・削除）

図 1-25 にシステム構成例を示します。

図 1-25 DBPARTNER Library を利用する場合のシステム構成例



1.4.7 XDM/RD E2 の外部表を利用する場合

データベースアクセスを、XDM/RD E2 の外部表を操作するように作成したアプリケーションプログラムから要求できます。

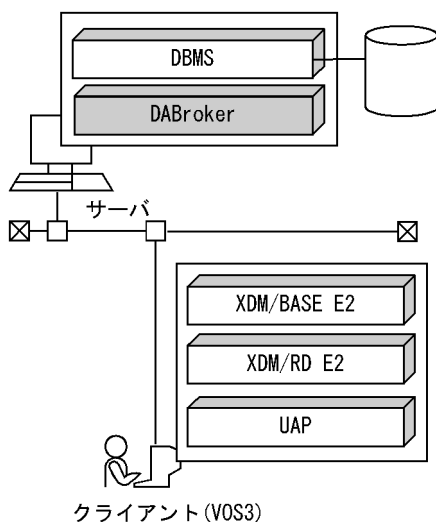
XDM/RD E2 の外部表を使うと、VOS3 上の XDM/RD E2 の外部表から DABroker を経由して、ワークステーションやパーソナルコンピュータ上のデータベースの検索、加工、及び編集ができます。

実行できるデータベース操作は、次のとおりです。

- データの検索

図 1-26 にシステム構成例を示します。

図 1-26 XDM/RD E2 の外部表を利用する場合のシステム構成例



2

DABroker の機能（共通）

この章では、DABroker の機能について説明します。

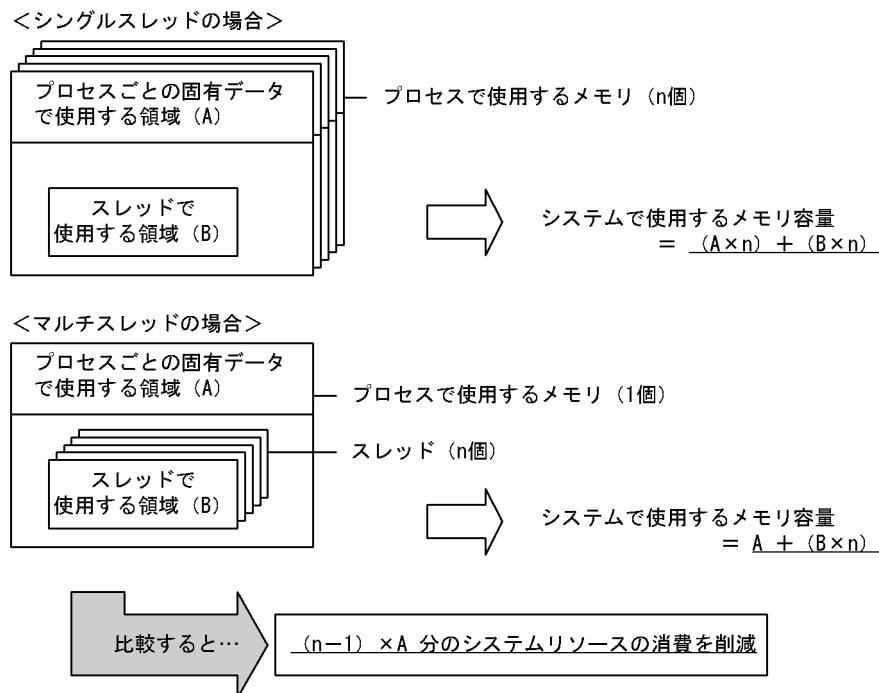
UNIX を使用する環境と Windows を使用する環境で共通の内容です。

2.1 マルチスレッドの実行

DABroker は、データベースアクセスをマルチスレッドで実行します。マルチスレッドで実行すると、シングルスレッドで実行する場合に比べてシステムリソースの消費を軽減でき、同時アクセス数（同時に実行できる処理数）の増加を実現します。

図 2-1 に示すように、シングルスレッドでは、プロセスごとにメモリ領域を確保しますが、マルチスレッドでは、一つのプロセスで使用するメモリ領域で複数スレッドのメモリ領域を確保できます。したがって、システムリソースの消費を軽減できます。

図 2-1 マルチスレッドとシングルスレッドでのシステムリソース消費量の比較

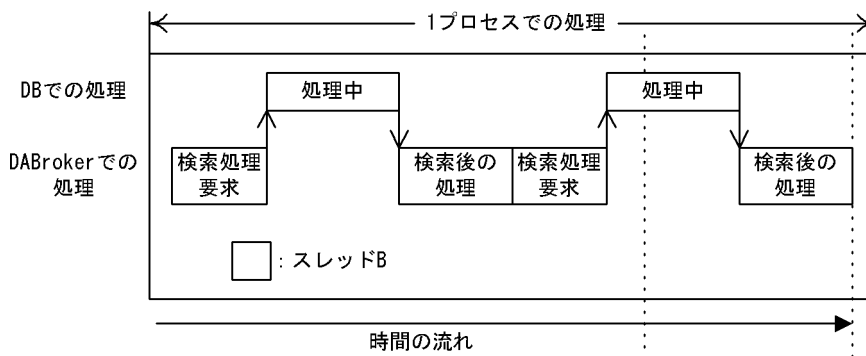


また、マルチスレッドで実行すると、シングルスレッドで実行する場合に比べて、処理にかかる時間を短縮できます。図 2-2 で同一処理を複数回実行する場合の例を示します。シングルスレッドでは、複数の処理を一つのスレッドで実行しますが、マルチスレッドでは、処理ごとにスレッドを生成し、スレッドを並行して実行できます。したがって、処理時間を短縮できます。スレッドを並行に実行することで処理時間を短縮するため、マルチスレッドは、非同期アクセス処理に適しています。

図 2-2 マルチスレッドとシングルスレッドでの処理時間の比較

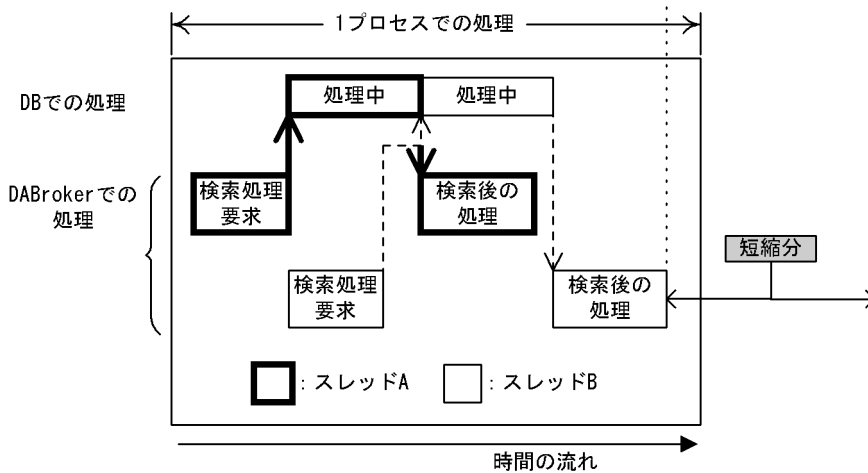
<シングルスレッドの場合>…(例) : 1プロセス1スレッドで2回の検索要求をした場合

この場合、1回目の処理が終わるまで2回目の処理を実行できません。



<マルチスレッドの場合>…(例) : 1プロセス2スレッドで2回の検索要求をした場合

この場合、1回目の処理と2回目の処理を並行して実行できます。



注 DBとは、データベースのことです。

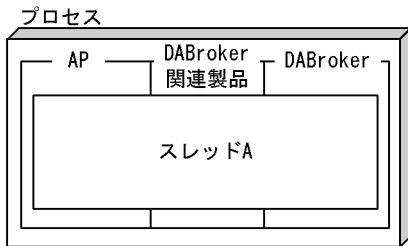
次に、DABroker で実現するスレッドの構成について説明します。

2.1.1 アプリケーションから実行するマルチスレッド

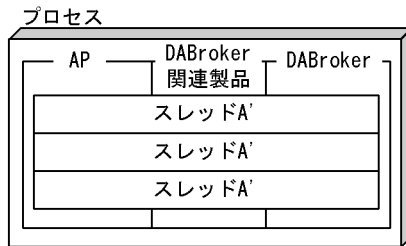
DABroker では、開発したアプリケーション、DABroker 関連製品、及び DABroker で一つのプロセスを生成して実行します。プロセスで生成するスレッドの構成は、開発するアプリケーションによって異なります。DABroker 関連製品が提供する API を使用すると、スレッドの制御方法などを意識することなく、マルチスレッドで実行するアプリケーションを開発できます。図 2-3 に、スレッドの生成例を示します。

図 2-3 スレッドの生成例

- 同期アクセス処理を実現する場合
(1プロセス = 1スレッド)

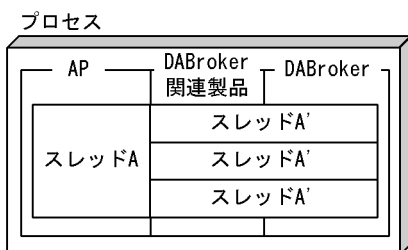


- DABroker関連製品のAPIを使用しないで
非同期アクセス処理を実現する場合
(1プロセス = nスレッド)
…複雑なコードを記述する必要がある

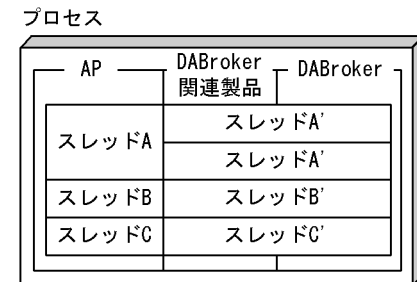


DABroker関連製品のAPIを使用すると…
スレッドの制御方法を意識することなく、
簡単にマルチスレッドで実行するアプリケーションが開発できる

- アプリケーションでDABroker関連製品のAPIを使って非同期アクセス処理を実現する場合 (1プロセス = nスレッド)



- アプリケーションでDABroker関連製品のAPIを使って非同期アクセス処理を実現する場合 (1プロセス = nスレッド)



アプリケーションからのマルチスレッドの実行は、同時に複数処理を実行したい場合に適しています。つまり、データベースへの非同期アクセス処理に適しています。

2.1.2 DBPARTNER から実行するマルチスレッド

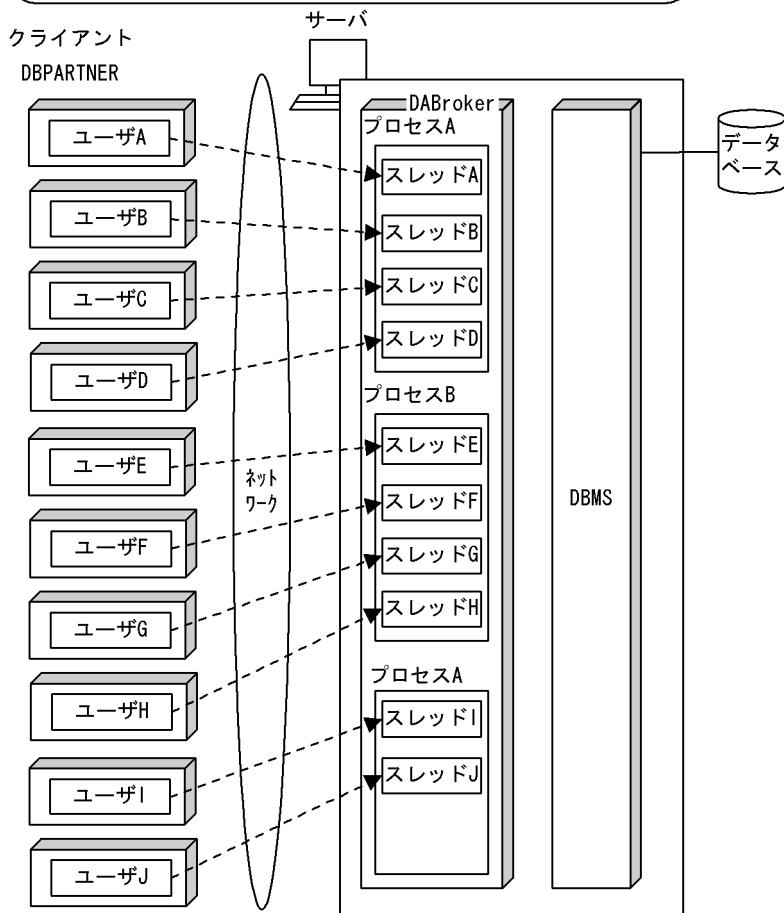
DABrokerは、DBPARTNERを使用する場合DABrokerにアクセスするユーザの単位にスレッドを生成します。図2-4に、DBPARTNERを使用する場合のスレッドの生成処理を示します。

図 2-4 DBPARTNER の場合の DABroker の処理方法

<クライアントから、10ユーザの処理要求がある場合>

<条件>

- ・最大接続ユーザ数 (DABMAXUSERS) : 10 (ユーザ)
- ・1プロセスあたりの実行ユーザ数 (DABMAXBLOCKUSERS) : 4 (ユーザ)
- ・最大処理プロセス数 (DABMAXPROCESS) : 3 (プロセス)



<参考>

- ・DABrokerで処理できる最大接続ユーザ数 : 12 (ユーザ)
- $$= \begin{matrix} 1 \text{プロセスあたりの実行ユーザ数} \\ : 4 \text{(ユーザ)} \end{matrix} * \begin{matrix} \text{最大処理プロセス数} \\ : 3 \text{(プロセス)} \end{matrix}$$

1 プロセスで処理するスレッド数 (= DABroker に接続するユーザ数) は決められているので、1 プロセスで処理するスレッド数を適切な値にすると DABroker に同時に接続できるユーザ数の増加にも効率的に対応できます。

1 プロセスで処理するスレッド数 (1 プロセスあたりの実行ユーザ数) は、DABroker 動作環境定義ファイルで設定します。

DABroker 動作環境定義ファイル、最大接続ユーザ数、最大処理プロセス数、1 プロセスあたりの実行ユーザ数については、次の箇所を参照してください。

UNIX の場合 : 「[3.4 DABroker の動作環境の設定](#)」

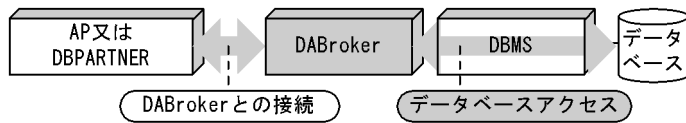
Windows の場合：「[4.5 DABroker 環境設定項目](#)」

設定値の見積もり方法については「[5.2 DBPARTNER を使用する場合のスレッドのチューニング](#)」を参照してください。

2.2 データベースアクセス

ここでは、DABroker のデータベースアクセスの処理方法について説明します。

図 2-5 データベースアクセス



データベースアクセスは、次のどれかの方法で実行します。なお、アクセスを要求するインタフェースによって実行できる処理方法は異なります。

2.2.1 同期アクセス処理

SQL の実行、ストアプロシジャの制御などを同期アクセスで実行します。

アプリケーション、DBPARTNER、HITSENER5、DBPARTNER Library、又は XDM/RD 外部表から要求を受けるごとに処理を実行し、アプリケーションなどに結果を返す処理を同期アクセス処理といいます。同期アクセス処理の場合、アプリケーションなどでは、結果が返ってくるまで（サーバでの処理が終了するまで）次の要求（又は別の操作）ができません。同期アクセス処理は、アプリケーション、DBPARTNER、HITSENER5、DBPARTNER Library、及び XDM/RD 外部表のどれからも要求できます。通常、DBPARTNER、HITSENER5、DBPARTNER Library、又は XDM/RD 外部表では同期アクセス処理になります。

2.2.2 非同期アクセス処理

SQL の実行、ストアプロシジャの制御などを非同期アクセスで実行します。

アプリケーションからの要求を受けると、先にアプリケーションに対して要求を受け付けたことを通知し、その後処理を実行します。アプリケーションでは、処理の結果を待たずに（サーバでの処理の終了を待たずに）次の要求（又は別の操作）ができます。アプリケーションから結果の確認要求があった場合、アプリケーションに対する処理結果を通知します。この処理を非同期アクセス処理といいます。なお、非同期アクセス処理は、アプリケーションから要求された場合だけ実行します。

2.2.3 DBPARTNER からの予約処理

DABroker が DBPARTNER の代わりにサーバに処理を要求し、処理結果を一括管理する処理のことです。検索したデータをデータベースに保存する処理を実行します。

DABroker は、DBPARTNER からの要求を受けると、先に DBPARTNER に対して要求を受け付けたことを通知し、その後処理を実行します。DBPARTNER では処理の結果を待たずに次の要求ができます。DBPARTNER に対する処理結果は、DBPARTNER からの結果の確認要求があった場合に通知します。なお、予約処理では、あらかじめ処理要求を登録しておくことはできません。

予約処理は、DBPARTNER/Client の GUI を使用しているクライアント、又は DBPARTNER/Web から実行できます。

2.2.4 DBPARTNER からのエージェント機能の実行

データ検索、更新などの実行したい処理内容をあらかじめエージェントとして登録できるため、移動先で公衆回線網を利用してデータベースをアクセスできます。また、定期的に行う定型業務としても活用できます。

エージェントは DBPARTNER (DBPARTNER2 Client, DBPARTNER/Client32, DBPARTNER2 Web, 又は DBPARTNER Mobile) から実行できます。なお、Database Connection Server 経由でメインフレームデータベースにアクセスする場合、エージェント機能は使用できません。

エージェントとは

データベースをアクセスする内容をエージェントとして DABroker に登録し、クライアントでエージェントの実行を指示するだけで、データベースをアクセスできます。

エージェントにはデータベースアクセスの実行時間、アクセス条件などの内容をあらかじめ登録しておきます。

エージェントは、登録された処理内容を定期的に行うことができるので、定型業務として活用できます。また、複数のユーザでエージェントを共有できます。さらに、エージェントで処理内容を設定できるため、クライアント・サーバ間のやり取りが減り、クライアントのシステム及び通信の負荷を軽減できます。

機能の内容

DBPARTNER からエージェント実行の要求を受けると、DABroker はエージェントを実行します。クライアントには、エージェントが成功したかどうかの実行結果、データベースの処理結果を返します。クライアントとの接続を解放してもエージェントの実行を受け付けている場合は、データベースをアクセスします。処理結果は、DABroker でためておき、次回接続の確立時にまとめてクライアントに返します。

また、クライアントは、処理結果を待たずに次のエージェントの実行を要求できます。DABroker はエージェント実行中に、別のエージェントを実行してデータベースをアクセスできます。

DABroker は次の内容を実行します。

- エージェントの実行

DBPARTNER を使用する場合は、DBPARTNER/Client32 又は DBPARTNER2 Client でエージェントを登録し、DBPARTNER Mobile - Client などからエージェントの実行を要求します。エージェントには、次の実行方法があります。

- **即時実行**

エージェント実行要求受け付け直後に実行します。

- **時間指定による実行**

エージェント実行要求を受け付けた後、指定時間にエージェントを実行します。

- **繰り返し実行**

エージェント実行要求を受け付けた後、指定時間にエージェントを実行し、次の実行時間が来るのを待ちます。

- **即時配信実行**

エージェント実行要求受け付け直後に実行し、その応答として結果を返します。なお、即時配信実行は、Mobile Message Queue を使用する場合だけ実行できます。

- **エージェント機能を使用するユーザのグループ管理**

DABroker では、エージェント機能を使用するユーザをグループで管理できます。グループで管理すると、グループに登録されていないユーザからの処理要求を拒否できます。グループ管理は、処理を要求するユーザ ID をグループ管理情報ファイルに登録して使用します。グループ管理については、「[5.1.3 エージェント機能を使用するユーザのグループ管理](#)」を参照してください。

2.2.5 分散トランザクション処理

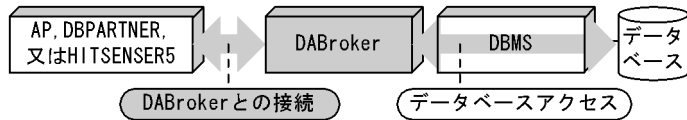
TPBroker の OTS 機能、又は Java JTS 機能を提供するトランザクションマネージャ (Java TM) を使って、TPBroker や Java TM にトランザクションの制御を任せることで、DABroker のデータベースアクセス処理と、TPBroker や Java TM と連携する他製品のリソースアクセス処理とを、一つのグローバルトランザクションとして同期制御できるようになります。

ワークフローを実現するアプリケーションを考えた場合、案件の詳細を登録するデータベースと、案件の流れを記録するメッセージキューが必要です。しかし、案件の詳細を登録するデータベースへのアクセスには DABroker を利用していて、メッセージキューへのアクセスには TPBroker と連携する他製品を利用していたとします。この場合でも、TPBroker の OTS 機能を両方のリソースアクセス処理で利用することで、二つのリソースアクセス処理を同期させることができます。

2.3 DABroker との接続

DABroker 関連製品を使用して開発するアプリケーション、DBPARTNER, HITSENSER5, DBPARTNER Library, 又は XDM/RD 外部表と、DABroker との接続について説明します。

図 2-6 DABroker との接続



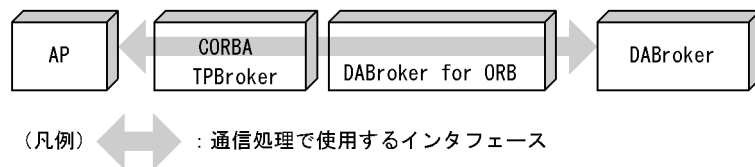
2.3.1 アプリケーションと DABroker との接続

アプリケーションと DABroker との接続（通信）処理は、API によって異なります。

- DABroker for .NET Framework, DABroker for C++及び DABroker for COBOL
DABroker for .NET Framework, DABroker for C++及び DABroker for COBOL で提供するサーバアプリケーション開発用インタフェースでは、DABroker との通信処理をサポートしていません。
- DABroker for ORB
DABroker for ORB が提供する Simple Interface 及びクライアントアプリケーション開発用部品は、CORBA オブジェクトです。DABroker for ORB が提供するオブジェクトを使ったアプリケーションと DABroker との接続には、CORBA オブジェクトを呼び出す TPBroker を利用します。

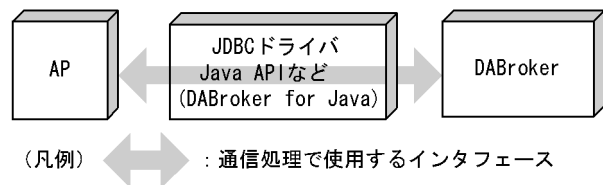
図 2-7 に、DABroker for ORB を使用した場合の通信処理について示します。

図 2-7 DABroker for ORB での通信処理



- DABroker for Java
DABroker for Java で開発したアプリケーションと DABroker の通信は、Java 標準 API (java.net クラスライブラリ), Java RMI (Java Remote Method Invocation) などを使用して実現します。DABroker for Java で Applet を開発した場合は、DABroker との通信トレースを取得できます。図 2-8 に、DABroker for Java を使用した場合の通信処理について示します。

図 2-8 DABroker for Java での通信処理

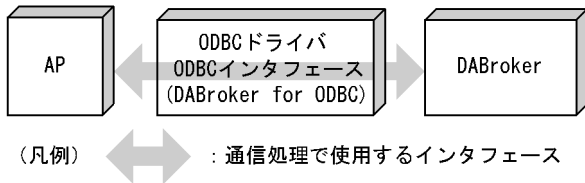


- DABroker for ODBC

DABroker for ODBC で開発したアプリケーションと DABroker の通信は、ODBC インタフェースを使用して実現します。DABroker for ODBC で LAN 接続用のサーバアプリケーションを開発した場合は、DABroker との通信トレースを取得できます。

図 2-9 に、DABroker for ODBC を使用した場合の通信処理について示します。

図 2-9 DABroker for ODBC での通信処理



2.3.2 DBPARTNER, HITSENSER5, DBPARTNER Library, 又は XDM/RD 外部表と DABroker との接続

DBPARTNER, HITSENSER5, DBPARTNER Library, 又は XDM/RD 外部表との通信時、DABroker は TCP/IP を使用して接続の確立と解放を次のように処理します。

同期アクセス、又は DBPARTNER からの予約処理の場合

クライアントからの要求に応じて接続を確立、解放します。接続の確立中にサーバはクライアントからのデータベースアクセス要求を受け付けます。解放した接続を再び確立するには、クライアントを再起動する、又はクライアントから再度、接続の確立を要求します。

DBPARTNER からのエージェント機能を使用している場合

エージェント実行要求に応じて、接続を確立、解放します。接続の確立中にサーバはクライアントからのデータベースアクセス要求を受け付けます。また、接続を解放しても、受け付けたデータベースのアクセスは実行します。解放した接続は、クライアントからのエージェント実行を受け付けた時に再び確立します。

3

DABroker の環境設定と運用 (UNIX)

この章では、UNIX で DABroker を使用するための環境設定の方法、及び運用の方法について説明します。

3.1 環境設定の手順

ここでは、DABroker を使用する環境の設定手順について説明します。なお、説明は、次のことを前提にしています。

- DBMS を使用できる環境が既に構築されていること
- 開発アプリケーション、及び DABroker を組み込むマシンをネットワーク上で使用できる環境が既に構築されていること
- DBPARTNER, HITSENSER5, DBPARTNER Library, 又は XDM/RD 外部表からのアクセスを利用する場合は、DBPARTNER, HITSENSER5, DBPARTNER Library, 又は XDM/RD 外部表を組み込むマシンをネットワーク上で使用できる環境が既に構築されていること
- RDA Link for Gateway 経由でメインフレーム系データベースにアクセスする場合、RDA Link for Gateway が DABroker を組み込むマシンに組み込まれていること

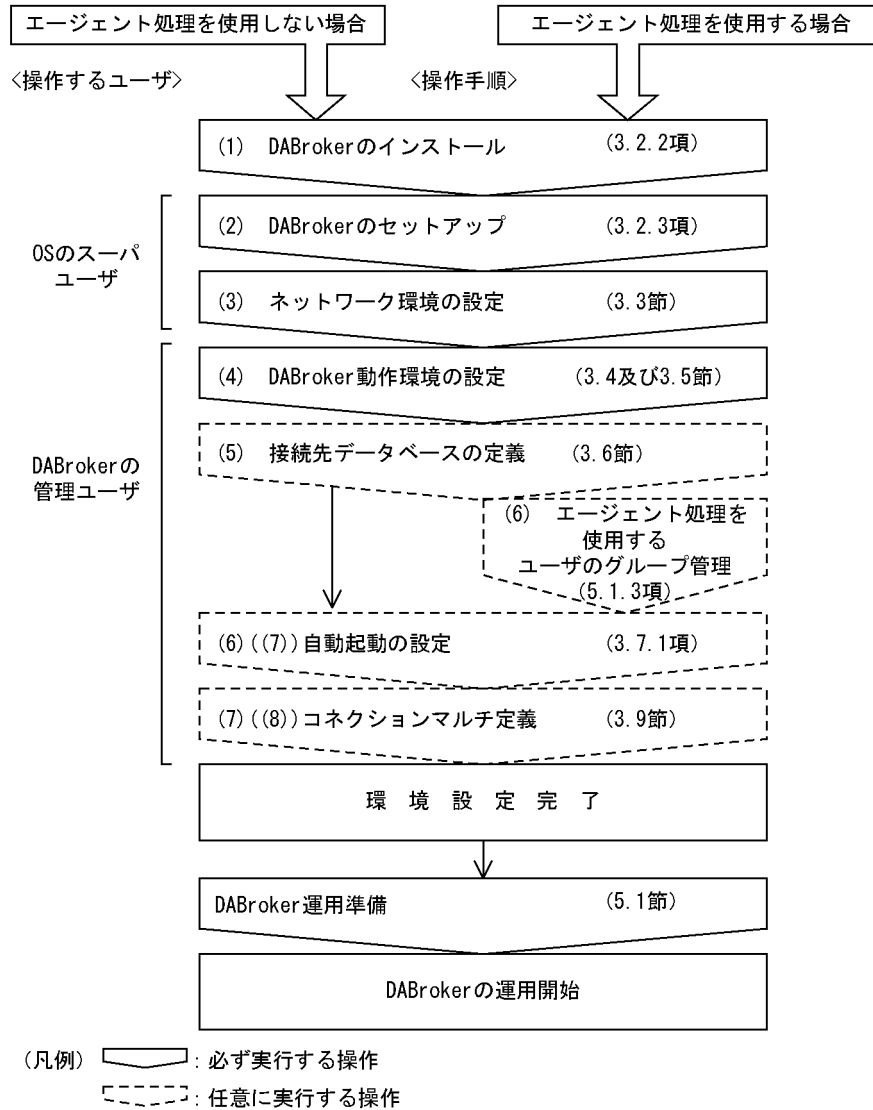
DABroker の環境設定の操作手順を、DABroker 関連製品を使用する場合と DBPARTNER, HITSENSER5, DBPARTNER Library, 又は XDM/RD 外部表を使用する場合に分けて説明します。なお、この章以降 DABroker と直接のインタフェースを持つプログラムのことをクライアント、DABroker を組み込むマシンをサーバと呼びます。

3.1.1 DABroker 関連製品を使用する場合の環境設定

DABroker 関連製品を使用する場合の環境設定の手順を図 3-1 に示します。

図 3-1 では、環境設定完了後、最初に DABroker を運用するまでの操作を示しています。

図 3-2 DBPARTNER, HITSENER5, DBPARTNER Library, 又は XDM/RD 外部表を使用する場合の環境設定の手順



3.2 DABroker のインストールとセットアップ

ここでは、DABroker のインストールとインストール後に実行するセットアップ操作について説明します。

3.2.1 DABroker をインストールするマシン

使用するデータベースによって、DABroker をインストールできるマシンが決まります。DABroker をインストールするマシンについて、使用するデータベースごとに説明します。

- HiRDB 又は ORACLE にアクセスする場合

DABroker は、DBMS のサーバプログラム又は DBMS のクライアントプログラムがインストールされたマシンにインストールしてください。なお、「1.2 DABroker の使用環境」では、DABroker を DBMS のサーバプログラムがインストールされたマシンにインストールする形態を示しています。

- RDA Link for Gateway を使ってメインフレーム系データベースにアクセスする場合

DABroker は、RDA Link for Gateway と同一マシンにインストールしてください。

3.2.2 DABroker のインストール

DABroker のインストールは、標準インストーラ機能 (Hitachi PP Installer) を使用して実行します。インストール先ディレクトリは、/opt/DABroker です。

表 3-1 に、DABroker のインストール時に作成されるファイル及びディレクトリの中で、DABroker の関連製品で開発するアプリケーションからデータベースにアクセスする場合に必要なファイルとディレクトリを示します。

表 3-1 DABroker のインストール時に作成されるディレクトリとファイル

インストール時に作成されるディレクトリとファイル		説明
/opt/DABroker	—	<インストール先ディレクトリ>
/bin	—	<実行コマンド格納ディレクトリ>
	/dabsetup	DABroker セットアップコマンド
	/dabstart	DABroker の起動
	/dabstop	DABroker の終了
	/dabstat	DABroker の動作状態の表示
	/dabcltrc	クライアントとの通信トレースの表示
	/dabdbtrc	データベースアクセストレースの表示
	/dabagtreinfo	エージェント登録情報の表示*

インストール時に作成されるディレクトリとファイル		説明
	/dabagtexeinfo	エージェント実行情報の表示※
	/dabcmstart	コネクションマルチ機能の起動
	/dabcmstop	コネクションマルチ機能の終了
	/dabcmstat	コネクションマルチ機能の動作状態の表示
	/lib	<ライブラリ格納ディレクトリ> ネイティブライブラリ (libdaJavaC.sl) など
	/obj	<オブジェクトファイル用ディレクトリ>
	/AGT	<エージェント処理用ディレクトリ>
	/AGTGRP	<グループ管理用ディレクトリ>
	/msg	<メッセージ用ディレクトリ>
	/sample	-
	/dasyconf	DABroker 動作環境定義ファイルのサンプル
	/dadbenv	接続先データベース定義ファイルのサンプル
	/damconenv	コネクションマルチ定義ファイルのサンプル
	/conf	<DABroker 動作環境定義ファイルなどの格納ディレクトリ>
	/spool	<DABroker スプール情報格納ディレクトリ>
	/patch_dab	<修正パッチ履歴ディレクトリ>
	./dabrokerenv	<DABroker の運用情報ファイル>
	./HTC_2340.inf	<DABroker トラブルシュート取得情報ファイル>

注※

Red Hat Linux では作成されません。

3.2.3 DABroker のセットアップ

インストール後、次の操作を実行します。

- DABroker セットアップコマンドの実行

DABroker をインストール後、DABroker セットアップコマンド (dabsetup コマンド) を実行し、DABroker を運用するディレクトリ及び管理ユーザを設定します。DABroker セットアップコマンドの実行方法については、「3.2.3(1) セットアップコマンドの実行手順」を参照してください。「DABroker 運用ディレクトリ」とは、DABroker が提供するファイル及びディレクトリを格納し、DABroker を実行するディレクトリのことです。

なお、DABroker セットアップコマンド実行後に DABroker 関連製品をインストールした場合は、DABroker 関連製品で提供するセットアップコマンドを実行してください。DABroker 関連製品で提供するセットアップコマンドについては、DABroker 関連製品の各マニュアルを参照してください。

- **ライブラリ格納ディレクトリの登録**

- **HP-UX の場合：SHLIB_PATH (OS の環境変数) への登録**

DABroker で提供する、ライブラリ格納ディレクトリ「DABroker 運用ディレクトリ/lib」を SHLIB_PATH に登録し、DABroker を運用できる状態にしてください。

また、HiRDB を使用する場合は HiRDB ライブラリのパス名 (標準値：/opt/HiRDB_P/client/lib)、ORACLE を使用する場合は ORACLE クライアントライブラリ (32 ビット) (libclntsh.sl) のパス名も SHLIB_PATH に登録してください。

- **Solaris の場合：LD_LIBRARY_PATH (OS の環境変数) への登録**

DABroker で提供する、ライブラリ格納ディレクトリ「DABroker 運用ディレクトリ/lib」を LD_LIBRARY_PATH に登録し、DABroker を運用できる状態にしてください。

また、HiRDB を使用する場合は HiRDB ライブラリのパス名 (標準値：/opt/HiRDB_P/client/lib)、ORACLE を使用する場合は ORACLE クライアントライブラリ (32 ビット) (libclntsh.so) のパス名も LD_LIBRARY_PATH に登録してください。

- **AIX の場合：LD_LIBRARY_PATH, 及び LIBPATH (OS の環境変数) への登録**

DABroker で提供する、ライブラリ格納ディレクトリ「DABroker 運用ディレクトリ/lib」を LD_LIBRARY_PATH, 及び LIBPATH に登録し、DABroker を運用できる状態にしてください。

また、HiRDB を使用する場合は HiRDB ライブラリのパス名 (標準値：/opt/HiRDB_P/client/lib)、ORACLE を使用する場合は ORACLE クライアントライブラリ (32 ビット) (libclntsh.a) のパス名も LD_LIBRARY_PATH, 及び LIBPATH に登録してください。

- **Red Hat Linux の場合：LD_LIBRARY_PATH (OS の環境変数) への登録**

DABroker で提供する、ライブラリ格納ディレクトリ「DABroker 運用ディレクトリ/lib」を LD_LIBRARY_PATH に登録し、DABroker を運用できる状態にしてください。

(1) セットアップコマンドの実行手順

DABroker セットアップコマンドの実行手順について次に示します。

1. DABroker を管理するユーザ (UNIX ログインユーザ) を OS に登録する。ユーザ ID 登録後、パスワードも登録する。

登録方法は、使用する OS のマニュアルを参照してください。

ここで登録したユーザでログインするユーザは、「DABroker の管理ユーザ」と呼びます。DABroker 管理ユーザに与えられる権限は次のとおりです。

- DABroker で提供するファイル及びディレクトリの所有者としてのアクセス権
- DABroker の運用コマンドの実行権

<注意>

ORACLE にアクセスする場合は、ユーザ ID を「3」以上にしてください。

2. DABroker インストール先ディレクトリのグループ/ユーザを、DABroker の管理グループ/ユーザに変更する。

DABroker インストール先ディレクトリのアクセス権を確認し、DABroker の管理グループ/ユーザに変更しておいてください。

3. DABroker セットアップコマンド (dabsetup) を実行する。

dabsetup コマンドによって、DABroker 運用ディレクトリがセットアップされます。

セットアップ後、DABroker 運用ディレクトリ、および DABroker 運用ディレクトリにあるディレクトリやファイルの所有者は、DABroker の管理グループ/ユーザになります。

また、dabsetup コマンドの引数にパスを指定することで、DABroker 運用ディレクトリを任意の場所に変更することもできます。

dabsetup コマンドの実行方法については、「[3.2.3\(1\)\(a\) DABroker セットアップコマンド](#)」を参照してください。

なお、DABroker インストール直後の DABroker 運用ディレクトリは、インストール先ディレクトリ (/opt/DABroker) に設定されています。

(a) DABroker セットアップコマンド

次に、dabsetup コマンドの使用方法を示します。

形式

```
# dabsetup [-d | パス名]
```

機能

dabsetup コマンドは、DABroker の運用ディレクトリを指定するコマンドです。

dabsetup コマンドは、インストール先ディレクトリにインストールされたディレクトリ及びファイルを、指定された運用ディレクトリに移動します。

また、dabsetup コマンドは、運用ディレクトリの削除も実行します。

使用できるユーザ

OS のスーパーユーザ

指定値

パス名

DABroker 運用ディレクトリを変更する場合に指定します。

dabsetup コマンドは、ここで指定された場所に DABroker 運用ディレクトリを作成します。

パス名は、フルパスで指定してください。

また、パス名に次の場所を指定することはできません。

- リモートファイルシステム上のディレクトリ
- リモートファイルシステム
- ローカルファイルとリンクしたディレクトリ

なお、パス名の指定を省略した場合、すでに設定されている DABroker 運用ディレクトリに再作成します。

-d

DABroker 運用ディレクトリを削除する場合に指定します。再度 DABroker を使用する場合は、DABroker を再インストールしてください。

注意事項

- dabsetup コマンドを実行すると、パス名に指定された場所に「DABroker」ディレクトリを作成し、DABroker の実行に必要なディレクトリやファイルをインストール先ディレクトリ (/opt/DABroker) からコピーします。
dabsetup コマンド実行後、インストール先ディレクトリ (/opt/DABroker) には、DABroker の実行に必要なディレクトリやファイルが残ったままになります。
- dabsetup コマンドは、DABroker が停止中に実行してください。
- dabsetup コマンドで DABroker を再度セットアップする場合、運用ディレクトリでの DABroker の動作状態を初期状態にします。ただし、ユーザが作成した DABroker 動作環境定義ファイル、エージェント関連ファイル、ログファイル及びトレースファイルなどはセットアップしません。

コマンド実行時の注意事項

- 運用ディレクトリとして指定するパスが、ファイルとして存在する場合は、dabsetup コマンドの実行を中止します。
- 運用ディレクトリとして指定するパスに、DABroker が提供するディレクトリと同一名称のファイル、又は DABroker が提供するファイルと同一名称のディレクトリが存在する場合は、dabsetup コマンドの実行を中止します。
- 運用ディレクトリとして指定したディレクトリが既に存在する場合は、上書きするかどうかをコマンド実行者に確認します。なお、上書きを指定した場合、指定したディレクトリに存在する、DABroker がセットアップしないファイル (DABroker 動作環境定義ファイル、エージェント関連ファイルなど) は保証されます。

コマンドの実行が失敗した場合の注意事項

- dabsetup コマンドの実行が失敗した場合は、DABroker は、dabsetup コマンドの実行を終了し、dabsetup コマンドを実行する前の状態に戻します。dabsetup コマンドの実行が完了する前に、DABroker 関連製品のセットアップコマンドを実行しないでください。
- DABroker 関連製品のセットアップコマンドの実行が失敗した場合は、DABroker は、dabsetup コマンドを実行する前の状態に戻します。したがって、DABroker のセットアップ (dabsetup コマンドの実行) からやり直してください。

DABroker 関連製品をセットアップする場合の注意事項

- dabsetup コマンド実行後に DABroker 関連製品をインストールした場合は、DABroker 関連製品で提供するセットアップコマンドを実行してください。

なお、dabsetup コマンド実行前に、DABroker 関連製品をインストールした場合は、dabsetup コマンドで DABroker 関連製品のセットアップも実行しますので、DABroker 関連製品で提供するセットアップコマンドを実行する必要はありません。

DABroker 運用ディレクトリを削除する場合の注意事項

- dabsetup コマンドでの運用ディレクトリの削除に失敗した場合は、残ったディレクトリ及びファイルを OS の rm コマンドで削除してください。
- dabsetup コマンドでの運用ディレクトリの削除に成功した場合も、インストール先ディレクトリに dabsetup コマンドなどのシステムで必要なファイルが残ります。インストール先ディレクトリに残ったファイルは、インストーラで削除してください（アンインストールしてください）。
- ユーザが作成した DABroker 動作環境定義ファイル、エージェント関連ファイルは、dabsetup コマンドの実行で削除されません。必要に応じて削除してください。

3.3 ネットワーク環境の設定

ここでは、DBPARTNER、HITSENER5、DBPARTNER Library、又は XDM/RD 外部表を使用する場合に、`/etc/services` ファイルに設定する DABroker のサービス名及びポート番号について説明します。

また、Database Connection Server との接続にサービス名を使用する場合の、`/etc/services` ファイルでの設定方法について説明します。

3.3.1 DBPARTNER, HITSENER5, DBPARTNER Library, 又は XDM/RD 外部表を使用する場合

DBPARTNER、HITSENER5、DBPARTNER Library、又は XDM/RD 外部表を使用する場合、`/etc/services` ファイルに、TCP/IP 環境での DABroker のサービス名及びポート番号として、次の記述をファイルの行末に追加してください。

```
DABROKER_SV XXXXX/tcp
```

DABROKER_SV : DABroker のサービス名です。

XXXXX : ポート番号です。

tcp : プロトコル名です。TCP/IP を使用することを表します。

注意事項

- XXXXX には DABroker がオープンするポート番号を指定します。推奨値は 40179 です。オペレーティングシステムによっては推奨値がエフェメラルポートの範囲に当たるので、その場合は別のポート番号を指定してください。このポート番号とクライアント側のポート番号には同じ値を指定します。
- NIS (Network Information Service) を使用している場合、ネットワーク環境を NIS のサーバで設定してください。NIS のサーバを使用している場合、ネットワーク環境の設定は、NIS のサーバでの設定が有効になります。
- クライアント側では、使用する通信ドライバに、DABroker の組み込まれたサーバのホスト名、IP アドレス及びポート番号を登録してください。

3.3.2 Database Connection Server を経由してメインフレーム系データベースにアクセスする場合

Database Connection Server との接続には、DABroker では、Database Connection Server のポート番号又はサービス名を指定します。ポート番号又はサービス名は、コネクションマルチ定義ファイル ([「3.9 DABroker コネクションマルチ定義\(Database Connection Server 経由の DB アクセス定義\)」](#) 参照)、又は DABroker 動作環境定義ファイルの DAB_DBMS_PORT ([「3.5.1 共通項目」](#) 参照) で指定します。このとき、サービス名を指定する場合は、`/etc/services` ファイルに、サービス名とそれに対応するポート番

号を設定してください。Database Connection Server との接続にポート番号を使用する場合は、ここでの設定は不要です。

例えば、ポート番号（接続する Database Connection Server のサーバ空間のポート番号(サーバスケジュール番号)）が 40179 に対して、サービス名：DBCSV を設定する場合は、次の記述をファイルの行末に追加します。

```
DBCSV 40179/tcp
```

注意事項

- サービス名には、任意の名称が設定できます。
- ここで設定したサービス名を、コネクションマルチ定義ファイル（[「3.9 DABroker コネクションマルチ定義\(Database Connection Server 経由の DB アクセス定義\)」](#) 参照）、及び DABroker 動作環境定義ファイルの DAB_DBS_PORT（[「3.5.1 共通項目」](#) 参照）で指定してください。
- ポート番号は、Database Connection Server のサーバ空間のポート番号(サーバスケジュール番号)にしてください。

3.4 DABroker の動作環境の設定

DABroker の動作環境を、システム構成に合わせて設定する必要があります。

DABroker が動作するために必要な項目（環境変数）とその値を、DABroker 動作環境定義ファイルに記述します。設定する項目については、「3.5 DABroker の動作環境の設定項目」を参照ください。

DBPARTNER, HITSENSER5, DBPARTNER Library, 又は XDM/RD 外部表を使用する場合、DABroker の起動時にこのファイルの内容が有効になります。

なお、DABroker 動作環境定義ファイルに記述する項目を、シェルの環境変数として設定することもできます。シェルの環境変数として設定した場合は、DABroker 動作環境定義ファイルで設定した値よりもシェルで設定した値を有効とします。

3.4.1 DABroker 動作環境定義ファイルの作成

DABroker では、DABroker 動作環境定義ファイルに、DABroker の動作に関する情報を定義します。DABroker 動作環境定義ファイルは、インストール時に提供されるサンプルを使用して作成します。

表 3-2 に、使用するファイルとファイルの格納先を示します。

表 3-2 DABroker 動作環境定義ファイルのファイル名

使用するサンプルファイル	DABroker 動作環境定義ファイル格納先
/opt/DABroker/sample/dasysconf	DABroker 運用ディレクトリ/conf

DABroker 動作環境定義ファイルの作成手順を次に示します。

1. サンプルファイルを、表 3-2 に示した DABroker 動作環境定義ファイルの格納先へコピーする。
2. コピー先ファイルの設定項目の標準値を必要に応じて変更する。また、値が設定されていない項目には値を入力する。
3. ファイルを保存する。

(1) ファイルの構成内容

提供するサンプルファイルにあるように、DABroker 動作環境定義ファイルは、「DAB」ブロックと「OTHERS」ブロックで構成されます。

「DAB」ブロック

DABroker の動作環境の設定項目を指定します。

DABroker では、設定項目のチェックを実行し、不正な名称などを指定した場合はエラーになります。

なお、ここで指定する項目については、「[3.5.1 共通項目](#)」及び「[3.5.2 リモートアクセス設定項目](#)」を参照してください。

「OTHERS」ブロック

DBMS などの設定項目を指定します。

DABroker では、設定項目のチェックを実行しません。このセクションに指定されたすべての項目を有効とします。

なお、ここで指定する項目については、「[3.5.4 HiRDB の設定](#)」及び「[3.5.5 ORACLE の設定](#)」を参照してください。

(2) ファイル作成時の注意事項

- 値を変更しない設定項目は削除しないでください。
- 設定項目の export は不要です。
- 「=」の前後に空白文字、又はタブコードを入れないでください。入れた場合はエラーとなります。
(エラーの例) `DABTMP = /tmp/dab`
- 設定値に半角空白文字を使用する場合は、「'(シングル引用符)」で設定値を囲んで指定してください。
(指定例) `DABTMP=' /tmp/da b'`
- 設定値の後ろに半角空白文字、タブコード、「#」以外の文字を記載した場合は、エラーとなります。
(エラーの例) `DABTMP = /tmp/dab;`
- 「#」以降は、コメントとして扱います。
- 設定値には、値だけを指定します。「\$設定項目」で指定する値を引き継ぐことはできません。

(3) 作成したファイルの変更

- 作成した DABroker 動作環境定義ファイルの内容の変更は、DABroker が起動していない時に行ってください。

3.5 DABroker の動作環境の設定項目

DABroker の動作環境を設定する項目には、次の種類があります。

- 共通項目

DABroker 関連製品で開発したアプリケーション、又はリモートアクセスのどの方法でデータベースにアクセスする場合にも設定します。

DABroker for Java Version 2 から DABroker にローカルアクセスする場合は、リモートアクセス設定項目で設定する次の項目についても有効になります。

- DABORADBA_LONGBUFSIZE
- DABHIRDBA_BLOBBUFSIZE
- DABDBSDBA_BLOBBUFSIZE

これらの項目の設定方法については、「[3.5.2 リモートアクセス設定項目](#)」を参照してください。

- リモートアクセス設定項目

リモートアクセス使用時に設定します。

リモートアクセスについては、「[1.2 DABroker の使用環境](#)」を参照してください。

- イベントトレースの設定

DABroker 関連製品のイベントトレースを取得する場合に設定します。

- for C++設定項目
- for COBOL 設定項目
- for ORB 設定項目

- HiRDB の設定項目

接続するデータベースに HiRDB を使用する場合に設定します。

- ORACLE の設定項目

接続するデータベースに ORACLE を使用する場合に設定します。

- Database Connection Server 接続設定項目

Database Connection Server を経由して、ホストのデータベースを使用する場合に設定します。

- エージェント処理機能の設定項目

DBPARTNER のエージェント処理機能を使用する場合に設定します。

3.5.1 共通項目

DABroker の動作環境での共通項目について説明します。

標準値は DABroker 動作環境定義ファイルのサンプルファイルにあらかじめ設定されている値です。

共通項目の一覧を、表 3-3 に示します。

表 3-3 共通項目の一覧

設定項目	説明	標準値	指定範囲
DAB_LANG	LANG 環境変数	HP-UX の場合 ja_JP.SJIS Solaris の場合 ja_JP.PCK AIX 又は Linux の場合 C	詳細は「3.5.1(1) DAB_LANG (LANG 環境 変数)」参照。
DABTMP	作業ディレクトリ指定	/tmp	任意ディレクトリ ※1
DABSQL_TRCSIZE※2	データベースアクセストレースの取得	-1	0 又は -1
DABSELECTCOLCNT	表検索カラム数	0	0,1~2147483647
DABSELECTINFCNT	更新・検索項目数	0	0,1~2147483647
DABORACLE_TYPE	使用する ORACLE のバージョン	805	詳細は「3.5.1(6) DABORACLE_TYPE (使 用する ORACLE のバー ジョン)」参照。
DABSQL_HiRDB_ DICTIONARY_ DATATYPE	HiRDB でのディクショナリ 表のデータ属性	C	M 又は C
DABSVLOGSIZE	DABroker のログ取得ファ イルサイズ	64 (キロバイト)	0, 64~32767 (キロバイト)
DABXA_LOG_ERROR	XA トレースを取得する	ON	ON 又は OFF
DABEXSQL_TRC_LINE	拡張データベースアクセス トレースを取得する	-1	-1, 0, 又は 1024~ 32767(行)
DAB_HIRDB_DBINF_ESC	HiRDB 環境変数区切り文字	@	@ ! \$ % & * + - / : ? ¥ ;

注※1

DABroker の管理ユーザに READ/WRITE 権限がある任意のディレクトリです。ただし、パーティションを圧迫しないところです。

注※2

DABroker 稼働時でも、シェルの環境変数で設定された値が有効になります。

(1) DAB_LANG (LANG 環境変数)

標準値

HP-UX の場合

ja_JP.SJIS

Solaris の場合

ja_JP.PCK

AIX の場合

C

Red Hat Linux の場合

C

指定範囲

HP-UX の場合

ja_JP.SJIS : SJIS モード

ja_JP.EUC, ja_JP.UJIS, 又は ja_JP.eucJP : EUC モード

C : ASCII モード

UTF-8 : UTF-8 モード

Solaris の場合

ja_JP.PCK : SJIS モード

ja 又は japanese : EUC モード

C : ASCII モード

UTF-8 : UTF-8 モード (HiRDB 使用時だけ指定できます)

AIX の場合

Ja_JP, Ja_JP.IBM-932, 又は Ja_JP.IBM-943 : SJIS モード

ja_JP, 又は ja_JP.IBM-eucJP : EUC モード

C : ASCII モード

UTF-8 : UTF-8 モード

Red Hat Linux の場合

ja_JP, ja_JP.ujis, japanese, japanese.euc, ja_JP.eucjp, 又は Ja_JP : EUC モード

C : ASCII モード

UTF-8 : UTF-8 モード

説明

使用する母国語（日本語、英語など）及び文字コード（SJIS コードなど）を設定します。

表 3-4 各言語モードで操作できるデータベース

データベース	DABroker での設定			
	SJIS モード	日本語 EUC モード	ASCII モード	UTF-8 モード
HiRDB	○	○	○	○
ORACLE	○	○	○	○*
VOS3 XDM/RD	○	×	○	×
VOS3 XDM/SD	○	×	○	×
VOSK SQL/K	○	×	○	×
VOS1 RDB1 E2	○	×	○	×
VOS1 PDM II E2	○	×	○	×
VOSK スプールファ イル	○	×	○	×

凡例

○：操作できる

×：操作できない

各データベースで使用する言語モードは、必ず DABroker の言語モードと合わせてください。データベースと DABroker の言語モードが異なっている場合は動作を保証しません。

クライアントが DABroker for Java の場合は、DAB_LANG の指定が JDBC の文字エンコーディングに影響します。詳細については、マニュアル「DABroker for Java™ Version 2 DABroker Driver for Java™ Technology」の「接続するデータベースの指定」の ENCODELANG の説明を参照してください。

DABroker を日本語 EUC モード、ASCII モード、又は UTF-8 モードで動作させた場合、メッセージは英文で出力されます。メッセージの詳細については、「7 メッセージ(共通)」を参照してください。

注※

Solaris の場合は設定できません。

(2) DABTMP (DABroker 作業ディレクトリ)

標準値

/tmp

指定範囲

DABroker の管理ユーザに READ/WRITE 権限がある任意のディレクトリ (ただし、パーティションを圧迫しないところ)

説明

DABroker の作業ディレクトリを指定します。このディレクトリは、予約処理用の SQL の格納（Red Hat Linux を除く）や、処理途中で作成される一時ファイルの格納などに使用されます。

なお、ここで/tmp を指定した場合は、OS の再起動時に、/tmp ディレクトリ下に作成されたすべてのファイルは削除されます。

(3) DABSQL_TRCSIZE（データベースアクセストレースの取得）

標準値

-1

指定範囲

0 又は -1

説明

データベースアクセストレース（dabdbtrc コマンドで表示）を取得するかどうかを設定します。

- 0 を設定した場合

すべてのアクセストレースを取得するファイルが、[DABroker 運用ディレクトリ/spool/db_access] の下に作成されます。ファイル名は、次のとおりです（XXXXXX はシステム任意の番号）。
db_XXXXXX

ただし、ハードディスクの空き容量がなくなった場合、それ以降のアクセストレースは取得しません。なお、格納する領域のサイズは設定できません。アプリケーション、又はクライアントからの要求量が多ければ、トレースのファイル容量も多くなるため、ハードディスクの容量に余裕をもたせておいてください。必要容量の概算方法は、「6.4.2 データベースアクセストレースの表示」を参照してください。また、アプリケーション、又はクライアントからの要求量が多ければ、DABroker を終了するまでにも時間がかかります。御注意ください。

- -1 を設定した場合

データベースアクセストレースを取得しません。

(4) DABSELECTCOLCNT（表検索カラム数）

標準値

0

指定範囲

0, 又は 1~2147483647

説明

表検索用の領域として確保する表カラムの個数を指定します。

- 0 を設定した場合

DBMS に対して表カラムの個数を要求する SQL を発行します。DBMS から返された結果を基に、表検索用の領域を確保します。

- 1～2147483647 を設定した場合

表カラムの個数が指定した値より少ない場合、DBMS に対して表カラムの個数を要求する SQL を発行しません。このため、SQL を発行する場合より処理性能が向上します。ただし、検索を実行するたびに次に示す大きさのメモリが必要になるため、動作環境を考慮して値を設定してください。よく検索する表のカラム数を大きく超える値を設定した場合、性能劣化の原因になることがあります。

HiRDB, ORACLE の場合：1 カラム当たり 128 バイト

RDA Link for Gateway の場合：1 カラム当たり 256 バイト

上記以外の DBMS の場合：0 だけしか有効となりません（項目の個数は設定できますが、無条件で 0 となります）。

表カラムの個数が指定した値より多い場合、DBMS に対して表カラムの個数を要求する SQL を発行します。DBMS から返された結果を基に、表検索用の領域を確保します。

(5) DABSELECTINFCNT (更新・検索項目数)

標準値

0

指定範囲

0, 又は 1～2147483647

説明

表の更新・検索用の領域として確保する、入力項目（パラメタ）の個数を指定します。

- 0 を設定した場合

DBMS に対して入力項目の個数を要求する SQL を発行します。DBMS から返された結果を基に、表の更新・検索用の領域を確保します。

- 1～2147483647 を設定した場合

入力項目の個数が指定した値より少ない場合、DBMS に対して入力項目の個数を要求する SQL を発行しません。このため、SQL を発行する場合より処理性能が向上します。ただし、更新・検索を実行するたびに次に示す大きさのメモリが必要になるため、動作環境を考慮して値を設定してください。よく更新・検索する表の入力項目の個数を大きく超える値を設定した場合、性能劣化の原因になることがあります。

HiRDB, ORACLE の場合：1 カラム当たり 128 バイト

RDA Link for Gateway の場合：1 カラム当たり 256 バイト

上記以外の DBMS の場合：0 だけしか有効となりません（項目の個数は設定できますが、無条件で 0 となります）。

入力項目の個数が指定した値より多い場合、DBMS に対して入力項目の個数を要求する SQL を発行します。DBMS から返された結果を基に、表の更新・検索用の領域を確保します。

(6) DABORACLE_TYPE (使用する ORACLE のバージョン)

標準値

HP-UX の場合

805

Solaris の場合

805

AIX の場合

817

Red Hat Linux の場合

920

指定範囲

HP-UX の場合

805, 806, 815, 816, 817, 901, 920, 101, 又は 111

Solaris の場合

805, 806, 815, 816, 817, 901, 920, 101, 又は 111

AIX の場合

817, 920, 101, 又は 111

Red Hat Linux の場合

920, 101, 111, 121 又は 190

説明

使用する ORACLE のバージョンを設定します。

805 : Oracle8 R8.0.5

806 : Oracle8 R8.0.6

815 : Oracle8i R8.1.5

816 : Oracle8i R8.1.6

817 : Oracle8i R8.1.7

901 : Oracle9i R9.0.1

920 : Oracle9i R9.2.0

101 : Oracle10g R10.1.0 又は R10.2.0

111 : Oracle11g R11.1.0 又は R11.2.0

121 : Oracle12c R12.1.0 又は R12.2.0

190 : Oracle19c R19.0.0 以降

(7) DABSQL_HiRDB_DICTIONARY_DATATYPE (HiRDB でのディクショナリ表のデータ属性)

標準値

C

指定範囲

M 又は C

説明

HiRDB のディクショナリ表の列属性を指定します。

- M を設定した場合
混在文字データ (MVARCHAR) として処理します。
- C を設定した場合
文字データ (VARCHAR) として処理します。

注意事項

ここでの指定と HiRDB で使用されているディクショナリ表のデータ属性が異なる場合は、結果不正や SQL エラーになる場合があります。

(8) DABSVLOGSIZE (DABroker のログ取得ファイルサイズ)

標準値

64 (キロバイト)

指定範囲

0, 又は 64~32767 (キロバイト)

説明

DABroker の処理のログを取得するファイルの容量を設定します。ログを [DABroker 運用ディレクトリ/spool/dabsv.log], および [DABroker 運用ディレクトリ/spool/dabsvdemon.log] に取得します。

- 0 を設定した場合
ファイルの容量は、無限大となります。
- 64~32767 を設定した場合
ファイルの容量は、設定した数値分となります。

注意事項

ログファイルが満杯になったらラップアラウンドでログを取得します。ラップアラウンドで取得した場合は、1 世代前のバックアップファイル (dabsv.logold, dabsvdemon.logold) を作成し、常に 1 世代だけを管理します。

(9) DABXA_LOG_ERROR (XA トレースを取得する)

標準値

ON (チェックあり)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)

(ON: 取得します OFF: 取得しません)

説明

XA インタフェースでエラーを検知した場合、エラートレースを取得するかどうかを設定します。

XA トレースは、DABroker 運用ディレクトリ下の spool ディレクトリに db_xainfo.log というファイル名で取得されます。XA トレースファイル(db_xainfo.log)はテキスト形式なので、任意のテキストエディタで参照してください。

XA トレースファイルは、運用期間に応じてファイルサイズが増加するため、定期的に削除する必要があります。

(10) DABEXSQL_TRC_LINE (拡張データベースアクセストレースを取得する)

標準値

-1 (トレースを取得しません)

指定範囲

-1 (トレースを取得しません), 0 (トレース取得行数を無限大にします), 又は 1024~32767 (行)

説明

拡張データベースアクセストレースを取得するかどうかを設定します。

- -1 を設定した場合
拡張データベースアクセストレースを取得しません。
- 0 を設定した場合
拡張データベースアクセストレースを取得します。出力行数は、無限大となります。
- 1024~32767 を設定した場合
拡張データベースアクセストレースを取得します。出力行数は、設定した数値分となります。

拡張データベースアクセストレースのファイルは、1 行に 134 バイトまで出力されます。拡張データベースアクセストレースのファイルの出力行数は、ここでの設定値に従います。ただし、ヘッダは行数に含まれません。

拡張データベースアクセストレースの詳細については、「[6.4.3 拡張データベースアクセストレース機能](#)」を参照してください。

(11) DAB_HIRDB_DBINF_ESC (HiRDB 環境変数区切り文字)

標準値

@

指定範囲

@ ! \$ % & * + - / : ? ¥ ;

説明

HiRDB 環境変数区切り文字を指定します。

注意事項

特に問題がない場合、区切り文字には@を使用してください。

区切り文字は必ず半角 1 文字で指定してください。全角文字は指定できません。

DABroker for Java から HiRDB クライアント環境変数グループを指定してデータベースと接続する場合は、標準値「@」を指定してください。

3.5.2 リモートアクセス設定項目

DABroker の動作環境でのリモートアクセス設定項目について説明します。

標準値は DABroker 動作環境定義ファイルのサンプルファイルにあらかじめ設定されている値です。

リモートアクセス設定項目の一覧を表 3-5 に示します。

表 3-5 リモートアクセス設定項目の一覧

設定項目	説明	標準値	指定範囲
DABCLIENT_TIMEOUT	クライアントからの要求待ちタイムアウト	3600(秒)	0, 又は 1~65535 (秒)
DABDBA_TIMEOUT	データベースからの実行終了待ちタイムアウト	1800(秒)	0, 又は 1~65535 (秒)
DABKEEPALIVE	TCP/IP 通信時キープアライブ	OFF	ON 又は OFF
DABTCP_TRCSIZE	通信トレースの取得	-1	0 又は -1
DABMAXUSERS	最大接続ユーザ数	10 (ユーザ)	1~100000 (ユーザ)
DABDBANUMBER	同時実行可能なデータベースアクセス数	20 (データベースアクセス数)	1~100000 (データベースアクセス数)
DABMAXBLOCKUSERS	1 プロセスあたりの実行ユーザ数	1 (ユーザ数)	1~100 (ユーザ数)

設定項目	説明	標準値	指定範囲
DABMAXPROCESS	最大処理プロセス数	10 (プロセス数)	1~1000 (プロセス数)
DABBATCH	予約処理作業ディレクトリ	/tmp	任意ディレクトリ ※
DABORADBA_ LONGBUFSIZE	ORACLE の LONG 型及び LONGRAW 型のデータの受け取りバッファサイズ	4096 (バイト)	0, 1~2147483647 (バイト)
DABHIRDBA_ BLOBBUFSIZE	HiRDB の BLOB 型データ受取バッファサイズ	4096 (バイト)	0, 1~2147483647 (バイト)
DABDBSDBA_ BLOBBUFSIZE	Database Connection Server の BLOB 型データ受取バッファサイズ	4096 (バイト)	0, 1~2147483647 (バイト)
DABIPCONVERT	IP アドレスからホスト名への変換オプション	ON	ON 又は OFF

注※

DABroker の管理ユーザに READ/WRITE 権限がある任意のディレクトリです。ただし、パーティションを圧迫しないところです。

(1) DABCLIENT_TIMEOUT (クライアントからの要求待ちタイムアウト)

標準値

3600 (秒)

指定範囲

0, 又は 1~65535 (秒)

説明

- 1~65535 (秒) を設定した場合

DABroker がアクセス要求に対する応答をクライアントに返してから、次のアクセス要求を受けるまでの時間を監視します。

ここで設定した時間を超えてクライアントからのアクセス要求がない場合、DABroker は次の処理をします。

クライアントに対しては、クライアントからの要求待ちタイムアウトの発生を通知して、接続を解放します。処理を続ける場合、クライアントは一度終了して再起動する必要があります。

- 0 を設定した場合

クライアントからの要求待ち時間を監視しません。

(2) DABDBA_TIMEOUT (データベースからの実行終了待ちタイムアウト)

標準値

1800 (秒)

指定範囲

0, 又は 1~65535 (秒)

説明

- 1~65535 (秒) を設定した場合
DABroker がデータベースに処理を要求してから、実行が終了するまでの時間を監視します。
ここで設定した時間を超えてデータベースからの応答がない場合、DABroker は次の処理をします。
クライアントに対しては、データベースからの実行終了待ちタイムアウトの発生を通知して、コネクションを解放します。処理を続ける場合、クライアントは一度終了して再起動する必要があります。
- 0 を設定した場合
データベースからの実行終了待ち時間を監視しません。

この項目は、Database Connection Server 使用時だけ有効です。

分散トランザクション連携機能使用時は、ここでの指定内容にかかわらず、0 が設定されます。

(3) DABKEEPALIVE (TCP/IP 通信時キープアライブ)

標準値

OFF

指定範囲

ON 又は OFF

説明

キープアライブによる監視を行うかどうかを設定します。キープアライブとは、クライアントとの通信経路が確立されているかどうかを常に監視する TCP/IP の機能です。

- ON を設定した場合
キープアライブによる監視が行われます (クライアントとの通信経路を監視する)。
- OFF を設定した場合
キープアライブによる監視は行われません (クライアントとの通信経路を監視しない)。

(4) DABTCP_TRCSIZE (通信トレースの取得)

標準値

-1

指定範囲

0 又は -1

説明

クライアントとの通信トレース (dabcltrc コマンドで表示) を取得するかどうかを設定します。コマンドについては、「[3.8.2 クライアントとの通信トレースの表示](#)」を参照してください。

- 0 を設定した場合

すべての通信トレースを蓄積するファイルが [DABroker 運用ディレクトリ /spool/dabcltrc] の下に作成されます。ファイル名は、次のとおりです (XXXXXX は接続元のクライアントの IP アドレス, AAAAAA は任意の名称)。

trs2340_tcXXX.XXX.XXX.XXX (DBPARTNER/Client を使用する場合)

trs2340_tcXXX.XXX.XXX.XXXAAAAA (DBPARTNER/Client 以外の場合)

ただし、ハードディスクの空き容量がなくなった場合、それ以降の通信トレースは取得しません。また、クライアントからの要求量が多ければ、トレースファイル容量も多くなるため、ハードディスクの容量に余裕をもたせておいてください。必要容量の概算方法は、「[3.8.2 クライアントとの通信トレースの表示](#)」を参照してください。また、クライアントからの要求量が多ければ、DABroker を終了するまでに時間がかかります。御注意ください。

- -1 を設定した場合

クライアントとの通信トレースを取得しません。

(5) DABMAXUSERS (最大接続ユーザ数)

標準値

10 (ユーザ)

指定範囲

1~100000 (ユーザ)

説明

コネクションを確立できる最大ユーザ数を設定します。ここで設定した値を超えるユーザが接続しようとした場合は、クライアントにエラーメッセージを返し、接続しません。

(6) DABDBANUMBER (同時実行可能なデータベースアクセス数)

標準値

20 (データベースアクセス数)

指定範囲

1~100000 (データベースアクセス数)

説明

DABroker が一度に受け付けられる、複数のクライアントからのデータベースアクセス処理要求の合計数を設定します。ここで設定した値を超える要求が行われた場合は、クライアントにエラーメッセージを返し、処理を受け付けません。

(7) DABMAXBLOCKUSERS (1 プロセスあたりの実行ユーザ数)

標準値

1 (ユーザ数)

指定範囲

1~100 (ユーザ数)

ORACLE を使用する場合は 1 を指定してください。

説明

一つのプロセスで処理する最大ユーザ数を設定します。ここで設定した値を超えた場合は、処理を受け付けません。

ORACLE を使用する場合は、必ず 1 を指定してください。1 以外を指定した場合は、クライアントにエラーメッセージを返し、処理を受け付けません。

注意事項

DABroker のデーモンプロセスが、サーバのメモリ又はキャッシュメモリ上で動作するように適切な値を設定してください。

(8) DABMAXPROCESS (最大処理プロセス数)

標準値

10 (プロセス数)

指定範囲

1~1000 (プロセス数)

説明

DABroker で起動する最大プロセス数を設定します。ここで設定した値を超えた場合は、クライアントにエラーメッセージを返し、処理を受け付けません。

注意事項

不要なデーモンプロセスの起動を制限するために適切な値を設定してください。

(9) DABBATCH (予約処理作業ディレクトリ)

標準値

/tmp

指定範囲

DABroker の管理ユーザに READ/WRITE 権限がある任意のディレクトリ (ただし、パーティションを圧迫しないところ)

説明

予約処理の終了ステータスを格納するディレクトリを指定します。標準値は/tmp ですが、クライアントが予約完了状況を次の日に確認する場合など、サーバマシンの終了後もファイルを保存する必要がある

る場合は、/tmp 以外のディレクトリを指定してください。指定したディレクトリの下には、次のようなファイルが作成されます。

- 予約処理用ステータスファイル

DABroker が内部的に使用します。ファイル名は次のとおりです。

\$DABBATCH/.yoyaku/.HHMSSnnnnnxx

このファイルは、クライアントからの要求で削除されます。

HHMSS：予約処理を受け付けた時刻（時 分 秒）

nnnnn：予約処理のプロセス ID

xx：HHMSSnnnnn が重複した場合の識別に使用する文字

<注意>

この予約処理用ステータスファイルを開いて内容を変更したり、ファイルを削除したりしないでください。予約処理が正常に実行できなくなります。

- 予約処理用システムファイル

DABroker が内部的に使用します。ファイル名は次のとおりです。

\$DABBATCH/.yoyaku/.system

このファイルはクライアントからの要求では削除されません。

<注意>

この予約処理用システムファイルを開いて内容を変更したり、ファイルを削除したりしないでください。予約処理が正常に実行できなくなります。

(10) DABORADBA_LONGBUFSIZE (ORACLE の LONG 型及び LONGRAW 型のデータの受け取りバッファサイズ)

標準値

4096 (バイト)

指定範囲

0, 又は 1~2147483647 (バイト)

説明

ORACLE から、LONG 型又は LONGROW 型を持つデータを抽出する場合の受け取りサイズを指定します。

これらのデータ型の場合、ORACLE では最大 2147483647 バイトのデータの存在が考えられますが、DABroker では、ここで指定した値までのデータを受け取り、以降のデータは切り捨てます。

0 を指定した場合、DBMS の定義長分の領域長を仮定します。ただし、実際に格納されているデータの大きさにかかわらず、無条件に定義長分の領域が確保されるため、メモリ効率が悪くなることがあります。格納されているデータの最大長が分かっている場合は、その値を指定することをお勧めします。この項目は、DABroker for Java Version 2 で DABroker にローカルアクセスした場合にも有効になります。

また、DABroker for Java Version 2 でデータベース種別に Oracle8i を設定している場合にも設定しません。

(11) DABHIRDBA_BLOBBUFSIZE (HiRDB の BLOB 型データ受取バッファサイズ)

標準値

4096 (バイト)

指定範囲

0, 又は 1~2147483647 (バイト)

説明

HiRDB から BLOB 型データを抽出する場合の受け取りサイズを指定します。

このデータ型の場合、HiRDB では最大 2147483647 バイトのデータの存在が考えられますが、DABroker ではここで指定した値までのデータを受け取り、以降のデータは切り捨てます。

0 を指定した場合、DBMS の定義長分の領域長を仮定します。ただし、実際に格納されているデータの大きさにかかわらず、無条件に定義長分の領域が確保されるため、メモリ効率が悪くなることがあります。格納されているデータの最大長が分かっている場合は、その値を指定することをお勧めします。この項目は、DABroker for Java Version 2 で DABroker にローカルアクセスした場合にも有効になります。

(12) DABDBSDBA_BLOBBUFSIZE (Database Connection Server の BLOB 型データ受取バッファサイズ)

標準値

4096 (バイト)

指定範囲

0, 又は 1~2147483647 (バイト)

説明

Database Connection Server から BLOB 型データを抽出する場合の受け取りサイズを指定します。

このデータ型の場合、最大 2147483647 バイトのデータの存在が考えられますが、DABroker ではここで指定した値までのデータを受け取り、以降のデータは切り捨てます。

0 を指定した場合、DBMS の定義長分の領域長を仮定します。ただし、実際に格納されているデータの大きさにかかわらず、無条件に定義長分の領域が確保されるため、メモリ効率が悪くなることがあります。格納されているデータの最大長が分かっている場合は、その値を指定することをお勧めします。この項目は、DABroker for Java Version 2 で DABroker にローカルアクセスした場合にも有効になります。

(13) DABIPCONVERT (IP アドレスからホスト名への変換オプション)

標準値

ON

指定範囲

ON 又は OFF

説明

メッセージ、及び DABroker の処理内容を表示するコマンドで表示される IP アドレスをホスト名に変換するかどうかを設定します。DABroker の処理内容を表示するコマンドについては「[3.8 DABroker の処理内容の表示](#)」を参照してください。

- ON を設定した場合
表示される IP アドレスをホスト名に変換します。
動作環境によってこの変換処理に時間がかかり、接続処理が遅くなることがあります。
- OFF を設定した場合
表示される IP アドレスをホスト名に変換しません。

3.5.3 イベントトレースの設定

DABroker 関連製品を使用して開発したアプリケーションからアクセスする場合の、イベントトレースの設定項目について説明します。イベントトレースを取得するかどうか、取得するときのファイルサイズ、取得レベル、及び取得するイベントの項目を設定します。

なお、DABroker for Java を使用する場合は、このイベントトレースを取得できません。

イベントトレースの取得形式については、「[6.3.1 イベントトレースの取得形式 \(共通\)](#)」を参照してください。

(1) for C++設定項目

for C++設定項目の一覧を表 3-6 に示します。

表 3-6 for C++設定項目の一覧

設定項目	説明	標準値	指定範囲
DABCPP_EVTTRC_SIZE	DABroker for C++のイベントトレース取得サイズ	32 (キロバイト)	-1, 0, 10~32767 (キロバイト)
DABCPP_EVTTRC_LEVEL	DABroker for C++のイベントトレース取得レベル	0	0 又は 1

設定項目	説明	標準値	指定範囲
DABCPP_EVTTRC_nn	DABroker for C++のイベントトレース取得種別	マニュアル「DABroker for C++」を参照してください	ON 又は OFF (ON:取得します OFF:取得しません)

(a) DABCPP_EVTTRC_SIZE (DABroker for C++のイベントトレース取得サイズ)

標準値

32 (キロバイト)

指定範囲

-1, 0, 10~32767 (キロバイト)

説明

DABroker for C++のイベントトレースを取得するかどうか、及び取得する場合は、トレースを取得するファイルの容量を設定します。トレースを [DABroker 運用ディレクトリ/spool/dabevtrccpp1] に取得します。

- -1 を設定した場合
イベントトレースを取得しません。
- 0 を設定した場合
イベントトレースを取得します。ファイルの容量は、無限大となります。
- 10~32767 を設定した場合
イベントトレースを取得します。ファイルの容量は、設定した数値分となります。

注意事項

トレースファイルが満杯になったらラップアラウンドでトレースを取得します。ラップアラウンドで取得した場合は、1世代前のバックアップファイル (dabevtrccpp2) を作成し、常に1世代だけを管理します。

(b) DABCPP_EVTTRC_LEVEL (DABroker for C++のイベントトレース取得レベル)

標準値

0

指定範囲

0 又は 1

説明

DABroker for C++のイベントトレースの取得レベルを設定します。
設定値が大きいほど多くのイベントトレースを取得します。

(c) DABCPP_EVTTRC_nn [nn : 00 から始まるイベント番号] (DABroker for C++のイベントトレース取得種別)

標準値

マニュアル「DABroker for C++」を参照してください。

指定範囲

ON 又は OFF (ON : 取得します OFF : 取得しません)

説明

各イベントのトレースを取得するかどうかを設定します。取得できるイベントトレースの内容については、マニュアル「DABroker for C++」を参照してください。

(2) for COBOL 設定項目

for COBOL 設定項目の一覧を表 3-7 に示します。

表 3-7 for COBOL 設定項目の一覧

設定項目	説明	標準値	指定範囲
DABCOBOL_EVTTRC_SIZE	DABroker for COBOL のイベントトレース取得サイズ	32 (キロバイト)	-1, 0, 10~32767 (キロバイト)
DABCOBOL_EVTTRC_LEVEL	DABroker for COBOL のイベントトレース取得レベル	0	0~2
DABCOBOL_EVTTRC_nn	DABroker for COBOL のイベントトレース取得種別	マニュアル「DABroker for COBOL」を参照してください	ON 又は OFF (ON : 取得します OFF : 取得しません)

(a) DABCOBOL_EVTTRC_SIZE (DABroker for COBOL のイベントトレース取得サイズ)

標準値

32 (キロバイト)

指定範囲

-1, 0, 10~32767 (キロバイト)

説明

DABroker for COBOL のイベントトレースを取得するかどうか、及び取得する場合は、トレースを取得するファイルの容量を設定します。トレースを [DABroker 運用ディレクトリ/spool/dabevtttrccobol1] に取得します。

- -1 を設定した場合
イベントトレースを取得しません。

- 0 を設定した場合
イベントトレースを取得します。ファイルの容量は、無限大となります。
- 10~32767 を設定した場合
イベントトレースを取得します。ファイルの容量は、設定した数値分となります。

<注意>

トレースファイルが満杯になったらラップアラウンドでトレースを取得します。ラップアラウンドで取得した場合は、1世代前のバックアップファイル (dabevttrccobol2) を作成し、常に1世代だけを管理します。

(b) DABCOBOL_EVTTRC_LEVEL (DABroker for COBOL のイベントトレース取得レベル)

標準値

0

指定範囲

0~2

説明

DABroker for COBOL のイベントトレースの取得レベルを設定します。
設定値が大きいほど多くのイベントトレースを取得します。

(c) DABCOBOL_EVTTRC_nn [nn : 00 から始まるイベント番号] (DABroker for COBOL のイベントトレース取得種別)

標準値

マニュアル「DABroker for COBOL」を参照してください。

指定範囲

ON 又は OFF (ON : 取得します OFF : 取得しません)

説明

各イベントのトレースを取得するかどうかを設定します。取得できるイベントトレースの内容については、マニュアル「DABroker for COBOL」を参照してください。

(3) for ORB 設定項目

for ORB 設定項目の一覧を表 3-8 に示します。

表 3-8 for ORB 設定項目の一覧

設定項目	説明	標準値	指定範囲
DABORB_EVTTRC_SIZE	DABroker for ORB のイベントトレース取得サイズ	128 (キロバイト)	-1, 0, 10~32767 (キロバイト)

設定項目	説明	標準値	指定範囲
DABORB_EVTTRC_LEVEL	DABroker for ORB のイベントトレース取得レベル	1	0 又は 1
DABORB_EVTTRC_nn	DABroker for ORB のイベントトレース取得種別	マニュアル「DABroker for ORB」を参照してください	ON 又は OFF (ON: 取得します OFF: 取得しません)

(a) DABORB_EVTTRC_SIZE (DABroker for ORB のイベントトレース取得サイズ)

標準値

128 (キロバイト)

指定範囲

-1, 0, 10~32767 (キロバイト)

説明

DABroker for ORB のイベントトレースを取得するかどうか、及び取得する場合は、トレースを取得するファイルの容量を設定します。トレースを [DABroker 運用ディレクトリ/spool/dabevttrcorb1] に取得します。

- -1 を設定した場合
イベントトレースを取得しません。
- 0 を設定した場合
イベントトレースを取得します。ファイルの容量は、無限大となります。
- 10~32767 を設定した場合
イベントトレースを取得します。ファイルの容量は、設定した数値分となります。

<注意>

トレースファイルが満杯になったらラップアラウンドでトレースを取得します。ラップアラウンドで取得した場合は、1 世代前のバックアップファイル (dabevttrcorb2) を作成し、常に 1 世代だけを管理します。

(b) DABORB_EVTTRC_LEVEL (DABroker for ORB のイベントトレース取得レベル)

標準値

1

指定範囲

0 又は 1

説明

DABroker for ORB のイベントトレースの取得レベルを設定します。
設定値が大きいほど多くのイベントトレースを取得します。

(c) DABORB_EVTTRC_nn [nn : 00 から始まるイベント番号] (DABroker for ORB のイベントトレース取得種別)

標準値

マニュアル「DABroker for ORB」を参照してください。

指定範囲

ON 又は OFF (ON : 取得します OFF : 取得しません)

説明

各イベントのトレースを取得するかどうかを設定します。取得できるイベントトレースの内容については、マニュアル「DABroker for ORB」を参照してください。

3.5.4 HiRDB の設定

HiRDB を使用する場合の設定項目を次に示します。HiRDB のバージョンによっては、必要な設定項目が異なることがありますのでご注意ください。

PDHOST と PDNAMEPORT の値は、HiRDB の環境で設定されている値を指定してください。そのほかの設定項目では、DABroker で設定した値が有効になります。各設定項目については、マニュアル「HiRDB Version x UAP 開発ガイド」を参照してください。

1. PDCLTPATH

トレースファイル格納ディレクトリ

2. PDHOST

ホスト名

注

ここで設定した値と異なるホスト名でアプリケーションが接続を要求した場合は、アプリケーションで指定されたホスト名が有効になります。

3. PDNAMEPORT

HiRDB のシステムポート番号

注

ここで設定した値と異なるポート番号でアプリケーションが接続を要求した場合は、アプリケーションで指定されたポート番号が有効になります。

4. PDSWAITTIME

サーバ最大待ち時間

5. PDSQLTRACE

SQL トレースファイルのサイズ

6. PDUAPERLOG

3.5.5 ORACLE の設定

ORACLE を使用する場合の設定項目を次に示します。

各設定項目には、ORACLE の環境で設定されている値を指定してください。各設定項目については、ORACLE のマニュアルを参照してください。

1. ORACLE_HOME

ORACLE のホームディレクトリ

2. ORACLE_SID

ORACLE のデータベース識別子

<注意>

ここで設定した値と異なるデータベース識別子でアプリケーションが接続を要求した場合は、アプリケーションで指定されたデータベース識別子が有効になります。

3. NLS_LANG

母国語指定

3.5.6 Database Connection Server 接続設定項目

DABroker の動作環境での Database Connection Server 接続設定項目について説明します。

標準値は DABroker 動作環境定義ファイルのサンプルファイルにあらかじめ設定されている値です。

Database Connection Server 接続設定項目の一覧を表 3-9 に示します。

表 3-9 Database Connection Server 接続設定項目の一覧

設定項目	説明	標準値	指定範囲
DAB_DBS_HOST	接続する Database Connection Server のホスト名, 又は IP アドレス	localhost	ホスト名, 又は IP アドレス
DAB_DBS_PORT	接続する Database Connection Server のサービス名, 又はポート番号	40179	サービス名, 又はポート番号
DAB_DBS_SEND_BUFFER_SIZE	Database Connection Server に送信するときの最大バッファサイズ	64 (キロバイト)	64~2097151 (キロバイト)

設定項目	説明	標準値	指定範囲
DAB_DBS_RECEIVE_BUFFER_SIZE	Database Connection Server から受信するときの最大バッファサイズ	64 (キロバイト)	64~2097151 (キロバイト)

(1) DAB_DBS_HOST (接続する Database Connection Server のホスト名, 又は IP アドレス)

標準値

localhost

指定範囲

ホスト名, 又は IP アドレス

説明

Database Connection Server 使用時, 接続先の Database Connection Server のホスト名, 又は IP アドレスを指定します。

(2) DAB_DBS_PORT (接続する Database Connection Server のサービス名, 又はポート番号)

標準値

40179

指定範囲

サービス名, 又はポート番号

説明

Database Connection Server 使用時, 接続先の Database Connection Server のサービス名, 又はポート番号を指定します。

(3) DAB_DBS_SEND_BUFFER_SIZE (Database Connection Server に送信するときの最大バッファサイズ)

標準値

64 (キロバイト)

指定範囲

64~2097151 (キロバイト)

説明

Database Connection Server 使用時, Database Connection Server にデータを送信する場合の最大バッファサイズを指定します。

(4) DAB_DBS_RECEIVE_BUFFER_SIZE (Database Connection Server に受信するときの最大バッファサイズ)

標準値

64 (キロバイト)

指定範囲

64~2097151 (キロバイト)

説明

Database Connection Server 使用時, Database Connection Server からデータを受信する場合の最大バッファサイズを指定します。

3.5.7 エージェント処理機能の設定

エージェント処理を使用する場合の設定項目について説明します。

エージェント処理機能の設定の一覧を表 3-10 に示します。

表 3-10 エージェント処理機能の設定の一覧

設定項目	説明	標準値	指定範囲
DABAGT	作業ディレクトリ	/opt/DABroker/AGT	DABroker の管理ユーザに READ/WRITE 権限がある任意のディレクトリ (ただし, パーティションを圧迫しないところ)
DABAGTREGMAX	登録可能エージェント数	1000 (エージェント数)	1~32767 (エージェント数)
DABAGTTIMMAX	実行予約可能最大エージェント数	250 (エージェント数)	1~32767 (エージェント数)
DABAGTFILMAX	実行結果格納ファイル容量	500 (キロバイト)	1~2097151 (キロバイト)
DABAGT_LOG_SIZE	エージェントログファイル容量	64 (キロバイト)	-1, 0, 又は 1~32767 (キロバイト)
DABAGT_LOG_INFO	エージェントログにインフォメーション情報を取得する	ON	ON 又は OFF (ON: 取得します OFF: 取得しません)
DABAGT_LOG_ERROR	エージェントログにエラー情報を取得する	ON	ON 又は OFF (ON: 取得します OFF: 取得しません)
DABAGTGRP	グループ管理用ディレクトリ	/opt/DABroker/AGTGRP	DABroker の管理ユーザに READ/WRITE 権限があ

設定項目	説明	標準値	指定範囲
			任意のディレクトリ（ただし、パーティションを圧迫しないところ）
DABAGTWCHALDAY	実行予約日を経過しても有効にする日数	0（日）	0～31（日）
DABAGTWCHALTIME	実行予約時刻を経過しても有効にする秒数	0（秒）	0～86400（秒）
DABAGTEXEDIRETIME	即時配信実行監視時間	1800（秒）	0, 又は 60～3600（秒）

(1) DABAGT（作業ディレクトリ）

標準値

/opt/DABroker/AGT

指定範囲

DABroker の管理ユーザに READ/WRITE 権限がある任意のディレクトリ（ただし、パーティションを圧迫しないところ）

説明

エージェント登録情報、実行情報、及び実行結果を格納するディレクトリを指定します。指定したディレクトリの下には、次のようなファイルが作成されます。

- エージェント登録情報用ファイル

DBPARTNER/Client で指定する、エージェント登録情報を格納します。エージェント登録要求時に作成されます。ファイル名は次のとおりです。

\$DABAGT/.REGaa・・・aa

aa・・・aa：登録したエージェント名（DBPARTNER/Client で指定した文字列が設定されます）
 なお、エージェント名は、一つの DABroker 内でユニークな名称にしてください。ユニークでない場合は、DBPARTNER/Client にエラーが返されます。

エージェント登録情報ファイルは、エージェント登録ごとにできるファイルです。エージェント登録数が多くなるにつれ、エージェント登録情報ファイル数も増え、サーバファイルシステムを圧迫します。このため、ルートディレクトリのパーティションとは別に、エージェント用ディレクトリのパーティションを確保するなどの運用をシステム管理者が行ってください。サーバファイルシステムの圧迫が解消しない場合は、使用しないファイルの退避又は削除などの対処をしてください。ファイル運用については「5.4 エージェント機能で使用するファイルの運用」、エージェント登録情報ファイルの容量の目安となる概算式については「5.4.5(2) ファイル容量の見積もり」を参照してください。

- エージェント登録情報管理ファイル

エージェント登録情報を管理し、DABroker が内部的に使用します。ファイル名は次のとおりです。
 \$DABAGT/.RCTRL

- エージェント実行情報ファイル

クライアントからのエージェント実行要求を受け付けたときに、実行情報を格納します。エージェント実行要求時に作成します。ファイル名は次のとおりです。

`$DABAGT/.EXEbb・・・bb`

bb・・・bb：エージェント実行名（DABroker が割り当てる識別名（番号）です）

エージェント実行情報ファイルは、エージェント実行要求受け付けごとにできるファイルです。エージェント実行要求数が多くなるにつれ、エージェント実行情報ファイル数も増え、サーバファイルシステムを圧迫します。このため、ルートディレクトリのパーティションとは別に、エージェント用ディレクトリのパーティションを確保するなどの運用をシステム管理者が行ってください。サーバファイルシステムの圧迫が解消しない場合は、使用しないファイルの退避又は削除などの対処をしてください。

エージェント実行情報ファイルの削除は、エージェント実行終了状態で操作してください。また、エージェント実行情報ファイルと対応するエージェントの実行結果格納ファイルも削除してください。ファイルの運用については「[5.4 エージェント機能で使用するファイルの運用](#)」，エージェント実行情報ファイルの容量の目安となる概算式については「[5.4.5\(2\) ファイル容量の見積もり](#)」を参照してください。

- エージェント実行情報管理ファイル

エージェント実行情報を管理し、DABroker が内部的に使用します。ファイル名は次のとおりです。

`$DABAGT/.ECTRL`

- エージェント実行結果格納ファイル

エージェントの実行結果を格納します。エージェント実行時に作成されます。ファイル名は次のとおりです。

`$DABAGT/.RESbb・・・bb`

bb・・・bb：エージェント実行名（DABroker が割り当てる識別名（番号）です）

エージェント実行結果格納ファイルを削除するときは、ファイル名、及び `dabagtexeinfo` コマンド（「[3.8.4 エージェント実行情報の表示](#)」参照）でエージェント実行情報を確認してから実行してください。

エージェント実行情報ファイルを削除する場合は、削除するエージェント実行情報に対応するエージェントの実行結果も削除してください。

ファイルの運用については、「[5.4 エージェント機能で使用するファイルの運用](#)」を参照してください。

- 実行予約指定エージェントの実行待ちキューファイル

DABroker の起動時に作成され、エージェント情報に実行予約を設定した（エージェントタイマー指定した）エージェントの、キュー情報について格納します。ファイル名は次のとおりです。

`$DABAGT/.timer_que`

- エージェントログファイル

エージェント機能を使う処理のログを取得します。エージェント機能を使う処理ごとに取得します。なお、エージェントログの取得形式、及び取得内容については、「[6.3.2 エージェントログの取得形式と取得内容（共通）](#)」を参照してください。ファイル名は次のとおりです。

\$DABAGT/agtlogfile

(バックアップファイル：\$DABAGT/agtlogfile.old)

エージェントログファイルが満杯になったらラップアラウンドで取得します。ラップアラウンドで取得した場合は、1世代だけを世代管理し、バックアップファイル (agtlogfile.old) を作成します。

(2) DABAGTREGMAX (登録可能エージェント数)

標準値

1000 (エージェント数)

指定範囲

1~32767 (エージェント数)

説明

登録可能な最大エージェント数を設定します。ここで設定した値を超える要求が行われた場合は、クライアントにエラーメッセージを返し、処理を受け付けません。

(3) DABAGTTIMMAX (実行予約可能最大エージェント数)

標準値

250 (エージェント数)

指定範囲

1~32767 (エージェント数)

説明

実行予約を設定して (エージェントタイマー指定して) 実行する、最大エージェント数を設定します。ここで設定した値を超える要求が行われた場合は、クライアントにエラーメッセージを返し、処理を受け付けません。既に登録済みの実行予約指定エージェントの最大エージェント数より少ない値を指定した場合は、設定値を変更しないで登録済みの値が設定されます。

(4) DABAGTFILMAX (実行結果格納ファイル容量)

標準値

500 (キロバイト)

指定範囲

1~2097151 (キロバイト)

説明

エージェントの実行結果格納ファイルの最大容量を設定します。ここで設定した値を超える要求が行われた場合は、クライアントにエラーメッセージを返し、設定サイズに入る限りのデータを保存して処理を終了します。

(5) DABAGT_LOG_SIZE (エージェントログファイル容量)

標準値

64 (キロバイト)

指定範囲

-1, 0, 又は 1~32767 (キロバイト)

説明

エージェントログを取得するかどうか、及び取得する場合は、エージェントログを取得するファイルの容量を設定します。

- -1 を設定した場合
エージェントログを取得しません。
- 0 を設定した場合
エージェントログを取得します。ファイルの容量は、無限大になります。
- 1~32767 の値を設定した場合
エージェントログを取得します。ファイルの容量は、設定した数値分となります。
ログを取得する設定をした場合、エージェントログファイルが満杯になったらラップアラウンドでログを取得します。ラップアラウンドで取得した場合は、1 世代前のバックアップファイル (agtlogfile.old) を作成し、常に 1 世代だけを世代管理します。

(6) エージェントログ選択オプション

エージェントログにはインフォメーション情報とエラー情報があり、それぞれを取得するかどうかを設定できます。エージェントログの取得内容を任意に選択して取得する機能をエージェントログ選択オプションといいます。エージェントログの取得内容については、「[6.3.2 エージェントログの取得形式と取得内容 \(共通\)](#)」を参照してください。

(a) DABAGT_LOG_INFO (エージェントログにインフォメーション情報を取得する)

標準値

ON

指定範囲

ON 又は OFF (ON : 取得します OFF : 取得しません)

説明

エージェントログにインフォメーション情報を取得するかどうかを設定します。

(b) DABAGT_LOG_ERROR (エージェントログにエラー情報を取得する)

標準値

ON

指定範囲

ON 又は OFF (ON : 取得します OFF : 取得しません)

説明

エージェントログに、エラー情報のログを取得するかどうかを設定します。

(7) DABAGTGRP (グループ管理用ディレクトリ)

標準値

/opt/DABroker/AGTGRP

指定範囲

DABroker の管理ユーザに READ/WRITE 権限がある任意のディレクトリ (ただし、パーティションを圧迫しないところ)

説明

複数ユーザをグループ化して管理するファイル (グループ管理情報ファイル) を格納するディレクトリを指定します。

複数ユーザをグループ化することで、各処理要求時にグループ外ユーザの要求を拒否できます。

ここで指定したディレクトリ下に、次のファイルを格納します。

- エージェントのグループ管理情報ファイル

エージェントの処理を要求するユーザ ID をグループ管理するファイルです。グループ ID ごとに作成します。ファイル名は次のとおりです。

\$DABAGTGRP/gg・・・gg

gg・・・gg : グループ ID

なお、性能劣化の原因となるため、このディレクトリには、グループ情報以外の不要なファイルを格納しないでください。

グループ管理情報ファイルの作成手順については、「[5.1.3 エージェント機能を使用するユーザのグループ管理](#)」を参照してください。

(8) DABAGTWCHALDAY (実行予約日を経過しても有効にする日数)

標準値

0 (日)

指定範囲

0~31 (日)

説明

エージェントの実行を予約した日が過ぎても実行予約処理を有効にする最大日数を指定します。実行予約日を有効にする日数を指定することで、サーバマシンとクライアントでユーザが認識する時間（日数）のずれを調整することができます。

(9) DABAGTWCHALTIME (実行予約時刻を経過しても有効にする秒数)

標準値

0 (秒)

指定範囲

0~86400 (秒)

説明

エージェントの実行を予約した時刻が過ぎても実行予約処理を有効にする最大時間を指定します。実行予約時刻を経過しても有効にする時間を指定することで、サーバマシンとクライアントでユーザ認識する時間のずれを調整することができます。

(10) DABAGTEXEDIRETIME (即時配信実行監視時間)

標準値

1800 (秒)

指定範囲

0, 又は 60~3600 (秒)

説明

DABroker がエージェント即時配信実行でのエージェント処理終了を待つ時間を設定します。

- 0 を設定した場合
実行時間の監視をしないで、エージェント処理の終了を待ちます。
- 60~3600 を設定した場合
ここで設定した時間を超えてもエージェント処理が終了しない場合、DABroker は次の処理をします。
クライアントに対しては、エージェント即時配信実行監視タイムアウトの発生を通知します。
データベースに対しては、エージェントの実行をキャンセルします。データベースでの処理は、DBMS でロールバックされます。

3.5.8 その他の設定

(1) DAB_DEMON_LOG_MODE (デーモンの生成・消滅・クリーンアップ実行結果ログ抑止)

標準値

なし

指定範囲

LIMITED

説明

デーモンの生成・消滅・クリーンアップ実行結果ログの出力によりディスクを圧迫したくない場合に指定します。

デーモンの生成・消滅・クリーンアップ実行結果ログは、本指定をしない限り、標準で出力されます。

デーモンの生成・消滅・クリーンアップ実行結果ログについては、「[6.3.4 デーモンの生成・消滅・クリーンアップ実行結果ログ \(共通\)](#)」を参照してください。

(2) DABPTHREAD_STACK_SIZE (スレッドスタックサイズ変更機能)

標準値

システムデフォルトのスレッドスタックサイズ

指定範囲

0~1024 (キロバイト)

説明

スレッドスタックサイズを変更する場合に指定します。

指定された数値がシステムの最小値より下回っている場合には、システムの最小値まで切り上げ、指定された値が 1024 を超えたときは 1024 になります。システムの最小値はプラットフォームによって異なります。

なお、本機能はプラットフォームが HP-UX の場合のみ有効となります。

(3) DAB_ORA7_XAHANDLE_OCI8USE (Oracle XA 接続時の Pro*COBOL, Pro*C/C++連携機能)

標準値

なし

指定範囲

ON (使用する)

説明

XA 環境下の同一プロセス内で、Oracle Pro*COBOL、又は Oracle Pro*C/C++で作成した UAP と、DABroker for C++ で作成した UAP から Oracle に接続をする場合に設定します。この設定をせず、XA 環境下の同一プロセス内で、Oracle Pro*COBOL、又は Oracle Pro*C/C++で作成した UAP と、DABroker for C++ で作成した UAP から Oracle に接続すると「ORA-03114: not connected to ORACLE」エラーになります。

Oracle9i 9.0.1 以前のバージョンのデータベースに接続する場合は、” ON” を指定しても本機能は無効となります。

本機能を有効にすると、DABroker for C++ を使用した UAP から Oracle に接続、切断する場合に、次のように処理が変更されます。なお、以下「Oracle Call Interface」を OCI と呼びます。

(a) 本機能が無効の場合

(i)接続時

OCI7 の次の関数を実行する。

sqlld2()…XA 環境で、OCI7 で使用する接続情報を取得する

(ii)切断時

OCI 関数の呼び出し処理なし。

(b) 本機能が有効の場合

(i)接続時

OCI8 の以下の関数を実行する。

xaoEnv()…指定された XA 接続の OCI 環境ハンドルを取得する

xaoSvcCtx()…指定された XA 接続の OCI サービス・ハンドルを取得する

OCIHandleAlloc()…エラー・レポート・ハンドルを取得する

OCISvcCtxToLda()…OCI8 で生成した接続情報を OCI7 で使用する接続情報に変換する

(ii)切断時

OCI8 の以下の関数を実行する。

OCILdaToSvcCtx()…OCI7 で生成した接続情報を OCI8 で使用する接続情報に変換する

OCIHandleFree()…エラー・レポート・ハンドルを開放する

注意事項

本機能を使用して、XA 環境下の同一プロセス内で、DABroker for C++ で作成した UAP から Oracle に接続をした後に、DABroker for C++ 以外のインタフェース (Pro*COBOL, Pro*C/C++, 又は OCI 関数) で作成した UAP から Oracle に接続をする場合は、DABroker for C++ を使用した UAP で Oracle から切断をした後に行ってください。

DABroker では、OCISvcCtxToLda 関数を使用して動作しています。そのため、DABroker for C++ を使用した UAP で Oracle に接続している間に、DABroker for C++ 以外のインタフェースから

Oracle に接続した場合、DABroker で管理している接続情報が変更されることがあるため、動作を保証できません。

3.6 DABroker 接続先データベースの定義

接続先データベース定義ファイルは、アプリケーションから接続するデータベース種別、ホスト名、ポート番号などを定義するファイルです。DABroker for C++、DABroker for COBOL、及び DABroker for ORB で開発するアプリケーションは、接続先データベース定義ファイルに定義されたデータベース種別名、及びデータベース名を使用してデータベースにアクセスします。また、DBPARTNER、及び DABroker for Java でも、Database Connection Server 経由でデータベースにアクセスする場合は、接続先データベース定義ファイルで設定した情報を使用して Database Connection Server に接続します。

アプリケーションでは、データベース種別名及びデータベース名を指定するため、アプリケーションを変更することなく、接続するデータベースの変更ができます。ただし、OTS 機能、又は Java JTS 機能を使用する場合は、接続先データベース定義ファイルで設定した情報を使用できません。

ここでは、接続先データベース定義ファイルの作成方法について説明します。

3.6.1 接続先データベース定義ファイルの作成

接続先データベース定義ファイルは、「3.6.1(2) ファイルでの指定形式」で示す形式で作成します。また、インストール時に提供されるサンプルを使用して作成することもできます。サンプルファイルを使う作成方法については、「3.6.1(4) サンプルファイルを使用した作成方法」を参照してください。

(1) 作成するファイルの名称及び格納先

次に、接続先データベース定義ファイルのファイル名及び格納先を示します。

作成するファイルの名称

dadbenv

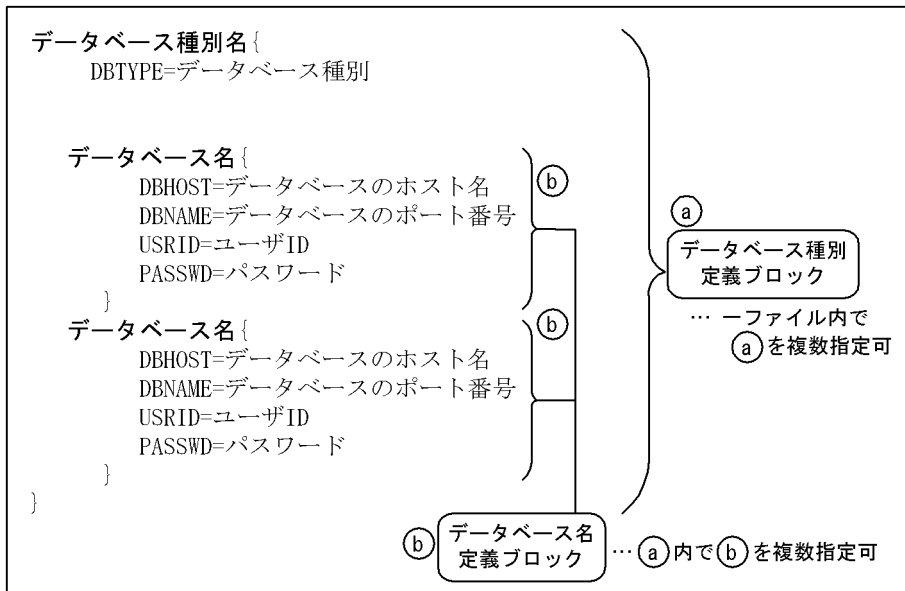
ファイルの格納先

DABroker 運用ディレクトリ/conf

(2) ファイルでの指定形式

接続先データベース定義ファイルでは、データベースに接続するための情報を次に示すような形式で指定します。データベース種別ごとにデータベース名を指定し、データベース名には、ホスト名、ポート番号などの情報を対応させます。

なお、ファイル内で複数のデータベース種別の指定ができます。また、一つのデータベース種別に対して、複数のデータベース名を指定できます。



(3) ファイル作成時の注意事項

- ファイル内では、大文字、小文字を区別します。
- 「#」以降は、コメントとして扱います。
- 「=」の前後に空白文字、又はタブコードなどの不要な文字を入れないでください。入れた場合はエラーとなります。

(4) サンプルファイルを使用した作成方法

提供するサンプルファイル (/opt/DABroker/sample/dadbenv) を使用して作成することもできます。次に、作成手順を示します。

1. サンプルファイルを「DABroker 運用ディレクトリ/conf」にコピーする。
2. コピー先ファイルの設定項目の標準値を必要に応じて変更する。また、値が設定されていない項目には値を入力する。
3. ファイルを保存する。

3.6.2 接続先データベース定義ファイルでの設定項目

ファイルには、データベース種別定義ブロックがあり、データベース種別定義ブロックの中で、データベース名定義ブロックを指定します。

(1) データベース種別定義ブロック

データベース種別、及びデータベース名定義ブロックを設定します。

次に、データベース種別を設定する項目について説明します。データベース名定義ブロックについては、「3.6.2(1)(a) データベース名定義ブロック」を参照してください。

データベース種別名（指定必須）

データベース種別を指定するブロックの名称です。アプリケーションでは、データベースに接続時にこの名称を使用します。

データベース種別名には、ファイル内でユニークな名称を付けてください。23文字まで指定できます。2バイト文字、半角カタカナ文字は使用しないでください。

DBTYPE=データベース種別（指定必須）

データベース種別を指定します。

データベース種別

次の文字列を指定してください。

使用するデータベース	指定内容
HiRDB にアクセスする場合	HIRDB
Oracle にアクセスする場合	ORACLE7
RDA Link for Gateway 経由で メインフレーム系データベースにアクセスする場合	RDALINK*
Database Connection Server 経由で メインフレーム系データベースにアクセスする場合	DBCONNECTSVR

注※

AIX 及び Linux では指定できません。

(a) データベース名定義ブロック

このブロックで指定する項目は、データベース種別によって異なります。

RDA Link for Gateway 経由でデータベースにアクセスする場合は、ここでデータベース種別を指定します。

次に、データベース名定義ブロックで設定できる項目について説明します。

データベース名

接続先データベースごとに指定する、データベースの接続情報（ホスト名、ポート番号など）を指定するブロックの名称です。アプリケーションでは、データベース接続時にこの名称を指定します。

データベース名には、データベース種別定義ブロック内でユニークな名称を付けてください。23文字まで指定できます。2バイト文字、半角カタカナ文字は使用しないでください。

◆HiRDB を使用する場合

HiRDB を使用する場合、DBHOST 及び DBNAME は必ず指定してください。

DBHOST=データベースのホスト名

接続する HiRDB が存在するホスト名称を指定します。指定を省略する場合は、HiRDB の環境変数 PDHOST で指定した値になります。

HiRDB クライアント環境変数使用時、及び HiRDB 環境変数グループ使用時は次のように設定してください。

HiRDB クライアント環境変数使用時

DBHOST=区切り文字+HiRDB クライアント環境変数=値+区切り文字+HiRDB クライアント環境変数=値

HiRDB クライアント環境定義の環境変数を指定するとき区切り文字として DABroker 環境変数 DAB_HIRDB_DBINF_ESC で指定した 1 バイトが区切り文字となります。

DAB_HIRDB_DBINF_ESC の指定がない場合は@が区切り文字となります。

@を区切り文字としたときの例を次に示します。

DBHOST=@PDHOST=host@PDNAMEPORT=20000

注意事項

- HiRDB で指定できる環境変数以外を指定した場合の動作は保証しません。
- 環境変数の指定値は DABroker ではエラーチェックを行わないため、環境変数に誤りがある場合は HiRDB のエラーとなります。
- この機能を使用する場合、従来 DABroker で設定していた、PDUSER、PDCLTAPNAME、PDHOST、PDNAMEPORT は設定されません。
- 指定値の中の空白 # " ' はサポートしていません。指定した場合の動作は保証しません。

HiRDB 環境変数グループ使用時

DBHOST=区切り文字+DABENVGRP=HiRDB 環境変数グループ

DABroker 環境変数 DAB_HIRDB_DBINF_ESC 環境変数で指定した 1 バイトが区切り文字となります。DAB_HIRDB_DBINF_ESC の指定がない場合は@が区切り文字となります。

@を区切り文字としたときの例を次に示します。

DBHOST=@DABENVGRP=XXXX

接続時に HiRDB 環境変数グループ識別子 XXXX を指定します。

注意事項

- HiRDB 環境変数グループの指定値は DABroker ではエラーチェックを行わないため、HiRDB 環境変数グループに誤りがある場合は HiRDB のエラーとなります。
- この機能を使用する場合、従来 DABroker で設定していた、PDUSER、PDCLTAPNAME、PDHOST、PDNAMEPORT は設定されません。
- 指定値の中の空白 # " ' はサポートしていません。指定した場合の動作は保証しません。
- DBHOST=@DABENVGRP=XXXX 以外を指定しないでください。

DBNAME=データベースのポート番号

接続する HiRDB が使用するポート番号を指定します。指定を省略する場合は、HiRDB の環境変数 PDNAMEPORT で指定した値になります。

DBHOST として HiRDB クライアント環境変数、又は HiRDB 環境変数グループ使用時の設定をした場合は、次のように設定してください。

DBNAME=DAB_HIRDBENV

USRID=ユーザ ID

データベースにアクセスするユーザ ID を指定します。

アプリケーションでユーザ ID の指定が省略された場合、ここで指定したユーザ ID でデータベースにアクセスします。

なお、指定できる文字列の規則については、使用するデータベースのマニュアルを参照してください。

PASSWD=パスワード

データベースにアクセスするユーザ ID のパスワードを指定します。

アプリケーションでパスワードの指定が省略された場合、ここで指定したパスワードでデータベースにアクセスします。

なお、指定できる文字列の規則については、使用するデータベースのマニュアルを参照してください。

◆ORACLE を使用する場合

DBNAME=ORACLE のリスナー名

SQL*NET 経由で接続する ORACLE のリスナー名称を指定します。

リモートアクセスをしない場合はここでの指定を省略してください。

USRID=ユーザ ID

データベースにアクセスするユーザ ID を指定します。

アプリケーションでユーザ ID の指定が省略された場合、ここで指定したユーザ ID でデータベースにアクセスします。

なお、指定できる文字列の規則については、使用するデータベースのマニュアルを参照してください。

PASSWD=パスワード

データベースにアクセスするユーザ ID のパスワードを指定します。

アプリケーションでパスワードの指定が省略された場合、ここで指定したパスワードでデータベースにアクセスします。

なお、指定できる文字列の規則については、使用するデータベースのマニュアルを参照してください。

◆RDA Link for Gateway 経由でメインフレーム系データベースを使用する場合

DBTYPE=データベース種別 (指定必須)

データベース種別を指定します。

データベース種別

次の文字列を指定してください。

使用するデータベース	指定内容
VOS3 XDM/RD にアクセスする場合	XDMRD
VOSK SQL/K にアクセスする場合	SQLK
VOS1 RDB1 E2 にアクセスする場合	RDB1

DBNAME=RDA Link for Gateway の分散定義ファイル名称

RDA Link for Gateway の分散定義ファイル名称を指定します。

USRID=ユーザ ID

データベースにアクセスするユーザ ID を指定します。

アプリケーションでユーザ ID の指定が省略された場合、ここで指定したユーザ ID でデータベースにアクセスします。

なお、指定できる文字列の規則については、使用するデータベースのマニュアルを参照してください。

PASSWD=パスワード

データベースにアクセスするユーザ ID のパスワードを指定します。

アプリケーションでパスワードの指定が省略された場合、ここで指定したパスワードでデータベースにアクセスします。

なお、指定できる文字列の規則については、使用するデータベースのマニュアルを参照してください。

◆Database Connection Server 経由でメインフレーム系データベースを使用する場合

DBTYPE=データベース種別 (指定必須)

データベース種別を指定します。

データベース種別

次の文字列を指定してください。

使用するデータベース	指定内容
VOS3 XDM/RD にアクセスする場合	XDMRD
VOS3 XDM/SD にアクセスする場合	XDMSD
VOSK SQL/K にアクセスする場合	SQLK
VOS1 PDM II E2 にアクセスする場合	VOS1PDM
VOS1 RDB1 E2 にアクセスする場合	RDB1
VOSK スプールファイルにアクセスする場合	VOSKSPPOOL

DBNAME=Database Connection Server のデータベース識別子

接続する Database Connection Server のデータベース識別子(RD ノード名称)を指定します。

DBHOST=コネクションマルチ名

USERID のユーザが使用する、コネクションマルチ定義に設定したコネクションマルチ名を指定します。

SNDBUF=送信するデータバッファのサイズ

Database Connection Server 側へ送信するデータバッファのサイズを指定します。
指定できる範囲は、64~2097151(キロバイト)です。

RCVBUF=受信するデータバッファのサイズ

Database Connection Server 側から受信するデータバッファのサイズを指定します。
指定できる範囲は、64~2097151(キロバイト)です。

USRID=ユーザ ID

データベースにアクセスするユーザ ID を指定します。

アプリケーションでユーザ ID の指定が省略された場合、ここで指定したユーザ ID でデータベースにアクセスします。

なお、指定できる文字列の規則については、使用するデータベースのマニュアルを参照してください。

PASSWD=パスワード

データベースにアクセスするユーザ ID のパスワードを指定します。

アプリケーションでパスワードの指定が省略された場合、ここで指定したパスワードでデータベースにアクセスします。

なお、指定できる文字列の規則については、使用するデータベースのマニュアルを参照してください。

3.6.3 接続先データベース定義ファイルの作成例

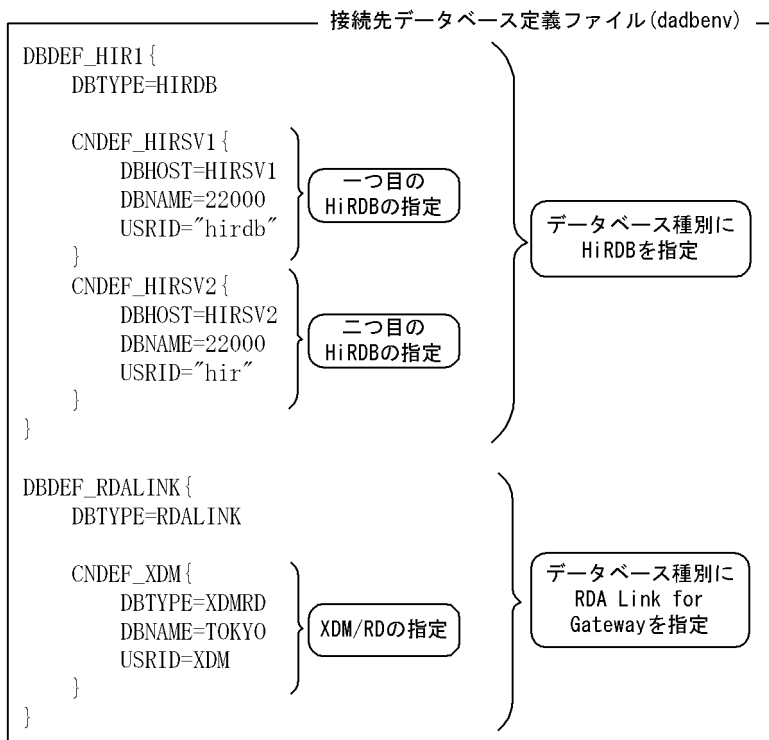
(1) HiRDB と XDM/RD の場合の指定例

HiRDB 及び VOS3 XDM/RD を使用する場合の指定例を次に示します。使用する HiRDB には 2 種類あり、次のような指定値を使用します。また、ここでは、DBTIMER、PASSWD の指定を省略しています。

(指定値)

- HiRDB のデータベース種別名：DBDEF_HIR1
 - 一つ目の HiRDB のデータベース名：CNDEF_HIRSV1
 - 二つ目の HiRDB のデータベース名：CNDEF_HIRSV2
- RDA Link for Gateway のデータベース種別名：DBDEF_RDALINK
 - VOS3 XDM/RD のデータベース名：CNDEF_XDM

(作成例)



(2) 同時に接続できない例

接続先データベース定義ファイル内で複数定義したデータベース種別名がユニークでも、実際のデータベース種別が同じ場合、同じプログラムから同時に接続できません。

次のような例では、HiRDB1 と HiRDB2 の両方を同時に接続できません。HiRDB1 と ORACLE1、又は HiRDB1 の Table1 と Table2 は同時に接続できます。

```
HiRDB1 {
    DBTYPE=HIRDB
    Table1 {
        DBHOST=' ITEMS'
        DBNAME=' 22200'
    }
    Table2 {
        DBHOST=' STOCK'
        DBNAME=' 22200'
    }
}
HiRDB2 {
    DBTYPE=HIRDB
    database2 {
        DBHOST=' EMPLOYEE'
        DBNAME=' 22200'
    }
}
ORACLE1 {
    DBTYPE=ORACLE7
    DB1 {
        DBHOST=' CUSTOMERS'
    }
}
}
```

3.7 DABroker の起動と終了

ここでは、DABroker の起動と終了について説明します。DABroker と同じマシンでサーバアプリケーションを実行するような場合、DABroker を起動する必要はありません。しかし、次のような場合は、DABroker を起動する必要があります。

- DBPARTNER, 又は HITSENSER5 を使用してデータベースにアクセスする場合
- DBPARTNER Library, 又は XDM/RD 外部表を使用してデータベースにアクセスする場合
- DABroker for Java を使用して WWW 環境でデータベースにアクセスする場合
- DABroker for ODBC を使用してリモート接続でデータベースにアクセスする場合

DABroker で提供するコマンドは、「DABroker 運用ディレクトリ/bin」に格納されています。

DABroker が実際に起動・終了されるのは、起動・終了コマンドの実行と同時ではありません。このため、起動・終了の各コマンドの実行が完了しても、しばらく DABroker の起動・終了は動作します。

DABroker の運用前の設定については、「[5 DABroker の運用 \(共通\)](#)」を参照してください。

3.7.1 DABroker の起動

DABroker を起動するには、起動コマンドを入力します。また、DABroker をインストールしたマシンの起動時に、自動的に DABroker を起動する方法もあります。

(1) 起動コマンド

コマンド形式

```
# dabstart [-t] [-a]
```

機能

DABroker を起動します。

使用できるユーザ

DABroker の管理ユーザ

オプション

-t

クライアントとの通信に TCP/IP を使用することを指定します。なお、-t を省略しても標準値として -t が設定されます。

-a

エージェント処理機能を使用することを指定します。なお、-a を省略すると、エージェント処理機能は使用できませんので御注意ください。

(2) 自動起動

自動起動とは、DABroker をインストールした端末の電源を入れた時に、自動的に DABroker を起動させることです。自動起動は、DBPARTNER、HITSENER5、DBPARTNER Library、又は XDM/RD 外部表からアクセスする場合だけ有効です。

自動起動の設定方法については、各 UNIX のマニュアルを参照してください。

エージェント処理を使用する場合は、dabstart コマンドに -a オプションを設定してください（「3.7.1(1) 起動コマンド」参照）。

この記述は、UNIX が実行する起動シェルスクリプトの中では、TCP/IP の起動よりも後に DABroker が起動するように追加してください。

注意事項

予約処理、又は実行予約を設定しているエージェントがある場合、DABroker を常に起動しておく必要があります。予約処理、又は実行予約を設定しているエージェントがある場合は、サーバを常時通電しておくか、自動起動設定にしておくことをお勧めします。

3.7.2 DABroker の終了

DABroker を終了するには、終了コマンドを入力します。

クライアント及び DBMS は、それぞれの環境で終了してください。

オプションの指定で、次のどれかの終了方式が選べます。

- 通常の終了

接続中のクライアントが存在せず、かつサーバで実行中の予約処理がない場合、及び実行予約を設定したエージェントがない場合にだけ、DABroker を終了します。それ以外の場合はエラーメッセージが表示され、終了できません。接続中のクライアント又は実行中の予約処理があるかどうかは、dabstat コマンドで確認できます。dabstat コマンドについては、「3.8.1 動作状態の表示」を参照してください。

- 計画終了

サーバで実行中の処理が終わり、すべてのクライアントとの接続が解放された時点で、DABroker を終了します。新しいクライアントからの接続要求、及び接続中のクライアントからの新しい処理要求は受け付けません。

- 強制終了

サーバで実行中の処理があってもすべて中断して、DABroker を終了します。クライアントとの接続は解放し、データベースに対しては処理のロールバック要求と切り離しをします。

このほかに、DABroker は終了しないで、サーバで実行中の特定のクライアントの処理だけを中断するオプションもあります。

(1) 終了コマンド

コマンド形式

```
# dabstop [-p|-f|-c IP アドレス又はホスト名]
```

機能

DABroker を終了します。

使用できるユーザ

DABroker の管理ユーザ

オプション

指定なし

通常の終了をします。

-p

DABroker を計画終了します。

-f

DABroker を強制終了します。

計画終了中に実行すると、計画終了から強制終了に切り替わります。

-c IP アドレス又はホスト名

指定した IP アドレス又はホスト上のクライアントから要求されている処理だけが強制的に中止されます。このとき、データベースの障害処理によってデータベースでの処理がロールバックされ、指定したクライアントとの接続は解放されます。この場合、DABroker は終了しません。ほかのクライアントからの要求は引き続き受け付け、処理の中断を指定したクライアントも再起動すれば接続を再び確立できます。

計画終了中に実行すると、指定した IP アドレス又はホスト上のクライアントの処理だけを中断し、その後は計画終了を続行します。

IP アドレス及びホスト名は dabstat コマンドで参照できます。dabstat コマンドについては、[\[3.8.1 動作状態の表示\]](#) を参照してください。

注意事項

設定項目 DABTCP_TRCSIZE, 又は DABSQL_TRCSIZE で 0 を設定し (通信トレース, 又は SQL トレースを取得), クライアントからの要求量が多い場合, クライアント及び DABroker を終了するまでに時間がかかります。

実行予約を設定したエージェントがある場合の計画終了及び強制終了時は, DABroker のエージェントタイマー監視がキャンセルされ, 再起動時にタイマー監視を再開します。なお, 実行予約を設定したエージェントの処理は, 再起動時に引き継がれます。再起動時に実行予約の期日が過ぎていた場合は, エージェントの実行はエラーとなり, メッセージを出力します。ただし, 設定項目 DABAGTWCHALDAY, 又は DABAGTWCHALTIME で指定した値の有効範囲である場合は, エージェントの処理は引き継がれます。

コマンドが終了しても、終了までに時間が掛かることがあります。このとき再起動しても、終了処理中のため起動できません。しばらく待ってから再起動してください。

3.8 DABroker の処理内容の表示

ここでは、DABroker の処理内容の表示に使うコマンドについて説明します。DABroker の運用前の設定については、「5 DABroker の運用 (共通)」を参照してください。

処理内容を表示するためのコマンドには次のものがあります。

- dabstat (動作状態の表示)
- dabcltrc (クライアントとの通信トレースの表示)
- dabdbtrc (データベースアクセストレースの表示)
- dabagtreinfo (エージェント登録情報の表示)
- dabagtexeinfo (エージェント実行情報の表示)

DABroker で提供するコマンドは、「DABroker 運用ディレクトリ/bin」に格納されています。

3.8.1 動作状態の表示

コマンド形式

```
$ dabstat
```

機能

DABroker と各クライアントとの接続時間、及び予約処理の状態を表示します。
dabstat コマンドは、DABroker の計画終了中にも実行できます。

使用できるユーザ

すべてのユーザ

出力例

\$dabstat	STIME	TIME	CLIENT	batchNo	association
1998/12/16	19:29:23	0:30:37	200.0.0.1	0	*
1998/12/16	17:59:04	2:00:56	host01	1	
1998/12/16	18:10:48	1:49:12	host02	0	*

STIME

そのクライアントからの最初の要求の受付開始日時 (年/月/日 時:分:秒)

TIME

クライアントと接続している時間 (時:分:秒)

CLIENT

クライアントの IP アドレス又はホスト名

hosts ファイル又はネームサーバにホスト名が登録されている場合はホスト名が表示されます。それ以外の場合は IP アドレスが表示されます。

batchNo

実行中の予約処理の本数

association

クライアントとの接続の確立状態

- *：接続を確立している
- 空白：接続は解放しているが、サーバでは要求された予約処理を実行中

注意事項

接続を解放しているクライアントの情報が出力されるのは、そのクライアントから要求された予約処理がサーバで実行中の時だけです。予約処理が実行されていない場合は、接続を解放しているクライアントの情報は表示されません。

3.8.2 クライアントとの通信トレースの表示

コマンド形式

```
# dabcltrc [-c IP アドレス又はホスト名]
```

機能

DABroker とクライアントとの間で行われた、データの送受信に関する情報（通信トレース）を表示します。

なお、通信トレースの取得は、クライアントからの要求受け付け直後、及び DABroker からの結果送信直後に、DABroker が行っています。

使用できるユーザ

DABroker の管理ユーザ

オプション

指定なし

すべてのクライアントとの通信トレースを表示します。

-c IP アドレス又はホスト名

指定した IP アドレス又はホスト名のクライアントとの通信トレースだけを表示します。

出力例

```
#dabcltrc
IPADDR=200.0.0.10 PROT=TCP PAP-NAME=DBPARTNER/Server Driver
DATE      TIME          EVT      SIZE  DATA
1998/12/02 11:38:31.13  PUT      32    Connection(0,0)
1998/12/02 11:39:00.25  GET     256    Connect(Database)
1998/12/02 11:38:20.56  PUT     256    Result(0,0)
1998/12/02 11:40:40.21  GET    1024    LIST (TABLE) <Prepare>
1998/12/02 11:45:23.22  PUT    1024    Result(0,0)
```

IPADDR=XXXXXXXX

クライアントの IP アドレス又はホスト名

PROT=XXX

通信プロトコル。TCP：TCP/IP

PAP-NAME=XXXXXX

次のどれかが表示されます。

- DABroker 関連製品を使用した場合：アプリケーションで設定した名称
- DBPARTNER/Client, DBPARTNER/Web, 又は HITSENSER5 からアクセスした場合：DBPARTNER/Client, DBPARTNER/Web, 又は HITSENSER5 の識別情報
- DBPARTNER Mobile からアクセスした場合：クライアント識別情報
- DBPARTNER Library からアクセスした場合：DBPARTNER Library の識別情報
- XDM/RD 外部表としてアクセスした場合：XDM/RD の識別情報

DATE

送受信の発生日付（年/月/日）

TIME

送受信の発生時刻（時:分:秒.n）

n は n/100 秒を表します。

EVT

DABroker での事象区分

- PUT：クライアントへの送信
- GET：クライアントからの受信

SIZE

送受信データ長（バイト）

DATA

送受信データ種別

aa…aa <bb…bb> (cc…cc,dd…dd) の形式で出力されます。

- aa…aa：オペレーション種別
表示されるオペレーション種別とその意味を次に示します。

オペレーション種別	意 味
Connect (Database)	データベースとの接続要求
Disconnect (Database)	データベースとの切り離し要求
Disconnect (Connection)	コネクションの解放要求
Commit	トランザクションのコミット要求
Rollback	トランザクションのロールバック要求
LIST (Table)	表一覧取得要求

オペレーション種別	意 味
LIST (Column)	列一覧取得要求
LIST (Index)	索引一覧取得要求
LIST (Table Privileges)	表権限一覧取得要求
LIST (Column Privileges)	列権限一覧取得要求
LIST (Inquiry)	予約完了状況一覧取得要求
LIST (PrimaryKey)	プライマリキー一覧取得要求
EXT-CMD	SQL 文の実行要求
DataDivision (Fetch)	データ分割取得
Save	データベースへの表の保存要求
Reserve	予約処理要求
Reserve Delete	予約処理結果の削除要求
Cancel	DBMS に対する処理の中断要求
Illegal Request	不正要求
Connection	コネクション確立応答
Result	正常応答
Error	異常応答
Result (Cancel)	正常応答 (DBMS に対する処理の中断要求時)
Error (Cancel)	異常応答 (DBMS に対する処理の中断要求時)
SQL Fetch Option	SQL 実行オプション実行
File Delete	SQL 実行結果格納ファイル削除
Procedure (List Prepare Procedure)	ストアードプロシジャ一覧取得準備要求
Procedure (List Fetch Procedure)	ストアードプロシジャ一覧取得要求
Procedure (List Close Procedure)	ストアードプロシジャ一覧取得後の処理要求
Procedure (List Prepare Column)	ストアードプロシジャ列一覧取得準備要求
Procedure (List Fetch Column)	ストアードプロシジャ列一覧取得要求
Procedure (List Close Column)	ストアードプロシジャ列一覧取得後の処理要求
Procedure (Parameter Get)	ストアードプロシジャ定義情報取得要求
Procedure (Prepare)	ストアードプロシジャ実行準備要求
Procedure (Execute)	ストアードプロシジャ実行要求
Procedure (Resume)	ストアードプロシジャ Resume 実行
Agent (Register)	エージェントの登録要求

オペレーション種別	意 味
Agent (Delete Registration)	エージェントの削除要求
Agent (List Prepare Registration)	エージェントの登録情報の取得準備要求
Agent (List Fetch Registration)	エージェントの登録情報の取得継続要求
Agent (List Close Registration)	エージェントの登録情報取得後の処理要求
Agent (Execute)	エージェントの実行要求
Agent (List Prepare Execution)	エージェントの実行状況の取得準備要求
Agent (List Fetch Execution)	エージェントの実行状況の取得継続要求
Agent (List Close Execution)	エージェントの実行状況取得後の処理要求
Agent (Get Prepare Result)	エージェント実行結果の取得準備要求
Agent (Get Fetch Result)	エージェント実行結果の取得継続要求
Agent (Get Close Result)	エージェント実行結果の取得後の処理要求
Agent (Delete Result)	エージェントの実行結果の削除要求
Agent (Option)	エージェントのオプション実行要求
Agent (Execute Direct Prepare)	エージェントの即時配信実行準備要求
Agent (Execute Direct Fetch)	エージェントの即時配信実行継続要求
Agent (Execute Direct Close)	エージェントの即時配信実行後処理要求

- bb…bb：付加情報
オペレーション種別が、LIST (xxxxx) 及び EXT-CMD の場合に表示されます。
- cc…cc：リターンコード
- dd…dd：詳細コード

注意事項

クライアントからの要求量が多いほど、クライアントとの通信トレースのファイル容量も多くなります。通信トレースのファイル容量の目安として、概算式を次に示します。

(概算式)

80 (ヘッダ情報長) + クライアントからの要求数 * 2 (PUT 及び GET 分) * 36 (PUT 及び GET 情報長) [バイト]

あらかじめ、この概算式で計算したサイズ以上のハードディスク容量を用意しておいてください。また、クライアントからの要求量が多いほど、トレース表示が終わるまでにも時間がかかります。御注意ください。

3.8.3 エージェント登録情報の表示

コマンド形式

```
$ dabagtreinfo [-r エージェント名] [-u ユーザID]
```

機能

エージェントの登録情報を表示します。DBPARTNER/Client32 又は DBPARTNER2 Client で登録を要求したエージェントの情報です。なお、このコマンドを実行する前に注意事項で示す項目を設定してください。

使用できるユーザ

すべてのユーザ

オプション

指定なし

DABroker に登録されているすべてのエージェント登録情報を表示します。

-r エージェント名

指定したエージェント名の登録情報を表示します。

指定したユーザ ID で登録されている、エージェント登録情報を表示します。

出力例

```
$dabagtreinfo
REGNAME  REGUSER  REGDATE/TIME  RESVDATE/TIME/DAY
在庫検索  USER123  1998/12/04 16:00  1998/12/04 12:00 FRI
:         :         :               :
:         :         :               :
```

REGNAME

エージェント名

REGUSER

登録したユーザ ID

REGDATE/TIME

登録した日付（年/月/日） 時刻（時:分）

RESVDATE/TIME/DAY

実行予約日付（年/月/日） 時刻（時:分） 曜日

実行予約を設定していない、又は実行予約を設定していない項目がある場合は、「*」が表示されます。

（例）

曜日（及び実行時刻）だけを設定した場合：****/**/** 10:15 MON

注意事項

このコマンドを実行する前に、コマンドを実行するウィンドウで次の項目を設定してください。

- DABTMP

- DABAGT
- DABAGTGRP

各設定項目では、DABroker 動作環境定義ファイルで設定した値を指定してください。ユーザのグループ管理をしていない場合、DABAGTGRP の指定は不要です。

3.8.4 エージェント実行情報の表示

コマンド形式

```
$ dabagtexeinfo [-r エージェント名] [-e エージェント実行名] [-u ユーザID]
```

機能

エージェントの実行情報を表示します。DBPARTNER から実行を要求したエージェントの情報です。なお、このコマンドを実行する前に注意事項で示す項目を設定してください。

使用できるユーザ

すべてのユーザ

オプション

指定なし

DABroker で実行されたすべてのエージェント実行情報を表示します。

-r エージェント名

指定した登録名の、エージェント実行情報を表示します。

-e エージェント実行名

指定した実行名の、エージェント実行情報を表示します。

-u ユーザ ID

指定したユーザ ID で実行された、エージェント実行情報を表示します。

出力例

EXENAME	REGNAME	EXECUSER	STATUS	EXECDATE/TIME	EXECPID
00D83667B2AC0000	在庫検索	K654321	TIMER-WATCH	1998/12/05 12:00	1024
⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮

EXECNAME

エージェント実行名

エージェントの実行時に、DABroker が割り当てる識別名（番号）です。

REGNAME

エージェント名

EXECUSER

エージェントの実行を要求したユーザ ID

STATUS

実行状態

- IDLE：初期状態（実行要求待ち状態）
- TIMER-WATCH：エージェントタイマ監視中
- RUN：処理イベントの実行中
- NORMAL-ENDED (xxxxxx)：正常終了
xxxxxx：警告メッセージ ID
警告メッセージ ID がある場合は表示します。
- ABNORMAL-ENDED (xxxxxx)：異常終了
xxxxxx：エラーメッセージ ID

EXECDATE/TIME

実行日付（年/月/日） 時刻（時:分）

EXECPID

実行プロセス ID

注意事項

このコマンドを実行する前に、コマンドを実行するウィンドウで次の項目を設定してください。

- DABTMP
- DABAGT
- DABAGTGRP

各設定項目では、DABroker 動作環境定義ファイルで設定した値を指定してください。ユーザのグループ管理をしていない場合、DABAGTGRP の指定は不要です。

3.9 DABroker コネクションマルチ定義(Database Connection Server 経由の DB アクセス定義)

Database Connection Server を経由してメインフレーム系データベースにアクセスする場合、アプリケーション、又は DBPARTNER は、接続先データベースの定義（「3.6 DABroker 接続先データベースの定義」参照）で定義したコネクションマルチ名を使用してアクセスします。コネクションマルチ名とは、接続する Database Connection Server のポート番号やホスト名などを定義した集まりに対して付ける任意名称です。コネクションマルチ名は、コネクションマルチ定義ファイルで定義します。ここでは、コネクションマルチ名の定義方法について説明します。

3.9.1 コネクションマルチ定義ファイルの作成

Database Connection Server 経由でメインフレーム系データベースに接続する場合、コネクションマルチ定義ファイルで、コネクションマルチ名を定義します。コネクションマルチ定義ファイルは、インストール時に提供されるサンプルを使用して作成します。

表 3-11 に、使用するファイルとファイルの格納先を示します。

表 3-11 DABroker 動作環境定義ファイルのファイル名

使用するサンプルファイル	コネクションマルチ定義ファイル格納先
/opt/DABroker/sample/damconenv	DABroker 運用ディレクトリ/conf

コネクションマルチ定義ファイルの作成手順を次に示します。

1. サンプルファイルを、表 3-11 に示したコネクションマルチ定義ファイルの格納先へコピーする。
2. コピー先ファイルの設定項目の標準値を必要に応じて変更する。また、値が設定されていない項目には値を入力する。
3. ファイルを保存する。

(1) ファイル作成時の注意事項

- 大文字と小文字は区別します。
- 定義名称は、255 文字まで指定できます。
- 定義名称に 2 バイト文字、及び半角の片仮名は使用しないでください。
- 「=」の前後に空白文字、又はタブコードを入れないでください。入れた場合はエラーとなります。

(エラーの例)

```
DABTMP = /tmp/dab
```

- 「#」以降は、コメントとして扱います。

(2) 作成したファイルの変更

- 作成した接続マルチ定義ファイルの内容の変更は、DABroker、及び接続マルチ機能が起動していない時に行ってください。

3.9.2 コネクションマルチ名の設定

一つの接続マルチ名には、次の内容を定義します。

- Database Connection Server のホスト名、又は IP アドレス (DBSHOST)
- Database Connection Server のサービス名、又はポート番号 (DBSPORT)
- Database Connection Server からの応答監視時間 (DBSTIMER)
- 最大仮想接続数 (MCONMAX)
- 通信処理プロセス数 (MCONPROC)
- プール接続数 (MCONPOOL)
- 予約プール接続数 (MRSVPOOL)

図 3-3 に「接続マルチ名」と DBSHOST、DBSPORT の指定内容の関係を示します。

図 3-3 指定内容の関係

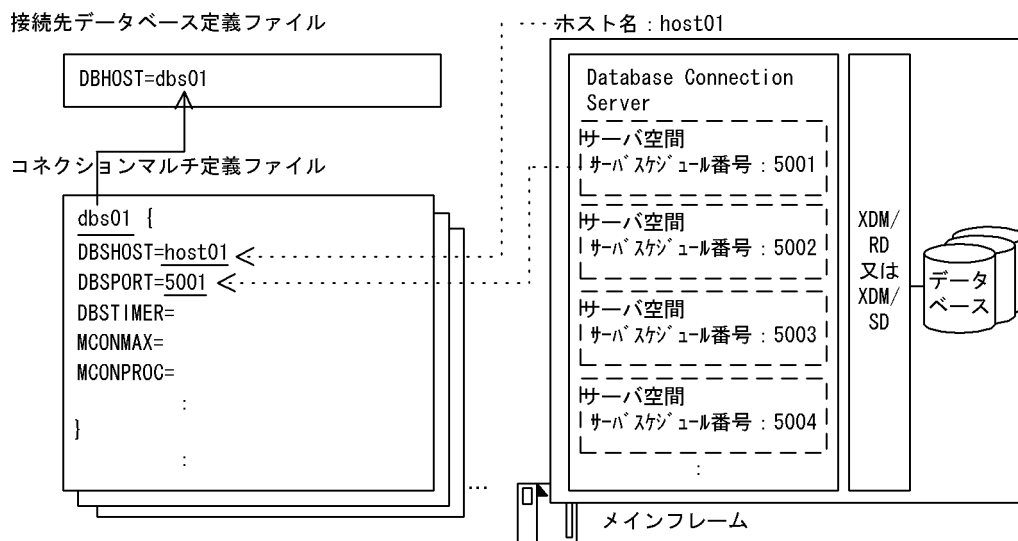


図 3-3 で示すように、Database Connection Server のホスト名 (DBSHOST) には、Database Connection Server、及びアクセスするメインフレーム系データベースが存在するホスト名を指定します。Database Connection Server のポート番号又はサービス名には、Database Connection Server のサーバ空間※に割り当てられているサーバスケジュール番号 (マニュアル「Database Connection Server」参照) を設定します。この例ではポート番号を指定していますが、サービス名を指定することもできます。サービス名を指定するときは、/etc/services ファイルに、サービス名とそれに対応するポート番号を設定してください (「3.3 ネットワーク環境の設定」参照)。

注※ プロセスに相当するメインフレーム側の処理単位です。

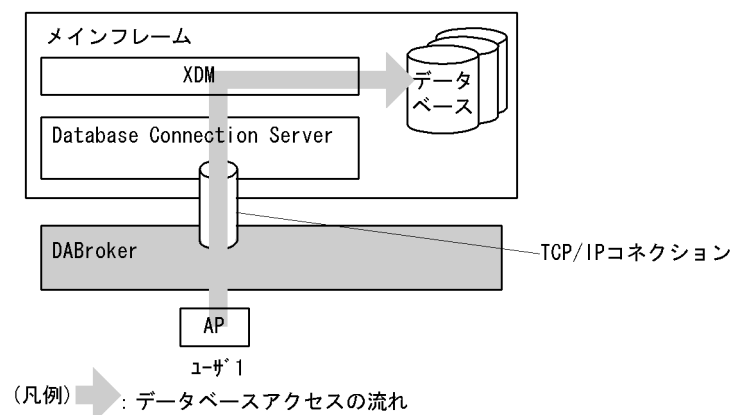
3.9.3 最大仮想コネクション数などを設定するときのガイドライン

最大仮想コネクション数 (MCONMAX), 通信処理プロセス数 (MCONPROC), プールコネクション数 (MCONPOOL), 予約プールコネクション数 (MRSVPOOL) は, 使用環境に合わせた適切な値を設定する必要があります。ここでは, 設定するときのガイドラインを説明します。

(1) DABroker と Database Connection Server の接続の仕組み

DABroker と Database Connection Server の接続時, Database Connection Server では, TCP/IP コネクションを確立します。1 ユーザからのアクセス要求に対して一つの TCP/IP コネクションを確立します (図 3-4 参照)。

図 3-4 DABroker と Database Connection Server の接続イメージ(1)



この場合, 1TCP/IP コネクションを 1 ユーザからのアクセス要求だけで使用でき, レスポンス性能が良くなりますが, 同時アクセスユーザ数が増えると, TCP/IP コネクションの確立数も多くなり Database Connection Server のメインフレーム側及び DABroker 側でのリソース消費量が多くなってしまいます。

そのため, DABroker では, 一つの TCP/IP コネクションを複数ユーザからのアクセス要求で共有できるように, 一つの TCP/IP コネクションの中で複数の DABroker 用のコネクションを確立できるようにしています。一つの TCP/IP コネクションに対して確立する DABroker 用のコネクションを**仮想コネクション**と呼びます。一つの TCP/IP コネクションの中で複数の仮想コネクションを確立するときには, DABroker では, **通信処理プロセス**を起動します。通信処理プロセスは, TCP/IP コネクションごとに起動します。

1TCP/IP コネクションを複数ユーザからのアクセス要求で共有する場合, DABroker では, 1 通信処理プロセス当たりで (=一つの TCP/IP コネクションに対して) 同時に確立できる仮想コネクション数 (=最大仮想コネクション数), 及び DABroker 内で同時に起動できる通信処理プロセス数 (=通信処理プロセス数) を決めて運用します。

図 3-5 に, 1TCP/IP コネクションを 2 ユーザからのアクセス要求で共有する場合 (1 通信処理プロセス当たりで二つの仮想コネクションを確立する場合) の, DABroker と Database Connection Server の接続

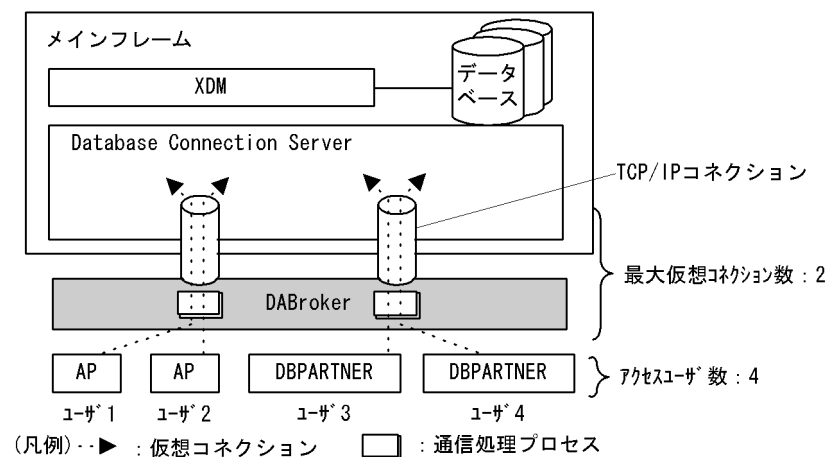
イメージを示します。ここでは、アプリケーション及び DBPARTNER から同時に四つのアクセス要求があり、二つの TCP/IP コネクションで対応しています。DABroker では、二つの通信処理プロセスを起動し、それぞれの通信処理プロセスで二つの仮想コネクションを確立しています。このとき、四つの TCP/IP コネクションを使用した場合に比べ、メインフレーム側及び DABroker 側の両方でリソース消費量を少なくできます。

DABroker は、次のタイミングで Database Connection Server に対して TCP/IP コネクションを確立します。

- 「予約プールコネクション数(MRSVPOOL)」が設定されている (>0) 場合
DABroker コネクションマルチ機能起動時に確立します。
- 「予約プールコネクション数(MRSVPOOL)」が設定されていない (=0) 場合
最初のユーザアクセス要求時に確立します。
- 起動中の通信処理プロセスで実行できるユーザ数(「最大仮想コネクション数(MCONMAX)」)を超えたユーザアクセスが要求された場合
通信処理プロセスを新しく起動して確立します。
(ただし、「通信処理プロセス数(MCONPROC)」の範囲内)

そして、これらの TCP/IP コネクションを確立する際に、1TCP/IP コネクション上に「最大仮想コネクション数(MCONMAX)」に設定された数だけ仮想コネクションを予約します。

図 3-5 DABroker と Database Connection Server の接続イメージ(2)



なお、一つの TCP/IP コネクションに対して一つの仮想コネクションを確立する形態をシングルコネクション (図 3-4 参照) と呼び、一つの TCP/IP コネクションに対して複数の仮想コネクションを確立する形態をコネクションマルチ (図 3-5 参照) と呼びます。

(2) 最大仮想コネクション数と通信処理プロセス数の決定

最大仮想コネクション数 (MCONMAX)

次のどちらかの値を設定します。

- 1TCP/IP コネクションを 1 ユーザからのアクセス要求だけで使用する場合（シングルコネクション）
最大仮想コネクション数を「1」に設定します。この場合、TCP/IP コネクションを 1 ユーザからのアクセス要求だけで使用できるためレスポンス性能が良いのですが、複数ユーザからの同時アクセス要求があるときには同時アクセス要求数分の TCP/IP コネクションを確立し、DABroker 及びメインフレームでのリソース消費量が多くなります。
- 1TCP/IP コネクションを複数ユーザからのアクセス要求で共有する場合（コネクションマルチ）
「2 以上の値」で、1 通信処理プロセス当たりで同時に確立できる仮想コネクション数を設定します。設定値については、下記を参照してください。なお、この場合は、運用時に、コネクションマルチ機能起動コマンド（「3.10.1(1) コネクションマルチ機能起動コマンド (dabcmstart)」参照）を実行してください。

通信処理プロセス数 (MCONPROC)

DABroker で同時に起動できる通信処理プロセス数の最大値を設定します。設定値については、下記を参照してください。なお、最大仮想コネクション数が「1」の場合は、設定する必要はありません。

1TCP/IP コネクションを複数ユーザからのアクセス要求で共有する場合、これらの値は、次の手順で決定してください。

1. DABroker のシステム全体で同時にアクセスするユーザ数（要求数）の最大値を見積もる

この最大値は、アクセス対象となる Database Connection Server の空間起動制御文で設定された「サーバユーザ数」以内となるようにしてください。サーバユーザ数の設定方法については、マニュアル「Database Connection Server」を参照してください。

2. 「1 で見積もった最大値 ≤ 最大仮想コネクション数 * 通信処理プロセス数」となるような最大仮想コネクション数と最大通信処理プロセス数の組み合わせを決定する

このとき、次の 2 点を考慮し、使用環境に合わせた適切な数値を検討してください。

- レスポンス性能
- DABroker が組み込まれたサーバで消費するリソース量、及び Database Connection Server が組み込まれたメインフレームで消費するリソース量

レスポンス性能は、1 通信処理プロセス当たりの仮想コネクション数を少なくすればするほど向上し、最大仮想コネクション数を 1 にする（シングルコネクション）とレスポンスが早くなります。しかし、同時アクセスユーザ数が多い場合に、TCP/IP コネクションも増えるため、メインフレーム側でリソース消費量が多くなります。このため、使用環境に合わせた適切な数値を検討してください。

なお、最大仮想コネクション数は、接続する Database Connection Server の空間起動制御文で設定された「ユーザ数(仮想ユーザ空間数など) ÷ TCP コネクション数」の倍数となるように設定してください。

(3) プールコネクション数と予約プールコネクション数の決定（仮想コネクションのプーリング機能の使用）

最大仮想コネクション数を 2 以上にした場合、次の 2 種類の機能が使用できます。この 2 種類は併用できません。

- 一度確立した仮想コネクションをプールする

この機能を使用すると、コネクション確立のためのオーバーヘッドを削減し、Database Connection Server との接続時間の短縮が図れます。この機能を使用するときは、**プールコネクション数 (MCONPOOL)** を設定してください。

- コネクションマルチ機能の起動時に、指定された数の仮想コネクションを確立しておく

この機能を使用すると、通信処理プロセスの起動のためのオーバーヘッドを削減し、Database Connection Server との接続時間を短縮できます。この機能を使用するときは、**予約プールコネクション数 (MRSVPOOL)** を設定してください。

仮想コネクションのプーリング機能を使用するときは、次に示す値を設定してください。

プールコネクション数 (MCONPOOL)

DABroker のシステム全体でプールする仮想コネクション数を設定します。プールした仮想コネクションに対応する TCP/IP コネクションのためのリソースを Database Connection Server (メインフレーム) 側で占有してしまうため、メインフレーム側でのリソース消費量を考慮して使用環境に合わせた値を設定してください。なお、リソースは、1 通信処理プロセスごと、1 TCP/IP コネクションごとに消費されるため、「最大仮想コネクション数」で設定した値の倍数をプールコネクション数に設定することをお勧めします。

予約プールコネクション数 (MRSVPOOL)

DABroker のシステム全体でコネクションマルチ機能起動時に確立しておく仮想コネクション数を設定します。コネクションマルチ機能起動時にアクセス要求がなくても仮想コネクションを確立してしまうことになるため、DABroker を組み込んだサーバで確立した分のリソースを占有してしまいます。したがって、DABroker 側のリソース消費量を考慮して使用環境に合わせた値を設定してください。なお、リソースは、1 通信処理プロセスごと、1 TCP/IP コネクションごとに消費されるため、「最大仮想コネクション数(MCONMAX)」で設定した値の倍数を予約プールコネクション数に設定することをお勧めします。

プールコネクション数及び予約プールコネクション数の両方に値を設定する場合は、「プールコネクション数 \geq 予約プールコネクション数」となるように設定してください。

3.9.4 コネクションマルチ定義ファイルでの設定項目

コネクションマルチ定義ファイルでは、次の項目を設定します。

コネクションマルチ名

接続する Database Connection Server に付ける任意名称を設定します。

<設定規則>DABroker のシステム内でユニークな名称にしてください。最大 255 文字まで指定できます。2 バイト文字、半角カタカナは使用しないでください。

DBSHOST=Database Connection Server のホスト名又は IP アドレス

接続する Database Connection Server が存在するホスト名、又は IP アドレスを設定します。

ここでの指定を省略した場合は、DABroker 動作環境定義ファイルの DAB_DB_HOST(接続するホスト名, 又は IP アドレス)に指定された値を仮定します。

DBSPORT=Database Connection Server のポート番号又はサービス名

接続する Database Connection Server のポート番号 (Database Connection Server ではサーバスケジューラ番号と呼ぶ) 又はサービス名 (ユーザの任意設定名称) を設定します。

ここで、サービス名を指定した場合は、`/etc/services` ファイルにサービス名と対応するポート番号を設定してください (「3.3 ネットワーク環境の設定」参照)。

ここでの指定を省略した場合は、DABroker 動作環境定義ファイルの DAB_DB_PORT(接続するサービス名, 又はポート番号)に指定された値を仮定します。

DBSTIMER=Database Connection Server からの応答監視時間

Database Connection Server からの処理結果の応答待ち時間を設定します。

指定範囲: 0, 1~65535 (秒)

0 を指定した場合: 応答待ち時間を監視しません。

MCONMAX=最大仮想コネクション数

1 通信処理プロセス当たりで同時に確立できる仮想コネクション数の最大値を設定します。ここで設定した値を超えるユーザからの要求があった場合は、別の通信処理プロセスを起動し、仮想コネクションを確立して Database Connection Server に接続します。なお、データベースには、「最大仮想コネクション数 * 通信処理プロセス数」で算出される数のユーザが同時にアクセスできます。

指定範囲: 1~100

- 1 を設定した場合
シングルコネクションとなります。
- 2 以上を設定した場合
1 通信処理プロセスにつき 2 以上の仮想コネクションを同時に確立できるマルチコネクション (コネクションマルチ) となります。このとき、「通信処理プロセス数(MCONPROC)」も設定してください。なお、この場合は、運用時に、コネクションマルチ機能起動コマンド (「3.10.1(1) コネクションマルチ機能起動コマンド (dabcmstart)」参照) を実行してください。

MCONPROC=通信処理プロセス数

最大仮想コネクション数を 2 以上の値にした場合に必ず設定する項目です。

Database Connection Server と接続するときに起動する最大通信処理プロセス数を設定します。ここで設定した値を超えた場合は、アプリケーション, 又は DBPARTNER にエラーメッセージを返し、処理を受け付けません。

指定範囲: 0, 1~1000

- 指定を省略した場合又は 0 を指定した場合
シングルコネクション (1 通信処理プロセス当たり 1 仮想コネクション) となります。

MCONPOOL=プールコネクション数

最大仮想コネクション数を 2 以上の値にした場合に設定できる項目です。

Database Connection Server に接続するときに確立する仮想コネクションを、データベースアクセスが終了しても、確立されたままの状態プールしておく数を設定します。確立済みの仮想コネクションをプールしておくことで、Database Connection Server と接続するときの仮想コネクション確立時間を短縮できます。

指定範囲：0, 1~32767

指定を省略した場合又は 0 を指定した場合：プール機能を使用できません。

MRSVPOOL=予約プールコネクション数

最大仮想コネクション数を 2 以上の値にした場合に設定できる項目です。

DABroker 起動時に仮想コネクションを確立しておく数を設定します。DABroker 起動時に仮想コネクションが確立されていると、Database Connection Server と接続するときの仮想コネクション確立時間を短縮できます。

指定範囲：0, 1~32767

指定を省略した場合又は 0 を指定した場合：予約プール機能を使用できません。

注意

「最大仮想コネクション数 * 通信処理プロセス数(下記参照)」で算出される値以上の数を設定できません。設定した場合は、コネクションマルチ機能の運用を開始した時点で、通信処理プロセスの起動エラーとなります。

プールコネクション数も設定した場合、予約プールコネクション数では、プールコネクション数より大きな値を設定できません。

3.9.5 コネクションマルチ定義ファイルの作成例

コネクションマルチ定義ファイルの作成例を次に示します。

コネクションマルチ名 db01 はコネクションマルチ機能を使用する例、コネクションマルチ名 db02 はコネクションマルチ機能を使用しない例です。

```
db01{
    DBSHOST='host01'
    DBSPORT='5001'
    DBSTIMER=180
    MCONMAX=2
    MCONPROC=10
    MCONPOOL=5
    MRSVPOOL=3
}
db02{
    DBSHOST='host02'
    DBSPORT='5002'
    DBSTIMER=180
    MCONMAX=1
}
```

3.10 コネクションマルチ機能の使用

コネクションマルチ定義ファイルで、「最大仮想コネクション数(MCONMAX)」を2以上に設定した場合は、コネクションマルチ機能起動コマンドを実行してください。この場合は、コネクションマルチ運用ユーティリティを使って、Database Connection Server と DABroker との接続状態を確認できます。

ここでは、コネクションマルチ機能の運用開始と終了方法、コネクションマルチ運用ユーティリティについて説明します。

DABroker で提供するコマンドは、「DABroker 運用ディレクトリ/bin」に格納されています。

Red Hat Linux の場合、コネクションマルチ機能は使用できません。

3.10.1 コネクションマルチ機能の運用開始と終了

コネクションマルチ機能の運用開始方法と終了方法について説明します。

注意事項

- コネクションマルチ機能が実際に起動・終了されるのは、起動・終了コマンドの実行と同時ではありません。このため、起動・終了の各コマンドの実行が完了しても、しばらくコネクションマルチ機能の起動・終了は動作します。
- 複数の仮想コネクション下に接続し、Database Connection Server 側のタイマ監視などによって、Database Connection Server 側で仮想コネクションが切断された場合、DABroker を使用しているクライアントから Disconnect を発行して処理を終了させないと、仮想コネクションは切断されません。

(1) コネクションマルチ機能起動コマンド(dabcmstart)

コマンド形式

```
# dabcmstart
```

機能

コネクションマルチ機能を起動します。

使用できるユーザ

DABroker の管理ユーザ

注意事項

コネクションマルチ機能は、二つ以上同時に起動できません。

(2) コネクションマルチ機能終了コマンド(dabcmstop)

コマンド形式

```
# dabcmstop [-f | -p]
```

機能

コネクションマルチ機能を終了します。

使用できるユーザ

DABroker の管理ユーザ

オプション

dabcmstop では次のオプションが指定できます。ただし、複数のオプションを同時に指定できません。

-f

起動中の処理すべてを強制終了します(強制終了)。

サーバプロセスは、接続中のユーザに強制終了を受信したことを通知して終了します。

データベースにアクセス中の処理は、この通知を受けた時点で処理を終了します。

計画終了中でも強制終了は実行できます。

-p

新しいユーザからの要求を受け付けないようにして、接続中のユーザの処理がすべて終了した時点で終了します(計画終了)。

計画終了の解除はできません。

省略

接続中のユーザがない場合だけ、終了します(正常終了)。

それ以外の場合はメッセージを出力し、終了コマンドが無効になります。

3.10.2 コネクションマルチ運用ユティリティ

コネクションマルチ運用ユティリティには次の機能があります。

- コネクションマルチ機能の運用状態の表示

接続ユーザ ID からのアクセス要求に対応する仮想コネクション ID ごとの、DABroker と Database Connection Server との接続時間を表示します。

(1) ステータス表示コマンド(dabcmstat)

コマンド形式

```
$ dabcmstat
```

機能

コネクションマルチ機能の動作状態 (ステータス) を表示します。

使用できるユーザ

すべてのユーザ

出力例

```
$ /DBBROKER/bin/dabcmstat
  CONNTIME          USERTIME  USERID  STATUS  VCONID  SVPROCID  DMNID  CMNAME
1999/09/01 17:02:02 1999/09/01 17:02:10 SYS1    USE     1         1001   231   CM001
1999/09/01 17:02:05                POOL    2         1002   231   CM001
1999/09/01 17:05:04 1999/09/01 17:05:10 SYS1    USE     1         2005   245   CM002
1999/09/01 17:05:05                POOL    2         2006   245   CM002
```

CONNTIME：接続開始時間

USERTIME：ユーザ開始時間※

USERID：接続ユーザ ID※

STATUS：仮想接続の状態

- USE：仮想接続の使用
- POOL：仮想接続のプール待機中

VCONID：仮想接続 ID

DMNID：通信デーモン ID

SVPROCID：サーバプロセス ID

CMNAME：接続マルチ名

注※ 仮想接続の状態が「USE（ユーザ使用中）」の場合だけ表示されます。

3.11 ORACLE アクセスライブラリ再作成方法

DABroker では、ORACLE をアクセスするためのライブラリとして ORACLE アクセスライブラリを提供しています。ORACLE アクセスライブラリは、ORACLE の OCI プログラムです。OCI プログラムは、ORACLE のインストール状況によって、その作成方法（make 方法）が異なる場合があります。ORACLE のインストールの状況によっては、DABroker が提供している ORACLE アクセスライブラリでは ORACLE にアクセスできず、ユーザの環境に合わせた ORACLE アクセスライブラリの再作成が必要になります。

ここでは、ORACLE アクセスライブラリを作成する機能について、UNIX の種類ごとに説明します。※

注※

Oracle12c 以降のバージョンの場合はアクセスライブラリを再作成する必要はありません。Oracle12c 以降のバージョンのサンプルメイクファイルは提供していません。

3.11.1 HP-UX の場合

HP-UX 使用時の ORACLE アクセスライブラリ再作成方法について説明します。

(1) 適用範囲

ORACLE アクセスライブラリを作成する機能は、次の ORACLE システムを運用している場合に適用してください。

- Oracle8i R8.1.5 を使用し、「Oracle Advanced Security Kerberos 機能」をインストールしていない場合
- DABroker を動作させた時、次のメッセージで ORACLE の関数名、又はライブラリ名が出力された場合

DABroker 出力メッセージ

```
KFDB005MM-E システムコールでエラーが発生しました。
```

```
[dlopen(/opt/DABroker/lib/libdaO7815.sl*) errno=ld.sl.1:
```

```
重大なエラー:再配置エラー:ファイル /opt/DABroker/lib/libdaO7815.sl*:
```

```
シンボル (ORACLE の関数):参照されたシンボルがありません]
```

注※

- Oracle8 R8.0.5 を使用している場合 : libdaO7805.sl と表示されます。
- Oracle8 R8.0.6 を使用している場合 : libdaO7806.sl と表示されます。
- Oracle8i R8.1.5 を使用している場合 : libdaO7815.sl と表示されます。
- Oracle8i R8.1.6 を使用している場合 : libdaO7816.sl と表示されます。
- Oracle8i R8.1.7 を使用している場合 : libdaO7817.sl と表示されます。

- Oracle9i R9.0.1 を使用している場合：libdaO7901.sl と表示されます。
- Oracle9i R9.2.0 を使用している場合：libdaO7920.sl と表示されます。
- Oracle10g R10.1.0 または R10.2.0 を使用している場合：libdaO7101.sl と表示されます。
- Oracle11g R11.1.0 または R11.2.0 を使用している場合：libdaO7111.sl と表示されます。
- DABroker for Java のデータベース種別に Oracle8i を指定して、Oracle8i R8.1.7 を使用している場合：libdaO8817.sl と表示されます。
- DABroker for Java のデータベース種別に Oracle8i を指定して、Oracle9i R9.0.1 を使用している場合：libdaO8901.sl と表示されます。
- DABroker for Java のデータベース種別に Oracle8i を指定して、Oracle9i R9.2.0 を使用している場合：libdaO8920.sl と表示されます。
- DABroker for Java のデータベース種別に Oracle8i を指定して、Oracle10g R10.1.0 または R10.2.0 を使用している場合：libdaO8101.sl と表示されます。
- DABroker for Java のデータベース種別に Oracle8i を指定して、Oracle11g R11.1.0 または R11.2.0 を使用している場合：libdaO8111.sl と表示されます。

(2) ORACLE アクセスライブラリ再作成の手順

ORACLE アクセスライブラリを再作成する手順を次に説明します。

次の手順は、必ず、運用ディレクトリが/opt/DABroker の環境で実行してください。

DABroker の運用ディレクトリを/opt/DABroker 以外に設定している場合は、いったん、DABroker のセットアップコマンドで運用ディレクトリを/opt/DABroker に戻してから実行してください。

1. DABroker 既存ライブラリのバックアップ

DABroker の既存ライブラリのバックアップを次のように取得してください。

```
cd /opt/DABroker/lib
```

- Oracle8 R8.0.5 を使用している場合
mv libdaO7805.sl libdaO7805.sl_bk
- Oracle8 R8.0.6 を使用している場合
mv libdaO7806.sl libdaO7806.sl_bk
- Oracle8i R8.1.5 を使用している場合
mv libdaO7815.sl libdaO7815.sl_bk
- Oracle8i R8.1.6 を使用している場合
mv libdaO7816.sl libdaO7816.sl_bk
- Oracle8i R8.1.7 を使用している場合
mv libdaO7817.sl libdaO7817.sl_bk
- Oracle9i R9.0.1 を使用している場合


```
mv libdaO7901.sl libdaO7901.sl_bk
```

- Oracle9i R9.2.0 を使用している場合

```
mv libdaO7920.sl libdaO7920.sl_bk
```

- Oracle10g R10.1.0 または R10.2.0 を使用している場合

```
mv libdaO7101.sl libdaO7101.sl_bk
```

- Oracle11g R11.1.0 または R11.2.0 を使用している場合

```
mv libdaO7111.sl libdaO7111.sl_bk
```

- DABroker for Java のデータベース種別に Oracle8i を指定して、Oracle8i R8.1.7 を使用している場合

```
mv libdaO8817.sl libdaO8817.sl_bk
```

- DABroker for Java のデータベース種別に Oracle8i を指定して、Oracle9i R9.0.1 を使用している場合

```
mv libdaO8901.sl libdaO8901.sl_bk
```

- DABroker for Java のデータベース種別に Oracle8i を指定して、Oracle9i R9.2.0 を使用している場合

```
mv libdaO8920.sl libdaO8920.sl_bk
```

- DABroker for Java のデータベース種別に Oracle8i を指定して、Oracle10g R10.1.0 または R10.2.0 を使用している場合

```
mv libdaO8101.sl libdaO8101.sl_bk
```

- DABroker for Java のデータベース種別に Oracle8i を指定して、Oracle11g R11.1.0 または R11.2.0 を使用している場合

```
mv libdaO8111.sl libdaO8111.sl_bk
```

2. 提供サンプルメイクファイルの修正

DABroker では、表 3-12~25 に示す、各 ORACLE アクセスライブラリのサンプルメイクファイルを提供しています。使用している環境に合わせて、修正して使用してください。

なお、表中の番号は、次の内容を示しています。

1 : DABroker アクセスライブラリ名称

修正しないでください。

2 : DABroker 共用ライブラリ

修正しないでください。

3 : ORACLE クライアントライブラリ、及び OS 提供ライブラリ

DABroker の ORACLE アクセスライブラリは、ORACLE の OCI プログラムです。ORACLE のマニュアルを参照し、OCI プログラム作成時に必要な ORACLE クライアントライブラリ、及び OS 提供ライブラリを確認の上、使用している環境に合わせて修正してください。

4: DABroker 共用オブジェクトファイル

修正しないでください。

5: リンケージパラメタ

修正しないでください。

- Oracle8 R8.0.5 を使用している場合

ファイル名: /opt/DABroker/sample/oramake7805

表 3-12 Oracle8 R8.0.5 サンプルメイクファイル

番号	定義文
1	PROGRAM = libda07805.sl
2	LIBS = -L/opt/DABroker/lib -ldaccommon
3	LIBS1 = -l:libcl.sl -lpthread
3	ORALIB = -L \$(ORACLE_HOME)/lib -l:libclntsh.sl
3	LDFLAGS = -b +s
4	OBJS = \$(DABDIR)/obj/dacdcntl.o \$(DABDIR)/obj/da07805.a
5	all: \$(PROGRAM)
5	<pre>\$(PROGRAM): \$(OBJS) echo "Linking \$(PROGRAM) ..." \$(LD) -o \$(PROGRAM) \$(OBJS) \$(LIBS) \$(LIBS1) \$(LDFLAGS) ¥ \$(ORALIB) echo "done"</pre>

- Oracle8 R8.0.6 を使用している場合

ファイル名: /opt/DABroker/sample/oramake7806

表 3-13 Oracle8 R8.0.6 サンプルメイクファイル

番号	定義文
1	PROGRAM = libda07806.sl
2	LIBS = -L/opt/DABroker/lib -ldaccommon
3	LIBS1 = -l:libcl.sl -lpthread
3	ORALIB = -L \$(ORACLE_HOME)/lib -l:libclntsh.sl
3	LDFLAGS = -b +s
4	OBJS = \$(DABDIR)/obj/dacdcntl.o \$(DABDIR)/obj/da07806.a
5	all: \$(PROGRAM)
5	<pre>\$(PROGRAM): \$(OBJS) echo "Linking \$(PROGRAM) ..."</pre>

番号	定義文
	<pre>\$(LD) -o \$(PROGRAM) \$(OBS) \$(LIBS) \$(LIBS1) \$(LDFLAGS) ¥ \$(ORALIB) echo "done"</pre>

- Oracle8i R8.1.5 を使用している場合
ファイル名：/opt/DABroker/sample/oramake7815

表 3-14 Oracle8i R8.1.5 サンプルメイクファイル

番号	定義文
1	PROGRAM = libda07815.sl
2	LIBS = -L/opt/DABroker/lib -ldaccommon
3	LIBS1 = -l:libcl.sl -lpthread
3	ORALIB = -L \$(ORACLE_HOME)/lib -l:libclntsh.sl
3	LDFLAGS = -b +s
4	OBS = \$(DABDIR)/obj/dacdcntl.o \$(DABDIR)/obj/da07815.a
5	all: \$(PROGRAM)
5	<pre>\$(PROGRAM): \$(OBS) echo "Linking \$(PROGRAM) ..." \$(LD) -o \$(PROGRAM) \$(OBS) \$(LIBS) \$(LIBS1) \$(LDFLAGS) ¥ \$(ORALIB) echo "done"</pre>

- Oracle8i R8.1.6 を使用している場合
ファイル名：/opt/DABroker/sample/oramake7816

表 3-15 Oracle8i R8.1.6 サンプルメイクファイル

番号	定義文
1	PROGRAM = libda07816.sl
2	LIBS = -L/opt/DABroker/lib -ldaccommon
3	LIBS1 = -l:libcl.sl -lpthread
3	ORALIB = -L \$(ORACLE_HOME)/lib -l:libclntsh.sl
3	LDFLAGS = -b +s
4	OBS = \$(DABDIR)/obj/dacdcntl.o \$(DABDIR)/obj/da07816.a
5	all: \$(PROGRAM)
5	<pre>\$(PROGRAM): \$(OBS) echo "Linking \$(PROGRAM) ..." \$(LD) -o \$(PROGRAM) \$(OBS) \$(LIBS) \$(LIBS1) \$(LDFLAGS) ¥</pre>

番号	定義文
	\$(ORALIB) echo "done"

- Oracle8i R8.1.7 を使用している場合
ファイル名：/opt/DABroker/sample/oramake7817

表 3-16 Oracle8i R8.1.7 サンプルメイクファイル

番号	定義文
1	PROGRAM = libda07817.sl
2	LIBS = -L/opt/DABroker/lib -ldaccommon
3	LIBS1 = -l:libcl.sl -lpthread
3	ORALIB = -L \$(ORACLE_HOME)/lib -l:libclntsh.sl
3	LDFLAGS = -b +s
4	OBJS = \$(DABDIR)/obj/dacdcntl.o \$(DABDIR)/obj/da07817.a
5	all: \$(PROGRAM)
5	\$(PROGRAM): \$(OBJS) echo "Linking \$(PROGRAM) ..." \$(LD) -o \$(PROGRAM) \$(OBJS) \$(LIBS) \$(LIBS1) \$(LDFLAGS) ¥ \$(ORALIB) echo "done"

- Oracle9i R9.0.1 を使用している場合
ファイル名：/opt/DABroker/sample/oramake7901

表 3-17 Oracle9i R9.0.1 サンプルメイクファイル

番号	定義文
1	PROGRAM = libda07901.sl
2	LIBS = -L/opt/DABroker/lib -ldaccommon
3	LIBS1 = -l:libcl.sl -lpthread
3	ORALIB = -L \$(ORACLE_HOME)/lib32 -l:libclntsh.sl
3	LDFLAGS = -b +s
4	OBJS = \$(DABDIR)/obj/dacdcntl.o \$(DABDIR)/obj/da07901.a
5	all: \$(PROGRAM)
5	\$(PROGRAM): \$(OBJS) echo "Linking \$(PROGRAM) ..." \$(LD) -o \$(PROGRAM) \$(OBJS) \$(LIBS) \$(LIBS1) \$(LDFLAGS) ¥ \$(ORALIB)

番号	定義文
	echo "done"

- Oracle9i R9.2.0 を使用している場合
ファイル名：/opt/DABroker/sample/oramake7920

表 3-18 Oracle9i R9.2.0 サンプルメイクファイル

番号	定義文
1	PROGRAM = libda07920.sl
2	LIBS = -L/opt/DABroker/lib -ldaccommon
3	LIBS1 = -l:libcl.sl -lpthread
3	ORALIB = -L \$(ORACLE_HOME)/lib32 -l:libclntsh.sl
3	LDFLAGS = -b +s
4	OBJS = \$(DABDIR)/obj/dacdcntl.o \$(DABDIR)/obj/da07920.a
5	all: \$(PROGRAM)
5	\$(PROGRAM): \$(OBJS) echo "Linking \$(PROGRAM) ..." \$(LD) -o \$(PROGRAM) \$(OBJS) \$(LIBS) \$(LIBS1) \$(LDFLAGS) ¥ \$(ORALIB) echo "done"

- Oracle 10g R10.1.0 または R10.2.0 を使用している場合
ファイル名：/opt/DABroker/sample/oramake7101

表 3-19 Oracle 10g サンプルメイクファイル

番号	定義文
1	PROGRAM = libda07101.sl
2	LIBS = -L /opt/DABroker/lib -ldaccommon
3	LIBS1 = -l:libcl.sl -lpthread
3	ORALIB = -L \$(ORACLE_HOME)/lib -l:libclntsh.sl
3	LDFLAGS = -b +s
4	OBJS = \$(DABDIR)/obj/dacdcntl.o \$(DABDIR)/obj/da07101.a
5	all: \$(PROGRAM)
5	\$(PROGRAM): \$(OBJS) echo "Linking \$(PROGRAM) ..." \$(LD) -o \$(PROGRAM) \$(OBJS) \$(LIBS) \$(LIBS1) \$(LDFLAGS) ¥ \$(ORALIB) echo "done"

- Oracle 11g R11.1.0 または R11.2.0 を使用している場合
ファイル名：/opt/DABroker/sample/oramake7111

表 3-20 Oracle 11g サンプルメイクファイル

番号	定義文
1	PROGRAM = libda07111.sl
2	LIBS = -L /opt/DABroker/lib -ldaccommon
3	LIBS1 = -l:libcl.sl -lpthread
3	ORALIB = -L \$(ORACLE_HOME)/lib -l:libclntsh.sl
3	LDFLAGS = -b +s
4	OBJS = \$(DABDIR)/obj/dacdctl.o \$(DABDIR)/obj/da07111.a
5	all: \$(PROGRAM)
5	\$(PROGRAM): \$(OBJS) echo "Linking \$(PROGRAM) ..." \$(LD) -o \$(PROGRAM) \$(OBJS) \$(LIBS) \$(LIBS1) \$(LDFLAGS) ¥ \$(ORALIB) echo "done"

- DABroker for Java のデータベース種別に Oracle8i を指定して、Oracle8i R8.1.7 を使用している場合

ファイル名：/opt/DABroker/sample/oramake8817

表 3-21 Oracle8i R8.1.7 サンプルメイクファイル (DABroker for Java のデータベース種別で Oracle8i 指定時)

番号	定義文
1	PROGRAM = libda08817.sl
2	LIBS = -L/opt/DABroker/lib -ldaccommon
3	LIBS1 = -l:libcl.sl -lpthread
3	ORALIB = -L \$(ORACLE_HOME)/lib -l:libclntsh.sl
3	LDFLAGS = -b +s
4	OBJS = \$(DABDIR)/obj/dacd8817.o \$(DABDIR)/obj/da08817.a
5	all: \$(PROGRAM)
5	\$(PROGRAM): \$(OBJS) echo "Linking \$(PROGRAM) ..." \$(LD) -o \$(PROGRAM) \$(OBJS) \$(LIBS) \$(LIBS1) \$(LDFLAGS) ¥ \$(ORALIB) echo "done"

- DABroker for Java のデータベース種別に Oracle8i を指定して、Oracle9i R9.0.1 を使用している場合

ファイル名：/opt/DABroker/sample/oramake8901

表 3-22 Oracle9i R9.0.1 サンプルメイクファイル (DABroker for Java のデータベース種別で Oracle8i 指定時)

番号	定義文
1	PROGRAM = libda08901.sl
2	LIBS = -L/opt/DABroker/lib -ldaccommon
3	LIBS1 = -l:libcl.sl -lpthread
3	ORALIB = -L \$(ORACLE_HOME)/lib32 -l:libclntsh.sl
3	LDFLAGS = -b +s
4	OBJS = \$(DABDIR)/obj/dacd8901.o \$(DABDIR)/obj/da08901.a
5	all: \$(PROGRAM)
5	\$(PROGRAM): \$(OBJS) echo "Linking \$(PROGRAM) ..." \$(LD) -o \$(PROGRAM) \$(OBJS) \$(LIBS) \$(LIBS1) \$(LDFLAGS) ¥ \$(ORALIB) echo "done"

- DABroker for Java のデータベース種別に Oracle8i を指定して、Oracle9i R9.2.0 を使用している場合

ファイル名：/opt/DABroker/sample/oramake8920

表 3-23 Oracle9i R9.2.0 サンプルメイクファイル (DABroker for Java のデータベース種別で Oracle8i 指定時)

番号	定義文
1	PROGRAM = libda08920.sl
2	LIBS = -L/opt/DABroker/lib -ldaccommon
3	LIBS1 = -l:libcl.sl -lpthread
3	ORALIB = -L \$(ORACLE_HOME)/lib32 -l:libclntsh.sl
3	LDFLAGS = -b +s
4	OBJS = \$(DABDIR)/obj/dacd8920.o \$(DABDIR)/obj/da08920.a
5	all: \$(PROGRAM)
5	\$(PROGRAM): \$(OBJS) echo "Linking \$(PROGRAM) ..." \$(LD) -o \$(PROGRAM) \$(OBJS) \$(LIBS) \$(LIBS1) \$(LDFLAGS) ¥

番号	定義文
	\$(ORALIB) echo "done"

- DABroker for Java のデータベース種別に Oracle8i を指定して、Oracle 10g R10.1.0 または R10.2.0 を使用している場合

ファイル名：/opt/DABroker/sample/oramake8101

表 3-24 Oracle 10g サンプルメイクファイル(DABroker for Java のデータベース種別に Oracle8i を指定時)

番号	定義文
1	PROGRAM = libda08101.sl
2	LIBS = -L /opt/DABroker/lib -ldaccommon
3	LIBS1 = -l:libcl.sl -lpthread
3	ORALIB = -L \$(ORACLE_HOME)/lib -l:libclntsh.sl
3	LDFLAGS = -b +s
4	OBJS = \$(DABDIR)/obj/dacd8101.o \$(DABDIR)/obj/da08101.a
5	all: \$(PROGRAM)
5	\$(PROGRAM): \$(OBJS) echo "Linking \$(PROGRAM) ..." \$(LD) -o \$(PROGRAM) \$(OBJS) \$(LIBS) \$(LIBS1) \$(LDFLAGS) ¥ \$(ORALIB) echo "done"

- DABroker for Java のデータベース種別に Oracle8i を指定して、Oracle 11g R11.1.0 または R11.2.0 を使用している場合

ファイル名：/opt/DABroker/sample/oramake8111

表 3-25 Oracle 11g サンプルメイクファイル(DABroker for Java のデータベース

番号	定義文
1	PROGRAM = libda08111.sl
2	LIBS = -L /opt/DABroker/lib -ldaccommon
3	LIBS1 = -l:libcl.sl -lpthread
3	ORALIB = -L \$(ORACLE_HOME)/lib -l:libclntsh.sl
3	LDFLAGS = -b +s
4	OBJS = \$(DABDIR)/obj/dacd8111.o \$(DABDIR)/obj/da08111.a
5	all: \$(PROGRAM)

番号	定義文
5	<pre>\$(PROGRAM): \$(OBJS) echo "Linking \$(PROGRAM) ..." \$(LD) -o \$(PROGRAM) \$(OBJS) \$(LIBS) \$(LIBS1) \$(LDFLAGS) ¥ \$(ORALIB) echo "done"</pre>

3. DABroker ORACLE アクセスライブラリの再作成

サンプルメイクファイルを修正後、次の環境変数を入力し、make コマンドを実行してください。

(メイクファイル実行手順 B シェルを使用の場合)

ORACLE_HOME = ORACLE HOME ディレクトリ

export ORACLE_HOME

DABDIR =/opt/DABroker

export DABDIR

- Oracle8 R8.0.5 を使用している場合
make -f oramake7805 (修正後のメイクファイルを実行する)
- Oracle8 R8.0.6 を使用している場合
make -f oramake7806 (修正後のメイクファイルを実行する)
- Oracle8i R8.1.5 を使用している場合
make -f oramake7815 (修正後のメイクファイルを実行する)
- Oracle8i R8.1.6 を使用している場合
make -f oramake7816 (修正後のメイクファイルを実行する)
- Oracle8i R8.1.7 を使用している場合
make -f oramake7817 (修正後のメイクファイルを実行する)
- Oracle9i R9.0.1 を使用している場合
make -f oramake7901 (修正後のメイクファイルを実行する)
- Oracle9i R9.2.0 を使用している場合
make -f oramake7920 (修正後のメイクファイルを実行する)
- Oracle 10g R10.1.0 または R10.2.0 を使用している場合
make -f Oramake7101 (修正後のメイクファイルを実行する)
- Oracle 11g R11.1.0 または R11.2.0 を使用している場合
make -f Oramake7111 (修正後のメイクファイルを実行する)
- DABroker for Java のデータベース種別に Oracle8i を指定して、Oracle8i R8.1.7 を使用している場合
make -f oramake8817 (修正後のメイクファイルを実行する)

- DABroker for Java のデータベース種別に Oracle8i を指定して、Oracle9i R9.0.1 を使用している場合
make -f oramake8901 (修正後のメイクファイルを実行する)
- DABroker for Java のデータベース種別に Oracle8i を指定して、Oracle9i R9.2.0 を使用している場合
make -f oramake8920 (修正後のメイクファイルを実行する)
- DABroker for Java のデータベース種別に Oracle8i を指定して、Oracle 10g R10.1.0 または R10.2.0 を使用している場合
make -f Oramake8101 (修正後のメイクファイルを実行する)
- DABroker for Java のデータベース種別に Oracle8i を指定して、Oracle 11g R11.1.0 または R11.2.0 を使用している場合
make -f Oramake8111 (修正後のメイクファイルを実行する)

4. 再作成後の ORACLE アクセスライブラリを DABroker のライブラリ格納用ディレクトリにコピーする

make コマンドを実行したディレクトリ環境下で再作成した ORACLE アクセスライブラリを次のようにコピーしてください。

再作成した ORACLE アクセスライブラリは、make コマンドを実行したカレントディレクトリに作成されます。

- Oracle8 R8.0.5 を使用している場合
cp libdaO7805.sl /opt/DABroker/lib/libdaO7805.sl
- Oracle8 R8.0.6 を使用している場合
cp libdaO7806.sl /opt/DABroker/lib/libdaO7806.sl
- Oracle8i R8.1.5 を使用している場合
cp libdaO7815.sl /opt/DABroker/lib/libdaO7815.sl
- Oracle8i R8.1.6 を使用している場合
cp libdaO7816.sl /opt/DABroker/lib/libdaO7816.sl
- Oracle8i R8.1.7 を使用している場合
cp libdaO7817.sl /opt/DABroker/lib/libdaO7817.sl
- Oracle9i R9.0.1 を使用している場合
cp libdaO7901.sl /opt/DABroker/lib/libdaO7901.sl
- Oracle9i R9.2.0 を使用している場合
cp libdaO7920.sl /opt/DABroker/lib/libdaO7920.sl
- Oracle 10g R10.1.0 または R10.2.0 を使用している場合
cp libdaO7101.sl /opt/DABroker/lib/libdaO7101.sl
- Oracle 11g R11.1.0 または R11.2.0 を使用している場合
cp libdaO7111.sl /opt/DABroker/lib/libdaO7111.sl

- DABroker for Java のデータベース種別に Oracle8i を指定して、Oracle8i R8.1.7 を使用している場合
`cp libdaO8817.sl /opt/DABroker/lib/libdaO8817.sl`
- DABroker for Java のデータベース種別に Oracle8i を指定して、Oracle9i R9.0.1 を使用している場合
`cp libdaO8901.sl /opt/DABroker/lib/libdaO8901.sl`
- DABroker for Java のデータベース種別に Oracle8i を指定して、Oracle9i R9.2.0 を使用している場合
`cp libdaO8920.sl /opt/DABroker/lib/libdaO8920.sl`
- DABroker for Java のデータベース種別に Oracle8i を指定して、Oracle 10g R10.1.0 または R10.2.0 を使用している場合
`cp libdaO8101.sl /opt/DABroker/lib/libdaO8101.sl`
- DABroker for Java のデータベース種別に Oracle8i を指定して、Oracle 11g R11.1.0 または R11.2.0 を使用している場合
`cp libdaO8111.sl /opt/DABroker/lib/libdaO8111.sl`

5. DABroker の再セットアップ

作成した ORACLE アクセスライブラリを /opt/DABroker/lib にコピーした後、再度、DABroker のセットアップコマンドを実行してください。

3.11.2 AIX の場合

AIX 使用時の ORACLE アクセスライブラリ再作成方法について説明します。

(1) 適用範囲

ORACLE アクセスライブラリを作成する機能は、次の ORACLE システムを運用している場合に適用してください。

- DABroker を動作させたとき、次のメッセージで ORACLE の関数名、またはライブラリ名が出力された場合

DABroker 出力メッセージ

KFDB005MM-E システムコールでエラーが発生しました。

[ldlopen(/opt/DABroker/lib/libdaO7101.sl*) errno=ld.sl.1:

重大なエラー：再配置エラー：ファイル /opt/DABroker/lib/libdaO7101.sl*：

シンボル (ORACLE の関数)：参照されたシンボルがありません]

注※

- Oracle10g R10.1.0 または R10.2.0 を使用している場合：libdaO7101.sl と表示されます。

- Oracle11g R11.1.0 または R11.2.0 を使用している場合：libdaO7111.sl と表示されます。
- DABroker for Java のデータベース種別に Oracle8i を指定して、Oracle10g R10.1.0 または R10.2.0 を使用している場合：libdaO8101.sl と表示されます。
- DABroker for Java のデータベース種別に Oracle8i を指定して、Oracle11g R11.1.0 または R11.2.0 を使用している場合：libdaO8111.sl と表示されます。

(2) ORACLE アクセスライブラリ再作成の手順

ORACLE アクセスライブラリを再作成する手順を次に説明します。

次の手順は、必ず、運用ディレクトリが/opt/DABroker の環境で実行してください。

DABroker の運用ディレクトリを/opt/DABroker 以外に設定している場合は、いったん、DABroker のセットアップコマンドで運用ディレクトリを/opt/DABroker に戻してから実行してください。

1. DABroker 既存ライブラリのバックアップ

DABroker の既存ライブラリのバックアップを次のように取得してください。

```
cd /opt/DABroker/lib
```

- Oracle8i R8.1.7 を使用している場合
mv libdaO7817.a libdaO7817.a_bk
- Oracle9i R9.2.0 を使用している場合
mv libdaO7920.a libdaO7920.a_bk
- Oracle10g R10.1.0 または R10.2.0 を使用している場合
mv libdaO7101.a libdaO7101.a_bk
- Oracle11g R11.1.0 または R11.2.0 を使用している場合
mv libdaO7111.a libdaO7111.a_bk
- DABroker for Java のデータベース種別に Oracle8i を指定して、Oracle8i R8.1.7 を使用している場合
mv libdaO8817.a libdaO8817.a_bk
- DABroker for Java のデータベース種別に Oracle8i を指定して、Oracle9i R9.2.0 を使用している場合
mv libdaO8920.a libdaO8920.a_bk
- DABroker のデータベース種別に ORACLE8i を指定して、Oracle10g R10.1.0 または R10.2.0 を使用している場合
mv libdaO8101.a libdaO8101.a_bk
- DABroker のデータベース種別に ORACLE8i を指定して、Oracle11g R11.1.0 または R11.2.0 を使用している場合
mv libdaO8111.a libdaO8111.a_bk

2. 提供サンプルメイクファイルの修正

DABrokerでは、表 3-26~33 に示す、ORACLE アクセスライブラリのサンプルメイクファイルを提供しています。使用している環境に合わせて、修正して使用してください。

なお、表中の番号は、次の内容を示しています。

1：DABroker アクセスライブラリ名称

修正しないでください。

2：DABroker 共用ライブラリ

修正しないでください。

3：ORACLE クライアントライブラリ，及び OS 提供ライブラリ

DABroker の ORACLE アクセスライブラリは、ORACLE の OCI プログラムです。ORACLE のマニュアルを参照し、OCI プログラム作成時に必要な ORACLE クライアントライブラリ，及び OS 提供ライブラリを確認の上、使用している環境に合わせて修正してください。

4：DABroker 共用オブジェクトファイル

修正しないでください。

5：リンケージパラメタ

修正しないでください。

- Oracle8i R8.1.7 を使用している場合

ファイル名：/opt/DABroker/sample/oramake7817

表 3-26 Oracle8i R8.1.7 サンプルメイクファイル

番号	定義文
1	PROGRAM = libda07817.a
1	PROGRAM1 = libda07817.o
2	LIBS = -L /opt/DABroker/lib -ldaccommon
3	LIBS1 = -lpthreads -bglink:/lib/glink.o -bM:SRE -bexpall ¥
3	-bnoentry -lm -lc_r
3	ORALIB = -L \$(ORACLE_HOME)/lib -lcIntsh
3	LDFLAGS = -G
4	OBJS = /opt/DABroker/obj/dacdcntl.o ¥
4	/opt/DABroker/obj/da07817.a
5	all: \$(PROGRAM)
5	\$(PROGRAM): \$(OBJS)
	echo "Linking \$(PROGRAM) ..."
	ld -o \$(PROGRAM1) \$(OBJS) \$(LIBS) \$(LIBS1) \$(LDFLAGS) ¥
	\$(ORALIB)
	ar rv \$(PROGRAM) \$(PROGRAM1)
	rm -f \$(PROGRAM1)
	echo "done"

- Oracle9i R9.2.0 を使用している場合

ファイル名：/opt/DABroker/sample/oramake7920

表 3-27 Oracle9i R9.2.0 サンプルメイクファイル

番号	定義文
1	PROGRAM = libda07920.a
1	PROGRAM1 = libda07920.o
2	LIBS = -L /opt/DABroker/lib -ldaccommon
3	LIBS1 = -lpthreads -bglink:/lib/glink.o -bM:SRE -bexpall ¥
3	-bnoentry -lm -lc_r
3	ORALIB = -L \$(ORACLE_HOME)/lib -lclntsh
3	LDFLAGS = -G
4	OBJS = /opt/DABroker/obj/dacdcntl.o ¥
4	/opt/DABroker/obj/da07920.a
5	all: \$(PROGRAM)
5	\$(PROGRAM): \$(OBJS)
	echo "Linking \$(PROGRAM) ..."
	ld -o \$(PROGRAM1) \$(OBJS) \$(LIBS) \$(LIBS1) \$(LDFLAGS) ¥
	\$(ORALIB)
	ar rv \$(PROGRAM) \$(PROGRAM1)
	rm -f \$(PROGRAM1)
	echo "done"

- Oracle 10g R10.1.0 または R10.2.0 を使用している場合

ファイル名：/opt/DABroker/sample/oramake7101

表 3-28 Oracle 10g サンプルメイクファイル

番号	定義文
1	PROGRAM = libda07101.a
1	PROGRAM1 = libda07101.o
2	LIBS = -L /opt/DABroker/lib -ldaccommon
3	LIBS1 = -lpthreads -bglink:/lib/glink.o -bM:SRE -bexpall ¥
3	-bnoentry -lm -lc_r
3	ORALIB = -L \$(ORACLE_HOME)/lib -lclntsh
3	LDFLAGS = -G
4	OBJS = /opt/DABroker/obj/dacdcntl.o ¥
4	/opt/DABroker/obj/da07101.a
5	all: \$(PROGRAM)
5	\$(PROGRAM): \$(OBJS)
	echo "Linking \$(PROGRAM) ..."
	ld -o \$(PROGRAM1) \$(OBJS) \$(LIBS) \$(LIBS1) \$(LDFLAGS) ¥
	\$(ORALIB)
	ar rv \$(PROGRAM) \$(PROGRAM1)
	rm -f \$(PROGRAM1)

番号	定義文
	echo "done"

- Oracle 11g R11.1.0 または R11.2.0 を使用している場合
ファイル名：/opt/DABroker/sample/oramake7111

表 3-29 Oracle 11g サンプルメイクファイル

番号	定義文
1	PROGRAM = libda08111.a
1	PROGRAM1 = libda08111.o
2	LIBS = -L /opt/DABroker/lib -ldaccommon
3	LIBS1 = -lpthreads -bglink:/lib/glink.o -bM:SRE -bexpall ¥
3	-bnoentry -lm -lc_r
3	ORALIB = -L \$(ORACLE_HOME)/lib -lclntsh
3	LDFLAGS = -G
4	OBJS = /opt/DABroker/obj/dacd8111.o ¥
4	/opt/DABroker/obj/da08111.a
5	all: \$(PROGRAM)
5	\$(PROGRAM): \$(OBJS)
	echo "Linking \$(PROGRAM) ..."
	ld -o \$(PROGRAM1) \$(OBJS) \$(LIBS) \$(LIBS1) \$(LDFLAGS) ¥
	\$(ORALIB)
	ar rv \$(PROGRAM) \$(PROGRAM1)
	rm -f \$(PROGRAM1)
	echo "done"

- DABroker for Java のデータベース種別に Oracle8i を指定して、Oracle8i R8.1.7 を使用している場合

ファイル名：/opt/DABroker/sample/oramake8817

表 3-30 Oracle8i R8.1.7 サンプルメイクファイル (DABroker for Java のデータベース種別で Oracle8i 指定時)

番号	定義文
1	PROGRAM = libda08817.a
1	PROGRAM1 = libda08817.o
2	LIBS = -L /opt/DABroker/lib -ldaccommon
3	LIBS1 = -lpthreads -bglink:/lib/glink.o -bM:SRE -bexpall ¥
3	-bnoentry -lm -lc_r
3	ORALIB = -L \$(ORACLE_HOME)/lib32 -lclntsh
3	LDFLAGS = -G
4	OBJS = /opt/DABroker/obj/dacd8817.o ¥
4	/opt/DABroker/obj/da08817.a
5	all: \$(PROGRAM)

番号	定義文
5	<pre>\$(PROGRAM): \$(OBJS) echo "Linking \$(PROGRAM) ..." ld -o \$(PROGRAM1) \$(OBJS) \$(LIBS) \$(LIBS1) \$(LDFLAGS) ¥ \$(ORALIB) ar rv \$(PROGRAM) \$(PROGRAM1) rm -f \$(PROGRAM1) echo "done"</pre>

- DABroker for Java のデータベース種別に Oracle9i を指定して、Oracle9i R9.2.0 を使用している場合

ファイル名：/opt/DABroker/sample/oramake8920

表 3-31 Oracle9i R9.2.0 サンプルメイクファイル (DABroker for Java のデータベース種別で Oracle8i 指定時)

番号	定義文
1	PROGRAM = libda08920.a
1	PROGRAM1 = libda08920.o
2	LIBS = -L /opt/DABroker/lib -ldaccommon
3	LIBS1 = -lpthreads -bglink:/lib/glink.o -bM:SRE -bexpall ¥
3	-bnoentry -lm -lc_r
3	ORALIB = -L \$(ORACLE_HOME)/lib32 -lclntsh
3	LDFLAGS = -G
4	OBJS = /opt/DABroker/obj/dacd8920.o ¥
4	/opt/DABroker/obj/da08920.a
5	all: \$(PROGRAM)
5	<pre>\$(PROGRAM): \$(OBJS) echo "Linking \$(PROGRAM) ..." ld -o \$(PROGRAM1) \$(OBJS) \$(LIBS) \$(LIBS1) \$(LDFLAGS) ¥ \$(ORALIB) ar rv \$(PROGRAM) \$(PROGRAM1) rm -f \$(PROGRAM1) echo "done"</pre>

- DABroker のデータベース種別に ORACLE8i を指定して、Oracle10g R10.1.0 または R10.2.0 を使用している場合

ファイル名：/opt/DABroker/sample/oramake8101

表 3-32 Oracle10g サンプルメイクファイル (DABroker のデータベース種別で ORACLE8i 指定時)

番号	定義文
1	PROGRAM = libda08101.a
1	PROGRAM1 = libda08101.o

番号	定義文
2	LIBS = -L /opt/DABroker/lib -ldaccommon
3	LIBS1 = -lthreads -bglink:/lib/glink.o -bM:SRE -bexpall ¥
3	-bnoentry -lm -lc_r
3	ORALIB = -L \$(ORACLE_HOME)/lib32 -lclntsh
3	LDFLAGS = -G
4	OBJS = /opt/DABroker/obj/dacd8101.o ¥
4	/opt/DABroker/obj/da08101.a
5	all: \$(PROGRAM)
5	\$(PROGRAM): \$(OBJS)
	echo "Linking \$(PROGRAM) ..."
	ld -o \$(PROGRAM1) \$(OBJS) \$(LIBS) \$(LIBS1) \$(LDFLAGS) ¥
	\$(ORALIB)
	ar rv \$(PROGRAM) \$(PROGRAM1)
	rm -f \$(PROGRAM1)echo "done"

- DABroker for Java のデータベース種別に Oracle8i を指定して、Oracle 11g R11.1.0 または R11.2.0 を使用している場合

ファイル名：/opt/DABroker/sample/oramake8111

表 3-33 Oracle 11g サンプルメイクファイル (DABroker for Java のデータベース種別に Oracle8i を指定時)

番号	定義文
1	PROGRAM = libda08111.a
1	PROGRAM1 = libda08111.o
2	LIBS = -L /opt/DABroker/lib -ldaccommon
3	LIBS1 = -lthreads -bglink:/lib/glink.o -bM:SRE -bexpall ¥
3	-bnoentry -lm -lc_r
3	ORALIB = -L \$(ORACLE_HOME)/lib -lclntsh
3	LDFLAGS = -G
4	OBJS = /opt/DABroker/obj/dacd8111.o ¥
4	/opt/DABroker/obj/da08111.a
5	all: \$(PROGRAM)
5	\$(PROGRAM): \$(OBJS)
	echo "Linking \$(PROGRAM) ..."
	ld -o \$(PROGRAM1) \$(OBJS) \$(LIBS) \$(LIBS1) \$(LDFLAGS) ¥
	\$(ORALIB)
	ar rv \$(PROGRAM) \$(PROGRAM1)
	rm -f \$(PROGRAM1)
	echo "done"

3. DABroker ORACLE アクセスライブラリの再作成

サンプルメイクファイルを修正後、次の環境変数を入力し、make コマンドを実行してください。

(メイクファイル実行手順 B シェルを使用の場合)

ORACLE_HOME = ORACLE HOME ディレクトリ

export ORACLE_HOME

DABDIR =/opt/DABroker

export DABDIR

- Oracle8i R8.1.7 を使用している場合
make -f oramake7817 (修正後のメイクファイルを実行する)
- Oracle9i R9.2.0 を使用している場合
make -f oramake7920 (修正後のメイクファイルを実行する)
- Oracle10g R10.1.0 または R10.2.0 を使用している場合
make -f oramake7101 (修正後のメイクファイルを実行する)
- Oracle 11g R11.1.0 または R11.2.0 を使用している場合
make -f Oramake7111 (修正後のメイクファイルを実行する)
- DABroker for Java のデータベース種別に Oracle8i を指定して、Oracle8i R8.1.7 を使用している場合
make -f orameke8817 (修正後のメイクファイルを実行する)
- DABroker for Java のデータベース種別に Oracle8i を指定して、Oracle9i R9.2.0 を使用している場合
make -f orameke8920 (修正後のメイクファイルを実行する)
- DABroker のデータベース種別に ORACLE8i を指定して、Oracle10g R10.1.0 または R10.2.0 を使用している場合
make -f orameke8101 (修正後のメイクファイルを実行する)
- DABroker for Java のデータベース種別に Oracle8i を指定して、Oracle 11g R11.1.0 または R11.2.0 を使用している場合
make -f Oramake8111 (修正後のメイクファイルを実行する)

4. 再作成後の ORACLE アクセスライブラリを DABroker のライブラリ格納用ディレクトリにコピーする

make コマンドを実行したディレクトリ環境下で再作成した ORACLE アクセスライブラリを次のようにコピーしてください。

再作成した ORACLE アクセスライブラリは、make コマンドを実行したカレントディレクトリに作成されます。

- Oracle8i R8.1.7 を使用している場合
cp libdaO7817.a /opt/DABroker/lib/libdaO7817.a
- Oracle9i R9.2.0 を使用している場合
cp libdaO7920.a /opt/DABroker/lib/libdaO7920.a
- Oracle10g R10.1.0 または R10.2.0 を使用している場合

```
cp libdaO7101.a /opt/DABroker/lib/libdaO7101.a
```

- Oracle 11g R11.1.0 または R11.2.0 を使用している場合

```
cp libdaO7111.sl /opt/DABroker/lib/libdaO7111.sl
```

- DABroker for Java のデータベース種別に Oracle8i を指定して、Oracle8i R8.1.7 を使用している場合

```
cp libdaO8817.a /opt/DABroker/lib/libdaO8817.a
```

- DABroker for Java のデータベース種別に Oracle8i を指定して、Oracle9i R9.2.0 を使用している場合

```
cp libdaO8920.a /opt/DABroker/lib/libdaO8920.a
```

- DABroker のデータベース種別に ORACLE8i を指定して、Oracle10g R10.1.0 または R10.2.0 を使用している場合

```
cp libdaO8101.a /opt/DABroker/lib/libdaO8101.a
```

- DABroker for Java のデータベース種別に Oracle8i を指定して、Oracle 11g R11.1.0 または R11.2.0 を使用している場合

```
cp libdaO8111.sl /opt/DABroker/lib/libdaO8111.sl
```

5. DABroker の再セットアップ

作成した ORACLE アクセスライブラリを /opt/DABroker/lib にコピーした後、再度、DABroker のセットアップコマンドを実行してください。

3.11.3 Solaris の場合

Solaris 使用時の ORACLE アクセスライブラリ再作成方法について説明します。

(1) 適用範囲

ORACLE アクセスライブラリを作成する機能は、次の ORACLE システムを運用している場合に適用してください。

- Oracle8 R8.0.5
- Oracle8 R8.0.6
- Oracle8i R8.1.5
- Oracle8i R8.1.6
- Oracle8i R8.1.7
- Oracle9i R9.0.1

また、ORACLE アクセスライブラリを作成する機能を使用するには、Sun WorkShop C コンパイラが必要です。

(2) ORACLE アクセスライブラリ再作成の手順

DABroker では、ORACLE が提供している各バージョンの OCI サンプルメイクファイルに DABroker 専用のリンケージラインを埋め込んで提供しています。使用している ORACLE のバージョンに従い、次の手順で実行してください。

次の手順は、必ず、運用ディレクトリが/opt/DABroker の環境で実行してください。

DABroker の運用ディレクトリを/opt/DABroker 以外に設定している場合は、いったん、DABroker のセットアップコマンドで運用ディレクトリを/opt/DABroker に戻してから実行してください。

(a) Oracle8 8.0.5 を使用している場合

- 実行手順

次の環境変数を入力して、実行してください。

(メイクファイル実行手順 標準シェルを使用する場合)

1. 環境変数の設定

```
$ ORACLE_HOME =<ORACLE インストールディレクトリ>
```

```
$ export ORACLE_HOME
```

```
$ DABLIB_HOME =<DABroker 運用ディレクトリ>
```

```
$ export DABLIB_HOME
```

2. ORACLE アクセスライブラリ作成の実行

カレントディレクトリはシェルファイルがあるところへ移動してください。

```
$ sh ./dab_rdbms7805.sh
```

これによって、dab_demo_rdbms7805.mk が実行されます。

3. 正常終了した場合

再作成用シェルを実行したディレクトリ環境下で、作成した ORACLE アクセスライブラリを次のようにコピーしてください。作成した ORACLE アクセスライブラリは、再作成用シェルを実行したカレントディレクトリに作成されます。

```
$ cp libdaO7805.so /opt/DABroker/lib/libdaO7805.so
```

4. 異常終了した場合

「●注意事項」を参照し、メイクファイルを再作成して実行してください。

- DABroker 専用のリンケージライン

DABroker の提供する dab_demo_rdbms7805.mk は、当製品開発時点の OCI サンプルメイクファイル (\$ORACLE_HOME/rdbms/demo/demo_rdbms.mk) の最終行に次のラインを追加しています。

```
#####
dab_link:
$(ECHO) $(CC) $(LDFLAGS) -o $(EXE) ¥
$(DABLIB_HOME)/obj/dacdcntl.o $(DABLIB_HOME)/obj/da07805.a ¥
$(OCISHAREDLIBS) ¥
-lposix4 -ldl -lc -lnsl -lsocket -lthread -G
#####
```

- 注意事項

ORACLE が提供している各バージョンの OCI サンプルメイクファイル (\$ORACLE_HOME/rdbms/demo/demo_rdbms.mk) は変更される場合があります。このため、DABroker が提供している dab_demo_rdbms7805.mk と比較してください。もし、DABroker 専用のリンケージライン以外が異なっていた場合、次の手順で dab_demo_rdbms7805.mk を再作成してください。

```
$ cp $ORACLE_HOME/rdbms/demo/demo_rdbms.mk dab_demo_rdbms7805.mk
```

コピー後、上記の DABroker 専用のリンケージラインを最終行に追加してください。

(b) Oracle8 8.0.6 を使用している場合

- 実行手順

次の環境変数を入力して、実行してください。

(メイクファイル実行手順 標準シェルを使用する場合)

1. 環境変数の設定

```
$ ORACLE_HOME =<ORACLE インストールディレクトリ>
```

```
$ export ORACLE_HOME
```

```
$ DABLIB_HOME =<DABroker 運用ディレクトリ>
```

```
$ export DABLIB_HOME
```

2. ORACLE アクセスライブラリ作成の実行

カレントディレクトリはシェルファイルがあるところへ移動してください。

```
$ sh ./dab_rdbms7806.sh
```

これによって、dab_demo_rdbms7806.mk が実行されます。

3. 正常終了した場合

再作成用シェルを実行したディレクトリ環境下で、作成した ORACLE アクセスライブラリを次のようにコピーしてください。作成した ORACLE アクセスライブラリは、再作成用シェルを実行したカレントディレクトリに作成されます。

```
$ cp libdaO7806.so /opt/DABroker/lib/libdaO7806.so
```

4. 異常終了した場合

「●注意事項」を参照し、メイクファイルを再作成して実行してください。

- DABroker 専用のリンケージライン

DABroker の提供する dab_demo_rdbms7806.mk は、当製品開発時点の OCI サンプルメイクファイル (\$ORACLE_HOME/rdbms/demo/demo_rdbms.mk) の最終行に次のラインを追加しています。

```
#####
dab_link:
$(ECHO$) ld $(LDFLAGS) -o $(EXE) ¥
$(DABLIB_HOME)/obj/dacdcntl.o $(DABLIB_HOME)/obj/da07806.a ¥
$(OCISHAREDLIBS) ¥
-lposix4 -ldl -lc -lnsl -lsocket -lthread -G
#####
```

- 注意事項

ORACLE が提供している各バージョンの OCI サンプルメイクファイル (\$ORACLE_HOME/rdbms/demo/demo_rdbms.mk) は変更される場合があります。このため、DABroker が提供している dab_demo_rdbms7806.mk と比較してください。もし、DABroker 専用のリンケージライン以外が異なっていた場合、次の手順で dab_demo_rdbms7806.mk を再作成してください。

```
$ cp $ORACLE_HOME/rdbms/demo/demo_rdbms.mk dab_demo_rdbms7806.mk
```

コピー後、上記の DABroker 専用のリンケージラインを最終行に追加してください。

(c) Oracle8i 8.1.5 を使用している場合

- 実行手順

次の環境変数を入力して、実行してください。

(メイクファイル実行手順 標準シェルを使用する場合)

1. 環境変数の設定

```
$ ORACLE_HOME =<ORACLE インストールディレクトリ>
$ export ORACLE_HOME
$ DABLIB_HOME =<DABroker 運用ディレクトリ>
$ export DABLIB_HOME
```

2. ORACLE アクセスライブラリ作成の実行

カレントディレクトリはシェルファイルがあるところへ移動してください。

```
$ sh ./dab_rdbms7815.sh
```

これによって、dab_demo_rdbms7815.mk が実行されます。

3. 正常終了した場合

再作成用シェルを実行したディレクトリ環境下で、作成した ORACLE アクセスライブラリを次のようにコピーしてください。作成した ORACLE アクセスライブラリは、再作成用シェルを実行したカレントディレクトリに作成されます。

```
$ cp libdaO7815.so /opt/DABroker/lib/libdaO7815.so
```

4. 異常終了した場合

「●注意事項」を参照し、メイクファイルを再作成して実行してください。

- DABroker 専用のリンケージライン

DABroker の提供する dab_demo_rdbms7815.mk は、当製品開発時点の OCI サンプルメイクファイル (\$ORACLE_HOME/rdbms/demo/demo_rdbms.mk) の最終行に次のラインを追加しています。

```
#####
dab_link:
$(ECHO) $(CC) $(LDFLAGS) -o $(EXE) ¥
$(DABLIB_HOME)/obj/dacdcntl.o $(DABLIB_HOME)/obj/da07815.a ¥
$(OCISHAREDLIBS) ¥
-lposix4 -ldl -lc -lnsl -lsocket -lthread -G
#####
```

- 注意事項

ORACLE が提供している各バージョンの OCI サンプルメイクファイル (\$ORACLE_HOME/rdbms/demo/demo_rdbms.mk) は変更される場合があります。このため、DABroker が提供している dab_demo_rdbms7815.mk と比較してください。もし、DABroker 専用のリンケージライン以外が異なっていた場合、次の手順で dab_demo_rdbms7815.mk を再作成してください。

```
$ cp $ORACLE_HOME/rdbms/demo/demo_rdbms.mk dab_demo_rdbms7815.mk
```

コピー後、上記の DABroker 専用のリンケージラインを最終行に追加してください。

(d) Oracle8i 8.1.6 を使用している場合

- 実行手順

次の環境変数を入力して、実行してください。

(メイクファイル実行手順 標準シェルを使用する場合)

1. 環境変数の設定

```
$ ORACLE_HOME =<ORACLE インストールディレクトリ>
```

```
$ export ORACLE_HOME
```

```
$ DABLIB_HOME =<DABroker 運用ディレクトリ>
```

```
$ export DABLIB_HOME
```

2. ORACLE アクセスライブラリ作成の実行

カレントディレクトリはシェルファイルがあるところへ移動してください。

```
$ sh ./dab_rdbms7816.sh
```

これによって、dab_demo_rdbms7816.mk が実行されます。

3. 正常終了した場合

再作成用シェルを実行したディレクトリ環境下で、作成した ORACLE アクセスライブラリを次のようにコピーしてください。作成した ORACLE アクセスライブラリは、再作成用シェルを実行したカレントディレクトリに作成されます。

```
$ cp libdaO7816.so /opt/DABroker/lib/libdaO7816.so
```

4. 異常終了した場合

「●注意事項」を参照し、メイクファイルを再作成して実行してください。

- DABroker 専用のリンケージライン

DABroker の提供する dab_demo_rdbms7816.mk は、当製品開発時点の OCI サンプルメイクファイル (\$ORACLE_HOME/rdbms/demo/demo_rdbms.mk) の最終行に次のラインを追加しています。

```
#####
dab_link:
$(ECHO) ld $(LD_FLAGS) -o $(EXE) ¥
$(DABLIB_HOME)/obj/dacdcntl.o $(DABLIB_HOME)/obj/da07816.a ¥
$(OCISHAREDLIBS) ¥
-lposix4 -ldl -lc -lnsl -lsocket -lthread -G
#####
```

- 注意事項

ORACLE が提供している各バージョンの OCI サンプルメイクファイル (\$ORACLE_HOME/rdbms/demo/demo_rdbms.mk) は変更される場合があります。このため、DABroker が提供している dab_demo_rdbms7816.mk と比較してください。もし、DABroker 専用のリンケージライン以外が異なっていた場合、次の手順で dab_demo_rdbms7816.mk を再作成してください。

```
$ cp $ORACLE_HOME/rdbms/demo/demo_rdbms.mk dab_demo_rdbms7816.mk
```

コピー後、上記の DABroker 専用のリンケージラインを最終行に追加してください。

(e) Oracle8i 8.1.7 を使用している場合

- 実行手順

次の環境変数を入力して、実行してください。

(メイクファイル実行手順 標準シェルを使用する場合)

1. 環境変数の設定

```
$ ORACLE_HOME =<ORACLE インストールディレクトリ>
$ export ORACLE_HOME
$ DABLIB_HOME =<DABroker 運用ディレクトリ>
$ export DABLIB_HOME
```

2. ORACLE アクセスライブラリ作成の実行

カレントディレクトリはシェルファイルがあるところへ移動してください。

```
$ sh ./dab_rdbms7817.sh
```

これによって、dab_demo_rdbms7817.mk が実行されます。

3. 正常終了した場合

再作成用シェルを実行したディレクトリ環境下で、作成した ORACLE アクセスライブラリを次のようにコピーしてください。作成した ORACLE アクセスライブラリは、再作成用シェルを実行したカレントディレクトリに作成されます。

```
$ cp libdaO7817.so /opt/DABroker/lib/libdaO7817.so
```


4. 異常終了した場合

メイクファイルを再作成し、実行してください。

Oracle8i 8.1.7 の dab_demo_rdbms7817.mk ファイルは、ORACLE の提供するサンプルメイクファイルを使用していません。このため、表 3-34 を参照して、修正が必要な場合、修正してください。

なお、表中の番号は、次の内容を示しています。

1：DABroker アクセスライブラリ名称

修正しないでください。

2：DABroker 共用ライブラリ

修正しないでください。

3：ORACLE クライアントライブラリ、及び OS 提供ライブラリ

DABroker の ORACLE アクセスライブラリは、ORACLE の OCI プログラムです。ORACLE のマニュアルを参照し、OCI プログラム作成時に必要な ORACLE クライアントライブラリ、及び OS 提供ライブラリを確認の上、使用している環境に合わせて修正してください。

4：DABroker 共用オブジェクトファイル

修正しないでください。

表 3-34 Oracle8i R8.1.7 サンプルメイクファイル

番号	定義文
1	PROGRAM = libda07817.so
2	LIBS = -L \$(DABLIB_HOME)/lib -ldaccommon
3	ORALIB = -L\$(ORACLE_HOME)/lib -L\$(ORACLE_HOME)/rdbms/lib -lclntsh
3	LDFLAGS = -lposix4 -lthread -ldl -lc -lnsl -lsocket -G
4	OBJS = \$(DABLIB_HOME)/obj/dacdcntl.o \$(DABLIB_HOME)/obj/da07817.a
4	all: \$(PROGRAM)
4	\$(PROGRAM): \$(OBJS) echo "Linking \$(PROGRAM) ..." ld -o \$(PROGRAM) \$(OBJS) \$(LIBS) \$(ORALIB) \$(LDFLAGS) echo "done"

(f) Oracle9i 9.0.1 を使用している場合

• 実行手順

次の環境変数を入力して、実行してください。

(メイクファイル実行手順 標準シェルを使用する場合)

1. 環境変数の設定

```
$ ORACLE_HOME =<ORACLE インストールディレクトリ>
```

```
$ export ORACLE_HOME
```



```
$ DABLIB_HOME =<DABroker 運用ディレクトリ>
$ export DABLIB_HOME
```

2. ORACLE アクセスライブラリ作成の実行

カレントディレクトリはシェルフファイルがあるところへ移動してください。

```
$ sh ./dab_rdbms7901.sh
```

これによって、dab_demo_rdbms7901.mk が実行されます。

3. 正常終了した場合

再作成用シェルを実行したディレクトリ環境下で、作成した ORACLE アクセスライブラリを次のようにコピーしてください。作成した ORACLE アクセスライブラリは、再作成用シェルを実行したカレントディレクトリに作成されます。

```
$ cp libdaO7901.so /opt/DABroker/lib/libdaO7901.so
```

4. 異常終了した場合

メイクファイルを再作成し、実行してください。

Oracle9i 9.0.1 の dab_demo_rdbms7901.mk ファイルは、ORACLE の提供するサンプルメイクファイルを使用していません。このため、表 3-35 を参照して、修正が必要な場合、修正してください。

なお、表中の番号は、次の内容を示しています。

1：DABroker アクセスライブラリ名称

修正しないでください。

2：DABroker 共用ライブラリ

修正しないでください。

3：ORACLE クライアントライブラリ、及び OS 提供ライブラリ

DABroker の ORACLE アクセスライブラリは、ORACLE の OCI プログラムです。ORACLE のマニュアルを参照し、OCI プログラム作成時に必要な ORACLE クライアントライブラリ、及び OS 提供ライブラリを確認の上、使用している環境に合わせて修正してください。

4：DABroker 共用オブジェクトファイル

修正しないでください。

表 3-35 Oracle9i R9.0.1 サンプルメイクファイル

番号	定義文
1	PROGRAM = libda07901.so
2	LIBS = -L \$(DABLIB_HOME)/lib -ldaccommon
3	ORALIB = -L\$(ORACLE_HOME)/lib -L\$(ORACLE_HOME)/rdbms/lib -lclntsh
3	LDFLAGS = -lposix4 -lthread -ldl -lc -lnsl -lsocket -G
4	OBJS = \$(DABLIB_HOME)/obj/dacdcntl.o \$(DABLIB_HOME)/obj/da07901.a
4	all: \$(PROGRAM)

番号	定義文
4	<pre>\$(PROGRAM): \$(OBJS) echo "Linking \$(PROGRAM) ..." ld -o \$(PROGRAM) \$(OBJS) \$(LIBS) \$(ORALIB) \$(LDFLAGS) echo "done"</pre>

3.11.4 Red Hat Linux の場合

(1) 適用範囲

ORACLE アクセスライブラリを作成する機能は、次の ORACLE システムを運用している場合に適用してください。

- DABroker を動作させたとき、次のメッセージで ORACLE の関数名、またはライブラリ名が出力された場合

DABroker 出力メッセージ

KFDB005MM-E システムコールでエラーが発生しました。

[dlopen(/opt/DABroker/lib/libdaO7101.sl*) ermo=ld.sl.1 :

重大なエラー：再配置エラー：ファイル /opt/DABroker/lib/libdaO7101.sl*:

シンボル (ORACLE の関数)：参照されたシンボルがありません]

注※

- Oracle10g R10.1.0 または R10.2.0 を使用している場合：libdaO7101.sl と表示されます。
- Oracle11g R11.1.0 または R11.2.0 を使用している場合：libdaO7111.sl と表示されます。
- DABroker for Java のデータベース種別に Oracle8i を指定して、Oracle10g R10.1.0 または R10.2.0 を使用している場合：libdaO8101.sl と表示されます。
- DABroker for Java のデータベース種別に Oracle8i を指定して、Oracle11g R11.1.0 または R11.2.0 を使用している場合：libdaO8111.sl と表示されます。

(2) ORACLE アクセスライブラリの作成手順

ORACLE アクセスライブラリを再作成する手順を次に示します。

次の手順は、必ず、運用ディレクトリが/opt/DABroker の環境で実行してください。

DABroker の運用ディレクトリを/opt/DABroker 以外に設定している場合は、いったん、DABroker のセットアップコマンドで運用ディレクトリを/opt/DABroker に戻してから実行してください。

1. DABroker 既存ライブラリのバックアップ

DABroker の既存ライブラリのバックアップを次のように取得してください。

cd /opt/DABroker/lib

- Oracle 10g R10.1.0 または R10.2.0 を使用している場合
mv libdaO7101.a libdaO7101.a_bk
- Oracle 11g R11.1.0 または R11.2.0 を使用している場合
mv libdaO7111a libdaO7111.a_bk
- DABroker for Java のデータベース種別に Oracle8i を指定して、Oracle 10g R10.1.0 または R10.2.0 を使用している場合
mv libdaO8101.a libdaO8101.a_bk
- DABroker for Java のデータベース種別に Oracle8i を指定して、Oracle 11g R11.1.0 または R11.2.0 を使用している場合
mv libdaO8111.a libdaO8111.a_bk

2. 提供サンプルメイクファイルの修正

DABroker では、表 3-36~39 に示す ORACLE アクセスライブラリのサンプルメイクファイルを提供しています。使用している環境に合わせて、修正して使用してください。

なお、表中の番号は、次の内容を示しています。

1：DABroker のアクセスライブラリ名称

修正しないでください。

2：DABroker 共用ライブラリ

修正しないでください。

3：ORACLE クライアントライブラリ、および OS 提供ライブラリ

DABroker の ORACLE アクセスライブラリは、ORACLE の OCI プログラムです。ORACLE のマニュアルを参照し、OCI プログラム作成時に必要な ORACLE クライアントライブラリ、および OS 提供ライブラリを確認の上、使用している環境に合わせて修正してください。

4：DABroker 共用オブジェクトファイル

修正しないでください。

5：リンクージパラメタ

修正しないでください。

- Oracle 10g R10.1.0 または R10.2.0 を使用している場合
ファイル名：/opt/DABroker/sample/oramake7101

表 3-36 Oracle 10g サンプルメイクファイル

番号	定義文
1	PROGRAM = libda07101.so
2	LIBS = -L \$(DABDIR)/lib -ldaccommon
3	ORALIB = -L\$(ORACLE_HOME)/lib -L\$(ORACLE_HOME)/rdbms/lib -lcintsh

番号	定義文
3	LDFLAGS = -ldl -lnsl -lpthread -G
4	OBJS = \$(DABDIR)/obj/dacdcntl.o \$(DABDIR)/obj/da07101.a
5	all: \$(PROGRAM)
5	\$(PROGRAM): \$(OBJS) echo "Linking \$(PROGRAM) ..." ld -o \$(PROGRAM) \$(OBJS) \$(LIBS) \$(ORALIB) \$(LDFLAGS) echo "done"

- Oracle 11g R11.1.0 または R11.2.0 を使用している場合
ファイル名：/opt/DABroker/sample/oramake7111

表 3-37 Oracle 11g サンプルメイクファイル

番号	定義文
1	PROGRAM = libda07111.so
2	LIBS = -L \$(DABDIR)/lib -ldaccommon
3	ORALIB = -L\$(ORACLE_HOME)/lib -L\$(ORACLE_HOME)/rdbms/lib -lcIntsh
3	LDFLAGS = -ldl -lnsl -lpthread -G
4	OBJS = \$(DABDIR)/obj/dacdcntl.o \$(DABDIR)/obj/da07111.a
5	all: \$(PROGRAM)
5	\$(PROGRAM): \$(OBJS) echo "Linking \$(PROGRAM) ..." ld -o \$(PROGRAM) \$(OBJS) \$(LIBS) \$(ORALIB) \$(LDFLAGS) echo "done"

- DABroker for Java のデータベース種別に Oracle8i を指定して、Oracle 10g R10.1.0 または R10.2.0 を使用している場合
ファイル名：/opt/DABroker/sample/oramake8101

表 3-38 Oracle 10g サンプルメイクファイル(DABroker for Java のデータベース種別に Oracle8i を指定時)

番号	定義文
1	PROGRAM = libda08101.so
2	LIBS = -L \$(DABDIR)/lib -ldaccommon
3	ORALIB = -L\$(ORACLE_HOME)/lib -L\$(ORACLE_HOME)/rdbms/lib -lcIntsh
3	LDFLAGS = -ldl -lnsl -lpthread -G
4	OBJS = \$(DABDIR)/obj/dacd8101.o \$(DABDIR)/obj/da08101.a

番号	定義文
5	all: \$(PROGRAM)
5	\$(PROGRAM): \$(OBJS) echo "Linking \$(PROGRAM) ..." ld -o \$(PROGRAM) \$(OBJS) \$(LIBS) \$(ORALIB) \$(LDFLAGS) echo "done"

- DABroker for Java のデータベース種別に Oracle8i を指定して、Oracle 11g R11.1.0 または R11.2.0 を使用している場合

ファイル名：/opt/DABroker/sample/oramake8111

表 3-39 Oracle 11g サンプルメイクファイル(DABroker for Java のデータベース種別に Oracle8i を指定時)

番号	定義文
1	PROGRAM = libda08111.so
2	LIBS = -L \$(DABDIR)/lib -ldaccommon
3	ORALIB = -L\$(ORACLE_HOME)/lib -L\$(ORACLE_HOME)/rdbms/lib -lcintsh
3	LDFLAGS = -ldl -lnsl -lpthread -G
4	OBJS = \$(DABDIR)/obj/dacd8111.o \$(DABDIR)/obj/da08111.a
5	all: \$(PROGRAM)
5	\$(PROGRAM): \$(OBJS) echo "Linking \$(PROGRAM) ..." ld -o \$(PROGRAM) \$(OBJS) \$(LIBS) \$(ORALIB) \$(LDFLAGS) echo "done"

3. DABroker ORACLE アクセスライブラリの再作成

サンプルメイクファイルを修正後、次の環境変数を入力し、make コマンドを実行してください。

(メイクファイル実行手順 B シェルを使用の場合)

ORACLE_HOME = ORACLE HOME ディレクトリ

export ORACLE_HOME

DABDIR = /opt/DABroker

export DABDIR

- Oracle 10g R10.1.0 または R10.2.0 を使用している場合
make -f Oramake7101 (修正後のメイクファイルを実行する)
- Oracle 11g R11.1.0 または R11.2.0 を使用している場合
make -f Oramake7111 (修正後のメイクファイルを実行する)

- DABroker for Java のデータベース種別に Oracle8i を指定して、Oracle 10g R10.1.0 または R10.2.0 を使用している場合
make -f Oramake8101 (修正後のメイクファイルを実行する)
- DABroker for Java のデータベース種別に Oracle8i を指定して、Oracle 11g R11.1.0 または R11.2.0 を使用している場合
make -f Oramake8111 (修正後のメイクファイルを実行する)

4. 再作成の ORACLE アクセスライブラリを DABroker のライブラリ格納用ディレクトリにコピーする

make コマンドを実行したディレクトリ環境下で再作成した、ORACLE アクセスライブラリを次のようにコピーしてください。

再作成した ORACLE アクセスライブラリは、make コマンドを実行したカレントディレクトリに作成されます。

- Oracle 10g R10.1.0 または R10.2.0 を使用している場合
cp libdaO7101.sl /opt/DABroker/lib/libdaO7101.sl
- Oracle 11g R11.1.0 または R11.2.0 を使用している場合
cp libdaO7111.sl /opt/DABroker/lib/libdaO7111.sl
- DABroker for Java のデータベース種別に Oracle8i を指定して、Oracle 10g R10.1.0 または R10.2.0 を使用している場合
cp libdaO8101.sl /opt/DABroker/lib/libdaO8101.sl
- DABroker for Java のデータベース種別に Oracle8i を指定して、Oracle 11g R11.1.0 または R11.2.0 を使用している場合
cp libdaO8111.sl /opt/DABroker/lib/libdaO8111.sl

5. DABroker の再セットアップ

作成した ORACLE アクセスライブラリを /opt/DABroker/lib にコピーしたあと、再度、DABroker のセットアップコマンドを実行してください。

3.11.5 Oracle9i 以降で追加されたデータ型への対応状況

Oracle9i, 又は Oracle10g で追加されたデータ型に対する DABroker の対応状況を表 3-40 と表 3-41 に示します。

表 3-40 Oracle9i で追加されたデータ型

データ型	DABroker でのサポート状況
TIMESTAMP(fractional_seconds_precision)	未サポート
TIMESTAMP(fractional_seconds_precision) WITH TIME ZONE	未サポート

データ型	DABroker でのサポート状況
TIMESTAMP(fractional_seconds_precision) WITH LOCAL TIME ZONE	未サポート
INTERVAL YEAR(year_precision) TO MONTH	未サポート
INTERVAL DAY(day_precision) TO SECOND (fractional_seconds_precision)	未サポート

表 3-41 Oracle10g で追加されたデータ型

データ型	DABroker でのサポート状況
BINARY_FLOAT	未サポート
BINARY_DOUBLE	未サポート

4

DABroker の環境設定と運用 (Windows)

この章では、Windows での DABroker の環境設定、及び運用の方法について説明します。

4.1 環境設定の手順

ここでは、DABroker を使用する環境の設定手順について説明します。なお、説明は、次のことを前提にしています。

- DBMS を使用できる環境が既に構築されていること
- 開発アプリケーション、DBPARTNER、又は HITSENER5 を組み込むマシン、DBPARTNER Library、又は XDM/RD 外部表を組み込むマシンをネットワーク上で使用できる環境が既に構築されていること
- RDA Link for Gateway 経由でメインフレーム系データベースにアクセスする場合、RDA Link for Gateway が DABroker を組み込むマシンに組み込まれていること
- Database Connection Server 経由でメインフレーム系データベースにアクセスする場合、Database Connection Server が接続先のメインフレームに組み込まれていること

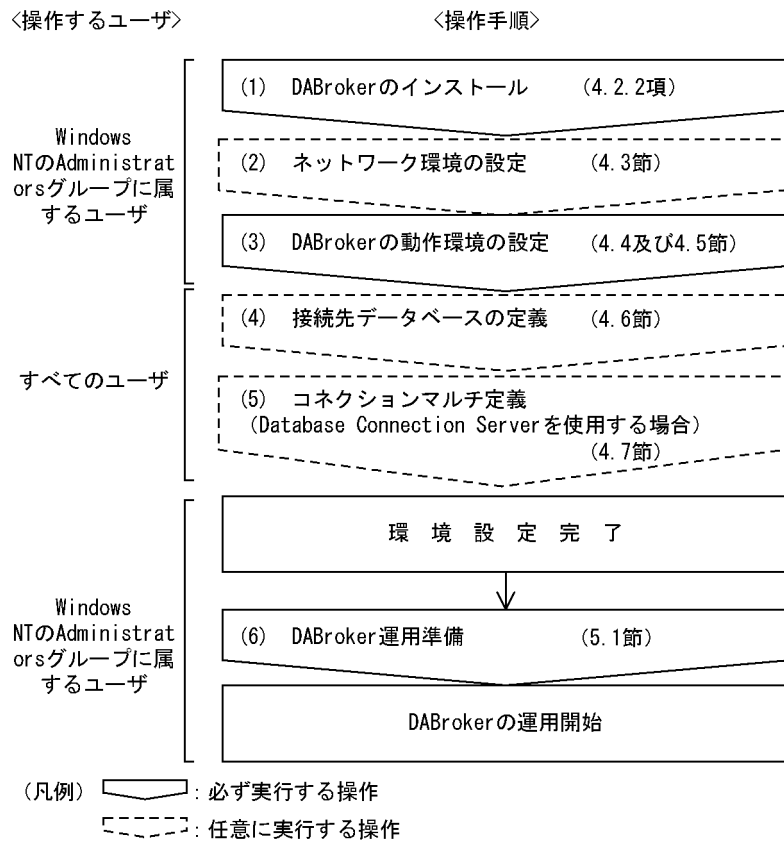
DABroker の環境設定の操作手順を、DABroker 関連製品を使用する場合と DBPARTNER、HITSENER5、DBPARTNER Library、又は XDM/RD 外部表を使用する場合に分けて説明します。

4.1.1 DABroker 関連製品を使用する場合の環境設定

DABroker 関連製品を使用する場合の環境設定の手順を図 4-1 に示します。

図 4-1 では、環境設定完了後、最初に DABroker を運用するまでの操作を示しています。

図 4-1 DABroker 関連製品を使用する場合の環境設定の手順



4.1.2 DBPARTNER, HITSENSER5, DBPARTNER Library, XDM/RD 外部表を使用する場合の環境設定

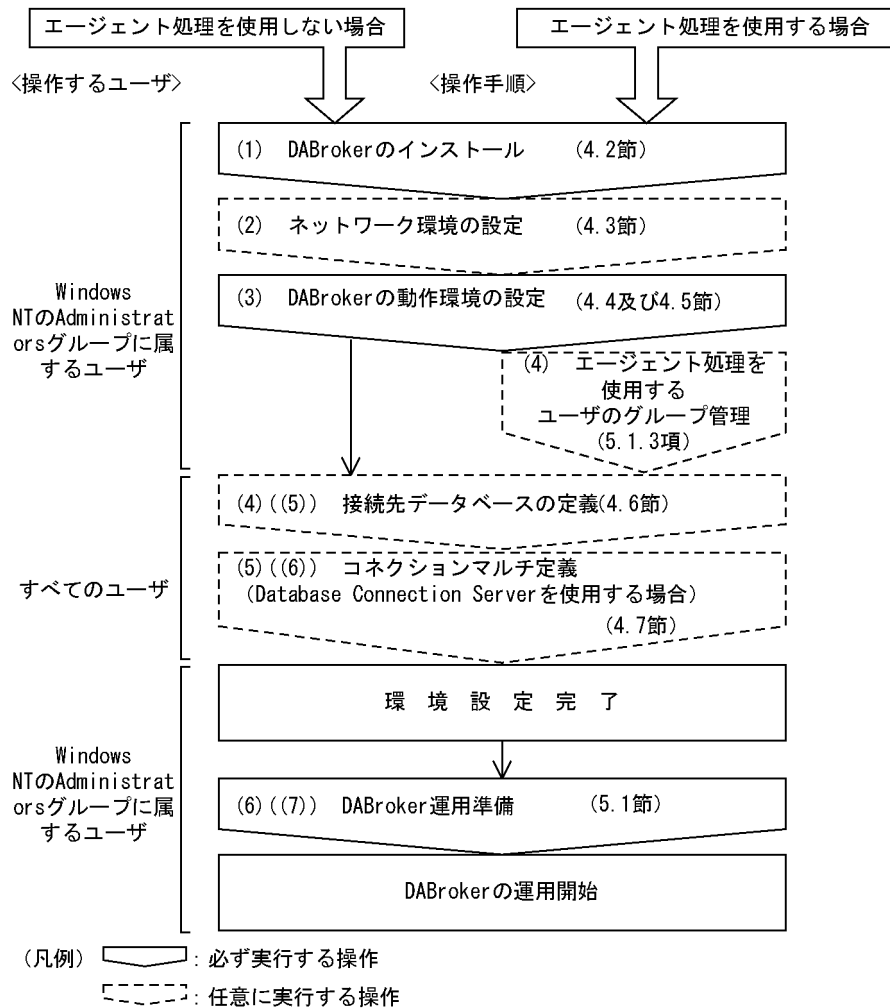
DBPARTNER, HITSENSER5, DBPARTNER Library, 又は XDM/RD 外部表を使用する場合は、図 4-2 に示す手順で DABroker の環境を設定してください。

ただし、(2)の「ネットワーク環境の設定」については、使用する環境によって設定が必要かどうか異なります。

エージェント機能を使用する場合は、(3)の「DABroker 動作環境の設定」でエージェント機能を設定してください。

図 4-2 では、環境設定完了後、最初に DABroker を運用するまでの操作を示しています。

図 4-2 DBPARTNER, HITSENER5, DBPARTNER Library, 又は XDM/RD 外部表を使用する場合の環境設定の手順



4.1.3 環境設定方法

環境設定はDABrokerが提供するユーティリティを使用します。環境設定で使用するユーティリティは、環境設定ユーティリティ、接続先データベース定義ユーティリティ、及びコネクションマルチ定義ユーティリティです。

これらユーティリティの詳細については、「4.4 DABroker 環境設定」, 「4.5 DABroker 環境設定項目」, 「4.6 DABroker 接続先データベース定義」, 及び「4.7 DABroker コネクションマルチ定義(Database Connection Server 経由の DB アクセス定義)」を参照してください。

4.2 DABroker のインストール

ここでは、DABroker のインストール操作について説明します。

4.2.1 DABroker をインストールするマシン

使用するデータベースによって、DABroker をインストールできるマシンが決まります。DABroker をインストールするマシンについて、使用するデータベースごとに説明します。

- **HiRDB, ORACLE などにアクセスする場合**

HiRDB, ORACLE, SQL Anywhere, Adaptive Server Anywhere, 又は SQL Server にアクセスする場合、DABroker は、DBMS のサーバプログラム、又は DBMS のクライアントプログラムがインストールされたマシンにインストールしてください。なお、「1.2 DABroker の使用環境」では、DABroker を DBMS のサーバプログラムがインストールされたマシンにインストールする形態を示しています。

- **RDA Link for Gateway を使ってメインフレーム系データベースにアクセスする場合**

DABroker は、RDA Link for Gateway と同一マシンにインストールしてください。

- **Database Connection Server を使ってメインフレーム系データベースにアクセスする場合**

DABroker は、任意のサーバマシン（Windows で動作しているパーソナルコンピュータ）にインストールしてください。

4.2.2 インストール

DABroker のインストールは、日立総合インストーラを使用して行います。インストールを実行すると、ダイアログボックスが表示されますので、ガイダンスに従ってください。

1. Administrators グループのメンバーであるユーザアカウントでログオンする。

2. 日立総合インストーラを使用して、DABroker のインストールを開始する。

[よろこそ] ダイアログボックスが表示されるので、[次へ] を選択します。

3. ユーザ情報を設定する。

[ユーザの情報] ダイアログボックスでインストールするユーザの個人名と会社名を入力します。

4. インストール先ディレクトリを指定する。

[インストール先のディレクトリ] ダイアログボックスでインストール先ディレクトリを指定します。標準のインストールディレクトリは、次のとおりです。

<システムドライブ>:\Program Files\HITACHI\DABroker

ここにインストールしない場合は、参照ボタンをクリックしてインストール先ディレクトリを指定してください。ただし、ネットワークドライブには DABroker を組み込みません。インストール先ディレクトリがない場合、ディレクトリを作成するかどうかのダイアログボックスが表示されます。

5. プログラムフォルダを選択する。

[プログラムフォルダの選択] 画面でプログラムを追加するプログラムフォルダを選択します。標準のプログラムフォルダは DABroker です。プログラムフォルダの選択はできません。

6. インストールを完了する。

[完了] ボタンをクリックしてインストールを完了します。なお、システムファイルを上書きする場合、即時にシステムを再起動するかを問い合わせます。再起動しなかった場合は、DABroker の動作を保証できませんので、必ず再起動してから使用してください。

7. 環境変数を設定する

DABroker をインストール後、システム環境変数の Path に"DABroker インストールディレクトリ¥lib" を必ず設定してください。設定しなかった場合は、各ユティリティが使用できません。

4.2.3 DABroker のファイル

(1) インストール時に作成されるディレクトリ

DABroker のインストール時に、指定したディレクトリに作成されるディレクトリを表 4-1 に示します。

なお、インストール先に指定したディレクトリをこれ以降、DABroker 運用ディレクトリと呼びます。

表 4-1 DABroker のディレクトリ

ディレクトリ	説明	
DABroker	—	インストール先ディレクトリ
¥bin		実行ファイル格納ディレクトリ
¥lib		ライブラリ格納用ディレクトリ
¥AGT		エージェント処理用ディレクトリ
¥AGTGRP		グループ管理用ディレクトリ
¥msg		メッセージ用ディレクトリ
¥conf		DABroker 動作環境定義ファイル格納ディレクトリ
¥spool		DABroker スプールディレクトリ
¥tmp		DABroker 標準作業ディレクトリ
¥patch_dab		パッチ履歴ディレクトリ

4.3 ネットワーク環境の設定

ここでは、DBPARTNER、HITSENER5、DBPARTNER Library、又は XDM/RD 外部表を使用する場合に、[Windows インストールディレクトリ]¥SYSTEM32¥DRIVERS¥ETC¥SERVICES ファイルに設定する DABroker のサービス名及びポート番号について説明します。

また、Database Connection Server との接続にサービス名を使用する場合の、[Windows インストールディレクトリ]¥SYSTEM32¥DRIVERS¥ETC¥SERVICES ファイルでの設定方法について説明します。

4.3.1 DBPARTNER, HITSENER5, DBPARTNER Library, 又は XDM/RD 外部表を使用する場合

DBPARTNER、HITSENER5、DBPARTNER Library、又は XDM/RD 外部表を使用する場合、[Windows インストールディレクトリ]¥SYSTEM32¥DRIVERS¥ETC¥SERVICES ファイルに、TCP/IP 環境での DABroker のサービス名及びポート番号として、次の記述をファイルの行末に追加してください。

```
DABROKER_SV XXXXX/tcp
```

DABROKER_SV : DABroker のサービス名です。

XXXXX : ポート番号です。

tcp : プロトコル名です。TCP/IP を使用することを表します。

注意事項

- XXXXX には DABroker がオープンするポート番号を指定します。推奨値は 40179 です。オペレーティングシステムによっては推奨値がエフェメラルポートの範囲に当たるので、問題がある場合は別のポート番号を指定してください。このポート番号とクライアント側のポート番号には同じ値を指定します。
- NIS (Network Information Service) を使用している場合、ネットワーク環境を NIS のサーバで設定してください。NIS のサーバを使用している場合、ネットワーク環境の設定は、NIS のサーバでの設定が有効になります。
- クライアント側では、使用する通信ドライバに、DABroker の組み込まれたサーバのホスト名、IP アドレス及びポート番号を登録してください。
- Windows ファイアウォールが有効になっていないか確認し、有効になっている場合は DABroker を使用する前に、DABroker のサービスポートを Windows ファイアウォールの例外に登録してください。

また、[Windows インストールディレクトリ]¥SYSTEM32¥DRIVERS¥ETC¥HOSTS ファイルにはクライアント側の PC 台数分の IP アドレスとホスト名を記述してください。HOSTS ファイルは IP アドレスからホスト名への変換オプションが有効な場合 (DABroker 環境設定項目の DABIPCONVERT に ON を指定した場合) に参照します。HOSTS ファイルについては、Windows のマニュアルを参照してください。

4.3.2 Database Connection Server を経由してメインフレーム系データベースにアクセスする場合

Database Connection Server との接続には、DABroker では、Database Connection Server のポート番号又はサービス名を指定します。ポート番号又はサービス名は、コネクションマルチ定義（「4.7 DABroker コネクションマルチ定義(Database Connection Server 経由の DB アクセス定義)」参照）又は環境設定の Database Connection Server 接続設定（「4.5.5 Database Connection Server 接続設定」参照）で指定します。このとき、サービス名を指定する場合は、[Windows インストールディレクトリ]¥SYSTEM32¥DRIVERS¥ETC¥SERVICES ファイルに、サービス名とそれに対応するポート番号を設定してください。Database Connection Server との接続にポート番号を使用する場合は、ここでの設定は不要です。

例えば、ポート番号（接続する Database Connection Server のサーバ空間のポート番号(サーバスケジュール番号)）が 40179 に対して、サービス名：DBC SV を設定する場合は、次の記述をファイルの行末に追加します。

```
DBC SV 40179/tcp
```

注意事項

- サービス名には、任意の名称が設定できます。
- ここで設定したサービス名を、DABroker 動作環境定義（「4.5.5 Database Connection Server 接続設定」参照）、コネクションマルチ定義（「4.7 DABroker コネクションマルチ定義(Database Connection Server 経由の DB アクセス定義)」参照）で指定してください。
- ポート番号は、Database Connection Server のサーバ空間のポート番号(サーバスケジュール番号)にしてください。
- Windows ファイアウォールが有効になっていないか確認し、有効になっている場合は DABroker を使用する前に、DABroker のサービスポートを Windows ファイアウォールの例外に登録してください。

また、[Windows インストールディレクトリ]¥SYSTEM32¥DRIVERS¥ETC¥HOSTS ファイルには接続する Database Connection Server の IP アドレスとホスト名を記述してください。HOSTS ファイルは IP アドレスからホスト名への変換オプションが有効な場合（DABroker 環境設定項目の DABIPCONVERT に ON を指定した場合）に参照します。HOSTS ファイルについては、Windows のマニュアルを参照してください。

4.4 DABroker 環境設定

システム構成に合わせた DABroker の動作環境を、**環境設定ユーティリティ**を使って設定してください。環境設定ユーティリティは、DABroker 動作環境定義ファイル：[DABroker 運用ディレクトリ¥conf] に dasysconf を作成します。ここでは、環境設定ユーティリティの操作方法について説明します。

4.4.1 環境設定ユーティリティの操作

環境設定ユーティリティは、プログラムフォルダに登録された [DABroker 環境設定] から起動します。

(1) 操作できるユーザ

環境設定ユーティリティを実行して、DABroker 動作環境定義ファイルを操作できるのは、Windows の Administrators グループに属するユーザだけです。

(2) 操作方法

環境設定ユーティリティは、DABroker が終了している状態で使用します。次回起動時に、変更した DABroker 動作環境定義ファイルの内容が有効になります。

操作方法を次に示します。

1. プログラムフォルダに登録された [環境設定] を起動する。

環境設定ユーティリティが起動し、[DABroker 動作環境設定] ダイアログボックスが表示されます。



2. 項目を設定する。

設定する項目については、「4.5 DABroker 環境設定項目」を参照してください。

3. 設定内容を確認して [OK] ボタンを選択する。

変更した内容が有効になり、環境設定ユーティリティは終了します。

4.5 DABroker 環境設定項目

環境設定ユーティリティで操作できる項目には次の種類があります。

- **共通設定の項目**

リモートアクセス、ローカルアクセスのどちらでデータベースにアクセスする場合でも共通に設定する項目です。

DABroker for Java Version 2 から DABroker にローカルアクセスする場合は、リモートアクセス設定で設定する次の項目についても有効になります。

- HiRDB の BLOB 型データ受取バッファサイズ
- ORACLE の LONG 型データ受取バッファサイズ
- ODBC3.0 の LONG 型データ受取バッファサイズ
- Database Connection Server の BLOB 型データ受取バッファサイズ

これらの項目の設定方法については、「[4.5.2 リモートアクセス設定](#)」を参照してください。

- **リモートアクセス設定の項目**

DBPARTNER, HITSENSER5, DBPARTNER Library, 又は XDM/RD 外部表を使用する場合に、[共通設定] タブで設定した項目以外に設定できる項目です。

- **for C++, for COBOL, for ORB 設定の項目 (アプリケーションの (イベントトレース取得設定))**

DABroker 関連製品のイベントトレースを取得する場合に設定する項目です。アプリケーションからのアクセスでエラーが発生した場合に、取得したトレースを参照してエラーに対処できます。

- **エージェント設定の項目**

DBPARTNER のエージェント機能を使用する場合に設定する項目です。

- **Database Connection Server 接続設定**

Database Connection Server を経由してメインフレーム系データベースにアクセスする場合に設定する項目です。

ここで示す標準値は、DABroker 環境設定ユーティリティ起動時に、あらかじめ表示されている値です。

4.5.1 共通設定の項目

(1) 作業ディレクトリ名 (DABTMP)

標準値

DABroker 運用ディレクトリ¥tmp

指定範囲

任意のディレクトリ（ただし、パーティションを圧迫しないところ）。ただし、ネットワークドライブ上のディレクトリは指定できません。また、指定したディレクトリのアクセス権限として、システムにフルコントロールを指定してください。

説明

DABroker の作業ディレクトリを指定します。このディレクトリは、予約処理用の SQL の格納や、処理途中で作成される一時ファイルの格納などに使用されます。

[...] ボタンを押すと、[ディレクトリ選択] ダイアログボックスが表示されます。

(2) データベースアクセス処理終了待ちタイムアウト (DABDBA_TIMEOUT)

標準値

1800 (秒)

指定範囲

0, 又は 1~65535 (秒)

説明

データベースアクセス処理の終了待ちタイムアウトを監視するかどうかを設定します。Database Connection Server にアクセスする場合だけ有効です。

- 1~65535 (秒) を設定した場合

DABroker がデータベースアクセス処理を開始してから終了するまでの時間を監視します。

ここで設定した時間を超えてもデータベースアクセス処理が終了しない場合、DABroker は次の処理をします。

クライアントに対しては、終了待ちタイムアウトの発生を通知して、コネクションを解放します。処理を続ける場合、クライアントは一度終了して再起動する必要があります。

データベースに対しては、データベースを切り離します。データベースでの処理は DBMS によってロールバックされます。

予約処理中、又はエージェント処理中にここで設定した時間を超えてもタイムアウトは発生しません。

- 0 を設定した場合

データベースアクセス処理の終了待ちタイムアウトを監視しません。

(3) 表検索カラム数 (DABSELECTCOLCNT)

標準値

0

指定範囲

0, 又は 1~2147483647

説明

表検索用の領域として確保する表カラムの個数を指定します。

- 0 を設定した場合

DBMS に対して表カラムの個数を要求する SQL を発行します。DBMS から返された結果を基に、表検索用の領域を確保します。SQL Anywhere 使用時は常に 0 を仮定します。

- 1～2147483647 を設定した場合

表カラムの個数が指定した値より少ない場合、DBMS に対して表カラムの個数を要求する SQL を発行しません。このため、SQL を発行する場合より処理性能が向上します。ただし、検索を実行するたびに次に示す大きさのメモリが必要になるため、動作環境を考慮して値を設定してください。よく検索する表のカラム数を大きく超える値を設定した場合、性能劣化の原因になることがあります。

HiRDB, ORACLE の場合：1 カラム当たり 128 バイト

RDA Link for Gateway の場合：1 カラム当たり 256 バイト

表カラムの個数が指定した値より多い場合、DBMS に対して表カラムの個数を要求する SQL を発行します。DBMS から返された結果を基に、表検索用の領域を確保します。

(4) 更新・検索項目数 (DABSELECTINFCNT)

標準値

0

指定範囲

0, 又は 1～2147483647

説明

表の更新・検索用の領域として確保する、入力項目（パラメタ）の個数を指定します。

- 0 を設定した場合

DBMS に対して入力項目の個数を要求する SQL を発行します。DBMS から返された結果を基に、表の更新・検索用の領域を確保します。SQL Anywhere 使用時は常に 0 を仮定します。

- 1～2147483647 を設定した場合

入力項目の個数が指定した値より少ない場合、DBMS に対して入力項目の個数を要求する SQL を発行しません。このため、SQL を発行する場合より処理性能が向上します。ただし、更新・検索を実行するたびに次に示す大きさのメモリが必要になるため、動作環境を考慮して値を設定してください。よく更新・検索する表の入力項目の個数を大きく超える値を設定した場合、性能劣化の原因になることがあります。

HiRDB, ORACLE の場合：1 カラム当たり 128 バイト

RDA Link for Gateway の場合：1 カラム当たり 256 バイト

入力項目の個数が指定した値より多い場合、DBMS に対して入力項目の個数を要求する SQL を発行します。DBMS から返された結果を基に、表の更新・検索用の領域を確保します。

(5) ログファイルサイズ [dabsv.log] (DABSVLOGSIZE)

標準値

64 (キロバイト)

指定範囲

0, 又は 64~32767 (キロバイト)

説明

DABroker のログを取得するファイルの容量を設定します。ログは DABroker 運用ディレクトリ %spool%dabsv.log, および %spool%dabsvdemon.log に取得されます。

- 0 を設定した場合
ファイルの容量は無限大になります。
- 64~32767 を設定した場合
設定した値がファイルの容量になります。

注意事項

ログファイルが満杯になったら、ラップアラウンドでログを取得します。ラップアラウンドした場合、1 世代前のバックアップファイル (dabsv.logold, dabsvdemon.logold) を作成し、1 世代だけを管理します。

(6) 使用する ORACLE のバージョン (DABORACLE_TYPE)

標準値

ORACLE9i

指定範囲

ORACLE9i, ORACLE10g, ORACLE11g, ORACLE12c 又は ORACLE19c

説明

使用する ORACLE のバージョンを設定します。

(7) HiRDB のディクショナリ表のデータ属性 (DABSQL_HiRDB_DICTIONARY_DATATYPE)

標準値

C : VARCHAR

指定範囲

M 又は C

説明

HiRDB のディクショナリ表の列属性を指定します。

- M を設定した場合

混在文字データ (MVARCHAR) として処理します。

- C を設定した場合

文字データ (VARCHAR) として処理します。

注意事項

ここでの指定と HiRDB で使用されているディクショナリ表のデータ属性が異なる場合は、結果不正や SQL エラーになる場合があります。

(8) HiRDB 環境変数区切り文字 (DAB_HIRDB_DBINF_ESC)

標準値

@

指定範囲

@ ! \$ % & * + - / : ? ¥ |

説明

HiRDB 環境変数区切り文字を指定します。

注意事項

特に問題がない場合、区切り文字には@を使用してください。

区切り文字は必ず半角 1 文字で指定してください。全角文字は指定できません。

DABroker for Java から HiRDB クライアント環境変数グループを指定してデータベースと接続する場合は、@を使用してください。

(9) データベースアクセストレースの取得 (DABSQL_TRCSIZE)

標準値

取得しない

指定範囲

取得しない又は取得する

説明

データベースアクセストレースを取得するかどうかを設定します。アクセストレースの出力については、[\[4.9.2 DABroker トレース出力ユーティリティの実行\]](#)を参照してください。

[データベースアクセストレース設定] ボタンを押すと [アクセストレース設定] ダイアログボックスが表示され、アクセストレースで出力する項目を選択できます。出力できる項目の詳細については、[\[6.5.1 アクセストレース設定\]](#)を参照してください。

- 取得するを設定した場合

次に示すファイル名ですべてのアクセストレースを取得するファイルが、DABroker 運用ディレクトリ¥spool¥db_access の下に作成されます (データベースアクセストレースは DABroker トレース出力ユーティリティで出力します)。

db_XXXXX

XXXXX：システム任意の番号。

ハードディスクの空き容量がなくなった場合、それ以降のアクセストレースは取得しません。なお、格納する領域のサイズは設定できません。

クライアントからの要求量が多い場合は、トレースのファイル容量も多く必要です。ハードディスクの容量に余裕をもたせてください。必要容量の概算方法については、「[4.9.2\(3\) データベースアクセストレースの出力方法](#)」のデータベースアクセストレースの出力の注意事項を参照してください。また、クライアントからの要求量が多ければ、DABroker を終了するまでに時間がかかります。御注意ください。

- 取得しないを設定した場合
データベースアクセストレースを取得しません。

(10) 拡張データベースアクセストレースを取得する (DABEXSQL_TRC_LINE)

標準値

OFF (取得しません)

指定範囲

OFF (チェックなし) 又は ON (チェックあり)

(OFF：取得しません ON：取得します)

説明

拡張データベースアクセストレースを取得するかどうかを設定します。

- OFF (チェックなし) を設定した場合
拡張データベースアクセストレースを取得しません。
- ON (チェックあり) を設定した場合
拡張データベースアクセストレースを取得します。出力行数の指定については、「[4.5.1\(11\) トレースファイルサイズ](#)」を参照してください。

拡張データベースアクセストレースの詳細については、「[6.5.2 拡張データベースアクセストレース機能](#)」を参照してください。

(11) トレースファイルサイズ

標準値

1024 (行)

指定範囲

0 (トレース取得行数を無限大にします), 又は 1024~32767 (行)

説明

「4.5.1(10) 拡張データベースアクセストレースを取得する (DABEXSQL_TRC_LINE)」で ON を設定した場合に、拡張データベースアクセストレースのトレースファイルサイズを指定します。

- 0 を設定した場合
出力行数は、無限大となります。
- 1024~32767 を設定した場合
拡張データベースアクセストレースを取得します。出力行数は、設定した数値分となります。

拡張データベースアクセストレースのファイルは、1 行に 134 バイトまで出力されます。拡張データベースアクセストレースのファイルの出力行数は、ここでの設定値に従います。ただし、ヘッダは行数に含まれません。

拡張データベースアクセストレースの詳細については、「6.5.2 拡張データベースアクセストレース機能」を参照してください。

(12) XA トレースを取得する (DABXA_LOG_ERROR)

標準値

ON (チェックあり)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)

(ON: 取得します OFF: 取得しません)

説明

XA インタフェースでエラーを検知した場合、エラートレースを取得するかどうかを設定します。

XA トレースは、DABroker 運用ディレクトリ下の spool ディレクトリに db_xainfo.log というファイル名で取得されます。XA トレースファイル(db_xainfo.log)はテキスト形式なので、任意のテキストエディタで参照してください。

XA トレースファイルは、運用期間に応じてファイルサイズが増加するため、定期的に削除する必要があります。

4.5.2 リモートアクセス設定

(1) 予約処理作業ディレクトリ名 (DABBATCH)

標準値

DABTMP が設定されているディレクトリ

指定範囲

任意のディレクトリ（ただし、パーティションを圧迫しないところ）。ただし、ネットワークドライブ上のディレクトリは指定できません。また、指定したディレクトリのアクセス権限として、システムにフルコントロールを指定してください。

説明

予約処理の終了ステータスを格納するディレクトリを指定します。クライアントが予約完了状況を次の日に確認する場合など、サーバマシンの終了後もファイルを保存する必要がある場合は、指定したディレクトリ下に作成されたファイルを削除しないでください。指定したディレクトリの下には、次のようなファイルが作成されます。

• 予約処理作業ファイル

DABroker が内部的に使用します。

`%DABTMP%¥hhmmssnnnnnxxxx. BC_`

hhmmss：予約処理を受け付けた時刻（時 分 秒）。

nnnnn：プロセス id。

xxxx：アルファベット。HHMMSSnnnnn が重複した場合の識別に使用する文字。

• 予約処理ステータスファイル

予約処理を実行する場合に DABroker が内部的に使用します。このファイルは、クライアントからの要求で削除されます。

この予約処理用ステータスファイルを開いて内容を変更したり、ファイルを削除したりしないでください。予約処理が正常に実行できなくなります。

`%DABBATCH%¥YOYAKU¥hhmmssnnnnnxxxx. BCH`

hhmmss：予約処理を受け付けた時刻（時 分 秒）。

nnnnn：プロセス id。

xxxx：アルファベット。HHMMSSnnnnn が重複した場合の識別に使用する文字。

• 予約処理用システムファイル

DABroker が内部的に使用します。このファイルはクライアントからの要求では削除されません。

この予約処理用システムファイルを開いて内容を変更したり、ファイルを削除したりしないでください。予約処理が正常に実行できなくなります。

`%DABBATCH%¥YOKAKU¥DBPSV. BCH`

(2) クライアントからの要求待ちタイムアウト (DABCLIENT_TIMEOUT)

標準値

3600 (秒)

指定範囲

0, 又は 1~65535 (秒)

説明

クライアントからの要求待ちタイムアウトを監視するかどうかを設定します。

- 1～65535（秒）を設定した場合

DABroker がアクセス要求に対する応答をクライアントに返してから、次のアクセス要求を受けるまでの時間を監視します。

ここで設定した時間を超えてクライアントからのアクセス要求がない場合、DABroker は次の処理をします。

クライアントに対しては、要求待ちタイムアウトの発生を通知して、コネクションを解放します。処理を続ける場合、クライアントは一度終了して再起動する必要があります。

データベースに対しては、このタイムアウトが発生したクライアントの処理がデータベースで実行中の場合は、トランザクションのロールバックを要求し、データベースを切り離します。

- 0 を設定した場合

クライアントからの要求待ちタイムアウトを監視しません。

(3) 最大接続ユーザ数 (DABMAXUSERS)

標準値

10（ユーザ）

指定範囲

1～100000（ユーザ）

説明

コネクションを確立できる最大ユーザ数を設定します。ここで設定した値を超えるユーザが接続しようとした場合、DABroker は、クライアントにエラーメッセージを返してコネクションを確立しません。

(4) 同時実行可能なデータベースアクセス数 (DABDBANUMBER)

標準値

20（データベース接続数）

指定範囲

1～100000（データベース接続数）

説明

DABroker が一度に受け付けられる、複数のクライアントからのデータベースアクセス要求（同期アクセス処理要求、及び予約処理要求）の合計数を設定します。ここで設定した値を超える要求があった場合は、クライアントにエラーメッセージを返し、処理を受け付けません。ただし、ここで設定した値を超えてもエージェント処理は受け付けます。

(5) 1 プロセスあたりの実行ユーザ数 (DABMAXBLOCKUSERS)

標準値

1 (ユーザ数)

指定範囲

1~100 (ユーザ数)

ORACLE を使用する場合は 1 を指定してください。

説明

一つのプロセスで処理する最大ユーザ数を設定します。ここで設定した値を超えた場合は、クライアントにエラーメッセージを返し、処理を受け付けません。

ORACLE を使用する場合は、必ず 1 を指定してください。1 以外を指定した場合は、クライアントにエラーメッセージを返し、処理を受け付けません。

Database Connection Server にアクセスする場合、接続先データベース定義のデータベース種別を複数使用するときは、必ず 1 を指定してください。

注意事項

DABroker のデーモンプロセスが、サーバのメモリ又はキャッシュメモリ上で動作するように適切な値を設定してください。

(6) 最大処理プロセス数 (DABMAXPROCESS)

標準値

10 (プロセス数)

指定範囲

1~1000 (プロセス数)

説明

DABroker で起動する最大プロセス数を設定します。ここで設定した値を超えた場合は、クライアントにエラーメッセージを返し、処理を受け付けません。

注意事項

不要なデーモンプロセスの起動を制限するために適切な値を設定してください。

(7) HIRDB の BLOB 型データ受取バッファサイズ (DABHIRDBA_BLOBBUFSIZE)

標準値

4096 (バイト)

指定範囲

0, 又は 1~2147483647 (バイト)

説明

HiRDB から BLOB 型データを抽出する場合の受け取りサイズを指定します。

このデータ型の場合、HiRDB では最大 2147483647 バイトのデータの存在が考えられますが、DABroker ではここで指定した値までのデータを受け取り、以降のデータは切り捨てます。

0 を指定した場合、DBMS の定義長分の領域長を仮定します。ただし、実際に格納されているデータの大きさにかかわらず、無条件に定義長分の領域が確保されるため、メモリ効率が悪くなることがあります。格納されているデータの最大長が分かっている場合は、その値を指定することをお勧めします。この項目は、DABroker for Java Version 2 で DABroker にローカルアクセスした場合にも有効となります。

(8) ORACLE の LONG 型データ受取バッファサイズ (DABORADBA_LONGBUFSIZE)

標準値

4096 (バイト)

指定範囲

0, 又は 1~2147483647 (バイト)

説明

ORACLE から、LONG 型データを抽出する場合の受け取りサイズを指定します。

このデータ型の場合、ORACLE では最大 2147483647 バイトのデータの存在が考えられますが、DABroker ではここで指定した値までのデータを受け取り、以降のデータは切り捨てます。

0 を指定した場合、DBMS の定義長分の領域長を仮定します。ただし、実際に格納されているデータの大きさにかかわらず、無条件に定義長分の領域が確保されるため、メモリ効率が悪くなることがあります。格納されているデータの最大長が分かっている場合は、その値を指定することをお勧めします。この項目は、DABroker for Java Version 2 で DABroker にローカルアクセスした場合にも有効となります。

この項目は、DABroker for Java Version 2 でデータベース種別に Oracle8i を設定している場合は設定しません。

(9) ODBC3.0 の LONG 型データ受取バッファサイズ (DABODBC30_LONGBUFSIZE)

標準値

4096 (バイト)

指定範囲

0, 又は 1~2147483647 (バイト)

説明

ODBC3.0 で接続したデータベースから LONG 型データを抽出する場合の受け取りサイズを指定します。

このデータ型の場合、最大 2147483647 バイトのデータの存在が考えられますが、DABroker ではここで指定した値までのデータを受け取り、以降のデータは切り捨てます。

0 を指定した場合、DBMS の定義長分の領域長を仮定します。ただし、実際に格納されているデータの大きさにかかわらず、無条件に定義長分の領域が確保されるため、メモリ効率が悪くなることがあります。格納されているデータの最大長が分かっている場合は、その値を指定することをお勧めします。この項目は、DABroker for Java Version 2 で DABroker にローカルアクセスした場合にも有効となります。

(10) Database Connection Server の BLOB 型データ受取バッファサイズ (DABDBSDBA_BLOBBUFSIZE)

標準値

4096 (バイト)

指定範囲

0, 又は 1~2147483647 (バイト)

説明

Database Connection Server で接続したデータベースから BLOB 型データを抽出する場合の受け取りサイズを指定します。

このデータ型の場合、最大 2147483647 バイトのデータの存在が考えられますが、DABroker ではここで指定した値までのデータを受け取り、以降のデータは切り捨てます。

0 を指定した場合、DBMS の定義長分の領域長を仮定します。ただし、実際に格納されているデータの大きさにかかわらず、無条件に定義長分の領域が確保されるため、メモリ効率が悪くなることがあります。格納されているデータの最大長が分かっている場合は、その値を指定することをお勧めします。

この項目は、DABroker for Java Version 2 で DABroker にローカルアクセスした場合にも有効となります。

(11) 漢字コードセット (DABLANGMODE)

標準値

SJIS (シフト JIS モード)

指定範囲

SJIS (シフト JIS モード), EUC (日本語 EUC モード), ASCII (ASCII モード), 又は UTF-8 (UTF-8 モード)

説明

DABroker が稼働する言語モードを設定します。

各言語モードで操作できるデータベースを表 4-2 に示します。

表 4-2 各言語モードで操作できるデータベース

データベース	DABroker での設定			
	シフト JIS モード	日本語 EUC モード	ASCII モード	UTF-8 モード
HiRDB	○	○	○	○
ORACLE	○	○	○	○
SQL Anywhere	○	×	○	×
Adaptive Server Anywhere	○	×	○	×
SQL Server	○	×	○	×
VOS3 XDM/RD	○	×	○	×
VOS3 XDM/SD	○	×	○	×
VOS1 RDB1 E2	○	×	○	×
VOSK SQL/K	○	×	○	×
VOS1 PDMII E2	○	×	○	×
VOSK スプールファイル	○	×	○	×

凡例

○：操作できる

×：操作できない

各データベースで使用する言語モードは、必ず DABroker の言語モードと合わせてください。データベースと DABroker の言語モードが異なっている場合は動作を保証しません。

クライアントが DABroker for Java の場合は、DAB_LANG の指定が JDBC の文字エンコーディングに影響します。詳細については、マニュアル「DABroker for Java™ Version 2 DABroker Driver for Java™ Technology」の「接続するデータベースの指定」の ENCODELANG の説明を参照してください。

DABroker を日本語 EUC モード、ASCII モード、又は UTF-8 モードで動作させた場合、メッセージは英文で出力されます。メッセージの詳細については、「7 メッセージ (共通)」を参照してください。

(12) TCP/IP 通信時キープアライブ (DABKEEPALIVE)

標準値

OFF

指定範囲

ON 又は OFF (ON：キープアライブによる監視を行います)

OFF：キープアライブによる監視を行いません)

説明

キープアライブによる監視を行うかどうかを設定します。キープアライブとは、クライアントとの通信経路が確立されているかどうかを常に監視する TCP/IP の機能です。

- ON を設定した場合
キープアライブによる監視が行われます（クライアントとの通信経路を監視する）。
- OFF を設定した場合
キープアライブによる監視は行われません（クライアントとの通信経路を監視しない）。

(13) IP アドレスからホスト名への変換オプション (DABIPCONVERT)

標準値

ON

指定範囲

ON 又は OFF

説明

メッセージ及びユティリティで表示される IP アドレスをホスト名に変換するかどうかを設定します。

- ON を設定した場合
表示される IP アドレスをホスト名に変換します。
動作環境によってこの変換処理に時間がかかり、接続処理が遅くなることがあります。
- OFF を設定した場合
表示される IP アドレスをホスト名に変換しません。

(14) 通信トレースの取得 (DABTCP_TRCSIZE)

標準値

取得しない

指定範囲

取得しない又は取得する

説明

クライアントとの通信トレースを取得するかどうかを設定します。通信トレースの出力については、[\[4.9.2 DABroker トレース出力ユティリティの実行\]](#) を参照してください。

- 「取得する」を設定した場合
次に示すファイル名ですべての通信トレースを取得するファイルが DABroker 運用ディレクトリ `¥spool¥dabcltrc` の下に作成されます（通信トレースは DABroker トレース出力ユティリティで出力します）。
`XXXXXXXX.cmt`
`XXXXXXXX.cmt_AA~AA`

XXXXXXXXX：任意の文字列。

AA～AA：任意の文字列。

ハードディスクの空き容量がなくなった場合、それ以降の通信トレースは取得しません。なお、格納するファイルサイズは設定できません。

クライアントからの要求量が多い場合は、トレースのファイルの容量も多く必要です。ハードディスクの容量は余裕を持たせてください。必要容量の概算方法については「4.9.2(2) クライアントとの通信トレースの出力方法」で通信トレースの出力の注意事項を参照してください。また、クライアントからの要求量が多ければ、DABrokerを終了するまでに時間がかかります。御注意ください。

- 取得しないを設定した場合
クライアントとの通信トレースを取得しません。

4.5.3 エージェント設定

(1) エージェント機能の使用 (DABAGT_USE)

標準値

使用しない

指定範囲

使用しない又は使用する

説明

エージェント機能を使用する場合の詳細設定をします。

(2) 作業ディレクトリ (DABAGT)

標準値

DABroker 運用ディレクトリ¥AGT

指定範囲

任意のディレクトリ（ただし、パーティションを圧迫しないところ）。ただし、ネットワークドライブ上のディレクトリは指定できません。また、指定したディレクトリのアクセス権限として、システムにフルコントロールを指定してください。

説明

次に示すエージェント登録情報、実行情報、及び実行結果を格納するディレクトリを指定します。

- エージェント登録情報用ファイル
エージェント登録情報を格納するファイルで、エージェント登録ごとにできるファイルです。エージェント登録数が多くなるにつれ、エージェント登録情報ファイル数も増え、サーバファイルシステムを圧迫するため、ルートディレクトリのパーティションとは別に、エージェント用ディレクトリのパーティションを確保するなどの運用をシステム管理者が行ってください。

ファイル運用の詳細については「[5.4 エージェント機能で使用するファイルの運用](#)」, エージェント登録情報ファイルの容量の目安となる概算式については「[5.4.5\(2\) ファイル容量の見積もり](#)」を参照してください。

`%DABAGT%\REGa1a2~an.AGT`

$a_1a_2 \sim a_n$: 登録したエージェント名 (DBPARTNER/Client で指定した文字列)。

それぞれ 0~9, A~F の文字。n は最大 128。

- **エージェント登録情報管理ファイル**

エージェント登録情報を管理するファイルで, DABroker が内部的に使用します。

`%DABAGT%\RCTRL.AGT`

`%DABAGT%\RCTRL.BAK`

`%DABAGT%\RCTRL.WK`

- **エージェント実行情報用ファイル**

クライアントから受けたエージェント実行要求の情報を格納します。

エージェント実行情報ファイルは, エージェント実行要求受け付けごとに行けるファイルです。エージェント実行要求数が多くなるにつれ, エージェント実行情報ファイル数も増え, サーバファイルシステムを圧迫するため, ルートディレクトリのパーティションとは別に, エージェント用ディレクトリのパーティションを確保するなどの運用をシステム管理者が行ってください。

ファイル運用の詳細については「[5.4 エージェント機能で使用するファイルの運用](#)」, エージェント実行情報ファイルの容量の目安となる概算式については「[5.4.5\(2\) ファイル容量の見積もり](#)」を参照してください。

`%DABAGT%\EXEa1a2~a16.AGT`

$a_{13} \sim a_{16}$: エージェント実行名 (DABroker が割り当てる識別名 (番号))。4 けたの数字。

- **エージェント実行情報管理ファイル**

エージェントの実行情報を管理するファイルで, DABroker が内部的に使用します。

`%DABAGT%\ECTRL.AGT`

`%DABAGT%\ECTRL.BAK`

`%DABAGT%\ECTRL.WK`

- **エージェント実行結果格納ファイル**

エージェントの実行結果を格納します。エージェント実行時に作成されます。

ファイル運用の詳細については, 「[5.4 エージェント機能で使用するファイルの運用](#)」を参照してください。

`%DABAGT%\RESa1a2~a16.AGT`

$a_{13} \sim a_{16}$: エージェント実行名 (DABroker が割り当てる識別名 (番号))。4 けたの数字。

- **実行予約指定エージェントの実行待ちキューファイル**

DABroker の起動時に作成され, エージェント情報に実行予約を設定した (エージェントタイマー指定した) エージェントの, キュー情報について格納します。

`%DABAGT%\TIMER_QUE`

%DABAGT%¥TIME_QUE. TMP

%DABAGT%¥TIMER_QUE. BAK

- エージェントログファイル

エージェント機能を使用したときのログを取得するファイルです。エージェント機能を使用するごとにログを取得します。

エージェントログファイルが満杯になったらラップアラウンドで取得します。ラップアラウンドで取得した場合は、1世代だけを世代管理し、バックアップファイル (agtlogfile.old) を作成します。

エージェントログの取得形式、及び取得内容については、「[6.3.2 エージェントログの取得形式と取得内容 \(共通\)](#)」を参照してください。

%DABAGT%¥AGTLOGFILE

%DABAGT%¥AGTLOGFILE. OLD

%DABAGT%¥AGTLOGLOCK

(3) 登録可能エージェント数 (DABAGTREGMAX)

標準値

1000 (エージェント数)

指定範囲

1~32767 (エージェント数)

説明

登録可能な最大エージェント数を設定します。ここで設定した値を超える要求があった場合は、クライアントにエラーメッセージを返し、処理を受け付けません。

(4) 実行予約可能最大エージェント数 (DABAGTTIMMAX)

標準値

250 (エージェント数)

指定範囲

1~32767 (エージェント数)

説明

実行予約を設定して (エージェントタイマー指定して) 実行する、最大エージェント数を設定します。ここで設定した値を超える要求があった場合は、クライアントにエラーメッセージを返し、処理を受け付けません。既に登録済みの実行予約指定エージェントの最大エージェント数より少ない値を指定した場合は、設定値を変更しないで登録済みの値が設定されます。

(5) エージェントログファイル容量 (DABAGT_LOG_SIZE)

標準値

64 (キロバイト)

指定範囲

-1, 0, 又は 1~32767 (キロバイト)

説明

エージェントログを取得するかどうか、及び取得する場合は、エージェントログを取得するファイルの容量を設定します。

ログを取得する設定をした場合、エージェントログファイルが満杯になった場合、ラップアラウンドしてバックアップファイル (agtlogfile.old) を作成します。バックアップファイルは 1 世代だけを管理します。

エージェントログとして取得する情報にはインフォメーション情報とエラー情報があり、それぞれ取得するかどうかを設定してください。エージェントログの取得形式については、「[6.3.2 エージェントログの取得形式と取得内容 \(共通\)](#)」を参照してください。

- -1 を設定した場合
エージェントログを取得しません。
- 0 を設定した場合
エージェントログを取得します。ファイルの容量は、無限大になります。
- 1~32767 の値を設定した場合
エージェントログを取得します。設定した値がファイルの容量になります。

(6) 実行結果格納ファイル容量 (DABAGTFILMAX)

標準値

500 (キロバイト)

指定範囲

1~2097151 (キロバイト)

説明

エージェントの実行結果格納ファイルの最大容量を設定します。ここで設定した値を超えて要求された場合は、クライアントにエラーメッセージを返し、設定サイズに入る限りのデータを保存して処理を終了します。

(7) エージェントログの選択

(a) エージェントログにインフォメーションメッセージを取得する (DABAGT_LOG_INFO)

標準値

ON

指定範囲

ON (チェックあり) 又は OFF (チェックなし)

(ON：取得します OFF：取得しません)

説明

エージェントログにインフォメーション情報を取得するかどうかを設定します。インフォメーション情報とは、DABroker で実行したエージェントのログです。

インフォメーション情報の取得形式については、「[6.3.2 エージェントログの取得形式と取得内容（共通）](#)」を参照してください。

(b) エージェントログにエラーメッセージを取得する (DABAGT_LOG_ERROR)

標準値

ON

指定範囲

ON（チェックあり）又は OFF（チェックなし）

(ON：取得します OFF：取得しません)

説明

エージェントログにエラー情報を取得するかどうかを設定します。エラー情報とは、DABroker で実行したエージェントのエラーログです。

エラー情報の取得形式については、「[6.3.2 エージェントログの取得形式と取得内容（共通）](#)」を参照してください。

(8) グループ管理用ディレクトリ (DABAGTGRP)

標準値

DABroker 運用ディレクトリ¥AGTGRP

指定範囲

任意のディレクトリ（ただし、パーティションを圧迫しないところ）。ただし、ネットワークドライブ上のディレクトリは指定できません。また、指定したディレクトリのアクセス権限として、システムにフルコントロールを指定してください。

説明

複数ユーザをグループ化して管理するファイル（グループ管理情報ファイル）を格納するディレクトリを指定します。複数ユーザをグループ化することで、各処理要求時にグループ以外のユーザの要求を拒否できます。グループ管理情報ファイルは、グループごとに、次のファイル名で作成します。作成手順については、「[5.1.3 エージェント機能を使用するユーザのグループ管理](#)」を参照してください。

%DABAGTGRP%¥/gg…gg

gg…gg：グループID

なお、性能劣化の原因となるため、このディレクトリには、グループ情報以外の不要なファイルを格納しないでください。

グループ管理情報ファイルの作成手順については、「[5.1.3 エージェント機能を使用するユーザのグループ管理](#)」を参照してください。

(9) 実行予約日を経過しても有効にする日数 (DABAGTWCHALDAY)

標準値

0 (日)

指定範囲

0~31 (日)

説明

エージェントの実行を予約した日が過ぎても実行予約処理を有効にする最大日数を指定します。実行予約日を経過しても有効にする日数を指定することで、サーバマシンとクライアントでユーザが認識する時間 (日数) のずれを調整できます。

(10) 実行予約日を経過しても有効にする秒数 (DABAGTWCHALTIME)

標準値

0 (秒)

指定範囲

0~86400 (秒)

説明

エージェントの実行を予約した時刻が過ぎても実行予約処理を有効にする最大時間を指定します。実行予約時刻を経過しても有効にする時間を設定することで、サーバマシンとクライアントでユーザが認識する時間のずれを調整できます。

(11) 即時配信実行監視時間 (DABAGTEXEDIRETIME)

標準値

1800 (秒)

指定範囲

0, 又は 60~3600 (秒)

説明

DABroker がエージェント即時配信実行でのエージェント処理終了を待つ時間を設定します。

- 0 を設定した場合
実行時間の監視をしないで、エージェント処理の終了を待ちます。
- 60~3600 を設定した場合
ここで設定した時間を超えてもエージェント処理が終了しない場合、DABroker は次の処理をします。
クライアントに対しては、エージェント即時配信実行監視タイムアウトの発生を通知します。
データベースに対しては、エージェントの実行をキャンセルします。データベースでの処理は、DBMS でロールバックされます。

4.5.4 for C++, for COBOL, for ORB 設定

DABroker 関連製品 (DABroker for C++, DABroker for ORB, DABroker for COBOL) を使用して開発したアプリケーションからアクセスする場合に、イベントトレースを取得するかどうか、取得する場合は、取得するファイルサイズ、取得レベル、取得するイベントの項目を設定します。

イベントトレースの取得形式については、「[6.3.1 イベントトレースの取得形式\(共通\)](#)」を参照してください。

(1) for C++設定 (DABroker for C++設定)

DABroker for C++のイベントトレースについて設定します。トレースの取得レベル、取得種別など、イベントトレースの詳細については、マニュアル「DABroker for C++」を参照してください。

- トレース取得サイズ (DABCPP_EVTTRC_SIZE)

標準値

32 (キロバイト)

指定範囲

-1, 0, 又は 10~32767 (キロバイト)

説明

イベントトレースを取得するかどうか、及び取得する場合は、トレースを取得するファイルの容量を設定します。トレースは DABroker 運用ディレクトリ¥spool¥dabevtttrccpp1 に取得されます。

- -1 を設定した場合
イベントトレースを取得しません。
- 0 を設定した場合
イベントトレースを取得します。ファイルの容量は無量大になります。
- 10~32767 を設定した場合
イベントトレースを取得します。設定した値がファイルの容量になります。

注意事項

トレースファイルが満杯になったらラップアラウンドでトレースを取得します。ラップアラウンドした場合は、1 世代前のバックアップファイル (dabevtttrccpp2) を作成し、常に 1 世代だけを管理します。

- トレース取得レベル (DABCPP_EVTTRC_LEVEL)

標準値

0

指定範囲

0 又は 1

説明

イベントトレースの取得レベルを設定します。

- トレース取得種別 00~03 (DABCPP_EVTTRC_nn, nn: 00~03)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)

(ON: 取得します OFF: 取得しません)

説明

00~03 の各イベントのトレースを取得するかどうかを設定します。

(2) for COBOL 設定 (DABroker for COBOL 設定)

DABroker for COBOL のイベントトレースについて設定します。トレースの取得レベル, 取得種別など, イベントトレースの詳細については, マニュアル「DABroker for COBOL」を参照してください。

- トレース取得サイズ (DABCOBOL_EVTTRC_SIZE)

標準値

32 (キロバイト)

指定範囲

-1, 0, 10~32767 (キロバイト)

説明

イベントトレースを取得するかどうか, 及び取得する場合は, トレースを取得するファイルの容量を設定します。トレースを DABroker 運用ディレクトリ¥spool¥dabevttrccobol1 に取得されます。

- -1 を設定した場合
イベントトレースを取得しません。
- 0 を設定した場合
イベントトレースを取得します。ファイルの容量は, 無限大になります。
- 10~32767 を設定した場合
イベントトレースを取得します。設定した値がファイルの容量になります。

注意事項

トレースファイルが満杯になったらラップアラウンドでトレースを取得します。ラップアラウンドした場合は, 1 世代前のバックアップファイル (dabevttrccobol2) を作成し, 常に 1 世代だけを管理します。

- トレース取得レベル (DABCOBOL_EVTTRC_LEVEL)

標準値

0

指定範囲

0~2

説明

DABroker for COBOL のイベントトレースの取得レベルを設定します。

- トレース取得種別 00,02 (DABCOBOL_EVTTRC_nn, nn : 00,02)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)

(ON : 取得します OFF : 取得しません)

説明

00,02 の各イベントのトレースを取得するかどうかを設定します。

(3) for ORB 設定 (DABroker for ORB 設定)

DABroker for ORB のイベントトレースについて設定します。トレースの取得レベル, 取得種別など, イベントトレースの詳細については, マニュアル「DABroker for ORB」を参照してください。

- トレース取得サイズ (DABORB_EVTTRC_SIZE)

標準値

128 (キロバイト)

指定範囲

-1, 0, 10~32767 (キロバイト)

説明

イベントトレースを取得するかどうか, 及び取得する場合は, トレースを取得するファイルの容量を設定します。トレースは DABroker 運用ディレクトリ¥spool¥dabevttrcorb1 に取得されます。

- -1 を設定した場合
イベントトレースを取得しません。
- 0 を設定した場合
イベントトレースを取得します。ファイルの容量は無量大になります。
- 10~32767 を設定した場合
イベントトレースを取得します。設定した値がファイルの容量になります。

注意事項

トレースファイルが満杯になったらラップアラウンドでトレースを取得します。ラップアラウンドした場合は, 1 世代前のバックアップファイル (dabevttrcorb2) を作成し, 常に 1 世代だけを管理します。

- トレース取得レベル (DABORB_EVTTRC_LEVEL)

標準値

1

指定範囲

0 又は 1

説明

イベントトレースの取得レベルを設定します。

- イベントトレース取得種別 00~03,10 (DABORB_EVTTRC_nn, nn:00~03,10)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)

(ON:取得します OFF:取得しません)

説明

00~03,10 の各イベントのトレースを取得するかどうかを設定します。

4.5.5 Database Connection Server 接続設定

ここでは、Database Connection Server に接続するときの接続情報の指定を、接続先データベース定義、コネクションマルチ定義で省略した場合に仮定される値を設定します。

(1) 接続先ホスト名または IP アドレス (DAB_DBS_HOST)

標準値

localhost

説明

接続する Database Connection Server のホスト名、又は IP アドレスを設定します。

(2) 接続先サービス名またはポート番号 (DAB_DBS_PORT)

標準値

0

説明

接続する Database Connection Server のサービス名又はポート番号を設定します。

サービス名を指定する場合は、[Windowsインストールディレクトリ]¥SYSTEM32¥DRIVERS¥ETC¥SERVICES ファイルにサービス名とそれに対応するポート番号を設定してください ([4.3.2 Database Connection Server を経由してメインフレーム系データベースにアクセスする場合] 参照)。

(3) 送信バッファサイズ (DAB_DBS_SEND_BUFFER_SIZE)

標準値

64 (キロバイト)

指定範囲

64~2097151 (キロバイト)

説明

Database Connection Server に送信するときの最大バッファサイズを設定します。

(4) 受信バッファサイズ (DAB_DBS_RECEIVE_BUFFER_SIZE)

標準値

64 (キロバイト)

指定範囲

64~2097151 (キロバイト)

説明

Database Connection Server から受信するときの最大バッファサイズを設定します。

4.5.6 その他の設定

その他の設定は、DABroker 動作環境定義の OTHERS 節に指定します。

DABroker 動作環境定義ファイル (dasysconf) は DABroker 運用ディレクトリ¥conf に格納されています。

(1) DAB_DEMON_LOG_MODE (デーモンの生成・消滅・クリーンアップ 実行結果ログ抑止)

標準値

なし

指定範囲

LIMITED

説明

デーモンの生成・消滅・クリーンアップ実行結果ログの出力によりディスクを圧迫したくない場合に指定します。

デーモンの生成・消滅・クリーンアップ実行結果ログは、本指定をしない限り、標準で出力されます。

デーモンの生成・消滅・クリーンアップ実行結果ログについては、「[6.3.4 デーモンの生成・消滅・クリーンアップ実行結果ログ \(共通\)](#)」を参照してください。

(2) DAB_ORA7_XAHANDLE_OCI8USE (Oracle XA 接続時の Pro*COBOL, Pro*C/C++連携機能)

標準値

なし

指定範囲

ON (使用する)

説明

XA 環境下の同一プロセス内で、Oracle Pro*COBOL, 又は Oracle Pro*C/C++で作成した UAP と、DABroker for C++ で作成した UAP から Oracle に接続をする場合に設定します。この設定をせず、XA 環境下の同一プロセス内で、Oracle Pro*COBOL, 又は Oracle Pro*C/C++で作成した UAP と、DABroker for C++ で作成した UAP から Oracle に接続すると「ORA-03114: not connected to ORACLE」エラーになります。

Oracle9i 9.0.1 に接続する場合は、” ON” を指定しても本機能は無効となります。

本機能を有効にすると、DABroker for C++ を使用した UAP から Oracle に接続、切断する場合に、次のように処理が変更されます。なお、以下「Oracle Call Interface」を OCI と呼びます。

(a) 本機能が無効の場合

(i)接続時

OCI7 の次の関数を実行する。

sqlld2()…XA 環境で、OCI7 で使用する接続情報を取得する

(ii)切断時

OCI 関数の呼び出し処理なし。

(b) 本機能が有効の場合

(i)接続時

OCI8 の以下の関数を実行する。

xaoEnv()…指定された XA 接続の OCI 環境ハンドルを取得する

xaoSvcCtx()…指定された XA 接続の OCI サービス・ハンドルを取得する

OCIHandleAlloc()…エラー・レポート・ハンドルを取得する

OCISvcCtxToLda()…OCI8 で生成した接続情報を OCI7 で使用する接続情報に変換する

(ii)切断時

OCI8 の以下の関数を実行する。

OCILdaToSvcCtx()…OCI7 で生成した接続情報を OCI8 で使用する接続情報に変換する

OCIHandleFree()…エラー・レポート・ハンドルを開放する

注意事項

本機能を使用して、XA 環境下の同一プロセス内で、DABroker for C++ で作成した UAP から Oracle に接続をした後に、DABroker for C++ 以外のインタフェース（Pro*COBOL, Pro*C/C++, 又は OCI 関数）で作成した UAP から Oracle に接続をする場合は、DABroker for C++ を使用した UAP で Oracle から切断をした後に行ってください。

DABroker では、OCISvcCtxToLda 関数を使用して動作しています。そのため、DABroker for C++ を使用した UAP で Oracle に接続している間に、DABroker for C++ 以外のインタフェースから Oracle に接続した場合、DABroker で管理している接続情報が変更されることがあるため、動作を保証できません。

4.6 DABroker 接続先データベース定義

DABroker for C++, DABroker for COBOL, DABroker for ORB, 及び DABroker for ODBC で開発するアプリケーションは、**接続先データベース定義ユーティリティ**で設定した**データベース種別名**、及び**データベース名**を使用してデータベースにアクセスします。また、DBPARTNER、及び DABroker for Java でも、Database Connection Server 経由でデータベースにアクセスする場合は、**接続先データベース定義ユーティリティ**で設定した情報を使用して Database Connection Server に接続します。接続先データベース定義ユーティリティで設定した情報を使用すると、接続条件などが変更されてもアプリケーションを変更する必要がありません。ただし、OTS 機能、又は Java JTS 機能を使用する場合は、接続先データベース定義ユーティリティで設定した情報を使用できません。

4.6.1 接続先データベース定義ユーティリティの操作

接続先データベース定義ユーティリティは、プログラムフォルダに登録された [DABroker 接続先データベース定義] から起動します。

接続先データベース定義ユーティリティで設定した内容は、接続先データベース定義ファイル：[DABroker 運用ディレクトリ¥conf¥dabenv]に出力されます。

(1) 操作できるユーザ

すべてのユーザ

ただし、操作する環境が Windows Vista, Windows 7, または Windows Server 2008 の場合は、Administrators グループに属するユーザだけです。

(2) 操作方法

1. プログラムフォルダに登録された [接続先データベース定義] を起動する。

DABroker 接続先データベース定義ユーティリティが起動し、[接続先データベース定義] ダイアログボックスが表示されます。



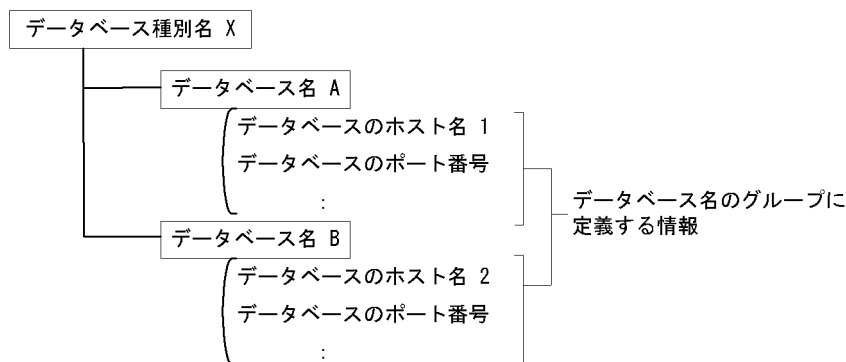
2. 使用するデータベースの製品名が記載されたタブを選択し、接続情報を設定する。

次の情報を定義します。

- データベース種別名：使用するデータベースに対して付ける任意名称
- データベース名：データベースごとに異なる情報の集まりに対して付ける任意名称
- データベース名のグループに定義する情報：データベース名のグループに定義する情報は、データベースごとに異なります。定義項目については、「4.6.2 設定する項目」を参照してください。

これらの情報は、次のような親子関係になります。

例)



この例で示すように、データベース名は、同一データベース種別名に対して、複数定義できます。例えば、ホスト名が異なる二つのHiRDBを、一つのデータベース種別として定義し、アプリケーションでは、データベース名でホストの異なるHiRDBを使い分けられます。

<設定方法>

- データベース種別名、データベース名を新規登録する場合
設定後、[追加] ボタンを選択します。
- 同一データベース種別名に対して、別のデータベース名を登録する場合

設定後、[追加] ボタンを選択します。

- 登録されている情報を変更する場合

登録内容を変更した後、[追加] ボタンを選択します。

- データベース種別名又はデータベース名を削除する場合

削除するデータベース種別又はデータベース名を選択して [削除] ボタンを選択してください。

4.6.2 設定する項目

データベースごとに設定する項目について説明します。

(1) HiRDB を使用する場合

設定項目名	指定内容
データベース種別名	使用するデータベースに対して付ける任意名称 <設定規則> 接続先データベース定義ユーティリティ内でユニークな名称にしてください。23文字まで指定できます。2バイト文字、半角カタカナ文字は使用しないでください。
データベース名	データベースごとに異なる情報の集まりに対して付ける任意名称 <設定規則> 同一データベース種別名内でユニークな名称にしてください。23文字まで指定できます。2バイト文字、半角カタカナ文字は使用しないでください。
接続先ホスト名 (DBHOST)	HiRDB のホスト名
接続先ポート番号 (DBNAME)	ポート番号
ユーザ ID(USERID)	接続ユーザ ID*
パスワード(PASSWD)	パスワード*
環境変数	HiRDB クライアントへ渡す環境変数を、下のテキストボックスに指定します。 HiRDB 使用時に指定できる環境変数の詳細については、HiRDB のマニュアルを参照してください。 <設定規則> 1024文字まで指定できます。 2バイト文字、及び半角の片仮名は使用しないでください。 各環境変数の先頭に、必ず区切り文字 (@) を記述してください。 PDHOST (ホスト名称、又は IP アドレス)、及び PDNAMEPORT (接続するポート番号) は、先頭か末尾で必ず指定します。 PDHOST、及び PDNAMEPORT は、重複して指定しないでください。 PDHOST、及び PDNAMEPORT 以外の環境変数は、DABroker で設定した値をそのまま使えます。

設定項目名	指定内容
	DABroker では、指定した環境変数の妥当性はチェックしません。
環境変数グループ	<p>HiRDB クライアントへ渡す環境変数グループ名を、下のテキストボックスで指定します。環境変数グループの名称や、環境変数グループで指定できる内容の詳細については、HiRDB のマニュアルを参照してください。</p> <p><設定規則></p> <p>30 文字まで指定できます。</p> <p>2 バイト文字、及び半角の片仮名は使用しないでください。</p> <p>DABroker では、指定した環境変数グループが有効かどうかはチェックしません。</p>

注※

アプリケーションでユーザ ID、パスワードの指定のない場合にここで指定されたユーザ ID、パスワードを使用してデータベースにアクセスします。また、DABroker では、接続先データベース定義ファイル出力時にパスワードを暗号化しないため、必要がない場合、パスワードはアプリケーション側で指定してください。

(2) ORACLE を使用する場合

設定項目名	指定内容
データベース種別名	<p>使用するデータベースに対して付ける任意名称</p> <p><設定規則></p> <p>接続先データベース定義ユーティリティ内でユニークな名称にしてください。23 文字まで指定できます。2 バイト文字、半角カタカナ文字は使用しないでください。</p>
データベース名	<p>データベースごとに異なる情報の集まりに対して付ける任意名称</p> <p><設定規則></p> <p>同一データベース種別名内でユニークな名称にしてください。23 文字まで指定できます。2 バイト文字、半角カタカナ文字は使用しないでください。</p>
リスナー名 (DBNAME)	<p>SQL*NET 経由でアクセスする ORACLE のリスナー名称</p> <p>ただし、リモートアクセスしない場合は省略します。</p>
ユーザ ID(USERID)	接続ユーザ ID※
パスワード(PASSWD)	パスワード※

注※

アプリケーションでユーザ ID、パスワードの指定のない場合にここで指定されたユーザ ID、パスワードを使用してデータベースにアクセスします。また、DABroker では接続先データベース定義ファイル出力時にパスワードを暗号化しないため、必要がない場合はパスワードはアプリケーション側で指定してください。

(3) SQL Anywhere を使用する場合

設定項目名	指定内容
データベース種別名	使用するデータベースに対して付ける任意名称 <設定規則> 接続先データベース定義ユーティリティ内でユニークな名称にしてください。23文字まで指定できます。2バイト文字、半角カタカナ文字は使用しないでください。
データベース名	データベースごとに異なる情報の集まりに対して付ける任意名称 <設定規則> 同一データベース種別名内でユニークな名称にしてください。23文字まで指定できます。2バイト文字、半角カタカナ文字は使用しないでください。
接続情報(DBHOST)	SQL Anywhere に渡す接続情報 接続情報のキーワードは ; (セミコロン) で区切ります。
ODBC のデータソース名 (DBNAME)	ODBC 接続のデータソース名
ユーザ ID(USERID)	接続ユーザ ID*
パスワード(PASSWD)	パスワード*

注※

アプリケーションでユーザ ID、パスワードの指定のない場合にここで指定されたユーザ ID、パスワードを使用してデータベースにアクセスします。また、DABroker では接続先データベース定義ファイル出力時にパスワードを暗号化しないため、必要がない場合はパスワードはアプリケーション側で指定してください。

(4) ODBC3.0 を使用する場合

設定項目名	指定内容
データベース種別名	使用するデータベースに対して付ける任意名称 <設定規則> 接続先データベース定義ユーティリティ内でユニークな名称にしてください。23文字まで指定できます。2バイト文字、半角カタカナ文字は使用しないでください。
データベース名	データベースごとに異なる情報の集まりに対して付ける任意名称 <設定規則> 同一データベース種別名内でユニークな名称にしてください。23文字まで指定できます。2バイト文字、半角カタカナ文字は使用しないでください。
データベース	次のどちらかを選択します。 <ul style="list-style-type: none">• SQLAnywhere : Adaptive Server Anywhere• SQLServer : SQL Server
接続情報(DBHOST)	ODBC3.0 に渡す接続情報 接続情報のキーワードは ; (セミコロン) で区切ります。

設定項目名	指定内容
ODBC のデータソース名 (DBNAME)	ODBC3.0 のデータソース名
ユーザ ID(USERID)	接続ユーザ ID*
パスワード(PASSWD)	パスワード*

注※

アプリケーションでユーザ ID、パスワードの指定のない場合にここで指定されたユーザ ID、パスワードを使用してデータベースにアクセスします。また、DABroker では接続先データベース定義ファイル出力時にパスワードを暗号化しないため、必要がない場合はパスワードはアプリケーション側で指定してください。

(5) RDA Link for Gateway 経由でメインフレーム系データベースを使用する場合

設定項目名	指定内容
データベース種別名	使用するデータベースに対して付ける任意名称 <設定規則> 接続先データベース定義ユーティリティ内でユニークな名称にしてください。23 文字まで指定できます。2 バイト文字、半角カタカナ文字は使用しないでください。
データベース名	データベースごとに異なる情報の集まりに対して付ける任意名称 <設定規則> 同一データベース種別名内でユニークな名称にしてください。23 文字まで指定できます。2 バイト文字、半角カタカナ文字は使用しないでください。
データベース	次のどれかを選択します。 <ul style="list-style-type: none"> • XDM/RD : VOS3 XDM/RD • RDB1 : VOS1 RDB1 E2 • SQL/K : VOSK SQL/K
分散定義名(DBNAME)	RDA Link for Gateway の分散定義名称
ユーザ ID(USERID)	接続ユーザ ID*
パスワード(PASSWD)	パスワード*

注※

アプリケーションでユーザ ID、パスワードの指定のない場合にここで指定されたユーザ ID、パスワードを使用してデータベースにアクセスします。また、DABroker では接続先データベース定義ファイル出力時にパスワードを暗号化しないため、必要がない場合はパスワードはアプリケーション側で指定してください。

(6) Database Connection Server 経由でメインフレーム系データベースを使用する場合

設定項目名	指定内容
データベース種別名	使用するデータベースに対して付ける任意名称 <設定規則> 接続先データベース定義ユーティリティ内でユニークな名称にしてください。23文字まで指定できます。2バイト文字、半角カタカナ文字は使用しないでください。
データベース名	データベースごとに異なる情報の集まりに対して付ける任意名称 <設定規則> 同一データベース種別名内でユニークな名称にしてください。23文字まで指定できます。2バイト文字、半角カタカナ文字は使用しないでください。
データベース	次のどれかを選択します。 <ul style="list-style-type: none"> • XDM/RD : VOS3 XDM/RD • XDM/SD : VOS3 XDM/SD • SQL/K : VOSK SQL/K • VOSKSPPOOL : VOSK スプールファイル • RDB1 : VOS1 RDB1 E2 • VOS1PDM2 : VOS1 PDMII E2 使用するアプリケーション開発用インタフェースによって、アクセスできるデータベースが異なります。詳細については、各 DABroker 関連製品、又は DBPARTNER のマニュアルを参照してください。
コネクションマルチ名 (DBHOST)	コネクションマルチ名 コネクションマルチ定義ユーティリティ (「4.7 DABroker コネクションマルチ定義 (Database Connection Server 経由の DB アクセス定義)」参照) で設定したコネクションマルチ名を指定してください。
分散定義名 (DBNAME)	分散定義名称 接続する Database Connection Server のデータベース識別子 (RD ノード名称)
ユーザ ID (USERID)	接続ユーザ ID*
パスワード (PASSWD)	パスワード*
送信バッファサイズ (SNDBUF)	Database Connection Server 側へ送信するデータバッファのサイズ(64~2097151 キロバイト) なお、ここでの指定を省略した場合又はコネクションマルチ機能 (「4.7 DABroker コネクションマルチ定義(Database Connection Server 経由の DB アクセス定義)」参照) を使用する場合は、DABroker 環境設定ユーティリティで設定した値が有効になります。
受信バッファサイズ (RCVBUF)	Database Connection Server 側から受信するデータバッファのサイズ(64~2097151 キロバイト) なお、ここでの指定を省略した場合又はコネクションマルチ機能 (「4.7 DABroker コネクションマルチ定義(Database Connection Server 経由の DB アクセス定義)」参照) を使用する場合は、DABroker 環境設定ユーティリティで設定した値が有効になります。

注※

アプリケーションでユーザ ID、パスワードの指定のない場合にここで指定されたユーザ ID、パスワードを使用してデータベースにアクセスします。また、DABroker では接続先データベース定義ファイル出力時にパスワードを暗号化しないため、必要がない場合はパスワードはアプリケーション側で指定してください。

(7) 注意事項

- 接続先データベース定義ユーティリティで複数定義したデータベース種別名がユニークでも、実際のデータベース種別が同じ場合、同じプログラムから同時に接続できません。

4.7 DABroker コネクションマルチ定義(Database Connection Server 経由の DB アクセス定義)

Database Connection Server を経由してメインフレーム系データベースにアクセスする場合、アプリケーション又は DBPARTNER は、接続先データベース定義ユーティリティ（「4.6 DABroker 接続先データベース定義」参照）に定義されたコネクションマルチ名を使用してアクセスします。コネクションマルチ名とは、接続する Database Connection Server のポート番号やホスト名などを定義した集まりに対して付ける任意名称で、コネクションマルチ定義ユーティリティで設定します。

コネクションマルチ定義ユーティリティで変更した内容は、DABroker の場合、コネクションマルチを再起動することで有効になります。アプリケーションの場合は、プロセスを再起動することで有効になります。

ここでは、コネクションマルチ定義ユーティリティを使ったコネクションマルチ名の設定方法について説明します。

4.7.1 コネクションマルチ名の設定

一つのコネクションマルチ名には、次の内容を定義します。

- Database Connection Server のホスト名 (1)
- Database Connection Server のポート番号又はサービス名 (2)
- Database Connection Server からの応答監視時間 (3)
- 最大仮想コネクション数 (4)
- 通信処理プロセス数 (5)
- プールコネクション数 (6)
- 予約プールコネクション数 (7)

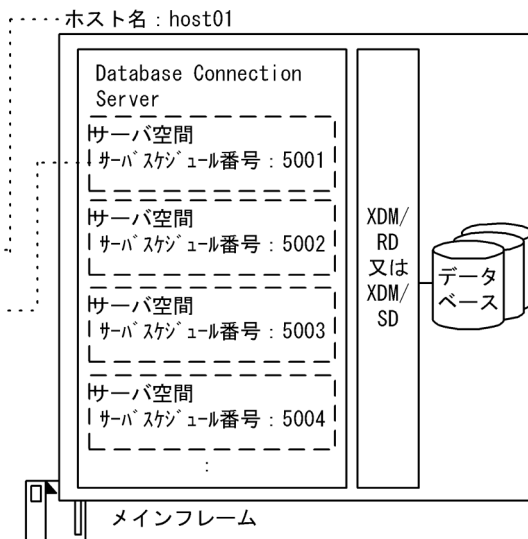
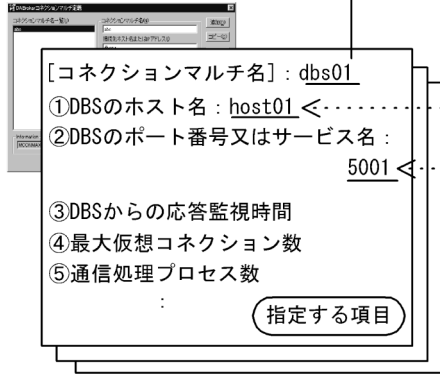
図 4-3 に「コネクションマルチ名」と(1)(2)の指定内容の関係を示します。

図 4-3 指定内容の関係

[DABroker接続先データベース定義]
ダイアログボックス



[DABroker接続マルチ定義]
ダイアログボックス



(表記) DBS : Database Connection Server

図 4-3 で示すように、Database Connection Server のホスト名 (1) には、Database Connection Server 及びアクセスするメインフレーム系データベースが存在するホスト名を指定します。この例ではホスト名を指定していますが、IP アドレスを指定することもできます。Database Connection Server のポート番号又はサービス名には、Database Connection Server のサーバ空間*に割り当てられているサーバスケジュール番号 (マニュアル「Database Connection Server」参照) を設定します。この例ではポート番号を指定していますが、サービス名を指定することもできます。サービス名を指定するときは、[Windows インストールディレクトリ]¥SYSTEM32¥DRIVERS¥ETC¥SERVICES ファイルに、サービス名とそれに対応するポート番号を設定してください ([4.3 ネットワーク環境の設定] 参照)。

注※ プロセスに相当するメインフレーム側の処理単位です。

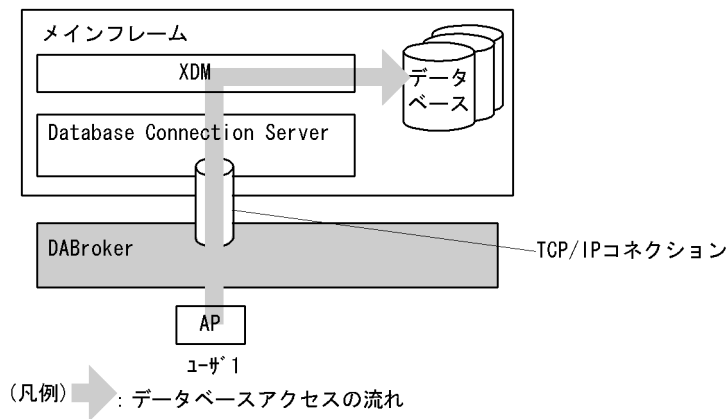
4.7.2 最大仮想接続数などを設定するときのガイドライン

最大仮想接続数 (4)、通信処理プロセス数 (5)、プール接続数 (6)、予約プール接続数 (7) は、使用環境に合わせた適切な値を設定する必要があります。ここでは、設定するときのガイドラインを説明します。

(1) DABroker と Database Connection Server の接続の仕組み

DABroker と Database Connection Server の接続時、Database Connection Server では、TCP/IP コネクションを確立します。1 ユーザからのアクセス要求に対して一つの TCP/IP コネクションを確立します (図 4-4 参照)。

図 4-4 DABroker と Database Connection Server の接続イメージ(1)



この場合、1TCP/IP コネクションを 1 ユーザからのアクセス要求だけで使用でき、レスポンス性能が良くなりますが、同時アクセスユーザ数が増えると、TCP/IP コネクションの確立数も多くなり Database Connection Server のメインフレーム側及び DABroker 側でのリソース消費量が多くなってしまいます。

そのため、DABroker では、一つの TCP/IP コネクションを複数ユーザからのアクセス要求で共有できるように、一つの TCP/IP コネクションのなかで複数の DABroker 用のコネクションを確立できるようにしています。一つの TCP/IP コネクションに対して確立する DABroker 用のコネクションを**仮想コネクション**と呼びます。一つの TCP/IP コネクションの中で複数の仮想コネクションを確立するときには、DABroker では、**通信処理プロセス**を起動します。通信処理プロセスは、TCP/IP コネクションごとに起動します。

1TCP/IP コネクションを複数ユーザからのアクセス要求で共有する場合、DABroker では、1 通信処理プロセス当たりで (=一つの TCP/IP コネクションに対して) 同時に確立できる仮想コネクション数 (=最大仮想コネクション数)、及び DABroker 内で同時に起動できる通信処理プロセス数 (=通信処理プロセス数) を決めて運用します。

図 4-5 に、1TCP/IP コネクションを 2 ユーザからのアクセス要求で共有する場合 (1 通信処理プロセス当たりで二つの仮想コネクションを確立する場合) の、DABroker と Database Connection Server の接続イメージを示します。ここでは、アプリケーション及び DBPARTNER から同時に四つのアクセス要求があり、二つの TCP/IP コネクションで対応しています。DABroker では、二つの通信処理プロセスを起動し、それぞれの通信処理プロセスで二つの仮想コネクションを確立しています。このとき、四つの TCP/IP コネクションを使用した場合に比べ、メインフレーム側及び DABroker 側の両方でリソース消費量を少なくできます。

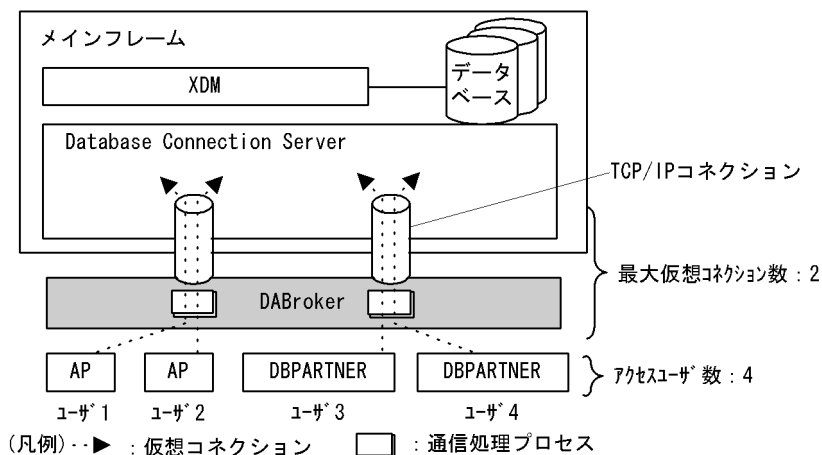
DABroker は、次のタイミングで Database Connection Server に対して TCP/IP コネクションを確立します。

- 「予約プールコネクション数」が設定されている (>0) 場合
DABroker コネクションマルチ機能起動時に確立します。
- 「予約プールコネクション数」が設定されていない (=0) 場合
最初のユーザアクセス要求時に確立します。

- 起動中の通信処理プロセスで実行できるユーザ数(「最大仮想コネクション数」)を超えたユーザアクセスが要求された場合
通信処理プロセスを新しく起動して確立します。
(ただし、「通信処理プロセス数」の範囲内)

そして、これらの TCP/IP コネクションを確立する際に、1TCP/IP コネクション上に「最大仮想コネクション数」に設定された数だけ仮想コネクションを予約します。

図 4-5 DABroker と Database Connection Server の接続イメージ(2)



なお、一つの TCP/IP コネクションに対して一つの仮想コネクションを確立する形態をシングルコネクション (図 4-4 参照) と呼び、一つの TCP/IP コネクションに対して複数の仮想コネクションを確立する形態をコネクションマルチ (図 4-5 参照) と呼びます。

(2) 最大仮想コネクション数と通信処理プロセス数の決定

最大仮想コネクション数 (4)

次のどちらかの値を設定します。

- 1TCP/IP コネクションを 1 ユーザからのアクセス要求だけで使用する場合 (シングルコネクション)
最大仮想コネクション数を「1」に設定します。この場合、TCP/IP コネクションを 1 ユーザからのアクセス要求だけで使用できるためレスポンス性能が良いのですが、複数ユーザからの同時アクセス要求があるときには同時アクセス要求数分の TCP/IP コネクションを確立し、DABroker 及びメインフレームでのリソース消費量が多くなります。
- 1TCP/IP コネクションを複数ユーザからのアクセス要求で共有する場合 (コネクションマルチ)
「2 以上の値」で、1 通信処理プロセス当たりで同時に確立できる仮想コネクション数を設定します。設定値については、下記を参照してください。なお、この場合は、運用時に、コネクションマルチ機能の運用開始ユーティリティ (「4.10 コネクションマルチ機能の使用」参照) を実行してください。

通信処理プロセス数 (5)

DABroker で同時に起動できる通信処理プロセス数の最大値を設定します。設定値については、下記を参照してください。なお、最大仮想コネクション数が「1」の場合は、設定する必要はありません。

1TCP/IP コネクションを複数ユーザからのアクセス要求で共有する場合、これらの値は、次の手順で決定してください。

1. DABroker のシステム全体で同時にアクセスするユーザ数 (要求数) の最大値を見積もる

この最大値は、アクセス対象となる Database Connection Server の空間起動制御文で設定された「サーバユーザ数」以内となるようにしてください。サーバユーザ数の設定方法については、マニュアル「Database Connection Server」を参照してください。

2. 「1 で見積もった最大値 ≤ 最大仮想コネクション数 * 通信処理プロセス数」となるような最大仮想コネクション数と最大通信処理プロセス数の組み合わせを決定する

このとき、次の 2 点を考慮し、使用環境に合わせた適切な数値を検討してください。

- レスポンス性能
- DABroker が組み込まれたサーバで消費するリソース量、及び Database Connection Server が組み込まれたメインフレームで消費するリソース量

レスポンス性能は、1 通信処理プロセス当たりの仮想コネクション数を少なくすればするほど向上し、最大仮想コネクション数を 1 にする (シングルコネクション) とレスポンスが早くなります。しかし、同時アクセスユーザ数が多い場合に、TCP/IP コネクションも増えるため、メインフレーム側でリソース消費量が多くなります。このため、使用環境に合わせた適切な数値を検討してください。

なお、最大仮想コネクション数は、接続する Database Connection Server の空間起動制御文で設定された「ユーザ数(仮想ユーザ空間数など) ÷ TCP コネクション数」の倍数となるように設定してください。

(3) プールコネクション数と予約プールコネクション数の決定 (仮想コネクションのプーリング機能の使用)

最大仮想コネクション数を 2 以上にした場合、次の 2 種類の機能が使用できます。この 2 種類は併用できます。

• 一度確立した仮想コネクションをプールする

この機能を使用すると、コネクション確立のためのオーバーヘッドを削減し、Database Connection Server との接続時間の短縮が図れます。この機能を使用するときは、**プールコネクション数 (6)** を設定してください。

• コネクションマルチ機能の起動時に指定された数の仮想コネクションを確立しておく

この機能を使用すると、通信処理プロセスの起動のためのオーバーヘッドを削減し、Database Connection Server との接続時間を短縮できます。この機能を使用するときは、**予約プールコネクション数 (7)** を設定してください。

仮想コネクションのプーリング機能を使用するときは、次に示す値を設定してください。

プールコネクション数 (6)

DABroker のシステム全体でプールする仮想コネクション数を設定します。プールした仮想コネクションに対応する TCP/IP コネクションのためのリソースを Database Connection Server (メインフレーム) 側で占有してしまうため、メインフレーム側でのリソース消費量を考慮して使用環境に合わせた値を設定してください。なお、リソースは、1 通信処理プロセスごと、1 TCP/IP コネクションごとに消費されるため、「最大仮想コネクション数」で設定した値の倍数をプールコネクション数に設定することをお勧めします。

予約プールコネクション数 (7)

DABroker のシステム全体でコネクションマルチ機能起動時に確立しておく仮想コネクション数を設定します。コネクションマルチ機能起動時にアクセス要求がなくても仮想コネクションを確立してしまうことになるため、DABroker を組み込んだサーバで確立した分のリソースを占有してしまいます。したがって、DABroker 側のリソース消費量を考慮して使用環境に合わせた値を設定してください。なお、リソースは、1 通信処理プロセスごと、1 TCP/IP コネクションごとに消費されるため、「最大仮想コネクション数」で設定した値の倍数を予約プールコネクション数に設定することをお勧めします。

プールコネクション数及び予約プールコネクション数の両方に値を設定する場合は、「プールコネクション数 \geq 予約プールコネクション数」となるように設定してください。

4.7.3 コネクションマルチ定義ユティリティの操作

コネクションマルチ定義ユティリティは、プログラムフォルダに登録された [コネクションマルチ定義] から起動します。

コネクションマルチ定義ユティリティで設定した内容は、コネクションマルチ定義ファイル：[DABroker 運用ディレクトリ¥conf¥damconenv] に出力されます。

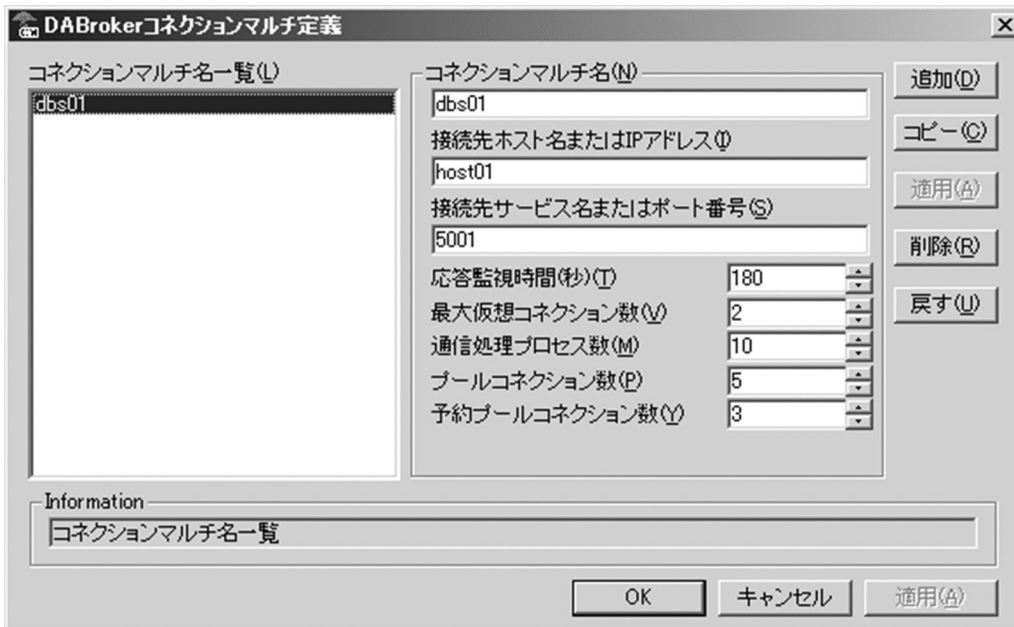
(1) 操作できるユーザ

すべてのユーザ

(2) 操作方法

1. プログラムフォルダに登録された [コネクションマルチ定義] を起動する。

コネクションマルチ定義ユティリティが起動し、[コネクションマルチ定義] ダイアログボックスが表示されます。



2. コネクションマルチ名, 及びコネクションマルチ名に対する接続情報を定義する。

設定内容については、「4.7.4 設定する項目」を参照してください。

<設定方法>

- コネクションマルチ名を新規登録する場合

[コネクションマルチ名] のテキストボックスに新しいコネクションマルチ名を定義します。既に登録されているコネクションマルチ名がある場合は、[追加] ボタンを選択してからコネクションマルチ名を定義します。

- 既にあるコネクションマルチ名の内容をコピーして新規登録する場合

コネクションマルチ名一覧からコピー元のコネクションマルチ名を選択します。コピー元の内容を表示してから [コピー] ボタンを選択します。次に [コネクションマルチ名] のテキストボックスに新しいコネクションマルチ名を定義します。

- 登録されているコネクションマルチ名の定義を変更する場合

コネクションマルチ名一覧から変更するコネクションマルチ名を選択します。

内容を変更後、[適用] ボタンを選択すると、編集した内容が保存されます。[戻す] ボタンを選択すると、編集前の状態に戻ります。

- コネクションマルチ名を削除する場合

コネクションマルチ名一覧から削除するコネクションマルチ名を選択して [削除] ボタンを選択してください。

4.7.4 設定する項目

コネクションマルチ定義ユーティリティでは、次の項目を設定します。

設定項目名	指定内容
コネクションマルチ名	接続する Database Connection Server に付ける任意名称を設定します。 <設定規則>DABroker のシステム内でユニークな名称にしてください。最大 255 文字まで指定できます。2 バイト文字、半角カタカナは使用しないでください。
Database Connection Server のホスト名	接続する Database Connection Server が存在するホスト名を設定します。 ここでの指定を省略した場合は、DABroker 環境設定ユーティリティの Database Connection Server 接続設定(接続先ホスト名または IP アドレス：DAB_DBS_HOST) に指定された値を仮定します。
Database Connection Server のポート番号又はサービス名	接続する Database Connection Server のポート番号 (Database Connection Server ではサーバスケジューラ番号と呼ぶ) 又はサービス名 (ユーザの任意設定名称) を設定します。 ここで、サービス名を指定した場合は、[Windows インストールディレクトリ] ¥SYSTEM32¥DRIVERS¥ETC¥SERVICES ファイルにサービス名と対応するポート番号を設定してください ([4.3 ネットワーク環境の設定] 参照)。 ここでの指定を省略した場合は、DABroker 環境設定ユーティリティの Database Connection Server 接続設定(接続先サービス名またはポート番号：DAB_DBS_PORT) に指定された値を仮定します。
Database Connection Server からの応答監視時間	Database Connection Server からの処理結果の応答待ち時間を設定します。 指定範囲：0, 1~65535 (秒) 0 を指定した場合：応答待ち時間を監視しません。
最大仮想コネクション数	1 通信処理プロセス当たりで同時に確立できる仮想コネクション数の最大値を設定します。ここで設定した値を超えるユーザからの要求があった場合は、別の通信処理プロセスを起動し、仮想コネクションを確立して Database Connection Server に接続します。なお、データベースには、「最大仮想コネクション数 × 通信処理プロセス数」で算出される数のユーザが同時にアクセスできます。 指定範囲：1~100 <ul style="list-style-type: none"> • 1 を設定した場合： シングルコネクションとなります。 • 2 以上を設定した場合： 1 通信処理プロセスにつき 2 以上の仮想コネクションを同時に確立できるマルチコネクション (コネクションマルチ) となります。このとき、「通信処理プロセス数」も設定してください。運用時は、コネクションマルチ機能の運用開始ユーティリティを実行してください。実行方法については、「4.10 コネクションマルチ機能の使用」を参照してください。
通信処理プロセス数 - 最大仮想コネクション数を 2 以上の値にした場合に必ず設定する項目です。	Database Connection Server と接続するとき起動する最大通信処理プロセス数を設定します。ここで設定した値を超えた場合は、アプリケーション又は DBPARTNER にエラーメッセージを返し、処理を受け付けません。 指定範囲：0, 1~1000 指定を省略した場合又は 0 を指定した場合：シングルコネクション (1 通信処理プロセス当たり 1 仮想コネクション) となります。
プールコネクション数 - 最大仮想コネクション数を 2 以上の値にした場合に設定できる項目です。	Database Connection Server に接続するとき確立する仮想コネクションを、データベースアクセスが終了しても、確立されたままの状態プールしておく数を設定します。確立済みの仮想コネクションをプールしておくことで、Database Connection Server と接続するときの仮想コネクション確立時間を短縮できます。 指定範囲：0, 1~32767

設定項目名	指定内容
	<p>指定を省略した場合又は 0 を指定した場合：プール機能を使用できません。</p>
<p>予約プールコネクション数 ー最大仮想コネクション数を 2 以上の値にした場合に設定できる項目です。</p>	<p>DABroker 起動時に仮想コネクションを確立しておく数を設定します。DABroker 起動時に仮想コネクションが確立されていると、Database Connection Server と接続するときの仮想コネクション確立時間を短縮できます。</p> <p>指定範囲：0, 1～32767</p> <p>指定を省略した場合又は 0 を指定した場合：予約プール機能を使用できません。</p> <p>注意：</p> <ul style="list-style-type: none"> • 「最大仮想コネクション数 × 通信処理プロセス数」で算出される値以上の数を設定できません。設定した場合は、コネクションマルチ機能の運用を開始した時点で、通信処理プロセスの起動エラーとなります。 • プールコネクション数も設定した場合、予約プールコネクション数では、プールコネクション数より大きな値を設定できません。

4.8 DABroker の起動と終了

DABroker の起動と終了方法について説明します

4.8.1 起動

(1) 操作できるユーザ

Windows の Administrators グループに属するユーザ

(2) 起動順序

DABroker を起動するときは次の順序で各プログラムを起動してください。

1. 接続するデータベースの DBMS
2. DABroker
3. DBPARTNER, HITSENSER5, DBPARTNER Library, 又は XDM/RD 外部表（それぞれの製品を使用している場合です）

1.と 2.の順序は前後してもかまいません。

(3) 起動方法

次のどちらかの方法で起動できます。

(a) プログラムフォルダからの起動

[スタート] - [プログラム] で表示されるフォルダ [DABroker] から [起動] を選択してください。

(b) サービスの開始による起動

[コントロールパネル] ダイアログボックスで [サービス] を選択します。[サービス] ダイアログボックスで [DABroker] を選択し、[開始] ボタンをクリックしてください。

[スタートアップの種別] を [自動] にすると、Windows の起動と同時に DABroker を起動できます。

4.8.2 終了

(1) 操作できるユーザ

Windows の Administrators グループに属するユーザ

(2) 終了順序

DABroker を終了するときは次の順序で各プログラムを終了してください。

1. DBPARTNER, HITSENSER5, DBPARTNER Library, 又は XDM/RD 外部表（それぞれの製品を使用している場合です）
2. DABroker
3. 接続するデータベースの DBMS

(3) 終了形態

DABroker の終了形態には次の三つがあります。

• 通常の終了

次の場合に DABroker を終了できます。

- クライアントが存在せず、かつ実行中又は実行待ちの予約処理がない場合
- 実行予約を設定したエージェントがない場合

この条件を満たさない場合はエラーメッセージが表示され、終了できません。

接続中のクライアント、又は受け付けている予約処理があるかどうかは、DABroker 管理ユティリティで確認できます。

• 計画終了

接続中のクライアントが存在する、又は予約処理がある場合、これらのすべての処理の終了を待って DABroker を終了します。新たなクライアントからの接続要求、及び接続中のクライアントからの新たな予約処理は受け付けません。

• 強制終了

接続中のクライアントが存在する、又は予約処理があっても、これらの処理を中断して DABroker を強制的に終了します。

計画終了中に [DABroker 終了] ダイアログボックスからこの強制終了を実行すると、計画終了を強制終了に変更できます。

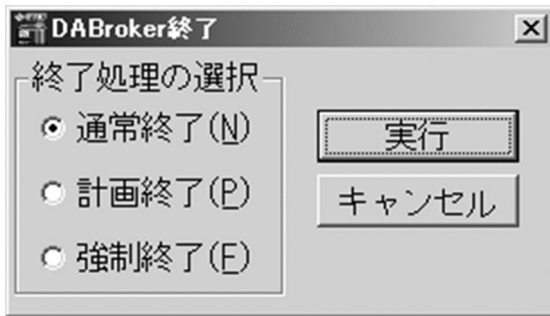
(4) 終了方法

起動方法にかかわらず、次のどちらかの方法で終了できます。

(a) プログラムフォルダからの終了

[スタート] - [プログラム] で表示されるフォルダ [DABroker] から [終了] を選択してください。[DABroker 終了] ダイアログボックスが表示されるので、終了形態を選択し、[実行] ボタンをクリックします。

その後、確認のためのダイアログボックスが表示されます。[はい] を選択して DABroker を終了します。



(b) サービスの停止による終了

[コントロールパネル] ダイアログボックスで [サービス] を選択します。[サービス] ダイアログボックスで [DABroker] を選択し、[停止] ボタンをクリックしてください。

ただし、この場合は強制終了になります。

4.9 DABroker の処理内容の表示

DABroker では、次の情報を取得できます。

4.9.1 DABroker 管理ユーティリティの実行

DABroker 管理ユーティリティには次の二つの機能があります。

- **DABroker の動作状態の表示**
各クライアントとの接続時間,及び予約処理の状態を表示します。
- **特定のクライアント（特定の IP アドレス又はホスト名）の処理の中断**
指定したクライアントとのコネクションを解放し、実行中の処理を中断します。このとき、DABroker はデータベースに要求した処理のロールバックを要求します。また、そのクライアントが要求した予約処理をすべて取り消します。

(1) 操作できるユーザとユーティリティの起動・終了方法

操作できるユーザ

操作する環境が Windows 2000, Windows Server 2003 または Windows XP の場合、すべてのユーザが管理ユーティリティを実行できます。

ただし、特定のクライアントの処理を中断できるのは、Windows の Administrators グループに属するユーザだけです。

操作する環境が Windows Vista, Windows 7, または Windows Server 2008 の場合は、Administrators グループに属するユーザだけ管理ユーティリティを実行、および特定のクライアントの処理を中断できます。

ユーティリティの起動と終了方法

起動

DABroker 管理ユーティリティは、プログラムフォルダに登録された [DABroker] から [DABroker 管理] を選択して起動します。DABroker が起動中に実行してください。

終了

[DABroker 管理] ダイアログボックスを閉じると DABroker 管理ユーティリティは終了します。

(2) 動作状態の表示方法

DABroker 管理ユーティリティを起動すると、次に示す [管理] ダイアログボックスが表示されます。ダイアログボックスに表示されている内容が、DABroker の動作状態です。

識別子	IPアドレス	接続時刻	接続時間	予約	状態	ホスト名
5444	10.209.30.67	2010/11/08 09:55:42	0:00:00	0	接続中	R15444.softhitachi.co.jp

ここでは、表示されている内容について説明します。

- IP アドレス

接続しているクライアントの IP アドレス。

- 接続時刻

クライアントと接続した日時（年／月／日 時：分：秒）

- 接続時間

接続してからの経過時間（時：分：秒）

- 予約

実行中の予約処理数。

- 状態

クライアントとの接続の確立状態。

「接続中」と表示されている場合は、クライアントとの接続が確立中であることを示します。何も表示されていない場合は、予約処理は実行中であるが、クライアントとの接続は解放されていることを示します。

- ホスト名


接続しているクライアントのホスト名。

次のフォルダの hosts ファイルにホスト名を登録している場合、ホスト名が表示されます。

[Windows をインストールしたフォルダ]¥SYSTEM32¥DRIVERS¥ETC

ホスト名の長さによっては、ホスト名の後方が表示できない場合があります。

アイコンについて

 アイコンをクリックすると最新の動作状態が表示されます。


注意事項

実行中の予約処理がある場合は、接続が解放されていてもクライアントの情報が出力されます。予約処理がない場合、又は予約処理があっても終了している場合は、接続を解放しているクライアントの情報は表示されません。

(3) 特定のクライアントの処理の中断方法

特定のクライアントの処理を中断したい場合は、DABroker の動作状態を表示し（「4.9.1(2)動作状態の表示方法」参照）、次に示す手順を実行してください。

1. [DABroker 管理] ダイアログボックスでクライアントを選択する。

動作状態を表示し、その中から処理を中断するクライアントを選択して  アイコンをクリックします。

2. [はい] ボタンを選択する。

確認のダイアログボックスが表示されるので、[はい] ボタンを選択すると、DABroker は選択したクライアントとの接続を解放します。同時にデータベースに処理のロールバックを要求して、そのクライアントが要求したすべての予約処理を取り消します。

[いいえ] ボタンについて

[いいえ] ボタンを選択すると、クライアントの処理の中断はキャンセルされます。

4.9.2 DABroker トレース出力ユティリティの実行

DABroker トレース出力ユティリティでは、次の二つのトレースを出力できます。

• 通信トレース

DABroker とクライアント間のデータの送受信に関する情報（通信トレース）を出力します。通信トレースはクライアントからの要求受付直後、及び DABroker からの結果送信直後に DABroker が取得します。出力されたトレースは、DABroker 運用ディレクトリ¥spool¥dabcltrc の下に格納されます。

• データベースアクセストレース

DABroker がデータベースに要求した処理、及びその処理に対するデータベースからの応答（アクセストレース）を表示します。アクセストレースは、DBMS に要求を出す直前及び DBMS から応答が返ってきた直後に、DABroker が取得します。出力されたトレースは、DABroker 運用ディレクトリ ¥spool¥db_access の下に格納されます。

また、DABroker では、拡張データベースアクセストレース機能でデータベースアクセストレースを出力することもできます。拡張データベースアクセストレース機能の詳細については、「6.5.2 拡張データベースアクセストレース機能」を参照してください。

(1) 操作できるユーザとユティリティの起動・終了方法

操作できるユーザ

すべてのユーザが DABroker トレース出力ユティリティを実行できます。

ユーティリティの起動と終了方法

起動

DABroker トレース出力ユーティリティは、プログラムフォルダに登録された [DABroker] から [トレース出力] を選択して起動します。DABroker 起動中に実行してください。

起動すると次のダイアログが表示されます。



終了

[DABroker トレース出力] ダイアログボックスを閉じると DABroker トレース出力ユーティリティは終了します。

(2) クライアントとの通信トレースの出力方法

[DABroker トレース出力] ユティリティを起動して表示されたダイアログで、次に示す手順を実行してください。

1. [通信トレースを出力する] チェックボックスをチェックする。
2. 取得する通信トレースを指定する。
 - すべての通信トレースを取得する場合
[IP アドレスまたはホスト名] を空白にしてください。
 - 特定の IP アドレスまたはホスト名の通信トレースを取得する
[IP アドレスまたはホスト名] に通信トレースを出力する IP アドレス又はホスト名を指定してください。
3. 通信トレースの出力先を指定する。

[出力先] ボタンをクリックすると、出力先のファイルを指定するダイアログボックスが表示されます。出力先のファイルを指定します。省略した場合は、DABroker 運用ディレクトリ¥spool¥DABTT.TXT に通信トレースが出力されます。
4. [実行] ボタンを選択する。

[実行] ボタンを選択すると指定した出力先に通信トレースが出力されます。

なお、データベースアクセストレースの出力設定（設定方法は、「4.9.2(3) データベースアクセストレースの出力方法」参照）もしてから [実行] ボタンを押すと、両方のトレースが出力できます。

出力例

DATE	TIME	EVT	SIZE	DATA
1998/12/02	11:38:31.13	PUT	32	Connection(0.0)
1998/12/02	11:39:00.25	GET	256	Connect(Database)
1998/12/02	11:38:20.56	PUT	256	Result(0.0)
1998/12/02	11:40:40.21	GET	1024	LIST (TABLE) <Prepare>
1998/12/02	11:45:23.22	PUT	1024	Result(0.0)

- **IPADDR=XXXXXXXX**

クライアントの IP アドレス又はホスト名。

- **PROT=XXX**

通信プロトコル。TCP：TCP/IP。

- **PAP-NAME=XXXXXX**

次のどれかが表示されます。

- DABroker 関連製品を使用した場合：アプリケーションで設定した名称
- DBPARTNER/Client 又は DBPARTNER/Web からアクセスした場合：DBPARTNER/Client 又は DBPARTNER/Web の識別情報
- DBPARTNER Mobile からアクセスした場合：クライアント識別情報
- DBPARTNER Library からアクセスした場合：DBPARTNER Library の識別情報
- XDM/RD 外部表としてアクセスした場合：XDM/RD の識別情報

- **DATE**

送受信の発生日付（年/月/日）。

- **TIME**

送受信の発生時刻（時:分:秒.n）。n は n/100 秒を表します。

- **EVT**

DABroker での事象区分。

- PUT：クライアントへの送信
- GET：クライアントからの受信

- **SIZE**

送受信データ長（バイト）。

- **DATA**

送受信データ種別。

aa…aa <bb…bb> (cc…cc,dd…dd) の形式で出力されます。

- aa…aa：オペレーション種別

表示されるオペレーション種別とその意味について、次に示します。

オペレーション種別	意 味
Connect (Database)	データベースとの接続要求
Disconnect (Database)	データベースとの切り離し要求
Disconnect (Connection)	コネクションの解放要求
Commit	トランザクションのコミット要求
Rollback	トランザクションのロールバック要求
LIST (Table)	表一覧取得要求
LIST (Column)	列一覧取得要求
LIST (Index)	索引一覧取得要求
LIST (Table Privileges)	表権限一覧取得要求
LIST (Column Privileges)	列権限一覧取得要求
LIST (Inquiry)	予約完了状況一覧取得要求
EXT-CMD	SQL 文の実行要求
SQL Fetch Option	SQL 実行オプション実行
File Delete	SQL 実行結果格納ファイル削除
Save	データベースへの表の保存要求
Reserve	予約処理要求
Reserve Delete	予約処理結果の削除要求
Agent (Register)	エージェントの登録要求
Agent (Delete Registration)	エージェントの削除要求
Agent (List Prepare Registration)	エージェントの登録情報の取得準備要求
Agent (List Fetch Registration)	エージェントの登録情報の取得継続要求
Agent (List Close Registration)	エージェントの登録情報取得後の処理要求
Agent (Execute)	エージェントの実行要求
Agent (List Prepare Execution)	エージェントの実行状況の取得準備要求
Agent (List Fetch Execution)	エージェントの実行状況の取得継続要求
Agent (List Close Execution)	エージェントの実行状況取得後の処理要求
Agent (Get Prepare Result)	エージェント実行結果の取得準備要求
Agent (Get Fetch Result)	エージェント実行結果の取得継続要求
Agent (Get Close Result)	エージェント実行結果の取得後の処理要求
Agent (Delete Result)	エージェントの実行結果の削除要求

オペレーション種別	意 味
Agent (Option)	エージェントのオプション実行要求
Agent (Execute Direct Prepare)	エージェントの即時配信実行準備要求
Agent (Execute Direct Fetch)	エージェントの即時配信実行継続要求
Agent (Execute Direct Close)	エージェントの即時配信実行後処理要求
Cancel	DBMS に対する処理の中断要求
Illegal Request	不正要求
Connection	コネクション確立応答
Result	正常応答
Error	異常応答
Result (Cancel)	正常応答 (DBMS に対する処理の中断要求時)
Error (Cancel)	異常応答 (DBMS に対する処理の中断要求時)
Procedure(List Prepare Procedure)	ストアードプロシジャ一覧取得準備要求
Procedure(List Fetch Procedure)	ストアードプロシジャ一覧取得要求
Procedure(List Close Procedure)	ストアードプロシジャ一覧取得後の処理要求
Procedure(List Prepare Column)	ストアードプロシジャ列一覧取得準備要求
Procedure(List Fetch Column)	ストアードプロシジャ列一覧取得要求
Procedure(List Close Column)	ストアードプロシジャ列一覧取得後の処理要求
Procedure(Parameter Get)	ストアードプロシジャ定義情報取得要求
Procedure(Prepare)	ストアードプロシジャ実行準備要求
Procedure(Execute)	ストアードプロシジャ実行要求
Procedure(Extended Execute)	ストアードプロシジャ拡張実行要求
AUTO Commit OFF	トランザクションの AUTO COMMIT モード解除要求
AUTO Commit ON	トランザクションの AUTO COMMIT モード設定要求

- bb…bb：付加情報。
オペレーション種別が、LIST (xxxxx) 及び EXT-CMD の場合に表示されます。
- cc…cc：リターンコード。
- dd…dd：詳細コード。

注意事項

クライアントからの要求量が多いほど、クライアントとの通信トレースのファイル容量も多くなります。通信トレースのファイル容量の目安として、概算式を次に示します。

(概算式)

80 (ヘッダ情報長) + クライアントからの要求数 * 2 (PUT 及び GET 分) * 36 (PUT 及び GET 情報長) [バイト]

あらかじめ、この概算式で計算したサイズ以上のハードディスク容量を用意しておいてください。また、クライアントからの要求量が多いほど、トレース出力が終わるまで時間がかかります。御注意ください。

(3) データベースアクセストレースの出力方法

[DABroker トレース出力] ユティリティを起動して表示されたダイアログで、次に示す手順を実行してください。

1. [アクセストレースを出力する] チェックボックスをチェックする。

2. 取得するアクセストレースを指定する。

- すべてのアクセストレースを取得する場合
[UAP 名], [ユーザ ID] を空白にしてください。
- 特定のアプリケーションのアクセストレースを取得する
[UAP 名] にアクセストレースを出力する UAP 名を指定してください。
指定できる文字列は次のとおりです。

アプリケーションの種類	指定する名称
DABroker 関連製品で開発したアプリケーションからアクセスする場合	アプリケーションで設定した名称 (各 DABroker 関連マニュアル参照)
DBPARTNER/Client 又は DBPARTNER/Web からアクセスする場合	DBPARTNER/Client 又は DBPARTNER/Web の識別情報
エージェント処理を実行した場合	エージェント実行名 (16 バイトの固定文字列)

- 特定のユーザ ID のアクセストレースを取得する場合
[ユーザ ID] にアクセストレースを出力するユーザ ID を指定してください。
- 特定のクライアントの特定のユーザ ID のアクセストレースを取得する場合
[UAP 名] にアクセストレースを出力する UAP 名を指定し、[ユーザ ID] にアクセストレースを出力するユーザ ID を指定してください。

3. アクセストレースの出力先を指定する。

[出力先] ボタンをクリックすると、出力先のファイルを指定するダイアログボックスが表示されます。出力先のファイルを指定します。省略した場合は、DABroker 運用ディレクトリ¥spool¥DABDT.TXT にアクセストレースが出力されます。

4. [実行] ボタンを選択する。

[実行] ボタンを選択すると指定した出力先にアクセストレースが出力されます。

すべてのクライアントから要求されたデータベースアクセストレースを出力します。

出力例

UID=HITACHI	PAPNAME=DBPARTNER/Server Driver	Driver	DBNAME=HiRDB	
DATE	TIME	EVT	RETCODE	INFORMATION
1998/04/02	11:38:30.10	PREPARE		
(SQL) SELECT * FROM ZAICO				
1998/04/02	11:38:31.22	PREPARE	0	
1998/04/02	11:38:31.30	DESCRIBE		
1998/04/02	11:38:32.43	DESCRIBE	0	
1998/04/02	11:38:32.50	DECLARE		
1998/04/02	11:38:33.70	DECLARE	0	
1998/04/02	11:38:34.00	OPEN		
1998/04/02	11:38:36.22	OPEN	0	
1998/04/02	11:38:36.50	FETCH		BLOCK COUNT (1)
1998/04/02	11:38:38.24	FETCH	0	BLOCK COUNT (1)
1998/04/02	11:38:38.70	FETCH		BLOCK COUNT (1)
1998/04/02	11:38:40.22	FETCH	100	BLOCK COUNT (0)
1998/04/02	11:38:40.50	CLOSE		
1998/04/02	11:38:42.22	CLOSE	0	
1998/04/02	11:38:50.55	COMMIT		
1998/04/02	11:38:52.74	COMMIT	0	
1998/04/02	11:42:12.32	DISCONNECT		

- **UID=XXXXXXXX**

ユーザ ID。

- **PAPNAME=XXXXXXXX**

アプリケーションを識別するための文字列が表示されます。表示される文字列は、[UAP 名]で指定できる名称です。

- **DBNAME=XXXXXXXX**

接続先データベース名。次のどれかが出力されます。

- HiRDB : HiRDB
- ORACLE7 : Oracle9i, Oracle10g, Oracle11g, もしくは Oracle12c で Oracle Call Interface 7 を使用している場合, 又は Oracle19c で Oracle Call Interface 8i を使用している場合。
- ORACLE8i : Oracle10g, Oracle11g, Oracle12c, 又は Oracle19c で Oracle Call Interface 8i を使用している場合。
- SQLAnywhere : SQL Anywhere
- ODBC3.0 : Adaptive Server Anywhere, SQL Server
- RDA_Link_GW : RDA Link for Gateway
- DBCSV : Database Connection Server
- **DATE**
アクセスの発生日 (年/月/日)。
- **TIME**
アクセスの発時刻 (時:分:秒.n)。n は n/100 秒を表します。
- **EVT**

DABroker がデータベースに要求した処理の内容。次のどれかが表示されます。

EVT の表示	処 理 内 容
CONNECT	データベースと接続します
PREPARE	SQL の前処理をします
DESCRIBE	前処理した SQL の情報を返します
DECLARE	前処理した SELECT に対してカーソルを宣言します
OPEN	カーソルを位置付けます
FETCH	行を読み込んでカーソルを進めます
CLOSE	カーソルを閉じます
EXECUTE	前処理した SQL を実行します
COMMIT	トランザクションを正常終了、更新を有効にします
ROLLBACK	トランザクションを取り消し、更新を無効にします
DISCONNECT	トランザクションを正常終了、切り離します
CANCEL	DBMS に対し、処理の中断します
LOGGING	DBMS に対し、ログ取得要求します ただし、DBMS が HiRDB の場合だけです
LOGLESS	DBMS に対し、ログレス要求します ただし、DBMS が HiRDB の場合だけです
AUTOCOMOFF	トランザクションの AUTO COMMIT モードを解除します
AUTOCOMON	トランザクションの AUTO COMMIT モードを設定します

データベースアクセストレースの取得内容は任意に選択できます。環境設定ユーティリティの [アクセストレースの種類] で選択します。アクセストレースの種類については、[4.5.1(9) データベースアクセストレースの取得 (DABSQL_TRCSIZE)] を参照してください。

• RETCODE

DABroker が要求した処理に対して、DBMS から返されるリターンコード。

- 0：正常終了。
- 0 以外：異常終了又はメッセージ付きの正常終了。

DABroker から DBMS への要求の送信を表示している行は、RETCODE は空白になります。

RDA Link for Gateway の場合は、RETCODE は空白になります。

• INFORMATION

FETCH 時の行取得要求数及び取得結果の情報。

ORACLE 又は RDA Link for Gateway の場合は、「BLOCK COUNT (XXX)」の形式で出力されま

- (SQL)

クライアントから要求のあった SQL の内容。この項目は、必要に応じて表示されます。

SQL Anywhere, Adaptive Server Anywhere, 又は SQL Server 使用時, EVT が PREPARE の場合は、次の情報が出力されます。

表示内容	説明
(SQL) ODBC:{SQLTables}	テーブル一覧情報
(SQL) ODBC:{SQLColums}	カラム一覧情報
(SQL) ODBC:{SQLTablePrivileges}	テーブル権限一覧情報
(SQL) ODBC:{SQLColumnPrivileges}	カラム権限一覧情報
(SQL) ODBC:{SQLStatistics}	インデクス一覧情報
(SQL) ODBC:{SQLProcedures}	プロシジャー一覧情報
(SQL) ODBC:{SQLProcedureColumns}	プロシジャ列一覧情報
(SQL) ODBC:{SQLPrimaryKeys}	プライマリキー一覧情報

Database Connection Server 使用時, EVT が PREPARE の場合は次の情報が出力されます。

表示内容	説明
(SQL) DBCSV:{SQLTable}	テーブル一覧情報
(SQL) DBCSV:{SQLColumns}	カラム一覧情報
(SQL) DBCSV:{SQLTablePrivileges}	テーブル権限一覧情報
(SQL) DBCSV:{SQLColumnsPrivileges}	カラム権限一覧情報
(SQL) DBCSV:{SQLStatistics}	インデクス一覧情報
(SQL) DBCSV:{ProcedureList}* [※]	外部プロシジャー一覧
(SQL) DBCSV:{ProcedureColumns}* [※]	外部プロシジャ列一覧

注※

Database Connection Server 経由で XDM/RD の外部プロシジャ機能を使用している場合に出力されます。

注意事項

クライアントからの要求量 (SQL 文) が多いほど、データベースアクセストレースのファイル容量も多くなります。トレースは、1 レコード 80 バイトで表示されます。クライアントからの要求 (SQL 文) が多い場合は、折り返して表示します。データベースアクセストレースのファイル容量の目安として、概算式を次に示します。

(概算式)

$$\{(\text{クライアントからの要求 (SQL 長など)} * 2) + 2 \lfloor \frac{\text{クライアントからの要求 SQL 長}}{73} \rfloor - 1 \} * 80 \text{ [バイト]}$$

注 下線は、SQL 長が 73 バイト以上の場合です。

あらかじめ、この概算式で計算したサイズ以上のハードディスク容量を用意しておいてください。また、クライアントからの要求量が多いほど、トレース表示が終わるまで時間がかかります。御注意ください。

4.9.3 DABroker エージェント管理ユーティリティの実行

DABroker エージェント管理ユーティリティには次の二つの機能があります。

- [エージェント登録一覧] ユティリティ

DABroker に登録されているすべてのエージェント登録情報を表示します。DBPARTNER Mobile - Client, 又は DBPARTNER/Client32 で登録を要求したエージェントの情報です。

- [エージェント実行情報] ユティリティ

エージェントの実行情報を表示します。DBPARTNER Mobile - Client, 又は DBPARTNER/Client32 から実行を要求したエージェントの情報です。

これらの情報を表示するためには、あらかじめ、DABroker 動作環境設定で次の項目を指定しておいてください。

- 作業ディレクトリ (DABTMP)
- 作業ディレクトリ (DABAGT)
- グループ管理用ディレクトリ (DABAGTGRP)

なお、グループ管理用ディレクトリ (DABAGTGRP) は、ユーザのグループ管理をしていない場合は設定不要です。

(1) エージェント登録一覧ユーティリティの実行

(a) 操作できるユーザと起動・終了方法

操作できるユーザ

すべてのユーザ。

ただし、操作する環境が Windows Vista, Windows 7, または Windows Server 2008 の場合は、Administrators グループに属するユーザだけです。

ユーティリティの操作方法

起動・操作方法

エージェント登録一覧ユーティリティはプログラムフォルダに登録された [DABroker] から [エージェント登録一覧] を選択して起動します。

起動すると [エージェント登録一覧] ダイアログボックスが表示されます。

終了方法

[エージェント登録一覧] ダイアログボックスを閉じると DABroker エージェント情報表示ユーティリティは終了します。

(b) エージェント登録情報の表示内容

[エージェント登録一覧] で表示されている内容が登録情報です。

各タイトルをクリックすると表示されている内容を昇順に並べ替えられます。さらにクリックすると今度は降順に並べ替えられます。また、タイトルの境界をドラックすると表示幅を変更できます。




登録名	登録ユーザ	登録日	実行予約日
在庫検索	USER1	2010/11/09 13:49	****/**/** **** *

ここでは、表示されている内容について説明します。

- **登録名**
エージェント名。
- **登録ユーザ**
登録したユーザ ID。
- **登録日**
登録した日付（年/月/日） 時刻（時:分）。
- **実行予約日**
実行予約日付（年/月/日） 時刻（時:分） 曜日。
実行予約を設定していない、又は実行予約を設定していない項目がある場合は、「*」が表示されます。

アイコンについて

 アイコンをクリックすると最新のエージェント登録情報一覧が表示されます。

(2) エージェント実行一覧ユーティリティの実行

(a) 操作できるユーザと起動・終了方法

操作できるユーザ

すべてのユーザ。

ただし、操作する環境が Windows Vista, Windows 7, または Windows Server 2008 の場合は、Administrators グループに属するユーザだけです。

ユーティリティの操作方法

起動・操作方法

エージェント登録一覧ユーティリティはプログラムフォルダに登録された [DABroker] から [エージェント実行一覧] を選択して起動します。

起動すると [エージェント実行一覧] ダイアログボックスが表示されます。

終了方法

[エージェント実行一覧] ダイアログボックスを閉じると DABroker エージェント情報表示ユーティリティは終了します。

(b) エージェント実行情報の表示

[エージェント実行一覧] で表示されている内容が実行情報です。

各タイトルをクリックすると表示されている内容を昇順に並べ替えられます。さらにクリックすると今度は降順に並べ替えられます。また、タイトルの境界をドラックすると表示幅を変更できます。

実行名	登録名	実行ユーザ	状態	実行日
0D244CD8D2D60000	在庫検索	USER1	RUN	2010/11/09 13:49

ここでは、表示されている内容について説明します。

- 実行名

エージェントの実行時に DABroker が割り当てる識別名（番号）です。

- 登録名

エージェント名。

- 実行ユーザ

エージェントの実行を要求したユーザ ID。


- 状態


実行状態です。次のどれかが表示されます。

- IDLE：初期状態（実行要求待ち状態）。
- TIMER-WATCH：エージェントタイマー監視中。
- RUN：処理イベントの実行中。
- NORMAL-ENDED (xxxxxx)：正常終了。
xxxxxx：警告メッセージ ID。警告メッセージ ID がある場合は表示します。
- ABNORMAL-ENDED (xxxxxx)：異常終了。
xxxxxx：エラーメッセージ ID。

- 実行日

実行日付（年/月/日） 時刻（時:分）。

 アイコンについて

 アイコンをクリックすると最新のエージェント実行情報一覧が表示されます。

4.10 コネクションマルチ機能の使用

コネクションマルチ定義ユーティリティで、「最大仮想コネクション数」を2以上に設定した場合は、コネクションマルチ機能の運用開始ユーティリティを実行してください。この場合は、コネクションマルチ管理ユーティリティを使って、Database Connection ServerとDABrokerとの接続状態を確認できます。

ここでは、コネクションマルチ機能の運用開始と終了方法、コネクションマルチ管理ユーティリティの操作方法について説明します。

4.10.1 コネクションマルチ機能の運用開始と終了

コネクションマルチ機能の運用開始方法と終了方法について説明します

(1) 運用開始

(a) 操作できるユーザ

WindowsのAdministratorsグループに属するユーザ

(b) 起動方法

次のどちらかの方法で起動できます。

- プログラムフォルダからの起動

[スタート] - [プログラム] で表示されるフォルダ [DABroker] から [コネクションマルチ起動] を選択してください。

- サービスの開始による起動

[コントロールパネル] ダイアログボックスで [サービス] を選択します。[サービス] ダイアログボックスで [DABroker Connection Multi] を選択し、[開始] ボタンをクリックしてください。

[スタートアップの種別] を [自動] にすると、Windowsの起動と同時にコネクションマルチ機能を運用開始できます。

(2) 終了

(a) 操作できるユーザ

WindowsのAdministratorsグループに属するユーザ

(b) 終了形態

コネクションマルチ機能の運用の終了形態には次の三つがあります。

- 通常の終了

次の場合にコネクションマルチ機能の運用を終了できます。

- Database Connection Server と接続中のコネクションが存在しない場合

この条件を満たさない場合はエラーメッセージが表示され、終了できません。

接続中のクライアントがあるかどうかは、DABroker コネクションマルチ管理ユーティリティで確認できます。

- **計画終了**

Database Connection Server と使用中のコネクションが存在する場合、これらのすべての処理の終了を待ってコネクションマルチ機能の運用を終了します。新たなコネクションを作成しません。

- **強制終了**

Database Connection Server と接続中のコネクションが存在しても、これらの処理を中断してコネクションマルチ機能の運用を強制的に終了します。

計画終了中に [DABroker コネクションマルチ終了] ダイアログボックスからこの強制終了を実行すると、計画終了を強制終了に変更できます。

(c) 終了方法

起動方法にかかわらず、次のどちらかの方法で終了できます。

- **プログラムフォルダからの終了**

[スタート] - [プログラム] で表示されるフォルダ [DABroker] から [DABroker コネクションマルチ終了] を選択してください。[DABroker コネクションマルチ終了] ダイアログボックスが表示されるので、終了形態を選択し、[実行] ボタンをクリックします。

その後、確認のためのダイアログボックスが表示されます。[はい] を選択して DABroker を終了します。

- **サービスの停止による終了**

[コントロールパネル] ダイアログボックスで [サービス] を選択します。[サービス] ダイアログボックスで [DABroker Connection Multi] を選択し、[停止] ボタンをクリックしてください。ただし、この場合は強制終了になります。

4.10.2 コネクションマルチ管理ユーティリティ

コネクションマルチ管理ユーティリティには次の二つの機能があります。

- **コネクションマルチ機能の運用状態の表示**

接続ユーザ ID からのアクセス要求に対応する仮想コネクション ID ごとの、DABroker と Database Connection Server との接続時間を表示します。

- **コネクションマルチ機能運用中の特定のクライアントの処理の中断**

特定の接続ユーザ ID からのアクセス要求に対して確立した仮想コネクションを解放し、実行中の処理を中断します。このとき、DABroker はデータベースに要求した処理のロールバックを要求します。

(1) 操作できるユーザとユティリティの起動・終了方法

操作できるユーザ

操作する環境が Windows 2000, Windows Server 2003 または Windows XP の場合, すべてのユーザが管理ユティリティを実行できます。

ただし, 特定のクライアントの処理を中断できるのは, Windows の Administrators グループに属するユーザだけです。

操作する環境が Windows Vista, Windows 7, または Windows Server 2008 の場合は, Administrators グループに属するユーザだけ管理ユティリティを実行, および特定のクライアントの処理を中断できます。

ユティリティの起動と終了方法

起動

DABroker 管理ユティリティは, プログラムフォルダに登録された [DABroker] から [コネクションマルチ管理] を選択して起動します。DABroker が起動中に実行してください。

終了

[コネクションマルチ管理] ダイアログボックスを閉じると DABroker 管理ユティリティは終了します。

(2) コネクションマルチ機能の運用状態の表示方法

コネクションマルチ管理ユティリティを起動すると, 次に示す [DABroker コネクションマルチ管理] ダイアログボックスが表示されます。ダイアログボックスに表示されている内容が, コネクションマルチ機能の運用状態です。

接続開始...	ユーザ開始時間	接続ユーザID	仮想コネ...	仮...	サーバIP...	デモンID	コネクションマルチ名
2010/11/09...	2010/11/09 13:00:00	K00D161	USE	1	65547	2008	DBS

ここでは, 表示されている内容について説明します。

- 接続開始時間

Database Connection Server の TCP/IP コネクションが確立された日時 (年/月/日 時:分:秒)。

- ユーザ開始時間

ユーザからのアクセス要求に対して仮想コネクションの使用を開始した日時 (年/月/日 時:分:秒)。

- 接続ユーザ ID

データベースに接続するときを使用したユーザ ID (データベースのログインユーザ ID)。

- 仮想接続の状態

仮想接続の状態を次のどちらかで示します。

- USE：仮想接続の使用済
- POOL：仮想接続のプール待機中

- 仮想接続 ID

- サーバプロセス ID


- デーモン ID

仮想接続 ID, サーバプロセス ID, デーモン ID のそれぞれは, DABroker が付与するユニークな ID。

- 接続マルチ名

接続している Database Connection Server の接続マルチ名 ([4.7 DABroker 接続マルチ定義(Database Connection Server 経由の DB アクセス定義)] 参照)。

アイコンについて

 アイコンをクリックすると最新の運用状態が表示されます。

5

DABroker の運用（共通）

この章では、DABroker の運用方法について説明します。

5.1 DABroker を運用する前に

ここでは、データベースをアクセスする前に設定及び確認しておく項目について説明します。

5.1.1 データベースにアクセスするユーザ ID の設定

データベースをアクセスする時のユーザ ID 及びパスワードを設定してください。ユーザ ID 及びパスワードは、DBMS に設定します。

5.1.2 クライアントからの接続先の指定

(1) DABroker 関連製品を使用する場合

アプリケーションは接続先データベース定義ファイルを使用して、データベースにアクセスします。接続先データベース定義ファイルについては、次の箇所を参照してください。

UNIX の場合：「[3.6 DABroker 接続先データベースの定義](#)」

Windows の場合：「[4.6 DABroker 接続先データベース定義](#)」

なお、アプリケーションでの接続先データベースの指定方法については、各 DABroker 関連製品のマニュアルを参照してください。

(2) DBPARTNER, HITSENSER5, DBPARTNER Library, 又は XDM/RD 外部表を使用する場合

クライアントからデータベースのアクセスを要求する場合は、使用する DABroker やデータベースのネットワーク上での位置を指定する必要があります。どこで指定するかを次に示します。

- サーバ (DABroker が組み込まれたマシン) のホスト名
クライアントで指定します。

- 使用するデータベース
クライアント、又はデータベースの定義で指定します。

ただし、UNIX で、HiRDB 又は ORACLE にアクセスする場合は、DABroker の動作環境定義ファイルで各データベースの設定項目を一つ指定します。クライアントでデータベースの指定が省略された場合は、ここで指定されたデータベースにアクセスします。なお、DABroker 動作環境定義ファイルで指定したデータベースとクライアントで指定されたデータベースとが異なる場合は、クライアントで指定されたデータベースにアクセスします。

各データベースの設定項目については、次の箇所を参照してください。

HiRDB の場合：「[3.5.4 HiRDB の設定](#)」

ORACLE の場合：「[3.5.5 ORACLE の設定](#)」

(3) Database Connection Server を使用する場合

Database Connection Server を使ってデータベースにアクセスするには、接続先データベース定義ファイルとコネクションマルチ定義ファイルに、接続条件などを定義しておく必要があります。接続先データベース定義ファイルについては次の箇所を参照してください。

UNIX の場合：「[3.6 DABroker 接続先データベースの定義](#)」

Windows の場合：「[4.6 DABroker 接続先データベース定義](#)」

コネクションマルチ定義ファイルについては次の箇所を参照してください。

UNIX の場合：「[3.9 DABroker コネクションマルチ定義\(Database Connection Server 経由の DB アクセス定義\)](#)」

Windows の場合：「[4.7 DABroker コネクションマルチ定義 \(Database Connection Server 経由の DB アクセス定義\)](#)」

(4) Adaptive Server Anywhere 又は SQL Server(ODBC3.0)を使用する場合(Windows 限定)

Adaptive Server Anywhere 又は SQL Server(ODBC3.0)を使ってデータベースにアクセスするには、アクセス対象のデータベースの ODBC データソースを設定しておく必要があります。設定したデータソース名を DABroker の接続先データベース定義、又は DBPARTNER の接続情報に設定することで、データベースにアクセスできます。

DBPARTNER からアクセスする場合には、アクセス対象のデータベースのデータソース名を、DABroker がインストールされているマシンのシステムデータソースに設定してください。

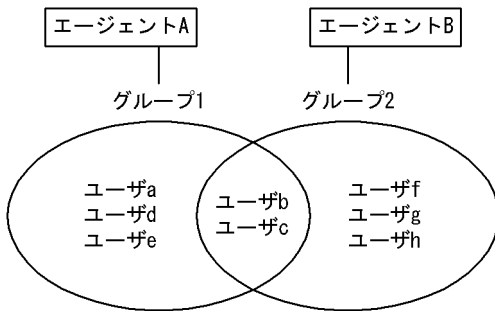
DBPARTNER の接続情報の設定については、各製品のマニュアルを参照してください。

5.1.3 エージェント機能を使用するユーザのグループ管理

エージェント機能を使用する複数ユーザをグループ化して管理できます。グループ管理すると、各エージェントの実行時にグループ外ユーザの要求を拒否できます。該当するグループ ID をエージェント登録時に指定します。

図 5-1 にグループ管理イメージを示します。

図 5-1 グループ管理イメージ



ユーザaはエージェントAを実行できるが、エージェントBは実行できない
ユーザbはエージェントA及びBを実行できる

(1) グループ管理情報ファイルの作成

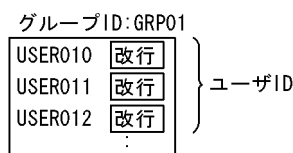
グループ管理で使用するファイルは、グループ管理情報ファイル（¥AGTGRP¥gg…gg）です。このファイルは、エージェント機能を使用する前又は該当エージェントの処理を要求する前までに、グループ ID ごとにサーバで作成します。

大文字/小文字の区別はありません。グループ ID の記述規則（属性）については、「5.4.5(1) エージェント機能のパラメーター一覧」を参照してください。

次にグループ管理情報ファイルの作成手順について示します。

1. グループ ID を用意する（グループ ID がファイル名になります）。
2. グループ管理情報ファイルに、該当エージェントを実行するユーザ ID を記述する。1 行につき、1 ユーザ ID を記述する。

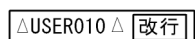
（記述例）



1 行に記述するすべての文字はユーザ ID を構成する文字として DABroker は認識します。

ユーザ ID として指定していない文字を記述した場合は、DABroker は記述したユーザ ID が存在しないと判断します。

（例）



と記述した（「USER010」の前後に空白「△」を入れた）場合、ユーザ ID は「△USER010△」と認識され、「USER010」とは別ユーザ ID となります。

3. グループ ID をつけたファイル名で、¥AGTGRP 下に格納する。

（例の場合：¥AGTGRP¥GRP01）

(2) グループ管理上の注意事項

エージェント登録時にグループ ID を設定したが、該当するグループ管理情報ファイルが存在しない場合は、エージェント実行及び実行結果取得権限を「公開しない」として扱います。

5.2 DBPARTNER を使用する場合のスレッドのチューニング

DABroker は、マルチスレッドに対応しています。DBPARTNER を使用する場合は、マルチスレッドの性能をチューニングする必要があります。

DBPARTNER を使用する場合、DABroker は、接続ユーザ（データベースにアクセスする処理）ごとにスレッドを生成します。接続ユーザ数とスレッドの対応については、「[2.1.2 DBPARTNER から実行するマルチスレッド](#)」を参照してください。

チューニングは、接続ユーザ数を考慮して DABroker 動作環境定義ファイルで次に示す項目に適切な値を指定してください。

- 最大処理プロセス数 (DABMAXPROCESS)
- 1 プロセスあたりの実行ユーザ数 (DABBLOCKUSERS)
- 最大接続ユーザ数 (DABMAXUSERS)

また、最大接続ユーザ数に設定する値は次の計算式の範囲にしてください。

(計算式)

最大接続ユーザ数 ≤ 最大処理プロセス数 × 1 プロセスあたりの最大接続ユーザ数

DABroker 動作環境定義ファイルについては、次の箇所を参照してください。

UNIX の場合：「[3.4 DABroker の動作環境の設定](#)」

Windows の場合：「[4.4 DABroker 環境設定](#)」

5.3 DBPARTNER を使用する場合のエージェントの実行権限

DBPARTNER を使用する場合、エージェントの実行内容によって、実行できるユーザが異なります。次に、DBPARTNER を使用する場合の、エージェントを実行する権限について説明します。

5.3.1 クライアントで指定するユーザ ID・パスワードと DBMS との関係

クライアントからのエージェント実行時に指定するユーザ ID・パスワードが、DBMS のユーザ ID・パスワードとなります。

5.3.2 エージェントの登録、削除

データベースアクセス権を持つユーザ ID・パスワードで DBPARTNER/Client32 又は DBPARTNER2 Client からエージェントを登録します。

エージェントを登録したユーザ ID・パスワードだけが DBPARTNER/Client32 又は DBPARTNER2 Client からエージェントを削除します。

5.3.3 エージェントの実行、実行結果の削除

データベースアクセス権を持つユーザ ID・パスワードで DBPARTNER/Client32, DBPARTNER2 Client, DBPARTNER2 Web 又は DBPARTNER Mobile からエージェントを実行します。

エージェントを実行したユーザ ID・パスワードだけが DBPARTNER Mobile 又は DBPARTNER2 Web からエージェント実行結果を削除できます。

5.3.4 エージェント実行権限の設定

エージェントの登録時に、エージェントの実行権限を設定します。エージェントの実行権限は、エージェントの公開／非公開の選択で設定します。

エージェントを公開する場合

エージェントを登録したユーザ ID・パスワード以外でも DBPARTNER/Client32 又は DBPARTNER2 Client から実行できます。

エージェントを公開しない場合

エージェントを登録したユーザ ID・パスワードだけが DBPARTNER/Client32 又は DBPARTNER2 Client から実行できます。

5.3.5 エージェント実行結果の取得権限の設定

エージェントの登録時又は実行時に、エージェント実行結果の取得権限を設定します。エージェント実行結果の公開／非公開の選択で設定します。

エージェント実行結果を公開する場合

エージェントを実行したユーザ ID・パスワード以外でも実行できます。

エージェント実行結果を公開しない場合

エージェントを実行したユーザ ID・パスワードだけが実行できます。

5.3.6 エージェントの実行権限一覧

表 5-1 及び表 5-2 に、エージェントの実行権限、及び権限の認証方法などについて一覧で示します。

表 5-1 エージェント実行権限

権限の種類	権限の所有者	権限の設定可否	設定タイミング	権限を設定できるクライアント
エージェントの登録権限	エージェント登録ユーザ ID・パスワードだけ (DB アクセスアクセス権あり)	×	—	—
エージェントの削除権限	エージェント登録ユーザ ID・パスワードだけ	×	—	—
エージェントの実行権限	エージェントを公開しない場合： エージェント登録ユーザ ID・パスワードだけ エージェントを公開する場合： エージェント登録ユーザ ID・パスワード以外のユーザ ID・パスワードも可能	○ (表 5-2 参照)	エージェント登録要求時	DBPARTNER/ Client32 DBPARTNER2 Client
エージェント一覧（登録情報）取得権限	該当するエージェントの実行権限のあるユーザ ID・パスワードだけ	×	—	—
エージェント実行結果取得権限	エージェント実行結果を公開しない場合： エージェント実行ユーザ ID・パスワードだけ	○ (表 5-2 参照)	エージェント登録要求時	DBPARTNER/ Client32 DBPARTNER Mobile

権限の種類	権限の所有者	権限の設定可否	設定タイミング	権限を設定できるクライアント
	エージェント実行結果を公開する場合： エージェント実行ユーザ ID・パスワード以外のユーザ ID・パスワードも可能		エージェント実行要求時	DBPARTNER2 Client
エージェント実行結果削除権限	エージェント実行ユーザ ID・パスワードだけ	×	—	—
エージェント実行状況取得権限	該当するエージェント一覧の取得権限のあるユーザ ID・パスワードだけ	×	—	—

(凡例)

○：できます。

×

—：該当しません。

表 5-2 権限を設定できるエージェントの権限の内容と権限の認証方法

設定できる権限		権限内容	権限の認証方法
エージェント実行権限	エージェントを公開しない場合	エージェント登録ユーザ ID・パスワードだけがエージェントを実行できます	エージェント実行時のユーザ ID・パスワードとエージェント登録時のユーザ ID・パスワードを照合します
	エージェントを公開する場合 (ユーザのグループ管理あり)	エージェントに付与されたグループ内のメンバ (ユーザ ID) だけがエージェントを実行できます	エージェント実行時のユーザ ID がエージェントに付与されたグループのメンバかどうかを照合します
	エージェントを公開する場合 (ユーザのグループ管理なし)	すべてのユーザ ID がエージェントを実行できます	照合しません
エージェント実行結果取得権限	エージェント実行結果を公開しない場合	エージェント実行ユーザ ID・パスワードだけがエージェント実行結果を取得できます	エージェント実行結果取得時のユーザ ID・パスワードとエージェント実行時のユーザ ID・パスワードを照合します
	エージェント実行結果を公開する場合 (ユーザのグループ管理あり)	エージェントに付与されたグループ内のメンバ (ユーザ ID) だけがエージェント実行結果を取得できます	エージェント実行結果取得時のユーザ ID がエージェントに付与されたグループのメンバかどうかを照合します
	エージェント実行結果を公開する場合	すべてのユーザ ID がエージェント実行結果を取得できます	照合しません

設定できる権限		権限内容	権限の認証方法
	(ユーザのグループ管理なし)		

5.4 エージェント機能で使用するファイルの運用

ここでは、エージェント機能で使用するファイルの運用について説明します。

5.4.1 ファイルのバックアップ

エージェント機能で使用する¥DABAGT 下のファイルは、定期的なバックアップの取得をお勧めします。ハードディスク、システムの破壊などでファイルが破壊された場合、エージェント登録情報、実行結果などは消滅します。バックアップを取得しておく、バックアップファイルで繰り返し実行するエージェントの実行予約を有効にするなどの回復ができます。

(1) 自動的にバックアップファイルが作成されるファイル

次のファイルには、自動的にバックアップファイルが作成されます。バックアップファイルは、DABAGT に指定したディレクトリ¥xxxxx.bak です。

- エージェント登録情報管理ファイル
(DABAGT に指定したディレクトリ¥DABAGT¥RCTRL.AGT)
このファイルが破壊された場合、エージェントの登録、削除、実行、登録情報の取得などができなくなります。
- エージェント実行情報管理ファイル
(DABAGT に指定したディレクトリ¥DABAGT¥ECTRL.AGT)
このファイルが破壊された場合、エージェントの実行、実行状況の取得、実行結果の取得、実行結果の削除などができなくなります。
- 実行予約をしたエージェントの実行待ちキューファイル
(DABAGT に指定したディレクトリ¥DABAGT¥timer_que)
このファイルが破壊された場合、実行予約をしたエージェントを実行できなくなります。

(2) バックアップファイルを使った回復方法

ファイルが破壊された場合、それぞれのバックアップファイルを使ってファイルを回復できます。

5.4.2 エージェント登録情報及び実行情報ファイルについて

エージェント登録情報及び実行情報ファイルは、エージェント登録又は実行ごとに生成されるファイルです。登録数又は実行数が多くなるにつれて、ファイル数も増えていき、ファイル数及びファイル容量がサーバファイルシステムを圧迫する恐れがあります。したがって、ルートディレクトリのパーティションとは別に、エージェント用ディレクトリのパーティションを確保するなどの運用をシステム管理者が行ってください。

それでもサーバファイルシステムの圧迫を解消できない場合は、使用しないファイルの退避又は削除などの対処をしてください。ファイルの削除については、「5.4.3 ファイルの削除」を参照してください。

なお、ファイル容量の目安となる概算式を「5.4.5(2) ファイル容量の見積もり」に示していますので、ファイル運用の参考にしてください。

5.4.3 ファイルの削除

エージェント登録情報、エージェント実行情報、及びエージェント実行結果格納ファイルを削除する時の注意事項について説明します。

- 各 OS の機能を使用して削除してください。
- 削除前に必ず、ファイル名、及びエージェント実行情報（実行状態、実行日、実行ユーザ ID など）を、次の方法で確認してください。

UNIX の場合

dabagtexeinfo コマンドで情報を確認できます。dabagtexeinfo コマンドについては、「3.8.4 エージェント実行情報の表示」を参照してください。

Windows の場合

エージェント情報表示ユーティリティを起動して情報を確認できます。エージェント情報表示ユーティリティについては、「4.9.3(2) エージェント実行一覧ユーティリティの実行」を参照してください。

(1) エージェント登録情報ファイルの削除

クライアントからエージェント登録情報を削除します。クライアントからエージェント登録情報（ファイル）を削除できない場合は、システム管理者がサーバ上でエージェント登録情報ファイル（REGaa…aa）を削除してください。

(2) エージェント実行情報ファイルの削除

- クライアントからエージェント実行情報を削除します。クライアントからエージェント実行情報（ファイル）が削除できない場合は、システム管理者がサーバ上でファイルを削除してください。
- 必ず実行終了状態のファイルを削除してください。誤って実行中のファイルを削除した場合は、DABroker の動作を保証しませんので御注意ください。
- エージェント実行情報ファイルを削除するときは、対応するエージェント実行結果格納ファイルも削除してください。対応するエージェント実行結果格納ファイルとは、エージェント実行情報ファイル（EXEbb…bb.AGT）のファイル名称に付いているエージェント実行名（bb…bb）と同じエージェント実行名が付いたファイル（RESbb…bb.AGT）です。
- エージェント実行情報ファイルを削除することで、実行予約を設定したエージェントタイマー監視を、サーバで強制的にキャンセルできます。

エージェントの実行予約時間が近づいている場合は、ファイルの削除中にエージェントが実行状態に移行することもあるため、エージェント実行情報ファイルを削除しないでください。

エージェントタイマー監視のキャンセルについては、「[6.2.2 エージェントタイマー監視](#)」を参照してください。

(3) エージェント実行結果格納ファイルの削除

- クライアントからエージェント実行結果を削除します。クライアントからエージェント実行結果（ファイル）を削除できない場合は、システム管理者がサーバ上でファイルを削除してください。
- エージェント実行結果格納ファイルを削除するときは、対応するエージェント実行情報ファイルも削除してください。

5.4.4 エージェント実行名の割り当て規則

DABroker でのエージェント実行名の割り当て方は、実行予約をしていないエージェント、実行予約を指定したエージェント、及び繰り返し実行するエージェントによって異なります。次に、エージェント実行名の割り当て形式及びそれぞれの場合の実行名（実行順序番号部分だけ）を示します。

DABroker でのエージェント実行名の割り当て形式

「XXXXXXXXXXXXnnnn」

XXXXXXXXXXXX

実行名基本部（16進数字 0~9, A~F）

nnnn

実行順序番号（数字 0001~9999）

- 実行予約をしていないエージェント
nnnn=0000
- 実行予約を指定したエージェント
nnnn=0001
- 繰り返し実行するエージェント
nnnn=0001~9999

繰り返すごとに「nnnn」の値に1が加算されます。

したがって、繰り返し実行するエージェントの実行結果（実行状況）を取得（削除）する場合は、クライアントで実行順序番号（繰り返し回数）を考慮したエージェント実行名を指定してください。

(例)

クライアントから繰り返し実行するエージェントの、3回目に実行したエージェントの実行結果を取得する場合

エージェント実行名：AABBCCDDEEFF0003

なお、繰り返し実行で「nnnn=FFFF（実行順序番号なし）」を指定した場合は、最新のエージェント実行結果（実行状況）を返します。

5.4.5 エージェント機能のパラメーター一覧及びファイル容量の見積もり（共通）

エージェント機能のすべての処理には、パラメータが関連付けられています。また、エージェント登録情報ファイル及びエージェント実行情報ファイルはパラメータで構成されています。したがってファイルの容量は、パラメータに設定する値を足した値になります。ここでは、エージェントのパラメーター一覧、及びファイル容量の見積もりについて説明します。

(1) エージェント機能のパラメーター一覧

エージェント機能に関連付けられているパラメーター一覧を、表 5-3 及び表 5-4 に示します。エージェント関連のコマンド、エージェントログ、及びエージェントのメッセージで表示されるパラメータを知る場合などに参照してください。パラメータは、エージェント登録情報（登録時の情報）とエージェント実行情報（実行時の情報）に分けて示しています。

表 5-3 エージェント登録情報のパラメータ

パラメータ名	説 明		データの属性
AgtRegName	エージェント名		最大 64 バイトの任意文字※
AgtRegUser	登録ユーザ情報	ユーザ ID	最大 30 バイトの任意文字※ (ただし、両端に「"」ダブルクォーテーションを使用する場合は、最大 32 バイトの任意文字)
AgtRegPass		パスワード	
AgtRegGroup	登録グループ情報	ユーザ ID	最大 30 バイトの英数字
AgtRegGPass		パスワード	
AgtRegDate	登録日時	日付	—
AgtRegTime		時刻	
AgtExeAuth	エージェント実行権限		—
AgtRefAuth	エージェント実行結果取得権限		
AgtComment	コメント		最大 255 バイトの任意文字
AgtConDBType	接続先情報	接続データベース種別	—
AgtConDBName		データベース識別名	
AgtConDBHost	実行予約日時	データベースホスト名	—
AgtRsvDate		日付	
AgtRsvTime		時刻	

パラメタ名	説 明		データの属性
AgtRsvDay		曜日	
AgtRsvRet		繰り返し	
AgtResWrite	実行結果保存形態		
AgtRegInParm	可変実行条件		
AgtSQLText	SQL テキスト		
AgtHeaderText	ヘッダテキスト		
AgtRetNum	リトライ情報	リトライ回数(回)	
AgtRetInt		リトライ間隔(秒)	

注※

次の文字はシステムの予約語のため使用できません。

[=], [:], [;], [/], ["], [*], [+], [-], [<], [>], [,]

表 5-4 エージェント実行情報のパラメタ

パラメタ名	説 明		データの属性	
AgtPriority	優先度		-	
AgtRegLimit	エージェント存続期限			
AgtResLimit	エージェント実行結果の保存期限			
AgtRegInfo	登録時のインフォメーション			
AgtExeName	エージェント実行名			
AgtExeNote	メモ		最大 64 バイトの任意文字※	
AgtExeUser	実行ユーザ情報	ユーザ ID	最大 30 バイトの任意文字※	
AgtExePass		パスワード		
AgtStatus	実行状況	状態	-	
AgtExePID		実行プロセス ID		
AgtErrMSG		メッセージ		
AgtExeSDate		実行日時		日付
AgtExeSTime				時間
AgtRsvCDate		カレントタイム		日付
AgtRsvCTime		実行待ち日時		時刻
AgtOutType		実行結果情報		出力先種別

パラメタ名	説明	データの属性
AgtOutName		出力名
AgtOutSize		出力サイズ
AgtOutColCnt		カラム情報
AgtOutColSize		カラムサイズ
AgtOutRecCnt		レコード数
AgtOutRecSize		レコードサイズ
AgtExeInfo	実行時のインフォメーション	
AgtExeInParm	可変実行条件	
登録情報に従うパラメタ	その他のエージェント情報	

注※

次の文字はシステムの予約語のため使用できません。

「=」, 「:」, 「;」, 「/」, 「"」, 「*」, 「+」, 「-」, 「<」, 「>」, 「,」

(2) ファイル容量の見積もり

エージェント登録情報ファイル及びエージェント実行情報ファイルは、次に示す概算式で容量を見積もります。概算式中のそれぞれの値は、表 5-5 及び表 5-6 に示すパラメタの見積もり値を使用して求めます。

- エージェント登録情報ファイル (DABAGT が設定されているディレクトリ¥REGaa…aa)**
 エージェント登録情報ファイル容量 =
 エージェント登録情報長値^{*1}+実行条件の値^{*2}+ヘッダテキスト値^{*3}
- エージェント実行情報ファイル (DABAGT が設定されているディレクトリ¥EXEbb…bb)**
 エージェント実行情報ファイル容量 =
 エージェント実行情報長値^{*4}+実行条件の値^{*2}+ヘッダテキスト値^{*3}

概算式中の値は次のように求めます。求めるときには、表 5-5、及び表 5-6 で示す値を使用してください。DABAGT は、DABroker 動作環境定義ファイルのエージェント機能の使用で指定する作業ディレクトリです。DABroker 動作環境定義ファイルについては、次の箇所を参照してください。

UNIX の場合：「[3.4 DABroker の動作環境の設定](#)」

Windows の場合：「[4.4 DABroker 環境設定](#)」

注※1 エージェント登録情報長値 = $\sum_{\text{使用パラメタ項目分}} (\text{固定部} + \text{可変部})$

注※2 実行条件の値 = $\sum_{i=1}^n (40 + C_i + L_i + V_i)$

注※3 ヘッダテキスト値 = $\sum_{i=1}^m (19 + M_i + L_i + V_i)$

- n : 実行条件項目数
- m : ヘッダ項目数
- C : 実行条件の項目名長
- L : 実行条件項目のデータ長
- M : 設定順序長
- V : パラメタ長

注※4 エージェント実行情報長値 = $\sum_{\text{使用パラメタ項目分}} (\text{固定部} + \text{可変部})$

表 5-5 エージェント登録情報ファイルのパラメタ及び見積もり値

パラメタ名	固定部 (バイト)	可変部	可変部の範囲 (バイト)			見積もり長 (バイト)		
			最大	最小	通常	最大	最小	通常
AgtRegName	14	RN	64	1	20	78	15	34
AgtRegUser	14	RU	32	1	8	46	15	22
AgtRegPass	14	RP	32	1	8	46	0	22
AgtRegGroup	15	RG	32	1	8	47	0	23
AgtRegGPass	15	GP	32	1	8	47	0	23
AgtRegDate	14*	10*	10*	10*	10*	24*	24*	24*
AgtRegTime	14*	8*	8*	8*	8*	22*	22*	22*
AgtExeAuth	14	2	2	2	2	16	0	16
AgtRefAuth	14	2	2	2	2	16	0	16
AgtComment	14	CM	255	1	8	269	0	22
AgtConDBType	16	2	2	2	2	18	18	18
AgtConDBName	16	DN	255	1	8	271	17	24
AgtConDBHost	16	DH	255	1	8	271	17	24
AgtRsvDate	14	10	10	10	10	24	0	24
AgtRsvTime	14	5	5	5	5	19	0	19
AgtRsvDay	13	2	2	2	2	15	0	15
AgtRsvRet	13	2	2	2	2	15	0	15
AgtResWrite	15	2	2	2	2	17	0	17
AgtSQLText	14	ST	32767	1	50	32781	15	64
AgtRegInfo	14	RI	255	1	8	269	0	22

パラメタ名	固定部 (バイト)	可変部	可変部の範囲 (バイト)			見積もり長 (バイト)		
			最大	最小	通常	最大	最小	通常
合計	287		34024	55	179	34311	143	466

(凡例)

CM：コメント長

DH：データベースホスト名

DN：接続先データベース名称長

GP：グループパスワード

RI：登録インフォメーション長

RG：グループID

RN：エージェント登録名称長

RP：エージェント登録時のユーザパスワード長

RU：エージェント登録時のユーザID長

ST：SQL テキスト長

※：DABroker が設定する項目

注

通常の値は、各可変名称及びSQLテキストは次の値で計算しています。

可変名称：8文字（エージェント名は2バイトコード10文字）

SQLテキスト：50バイト

表 5-6 エージェント実行情報ファイルのパラメタ及び見積もり値

パラメタ名	固定部 (バイト)	可変部	可変部の範囲 (バイト)			見積もり長 (バイト)		
			最大	最小	通常	最大	最小	通常
AgtExeName	14	16	16	16	16	30	30	30
AgtExeNote	14	ET	64	1	20	78	0	34
AgtExeUser	14	EU	32	1	8	46	15	22
AgtExePass	14	EP	32	1	8	46	0	22
AgtStatus	13※	12※	14※	3※	12※	27※	16※	25※
AgtExePID	13※	4※	6※	1※	4※	19※	14※	17※
AgtErrMsg	13※	0※	512※	1※	0※	525※	0※	13※
AgtExeSDate	15※	10※	10※	10※	10※	25※	25※	25※
AgtExeStime	15※	8※	8※	8※	8※	23※	23※	23※
AgtRsvCDate	15※	CD※	10※	10※	10※	25※	0※	25※

パラメタ名	固定部 (バイト)	可変部	可変部の範囲 (バイト)			見積もり長 (バイト)		
			最大	最小	通常	最大	最小	通常
AgtRsvCTime	15*	CT*	5*	5*	5*	20*	0*	20*
AgtOutType	14	3	3	3	3	17	17	17
AgtOutName	14*	30*	255*	1*	30*	269*	15*	44*
AgtOutSize	14*	OS*	10*	1*	4*	24*	15*	18*
AgtOutColCnt	16*	CC*	10*	1*	4*	26*	17*	20*
AgtOutColSize	17*	CS*	10*	1*	4*	27*	18*	21*
AgtOutRecCnt	16*	RC*	10*	1*	4*	26*	17*	20*
AgtOutRecSize	17*	RS*	10*	1*	4*	27*	18*	21*
AgtExeInfo	14	EI	255	1	8	269	0	22
合計	277		1272	67	162	1549	240	439

(凡例)

CC：カラム情報数

CD：タイム実行待ちの場合設定 (10 バイト固定)

CS：カラム情報サイズ

CT：エージェントタイム実行待ちの場合設定 (5 バイト固定)

EI：実行インフォメーション

EP：エージェント実行時のユーザパスワード長

ET：メモ長

EU：エージェント実行時のユーザ ID 長

OS：実行結果のサイズ (10 進文字長)

RC：レコード情報数

RS：レコード情報サイズ

※：状況に応じて DABroker が設定する項目

注

通常の値では、各可変名称及びシステム可変値は次の値で計算しています。

可変名称：8 文字 (メモは 2 バイトコード 10 文字)

システム可変値：平均長

6

DABroker の障害対策（共通）

この章では、DABroker での障害が発生したときの対処方法、及びタイマー監視について説明します。

UNIX を使用する環境と Windows を使用する環境で共通の内容です。

6.1 障害対策

DABroker の運用中にエラーが発生した場合、エラーメッセージが出力されます。「7 メッセージ (共通)」を参考にしてエラー原因を取り除いてください。

ここでは、エラー時に DABroker が実行する処理内容、及び保守員への連絡時に取得する障害情報について説明します。なお、このマニュアルでの保守員とは、システム管理者がご購入契約に基づくお問い合わせ窓口のことです。

6.1.1 エラー時に DABroker が実行する処理

システムで次のようなエラーが発生した場合、DABroker は、データベースを切り離します。

- DABroker が異常終了した場合
データベースでの実行中の処理はロールバックされます。
- DBMS が異常終了した場合
データベースでの処理は DBMS に任せます。

通常、クライアントからの要求でデータベースを切り離しますが、これらのエラーの場合及び DABroker を強制終了する場合は、DABroker が判断してデータベースを切り離します。

6.1.2 障害情報の取得

次のような場合は、システム管理者が保守員に連絡してください。

- メッセージの対処に、「保守員に連絡してください。」と記載されている場合
- 障害の要因が分からない場合
- メッセージの内容で対処できない場合

保守員に連絡する場合は、次に示す障害情報をコピーして保存してください。

UNIX の場合は、DABroker 運用ディレクトリにある `.HTC_2340.inf` についても取得してください。

(1) DABroker のトラブルシューティング情報

(a) スプール情報

DABroker 運用ディレクトリの `spool` ディレクトリ下のすべてのファイルをコピーして保存してください。

- メッセージログ

UNIX :

DABroker 運用ディレクトリ/spool/dabsv.log, 及びdabsv.logold

Windows :

DABroker 運用ディレクトリ¥spool¥dabsv.log, 及びdabsv.logold

- データベースアクセストレース, 拡張データベースアクセストレース

UNIX :

DABroker 運用ディレクトリ/spool/db_access 下のファイル

Windows :

DABroker 運用ディレクトリ¥spool¥db_access 下のファイル

- 通信トレース

UNIX :

DABroker 運用ディレクトリ/spool/dabcltrc

Windows :

DABroker 運用ディレクトリ¥spool¥dabcltrc

- イベントトレース

UNIX :

DABroker 運用ディレクトリ/spool/dabevttrcXXX1, 及びdabevttrcXXX2

Windows :

DABroker 運用ディレクトリ¥spool¥dabevttrcXXX1, 及びdabevttrcXXX2

イベントトレースの取得形式については、「[6.3.1 イベントトレースの取得形式 \(共通\)](#)」を参照してください。

- XA トレース

UNIX :

DABroker 運用ディレクトリ/spool/db_xainfo.log

Windows :

DABroker 運用ディレクトリ¥spool¥db_xainfo.log

(b) DABroker の定義ファイル

次のファイルをコピーして保存してください。

- DABroker 動作環境定義ファイル

UNIX :

DABroker 運用ディレクトリ/conf/dasysconf

Windows :

DABroker 運用ディレクトリ¥conf¥dasysconf

- 接続先データベース定義ファイル

障害対策用として保存するファイルには、パスワードなどの公開できない情報は削除しておくことをお勧めします。

UNIX :

DABroker 運用ディレクトリ/conf/dadbenv

Windows :

DABroker 運用ディレクトリ¥conf¥dadbenv

- 運用情報ファイル (UNIX)

/opt/DABroker/.dabrokerenv

- コネクションマルチ定義ファイル

UNIX :

DABroker 運用ディレクトリ/conf/damconenv

Windows :

DABroker 運用ディレクトリ¥conf¥damconenv

(c) トレースを取得している場合

トレースの取得を設定している場合は、データベースアクセストレース、通信トレースをファイルに出力して保存してください。

- データベースのアクセストレースの出力方法

UNIX の場合

dabdbtrc コマンドを使用します。dabdbtrc コマンドについては「[6.4.2 データベースアクセストレースの表示](#)」を参照してください。

Windows の場合

DABroker トレース出力ユーティリティを使用します。ユーティリティについては、「[4.9.2\(3\) データベースアクセストレースの出力方法](#)」を参照してください。

- 通信トレースの出力方法

UNIX の場合

dabcltrc コマンドを使用します。dabcltrc コマンドについては「[3.8.2 クライアントとの通信トレースの表示](#)」を参照してください。

Windows の場合

DABroker トレース出力ユーティリティを使用します。ユーティリティについては、「[4.9.2\(2\) クライアントとの通信トレースの出力方法](#)」を参照してください。

(2) システムのトラブルシューティング情報

(a) UNIX の場合

- システムログファイル

システムログファイルをコピーして保存してください。

HP-UX :

`/var/adm/syslog/syslog.log`

Solaris :

`/var/adm/messages`

Linux :

`/var/log`

AIX :

`/var/adm/ras`

- システムダンプ

DABroker の異常終了時の情報が `core` ファイルに格納されています。この情報をコピーして保存してください。標準では、開発したアプリケーションを起動したディレクトリに格納されますが、格納先はアプリケーションに依存します。

(b) Windows の場合

- イベントログ

イベントログをファイルに取得してください。取得方法は次のとおりです。

- Windows の [管理ツール] から [イベントビューア] を開く。
- [ログ] メニューの [アプリケーション] を選択し、表示をアプリケーションに変更する。
- [ログ] メニューの [名前をつけて保存] を選択し、ログをファイルに保存する。

(3) エージェント機能のトラブルシューティング情報

(a) エージェントの表示情報

UNIX の場合

`dabagtreinfo` コマンドと `dabagtexeinfo` コマンドを実行して、エージェントの登録情報と実行情報をそれぞれ取得してください。

`dabagtreinfo` コマンドと `dabagtexeinfo` コマンドについては、「[3.8.3 エージェント登録情報の表示](#)」を参照してください。

Windows の場合

エージェント情報表示ユーティリティを実行して「エージェント実行一覧」, 「エージェント登録一覧」のハードコピーを取得してください。

エージェント情報表示ユーティリティについては、「4.9.3(1) エージェント登録一覧ユーティリティの実行」を参照してください。

(b) エージェントのログ

エージェントのログファイルをコピーして保存してください。

- DABroker 動作環境定義ファイル DABAGT の使用の作業ディレクトリに指定したディレクトリ `¥agtlogfile`, 及び `agtlogfile.old`

DABAGT は、DABroker 動作環境定義ファイルのエージェント機能の使用で指定する作業ディレクトリです。DABroker 動作環境定義ファイルについては、次の箇所を参照してください。

UNIX の場合：「3.4 DABroker の動作環境の設定」

Windows の場合：「4.4 DABroker 環境設定」

エージェントログの取得形式と取得内容については、「6.3.2 エージェントログの取得形式と取得内容（共通）」を参照してください。

(c) エージェント機能のファイル

次に示すエージェント機能のファイルをコピーして保存してください。

- DABAGT に指定したディレクトリ下のすべてのファイル
 - REGaa...aa.*
 - RCTRL.*
 - EXEbb...bb.*
 - ECTRL.*
 - RESbb...bb.*
 - timer_que.*

DABAGT は、DABroker 動作環境定義ファイルのエージェント機能の使用で指定する作業ディレクトリです。DABroker 動作環境定義ファイルについては、次の箇所を参照してください。

UNIX の場合：「3.4 DABroker の動作環境の設定」

Windows の場合：「4.4 DABroker 環境設定」

(4) XA インタフェースでのトラブルシューティング情報

XA インタフェースでのトラブルシューティング情報は、XA トレースファイルとして取得します。

XA トレースファイルは次のファイルに取得されます。

UNIX：DABroker 運用ディレクトリ/spool/db_xainfo.log

Windows : DABroker 運用ディレクトリ¥spool¥db_xainfo.log

XA トレースの取得方法については、次の箇所を参照してください。

UNIX の場合 : 「[3.4 DABroker の動作環境の設定](#)」

Windows の場合 : 「[4.4 DABroker 環境設定](#)」

XA トレースの取得形式については、「[6.3.3 XA トレースの取得形式 \(共通\)](#)」を参照してください。

6.2 DABroker でのタイマー監視

DABroker では、タイマー監視を実行しています。

6.2.1 DBPARTNER と DABroker の接続時のタイマー監視

DABroker は、DBPARTNER とのコネクションが確立されている間、DBPARTNER からの要求待ちタイムアウトを監視します。このタイムアウトが発生するのは、DABroker が DBPARTNER からの要求に対して応答を返してから、一定時間 DBPARTNER からの要求がない場合です。

DBPARTNER からの要求待ちタイムアウトが発生した場合、DABroker は DBPARTNER とのコネクションを解放します。データベースにはロールバックを要求して、データベースを切り離します。

6.2.2 エージェントタイマー監視

実行予約を設定した（エージェントタイマー指定）エージェントの実行要求を受けたときに、DABroker はエージェントタイマー監視を開始します。

(1) エージェントタイマー監視のキャンセル

次に示す操作でエージェントタイマー監視をキャンセルできます。

- DBPARTNER Mobile - Client からキャンセルを要求

DBPARTNER Mobile - Client で、エージェント実行状況一覧の中のエージェントを削除したすると、DABroker では、実行中又はタイマー監視中のエージェントの実行をキャンセルし、エージェントタイマー監視をキャンセルします。

- エージェント実行情報ファイルを削除する

エージェント実行情報ファイルを削除すると、サーバで強制的に、エージェントタイマー監視をキャンセルします。キャンセルしたエージェントの実行予約時間になったときは、エージェントタイマー監視の実行エラーとして以降の監視を中止します。エージェント実行情報ファイルの削除については、「[5.4.3 ファイルの削除](#)」を参照してください。

- DABroker を計画終了又は強制終了する

DABroker を計画終了及び強制終了すると、エージェントタイマー監視はキャンセルされます。

6.2.3 Database Connection Server 接続時のタイマー監視

DABroker は、Database Connection Server との TCP/IP コネクションが確立されている間、Database Connection Server からの処理結果応答待ちタイムアウトを監視します。指定した時間内に Database

Connection Server から応答がない場合、DABroker は Database Connection Server との TCP/IP コネクションを解放します。

6.3 トレースの取得形式 (共通)

6.3.1 イベントトレースの取得形式 (共通)

DABroker は、DABroker 関連製品を使って開発したアプリケーションからデータベースにアクセスする場合のイベントトレースを取得します。イベントトレースは、次のファイルに取得されます。

UNIX : DABroker 運用ディレクトリ/spool/dabevttrcXXX1

Windows : DABroker 運用ディレクトリ¥spool¥dabevttrcXXX1

XXX :

DABroker for C++の場合 : cpp

DABroker for ORB の場合 : orb

DABroker for COBOL の場合 : cobol

なお、イベントトレースファイルが満杯になったら、ラップアラウンドでトレースを取得します。ラップアラウンドで取得した場合は、バックアップファイル (dabevttrcXXX2) を作成し、一世代だけ (dabevttrcXXX1) を管理します。

取得するイベントトレースの設定については、「[3.5.3 イベントトレースの設定](#)」(UNIX の場合)、「[4.5.4 for C++, for COBOL, for ORB 設定](#)」(Windows の場合) 及び DABroker 関連製品の各マニュアルを参照してください。

次に、イベントトレースの取得例を示します。

取得例

LEVEL	VPID	PID	TID	DATE	TIME	EVENT
40000010	00120000	711a	0001	1998/06/01	11:38:29.300000	イベント 1
40000010	00120000	711a	0002	1998/06/01	11:38:30.100000	イベント 2
40000010	00120000	711b	0001	1998/06/01	11:38:30.150000	イベント 3
40000010	00120000	711b	0002	1998/06/01	11:38:30.250000	イベント 4

LEVEL :

トレースに出力したレベル

VPID :

仮想プロセス ID

PID :

プロセス ID

TID :

スレッド ID

DATE :

取得した日付

TIME :

取得した時間

EVENT :

実行時の動作結果のログ

6.3.2 エージェントログの取得形式と取得内容（共通）

エージェントログを取得する設定にした場合、エージェント機能を使用したその内容がエージェントログファイル（DABAGT に指定したディレクトリ¥agtlogfile）に取得されます。

DABAGT は、DABroker 動作環境定義ファイルのエージェント機能の使用で指定する作業ディレクトリです。DABroker 動作環境定義ファイルについては、次の箇所を参照してください。

UNIX の場合：「[3.4 DABroker の動作環境の設定](#)」

Windows の場合：「[4.4 DABroker 環境設定](#)」

エージェントログ情報としてインフォメーション情報とエラー情報があり、それぞれ、情報を取得するかどうかを、DABroker 動作環境定義ファイルに設定します。

エージェントログファイルが満杯になった場合は、ラップアラウンドしてバックアップファイル（agtlogfile.old）を作成します。バックアップファイルは一世代だけ管理をします。また、DABroker を終了後、再起動した場合もバックアップファイルを作成します。

ここでは、取得する形式、及び内容を示します。エージェントログファイルはテキスト形式のファイルなので参照できます。

(1) エージェントログの取得形式

エージェントのログは次のような形式で取得されます。

日時	プロセスID	KFDBnnnAL-m	AgentXXX	イベント名	パラメタ=値	...
----	--------	-------------	----------	-------	--------	-----

日時

エージェントログを取得した日時を示します。（年/月/日 時:分:秒）

プロセスID

エージェントログを取得したプロセスのプロセスIDを示します。

エージェントログID

エージェントログの種別を示します。

KFDB

DABrokerのエージェントログ又はメッセージであることを示します。

nnnAL-m AgentXXX

エージェントログの種別を示します。

001AL-I AgentLog：インフォメーション情報

DABrokerで実行するエージェントのログです。

- UNIXの場合

DABroker動作環境定義ファイルのエージェント機能の使用で、「エージェントログにインフォメーションメッセージを取得する」をONにしている場合だけインフォメーション情報を取得します。

- Windowsの場合

DABroker動作環境設定ユーティリティの「エージェント機能の使用」－「ログにインフォメーションメッセージを取得する」チェックボックスをチェックあり（ON）にした場合だけインフォメーション情報を取得します。

002AL-E AgentErr：エラー情報

DABrokerで実行するエージェントが正常に動作していないことを示すエラーログです。

- UNIXの場合

DABroker動作環境定義ファイルのエージェント機能の使用で、「エージェントログにエラーメッセージを取得する」をONにしている場合だけエラー情報を取得します。

- Windowsの場合

DABroker動作環境定義ユーティリティの「エージェント機能の使用」－「ログにエラーメッセージを取得する」チェックボックスをチェックあり（ON）にした場合だけ、エラー情報を取得します。

イベント名

表6-1の「イベント名」を示します。エージェント機能を使った処理で発生するイベント名を示します。

パラメタ

表6-1の「取得内容（パラメタ=値）」を示します。

「パラメタ=値」で、登録、実行情報の項目とその値を、カンマ（,）で区切って表示します。パラメタについては、「[5.4.5 エージェント機能のパラメタ一覧及びファイル容量の見積もり（共通）](#)」を参照してください。

次に、エージェント名「在庫検索」のエージェント登録、登録完了、登録結果の応答、及び実行要求の受付までのエージェントログの取得例を示します。

```
1998/12/01 12:00:00 123 KFDB001AL-I AgentLog RegReq AgtRegName=在庫検索,
AgtRegUser=K654321
1998/12/01 12:03:48 123 KFDB001AL-I AgentLog RegCmp AgtRegName=在庫検索,
AgtRegUser=K654321, RC=0
1998/12/01 12:05:06 123 KFDB001AL-I AgentLog RegRsp AgtRegName=在庫検索,
AgtRegUser=K654321, RC=0
1998/12/01 12:30:12 12345 KFDB001AL-I AgentLog ExeRsp AgtRegName=在庫検索,
AgtExeUser=K654321
:
:
```

(2) エージェントログの取得内容

イベントごとに、取得するエージェントログの内容を表 6-1 に示します。

表 6-1 エージェントログの取得内容

イベント名	イベント内容	取得内容 (パラメータ=値)	取得タイミング
RegReq	エージェント登録要求の受付	AgtRegName=登録名 AgtRegUser=ユーザ ID	DABroker で、エージェントの登録要求を受け取った直後
RegRsp	エージェント登録結果の応答	AgtRegName=登録名 AgtRegUser=ユーザ ID RC=登録処理結果	エージェントの登録処理結果を応答する直前
RegDelReq	エージェント削除要求の受付	AgtRegName=登録名 AgtRegUser=ユーザ ID	DABroker で、エージェントの削除要求を受け取った直後
RegDelRsp	エージェント削除結果の応答	AgtRegName=登録名 AgtRegUser=ユーザ ID RC=削除処理結果	エージェントの削除処理結果を応答する直前
ExeReq	エージェント実行要求の受付	AgtRegName=登録名 AgtExeUser=ユーザ ID	DABroker で、エージェントの実行要求を受け取った直後
ExeRsp	エージェント実行結果の応答	AgtRegName=登録名 AgtExeName=実行名 AgtExeUser=ユーザ ID RC=実行受付処理結果	エージェントの実行要求の受付処理結果を応答する直前
ActAgt	エージェント実行の開始	AgtRegName=登録名 AgtExeName=実行名 AgtExePID=PID AgtExeUser=ユーザ ID	データベースアクセスエージェントの起動時
SQLRes	エージェント機能を使った SQL 実行結果	AgtRegName=登録名 AgtExeName=実行名 AgtExePID=PID AgtExeUser=ユーザ ID RC=実行処理結果	データベースアクセスエージェントでの、SQL 実行を終了すること
EndAgt	エージェント実行の完了	AgtRegName=登録名 AgtExeName=実行名 AgtExePID=PID AgtExeUser=ユーザ ID RC=実行処理結果	データベースアクセスエージェントの実行完了時
RegLstReq (Pre/Cnt/ End)	エージェント登録情報の取得要求の受付	AgtRegName=登録名* AgtRegUser=ユーザ ID	エージェント登録情報の取得要求を受け取った直後

イベント名	イベント内容	取得内容 (パラメタ=値)	取得タイミング
RegLstRsp (Opn/Pre/Cnt/End)	エージェント登録情報の取得要求の応答	AgtRegName=登録名 AgtRegUser=ユーザ ID RC=処理結果	エージェント登録情報の取得要求の処理結果を応答する直前
ExeLstReq (Opn/Pre/Cnt/End)	エージェント実行状況の取得要求の受付	AgtExeName=実行名※ AgtExeNote=メモ※ AgtExeUser=ユーザ ID	エージェント実行状況の取得要求を受け取った直後
ExeLstRsp (Opn/Pre/Cnt/End)	エージェント実行状況の取得要求の応答	AgtExeName=実行名 AgtExeNote=メモ AgtExeUser=ユーザ ID RC=処理結果	エージェント実行状況の取得要求処理結果を応答する直前
ResLstReq (Opn/Pre/Cnt/End)	エージェント実行結果の取得要求の受付	AgtExeName=実行名※ AgtExeNote=メモ※ AgtExeUser=ユーザ ID	エージェント実行結果の取得要求を受け取った直後
ResLstRsp (Opn/Pre/Cnt/End)	エージェント実行結果の取得要求の応答	AgtExeName=実行名 AgtExeNote=メモ AgtExeUser=ユーザ ID RC=処理結果	エージェント実行結果の取得要求処理結果を応答する直前
ExeDelReq	エージェント実行状況及び実行結果の削除要求の受付	AgtExeName=実行名 AgtExeNote=メモ AgtExeUser=ユーザ ID	エージェント実行状況及び実行結果の削除要求を受け取った直後
ExeDelRsp	エージェントの実行状況及び実行結果の削除要求の応答	AgtExeName=実行名 AgtExeNote=メモ AgtExeUser=ユーザ ID RC=処理結果	エージェント実行状況及び実行結果の削除結果を応答する直前
ExeCanReq	エージェント実行のキャンセル要求の受付	AgtExeName=実行名 AgtExeNote=メモ AgtExeUser=ユーザ ID	エージェント実行のキャンセル要求を受け取った直後
ExeCanRsp	エージェント実行のキャンセル要求の応答	AgtExeName=実行名 AgtExeNote=メモ AgtExeUser=ユーザ ID RC=処理結果	エージェント実行のキャンセル結果を応答する直前
ExeDirReq (Pre/Cnt/End)	エージェントの即時配信実行要求の受付 (Windows 限定)	AgtExeName=実行名※ AgtExeUser=ユーザ ID	DABroker でエージェントの即時配信実行要求を受け取った直後
ExeDirRsp (Pre/Cnt/End)	エージェントの即時配信実行結果の応答 (Windows 限定)	AgtExeName=実行名※ AgtExeUser=ユーザ ID RC=処理結果	DABroker でエージェントの即時配信実行結果を応答する直前

(凡例)

取得内容のパラメタについて示します。なお、パラメタ一覧については、「[5.4.5\(1\) エージェント機能のパラメタ一覧](#)」を参照してください。

AgtRegName：エージェント名

AgtRegUser：エージェントを登録したユーザ ID

AgtExeName：エージェント実行名 (DABroker が割り当てる識別名 (番号))

AgtExeNote：メモ

AgtExeUser：エージェント実行を要求したユーザ ID

AgtExePID：実行プロセス ID

RC：各イベントの処理結果

0：正常終了 (インフォメーションのエージェントログを取得します。)

正：警告あり (インフォメーションのエージェントログを取得します。)

負：異常終了 (エラーのエージェントログを取得します。)

注※

イベントが (Cnt/End) の場合には出力されません。

6.3.3 XA トレースの取得形式 (共通)

XA インタフェースでのトラブルシュート情報は、XA トレースファイルとして取得されます。

XA トレースファイルは次のファイルに取得されます。

UNIX：DABroker 運用ディレクトリ/spool/db_xainfo.log

Windows：DABroker 運用ディレクトリ¥spool¥db_xainfo.log

XA トレースの取得方法については、次の箇所を参照してください。

UNIX の場合：「[3.4 DABroker の動作環境の設定](#)」

Windows の場合：「[4.4 DABroker 環境設定](#)」

XA トレースファイルの取得例を次に示します。

取得例

```
1999/10/20 10:27:12.156000 [226:225] [end :xa_commit ] rmid=1 flags=0
RMName="Oracle_XA"
xid=[00000001:10:8:313939392D31302D323031303A32363A34371906B8003C1BAB000000
000009400000006000000010000003C000000020000003C00000004000000810000008000
0003C0000000B0000003C0000000600000003C2A1B1C2D0303B600373A0A190AB8003D390BA7
002B57BB0023591201B7002B1906B60032B600403A072AB400]
return code = -4
1999/10/20 10:27:19.947000 [226:225] [end :xa_recover ] rmid=1 flags=0
count=1 RMName="Oracle_XA"
xid=[00000000:0:0:0000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000
return code = -5
```

- XXXX/XX/XX

取得年月日

- XX:XX:XX.XXXXXX

取得時間

- [XXX:XXX]

プロセス ID : スレッド ID

- [XXXX:FFFF]

XXXX : イベント種別

- "call" : 関数呼び出し
- "end" : 関数リターン

FFFF : XA 関数名

- rmid=rrrr

RMID

- flags=ffff

フラグパラメタ

- RMName="nnnn"

RM 名称

- xa_info="iiii"

XA 文字列

xa_open(), 又は xa_close()関数の場合だけ, 出力されます。

- xid=[ffff:gggg:bbbb:dddd]

XID

ffff : フォーマット ID

gggg : グローバルトランザクション ID 長

bbbb : ブランチ修飾子長

dddd : XID データ

xa_open(), 又は xa_close()関数以外の場合だけ, 出力されます。

- count=cccc

リカバリ情報取得カウント

xa_recover()関数の場合だけ, 出力されます。

- return code = rrrr

XA 関数の戻り値

イベント種別が"end"の場合だけ, 出力されます。

6.3.4 デーモンの生成・消滅・クリーンアップ実行結果ログ (共通)

障害発生からある一定の時間までをさかのぼって, 共有メモリがどういう状態であったかを示す情報が次のログファイルに取得されます。

UNIX : DABroker 運用ディレクトリ/spool/dabsvdemon.log

Windows : DABroker 運用ディレクトリ¥spool¥dabsvdemon.log

ログファイルの大きさは, 環境変数 DABSVLOGSIZE で指定します。指定方法は「3.5 DABroker の動作環境の設定項目 (UNIX)」または, 「4.5 DABroker 環境設定項目 (Windows)」参照してください。

メッセージは次の形式で出力されます。

日付	時間	出力ホスト名	製品名	メッセージを出力したプロセス番号	メッセージID	メッセージ内容
oct 12	17:52:15	ikn0205	DABroker	[10434]	KFDB302YM-I dabtcdmn	{引数の内容} start. [プロセスID]

次にデーモンの生成・消滅・クリーンアップ実行結果ログの出力例を以下に示します。

```
oct 26 15:40:00 ikn0205 DABroker [27190] KFDB302YM-I dabtcdmn [-z 1 -n 3 -c 1] start. [27190]
oct 26 15:40:03 ikn0205 DABroker [27174] KFDB301YM-I Clean up. [1, 0, 1], [0]
oct 26 15:40:08 ikn0205 DABroker [27174] KFDB301YM-I Clean up. [1, 0, 1], [0]
oct 26 15:40:13 ikn0205 DABroker [27174] KFDB301YM-I Clean up. [1, 0, 1], [0]
oct 26 15:40:18 ikn0205 DABroker [27174] KFDB301YM-I Clean up. [1, 0, 1], [0]
oct 26 15:40:23 ikn0205 DABroker [27174] KFDB301YM-I Clean up. [1, 0, 1], [0]
oct 26 15:40:28 ikn0205 DABroker [27174] KFDB301YM-I Clean up. [1, 0, 1], [0]
oct 26 15:40:33 ikn0205 DABroker [27174] KFDB301YM-I Clean up. [1, 0, 1], [0]
oct 26 15:40:38 ikn0205 DABroker [27174] KFDB301YM-I Clean up. [1, 0, 1], [0]
oct 26 15:40:43 ikn0205 DABroker [27174] KFDB301YM-I Clean up. [1, 0, 1], [0]
oct 26 15:40:48 ikn0205 DABroker [27174] KFDB301YM-I Clean up. [1, 0, 1], [0]
oct 26 15:40:53 ikn0205 DABroker [27174] KFDB301YM-I Clean up. [1, 0, 1], [0]
oct 26 15:40:58 ikn0205 DABroker [27174] KFDB301YM-I Clean up. [1, 0, 1], [0]
oct 26 15:40:00 ikn0205 DABroker [27190] KFDB303YM-I dabtcdmn end [27190]
oct 26 15:40:03 ikn0205 DABroker [27174] KFDB301YM-I Clean up. [1, 0, 1], [0]
```

出力されるメッセージ内容を以下に示します。

出力プログラム	DABroker
---------	----------

メッセージ ID	KFDB301YM-I
メッセージ	Clean up. {xxx,yyy,zzz} , {sss}
内容	<p>クライアントとの通信サーバデーモンのクリーンアップを実行しました。</p> <p>xxx：クリーンアップ前のデーモン数。</p> <p>yyy：クリーンアップしたデーモン数。</p> <p>zzz：クリーンアップ実行後のデーモン数。</p> <p>sss：DABroker の状態。</p> <p>0：正常実行中。</p> <p>1：正常終了中。</p> <p>2：計画終了中。</p> <p>3：強制終了中。</p>

出力プログラム	DABroker
メッセージ ID	KFDB302YM-I
メッセージ	dabtcpdmn {xxx} start. [yyyy]
内容	<p>クライアントとの通信サーバデーモンを起動しました。</p> <p>xxx：dabtcpdmn の引数</p> <p>yyyy：起動したプロセス ID</p>

出力プログラム	DABroker
メッセージ ID	KFDB303YM-I
メッセージ	dabtcpdmn end. [xxxx]
内容	<p>クライアントとの通信サーバデーモンを終了しました。</p> <p>yyyy：起動したプロセス ID</p>

注意事項

- DABroker 動作環境定義の DABMAXBLOCKUSERS に 2 以上の値が設定されている場合、DABroker は 1 つのプロセスで複数のユーザとの接続を実現しているため、クライアントが接続していない状態でも、クリーンアップの対象ログ (KFDB301YM-I メッセージ) が出力されます。
- 本ログ出力によりディスクを圧迫したくない場合は、DABroker 動作環境定義ファイル (dasysconf) の OTHERS 節に、DAB_DEMON_LOG_MODE=LIMITED の定義を追加することにより抑止することができます。

DAB_DEMON_LOG_MODE の設定方法については「[3.5.8 その他の設定 \(UNIX\)](#)」または、「[4.5.6 その他の設定 \(Windows\)](#)」を参照してください。

6.4 データベースアクセストレース (UNIX)

ここでは、UNIX の場合のデータベースアクセストレースについて説明します。

6.4.1 データベースアクセストレース選択オプション

ここでは、データベースアクセストレース情報を任意に選択して出力する場合に設定する項目について示します。

データベースアクセストレース情報を任意に選択して出力する機能を、データベースアクセストレース選択オプションといいます。

データベースアクセストレース選択オプションは、DABroker の動作環境の基本項目で設定します。設定した項目は、「3.5.1(3) DABSQL_TRCSIZE (データベースアクセストレースの取得)」の設定項目の値を「0」と指定した場合に有効になります。

拡張データベースアクセストレース機能使用時に出力できる項目については、「3.5.1(10) DABEXSQL_TRC_LINE (拡張データベースアクセストレースを取得する)」に 0、又は 1024~32767 を設定しておく必要があります。

データベースアクセストレース選択オプション使用時の注意事項

- データベースアクセストレース選択オプションでは、すべての設定項目の値を指定する必要はありません。指定を省略すると、標準値「ON」が仮定されます。
- すべての設定項目の値を「OFF」にし、「3.5.1(3) DABSQL_TRCSIZE (データベースアクセストレースの取得)」の設定項目の値を「0」と指定した場合も文法エラーにはなりません。この場合、データベースアクセストレースは取得しません。
- 「ON」が指定された設定項目に該当する、すべてのデータベースへの要求がなかった場合、データベースアクセストレースにはヘッダ（出力項目名）だけが出力されます。

(1) DABSQL_TRC_CONNECT (CONNECT 情報出力要求)

標準値

ON

指定範囲

ON 又は OFF (ON：取得します OFF：取得しません)

説明

データベースアクセストレースに、データベースへの接続要求の情報を取得するかどうかを設定します。

(2) DABSQL_TRC_DISCONNECT (DISCONNECT 情報出力要求)

標準値

ON

指定範囲

ON 又は OFF (ON:取得します OFF:取得しません)

説明

データベースアクセストレースに、トランザクションの正常終了と切り離し要求の情報を取得するかどうかを設定します。

(3) DABSQL_TRC_COMMIT (COMMIT 情報出力要求)

標準値

ON

指定範囲

ON 又は OFF (ON:取得します OFF:取得しません)

説明

データベースアクセストレースに、トランザクションの正常終了と更新の情報を取得するかどうかを設定します。

(4) DABSQL_TRC_ROLLBACK (ROLLBACK 情報出力要求)

標準値

ON

指定範囲

ON 又は OFF (ON:取得します OFF:取得しません)

説明

データベースアクセストレースに、トランザクションを取り消し、更新を無効にする要求の情報を取得するかどうかを設定します。

(5) DABSQL_TRC_PREPARE (PREPARE 情報出力要求)

標準値

ON

指定範囲

ON 又は OFF (ON:取得します OFF:取得しません)

説明

データベースアクセストレースに、SQL の前処理要求の情報を取得するかどうかを設定します。

(6) DABSQL_TRC_DESCRIBE (DESCRIBE 情報出力要求)

標準値

ON

指定範囲

ON 又は OFF (ON:取得します OFF:取得しません)

説明

データベースアクセストレースに、前処理した SQL の情報を取得する要求の情報を取得するかどうかを設定します。

注意事項

前処理した SQL の情報には、入力情報 (DESC_IN) と出力情報 (DESC_OUT) があります。どちらか一方だけの出力設定はできません。

(7) DABSQL_TRC_DECLARE (DECLARE 情報出力要求)

標準値

ON

指定範囲

ON 又は OFF (ON:取得します OFF:取得しません)

説明

データベースアクセストレースに、前処理した SELECT に対してカーソルを宣言する要求の情報を取得するかどうかを設定します。

(8) DABSQL_TRC_OPEN (OPEN 情報出力要求)

標準値

ON

指定範囲

ON 又は OFF (ON:取得します OFF:取得しません)

説明

データベースアクセストレースに、カーソルの位置付け要求の情報を取得するかどうかを設定します。

(9) DABSQL_TRC_FETCH (FETCH 情報出力要求)

標準値

ON

指定範囲

ON 又は OFF (ON:取得します OFF:取得しません)

説明

データベースアクセストレースに、行を読み込んでカーソルを進める要求の情報を取得するかどうかを設定します。

(10) DABSQL_TRC_EXECUTE (EXECUTE 情報出力要求)

標準値

ON

指定範囲

ON 又は OFF (ON：取得します OFF：取得しません)

説明

データベースアクセストレースに、前処理した SQL の実行要求の情報を取得するかどうかを設定します。

(11) DABSQL_TRC_CLOSE (CLOSE 情報出力要求)

標準値

ON

指定範囲

ON 又は OFF (ON：取得します OFF：取得しません)

説明

データベースアクセストレースに、カーソルを閉じる要求の情報を取得するかどうかを設定します。

(12) DABSQL_TRC_CANCEL (CANCEL 情報出力要求)

標準値

ON

指定範囲

ON 又は OFF (ON：取得します OFF：取得しません)

説明

データベースアクセストレースに、データベースに対する処理中断要求の情報を取得するかどうかを設定します。

(13) DABSQL_TRC_LOGGING (LOGGING 情報出力要求)

標準値

ON

指定範囲

ON 又は OFF (ON：取得します OFF：取得しません)

説明

データベースアクセストレースに LOGGING 要求の情報を取得するかどうかを設定します。

(14) DABSQL_TRC_LOGLESS (LOGLESS 情報出力要求)

標準値

ON

指定範囲

ON 又は OFF (ON : 取得します OFF : 取得しません)

説明

データベースアクセストレースに LOGLESS 要求の情報を取得するかどうかを設定します。

(15) DABSQL_TRC_ATTACH (ATTACH 情報出力要求)

標準値

ON

指定範囲

ON 又は OFF (ON : 取得します OFF : 取得しません)

説明

データベースアクセストレースに、サーバとのアクセスパス作成要求の情報を取得するかどうかを設定します。

この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

(16) DABSQL_TRC_ATTRGET (ATTRGET 情報出力要求)

標準値

ON

指定範囲

ON 又は OFF (ON : 取得します OFF : 取得しません)

説明

データベースアクセストレースに、ハンドルの属性取得要求の情報を取得するかどうかを設定します。

この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

(17) DABSQL_TRC_ATTRSET (ATTRSET 情報出力要求)

標準値

ON

指定範囲

ON 又は OFF (ON : 取得します OFF : 取得しません)

説明

データベースアクセストレースに、ハンドルの属性設定要求の情報を取得するかどうかを設定します。
この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

(18) DABSQL_TRC_BIND (BIND 情報出力要求)

標準値

ON

指定範囲

ON 又は OFF (ON : 取得します OFF : 取得しません)

説明

データベースアクセストレースに、パラメタのバインド要求の情報を取得するかどうかを設定します。
この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

(19) DABSQL_TRC_CHARFORM (CHARFORM 情報出力要求)

標準値

ON

指定範囲

ON 又は OFF (ON : 取得します OFF : 取得しません)

説明

データベースアクセストレースに、キャラクタセットフォームの取得要求の情報を取得するかどうかを設定します。
この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

(20) DABSQL_TRC_CHARID (CHARID 情報出力要求)

標準値

ON

指定範囲

ON 又は OFF (ON : 取得します OFF : 取得しません)

説明

データベースアクセストレースに、キャラクタセット ID の取得要求の情報を取得するかどうかを設定します。
この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

(21) DABSQL_TRC_DEFINE (DEFINE 情報出力要求)

標準値

ON

指定範囲

ON 又は OFF (ON：取得します OFF：取得しません)

説明

データベースアクセストレースに、パラメタの定義要求の情報を取得するかどうかを設定します。
この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

(22) DABSQL_TRC_DESALLOC (DESALLOC 情報出力要求)

標準値

ON

指定範囲

ON 又は OFF (ON：取得します OFF：取得しません)

説明

データベースアクセストレースに、記述子の領域確保の情報を取得するかどうかを設定します。
この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

(23) DABSQL_TRC_DESCANY (DESCANY 情報出力要求)

標準値

ON

指定範囲

ON 又は OFF (ON：取得します OFF：取得しません)

説明

データベースアクセストレースに、オブジェクトの記述要求の情報を取得するかどうかを設定します。
この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

(24) DABSQL_TRC_DESFREE (DESFREE 情報出力要求)

標準値

ON

指定範囲

ON 又は OFF (ON：取得します OFF：取得しません)

説明

データベースアクセストレースに、記述子の領域解放の情報を取得するかどうかを設定します。
この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

(25) DABSQL_TRC_DETACH (DETACH 情報出力要求)

標準値

ON

指定範囲

ON 又は OFF (ON : 取得します OFF : 取得しません)

説明

データベースアクセストレースに、サーバとのアクセスパス削除要求の情報を取得するかどうかを設定します。

この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

(26) DABSQL_TRC_ENVCREATE (ENVCREATE 情報出力要求)

標準値

ON

指定範囲

ON 又は OFF (ON : 取得します OFF : 取得しません)

説明

データベースアクセストレースに、環境作成要求の情報を取得するかどうかを設定します。

この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

(27) DABSQL_TRC_ERRGET (ERRGET 情報出力要求)

標準値

ON

指定範囲

ON 又は OFF (ON : 取得します OFF : 取得しません)

説明

データベースアクセストレースに、エラー情報取得要求の情報を取得するかどうかを設定します。

この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

(28) DABSQL_TRC_GETPIECE (GETPIECE 情報出力要求)

標準値

ON

指定範囲

ON 又は OFF (ON : 取得します OFF : 取得しません)

説明

データベースアクセストレースに、ピース情報取得要求の情報を取得するかどうかを設定します。

HiRDB 以外に接続している場合

この設定項目は、DABroker for Java からデータベースの機能を使用する場合だけ有効です。

トレースに、ピース情報取得要求の情報を取得するかどうかを設定します。

HiRDB に接続している場合

スカラ関数 POSITION 実行要求の情報を取得するかどうかを設定します。

この情報は、クライアントが DABroker for Java で次の条件すべてを満たす場合に有効になります。

- 接続しているデータベースが HiRDB で、HiRDB のサーバとクライアントのバージョンが共に 07-01 以上
- DABroker for Java のバージョンが 02-12 以上
- DABroker が暗黙的に次の SQL を実行した場合
SET 文を実行し、代入値にスカラ関数 POSITION を使用します。

(29) DABSQL_TRC_HDLALLOC (HDLALLOC 情報出力要求)

標準値

ON

指定範囲

ON 又は OFF (ON : 取得します OFF : 取得しません)

説明

データベースアクセストレースに、ハンドルの領域確保の情報を取得するかどうかを設定します。

この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

(30) DABSQL_TRC_HDLFREE (HDLFREE 情報出力要求)

標準値

ON

指定範囲

ON 又は OFF (ON : 取得します OFF : 取得しません)

説明

データベースアクセストレースに、ハンドルの領域解放の情報を取得するかどうかを設定します。
この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

(31) DABSQL_TRC_LOBCLOSE (LOBCLOSE 情報出力要求)

標準値

ON

指定範囲

ON 又は OFF (ON : 取得します OFF : 取得しません)

説明

データベースアクセストレースに、BFILE データのクローズ要求の情報を取得するかどうかを設定します。

この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

(32) DABSQL_TRC_LOBLEN (LOBLEN 情報出力要求)

標準値

ON

指定範囲

ON 又は OFF (ON : 取得します OFF : 取得しません)

説明

データベースアクセストレースに、LOB データの長さ取得要求の情報を取得するかどうかを設定します。

HiRDB 以外に接続している場合

この設定項目は、DABroker for Java からデータベースの機能を使用する場合だけ有効です。
トレースに、LOB データの長さ取得要求の情報を取得するかどうかを設定します。

HiRDB に接続している場合

スカラ関数 LENGTH 実行要求の情報を取得するかどうかを設定します。

この情報は、クライアントが DABroker for Java で次の条件すべてに一致する場合に有効になります。

- 接続しているデータベースが HiRDB で、HiRDB のサーバとクライアントのバージョンが共に 07-01 以上
- DABroker for Java のバージョンが 02-12 以上
- DABroker が暗黙的に次の SQL を実行した場合
SET 文を実行し、代入値にスカラ関数 LENGTH を使用します。

(33) DABSQL_TRC_LOBOPEN (LOBOPEN 情報出力要求)

標準値

ON

指定範囲

ON 又は OFF (ON：取得します OFF：取得しません)

説明

データベースアクセストレースに、BFILE データのオープン要求の情報を取得するかどうかを設定します。

この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

(34) DABSQL_TRC_LOBREAD (LOBREAD 情報出力要求)

標準値

ON

指定範囲

ON 又は OFF (ON：取得します OFF：取得しません)

説明

データベースアクセストレースに、LOB データの読み込み要求の情報を取得するかどうかを設定します。

HiRDB 以外に接続している場合

この設定項目は、DABroker for Java からデータベースの機能を使用する場合だけ有効です。

トレースに、LOB データの読み込み要求の情報を取得するかどうかを設定します。

HiRDB に接続している場合

スカラ関数 SUBSTR 実行要求の情報を取得するかどうかを設定します。

この情報は、クライアントが DABroker for Java で、次の条件すべてに一致する場合に有効になります。

- 接続しているデータベースが HiRDB で、HiRDB のサーバとクライアントのバージョンが共に 07-01 以上
- DABroker for Java のバージョンが 02-12 以上
- DABroker が暗黙的に下記 SQL を実行した場合
SET 文を実行し、代入値にスカラ関数 SUBSTR を使用します。

(35) DABSQL_TRC_LOBWRITE (LOBWRITE 情報出力要求)

標準値

ON

指定範囲

ON 又は OFF (ON : 取得します OFF : 取得しません)

説明

データベースアクセストレースに、LOB データの書き込み要求の情報を取得するかどうかを設定します。
この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

(36) DABSQL_TRC_PARAMGET (PARAMGET 情報出力要求)

標準値

ON

指定範囲

ON 又は OFF (ON : 取得します OFF : 取得しません)

説明

データベースアクセストレースに、パラメタ情報取得要求の情報を取得するかどうかを設定します。
この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

(37) DABSQL_TRC_SESBEGIN (SESBEGIN 情報出力要求)

標準値

ON

指定範囲

ON 又は OFF (ON : 取得します OFF : 取得しません)

説明

データベースアクセストレースに、セッション開始要求の情報を取得するかどうかを設定します。
この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

(38) DABSQL_TRC_SESEND (SESEND 情報出力要求)

標準値

ON

指定範囲

ON 又は OFF (ON : 取得します OFF : 取得しません)

説明

データベースアクセストレースに、セッション終了要求の情報を取得するかどうかを設定します。
この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

(39) DABSQL_TRC_SETPIECE (SETPIECE 情報出力要求)

標準値

ON

指定範囲

ON 又は OFF (ON：取得します OFF：取得しません)

説明

データベースアクセストレースに、ピース情報設定要求の情報を取得するかどうかを設定します。
この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

(40) DABSQL_TRC_GETDIAG (GETDIAG 情報出力要求)

標準値

ON

指定範囲

ON 又は OFF (ON：取得します OFF：取得しません)

説明

データベースアクセストレースに、エラー情報取得要求の情報を取得するかどうかを設定します。
この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

(41) DABSQL_TRC_AUTO_COMMIT_OFF (AUTO COMMIT OFF 出力要求)

標準値

ON

指定範囲

ON 又は OFF (ON：取得します OFF：取得しません)

説明

データベースアクセストレースに、AUTO COMMIT OFF 設定要求の情報を取得するかどうかを設定します。
この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

(42) DABSQL_TRC_AUTO_COMMIT_ON (AUTO COMMIT ON 出力要求)

標準値

ON

指定範囲

ON 又は OFF (ON : 取得します OFF : 取得しません)

説明

データベースアクセストレースに、AUTO COMMIT ON 設定要求の情報を取得するかどうかを設定します。

この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

(43) DABSQL_TRC_GET_INFO (GET INFO 出力要求)

標準値

ON

指定範囲

ON 又は OFF (ON : 取得します OFF : 取得しません)

説明

データベースアクセストレースに、接続後の動作情報取得要求の情報を取得するかどうかを設定します。

この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

(44) DABSQL_TRC_SET_INFO (SET INFO 出力要求)

標準値

ON

指定範囲

ON 又は OFF (ON : 取得します OFF : 取得しません)

説明

データベースアクセストレースに、接続後の動作情報設定要求の情報を取得するかどうかを設定します。

この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

(45) DABSQL_TRC_EXECUTE_DIRECT (EXECUTE DIRECT 出力要求)

標準値

ON

指定範囲

ON 又は OFF (ON : 取得します OFF : 取得しません)

説明

データベースアクセストレースに、SQL の前処理後に実行する要求の情報を取得するかどうかを設定します。

この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

6.4.2 データベースアクセストレースの表示

コマンド形式

```
# dabdbtrc [-u ユーザID] [-p アプリケーションの名称]
```

機能

DABroker がデータベースに要求した処理、及びその処理に対するデータベースからの応答（アクセストレース）を表示します。

なお、アクセストレースの取得は、DBMS に要求を出す直前及び DBMS から応答が返ってきた直後に、DABroker が行っています。

使用できるユーザ

DABroker の管理ユーザ

オプション

指定なし

すべてのアプリケーションからの要求に対して行ったデータベースアクセストレースを表示します。

-u ユーザ ID

指定したユーザ ID を持つアプリケーションからの要求で行った、データベースアクセストレースを表示します。

-p アプリケーションの名称

指定した名称のアプリケーションからの要求で行った、データベースアクセストレースを表示します。指定できる名称は次のとおりです。

アプリケーションの種類	指定する名称
DABroker 関連製品で開発したアプリケーションからアクセスする場合	アプリケーションで設定した名称 (各 DABroker 関連マニュアル参照)
DBPARTNER/Client 又は DBPARTNER/Web からアクセスする場合	DBPARTNER/Client 又は DBPARTNER/Web の識別情報
エージェント処理を実行した場合	エージェント実行名 (16 バイトの固定文字列)

ここで入力した名称は、PAPNAME に出力されます。

出力例

```
# dabdbtrc -uHITACHI
UID=HITACHI PAPNAME=XXXXX DBNAME=HiRDB
DATE        TIME        EVT          RETCODE      INFORMATION
1998/04/02 11:38:30.10 PREPARE
(SQL)SELECT * FROM ZAIKO
1998/04/02 11:38:31.22 PREPARE          0
1998/04/02 11:38:31.30 DESCRIBE
1998/04/02 11:38:32.43 DESCRIBE          0
1998/04/02 11:38:32.50 DECLARE
1998/04/02 11:38:33.70 DECLARE          0
1998/04/02 11:38:34.00 OPEN
1998/04/02 11:38:36.22 OPEN          0
1998/04/02 11:38:36.50 FETCH          BLOCK COUNT(1)
1998/04/02 11:38:38.24 FETCH          0          BLOCK COUNT(1)
1998/04/02 11:38:38.70 FETCH          BLOCK COUNT(1)
1998/04/02 11:38:40.22 FETCH          100         BLOCK COUNT(0)
1998/04/02 11:38:40.50 CLOSE
1998/04/02 11:38:42.22 CLOSE          0
1998/04/02 11:38:50.55 COMMIT
1998/04/02 11:38:52.74 COMMIT          0
1998/04/02 11:42:12.32 DISCONNECT
```

UID=XXXXXXXX

ユーザ ID

PAPNAME=XXXXXXXX

アプリケーションを識別するための文字列が表示されます。表示される文字列は、-p オプションで指定できる名称です。

DBNAME=XXXXXXXX

接続先データベース

次のどれかが表示されます。

- HiRDB : HiRDB
- ORACLE7 : Oracle8, Oracle8i, Oracle9i, Oracle10g, Oracle11g, もしくは Oracle12c で Oracle Call Interface 7 を使用している場合, 又は Oracle19c で Oracle Call Interface 8i を使用している場合。
- ORACLE8i : Oracle10g, Oracle11g, Oracle12c, 又は Oracle19c で Oracle Call Interface 8i を使用している場合。
- RDA_Link_GW : RDA Link for Gateway
- DBCSV : Database Connection Server

DATE

アクセスの発生日 (年/月/日)

TIME

アクセスの発生時刻 (時:分:秒.n)

n は n/100 秒を表します。

EVT

DABroker がデータベースに要求した処理の内容

次のどれかが表示されます。

EVT の表示	処 理 内 容
CONNECT	データベースと接続します。 分散トランザクション連携機能使用時は、データベースに対するアクセス要求ではなく、アプリケーションからの CONNECT 要求に対応したイベントになります。
PREPARE	SQL の前処理をします。
DESCRIBE	前処理した SQL の情報を返します。
DECLARE	前処理した SELECT に対してカーソルを宣言します。
OPEN	カーソルを位置付けます。
FETCH	行を読み込んでカーソルを進めます。
CLOSE	カーソルを閉じます。
EXECUTE	前処理した SQL を実行します。
COMMIT	トランザクションを正常終了、更新を有効にします。
ROLLBACK	トランザクションを取り消し、更新を無効にします。
DISCONNECT	トランザクションを正常終了、切り離します。 分散トランザクション連携機能使用時は、データベースに対するアクセス要求ではなく、アプリケーションからの DISCONNECT 要求に対応したイベントになります。
CANCEL	DBMS に対し、処理を中断します。

データベースアクセストレース情報を任意に選択して出力することもできます。任意に選択して出力する場合は、DABroker 動作環境定義ファイルに、データベースアクセストレース選択オプションの項目を設定する必要があります。データベースアクセストレース選択オプションについては、[「6.4.1 データベースアクセストレース選択オプション」](#)を参照してください。

RETCODE

DABroker が要求した処理に対して、DBMS から返されるリターンコード

- 0 : 正常終了
- 0 以外 : 異常終了又はメッセージ付きの正常終了

なお、DABroker から DBMS への要求の送信を表す行では、RETCODE は空白になります。また、RDA Link for Gateway の場合は、RETCODE は空白になります。

INFORMATION

Fetch 時の行取得要求数及び取得結果の情報

「BLOCK COUNT (XXX)」の形式で出力されます。

(SQL)

アプリケーションから要求のあった SQL の内容。この項目は、必要に応じて表示されます。

注意事項

アプリケーションからの要求量 (SQL 文) が多いほど、データベースアクセストレースのファイル容量も多くなります。トレースは、1 レコード 80 バイトで表示されます。アプリケーションからの要求 (SQL 文) が多い場合は、折り返して表示します。データベースアクセストレースのファイル容量の目安として、概算式を次に示します。

(概算式)

$$\{(\text{アプリケーションからの要求 (SQL 長など)} * 2) + 2 \lfloor \frac{\text{アプリケーションからの要求 SQL 長}}{73} \rfloor - 1 \} * 80 \text{ [バイト]}$$

注 下線は、SQL 長が 73 バイト以上の場合です。

あらかじめ、この概算式で計算したサイズ以上のハードディスク容量を用意しておいてください。また、アプリケーションからの要求量が多いほど、トレース表示が終わるまでにも時間がかかります。御注意ください。

6.4.3 拡張データベースアクセストレース機能

(1) 機能概要

拡張データベースアクセストレースでは、従来の DABroker のデータベースアクセストレースと比較して、次の機能が拡張されています。

- データベースへの接続ごと (接続から切断まで) にトレースを取得できます。
- データベース接続中でも、トレースファイルの参照やコピーができます。
- マルチスレッドでアプリケーションが動作している場合、データベースアクセスを実行したスレッドを特定できます。
- DABroker 動作環境定義ファイルの設定によって、出力するイベントの種類を選択できます。
- DABroker 動作環境定義ファイルの設定によって、ファイルの最大容量を設定できます。トレース情報が最大容量を超える場合、最大 2 世代で取得ファイルを切り替えてトレースを取得できます。

(2) 適用範囲

拡張データベースアクセストレース機能は、次のような場合に有効です。

- データベースのアクセス中に障害が発生した場合、迅速に障害を切り分ける必要がある。
- 24 時間稼働などのために、稼働中のアプリケーションを終了しないで障害を調査する必要がある。
- マルチスレッド環境でアプリケーションが動作していて、スレッドごとのデータベースアクセス履歴を取得したい。
- コネクションプーリングなどを使用していて、データベースへの接続ごとのデータベースアクセス履歴を取得する必要がある。

- 障害の再現調査ができないため、常時トレースを取得する必要がある。

(3) DABroker 動作環境定義

拡張データベースアクセストレースを取得する設定は、DABroker 動作環境定義ファイルで設定します。

DABroker 動作環境定義ファイルでの設定方法については、「[3.5 DABroker の動作環境の設定項目](#)」を参照してください。

(4) 出力形式

拡張データベースアクセストレースのファイル出力形式を次に示します。

Oracle19c で Oracle7 接続の場合、Oracle12c 以前とは出力するイベントが変わるなど、出力内容が異なります。

- ヘッダ情報

```

*-----*
*- DataBase Access Information (DRV)          -*
*- Date   YYYY/MM/DD hh:mm:ss.nnnnnn        -*
*-----*
DABroker Connect ID   : AAAAA(BBBBB)
Process ID            : CCCCC
UserID                : DDDDD
Client Name           : EEEEE
(DB別出力情報)
Lang Mode             : GGGGG

```

- 個々のトレース情報

```

THREAD-ID  CID  EVT  START-TIME  END-TIME  RETCODE
BLOCKCNT  HiRDB_CONNECTION_ID (SQL)SS...SS

```

DRV

DABroker のデータベースアクセスドライバの種別に従い、次の識別子が出力されます。

HiRDB 接続の場合

HiRDB Driver

Oracle8i 接続の場合

ORACLE8i Driver[※] (Oracle Call Interface 8i を使用)

Oracle7 接続の場合

ORACLE7 Driver (Oracle19c 以外で Oracle Call Interface 7 を使用している場合、又は Oracle19c で Oracle Call Interface 8i を使用している場合)

ODBC3.0 の場合

ODBC30

RDA Link for Gateway 経由メインフレーム系データベース接続の場合

RDALink Driver

Database Connection Server 経由メインフレーム系データベース接続の場合

DBS

注※

DABroker for Java から Oracle8i の機能を使用する場合だけ出力されます。

Date

ファイル作成日時 (YYYY/MM/DD hh:mm:ss.nnnnnn) が出力されます。

- YYYY：西暦年，MM：月，DD：日
- hh：時，mm：分，ss：秒，nnnnnn：マイクロ秒

DABroker Connect ID

データベース接続識別子が出力されます。

Process ID

実行プロセス ID が出力されます。

UserID

接続ユーザ ID が出力されます。

DABroker for Java で ORACLE の OS 認証機能を使用した場合は、「OS authority」となります。

Client Name

UAP 名称が出力されます。名称が設定されていない場合は空白になります。

(DB 別出力情報)

接続データベースによって異なる情報が出力されます。出力内容について、表 6-2 に示します。

表 6-2 接続データベースごとのトレース出力内容

接続データベース	出力内容	説明
HiRDB	PDHOST :	HiRDB 接続時の PDHOST の設定内容です。未設定の場合は、"Not Indicator"が出力されます。
	PDNAMEPORT :	HiRDB 接続時の PDNAMEPORT の設定内容です。未設定の場合は、"Not Indicator"が出力されます。
Oracle	SQLNET :	Oracle リスナー名が出力されます。未設定の場合は、"Not Indicator"が出力されます。
RDA Link	Access DataBase :	RDA Link で接続しているホスト DB 名称が出力されます。 VOS3 XDM RD E2 接続の場合：VOS3 XDM/RD E2 VOS1 RDB1 E2 接続の場合：VOS1 RDB1 E2 VOSK SQL/K 接続の場合：VOSK SQL/K 未設定の場合：Unknown DBMS

接続データベース	出力内容	説明
	Node Name :	データベース識別子の情報が出力されます。未設定の場合は, "Not Indicator"が出力されます。
Database Connction Server	Access DataBase :	接続しているホスト DB 名称が出力されます。 VOS3 XDM RD E2 接続の場合 : VOS3 XDM/RD E2 VOS3 XDM SD E2 接続の場合 : VOS3 XDM/SD E2 VOS1 RDB1 E2 接続の場合 : VOS1 RDB1 E2 VOS1 PDMII E2 接続の場合 : VOS1 PDM2 E2 VOSK SQL/K 接続の場合 : VOSK SQL/K VOSK スプールファイル接続の場合 : VOSK SPOOL 未設定の場合 : Unknown DBMS
	Node Name :	データベース識別子の情報が出力されます。未設定の場合は, "Not Indicator"が出力されます。

Lang Mode

実行言語モードが出力されます。

THREAD-ID

実行スレッド ID が出力されます。

CID

カーソル番号が出力されます。

EVT

DABroker がデータベースに要求した処理の内容です。

出力するイベントの種類は任意に選択できます。

選択できるイベントについては, 「[6.4.1 データベースアクセストレース選択オプション](#)」を参照してください。

また, 拡張データベースアクセストレースで出力イベントの種類を変更すると, 従来のデータベースアクセストレースの出力イベントの種類も変更されます。

すべてのイベント出力を OFF にした場合はヘッダだけが出力されます。

次のイベントは, 既存のデータベースアクセストレースにはありません。また, イベント出力を一つでも ON にした場合, GET_DIAG イベントの出力を抑止できません。

ATTACH : サーバとのアクセスパス作成要求を実行します。

ATTRGET : ハンドル属性の取得要求を実行します。

ATTRSET : ハンドル属性の設定要求を実行します。

AUTOCOMOFF : AUTO COMMIT モードの解除要求を実行します。

AUTOCOMON : AUTO COMMIT モードの設定要求を実行します。

BIND : パラメタのバインド要求を実行します。

CHARFORM：キャラクタセットフォームの取得要求を実行します。
CHARID：キャラクタセット ID の取得要求を実行します。
DEFINE：パラメタの定義要求を実行します。
DESALLOC：記述子の領域確保を実行します。
DESC_OUT：前処理した SQL の出力情報を返します。
DESC_IN：前処理した SQL の入力情報を返します。
DESCANY：オブジェクトの記述要求を実行します。
DESFREE：記述子の領域解放を実行します。
DETACH：データベースアクセスパス削除要求を実行します。
ENVCREATE：環境作成要求を実行します。
ERRGET：エラー情報取得要求を実行します。
EXECDIRECT：SQL の前処理後の実行処理です。
GET_DIAG：エラー詳細情報の取得処理を実行します。
GETINFO：接続後の動作情報の取得要求を実行します。
GETPIECE：ピース情報の取得要求を実行します。
HDLALLOC：ハンドルの領域確保を実行します。
HDLFREE：ハンドルの領域解放を実行します。
LOBCLOSE：BFILE データのクローズ要求を実行します。
LOBLEN：LOB データの長さ取得要求を実行します。
LOBOPEN：BFILE データのオープン要求を実行します。
LOBREAD：LOB データの読み込み要求を実行します。
LOBWRITE：LOB データの書き込み要求を実行します。
PARAMGET：パラメタ情報の取得要求を実行します。
SESBEGIN：セッション開始要求を実行します。
SESEND：セッション終了要求を実行します。
SETINFO：接続後の動作情報の設定要求を実行します。
SETPIECE：ピース情報の設定要求を実行します。

START-TIME

アクセス開始日時 (YYYY/MM/DD hh:mm:ss.nnnnnn) が出力されます。

- YYYY：西暦年，MM：月，DD：日
- hh：時，mm：分，ss：秒，nnnnnn：マイクロ秒

END-TIME

アクセス終了日時 (YYYY/MM/DD hh:mm:ss.nnnnnn) が出力されます。

- YYYY：西暦年，MM：月，DD：日
- hh：時，mm：分，ss：秒，nnnnnn：マイクロ秒

RETCODE

データベースから返されるリターンコードが出力されます。

BLOCKCNT

FETCH 時の行取得件数が出力されます。

HiRDB_CONNECTION_ID

(a)HiRDB サーバに接続する場合

コネクション ID (aaaa : bbbbb : cccc) が出力されます。

- aaaa : フロントエンドサーバ名 (HiRDB/Paraller Server 使用時), 又はシングルサーバ (HiRDB/Single Server 使用時)
- bbbbb : aaaa で示すサーバの接続通番
- cccc : aaaa で示すサーバのプロセス ID, 及びコネクション ID プロセス ID とコネクション ID は, HiRDB クライアントのバージョンが 07-01 以降で, サーバが HiRDB の場合だけ出力されます。HiRDB クライアントのバージョンが 07-01 未満で, サーバが HiRDB の場合は, ヘッダ (HiRDB_CONNECTION_ID) だけが出力され, コネクション ID は出力されません。なお, この情報は, CONNECT, COMMIT, ROLLBACK, PREPARE, EXECUTE, 又は DIRECT を実行したときに, 最新の情報に更新されます。

(b)HiRDB クライアント経由で XDM/RD E2 に接続する場合

コネクション ID (aaaa : bbbbb : cccc) が出力されます。

- aaaa : JOB 名称
- bbbbb : コネクト通番
- cccc : サーバユーザ番号

この情報は, HiRDB クライアントのバージョンが 07-02 以降で, サーバが XDM/RD E2 11-00 以降, かつ, Database Connection Server05-00 以降の場合のみ出力されます。

この条件に満たない場合, ヘッダ (HiRDB_CONNECTION_ID) だけが出力され, コネクション ID は出力されません。なお, この情報は, CONNECT, COMMIT, ROLLBACK, PREPARE, EXECUTE, 又は DIRECT を実行したときに, 最新の情報に更新されます。

(SQL)

クライアントから要求のあった SQL の内容が出力されます。この項目は, 必要に応じて出力されます。

(5) ファイル名称

拡張データベースアクセストレースのファイルは, 次のディレクトリに作成されます。

DABroker運用ディレクトリ/spool/db_access

拡張データベースアクセストレースのファイル名称は, 次の規則に従います。

spdb_XXXXXXXX_YYYYYYYY_****ZZZZ.IIIII

****：接続データベース名称

HiRDB の場合

HiRDB

Oracle8i 接続の場合 (Oracle Call Interface 8i を使用)

ORACLE8i※

Oracle7 の場合 (Oracle19c 以外で Oracle Call Interface 7 を使用している場合、又は Oracle19c で Oracle Call Interface 8i を使用している場合)

ORACLE7

ODBC3.0 の場合

ODBC30

RDA Link for Gateway 経由メインフレーム系データベースの場合

RDALink

Database Connection Server 経由メインフレーム系データベースの場合

DBS

注※

DABroker for Java から Oracle8i の機能を使用する場合だけ出力されます。

XXXXXXXX：実行プロセス ID

YYYYYYYY：データベース接続識別子

ZZZZZ：識別順序番号

同じプロセス ID で、かつ同じデータベース接続識別子のファイルがある場合に、00000 から 32767 までの番号を付加します。

IIIII：ファイル拡張子

log：最新のトレースファイル

logold：1 世代前のトレースファイル

(6) ファイルの運用方法

- 拡張データベースアクセストレースは、DABroker 動作環境定義ファイルで設定した行数を超える出力があった場合、現在のトレースファイルをバックアップファイルとして別名で保存します(バックアップファイルの名称については、「6.4.3(5) ファイル名称」参照)。トレース内容は、トレースファイルの先頭から再び出力します。
- バックアップファイルは 1 世代前までしか残しません。このため、既にバックアップファイルがあり、現在のトレースファイルが設定した行数を超えると、バックアップファイルを上書きします。バックアップファイルを保存する必要がある場合は、あらかじめ別のディレクトリに退避してください。
- トレースファイルは実行しているコネクションの数だけ作成されます。

- トレースファイルは自動的に削除されません。ユーザが不要になった時点で削除してください。
- ハードディスクの空き容量が十分でない場合、トレースの出力が停止されることがあります。

6.5 データベースアクセストレース (Windows)

ここでは、Windows の場合のデータベースアクセストレースについて説明します。

6.5.1 アクセストレース設定

アクセストレースに取得する情報の種類を選択するには、[DABroker 動作環境設定] ダイアログボックスの [共通設定] タブから、[アクセストレース] ボタンを押して、[アクセストレース設定] ダイアログボックスを表示します。アクセストレースを取得する場合は、[アクセストレース設定] ダイアログボックスで、取得する情報の種類を必ず一つ以上 ON (チェックあり) にしてください。

拡張データベースアクセストレース機能使用時に出力できる項目については、「4.5.1(10) 拡張データベースアクセストレースを取得する (DABEXSQL_TRC_LINE)」を ON (チェックあり) にしておく必要があります。

取得できる情報の種類は次のとおりです。すべての情報の初期値は ON (チェックあり) です。

(1) CANCEL (DABSQL_TRC_CANCEL)

標準値

ON (チェックあり)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)

(ON : 取得します OFF : 取得しません)

説明

データベースアクセストレースに、DBMS に対する処理中断要求の情報を取得するかどうかを設定します。

(2) CLOSE (DABSQL_TRC_CLOSE)

標準値

ON (チェックあり)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)

(ON : 取得します OFF : 取得しません)

説明

データベースアクセストレースに、カーソルを閉じる要求の情報を取得するかどうかを設定します。

(3) COMMIT (DABSQL_TRC_COMMIT)

標準値

ON (チェックあり)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)

(ON:取得します OFF:取得しません)

説明

データベースアクセストレースに、トランザクションの正常終了、更新を有効にする要求の情報を取得するかどうかを設定します。

(4) CONNECT (DABSQL_TRC_CONNECT)

標準値

ON (チェックあり)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)

(ON:取得します OFF:取得しません)

説明

データベースアクセストレースに、データベースとの接続要求の情報を取得するかどうかを設定します。

(5) DECLARE (DABSQL_TRC_DECLARE)

標準値

ON (チェックあり)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)

(ON:取得します OFF:取得しません)

説明

データベースアクセストレースに、前処理した SELECT に対してカーソルを宣言する要求の情報を取得するかどうかを設定します。

(6) DESCRIBE (DABSQL_TRC_DESCRIBE)

標準値

ON (チェックあり)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)

(ON:取得します OFF:取得しません)

説明

データベースアクセストレースに、前処理した SQL の情報を取得する要求の情報を取得するかどうかを設定します。

(7) DISCONNECT (DABSQL_TRC_DISCONNECT)

標準値

ON (チェックあり)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)

(ON:取得します OFF:取得しません)

説明

データベースアクセストレースに、トランザクションを正常終了、切り離し要求の情報を取得するかどうかを設定します。

(8) EXECUTE (DABSQL_TRC_EXECUTE)

標準値

ON (チェックあり)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)

(ON:取得します OFF:取得しません)

説明

データベースアクセストレースに、前処理した SQL の実行要求の情報を取得するかどうかを設定します。

(9) FETCH (DABSQL_TRC_FETCH)

標準値

ON (チェックあり)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)

(ON:取得します OFF:取得しません)

説明

データベースアクセストレースに、行を読み込んでカーソルを進める要求の情報を取得するかどうかを設定します。

(10) LOGGING (DABSQL_TRC_LOGGING)

標準値

ON (チェックあり)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)

(ON: 取得します OFF: 取得しません)

説明

データベースアクセストレースに LOGGING 要求の情報を取得するかどうかを設定します。

(11) LOGLESS (DABSQL_TRC_LOGLESS)

標準値

ON (チェックあり)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)

(ON: 取得します OFF: 取得しません)

説明

データベースアクセストレースに LOGLESS 要求の情報を取得するかどうかを設定します。

(12) PREPARE (DABSQL_TRC_PREPARE)

標準値

ON (チェックあり)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)

(ON: 取得します OFF: 取得しません)

説明

データベースアクセストレースに、SQL の前処理要求の情報を取得するかどうかを設定します。

(13) ROLLBACK (DABSQL_TRC_ROLLBACK)

標準値

ON (チェックあり)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)

(ON: 取得します OFF: 取得しません)

説明

データベースアクセストレースに、トランザクションを取り消し、更新を無効にする要求の情報を取得するかどうかを設定します。

(14) OPEN (DABSQL_TRC_OPEN)

標準値

ON (チェックあり)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)
(ON : 取得します OFF : 取得しません)

説明

データベースアクセストレースに、カーソルの位置付け要求の情報を取得するかどうかを設定します。

(15) ATTACH (DABSQL_TRC_ATTACH)

標準値

ON (チェックあり)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)
(ON : 取得します OFF : 取得しません)

説明

データベースアクセストレースに、サーバとのアクセスパス作成要求の情報を取得するかどうかを設定します。

この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

この設定項目は、DABroker for Java から Oracle8i の機能を使用する場合だけ有効です。

(16) ATTRGET (DABSQL_TRC_ATTRGET)

標準値

ON (チェックあり)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)
(ON : 取得します OFF : 取得しません)

説明

データベースアクセストレースに、ハンドルの属性取得要求の情報を取得するかどうかを設定します。

この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

この設定項目は、DABroker for Java から Oracle8i の機能を使用する場合だけ有効です。

(17) ATTRSET (DABSQL_TRC_ATTRSET)

標準値

ON (チェックあり)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)

(ON: 取得します OFF: 取得しません)

説明

データベースアクセストレースに、ハンドルの属性設定要求の情報を取得するかどうかを設定します。

この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

この設定項目は、DABroker for Java から Oracle8i の機能を使用する場合だけ有効です。

(18) BIND (DABSQL_TRC_BIND)

標準値

ON (チェックあり)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)

(ON: 取得します OFF: 取得しません)

説明

データベースアクセストレースに、パラメタのバインド要求の情報を取得するかどうかを設定します。

この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

(19) CHARFORM (DABSQL_TRC_CHARFORM)

標準値

ON (チェックあり)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)

(ON: 取得します OFF: 取得しません)

説明

データベースアクセストレースに、キャラクタセットフォームの取得要求の情報を取得するかどうかを設定します。

この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

この設定項目は、DABroker for Java から Oracle8i の機能を使用する場合だけ有効です。

(20) CHARID (DABSQL_TRC_CHARID)

標準値

ON (チェックあり)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)

(ON : 取得します OFF : 取得しません)

説明

データベースアクセストレースに、キャラクタセット ID の取得要求の情報を取得するかどうかを設定します。

この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

この設定項目は、DABroker for Java から Oracle8i の機能を使用する場合だけ有効です。

(21) DEFINE (DABSQL_TRC_DEFINE)

標準値

ON (チェックあり)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)

(ON : 取得します OFF : 取得しません)

説明

データベースアクセストレースに、パラメタの定義要求の情報を取得するかどうかを設定します。

この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

この設定項目は、DABroker for Java から Oracle8i の機能を使用する場合だけ有効です。

(22) DESALLOC (DABSQL_TRC_DESALLOC)

標準値

ON (チェックあり)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)

(ON : 取得します OFF : 取得しません)

説明

データベースアクセストレースに、記述子の領域確保の情報を取得するかどうかを設定します。

この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

この設定項目は、DABroker for Java から Oracle8i の機能を使用する場合だけ有効です。

(23) DESCANY (DABSQL_TRC_DESCANY)

標準値

ON (チェックあり)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)

(ON : 取得します OFF : 取得しません)

説明

データベースアクセストレースに、オブジェクトの記述要求の情報を取得するかどうかを設定します。

この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

この設定項目は、DABroker for Java から Oracle8i の機能を使用する場合だけ有効です。

(24) DESFREE (DABSQL_TRC_DESFREE)

標準値

ON (チェックあり)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)

(ON : 取得します OFF : 取得しません)

説明

データベースアクセストレースに、記述子の領域解放の情報を取得するかどうかを設定します。

この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

この設定項目は、DABroker for Java から Oracle8i の機能を使用する場合だけ有効です。

(25) DETACH (DABSQL_TRC_DETACH)

標準値

ON (チェックあり)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)

(ON : 取得します OFF : 取得しません)

説明

データベースアクセストレースに、サーバとのアクセスパス削除要求の情報を取得するかどうかを設定します。

この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

この設定項目は、DABroker for Java から Oracle8i の機能を使用する場合だけ有効です。

(26) ENVCREATE (DABSQL_TRC_ENVCREATE)

標準値

ON (チェックあり)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)

(ON: 取得します OFF: 取得しません)

説明

データベースアクセストレースに、環境作成要求の情報を取得するかどうかを設定します。

この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

この設定項目は、DABroker for Java から Oracle8i の機能を使用する場合だけ有効です。

(27) ERRGET (DABSQL_TRC_ERRGET)

標準値

ON (チェックあり)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)

(ON: 取得します OFF: 取得しません)

説明

データベースアクセストレースに、エラー情報取得要求の情報を取得するかどうかを設定します。

この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

この設定項目は、DABroker for Java から Oracle8i の機能を使用する場合だけ有効です。

(28) GETPIECE (DABSQL_TRC_GETPIECE)

標準値

ON (チェックあり)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)

(ON: 取得します OFF: 取得しません)

説明

データベースアクセストレースに、ピース情報取得要求の情報を取得するかどうかを設定します。

この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

HiRDB 以外に接続している場合

この設定項目は、DABroker for Java からデータベースの機能を使用する場合だけ有効です。

トレースに、ピース情報取得要求の情報を取得するかどうかを設定します。

HiRDB に接続している場合

スカラ関数 POSITION 実行要求の情報を取得するかどうかを設定します。

この情報は、クライアントが DABroker for Java で次の条件すべてを満たす場合に有効になります。

- 接続しているデータベースが HiRDB で、HiRDB のサーバとクライアントのバージョンが共に 07-01 以上
- DABroker for Java のバージョンが 02-12 以上
- DABroker が暗黙的に次の SQL を実行した場合
SET 文を実行し、代入値にスカラ関数 POSITION を使用します。

(29) HDLALLOC (DABSQL_TRC_HDLALLOC)

標準値

ON (チェックあり)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)

(ON : 取得します OFF : 取得しません)

説明

データベースアクセストレースに、ハンドルの領域確保の情報を取得するかどうかを設定します。

この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

この設定項目は、DABroker for Java から Oracle8i の機能を使用する場合だけ有効です。

(30) HDLFREE (DABSQL_TRC_HDLFREE)

標準値

ON (チェックあり)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)

(ON : 取得します OFF : 取得しません)

説明

データベースアクセストレースに、ハンドルの領域解放の情報を取得するかどうかを設定します。

この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

この設定項目は、DABroker for Java から Oracle8i の機能を使用する場合だけ有効です。

(31) LOBCLOSE (DABSQL_TRC_LOBCLOSE)

標準値

ON (チェックあり)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)
(ON: 取得します OFF: 取得しません)

説明

データベースアクセストレースに、BFILE データのクローズ要求の情報を取得するかどうかを設定します。

この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

この設定項目は、DABroker for Java から Oracle8i の機能を使用する場合だけ有効です。

(32) LOBLEN (DABSQL_TRC_LOBLEN)

標準値

ON (チェックあり)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)
(ON: 取得します OFF: 取得しません)

説明

データベースアクセストレースに、LOB データの長さ取得要求の情報を取得するかどうかを設定します。
この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

HiRDB 以外に接続している場合

この設定項目は、DABroker for Java からデータベースの機能を使用する場合だけ有効です。
トレースに、LOB データの長さ取得要求の情報を取得するかどうかを設定します。

HiRDB に接続している場合

スカラ関数 LENGTH 実行要求の情報を取得するかどうかを設定します。

この情報は、クライアントが DABroker for Java で次の条件すべてに一致する場合に有効になります。

- 接続しているデータベースが HiRDB で、HiRDB のサーバとクライアントのバージョンが共に 07-01 以上
- DABroker for Java のバージョンが 02-12 以上
- DABroker が暗黙的に次の SQL を実行した場合
SET 文を実行し、代入値にスカラ関数 LENGTH を使用します。

(33) LOBOPEN (DABSQL_TRC_LOBOPEN)

標準値

ON (チェックあり)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)
(ON: 取得します OFF: 取得しません)

説明

データベースアクセストレースに、BFILE データのオープン要求の情報を取得するかどうかを設定します。

この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

この設定項目は、DABroker for Java から Oracle8i の機能を使用する場合だけ有効です。

(34) LOBREAD (DABSQL_TRC_LOBREAD)

標準値

ON (チェックあり)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)
(ON: 取得します OFF: 取得しません)

説明

データベースアクセストレースに、LOB データの読み込み要求の情報を取得するかどうかを設定します。
この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

HiRDB 以外に接続している場合

この設定項目は、DABroker for Java からデータベースの機能を使用する場合だけ有効です。
トレースに、LOB データの読み込み要求の情報を取得するかどうかを設定します。

HiRDB に接続している場合

スカラ関数 SUBSTR 実行要求の情報を取得するかどうかを設定します。

この情報は、クライアントが DABroker for Java で、次の条件すべてに一致する場合に有効になります。

- 接続しているデータベースが HiRDB で、HiRDB のサーバとクライアントのバージョンが共に 07-01 以上
- DABroker for Java のバージョンが 02-12 以上
- DABroker が暗黙的に下記 SQL を実行した場合
SET 文を実行し、代入値にスカラ関数 SUBSTR を使用します。

(35) LOBWRIT (DABSQL_TRC_LOBWRITE)

標準値

ON (チェックあり)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)
(ON: 取得します OFF: 取得しません)

説明

データベースアクセストレースに、LOB データの書き込み要求の情報を取得するかどうかを設定します。
この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。
この設定項目は、DABroker for Java から Oracle8i の機能を使用する場合だけ有効です。

(36) PARAMGET (DABSQL_TRC_PARAMGET)

標準値

ON (チェックあり)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)
(ON: 取得します OFF: 取得しません)

説明

データベースアクセストレースに、パラメタ情報取得要求の情報を取得するかどうかを設定します。
この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。
この設定項目は、DABroker for Java から Oracle8i の機能を使用する場合だけ有効です。

(37) SESBEGIN (DABSQL_TRC_SESBEGIN)

標準値

ON (チェックあり)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)
(ON: 取得します OFF: 取得しません)

説明

データベースアクセストレースに、セッション開始要求の情報を取得するかどうかを設定します。
この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。
この設定項目は、DABroker for Java から Oracle8i の機能を使用する場合だけ有効です。

(38) SESEND (DABSQL_TRC_SESEND)

標準値

ON (チェックあり)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)
(ON: 取得します OFF: 取得しません)

説明

データベースアクセストレースに、セッション終了要求の情報を取得するかどうかを設定します。
この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。
この設定項目は、DABroker for Java から Oracle8i の機能を使用する場合だけ有効です。

(39) SETPIECE (DABSQL_TRC_SETPIECE)

標準値

ON (チェックあり)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)
(ON: 取得します OFF: 取得しません)

説明

データベースアクセストレースに、ピース情報設定要求の情報を取得するかどうかを設定します。
この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。
この設定項目は、DABroker for Java から Oracle8i の機能を使用する場合だけ有効です。

(40) GETDIAG (DABSQL_TRC_GETDIAG)

標準値

ON (チェックあり)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)
(ON: 取得します OFF: 取得しません)

説明

データベースアクセストレースに、エラー情報取得要求の情報を取得するかどうかを設定します。
この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

(41) AUTO COMMIT OFF (DABSQL_TRC_AUTO_COMMIT_OFF)

標準値

ON (チェックあり)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)

(ON：取得します OFF：取得しません)

説明

データベースアクセストレースに、AUTO COMMIT モード解除要求の情報を取得するかどうかを設定します。

この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

(42) AUTO COMMIT ON (DABSQL_TRC_AUTO_COMMIT_ON)

標準値

ON (チェックあり)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)

(ON：取得します OFF：取得しません)

説明

データベースアクセストレースに、AUTO COMMIT モード設定要求の情報を取得するかどうかを設定します。

この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

(43) GET INFO (DABSQL_TRC_GET_INFO)

標準値

ON (チェックあり)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)

(ON：取得します OFF：取得しません)

説明

データベースアクセストレースに、接続後の動作情報取得要求の情報を取得するかどうかを設定します。

この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

(44) SET INFO (DABSQL_TRC_SET_INFO)

標準値

ON (チェックあり)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)

(ON：取得します OFF：取得しません)

説明

データベースアクセストレースに、接続後の動作情報設定要求の情報を取得するかどうかを設定します。

この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

(45) EXECUTE DIRECT (DABSQL_TRC_EXECUTE_DIRECT)

標準値

ON (チェックあり)

指定範囲

ON (チェックあり) 又は OFF (チェックなし)

(ON : 取得します OFF : 取得しません)

説明

データベースアクセストレースに、SQL の前処理後に実行する要求の情報を取得するかどうかを設定します。

この設定項目は、拡張データベースアクセストレース機能使用時だけ有効です。

6.5.2 拡張データベースアクセストレース機能

(1) 機能概要

拡張データベースアクセストレースでは、従来の DABroker のデータベースアクセストレースと比較して、次の機能が拡張されています。

- データベースへの接続ごと（接続から切断まで）にトレースを取得できます。
- データベース接続中でも、トレースファイルの参照やコピーができます。
- マルチスレッドでアプリケーションが動作している場合、データベースアクセスを実行したスレッドを特定できます。
- DABroker 動作環境定義ファイルの設定によって、出力するイベントの種類を選択できます。
- DABroker 動作環境定義ファイルの設定によって、ファイルの最大容量を設定できます。トレース情報が最大容量を超える場合、最大 2 世代で取得ファイルを切り替えてトレースを取得できます。

(2) 適用範囲

拡張データベースアクセストレース機能は、次のような場合に有効です。

- データベースのアクセス中に障害が発生した場合、迅速に障害を切り分ける必要がある場合
- 24 時間稼働などのために、稼働中のアプリケーションを終了しないで障害を調査する必要がある場合
- マルチスレッド環境でアプリケーションが動作していて、スレッドごとのデータベースアクセス履歴を取得したい場合

- コネクションプーリングなどを使用していて、データベースへの接続ごとのデータベースアクセス履歴を取得する必要がある場合
- 障害の再現調査ができないため、常時トレースを取得する必要がある場合

(3) DABroker 動作環境定義

拡張データベースアクセストレースを取得する設定は、DABroker 動作環境定義ファイルで設定します。

DABroker 動作環境定義ファイルでの設定方法については、「[4.5 DABroker 環境設定項目](#)」を参照してください。

(4) 出力形式

拡張データベースアクセストレースのファイル出力形式を次に示します。

Oracle19c で Oracle7 接続の場合、Oracle12c 以前とは出力するイベントが変わるなど、出力内容が異なります。

- ヘッダ情報

```

*-----*
*- DataBase Access Information (DRV)          -*
*- Date   YYYY/MM/DD hh:mm:ss.nnnnnn        -*
*-----*
DABroker Connect ID   : AAAAA(BBBBB)
Process ID            : CCCCC
UserID                : DDDDD
Client Name           : EEEEE
(DB別出力情報)
Lang Mode             : GGGGG

```

- 個々のトレース情報

```

THREAD-ID  CID  EVT  START-TIME  END-TIME  RETCODE
BLOCKCNT  HiRDB_CONNECTION_ID (SQL)SS...SS

```

DRV

DABroker のデータベースアクセスドライバの種別に従い、次の識別子が出力されます。

HiRDB 接続の場合

HiRDB Driver

Oracle8i 接続の場合

ORACLE8i Driver^{*} (Oracle Call Interface 8i を使用)

Oracle7 接続の場合

ORACLE7 Driver (Oracle19c 以外で Oracle Call Interface 7 を使用している場合、又は Oracle19c で Oracle Call Interface 8i を使用している場合)

SQL Anywhere 接続の場合

SQLAnywhere

ODBC3.0 接続の場合

ODBC30

RDA Link for Gateway 経由メインフレーム系データベース接続の場合

RDALink Driver

Database Connection Server 経由メインフレーム系データベース接続の場合

DBS

注※

DABroker for Java から Oracle8i の機能を使用する場合だけ出力されます。

Date

ファイル作成日時 (YYYY/MM/DD hh:mm:ss.nnnnnn) が出力されます。

- YYYY : 西暦年, MM : 月, DD : 日
- hh : 時, mm : 分, ss : 秒, nnnnnn : マイクロ秒

DABroker Connect ID

データベース接続識別子が出力されます。

Process ID

実行プロセス ID が出力されます。

UserID

接続ユーザ ID が出力されます。

DABroker for Java で ORACLE の OS 認証機能を使用した場合は、「OS authority」となります。

Client Name

UAP 名称が出力されます。名称が設定されていない場合は空白になります。

(DB 別出力情報)

接続データベースによって異なる情報が出力されます。出力内容について、表 6-3 に示します。

表 6-3 接続データベースごとのトレース出力内容

接続データベース	出力内容	説明
HiRDB	PDHOST :	HiRDB 接続時の PDHOST の設定内容です。未設定の場合は, "Not Indicator"が出力されます。
	PDNAMEPORT :	HiRDB 接続時の PDNAMEPORT の設定内容です。未設定の場合は, "Not Indicator"が出力されます。
Oracle	SQLNET :	Oracle リスナー名が出力されます。未設定の場合は, "Not Indicator"が出力されます。
SQL Anywhere	ODBC Data Source :	ODBC 設定のデータソース名が出力されます。未設定の場合は, "Not Indicator"が出力されます。

接続データベース	出力内容	説明
	ODBC Key Word :	ODBC 設定の接続情報が出力されます。未設定の場合は, "Not Indicator"が出力されます。
ODBC3.0	ODBC Data Source :	ODBC 設定のデータソース名が出力されます。未設定の場合は, "Not Indicator"が出力されます。
	ODBC Key Word :	ODBC 設定の接続情報が出力されます。未設定の場合は, "Not Indicator"が出力されます。
RDA Link	Access DataBase :	RDA Link で接続しているホスト DB 名称が出力されます。 VOS3 XDM RD E2 接続の場合 : VOS3 XDM/RD E2 VOS1 RDB1 E2 接続の場合 : VOS1 RDB1 E2 VOSK SQL/K 接続の場合 : VOSK SQL/K 未設定の場合 : Unknown DBMS
	Node Name :	データベース識別子の情報が出力されます。未設定の場合は, "Not Indicator"が出力されます。
Database Connction Server	Access DataBase :	接続しているホスト DB 名称が出力されます。 VOS3 XDM RD E2 接続の場合 : VOS3 XDM/RD E2 VOS3 XDM SD E2 接続の場合 : VOS3 XDM/SD E2 VOS1 RDB1 E2 接続の場合 : VOS1 RDB1 E2 VOS1 PDMII E2 接続の場合 : VOS1 PDM2 E2 VOSK SQL/K 接続の場合 : VOSK SQL/K VOSK スプールファイル接続の場合 : VOSK SPOOL 未設定の場合 : Unknown DBMS
	Node Name :	データベース識別子の情報が出力されます。未設定の場合は, "Not Indicator"が出力されます。

Lang Mode

実行言語モードが出力されます。

THREAD-ID

実行スレッド ID が出力されます。

CID

カーソル番号が出力されます。

EVT

DABroker がデータベースに要求した処理の内容です。

出力するイベントの種類は任意に選択できます。

選択できるイベントについては、[「4.5.1\(9\) データベースアクセストレースの取得 \(DABSQL_TRCSIZE\)」](#)を参照してください。ただし、データベースアクセストレースを取得してい

ない場合は、DABroker 動作環境定義ファイルを直接編集します。また、拡張データベースアクセストレースで出力イベントの種類を変更すると、従来のデータベースアクセストレースの出力イベントの種類も変更されます。

すべてのイベント出力を OFF にした場合はヘッダだけが出力されます。

次のイベントは、既存のデータベースアクセストレースにはありません。また、イベント出力を一つでも ON にした場合、GET_DIAG イベントの出力を抑止できません。

ATTACH：サーバとのアクセスパス作成要求を実行します。

ATTRGET：ハンドル属性の取得要求を実行します。

ATTRSET：ハンドル属性の設定要求を実行します。

AUTOCOMOFF：AUTO COMMIT モードの解除要求を実行します。

AUTOCOMON：AUTO COMMIT モードの設定要求を実行します。

BIND：パラメタのバインド要求を実行します。

CHARFORM：キャラクタセットフォームの取得要求を実行します。

CHARID：キャラクタセット ID の取得要求を実行します。

DEFINE：パラメタの定義要求を実行します。

DESALLOC：記述子の領域確保を実行します。

DESC_OUT：前処理した SQL の出力情報を返します。

DESC_IN：前処理した SQL の入力情報を返します。

DESCANY：オブジェクトの記述要求を実行します。

DESFREE：記述子の領域解放を実行します。

DETACH：データベースアクセスパス削除要求を実行します。

ENVCREATE：環境作成要求を実行します。

ERRGET：エラー情報取得要求を実行します。

EXECDIRECT：SQL の前処理後の実行処理です。

GET_DIAG：エラー詳細情報の取得処理を実行します。

GETINFO：接続後の動作情報の取得要求を実行します。

GETPIECE：ピース情報の取得要求を実行します。

HDLALLOC：ハンドルの領域確保を実行します。

HDLFREE：ハンドルの領域解放を実行します。

LOBCLOSE：BFILE データのクローズ要求を実行します。

LOBLEN：LOB データの長さ取得要求を実行します。

LOBOPEN：BFILE データのオープン要求を実行します。

LOBREAD：LOB データの読み込み要求を実行します。

LOBWRITE：LOB データの書き込み要求を実行します。

PARAMGET：パラメタ情報の取得要求を実行します。

SESBEGIN：セッション開始要求を実行します。

SESEND：セッション終了要求を実行します。

SETINFO：接続後の動作情報の設定要求を実行します。

SETPIECE：ピース情報の設定要求を実行します。

START-TIME

アクセス開始日時 (YYYY/MM/DD hh:mm:ss.nnnnnn) が出力されます。

- YYYY：西暦年，MM：月，DD：日
- hh：時，mm：分，ss：秒，nnnnnn：マイクロ秒

END-TIME

アクセス終了日時 (YYYY/MM/DD hh:mm:ss.nnnnnn) が出力されます。

- YYYY：西暦年，MM：月，DD：日
- hh：時，mm：分，ss：秒，nnnnnn：マイクロ秒

RETCODE

データベースから返されるリターンコードが出力されます。

BLOCKCNT

FETCH 時の行取得件数が出力されます。

HiRDB_CONNECTION_ID

コネクション ID (aaaa：bbbb：cccc) が出力されます。

- aaaa：フロントエンドサーバ名 (HiRDB/Parallel Server 使用時)，又はシングルサーバ (HiRDB/Single Server 使用時)
- bbbb：aaaa で示すサーバの接続通番
- cccc：aaaa で示すサーバのプロセス ID，及びコネクション ID

プロセス ID とコネクション ID は，HiRDB クライアントのバージョンが 07-01 以降で，サーバが HiRDB の場合だけ出力されます。HiRDB クライアントのバージョンが 07-01 未満で，サーバが HiRDB の場合は，ヘッダ (HiRDB_CONNECTION_ID) だけが出力され，コネクション ID は出力されません。また，サーバが XDM/RD の場合は，ヘッダ (HiRDB_CONNECTION_ID)，及びコネクション ID は出力されません。なお，この情報は，CONNECT，COMMIT，ROLLBACK，PREPARE，EXECUTE，又は DIRECT を実行したときに，最新の情報に更新されます。

(SQL)

クライアントから要求のあった SQL の内容が出力されます。この項目は，必要に応じて出力されます。

(Windows QueryPerformance Counter)

高分解パフォーマンス情報が表示されます。

(5) ファイル名称

拡張データベースアクセストレースのファイルは，次のディレクトリに作成されます。

DABroker運用ディレクトリ¥spool¥db_access

拡張データベースアクセストレースのファイル名称は、次の規則に従います。

spdb_XXXXXXXX_YYYYYYY_****ZZZZ.IIIII

****：接続データベース名称

HiRDB の場合

HiRDB

Oracle8i 接続の場合

ORACLE8i※

Oracle7 接続の場合

ORACLE7

SQL Anywhere の場合

SQLAnywher

ODBC3.0 の場合

ODBC30

RDA Link for Gateway 経由メインフレーム系データベースの場合

RDALink

Database Connection Server 経由メインフレーム系データベースの場合

DBS

注※

DABroker for Java から Oracle8i の機能を使用する場合だけ出力されます。

XXXXXXXX：実行プロセス ID

YYYYYYY：データベース接続識別子

ZZZZZ：識別順序番号

同じプロセス ID で、かつ同じデータベース接続識別子のファイルがある場合に、00000 から 32767 までの番号を付加します。

IIIII：ファイル拡張子

log：最新のトレースファイル

logold：1 世代前のトレースファイル

(6) ファイルの運用方法

- 拡張データベースアクセストレースは、DABroker 動作環境定義ファイルで設定した行数を超える出力があった場合、現在のトレースファイルをバックアップファイルとして別名で保存します(バックアップファイルの名称については、「6.5.2(5) ファイル名称」参照)。トレース内容は、トレースファイルの先頭から再び出力します。

- バックアップファイルは 1 世代前までしか残しません。このため、既にバックアップファイルがあり、現在のトレースファイルが設定した行数を超えると、バックアップファイルを上書きします。バックアップファイルを保存する必要がある場合は、あらかじめ別のディレクトリに退避してください。
- トレースファイルは実行しているコネクションの数だけ作成されます。
- トレースファイルは自動的に削除されません。ユーザが不要になった時点で削除してください。
- ハードディスクの空き容量が十分でない場合、トレースの出力が停止されることがあります。

6.6 エラートレース機能

6.6.1 機能概要

エラートレース機能は、次の場合に DABroker のエラー情報をログファイルに出力します。

- リモートアクセス時にエラーが発生した場合
- DABroker Driver for Java Technology を使用したアクセス時にエラーが発生した場合

エラートレース機能で出力する情報を次に示します。

- エラーメッセージ
- データベース接続情報
- エラーが発生する前に要求を受けた情報
- 前回コミット，又はロールバックを実行した時間
- SQL 文の実行要求処理中の場合は，実行中のカーソルでの要求ごとの履歴情報

6.6.2 適用範囲

エラートレース機能は，エラー発生時の原因調査をするような場合に有効です。

6.6.3 出力形式

DABroker エラーログのファイル出力形式を次に示します。

履歴情報は，エラー発生時に履歴が存在する情報だけ出力されます。

(1) エラー事象

- 出力形式

```
YYYY/MM/DD hh:mm:ss.ssssss TID=AAAA SS..SS
```

YYYY/MM/DD hh:mm:ss.ssssss

エラー発生日時が出力されます。

- YYYY：西暦年，MM：月，DD：日
- hh：時，mm：分，ss：秒，ssssss：マイクロ秒

AAAA

エラー発生スレッド ID (16 進数値) が出力されます。

SS...SS

エラーメッセージが出力されます。

(2) データベース接続情報

• 出力形式

```
YYYY/MM/DD hh:mm:ss.ssssss PID=BBBB InitID=CCCC ConID=DDDD CName=EEEE DBID=FFFF CID=GGGG(HHHH,IIII)
```

YYYY/MM/DD hh:mm:ss.ssssss

データベース接続日時が出力されます。ただし、データベースと接続していない場合は、"----/--/--
--:--:--.-----"が出力されます。

- YYYY：西暦年，MM：月，DD：日
- hh：時，mm：分，ss：秒，ssssss マイクロ秒

BBBB

プロセス ID (10 進数値) が出力されます。

CCCC

DABroker 接続 ID (16 進数値) が出力されます。

DDDD

データベース接続識別子 (16 進数値) が出力されます。

EEEE

UAP 名称が出力されます。ただし、名称なしの場合は空白となります。

FFFF

データベース ID (10 進数値) が出力されます。

出力されるデータベース ID を次に示します。

1：HiRDB

2：Oracle

4：VOS3 XDM/RD

5：VOS1 RDB1 E2

6：VOSK SQL/K

7：Oracle8i (DABroker for Java からの Oracle8i 接続の場合)

68：VOS3 XDM/RD*

69：VOS1 RDB1 E2*

70：VOSK SQL/K*

74：VOS3 XDM/SD*

75 : VOS1 PDM II E2*

76 : VOSK スプールファイル*

注*

Database Connection Server 経由

GGGG

実行中カーソル ID (10 進数値) が出力されます。

HHHH,IIII

保守用情報が出力されます。

(3) 履歴情報

- 出力形式

START-TIME	EXEC-TIME	TID	COUNT	REQ-CODE	DATA
YYYY/MM/DD hh:mm:ss.ssssss	mmm:ss.ssssss	JJJJJJJJ	KKKKKKKK	LLLL MMMM NNNNNNNN	00000000 PP..PP

履歴情報は、履歴種別によって出力される情報の意味が変わります。履歴種別を次に示します。

履歴種別 (MMMM)

大きくわけて次の 3 種類に分類されます。

- 共通履歴情報
 - NOW : 実行中の情報
 - OLD : 一つ前の要求の情報
- SQL 文の実行要求時に指定されたカーソルの履歴情報
 - PRE : SQL PREPARE 処理
 - EXE : SQL EXECUTE 処理
 - FET1 : SQL FETCH 処理初回
 - FET : SQL FETCH 処理
 - CLS : SQL CLOSE 処理
- コミット/ロールバック情報
 - CMT : コミット処理
 - RBK : ロールバック処理

共通情報

START-TIME : 要求開始日時

EXEC-TIME : 要求実行時間

FET1 の場合は、最初の 1 件を検索した時間

FET の場合は、FETCH 処理の累計実行時間

TID：要求実行スレッド ID (16 進数値)

CMT, RBK の場合は, 空欄

COUNT：コネクト要求からの累計要求カウント数 (16 進数値)

FET の場合は, SQL 文実行時の FET の要求回数

0xFFFFFFFF を超えた場合は, 0 から循環します。

REQ-CODE：

LLLL：要求コード (10 進数値)

「7.5.2 要求コード/応答コード」を参照してください。

MMMM：履歴種別

DATA：

NNNNNNNN：情報 1 (16 進数値)

OOOOOOOO：情報 2 (16 進数値)

PP..PP：詳細情報 (任意文字列)

詳細を表 6-4 に示します。

表 6-4 各履歴種別による DATA 内容

履歴種別(MMMM)	情報 1 (NNNNNNNN)	情報 2 (OOOOOOOO)	詳細情報(P..PP)
PRE	検索カラム数	?パラメタ定義数	SQL の先頭 16 バイト (マルチバイト文字コードの途中の場合はその文字は出力しません)
EXE	SQL 実行件数*	?パラメタ指定数	非表示
FET1	要求時検索件数	最大検索件数	非表示
FET	累計検索件数	要求時検索件数	非表示
CLS	0 固定	0 固定	非表示
CMT	非表示		
RBK	非表示		
NOW	今回の要求コードで取得した情報		
OLD	前回の要求コードで取得した情報		

注※

INSERT, UPDATE, DELETE 実行時の件数

6.6.4 出力例

(1) コネクト時にエラーになった場合

```
2004/05/07 13:30:09.555000 TID=4040 KFP A11561-E Specified authorization identifier has no connect privilege
----/--/-- --:--:--.----- PID=1296 InitID=701 ConID=0 CName=DABLIBAP DBID=1 CID=0(0,0)
START-TIME EXEC-TIME TID COUNT REQ-CODE DATA
2004/05/07 13:30:09.195000 0:00.350000 4040 1 1002 NOW 00000000 00000000
```

(2) 最初の SQL 文の実行要求時にエラーになった場合

```
2004/05/07 13:30:25.478000 TID=1812 KFP A11204-E Table/index HS5."A" not found in system
2004/05/07 13:30:21.232000 PID=1752 InitID=701 ConID=10701 CName=DABLIBAP DBID=1 CID=1(0,0)
START-TIME EXEC-TIME TID COUNT REQ-CODE DATA
2004/05/07 13:30:25.438000 0:00.040000 1812 FFFFFFFF 1021 NOW 00000000 00000000
select * from a2004/05/07 13:30:21.232000 0:00.280000 1812 1 1002 OLD 00000000 00000000
```

(3) FETCH 要求が 2 回以上となる SQL 文を実行した後にエラーとなった場合

```
2004/05/07 13:31:04.324000 TID=1812 KFP A11204-E Table/index HS5."B" not found in system
2004/05/07 13:30:21.232000 PID=1752 InitID=701 ConID=10701 CName=DABLIBAP DBID=1 CID=1(0,0)
START-TIME EXEC-TIME TID COUNT REQ-CODE DATA
2004/05/07 13:31:04.304000 0:00.010000 1812 e 1021 NOW 00000000 00000000select * from b
2004/05/07 13:30:57.745000 0:00.040000 1812 d 1004 OLD 00000000 00000000
2004/05/07 13:30:41.892000 0:00.040000 1812 8 1021 PRE 00000009 00000000
SELECT * FROM "H
2004/05/07 13:30:41.942000 0:00.020000 1812 9 1022 EXE 00000000 00000000
2004/05/07 13:30:42.022000 0:00.140000 1812 a 1023 FET1 00006eb3 00000000
2004/05/07 13:30:50.905000 0:06.549000 1812 2 1023 FET 0000c350 0000549d
2004/05/07 13:30:57.745000 0:00.000000 1812 c 1024 CLS 00000000 00000000
2004/05/07 13:30:57.745000 0:00.040000 d 1004 CMT
2004/05/07 13:30:25.488000 0:00.040000 3 1005 RBK
```

(4) FETCH 要求が 1 回で終了する SQL 文を実行した後にエラーとなった場合

```
2004/05/07 13:31:25.645000 TID=1812 KFP A11204-E Table/index HS5."C" not found in system
2004/05/07 13:30:21.232000 PID=1752 InitID=701 ConID=10701 CName=DABLIBAP DBID=1 CID=1(0,0)
START-TIME EXEC-TIME TID COUNT REQ-CODE DATA
2004/05/07 13:31:25.635000 0:00.010000 1812 19 1021 NOW 00000000 00000000
select * from c
2004/05/07 13:31:19.015000 0:00.010000 1812 18 1004 OLD 00000000 00000000
2004/05/07 13:31:18.815000 0:00.020000 1812 14 1021 PRE 00000009 00000000
SELECT * FROM "H
2004/05/07 13:31:18.835000 0:00.020000 1812 15 1022 EXE 00000000 00000000
2004/05/07 13:31:18.895000 0:00.050000 1812 16 1023 FET1 0000014d 00000000
2004/05/07 13:31:18.875000 0:00.070000 1812 1 1023 FET 0000014d 0000014d
2004/05/07 13:31:19.015000 0:00.000000 1812 17 1024 CLS 00000000 00000000
2004/05/07 13:31:19.015000 0:00.010000 18 1004 CMT
2004/05/07 13:31:04.324000 0:00.010000 f 1005 RBK
```

(5) INSERT 文実行時?パラメタの数がアンマッチとなった場合 (EXECUTE 複数回実行後)

```
2004/05/07 15:37:16.773000 TID=1712 KFPA11313-E Number of variables not equal to number of parameters
2004/05/07 15:35:48.966000 PID=584 InitID=701 ConID=10701 CName=Select_load DBID=1 CID=1(1,0)
START-TIME          000 EXEC-TIME          TID      COUNT    REQ-CODE  DATA
2004/05/07 15:37:16.733000    0:00.030000    1712          7    1022 NOW  00000000 00000001
2004/05/07 15:36:18.369000    0:00.000000    1712          6    1022 OLD  00000001 00000002
2004/05/07 15:35:59.191000    0:00.281000    1712          2    1021 PRE  00000000 00000002
INSERT INTO "HS"
2004/05/07 15:36:18.369000    0:00.000000    1712          6    1022 EXE  00000001 00000002
```

(6) CLOSE 要求を 2 回連続して発行した場合

```
2004/05/07 16:00:39.720000 TID=1040 KFDB010TD-E 要求コードのシーケンスエラーが発生しました。[RequestCode=1024-1024]
2004/05/07 15:58:25.727000 PID=4164 InitID=701 ConID=10701 CName=Select_load DBID=1 CID=1(4,0)
START-TIME          EXEC-TIME          TID      COUNT    REQ-CODE  DATA
2004/05/07 16:00:39.700000    0:00.020000    1040          6    1024 NOW  00000000 00000000
2004/05/07 15:58:30.755000    0:00.000000    1040          5    1024 OLD  00000000 00000000
2004/05/07 15:58:26.749000    0:00.000000    1040          2    1021 PRE  00000009 00000000
SELECT * FROM "H2004/05/07 15:58:26.749000    0:00.000000    1040          3    1022 EXE  00000000 00000000
2004/05/07 15:58:26.759000    0:00.030000    1040          4    1023 FET1 0000014d 00000000
2004/05/07 15:58:26.759000    0:00.030000    1040          1    1023 FET  0000014d 0000014d
2004/05/07 15:58:30.755000    0:00.000000    1040          5    1024 CLS  00000000 00000000
```

6.6.5 ファイル名称

エラートレースファイルは、次のディレクトリに作成されます。

DABroker 運用ディレクトリ/spool

DABroker エラーログのファイル名称は、次のようになります。

- dabevtrcerr1 : ログ出力用ファイル
- dabevtrcerr2 : バックアップ用ファイル
- dabevtrcerr1.lock : ワークファイル

6.6.6 ファイルの運用方法

- dabevtrcerr1 ファイルのサイズが 512 キロバイトを超えている場合にログを出力すると、dabevtrcerr1 ファイルをバックアップ用ファイル (dabevtrcerr2) に名称を変更した後、新規作成した dabevtrcerr1 ファイルにログを出力します。
- バックアップ用ファイルは 1 世代前までだけ残せます。バックアップファイルを保存する必要がある場合は、あらかじめ別のディスクに退避してください。

- ハードディスクの空き容量が十分でない場合、ログの出力を停止します。

7

メッセージ (共通)

この章では、DABroker が出力するメッセージの要因、システムでの処理、及びユーザが取る対処を説明します。

7.1 メッセージの出力先

DABroker が発行するメッセージは、次のどれかに出力されます。

UNIX の場合

- サーバの標準エラー出力
- クライアント
- コンソール (/dev/console)
- Linux の場合のシステムログ (/var/log)
- HP-UX の場合のシステムログ (/var/adm/syslog/syslog.log)
- Solaris の場合のシステムログ (/var/adm/messages)
- AIX の場合のシステムログ (/var/adm/ras)
- メッセージログ (DABroker 運用ディレクトリ/spool/dabsv.log, 及び dabsv.logold)

Windows の場合

- クライアント
- メッセージダイアログボックス
- イベントログ (Windows のイベントビューア)
- メッセージログ (DABroker 運用ディレクトリ¥spool¥dabsv.log, 及び dabsv.logold)

このメッセージログは、イベントログビューア (プログラムフォルダに登録された[DABroker]から [イベントログビューア]を選択して起動すると表示されるダイアログボックス) で参照できます。

出力されたメッセージの要因、システムでの処理、及びユーザが取る対処を知る場合は、「7.3 メッセージ一覧」で該当するメッセージを探してください。「7.3 メッセージ一覧」では、DABroker が発行するメッセージを、メッセージ ID によって昇順に記載しています。

7.2 メッセージの形式及び注意事項

7.2.1 メッセージの形式

このマニュアルでは、メッセージを次の形式で説明します。

「KFDBnnnnn-m:メッセージテキスト」

KFDBnnnnn-m

メッセージ ID

KFDB

DABroker のメッセージであることを示します。

nnnnn

メッセージの通し番号を示します。

m

メッセージの種別を示します。

E:エラー

DABroker が正常に動作できないような障害の発生を通知するメッセージです。

I:インフォメーション

DABroker の動作を報告するメッセージです。

W:ワーニング

DABroker の処理で、ユーザに警告が必要な事象が発生したことを通知するメッセージです。

Q:クエスチョン

ユーザに応答を求めるメッセージです。

メッセージテキスト

メッセージの内容です。出力先に出力されるのは、メッセージ ID と、このメッセージテキストです。メッセージテキストは、上段に日本語、下段に英語を記載しています。

要因:メッセージが出力された原因を示します。

処理:メッセージが出力されたときにシステムの処理を示します。

対処:メッセージが出力されたときにユーザが取るべき処置を示します。

7.2.2 注意事項

- コンソールやメッセージログファイルには、サーバ又はクライアントの標準エラー出力に出力されないメッセージも出力されます。ただし、コンソールやメッセージログファイルには出力されずに、サーバ又はクライアントの標準エラー出力にだけ出力されるメッセージもあります。

- 「保守員に連絡してください」とは、システム管理者が、ご購入契約に基づくお問い合わせ窓口に連絡することを示します。保守員に連絡する場合、「6. DABroker の障害対策（共通）」に示した手順で、障害情報を取得してください。
- 「保守用情報」、「エラー番号」及び「シグナル番号」は、保守員が使用するコードです。
- データベース側で発行されたメッセージについては、各 DBMS のマニュアルを参照してください。
- 可変情報に表示する情報がない場合「*」を表示することがあります。
- ファイル名など、可変情報に表示する情報が長い場合、先頭 79 バイトだけを表示し、残りは切り捨てます。
- エージェント機能を使用する場合、メッセージに出力されるイベント名については「6.3.2(2) エージェントログの取得内容」を、パラメタについては、「5.4.5(1) エージェント機能のパラメター一覧」を参照してください。

7.2.3 Windows を使用している場合の注意事項

(1) 環境変数

出力されたメッセージの内容に「DABroker 動作環境定義ファイルの環境変数」といった記述がある場合は、Windows 環境では、環境設定ユーティリティで設定する項目を指します。環境変数と環境設定ユーティリティの対応を表 7-1 に示します。

環境設定ユーティリティ及び項目の詳細については、「4.4 DABroker 環境設定」及び「4.5 DABroker 環境設定項目」を参照してください。

表 7-1 環境変数と環境設定ユーティリティの項目の対応表

環境変数名	環境設定ユーティリティで対応する項目
DABTMP	作業ディレクトリ
DABSVLOGSIZE	ログファイルサイズ [dabsv.log]
DABDBA_TIMEOUT	データベースアクセス処理終了待ちタイムアウト
DABSELECTCOLCNT	表検索カラム数
DABSELECTINFCNT	更新・検索条件カラム数
DABORACLE_TYPE	使用する ORACLE のバージョン
DABSQL_HiRDB_DICTIONARY_DATATYPE	HiRDB のディクショナリ表のデータ属性
DAB_HIRDB_DBINF_ESC	HiRDB 環境変数区切り文字
DABSQL_TRCSIZE	アクセストレースを取得する
DABSQL_TRC_CANCEL	CANCEL (アクセストレースの種類)
DABSQL_TRC_CLOSE	CLOSE (アクセストレースの種類)

環境変数名	環境設定ユーティリティで対応する項目
DABSQL_TRC_COMMIT	COMMIT (アクセストレースの種類)
DABSQL_TRC_CONNECT	CONNECT (アクセストレースの種類)
DABSQL_TRC_DECLARE	DECLARE (アクセストレースの種類)
DABSQL_TRC_DESCRIBE	DESCRIBE (アクセストレースの種類)
DABSQL_TRC_DISCONNECT	DISCONNECT (アクセストレースの種類)
DABSQL_TRC_EXECUTE	EXECUTE (アクセストレースの種類)
DABSQL_TRC_FETCH	FETCH (アクセストレースの種類)
DABSQL_TRC_LOGGING	LOGGING (アクセストレースの種類)
DABSQL_TRC_LOGLESS	LOGLESS (アクセストレースの種類)
DABSQL_TRC_PREPARE	PREPARE (アクセストレースの種類)
DABSQL_TRC_ROLLBACK	ROLLBACK (アクセストレースの種類)
DABSQL_TRC_OPEN	OPEN (アクセストレースの種類)
DABSQL_TRC_ATTACH	ATTACH (アクセストレースの種類)
DABSQL_TRC_ATTRGET	ATTRGET (アクセストレースの種類)
DABSQL_TRC_ATTRSET	ATTRSET (アクセストレースの種類)
DABSQL_TRC_BIND	BIND (アクセストレースの種類)
DABSQL_TRC_CHARFORM	CHARFORM (アクセストレースの種類)
DABSQL_TRC_CHARID	CHARID (アクセストレースの種類)
DABSQL_TRC_DEFINE	DEFINE (アクセストレースの種類)
DABSQL_TRC_DESALLOC	DESALLOC (アクセストレースの種類)
DABSQL_TRC_DESCANY	DESCANY (アクセストレースの種類)
DABSQL_TRC_DESFREE	DESFREE (アクセストレースの種類)
DABSQL_TRC_DETACH	DETACH (アクセストレースの種類)
DABSQL_TRC_ENVCREATE	ENVCREATE (アクセストレースの種類)
DABSQL_TRC_ERRGET	ERRGET (アクセストレースの種類)
DABSQL_TRC_GETPIECE	GETPIECE (アクセストレースの種類)
DABSQL_TRC_HDLALLOC	HDLALLOC (アクセストレースの種類)
DABSQL_TRC_HDLFREE	HDLFREE (アクセストレースの種類)
DABSQL_TRC_LOBCLOSE	LOBCLOSE (アクセストレースの種類)
DABSQL_TRC_LOBLEN	LOBLEN (アクセストレースの種類)
DABSQL_TRC_LOBOPEN	LOBOPEN (アクセストレースの種類)

環境変数名	環境設定ユーティリティで対応する項目
DABSQL_TRC_LOBREAD	LOBREAD (アクセストレースの種類)
DABSQL_TRC_LOBWRITE	LOBWRITE (アクセストレースの種類)
DABSQL_TRC_PARAMGET	PARAMGET (アクセストレースの種類)
DABSQL_TRC_SESBEGIN	SESBEGIN (アクセストレースの種類)
DABSQL_TRC_SESEND	SESEND (アクセストレースの種類)
DABSQL_TRC_SETPIECE	SETPIECE (アクセストレースの種類)
DABSQL_TRC_GETDIAG	GETDIAG (アクセストレースの種類)
DABSQL_TRC_AUTO_COMMIT_OFF	AUTO_COMMIT_OFF (アクセストレースの種類)
DABSQL_TRC_AUTO_COMMIT_ON	AUTO_COMMIT_ON (アクセストレースの種類)
DABSQL_TRC_GET_INFO	GET_INFO (アクセストレースの種類)
DABSQL_TRC_SET_INFO	SET_INFO (アクセストレースの種類)
DABSQL_TRC_EXECUTE_DIRECT	EXECUTE_DIRECT (アクセストレースの種類)
DABBATCH	予約処理作業ディレクトリ
DABCLIENT_TIMEOUT	クライアントからの要求待ちタイムアウト
DABMAXUSERS	最大接続ユーザ数
DABDBANUMBER	同時実行可能なデータベースアクセス数
DABMAXBLOCKUSERS	1 プロセスあたりの実行ユーザ数
DABMAXPROCESS	最大処理プロセス数
DABHIRDBA_BLOBBUFSIZE	BLOB 型データ受取バッファサイズ(HiRDB)
DABORADBA_LONGBUFSIZE	LONG 型データ受取バッファサイズ(ORACLE)
DABODBC30_LONGBUFSIZE	LONG 型データ受取バッファサイズ(ODBC3.0)
DABDBSDBA_BLOBBUFSIZE	BLOB 型データ受取バッファサイズ(Database Connection Server)
DABLANGMODE	漢字コードセット
DABKEEPALIVE	TCP/IP 通信時キープアライブを監視する
DABIPCONVERT	IP アドレスからホスト名へ変換する
DABTCP_TRCSIZE	通信トレースを取得する
DBPAGT_USE	エージェント機能を使用
DABAGT	作業ディレクトリ(エージェント設定)
DABAGTREGMAX	登録可能エージェント数
DABAGTTIMMAX	実行予約可能最大エージェント数

環境変数名	環境設定ユーティリティで対応する項目
DABAGT_LOG_SIZE	ログファイル容量(エージェントログを取得する)
DABAGTFILMAX	実行結果格納ファイル容量
DABAGT_LOG_INFO	取得メッセージ種別(インフォメーション)
DABAGT_LOG_ERROR	取得メッセージ種別(エラー)
DABAGTGRP	グループ管理用ディレクトリ
DABAGTWCHALDAY	実行予約日を経過しても有効にする日数
DABAGTWCHALTIME	実行予約日を経過しても有効にする秒数
DABAGTEXEDIRETIME	即時配信実行監視時間
DABCPP_EVTTRC_SIZE	トレース取得サイズ(DABroker for C++)
DABCPP_EVTTRC_LEVEL	トレース取得レベル(DABroker for C++)
DABCPP_EVTTRC_nn	トレース取得種別(DABroker for C++)
DABCOBOL_EVTTRC_SIZE	トレース取得サイズ(DABroker for COBOL)
DABCOBOL_EVTTRC_LEVEL	トレース取得レベル(DABroker for COBOL)
DABCOBOL_EVTTRC_nn	トレース取得種別(DABroker for COBOL)
DABORB_EVTTRC_SIZE	トレース取得サイズ(DABroker for ORB)
DABORB_EVTTRC_LEVEL	トレース取得レベル(DABroker for ORB)
DABORB_EVTTRC_nn	トレース取得種別(DABroker for ORB)
DAB_DBS_HOST	接続先ホスト名または IP アドレス(Database Connection Server 接続設定)
DAB_DBS_PORT	接続先サービス名またはポート番号(Database Connection Server 接続設定)
DAB_DBS_SEND_BUFFER_SIZE	送信バッファサイズ(Database Connection Server 接続設定)
DAB_DBS_RECEIVE_BUFFER_SIZE	受信バッファサイズ(Database Connection Server 接続設定)
DAB_XA_ERROR	XA トレースを取得する

(2) 接続先データベース定義

出力されたメッセージの内容に「接続先データベース定義ファイルの設定」といった記述がある場合は、接続先データベース定義ユーティリティで設定することを指します。

接続先データベース定義ユーティリティ、及び項目の詳細については、「[4.6 DABroker 接続先データベース定義](#)」を参照してください。

(3) コネクションマルチ定義

出力されたメッセージの内容に「コネクションマルチ定義ファイルの設定」といった記述がある場合は、コネクションマルチ定義ユーティリティで設定することを指します。

コネクションマルチ定義ユーティリティ、及び項目の詳細については、「[4.7 DABroker コネクションマルチ定義 \(Database Connection Server 経由の DB アクセス定義\)](#)」を参照してください。

7.3 メッセージ一覧

KFDB002AG-E

必要な引数が足りないため、エージェントは実行しませんでした。要因・・・[XXXX]
The agent was not executed because a required argument was not specified. Reason ---
[XXXX]

XXXX：要因

要因

必要な引数が足りません。

処理

エージェント処理を終了します。

対処

保守員に連絡してください。

KFDB003AG-E

処理に必要なメモリが不足しているため、エージェントの実行を中止しました。
Agent execution was canceled because memory was insufficient for processing.

要因

エージェント処理の実行中に、メモリ不足が発生しました。

処理

エージェント処理を終了します。

対処

実行中のほかの処理が終了してから、再実行してください。

KFDB005AG-E

エージェント実行ファイルの取得でエラーが発生したため、エージェントの実行を中止しました。
要因・・・[XXXX], エラーコード・・・[YYYY]
Agent execution was canceled because an error occurred while obtaining the agent execution
file.Reason --- [XXXX], error code --- [YYYY]

XXXX：要因

YYYY：エラーコード

要因

エージェント実行ファイルの取得に失敗しました。

処理

エージェント処理を終了します。

対処

再度、処理を実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB006AG-E

エージェント実行結果ファイル出力でエラーが発生したため、エージェントの実行を中止しました。
要求・・・[XXXX], エラーコード・・・[YYYY]
Agent execution was canceled because an error occurred while outputting the agent execution result file.Request --- [XXXX], error code --- [YYYY]

XXXX：要求

YYYY：エラーコード

要因

エージェント実行結果ファイル出力でエラーが発生しました。

処理

エージェント処理を終了します。

対処

再度、処理を実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB007AG-E

データベースアクセス処理要求でイニシャライズ処理を行っていません。エージェントの実行を中止しました。要求関数・・・[XXXX], エラーコード・・・[YYYY], エラーコード名称・・・[ZZZZ]
The database access was requested without initialization. Agent execution was canceled.
Request function --- [XXXX], error code --- [YYYY], error code name --- [ZZZZ]

XXXX：要求関数

YYYY：エラーコード

ZZZZ：エラーコード名称

要因

データベースアクセス処理要求でイニシャライズ処理を行っていません。

処理

エージェント処理を中止します。

対処

保守員に連絡してください。

KFDB008AG-E

DBMS へのアクセスでエラーが発生しましたが、メモリ不足のためエラーメッセージが取得できません。エージェントの実行を中止しました。エラーコード・・・[XXXX]

An error occurred during access to the DBMS but the error message cannot be obtained because memory is insufficient. Agent execution is canceled. Error code --- [XXXX]

XXXX：エラーコード

要因

DBMS へのアクセスでエラーコードに示すエラーが発生しましたが、メモリ不足のためエラーメッセージが取得できません。

処理

エージェント処理を中止します。

対処

実行中のほかの処理が終了してから、再実行してください。

KFDB010AG-E

データベースアクセス処理に必要なメモリが不足しているため、エージェントの実行を中止しました。要求関数・・・[XXXX]、エラーコード・・・[YYYY]、エラーコード名称・・・[ZZZZ]

Agent execution is canceled because memory is insufficient for the database access processing. Request function --- [XXXX], error code --- [YYYY], error code name --- [ZZZZ]

XXXX：要求関数

YYYY：エラーコード

ZZZZ：エラーコード名称

要因

データベースアクセス処理の実行中にメモリ不足が発生しました。

処理

エージェント処理を中止します。

対処

実行中のほかの処理が終了してから、再実行してください。

KFDB012AG-E

データベースアクセス処理要求でシステムエラーが発生しました。エージェントの実行を中止しました。要求関数・・・[XXXX], エラーコード・・・[YYYY], エラーコード名称・・・[ZZZZ]

A system error occurred in the database access processing request. Agent execution was canceled. Request function --- [XXXX], error code --- [YYYY], error code name --- [ZZZZ]

XXXX：要求関数

YYYY：エラーコード

ZZZZ：エラーコード名称

要因

データベースアクセス処理要求でシステムのエラーが発生しました。

処理

エージェント処理を中止します。

対処

クライアントからの要求を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB013AG-E

データベースアクセス処理要求での関数の引数が不正なため、エージェントの実行を中止しました。要求関数・・・[XXXX], エラーコード・・・[YYYY], エラーコード名称・・・[ZZZZ]

Agent execution was canceled because an argument of the function for the database access processing request is invalid.

Request function --- [XXXX], error code --- [YYYY], error code name --- [ZZZZ]

XXXX：要求関数

YYYY：エラーコード

ZZZZ：エラーコード名称

要因

データベースアクセス処理要求での関数の引数が不正なため、エージェントの実行を中止しました。

処理

エージェント処理を中止します。

対処

保守員に連絡してください。

KFDB014AG-E

予期しないエラーが発生したため、エージェントの実行を中止しました。
Agent execution was canceled because an unexpected error occurred.

要因

予期しないエラーが発生しました。

処理

エージェント処理を中止します。

対処

保守員に連絡してください。

KFDB015AG-W

エージェント実行結果ファイルの最大容量を超えるため、実行結果情報を切り捨てました。
The execution result is truncated because the agent execution result file exceeded the maximum size.

要因

実行結果情報がエージェント実行結果ファイルの最大容量を超えました。

処理

エージェント実行結果情報を切り捨てました。

対処

エージェント実行結果ファイルの最大容量を増やすか、実行結果情報を絞り込んで検索してください。

KFDB016AG-E

データ変換でアンダーフローエラーが発生したため、エージェントの実行を中止しました。
Agent execution was canceled because an underflow error occurred during data conversion.

要因

データ変換でアンダーフローエラーが発生しました。

処理

エージェント処理を中止します。

対処

エージェントの登録内容と、DBMS のテーブルの定義情報を確認してください。

KFDB017AG-E

データ変換でオーバフローエラーが発生したため、エージェントの実行を中止しました。

Agent execution was canceled because an overflow error occurred during data conversion.

要因

データ変換でオーバフローエラーが発生しました。

処理

エージェントの処理を中止します。

対処

エージェントの登録内容と、DBMS のテーブルの定義情報を確認してください。

KFDB018AG-E

データ変換でメモリ不足が発生したため、エージェントの実行を中止しました。

Agent execution was canceled because memory became insufficient during data conversion.

要因

データ変換に必要なメモリが不足しているため、データ変換処理ができません。

処理

エージェント処理を中止します。

対処

しばらく時間を空けてから、再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB019AG-E

データ変換で NULL を指定できないデータに NULL 値を入れようとしたため、エージェントの実行を中止しました。

Agent execution was canceled because an invalid 'NULL' was set in the data during data conversion.

要因

データ変換で NULL を指定できないデータに NULL 値を入れようとしたため。

処理

エージェント処理を中止します。

対処

エージェントの登録内容と、DBMS のテーブルの定義情報を確認してください。

KFDB020AG-E

データ変換でデータ不正エラーが発生したため、エージェントの実行を中止しました。

Agent execution was canceled because an invalid data error occurred during data conversion.

要因

データ変換でデータ不正エラーが発生しました。

処理

エージェント処理を中止します。

対処

エージェントの登録内容と、DBMS のテーブルの定義情報を確認してください。

KFDB021AG-E

データ変換でパラメタ不正エラーが発生したため、エージェントの実行を中止しました。
Agent execution was canceled because an invalid parameter error occurred during data conversion.

要因

データ変換でパラメタ不正エラーが発生しました。

処理

エージェント処理を中止します。

対処

保守員に連絡してください。

KFDB022AG-E

データ変換で予期しないエラーが発生したため、エージェントの実行を中止しました。
Agent execution was canceled because an unexpected error occurred during data conversion.

要因

データ変換で予期しないエラーが発生しました。

処理

エージェント処理を中止します。

対処

保守員に連絡してください。

KFDB023AG-E

サポートしていないデータ型が含まれています。 [XXXX]
A data type that DABroker does not support was found. [XXXX]

XXXX：列名

要因

エージェントでサポートしていないデータ型が含まれています。

処理

エージェント処理を中止します。

対処

エージェント登録内容を、表示された列を使用しないエージェントに変更し、再実行してください。

KFDB024AG-E

指定された DBMS に対する環境変数の指定が誤っています。 [XXXX]

The specified DBMS environment variable is incorrect. [XXXX]

XXXX：環境変数名

要因

接続しようとした DBMS と DABroker の環境設定とが一致しません。

処理

エージェント処理を中止します。

対処

環境設定を変更して、DABroker を起動し直した後に、エージェントを再実行してください。

KFDB001AL-I

AgentLog XXXX YYYY

XXXX：イベント名

YYYY：パラメタの並び

要因

エージェント関連のイベントが発生しました。

処理

エージェント処理を継続します。

KFDB002AL-E

AgentErr XXXX YYYY

XXXX：イベント名

YYYY：パラメタの並び

要因

エージェント関連のエラーイベントが発生しました。

処理

エージェント処理を継続します。

KFDB003AL-E

```
エージェントログの取得に失敗しました。[XXXX() errno=YYYY] [EVENT:ZZZZ]  
Agent log cannot be obtained. [XXXX() errno=YYYY] [EVENT:ZZZZ]
```

XXXX：システムコール名称

YYYY：エラー番号

ZZZZ：イベント名

要因

エージェントログの取得に失敗しました。主な原因として、次のことが考えられます。

- メモリ不足
- ディスク容量不足

処理

エージェント処理を継続します。

対処

ディスク容量不足が考えられる場合、DABAGT の割り当てられたパーティション内の不要なファイルを削除してください。このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB010AL-I

```
DABroker エージェントサービスを開始します。  
Starting the DABroker agent service.
```

要因

エージェント機能を使用する指定して DABroker を起動しました。

処理

クライアントからのエージェント操作要求の受け付けを開始します。

KFDB011AL-I

```
DABroker エージェントサービスを終了します。  
Terminating the DABroker agent service.
```


要因

DABroker の終了操作を実行しました。

処理

クライアントからのエージェント操作要求の受け付けを終了します。

KFDB012AL-E

エージェントサービスが開始されていないため、エージェント処理を受け付けられません。

Agent processing cannot be accepted because the agent service is not running.

要因

DABroker 開始時に、エージェントサービスが開始されていません。または、計画終了によってエージェントサービスが終了しました。

処理

エージェントの処理要求を受け付けません。

対処

エージェントサービスの起動を指定して、DABroker を再起動してください。

KFDB020AL-E

DABroker のロードモジュールに実行権限がありません。

The user does not have authorization to execute the DABroker load module.

要因

DABroker のロードモジュールに実行権限がないため、クライアントからの要求を処理できません。

処理

クライアントからの要求を受け付けません。

対処

DABroker を組み込み直してください。

KFDB022AL-E

DABroker のロードモジュールがありません。

There is no load module for DABroker.

要因

DABroker の環境が破壊されているため、クライアントからの要求を処理できません。

処理

クライアントからの要求を受け付けません。

対処

DABroker を組み込み直してください。

KFDB024AL-E

```
エージェント処理の実行中に main プロセスが異常終了しました。  
要因・・・XXXX [REGNAME:YYYY] (又は [EXENAME:ZZZZ])  
The main process terminated abnormally while the agent was processing.  
Reason --- XXXX [REGNAME:YYYY] (又は[EXENAME:ZZZZ])
```

XXXX：シグナル番号

YYYY：エージェント名

ZZZZ：エージェント実行名

要因

エージェント処理中にシグナルを受信し、処理を中断しました。

処理

エージェント処理を中断します。

対処

エージェント処理を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB025AL-E

```
エージェントサービスの開始処理中に main プロセスが異常終了しました。  
要因・・・catch signal NO=XXXX(又は EXIT=YYYY)  
The main process terminated abnormally while the agent service was starting.  
Reason --- catch signal NO=XXXX (又は EXIT=YYYY)
```

XXXX：受信したシグナルの番号

YYYY：終了コード(保守用情報)

要因

エージェントサービスの開始処理中に main プロセスが異常終了しました。

処理

DABroker を終了します。

対処

システム管理者に連絡してください。

KFDB030AL-E

エージェント処理の実行中にシステムコールでエラーが発生しました。
[XXXX() errno= YYYYY] [REGNAME: RRRR] (又は [EXENAME: EEEEE])
An error occurred in a system call while the agent was processing.
[XXXX() errno=YYYYY] [REGNAME:RRRR] (又は, [EXENAME:EEEE])

XXXX：システムコール名称

YYYY：エラー番号

RRRR：エージェント名

EEEE：エージェント実行名

要因

OS のシステムコール関数でエラーが発生しました。

処理

処理を終了します。

対処

操作を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB032AL-E

エージェント処理で使用した共有メモリを削除できませんでした。
[shmctl() errno=XXXX] [REGNAME:YYYYY] (又は [EXENAME:ZZZZ])
The shared memory used for agent processing cannot be deleted.
[shmctl() errno=XXXX] [REGNAME:YYYYY] (又は[EXENAME:ZZZZ])

XXXX：エラー番号

YYYY：エージェント名

ZZZZ：エージェント実行名

要因

エージェント処理の実行中に OS のシステムコール関数(shmctl)でエラーが発生しました。

処理

処理を続行します。

対処

このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB036AL-E

エージェント処理の実行中にメモリ不足が発生したため、処理を中止しました。
[REGNAME=XXXX] (又は [EXENAME=YYYY])
Agent processing was canceled because memory became insufficient.
[REGNAME=XXXX] (又は [EXENAME=YYYY])

XXXX：エージェント名

YYYY：エージェント実行名

要因

エージェント処理の実行中にメモリ不足が発生しました。

処理

エージェント処理を中断します。

対処

他の処理が終了してから、再実行してください。

KFDB040AL-E

登録できる最大エージェント数を超過しました。エージェントは登録できません。[XXXX]
The maximum number of registered agents was exceeded. No more agents can be registered.
[XXXX]

XXXX：登録できる最大数

要因

DABAGTREGMAX に指定した値を超えて、エージェントを登録しようとしてしました。

処理

エージェント登録処理を中断します。

対処

不要なエージェントを削除して、エージェント実行処理を再実行してください。または、DABAGTREGMAX の指定値を変更し、DABroker を再起動してからエージェント登録処理を再実行してください。

KFDB041AL-E

設定できる最大タイマ指定エージェント数を超過しました。タイマ設定できません。
[DABAGTTIMMAX=XXXX] [EXENAME=YYYY]
The maximum number of agents set in the timer was exceeded.No agent can be set in the timer.

[DABAGTTIMMAX=XXXX] [EXENAME=YYYY]

XXXX：登録できる最大数

YYYY：エージェント実行名

要因

DABAGTTIMMAX に指定した値を超えて、タイマ指定でエージェントを実行しようとした。

処理

エージェントの実行を中断します。

対処

不要なタイマ指定のエージェント実行を削除して、エージェント実行処理を再実行してください。または、DABAGTTIMMAX の指定値を変更し、DABroker を再起動してからエージェント登録処理を再実行してください。

KFDB046AL-E

エージェント名はすでに使用されています。[REGNAME:XXXX] [USERID:YYYY]

The agent name specified by the client is already used. [REGNAME:XXXX] [USERID:YYYY]

XXXX：エージェント名

YYYY：ユーザ ID

要因

登録しようとしたエージェントは、すでに登録済みです。

処理

エージェント登録処理を中断します。

対処

他のエージェント名に変更し、エージェント登録処理を再実行してください。

KFDB047AL-E

指定されたエージェント名が見つかりません。[REGNAME:XXXX]

The specified agent name was not found. [REGNAME:XXXX]

XXXX：エージェント名

要因

指定されたエージェント名が見つかりません。

処理

エージェント処理を中断します。

対処

エージェント名を修正し、エージェント処理を再実行してください。

KFDB050AL-E

エージェント処理用システムファイルの書き込みで LOCK エラーが発生したため、エージェント処理を受け付けられません。[FILENAME=XXXX] [errno=YYYY]

Agent processing cannot be accepted because a lock error occurred while writing the system file for agent processing. [FILENAME=XXXX] [errno=YYYY]

XXXX：ファイル名

YYYY：lockf()システムコールのエラー番号

要因

エージェント処理用システムファイルの書き込み時に LOCK エラーが発生しました。

処理

クライアントからの要求を受け付けません。

対処

エラー番号を調査して対策後、クライアントからの要求を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB051AL-E

エージェント処理用システムファイルの書き込みで UNLOCK エラーが発生したため、エージェント処理を受け付けられません。[FILENAME=XXXX] [errno=YYYY]

Agent processing cannot be accepted because an unlock error occurred while writing the system file used for agent processing. [FILENAME=XXXX] [errno=YYYY]

XXXX：ファイル名

YYYY：lockf()システムコールのエラー番号

要因

エージェント処理用システムファイルの書き込み時に UNLOCK エラーが発生しました。

処理

クライアントからの要求を受け付けません。

対処

エラー番号を調査して対策後、クライアントからの要求を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB052AL-E

エージェント処理用システムファイルの読み込みで LOCK エラーが発生したため、エージェント処理を受け付けられません。[FILENAME=XXXX] [errno=YYYY]

Agent processing cannot be accepted because a lock error occurred while reading the system file used for agent processing. [FILENAME=XXXX] [errno=YYYY]

XXXX：ファイル名

YYYY：lockf()システムコールのエラー番号

要因

エージェント処理用システムファイルの読み込み時に LOCK エラーが発生しました。

処理

クライアントからの要求を受け付けません。

対処

エラー番号を調査して対策後、クライアントからの要求を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB053AL-E

エージェント処理用システムファイルの読み込みで UNLOCK エラーが発生したため、エージェント処理を受け付けられません。[FILENAME=XXXX] [errno=YYYY]

Agent processing cannot be accepted because an unlock error occurred while reading the system file used for agent processing. [FILENAME=XXXX] [errno=YYYY]

XXXX：ファイル名

YYYY：lockf()システムコールのエラー番号

要因

エージェント処理用システムファイルの読み込み時に UNLOCK エラーが発生しました。

処理

クライアントからの要求を受け付けません。

対処

エラー番号を調査して対策後、クライアントからの要求を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB054AL-E

エージェント処理用システムファイルで書き込みエラーが発生したため、エージェント処理を受け付けられません。[FILENAME=XXXX] [errno=YYYY]

Agent processing cannot be accepted because a write error occurred in the system file used for agent processing. [FILENAME=XXXX] [errno=YYYY]

XXXX：ファイル名

YYYY：open()又はwrite()システムコールのエラー番号

要因

エージェント処理用システムファイルで書き込みエラーが発生しました。

処理

クライアントからの要求を受け付けません。

対処

エラー番号を調査して対策後、クライアントからの要求を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB055AL-E

エージェント処理用システムファイルで読み込みエラーが発生したため、エージェント処理を受け付けられません。[FILENAME=XXXX] [errno=YYYY]

Agent processing cannot be accepted because a read error occurred in the system file used for agent processing. [FILENAME=XXXX] [errno=YYYY]

XXXX：ファイル名

YYYY：open()又はread()システムコールのエラー番号

要因

エージェント処理用システムファイルで読み込みエラーが発生しました。

処理

クライアントからの要求を受け付けません。

対処

エラー番号を調査して対策後、クライアントからの要求を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB056AL-E

エージェント処理用システムファイルが作成できなかったため、エージェント処理を受け付けられません。[FILENAME=XXXX] [errno=YYYY]

Agent processing cannot be accepted because the system file used for agent processing could not be created. [FILENAME=XXXX] [errno=YYYY]

XXXX：ファイル名

YYYY : creat()システムコールのエラー番号

要因

DABroker で、エージェント処理に使用するシステムファイルが作成できませんでした。主な原因として、DABroker 動作環境定義ファイルの DABTMP、又は DABAGT に指定されているディレクトリが、次の状態であることが考えられます。

- 書き込み権限がない
- ディレクトリが存在しない
- ディレクトリの容量が不足している

処理

クライアントからの要求を受け付けません。

対処

エラー番号を調査して対策後、クライアントからの要求を再実行してください。

主な対策として DABroker 動作環境定義ファイルの DABTMP 又は DABAGT に指定されているディレクトリに、次のどれかの操作をしてください。

- 書き込み権限を付与する
- ディレクトリを作成する
- ディレクトリの容量を増やす

なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB057AL-E

エージェント処理用システムファイルのレコードの位置付けでエラーが発生したため、エージェント処理を受け付けられません。[FILENAME=XXXX] [errno=YYYY]

Agent processing cannot be accepted because a record positioning error occurred in the system file used for agent processing. [FILENAME=XXXX] [errno=YYYY]

XXXX : ファイル名

YYYY : lseek()システムコールのエラー番号

要因

エージェント処理用システムファイルのレコードの位置付けでエラーが発生しました。

処理

クライアントからの要求を受け付けません。

対処

エラー番号を調査して対策後、クライアントからの要求を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB058AL-E

エージェント処理用システムファイルが破壊されているため、エージェント処理を受け付けられません。[FILENAME=XXXX]

Agent processing cannot be accepted because the system file used for agent processing is damaged. [FILENAME=XXXX]

XXXX：ファイル名

要因

エージェント処理用システムファイルが破壊されています。

処理

クライアントからの要求を受け付けません。

対処

操作を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB060AL-E

エージェントの実行権限がありません。[USERID:XXXX] [REGNAME:YYYY]

Agent has no execution authorization. [USERID:XXXX] [REGNAME:YYYY]

XXXX：エージェント実行者のユーザ ID

YYYY：エージェント名

要因

この実行ユーザはエージェントの実行権限がありません。

処理

クライアントからの要求を受け付けません。

対処

ユーザ ID 及びパスワードを確認してください。

KFDB061AL-E

エージェントの参照権限がありません。

[USERID:XXXX] [REGNAME:YYYY] (又は [EXENAME:ZZZZ])

Agent has no reference authorization. [USERID:XXXX] [REGNAME:YYYY]

(又は[EXENAME:ZZZZ])

XXXX：エージェント利用者のユーザ ID

YYYY：エージェント名

ZZZZ：エージェント実行名

要因

このユーザはエージェントの参照権限がありません。

処理

クライアントからの要求を受け付けません。

対処

ユーザ ID 及びパスワードを確認してください。

KFDB062AL-E

エージェントのグループ管理情報ファイルにアクセスできないため、エージェント処理を受け付けられません。[FILENAME:XXXX] [errno=YYYY]

Agent processing cannot be accepted because the group information file of the agent cannot be accessed. [FILENAME=XXXX] [errno=YYYY]

XXXX：グループ管理情報ファイルの名称

YYYY：エラー番号

要因

エージェントのグループ管理情報ファイルにアクセスできないため、エージェント処理を受け付けられません。主な原因として、次のことが考えられます。

- DABAGTGRP で指定したディレクトリが存在しない
- グループ管理情報ファイルが存在しない
- グループ管理情報ファイルに読み取り権限がない
- エージェントに登録されたグループ ID が間違っている

処理

クライアントからの要求を受け付けません。

対処

次の対策でグループ管理情報ファイルにアクセスできるようにした後、エージェント処理を行ってください。

- DABAGTGRP で指定したディレクトリを作成する
- グループ管理情報ファイルを作成する
- グループ管理情報ファイルに読み取り権限を与える
- エージェントに登録されたグループ ID が間違っている場合は、いったんエージェントを削除した後に、再度エージェントを登録する

KFDB063AL-E

このエージェントはキャンセルできない状態です。

[EXESTAT:XXXX] [EXENAME:YYYY] [EXEMEMO:ZZZZ]

This agent cannot be canceled currently. [EXESTAT:XXXX] [EXENAME:YYYY]

[EXEMEMO:ZZZZ]

XXXX：エージェントの状態を表す値

ABNORMAL-ENDED：異常終了(キャンセル終了含む)

IDLE：初期状態(実行要求待ち)

NORMAL-ENDED：正常終了

YYYY：キャンセルしようとしたエージェント実行名

ZZZZ：キャンセルしようとしたエージェント実行のメモ名

要因

現在のエージェントがキャンセルを実行できない状態(ABNORMAL-ENDED:異常終了, IDLE:初期状態, 又は NORMAL-ENDED:正常終了)のため, キャンセルできません。

処理

キャンセル処理を中断します。

対処

エージェントがキャンセルを実行できる状態の時に, キャンセルを実行してください。

KFDB064AL-E

エージェントの実行をキャンセルする権限がありません。

[EXENAME:XXXX] [EXEMEMO:YYYY]

The user does not have authorization to cancel the execution of the agent.

[EXENAME:EEEE] [EXEMEMO:MMMM]

XXXX：キャンセルしようとしたエージェント実行名

YYYY：キャンセルしようとしたエージェント実行のメモ名

要因

エージェントを実行したユーザとキャンセルするユーザが異なるため, キャンセルできません。

処理

キャンセル処理を中断します。

対処

エージェントを実行したユーザでキャンセルを実行してください。

KFDB065AL-E

```
エージェントの実行は既に終了しています。  
[EXESTAT: YYYYY] [EXENAME: EEEEE] [EXEMEMO: MMMMM]  
The agent has already been terminated.  
[EXESTAT:YYYYY] [EXENAME:EEEE] [EXEMEMO:MMMMM]
```

YYYYY：エージェントの状態を表す値

RUN：実行中

TIMER-WATCH：タイマ監視中

EEEE：キャンセルしようとしたエージェント実行名

MMMMM：キャンセルしようとしたエージェント実行のメモ名

要因

エージェントの実行状態が YYYYY の時にキャンセルを実行しましたが、既にエージェントの実行が終了しているため、キャンセルされませんでした。

処理

キャンセル処理を終了します。

対処

エージェントの状態を確認してください。エージェントの状態がまだ実行中の時(RUN:実行中, TIMER-WATCH:タイマ監視中)は、再度キャンセルを実行してください。

KFDB066AL-E

```
エージェントを削除する権限、又は上書きする権限がありません。  
[USERID:UUUUU] [REGNAME:RRRRR]  
No authorization is granted to delete or overwrite the agent.  
[USERID:UUUUU] [REGNAME:RRRRR])
```

UUUUU：削除要求したユーザのユーザID

RRRRR：エージェント名

要因

この実行ユーザには、指定されたエージェントを削除、又は上書きする権限がありません。

処理

クライアントからの要求を受け付けません。

対処

ユーザID、及びパスワードを確認してください。

KFDB067AL-E

エージェントの実行を削除する権限がありません。

[USERID: UUUU] [EXENAME: EEEE] [EXEMEMO: MMMM]

The user does not have authorization to delete the execution of the agent.

[USERID:UUUU] [EXENAME:EEEE] [EXEMEMO:MMMM]

UUUU：削除要求したユーザのユーザ ID

EEEE：削除しようとしたエージェント実行名

MMMM：削除しようとしたエージェント実行のメモ名

要因

このユーザは指定されたエージェントの実行を削除する権限がありません。

処理

クライアントからの要求を受け付けません。

対処

ユーザ ID 及びパスワードを確認してください。

KFDB068AL-E

エージェントの実行状態が変化したため、エージェント実行の削除処理又はキャンセル処理を中止しました。[EXESTAT:XXXX] [EXENAME:YYYY] [EXEMEMO:ZZZZ]

Deletion or cancellation of the agent execution is canceled because the execution status of the agent has changed. [EXESTAT:XXXX] [EXENAME:YYYY] [EXEMEMO:ZZZZ]

XXXX：エージェントの状態を表す値

RUN：実行中

TIMER-WATCH：タイマ監視中

YYYY：削除又はキャンセルしようとしたエージェント実行名

ZZZZ：削除又はキャンセルしようとしたエージェント実行のメモ名

要因

エージェントの実行状態が XXXX からほかの状態に変化したため、エージェント実行の削除又はキャンセルができませんでした。

処理

削除処理、又はキャンセル処理を終了します。ただし、複数のエージェントを対象にしていた場合、このエージェント以前の削除処理又はキャンセル処理は実行しました。

対処

エージェント実行の削除又はキャンセルを再実行してください。

KFDB070AL-E

エージェントの実行予約日時が既に過ぎているため、エージェントを実行しませんでした。

[EXENAME:EEEE] [XXXX]

The agent was not executed because the reserved time for execution of the agent has already passed.

[EXENAME:EEEE] [XXXX]

EEEE : エージェント実行名

XXXX : 実行予約日時(不特定の場合は' * ' を表示します)

要因

指定されたエージェントの実行予約日時が既に過ぎているため、エージェントを実行しませんでした。

処理

このエージェントを実行しません。

対処

予約日時を確認して、必要に応じて再実行してください。

KFDB071AL-E

エージェント実行予約処理でエラーが発生したため、エージェント処理を受け付けられません。

[EXENAME:XXXX]

Agent processing cannot be accepted because an error occurred in the agent execution reservation. [EXENAME:XXXX]

XXXX : エージェント実行名

要因

エージェント実行予約処理でエラーが発生したため、エージェント処理を受け付けられません。

処理

このエージェントを実行しません。

対処

このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB072AL-W

指定されたタイマ指定エージェントの最大数が、既に登録済みのタイマ指定エージェント数より少ないため、既定値を変更しませんでした。

[REGTIMAGTNUM=XXXX] [DABAGTTIMMAX=YYYY]

The specified maximum number of timer agents is denied because it is less than the number of timer agents already set. [REGTIMAGTNUM=XXXX] [DABAGTTIMMAX=YYYY]

XXXX：登録済みのタイマ指定エージェント数

YYYY：指定されたタイマ指定エージェントの最大数

要因

指定されたタイマ指定エージェントの最大数が、既に登録済みのタイマ指定エージェント数より少ないため、既定値を変更しませんでした。

処理

指定されたタイマ指定エージェントの最大数(DABAGTTIMMAX)を無視して処理を続行します。

対処

必要に応じて、適切なタイマ指定エージェントの最大数を指定して DABroker を再起動してください。

KFDB073AL-W

タイマ実行予約処理がキャンセルされました。 [EXENAME: EEEE]

Reservation processing was canceled. [EXENAME:EEEE]

EEEE：エージェント実行名

要因

エージェントタイマ実行予約処理が、エージェント実行削除要求などによってキャンセルされました。

処理

このエージェントを実行しません。

対処

必要に応じてエージェントを再実行してください。

KFDB074AL-W

エージェントの実行予約日時は、現在 DABroker が稼働しているプラットフォームでは実行できません。 [EXENAME EEEE]

The reserved time and date to execute the agent are invalid for the current platform on which DABroker is operating. [EXENAME:EEEE]

EEEE：エージェント実行名

要因

タイマ予約しようとしたエージェントの実行予約日時は、現在 DABroker が稼働しているプラットフォーム(OS)では実行できません。

処理

このエージェントを実行しません。

対処

予約日時を確認して、必要に応じて再実行してください。

KFDB080AL-E

指定されたエージェント名が見つかりません。 [REGNAME:XXXX]

The specified agent name was not found. [REGNAME:XXXX]

XXXX：エージェント名

要因

指定されたエージェント名が見つかりません。

処理

このエージェントを実行しません。

対処

エージェント名を再度確認してください。

KFDB081AL-E

指定されたエージェント実行名が見つかりません。 [EXENAME:XXXX] [EXEMEMO:YYYY]

The specified agent execution name was not found. [EXENAME:XXXX] [EXEMEMO:YYYY]

XXXX：エージェント実行名

YYYY：エージェント実行のメモ名

要因

指定されたエージェント実行名が見つかりません。

処理

このエージェントを実行しません。

対処

エージェント実行名を再度確認してください。

KFDB082AL-E

エージェントの状態が'ZZZZ'のためエージェント実行結果が削除できません。 [EXENAME:XXXX]
[EXEMEMO:YYYY]

The agent execution result cannot be deleted because the agent status is 'ZZZZ'.

[EXENAME:XXXX] [EXEMEMO:YYYY]

XXXX：エージェント実行名

YYYY：エージェント実行のメモ名

ZZZZ：エージェントの状態を表す値

IDLE：初期状態

RUN：実行中

TIMER-WATCH：タイマ監視中

要因

現在のエージェントが実行結果を削除できない状態(ZZZZ)のため、削除を実行できません。

処理

このエージェントを実行しません。

対処

エージェントの実行をキャンセルするか、又はエージェントの実行が終了してから、実行結果の削除を要求してください。

KFDB083AL-E

エージェント実行結果を削除する権限がありません。[EXENAME:XXXX] [EXEMEMO:YYYY]

The user does not have authorization to delete the agent execution result.

[EXENAME:XXXX] [EXEMEMO:YYYY]

XXXX：エージェント実行名

YYYY：エージェント実行のメモ名

要因

エージェント実行結果を削除する権限がありません。

処理

このエージェントを実行しません。

対処

エージェントを実行したユーザで再実行してください。

KFDB084AL-E

エージェントの状態が'ZZZZ'のためエージェント結果が取得できません。[EXENAME:XXXX]

[EXEMEMO:YYYY]

The agent result cannot be obtained because the agent status is 'ZZZZ'.

[EXENAME:XXXX] [EXEMEMO:YYYY]

XXXX：エージェント結果取得を要求したエージェント実行名

YYYY：エージェント結果取得を要求したエージェント実行のメモ名

ZZZZ：エージェントの状態を表す値

ABNORMAL-ENDED：異常終了

IDLE：初期状態

RUN：実行中

TIMER-WATCH：タイマ監視中

要因

エージェントがエージェント結果を取得できない状態(正常終了以外)のため、エージェント結果取得が実行できません。

処理

このエージェントを実行しません。

対処

エージェントが結果を取得できる状態の時に、エージェント結果取得を実行してください。

KFDB085AL-E

エージェント結果を取得する権限がありません。 [EXENAME:XXXX] [EXEMEMO:YYYY]

The user does not have authorization to obtain the agent result. [EXENAME:XXXX]
[EXEMEMO:YYYY]

XXXX：エージェント結果取得を要求したエージェント実行名

YYYY：エージェント結果取得を要求したエージェント実行のメモ名

要因

エージェント結果を取得する権限がありません。

処理

このエージェントを実行しません。

対処

エージェントを実行したユーザで再実行してください。

KFDB086AL-E

ユーザ ID が指定されていません。

The user ID is not specified.

要因

ユーザ ID が指定されていません。

処理

このエージェントを実行しません。

対処

ユーザ ID を指定して再実行してください。

KFDB087AL-W

エージェントの実行はキャンセルされました。 [EXENAME:XXXX]

Agent execution was canceled. [EXENAME:XXXX]

XXXX：エージェント実行名

要因

このエージェント実行はキャンセルされました。

処理

このエージェントの実行を中止します。

対処

必要に応じて再実行してください。

KFDB100AL-E

エージェント処理の実行中に内部エラーが発生しました。 [XXXX]

An internal error occurred while the agent was processing. [XXXX]

XXXX：保守用情報

要因

エージェント処理の実行中に DABroker システムの内部エラーが発生しました。

処理

処理を終了します。

対処

保守員に連絡してください。

KFDB201AL-E

コマンドの引数に誤りがあります。 使用法：dabagtreinfo [-r エージェント名] [-u ユーザ ID]

A command argument is incorrect. Usage: dabagtreinfo[-r agent-name][-u user-ID]

要因

dabagtreinfo コマンドのオプションに誤りがあります。

処理

処理を終了します。

対処

コマンドを正しく指定して、再実行してください。

KFDB202AL-E

コマンドの引数に誤りがあります。

使用法:dabagtxeinfo [-r エージェント名] [-e エージェント実行名] [-u ユーザ ID]

A command argument is incorrect.

Usage: dabagtxeinfo [-r agent-name][-e agent-execution-name][-u user-ID]

要因

dabagtxeinfo コマンドのオプションに誤りがあります。

処理

処理を終了します。

対処

コマンドを正しく指定して、再実行してください。

KFDB300AL-E

システムによりエージェント処理が中断されました。または、エージェント実行結果ファイルの入力処理でエラーが発生しました。要求… [XXXX], エラーコード… [YYYY]

The system canceled agent processing or an error occurred while inputting the agent execution result file. Request --- [XXXX], error code --- [YYYY]

XXXX：要因

YYYY：エラーコード

要因

システムに負荷が掛かり過ぎてエージェント処理が中断されました。または、エージェント実行結果ファイルの入力処理でエラーが発生しました。

処理

エージェント処理を終了します。

対処

システムに負荷が掛かり過ぎていないか確認して、処理を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB001CM-E

メモリが不足しているため、処理に必要な領域が確保できません。 [errno=XXXX]
Cannot allocate memory area because of insufficient memory. [errno=XXXX]

XXXX:エラー番号

要因

メモリ不足が発生しました。

処理

コネクションマルチ通信処理プロセスを終了します。

対処

DABroker で実行中のほかの処理が終了してから、再実行してください。

KFDB002CM-E

システムコールでエラーが発生しました。 [XXXX() errno= YYYY]
An error occurred in a system call. [XXXX() errno=YYYY]

XXXX：システムコール関数名

YYYY：エラー番号

要因

OS のシステムコール関数でエラーが発生しました。

処理

コネクションマルチ通信処理プロセスを終了します。

対処

操作を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB003CM-E

コネクションマルチ機能のソケット作成でエラーが発生しました。
An error occurred during the making of a socket for the Connection-Multi function.

要因

コネクションマルチ機能で使用するソケットを作成する OS のシステムコール関数(socket)でエラーが発生しました。

処理

コネクションマルチ通信処理プロセスを終了します。

対処

メインフレーム側の通信プログラムを起動していない場合にこのエラーが発生します。コネクションマルチ機能を使用する前にメインフレーム側の通信プログラムを起動しておいてください。

KFDB004CM-E

コネクションマルチ通信処理スレッドの起動でエラーが発生しました。 [errno=XXXX]
An error occurred during the start of a Connection-Multi communication thread.
[errno=XXXX]

XXXX:エラー番号

要因

コネクションマルチ機能で使用する通信スレッドの起動に失敗しました。

処理

コネクションマルチ通信処理プロセスを終了します。

対処

操作を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB005CM-E

プロセスの初期化でエラーが発生しました。 [damcInitProc() errno=XXXX]
An error occurred during process initialization. [damcInitProc() errno=XXXX]

XXXX:エラー番号

要因

コネクションマルチ通信処理プロセスの初期化に失敗しました。

処理

コネクションマルチ通信処理プロセスを終了します。

対処

操作を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB006CM-E

名前付きパイプのオープンでエラーが発生しました。 [errno=XXXX]
An error occurred during the opening of a named pipe. [errno=XXXX]

XXXX: エラー番号

要因

名前付きパイプのオープンに失敗しました。

処理

コネクションマルチ通信処理プロセスを終了します。

対処

操作を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB007CM-E

名前付きパイプのクローズでエラーが発生しました。 [errno=XXXX]
An error occurred during the closing of a named pipe. [errno=XXXX]

XXXX: エラー番号

要因

名前付きパイプのクローズに失敗しました。

処理

コネクションマルチ通信処理プロセスを終了します。

対処

操作を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB008CM-E

名前付きパイプの通信でエラーが発生しました。 [errno=XXXX]
An error occurred during communication of a named pipe. [errno=XXXX]

XXXX: エラー番号

要因

名前付きパイプでの通信に失敗しました。

処理

コネクションマルチ通信処理プロセスを終了します。

対処

操作を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB009CM-I

コネクションマルチ通信処理スレッドを終了します。要因： [XXXX]

The Connection-Multi communication thread will terminate. Cause:[XXXX]

XXXX:終了要因

CANCEL_REQUEST : ユーザキャンセル要求

TERM_REQUEST : 強制終了要求

要因

強制終了, 又はユーザキャンセル要求によりコネクションマルチ機能の通信処理スレッドが終了しました。

処理

コネクションマルチ通信処理スレッドを終了します。

対処

システム管理者が強制終了, 又はユーザキャンセル要求を行っていないか確認してください。

KFDB010CM-E

コネクションマルチ通信処理プロセスを終了します。

The Connection-Multi communication process will terminate.

要因

コネクションマルチ機能の通信処理プロセスが異常終了しました。

処理

コネクションマルチ機能のコネクションを解放します。以降のクライアントからの要求を受け付けません。

対処

kill コマンドなどを入力していないか, 又はシステムに負荷が掛かり過ぎていないか確認してください。エラーの原因が分からない場合は, 保守員に連絡してください。

KFDB011CM-E

コネクションマルチ受信処理スレッドの起動でエラーが発生しました。 [XXXX]

An error occurred during the start of a Connection-Multi receive thread. [XXXX]

XXXX:エラー情報

要因

コネクションマルチ機能で使用する受信処理スレッドの起動に失敗しました。

処理

コネクションマルチ通信処理プロセスを終了します。

対処

操作を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB012CM-E

接続したサーバシステムでシステムエラーが発生しました。 [XXXX]

A system error occurred in the connected server system. [XXXX]

XXXX:保守用情報

要因

接続したサーバシステムでエラーが発生しました。

処理

ユーザからの要求を受け付けません。

対処

接続した DBMS を見直し、指定し直してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB013CM-E

コネクションマルチ通信処理でシステムエラーが発生しました。 [XXXX]

A system error occurred during Connection-Multi communication processing. [XXXX]

XXXX:保守用情報

要因

コネクションマルチ機能の通信処理でエラーが発生しました。

処理

ユーザからの要求を受け付けません。

対処

接続した DBMS 及びコネクションマルチ機能の設定を見直し、再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB001CS-E

コマンドの引数に誤りがあります。 使用法：dabcmstart

Invalid argument. Usage:dabcmstart

要因

dabcmstart コマンドのオプションに誤りがあります。

処理

処理を終了します。

対処

コマンドを正しく指定して、再実行してください。

KFDB002CS-E

有効な定義がありません。DABroker コネクションマルチ機能を終了します。

A valid definition does not exist. The DABroker Connection-Multi function will terminate.

要因

コネクションマルチ機能を使用するための、有効な DABroker コネクションマルチ定義がありません。

処理

処理を終了します。

対処

DABroker コネクションマルチ定義で、最大仮想コネクション数 (MCONMAX) を 2 以上及び通信処理プロセス数 (MCONPROC) に 1 以上の値を指定してください。

KFDB003CS-I

DABroker コネクションマルチ機能の初期化処理が完了しました。

Initialization processing of the DABroker Connection-Multi function has finished.

要因

DABroker コネクションマルチ機能の起動要求[※]を受け付け、初期化処理が完了しました。

処理

DABroker コネクションマルチ機能を起動し、ユーザからのアクセス要求を受け付けます。

注※

Windows 対応の DABroker の場合は、プログラムフォルダに登録された[コネクションマルチ起動]を選択、又はサービスの開始を実行したときです。UNIX 対応の DABroker の場合は、dabcmstart コマンドを実行したときです。

KFDB004CS-I

DABroker コネクションマルチ機能の終了処理を開始しました。mode=NORMAL

Termination processing of the DABroker Connection-Multi function has started.

mode=NORMAL

要因

DABroker コネクションマルチ機能の通常終了処理[※]を開始しました。

処理

DABroker コネクションマルチ機能を終了します。

注※

Windows 対応の DABroker の場合は、プログラムフォルダに登録された[コネクションマルチ終了]を選択し、通常終了を実行したときです。UNIX 対応の DABroker の場合は、dabcmstop コマンドをオプション指定なしで実行したときです。

KFDB005CS-I

```
DABroker コネクションマルチ機能の終了処理を開始しました。mode=FORCED  
Termination processing of the DABroker Connection-Multi function has started.  
mode=FORCED
```

要因

DABroker コネクションマルチ機能の強制終了処理※を開始しました。

処理

DABroker コネクションマルチ機能を終了します。

注※

Windows 対応の DABroker の場合は、[コネクションマルチ終了]ダイアログボックスで強制終了を選択したとき、又はサービスを停止させたときです。UNIX 対応の DABroker の場合は、dabcmstop コマンドを-f オプションを指定して実行したときです。

KFDB006CS-I

```
DABroker コネクションマルチ機能の終了処理を開始しました。mode=PLANNED  
Termination processing of the DABroker Connection-Multi function has started.  
mode=PLANNED
```

要因

DABroker コネクションマルチ機能の計画終了処理※を開始しました。

処理

DABroker コネクションマルチ機能を終了します。

注※

Windows 対応の DABroker の場合は、[コネクションマルチ終了]ダイアログボックスで計画終了を選択したときです。UNIX 対応の DABroker の場合は、dabcmstop コマンドを-p オプションを指定して実行したときです。

KFDB007CS-E

```
DABroker コネクションマルチ機能は計画終了中です。接続できません。
```

The DABroker Connection-Multi function is terminating in the planned termination mode.
Connection is not possible.

要因

Database Connection Server 経由でアクセスするデータベースに接続しようとしたが、DABroker コネクションマルチ機能が計画終了中です。計画終了中は、新たなユーザからの接続要求は受け付けません。

処理

ユーザからの要求を受け付けません。

対処

DABroker コネクションマルチ機能の計画終了が終わってから、DABroker コネクションマルチ機能を再起動してください。その後、Database Connection Server 経由でアクセスするデータベースへの接続を再実行してください。

KFDB008CS-E

接続できる最大仮想コネクション数を超過しました。接続できません。

[XXXX][MCONMAX=YYYY,MCONPROC=ZZZZ]

The number of virtual connections exceeds the maximum that can be connected. Connection is not possible.

[XXXX] [MCONMAX=YYYY,MCONPROC=ZZZZ]

XXXX:コネクションマルチ名

YYYY: 1 通信プロセス当たりで同時に確立できる最大仮想コネクション数

ZZZZ:通信処理プロセス数

要因

接続可能な最大仮想コネクション数(DABroker コネクションマルチ定義で設定した MCONMAX*MCONPROC)を超過して、ユーザからの接続要求が発生しました。

処理

ユーザからの要求を受け付けません。

対処

ほかのユーザが終了するのを待ってから、再実行してください。または、DABroker コネクションマルチ定義の MCONMAX または、MCONPROC の値を大きくしてから、再実行してください。

KFDB009CS-E

DABroker コネクションマルチ機能は既に起動しています。

The DABroker Connection-Multi function is already running.

要因

DABroker コネクションマルチ機能が既に起動されているマシン上で、DABroker コネクションマルチ機能の起動要求※を受け付けました。

処理

処理を終了します。

対処

一つのマシン上では、複数の DABroker コネクションマルチ機能は起動できません。

注※

Windows 対応の DABroker の場合は、プログラムフォルダに登録された[コネクションマルチ起動]を選択、又はサービスの開始を実行したときです。UNIX 対応の DABroker の場合は、dabcmstart コマンドを実行したときです。

KFDB010CS-E

DABroker コネクションマルチ機能のロードモジュールに実行権限がありません。ロードモジュール名:[XXXX]

The user does not have authorization to execute the load module of the DABroker Connection-Multi function. Name:[XXXX]

XXXX:エラーとなったロードモジュールのファイル名称

要因

DABroker コネクションマルチ機能のロードモジュールに実行権限がないため、ユーザからの要求を処理できません。

処理

ユーザからの要求を受け付けません。

対処

DABroker を組み込み直してください。UNIX 対応 DABroker の場合は、その後 dabsetup コマンドで DABroker のセットアップを実行してください。

KFDB011CS-E

シンボリックリンクが多過ぎるため、DABroker コネクションマルチ機能を実行できません。ロードモジュール名:[XXXX]

The DABroker Connection-Multi function cannot start because there are too many symbolic links. Name:[XXXX]

XXXX:エラーとなったロードモジュールのファイル名称

要因

DABroker 組み込み先ディレクトリのシンボリックリンクが多過ぎるため、ユーザからの要求を処理できません。

処理

ユーザからの要求を受け付けません。

対処

システム管理者に連絡し、シンボリックリンクの少ないディレクトリに DABroker を組み込み直してもらってください。

KFDB012CS-E

```
DABroker コネクションマルチ機能のロードモジュールがありません。ロードモジュール名:[XXXX]  
A load module of the DABroker Connection-Multi function was not found. Name:[XXXX]
```

XXXX:エラーとなったロードモジュールのファイル名称

要因

DABroker の環境が破壊されているため、ユーザからの要求を処理できません。

処理

ユーザからの要求を受け付けません。

対処

DABroker を組み込み直してください。UNIX 対応 DABroker の場合は、その後 dabsetup コマンドで DABroker のセットアップを実行してください。

KFDB013CS-E

```
メモリ不足のため、DABroker コネクションマルチ機能を実行できません。ロードモジュール名:  
[XXXX]  
The DABroker Connection-Multi function cannot start because of insufficient memory. Name:  
[XXXX]
```

XXXX:エラーとなったロードモジュールのファイル名称

要因

DABroker コネクションマルチ機能の動作に必要なメモリが不足しています。

処理

ユーザからの要求を受け付けません。

対処

しばらく時間を空けてから、再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB014CS-E

DABroker コネクションマルチ機能を起動する権限がありません。

The user does not have authority to execute the DABroker Connection-Multi function.

要因

DABroker コネクションマルチ機能を起動しようとしたユーザが、DABroker の管理者ユーザ[※]ではないため、DABroker コネクションマルチ機能を起動できません。

処理

処理を終了します。

対処

DABroker コネクションマルチ機能の起動は、DABroker の管理者ユーザが実行してください。

注※

Windows 対応の DABroker の場合は、Administrators グループに属しているユーザ
UNIX 対応の DABroker の場合は、DABroker の管理ユーザ

KFDB015CS-E

コネクションマルチ通信プロセスが異常終了しました。要因:XXXX

The Connection-Multi communication processing terminated abnormally. Cause:XXXX

XXXX:終了要因（保守用情報）

要因

コネクションマルチ通信プロセスが異常終了しました。

処理

コネクションを解放します。処理を続行します。

対処

kill コマンドなどを入力していないか、又はシステムに負荷が掛かり過ぎていないか確認してください。
エラーの原因が分からない場合は、保守員に連絡してください。

KFDB016CS-E

DABroker コネクションマルチ機能が異常終了しました。要因:XXXX

The DABroker Connection-Multi function terminated abnormally. Cause:XXXX

XXXX:終了要因（保守用情報）

要因

kill コマンドを入力したか、又はシステムに負荷が掛かり過ぎて、DABroker コネクションマルチ機能が異常終了しました。

処理

DABroker コネクションマルチ機能を終了します。

対処

DABroker コネクションマルチ機能を再起動する場合は、強制終了を実行後、再起動してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。UNIX 対応の DABroker で保守員に連絡する場合は、以下のファイルを退避しておいてください。

- \$DABROKER/spool 以下のファイル(\$DABROKER : DABroker インストールディレクトリ)
- システムダンプ(DABroker を起動したカレントディレクトリ下の "core" ファイル)

KFDB018CS-W

```
実行可能な最大ユーザ処理プロセス数を超過しました。[XXXX]
[MCONMAX=AAAA,MCONPROC=BBBB,MCONPOOL=CCCC,MRSVPOOL=DDDD]
The number of users exceeds the maximum that can be executed.
[XXXX]
[MCONMAX=AAAA,MCONPROC=BBBB,MCONPOOL=CCCC,MRSVPOOL=DDDD]
```

XXXX:コネクションマルチ名

AAAA: 1 通信プロセス当たりで同時に確立できる最大仮想コネクション数

BBBB:通信処理プロセス数

CCCC:プールコネクション数

DDDD:予約プールコネクション数

要因

DABroker コネクションマルチ定義の予約プールコネクション数に指定した値が、接続可能な最大仮想コネクション数(DABroker コネクションマルチ定義で設定した MCONMAX*MCONPROC)を超えています。

処理

接続可能な最大仮想コネクション数分だけプールし、処理を続行します。

対処

DABroker コネクションマルチ定義の予約プールコネクション数が、最大仮想コネクション数以下になるようにしてください。

KFDB020CS-E

```
DABroker コネクションマルチ機能を起動できません。
The DABroker Connection-Multi function cannot start.
```

要因

DABroker コネクションマルチ機能の起動に失敗しました。

処理

DABroker コネクションマルチ機能を終了します。

対処

直前に出力されたメッセージの指示に従って対策してください。対策後、DABroker コネクションマルチ機能を再起動してください。

KFDB022CS-I

DABroker コネクションマルチ機能の終了処理が完了しました。
Termination processing of the DABroker Connection-Multi function has finished.

要因

DABroker コネクションマルチ機能が終了しました。

KFDB023CS-E

コネクションマルチ通信プロセスを起動できません。
The Connection-Multi communication process cannot start.

要因

コネクションマルチ通信プロセスの起動に失敗しました。

処理

処理を続行します。

対処

直前に出力されたメッセージに対する指示に従ってください。

KFDB024CS-E

DABroker コネクションマルチ機能は起動中です。接続できません。
The DABroker Connection-Multi function is starting. Connection is not possible.

要因

Database Connection Server 経由でアクセスするデータベースに接続しようとしたが、DABroker コネクションマルチ機能が起動中です。起動中は、ユーザからの接続要求は受け付けません。

処理

ユーザからの要求を受け付けません。

対処

しばらく時間を空けてから、Database Connection Server 経由でアクセスするデータベースへの接続を再実行してください。

KFDB025CS-E

DABroker コネクションマルチ機能は終了中です。接続できません。

The DABroker Connection-Multi function is terminating. Connection is not possible.

要因

Database Connection Server 経由でアクセスするデータベースに接続しようとしたが、DABroker コネクションマルチ機能が終了処理中です。終了処理中は、ユーザからの接続要求は受け付けません。

処理

ユーザからの要求を受け付けません。

対処

DABroker コネクションマルチ機能の終了処理が終わってから、DABroker コネクションマルチ機能を再起動してください。その後、Database Connection Server 経由でアクセスするデータベースへの接続を再実行してください。

KFDB026CS-E

DABroker コネクションマルチ機能は強制終了中です。接続できません。

The DABroker Connection-Multi function is terminating in the forced termination mode.

Connection is not possible.

要因

Database Connection Server 経由でアクセスするデータベースに接続しようとしたが、DABroker コネクションマルチ機能が強制終了中です。強制終了中は、ユーザからの接続要求は受け付けません。

処理

ユーザからの要求を受け付けません。

対処

DABroker コネクションマルチ機能の強制終了処理が終わってから、DABroker コネクションマルチ機能を再起動してください。その後、Database Connection Server 経由でアクセスするデータベースへの接続を再実行してください。

KFDB027CS-E

DABroker コネクションマルチ機能は起動していません。接続できません。

The DABroker Connection-Multi function is not running. Connection is not possible.

要因

Database Connection Server 経由でアクセスするデータベースに接続しようとしたが、DABroker コネクションマルチ機能が起動されていません。

処理

ユーザからの要求を受け付けません。

対処

DABroker コネクションマルチ機能を起動してください。その後、Database Connection Server 経由でアクセスするデータベースへの接続を再実行してください。

KFDB028CS-E

コネクションマルチ通信処理プロセスを起動できません。接続できません。

The DABroker Connection-Multi function communication process cannot start. Connection is not possible.

要因

Database Connection Server 経由でアクセスするデータベースに接続しようとしたが、コネクションマルチ機能の通信処理プロセスを起動できませんでした。そのため、ユーザからの接続要求を受け付けられません。

処理

ユーザからの要求を受け付けません。

対処

直前に出力されたメッセージに対する指示に従ってください。その後、Database Connection Server 経由でアクセスするデータベースへの接続を再実行してください。

KFDB029CS-E

仮想コネクションの割り当てに失敗しました。接続できません。

The virtual connection-allocation failed. Connection is not possible.

要因

Database Connection Server 経由でアクセスするデータベースに接続しようとしたが、仮想コネクションの割り当てに失敗したため、ユーザからの接続要求を受け付けられません。

処理

ユーザからの要求を受け付けません。

対処

直前に出力されたメッセージに対する指示に従ってください。その後、Database Connection Server 経由でアクセスするデータベースへの接続を再実行してください。

KFDB100CS-E

システムコールでエラーが発生しました。(XXXX() errno=YYYY)

An error occurred in a system call. (XXXX() errno=YYYY)

XXXX:関数名称(保守用情報)

YYYY:エラー番号(保守用情報)

要因

OS のシステムコール関数、又は DABroker の共通関数でエラーが発生しました。

処理

処理を終了します。

対処

操作を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB101CS-E

メモリを確保できません。

Required memory cannot be obtained.

要因

DABroker コネクションマルチ機能の動作に必要なメモリが不足しています。

処理

DABroker コネクションマルチ機能を終了します。

対処

しばらく時間を空けてから、再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB210CS-E

不正なデータを受信しました。[XXXX] エラーコード=YYYY

Invalid data was received. [XXXX] Error code=YYYY

XXXX:コネクションマルチ名

YYYY:保守用情報

要因

ユーザから不正な通信データを受信しました。

処理

ユーザからの要求を無視します。

対処

保守員に連絡してください。

KFDB211CS-E

無効なコネクションマルチ名を指定したため、処理要求を受け付けられません。[XXXX]
The processing request cannot be accepted because a specified Connection-Multi-name is invalid. [XXXX]

XXXX:コネクションマルチ名

要因

コネクションマルチ機能対象外のコネクションマルチ名か、未定義のコネクションマルチ名が指定されました。

処理

処理要求を受け付けません。

対処

有効なコネクションマルチ名を指定して再実行してください。DABroker コネクションマルチ定義の、最大仮想コネクション数を 2 以上に、通信処理プロセス数を 1 以上に設定してください。

KFDB213CS-E

DABroker コネクションマルチ機能が計画終了中のため、処理要求を受け付けられません。[XXXX]
The processing request cannot be accepted because the DABroker Connection-Multi function is terminating in the planned termination mode. [XXXX]

XXXX:コネクションマルチ名

要因

データベースと接続しようとしたのですが、DABroker コネクションマルチ機能が計画終了中です。計画終了中は、新たなユーザからの処理要求を受け付けません。

処理

ユーザからの要求を無視します。

対処

DABroker コネクションマルチ機能の計画終了が終わってから、DABroker コネクションマルチ機能を再起動してください。その後、データベースへの接続を再実行してください。

KFDB300CS-W

services ファイルに、サービス名とポート番号が登録されていません。[XXXX][DBSPORT=YYYY]

The services name and port number are not listed in the "services" file. [XXXX]
[DBSPORT=YYYY]

XXXX:コネクションマルチ名

YYYY:サービス名

要因

コネクションマルチ定義に設定したサービス名が services ファイルに登録されていません。

処理

ユーザからの要求を受け付けません。

対処

コネクションマルチ定義を確認し、再実行してください。

KFDB301CS-W

hosts ファイルに、ホスト名と IP アドレスが登録されていません。 [XXXX][DBSHOST=YYYY]
The host name and IP address are not listed in the "hosts" file. [XXXX][DBSHOST=YYYY]

XXXX:コネクションマルチ名

YYYY:ホスト名

要因

コネクションマルチ定義に設定したホスト名が hosts ファイルに登録されていません。

処理

ユーザからの要求を受け付けません。

対処

コネクションマルチ定義を確認し、再実行してください。

KFDB001DB-E

Required memory cannot be obtained. ErrInfo [XXXX]

XXXX: 保守用情報

要因

処理に必要な領域の確保時、メモリ不足が発生しました。

処理

処理を終了します。

対処

実行中の不要なプログラムを終了してから、再実行してください。なお、このメッセージが繰り返し発生する場合は、実メモリ、又は仮想メモリの設定を見直してください。

KFDB002DB-E

```
RDBMS に対する要求シーケンスが不正です。 ErrInfo [XXXX]  
The request sequence for the RDBMS is invalid. ErrInfo[XXXX]
```

XXXX：保守用情報

要因

ユーザから RDBMS に対する要求シーケンスが誤っています。

処理

ユーザからの要求を受け付けません。

対処

RDBMS に対する要求シーケンスを見直してください。

KFDB003DB-E

```
システムコールでエラーが発生しました。 ErrInfo [XXXX() errno=YYYY]  
An error occurred in a system call. ErrInfo[XXXX() errno=YYYY]
```

XXXX：システムコール名称

YYYY：エラー番号

要因

OS のシステムコール関数でエラーが発生しました。

処理

処理を終了します。

対処

使用する OS のマニュアルを参照しエラー原因を取り除いてから、再実行してください。なお、代表的なシステムコールのエラー番号については、「[7.5.3 システムコールのエラー番号](#)」を参照してください。

KFDB004DB-I

```
DBMS に対する処理を中断しました。  
DBMS processing was canceled.
```


要因

DBMS に対する処理を中断しました。

処理

DBMS に対する処理を中断します。

対処

必要に応じて再実行してください。

KFDB005DB-E

DABroker でサポートされていないデータ属性がありました。 [XXXX]

A data attribute that is not supported by DABroker was found. [XXXX]

XXXX：サポートされていない列属性コード(16進表示)

要因

指定した表列属性(XXXX)は、DABroker でサポートしていません。

処理

ユーザからの要求を受け付けません。

対処

列属性、又は DABroker のバージョンを見直してください。また、列属性コードについては接続先のデータベースのマニュアルを参照してください。

KFDB006DB-E

指定した DBMS と接続した DBMS が異なります。user DBMS=XXXX system DBMS=YYYY
nodename=ZZZZ

The specified DBMS differs from the connected DBMS.

user DBMS=XXXX system DBMS=YYYY nodename=ZZZZ

XXXX：ユーザが指定した DBMS

YYYY：DABroker が接続した DBMS

ZZZZ：RD ノード名称

要因

ユーザが指定した DBMS と接続した DBMS が異なります。

処理

ユーザの要求を受け付けません。

対処

接続する DBMS を見直し、指定し直してください。

KFDB007DB-E

DABroker では、繰り返し列、及び配列列を含む表はサポートしていません。
DABroker does not support tables containing multivalue columns and array columns.

要因

DABroker では繰り返し列、及び配列列を含む表はサポートしていません。

処理

ユーザからの要求を受け付けません。

対処

列属性を修正し、ユーザからの要求を再実行してください。

KFDB008DB-E

RDA Link for Gateway で障害が発生しました。関数名=XXXX エラーコード=YYYY
nodename=ZZZZ
An error occurred in RDA Link for Gateway.
Function name=XXXX Error code=YYYY nodename=XXXX

XXXX : RDA Link for Gateway の関数名(保守用情報)

YYYY : RDA Link for Gateway のエラーコード(保守用情報)

ZZZZ : RD ノード名称

要因

RDA Link for Gateway で障害が発生しました。

処理

ユーザからの要求を受け付けません。

対処

保守員に連絡してください。

KFDB009DB-E

RDA Link for Gateway で障害が発生しました。
nodename=XXXX 障害コード=YYYY メッセージテキスト=ZZZZ
An error occurred in RDA Link for Gateway.
nodename=XXXX Error code=YYYY Message text=ZZZZ

XXXX : RD ノード名称

YYYY : RDA Link for Gateway の障害コード

ZZZZ : RDA Link for Gateway からのメッセージテキスト

要因

RDA Link for Gateway で障害が発生しました。

処理

ユーザからの要求を受け付けません。

対処

障害コード、メッセージテキストについては、RDA Link for Gateway のマニュアルを参照してください。

KFDB010DB-E

```
RDA Link for Gateway で障害が発生しました。  
nodename=XXXX 障害コード=YYYY IPADDR=ZZZZ  
An error occurred in RDA Link for Gateway.  
nodename=XXXX Error code=YYYY IPADDR=ZZZZ
```

XXXX : RD ノード名称

YYYY : RDA Link for Gateway の障害コード

ZZZZ : IP アドレス又はホスト名

要因

RDA Link for Gateway で障害が発生しました。

処理

ユーザからの要求を受け付けません。

対処

障害コードについては、RDA Link for Gateway のマニュアルを参照してください。

KFDB011DB-E

```
接続している DBMS でエラーが発生しました。nodename=XXXX SQLCODE=YYYY  
Info=[ZZZZ]  
An error occurred in the connected DBMS. nodename=XXXX SQLCODE=YYYY Info=[ZZZZ]
```

XXXX : RD ノード名称

YYYY : 接続している DBMS の SQL コード

ZZZZ : 接続している DBMS の SQL コードに対する付加情報 (付加情報がない場合は出力されません。)

要因

接続している DBMS でエラーが発生しました。

処理

ユーザからの要求を受け付けません。

対処

接続している DBMS のマニュアルを参照してください。

KFDB012DB-E

接続している DBMS でエラーが発生しました。

nodename=XXXX SQLCODE=YYYY IPADDR=ZZZZ

An error occurred in the connected DBMS. nodename=XXXX SQLCODE=YYYY
IPADDR=ZZZZ

XXXX：RD ノード名称

YYYY：接続している DBMS の SQL コード

ZZZZ：IP アドレス又はホスト名

要因

接続している DBMS でエラーが発生しました。

処理

ユーザからの要求を受け付けません。

対処

接続している DBMS のマニュアルを参照してください。

KFDB013DB-E

ユーザが指定した、認可識別子、パスワード、又は DBMS 情報のどれかが 32 バイトを超えています。

The permission identifier, password or DBMS information specified by the user exceeded 32 bytes.

要因

ユーザが指定した、認可識別子、パスワード、又は DBMS 情報のどれかが 32 バイトを超えています。

処理

ユーザからの要求を受け付けません。

対処

認可識別子、パスワード、又は DBMS 情報の文字列の長さを確認し、制限バイト以内になるように指定し直してください。

KFDB014DB-E

分散サーバでエラーが発生しました。nodename=XXXX エラーテキスト=YYYY
An error occurred in the distributed server. nodename=XXXX error text=YYYY

XXXX：RD ノード名称

YYYY：エラー発生システムのエラーメッセージ

要因

分散サーバでエラーが発生しました。

処理

ユーザからの要求を受け付けません。

対処

エラーが発生したシステムのマニュアルを参照してください。

KFDB015DB-E

RDA Link for Gateway で Loading エラーが発生しました。ErrInfo [XXXX() errno=YYYY]
A loading error occurred in RDA Link for Gateway. ErrInfo[XXXX() errno=YYYY]

XXXX：エラーが発生した関数名

YYYY：エラーコード

要因

RDA Link for Gateway のライブラリのロード処理でエラーが発生しました。

処理

ユーザからの要求を受け付けません。

対処

RDA Link for Gateway を組み込み直してください。

KFDB016DB-E

Agent で Loading エラーが発生しました。ErrInfo [XXXX() errno=YYYY]
A loading error occurred in Agent. ErrInfo[XXXX() errno=YYYY]

XXXX：DABroker の関数名(保守用情報)

YYYY：DABroker のエラーコード(保守用情報)

要因

Agent 機能を使用する場合に必要なライブラリの処理でエラーが発生しました。

処理

ユーザからの要求を受け付けません。

対処

DABroker を組み込み直してください。

KFDB017DB-E

```
表属性の指定が誤っています。ErrInfo [XXXX]
The specified table attribute is incorrect. ErrInfo[XXXX]
```

XXXX：エラーコード

要因

表属性の指定が誤っています。

処理

ユーザからの要求を受け付けません。

対処

表属性を正しく指定してから、再実行してください。

なお、出力されたエラーコードによって次の内容を見直してください。

- エラーコードが 1 の場合
表属性の前後が ` で囲まれているか確認してください。
- エラーコードが 2 の場合
指定されている表属性が誤っていないか確認してください。
- エラーコードが 3 の場合
複数の表属性を指定した場合の区切り文字を確認してください。

KFDB018DB-E

```
入力パラメタの指定が誤っています。ErrInfo [XXXX, YYYY, ZZZZ]
The specified input parameter is incorrect. ErrInfo [XXXX, YYYY, ZZZZ]
```

XXXX：入力パラメタの行番号

YYYY：入力パラメタの列番号

ZZZZ：エラー項目名

要因

入力パラメタの指定が誤っています。

処理

ユーザからの要求を受け付けません。

対処

入力パラメタを正しく指定してから、再実行してください。

KFDB019DB-E

出力パラメタの指定が誤っています。ErrInfo [XXXX, YYYY]
The specified output parameter is incorrect. ErrInfo[XXXX, YYYY]

XXXX：出力パラメタの列番号

YYYY：エラー項目名

要因

出力パラメタの指定が誤っています。

処理

ユーザからの要求を受け付けません。

対処

出力パラメタを正しく指定してから、再実行してください。

KFDB020DB-E

DBMS のバージョンが違います。ErrInfo [XXXX]
Incorrect version of the DBMS. ErrInfo[XXXX]

XXXX：保守用情報

要因

DBMS は接続できるバージョンではありません。

処理

ユーザからの要求を受け付けません。

対処

システム管理者に DBMS のバージョンを確認してください。

KFDB021DB-E

複数 SQL 実行の指定が誤っています。ErrInfo [XXXX, YYYY]
The specified multiple SQL execution is incorrect. ErrInfo[XXXX,YYYY]

XXXX：SQL 番号

YYYY：詳細コード

要因

複数 SQL 実行の指定が誤っています。

処理

ユーザからの要求を受け付けません。

対処

複数 SQL 実行を正しく指定してから、再実行してください。

KFDB022DB-E

```
RDA Link for Gateway で障害が発生しました。関数名=XXXX エラーコード=YYYY  
nodename=ZZZZ
```

```
An error occurred in RDA Link for Gateway.
```

```
Function name=XXXX Error code=YYYY nodename=ZZZZ
```

XXXX : RDA Link for Gateway の関数名 (保守用情報)

YYYY : RDA Link for Gateway のエラーコード (保守用情報)

ZZZZ : RD ノード名称

要因

RDA Link for Gateway で障害が発生しました。

処理

ユーザからの要求を受け付けません。

対処

保守員に連絡してください。

KFDB023DB-E

```
RDA Link for Gateway で障害が発生しました。nodename=XXXX 障害コード=YYYY メッセージ  
テキスト=ZZZZ
```

```
An error occurred in RDA Link for Gateway.
```

```
nodename=XXXX Error code=YYYY Message text=ZZZZ
```

XXXX : RD ノード名称

YYYY : RDA Link for Gateway の障害コード

ZZZZ : RDA Link for Gateway からのメッセージテキスト

要因

RDA Link for Gateway で障害が発生しました。

処理

ユーザからの要求を受け付けません。

対処

障害コード、メッセージテキストについては、RDA Link for Gateway のマニュアルを参照してください。

KFDB024DB-E

```
接続している DBMS の制限のため、 ? パラメタで NULL を指定できません。 [XXXX]
You cannot specify NULL in the ? parameter because of a limit of the connected DBMS.
[XXXX]
```

XXXX：NULL が指定された ? パラメタの位置

要因

? パラメタに NULL が指定されました。

処理

ユーザからの要求を受け付けません。

対処

? パラメタを見直して再実行してください。

KFDB025DB-E

```
HiRDB または ORACLE に対する接続環境が整っていません。エラーコード=XXXX
Not prepared to connect to HiRDB or ORACLE. Error code=XXXX
```

XXXX：保守コード

DBMS が ORACLE の場合、次の OCI 関数名を表示します。

- 1：xaoEnv
- 2：xaoSvcCtx
- 3：OCIHandleAlloc
- 4：OCISvcCtxToLda

要因

次のどれかの要因が考えられます。

- 1.OTS の初期化が完了していない。
- 2.HiRDB が TPBroker(OTS)に登録されていない。
- 3.DABroker for C++ を使用した UAP をご利用の場合、DBRDatabase クラスの Connect メソッドの引数 lpctDBN に指定した xa_open 文字列の DB フィールドで指定するデータベース名が誤っている。

処理

ユーザからの要求を受け付けません。

対処

1. OTS の初期化完了前に DABroker にアクセスを要求している可能性があります。アプリケーションを確認してください。OTS の初期化については、接続するデータベースのマニュアルを参照してください。
2. TPBroker に HiRDB を登録しているかを確認してください。TPBroker への HiRDB の登録については、TPBroker および HiRDB のマニュアルを参照してください。
3. DABroker for C++ を使用した UAP を利用した場合、DBRDatabase クラスの Connect メソッドの引数 lpctDBN に指定した xa_open 文字列の DB フィールドで指定するデータベース名を見直してください。

KFDB026DB-E

```
ORACLE の OCI 関数でエラーが発生しました。関数名=XXXX リターンコード=YYYY  
The error occurred by the OCI function of ORACLE.Function name=XXXX return  
code=YYYY
```

XXXX : OCI 関数名

YYYY : OCI 関数のリターンコード

要因

OCI 関数 XXXX でエラーが発生しました。

処理

ユーザからの要求を受け付けません。

対処

Oracle Call Interface のマニュアルを参照してください。

KFDB027DB-E

```
LONG 型または LONG RAW 型のデータサイズが大きすぎるため、メモリを確保できません。  
ErrInfo[XXXX]  
Required memory cannot be obtained because the data size of the LONG type or LONG RAW  
type is too large. ErrInfo[XXXX]
```

XXXX : 保守情報

要因

処理に必要な領域の確保時、メモリ不足が発生しました。

処理

ユーザからの要求を受け付けません。

対処

1. 実行中の不要なプログラムを終了してから、再実行してください。
2. 一度に検索するレコード数を少なくしてください。

なお、このメッセージが繰り返し発生する場合は、LONG 型または LONG RAW 型を持つデータのサイズを確認して、実メモリ、または仮想メモリの設定を見直してください。

KFDB030DB-E

ODBC 接続でエラーが発生しました。

```
DATASOURCE NAME=WWWW SQL CODE=XXXX SQLSTATE=YYYY MESSAGE  
TEXT=ZZZZ
```

```
An error occurred in an ODBC access. DATASOURCE NAME=WWWW SQL CODE=XXXX  
SQLSTATE=YYYY MESSAGE TEXT=ZZZZ
```

WWWW：接続データソース名

XXXX：SQL コード，又は DABroker の保守用情報

YYYY：SQLSTATE

ZZZZ：メッセージテキスト(エラー発生システムのエラーメッセージ)

要因

ODBC で障害が発生しました。

処理

ユーザからの要求を受け付けません。

対処

接続している DBMS のマニュアルを参照してください。

KFDB031DB-E

ODBC 接続でエラーが発生しました。

```
SQL CODE=XXXX SQLSTATE=YYYY MESSAGE TEXT=ZZZZ
```

```
An error occurred in an ODBC access.
```

```
SQL CODE=XXXX SQLSTATE=YYYY MESSAGE TEXT=ZZZZ
```

XXXX：SQL コード，又は DABroker の保守用情報

YYYY：SQLSTATE

ZZZZ：メッセージテキスト(エラー発生システムのエラーメッセージ)

要因

ODBC で障害が発生しました。

処理

ユーザからの要求を受け付けません。

対処

接続している DBMS のマニュアルを参照してください。

KFDB032DB-E

```
ODBC 接続で要求パラメタが不正です。 ErrInfo=XXXX  
A request parameter is invalid for ODBC access. ErrInfo=XXXX
```

XXXX：保守用情報

要因

ODBC でパラメタが誤っています。

処理

ユーザからの要求を受け付けません。

対処

保守員に連絡してください。

KFDB040DB-E

```
DABroker ODBC 接続でエラーが発生しました。  
DATASOURCE NAME=WWWW SQL CODE=XXXX SQLSTATE=YYYY  
MESSAGE TEXT=ZZZZ  
An error occurred in a DABroker ODBC access.  
DATASOURCE NAME=WWWW SQL CODE=XXXX SQLSTATE=YYYY  
MESSAGE TEXT=ZZZZ
```

WWWW：接続データソース名

XXXX：SQL コード，又は DABroker の保守用情報

YYYY：SQLSTATE

ZZZZ：メッセージテキスト(エラー発生システムのエラーメッセージ)

要因

DABroker ODBC で障害が発生しました。

処理

ユーザからの要求を受け付けません。

対処

接続している DBMS のマニュアルを参照してください。

KFDB041DB-E

```
DABroker ODBC 接続でエラーが発生しました。  
SQL CODE=XXXX SQLSTATE=YYYY MESSAGE TEXT=ZZZZ  
An error occurred in a DABroker ODBC access.  
SQL CODE=XXXX SQLSTATE=YYYY MESSAGE TEXT=ZZZZ
```

XXXX：SQL コード，又は DABroker の保守用情報

YYYY：SQLSTATE

ZZZZ：メッセージテキスト(エラー発生システムのエラーメッセージ)

要因

DABroker ODBC で障害が発生しました。

処理

ユーザからの要求を受け付けません。

対処

接続している DBMS のマニュアルを参照してください。

KFDB042DB-E

```
DABroker ODBC 接続で要求パラメタが不正です。  
ErrInfo=XXXX  
A request parameter is invalid for DABroker ODBC access.  
ErrInfo=XXXX
```

XXXX：保守用情報

要因

DABroker ODBC 接続でパラメタが誤っています。

処理

ユーザからの要求を受け付けません。

対処

保守員に連絡してください。

KFDB043DB-E

接続した ODBC Driver のバージョンは未サポートです。[XXXX]
DABroker does not support the ODBC Driver version.[XXXX]

XXXX:ODBC Driver のバージョン

要因

接続した ODBC Driver はサポートされていないバージョンです。

処理

ユーザからの要求を受け付けません。

対処

ODBC Driver のバージョンを確認してください。

KFDB044DB-E

接続した DBMS は未サポートです。[XXXX]
DABroker does not support the DBMS. [XXXX]

XXXX：保守用情報

要因

接続した DBMS はサポートしていません。

処理

ユーザからの要求を受け付けません。

対処

接続した DBMS を見直し、指定し直してください。

KFDB045DB-E

指定した DBMS と接続した DBMS が異なります。
user DBMS=XXXX system DBMS=YYYY
The specified DBMS differs from the connected DBMS.
user DBMS=XXXX system DBMS=YYYY

XXXX：ユーザが指定した DBMS

YYYY：DABroker が接続した DBMS

要因

ユーザが指定した DBMS と接続した DBMS が異なります。

処理

ユーザからの要求を受け付けません。

対処

接続する DBMS を見直し、指定し直してください。

KFDB050DB-E

```
[DABroker.libdaTCPS]で Loading エラーが発生しました。ErrInfo[XXXX]  
A loading error occurred in [libdaTCPS]. ErrInfo[XXXX]
```

XXXX：保守用情報

要因

Database Connection Server アクセスに必要なライブラリの処理でエラーが発生しました。

処理

ユーザからの要求を受け付けません。

対処

DABroker を組み込み直してください。

KFDB051DB-E

```
接続したサーバーシステムはサポートしていません。 [XXXX,YYYY]  
DABroker does not support the connected server system. [XXXX,YYYY]
```

XXXX：接続サーバ名

YYYY：保守用情報

要因

接続しようとしたサーバシステムはサポートしていません。

処理

ユーザからの要求を受け付けません。

対処

接続する DBMS を見直し、指定し直してください。

KFDB052DB-E

```
接続したサーバーシステムでシステムエラーが発生しました。 [XXXX]  
A system error occurred in the connected server system. [XXXX]
```

XXXX：保守用情報

要因

接続したサーバーシステムでエラーが発生しました。

処理

ユーザからの要求を受け付けません。

対処

接続した DBMS を見直し、指定し直してください。

KFDB053DB-E

DABroker DBMS ドライバでシステムエラーが発生しました。 [XXXX]

A system error occurred in the DABroker DBMS driver. [XXXX]

XXXX：保守用情報

要因

DBMS ドライバでエラーが発生しました。

処理

ユーザからの要求を受け付けません。

対処

接続する DBMS を見直し、指定し直してください。

KFDB054DB-E

タイムオーバーのため処理を中断しました。

Processing was canceled because a timeout occurred.

要因

タイムアウトが発生しました。

処理

ユーザからの要求を受け付けません。

対処

再度実行してください。なお、このエラーが繰り返し発生する場合は、応答監視時間(DBSTIMER、又は DABDBA_TIMEOUT)の値を大きくしてください。

KFDB060DB-E

XA 要求でエラーが発生しました。 [XXXX]

An XA request error occurred. [XXXX]

XXXX：保守用情報

要因

XA 要求でエラーが発生しました。

処理

ユーザからの要求を受け付けません。

対処

「7.5.4 XA 要求時のエラー要因」を参照してください。

KFDB070DB-E

配列の要素に、異なるデータ型があります。[XXXX]

There is a data type which is different in element of arrangement.[XXXX]

XXXX：エラーの原因となったパラメタインデックス

要因

HiRDB の配列インサート機能使用時、配列に指定するデータ型が一致していません。

処理

ユーザからの要求を受け付けません。

対処

配列に指定するデータ型をすべて同じにして、再度実行してください。

KFDB071DB-E

セットされたデータのサイズが、文字データ型の有効範囲を越えています。[XXXX]

The size of data is over the effective range of character data. [XXXX]

XXXX：エラーの原因となったパラメタインデックス

要因

?パラメタにセットされた文字データのサイズが、文字データ型サイズの有効範囲を越えています。

処理

クライアントからの要求を受け付けません。

対処

?パラメタにセットしたデータのサイズを確認し、再度実行してください。

KFDB101DB-E

メモリが不足しているため、処理に必要な領域を確保できません。

Memory is insufficient for processing.

要因

処理に必要な領域の確保時に、メモリ不足が発生しました。

処理

処理を終了します。

対処

実行中のほかの処理を終了してから、再実行してください。

KFDB102DB-E

予約処理用システムファイルが作成できなかったため、予約処理を受け付けられません。ErrInfo [XXXX, YYYY]

Reserved processing cannot be accepted because the system file for reserved processing cannot be created. ErrInfo[XXXX,YYYY]

XXXX：保守用情報

YYYY：保守用情報

要因

予約処理用システムファイルが作成できませんでした。

DABroker 動作環境定義ファイルの DABBATCH に指定されているディレクトリが、次の状態であることが考えられます。

- 書き込み権限がない
- ディレクトリが存在しない
- ディレクトリの容量が不足している
- ディレクトリの下に「.yoyaku」というファイルが存在する
- ディレクトリの下に「.yoyaku/.system」というディレクトリが存在する

処理

クライアントからの要求を受け付けません。

対処

DABroker 動作環境定義ファイルの DABBATCH に指定されているディレクトリに、次のどれかの操作をしてください。

- 書き込み権限を付与する
- ディレクトリを作成する
- ディレクトリの容量を増やす
- ディレクトリの下にファイル「.yoyaku」を削除する
- ディレクトリの下にディレクトリ「.yoyaku/.system」を削除する

KFDB103DB-E

予約処理用システムファイルの書き込み LOCK にエラーが発生したため、予約処理を受け付けられません。ErrInfo [XXXX]

Reserved processing cannot be accepted because an error occurred in a write lock of the system file used for reserved processing. ErrInfo[XXXX]

XXXX：エラー番号

要因

予約処理用システムファイルの書き込み LOCK にエラーが発生しました。

処理

クライアントからの要求を受け付けません。

対処

クライアントからの要求を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB104DB-E

予約処理用システムファイルで書き込みエラーが発生したため、予約処理を受け付けられません。ErrInfo [XXXX]

Reserved processing cannot be accepted because a write error occurred in the system file used for reserved processing. ErrInfo[XXXX]

XXXX：エラー番号

要因

予約処理用システムファイルで書き込みエラーが発生しました。

処理

クライアントからの要求を受け付けません。

対処

クライアントからの要求を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB105DB-E

予約処理用システムファイルでオープンエラーが発生したため、予約処理は受け付けられません。ErrInfo [XXXX]

Reserved processing cannot be accepted because an open error occurred in the system file used for reserved processing. ErrInfo[XXXX]

XXXX：エラー番号

要因

予約処理用システムファイルでオープンエラーが発生しました。

処理

クライアントからの要求を受け付けません。

対処

クライアントからの要求を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB106DB-E

予約処理用システムファイルのファイルモード変更エラーが発生したため、予約処理は受け付けられません。ErrInfo [XXXX]

Reserved processing cannot be accepted because a mode change error occurred in the system file used for reserved processing. ErrInfo[XXXX]

XXXX：エラー番号

要因

予約処理用システムファイルのファイルモード変更エラーが発生しました。

処理

クライアントからの要求を受け付けません。

対処

クライアントからの要求を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB107DB-E

予約処理用作業ファイルが作成できなかったため、予約処理を受け付けられません。ErrInfo [XXXX]

Reserved processing cannot be accepted because the work file cannot be created.
ErrInfo[XXXX]

XXXX：エラー番号

要因

予約処理用作業ファイルが作成できませんでした。DABroker 動作環境定義ファイルの DABTMP に指定されているディレクトリが、次の状態であることが考えられます。

- 書き込み権限がない
- ディレクトリが存在しない
- ディレクトリの容量が不足している

処理

クライアントからの要求を受け付けません。

対処

DABroker 動作環境定義ファイルの DABTMP に指定されているディレクトリに、次のどれかの操作をしてください。

- 書き込み権限を付与する
- ディレクトリを作成する
- ディレクトリの容量を増やす

KFDB108DB-E

予約処理用作業ファイルにレコードの書き込みエラーが発生したため予約処理を受け付けられません。

ErrInfo [XXXX]

Reserved processing cannot be accepted because an error occurred in a record write of the work file used for reserved processing. ErrInfo[XXXX]

XXXX：エラー番号

要因

予約処理用作業ファイルで書き込みエラーが発生しました。

処理

クライアントからの要求を受け付けません。

対処

クライアントからの要求を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB109DB-E

予約処理用システムファイルでレコードの読み込みエラーが発生したため、予約処理を受け付けられません。ErrInfo [XXXX]

Reserved processing cannot be accepted because an error occurred in the record read of the system file used for reserved processing. ErrInfo[XXXX]

XXXX：エラー番号

要因

予約処理用システムファイルで読み込みエラーが発生しました。

処理

クライアントからの要求を受け付けません。

対処

クライアントからの要求を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB110DB-E

予約処理用システムファイルが破壊されているため、予約処理を受け付けられません。

Reserved processing cannot be accepted because the system file used for reserved processing is corrupted.

要因

予約処理用システムファイルが破壊されています。

処理

クライアントからの要求を受け付けません。

対処

クライアントからの要求を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB111DB-E

予約処理用システムファイルのレコードの位置付けにエラーが発生したため、予約処理を受け付けられません。ErrInfo [XXXX]

Reserved processing cannot be accepted because an error occurred in the record position of the system file used for reserved processing. ErrInfo[XXXX]

XXXX：エラー番号

要因

予約処理用システムファイルのレコードの位置付けにエラーが発生しました。

処理

クライアントからの要求を受け付けません。

対処

クライアントからの要求を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB112DB-E

予約処理用プロセスの起動に失敗したため、予約処理を受け付けられません。ErrInfo [XXXX]

Reserved processing cannot be accepted because an error occurred at the start of the process used for reserved processing. ErrInfo[XXXX]

XXXX：エラー番号

要因

予約処理用プロセスの起動に失敗しました。

処理

クライアントからの要求を受け付けません。

対処

クライアントからの要求を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB113DB-E

予約処理用システムファイルのレコードの位置付けにエラーが発生したため、予約処理状態の保存ができません。ErrInfo [XXXX]

Reserved processing status cannot be saved because an error occurred in the record position of the system file used for reserved processing. ErrInfo[XXXX]

XXXX：エラー番号

要因

予約処理用システムファイルのレコードの位置付けにエラーが発生しました。

処理

処理を続行します。

対処

データベースにアクセスし、実行した予約処理どおりにデータベースが更新されているかどうか確認してください。

KFDB114DB-E

予約処理用システムファイルのレコードの書き込みエラーが発生したため、予約処理状態の保存ができません。ErrInfo [XXXX]

Reserved processing status cannot be saved because an error occurred in the record write of the system file used for reserved processing. ErrInfo[XXXX]

XXXX：エラー番号

要因

予約処理用システムファイルのレコードの書き込みエラーが発生したため、予約処理の状態を示す情報が保存できません。

処理

処理を続行します。

対処

データベースにアクセスし、実行した予約処理どおりにデータベースが更新されているかどうか確認してください。

KFDB115DB-E

セマフォ ID の LOCK にエラーが発生したため、予約処理を受け付けられません。ErrInfo [XXXX]
Reserved processing cannot be accepted because an error occurred in a lock of the semaphore ID. ErrInfo[XXXX]

XXXX：エラー番号

要因

セマフォの LOCK にエラーが発生しました。

処理

クライアントからの要求を受け付けません。

対処

クライアントからの要求を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB116DB-E

DABroker が同時に実行できるデータベースアクセス数を越えたため予約処理を受け付けられません。
Reserved processing cannot be accepted because the number of database accesses exceeds the number that DABroker executes simultaneously.

要因

DABroker が同時に実行できるデータベースアクセス数を超過しました。

処理

クライアントからの要求を受け付けません。

対処

しばらく時間を空けてから、再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡し、DABroker 動作環境定義ファイルの DABDBANUMBER の値を大きくしてもらってください。

KFDB117DB-E

セマフォ ID の LOCK にエラーが発生しました。ErrInfo [XXXX]
An error occurred in a lock of the semaphore ID. ErrInfo[XXXX]

XXXX：エラー番号

要因

セマフォ ID の LOCK にエラーが発生しました。

処理

処理を続行します。

対処

データベースにアクセスし、実行した予約処理どおりにデータベースが更新されているかどうか確認してください。

KFDB118DB-E

共有メモリの情報が取得できません。

Information in shared memory cannot be obtained.

要因

共有メモリの情報が取得できませんでした。

処理

処理を続行します。

対処

データベースにアクセスし、実行した予約処理どおりにデータベースが更新されているかどうか確認してください。

KFDB119DB-E

予約処理用システムファイル更新のために必要な領域が確保できなかったため、予約処理状態の保存ができません。

Reserved processing status cannot be saved because the area is insufficient for updating the system file used for reserved processing.

要因

予約処理用システムファイル更新のために必要な領域が確保できませんでした。

処理

処理を続行します。

対処

データベースにアクセスし、実行した予約処理どおりにデータベースが更新されているかどうか確認してください。

KFDB120DB-E

予約処理用システムファイルのオープンにエラーが発生したため、予約処理状態の保存ができません。

ErrInfo [XXXX]

Reserved processing status cannot be saved because an error occurred during opening of the system file used for reserved processing. ErrInfo[XXXX]

XXXX：エラー番号

要因

予約処理用システムファイルのオープンにエラーが発生しました。

処理

処理を続行します。

対処

データベースにアクセスし、実行した予約処理どおりにデータベースが更新されているかどうか確認してください。

KFDB121DB-E

予約処理用システムファイルが破壊されているため、予約処理状態の保存ができません。

Reserved processing status cannot be saved because the system file used for reserved processing is corrupted.

要因

予約処理用システムファイルが破壊されているため、予約処理の状態を示す情報が保存できません。

処理

処理を続行します。

対処

データベースにアクセスし、実行した予約処理どおりにデータベースが更新されているかどうか確認してください。

KFDB122DB-E

予約処理の実行中に SQL 実行エラーが発生したため、予約実行処理を中止しました。

Reserved execute processing is canceled because an error occurred during execution of the SQL statements.

要因

予約処理の実行中に、SQL 実行エラーが発生しました。

処理

処理を終了します。

対処

エラーの原因を取り除いてから、再実行してください。

KFDB123DB-E

予約処理の実行に必要な領域を確保できなかったため、予約実行処理を中止しました。

Reserved execute processing is canceled because the area is insufficient for execution of reserved processing.

要因

予約処理の実行中に必要な領域を確保できませんでした。

処理

処理を終了します。

対処

実行中のほかの処理を終了してから、再実行してください。

KFDB124DB-E

予約処理の実行中に内部エラーが発生したため、予約実行処理を中止しました。ErrInfo [XXXX, YYYY]

Reserved execute processing was canceled because an internal error occurred.
ErrInfo[XXXX,YYYY]

XXXX：保守用情報

YYYY：保守用情報

要因

予約処理の実行中に、内部エラーが発生しました。

処理

処理を終了します。

対処

保守員に連絡してください。

KFDB125DB-E

予約処理用システムファイルのレコードの、書き込み LOCK にエラーが発生したため、予約処理状態の保存ができません。ErrInfo [XXXX]

Reserved processing status cannot be saved because an error occurred in a record write lock of the system file used for reserved processing. ErrInfo[XXXX]

XXXX：エラー番号

要因

予約処理用システムファイルのレコードの書き込み LOCK にエラーが発生しました。

処理

処理を続行します。

対処

データベースにアクセスし、実行した予約処理どおりにデータベースが、更新されているかどうか確認してください。

KFDB126DB-E

予約処理用システムファイルのレコードの読み込みにエラーが発生したため、予約処理状態の保存ができません。ErrInfo [XXXX]

Reserved processing status cannot be saved because an error occurred in a record read of the system file used for reserved processing. ErrInfo[XXXX]

XXXX：エラー番号

要因

予約処理用システムファイルのレコードの読み込みにエラーが発生しました。

処理

処理を続行します。

対処

データベースにアクセスし、実行した予約処理どおりにデータベースが更新されているかどうか確認してください。

KFDB127DB-I

予約処理用システムファイルを作成しました。

The system file used for reserved processing has been created.

要因

予約処理用システムファイルを作成しました。

処理

処理を続行します。

KFDB128DB-E

予約処理用作業ファイルでオープンエラーが発生したため、予約実行処理を中止しました。ErrInfo [XXXX]

Reserved execute processing was canceled because an error occurred during opening of the work file used for reserved processing. ErrInfo[XXXX]

XXXX：エラー番号

要因

予約処理用作業ファイルでオープンエラーが発生しました。

処理

処理を終了します。

対処

再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB129DB-E

予約処理用作業ファイルでレコードの読み込みエラーが発生したため、予約実行処理を中止しました。

ErrInfo [XXXX]

Reserved execute processing was canceled because an error occurred during a read of the work file used for reserved processing. ErrInfo[XXXX]

XXXX：エラー番号

要因

予約処理用作業ファイルで読み込みエラーが発生しました。

処理

処理を終了します。

対処

再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB130DB-E

予約処理の実行エラー情報格納ファイルが作成できなかったため、実行エラー情報は格納しません。

ErrInfo [XXXX]

Execute error information cannot be saved because an error occurred during creation of the execute error information file used for reserved processing. ErrInfo[XXXX]

XXXX：エラー番号

要因

予約処理の実行エラー情報格納ファイルが作成できなかった。

処理

処理を続行します。

対処

既にある予約処理の実行エラー情報を削除したあと、再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB131DB-E

予約処理用システムファイルの操作で内部エラーが発生したため、予約処理状態の保存ができません。
ErrInfo [XXXX, YYYY, ZZZZ]

Reserved processing status cannot be saved because an internal error occurred during operation of the system file used for reserved processing. ErrInfo[XXXX,YYYY,ZZZZ]

XXXX：保守用情報

YYYY：保守用情報

ZZZZ：保守用情報

要因

予約処理用システムファイルの操作で内部エラーが発生しました。

処理

処理を続行します。

対処

データベースにアクセスし、実行した予約処理どおりにデータベースが更新されているかどうか確認してください。

KFDB132DB-E

予約処理用作業ファイルの操作で内部エラーが発生したため、予約実行処理を中止しました。ErrInfo [XXXX, YYYY, ZZZZ]

Reserved execute processing was canceled because an internal error occurred during operation of the work file used for reserved processing. ErrInfo[XXXX,YYYY,ZZZZ]

XXXX：保守用情報

YYYY：保守用情報

ZZZZ：保守用情報

要因

予約処理用作業ファイルの操作で内部エラーが発生しました。

処理

処理を終了します。

対処

再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB133DB-E

予約処理用システムファイルの操作で内部エラーが発生したため、予約処理は受け付けられません。

ErrInfo [XXXX, YYYY, ZZZZ]

Reserved processing cannot be accepted because an internal error occurred during operation of the system file used for reserved processing. ErrInfo[XXXX,YYYY,ZZZZ]

XXXX：保守用情報

YYYY：保守用情報

ZZZZ：保守用情報

要因

予約処理用システムファイルの操作で内部エラーが発生しました。

処理

クライアントからの要求を受け付けません。

対処

クライアントからの要求を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB134DB-E

予約処理用作業ファイルの操作で内部エラーが発生したため、予約処理は受け付けられません。ErrInfo [XXXX, YYYY, ZZZZ]

Reserved processing cannot be accepted because an internal error occurred during operation of the work file used for reserved processing. ErrInfo[XXXX,YYYY,ZZZZ]

XXXX：保守用情報

YYYY：保守用情報

ZZZZ：保守用情報

要因

予約処理用作業ファイルの操作で内部エラーが発生しました。

処理

クライアントからの要求を受け付けません。

対処

クライアントからの要求を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB135DB-E

予約処理の実行エラー情報の書き込みに失敗したため、予約処理の実行エラー情報格納ファイルを削除しました。ErrInfo [XXXX]

The execution error information file for reserved processing was deleted because an error occurred during a write of the execution error information for reserved processing.
ErrInfo[XXXX]

XXXX：エラー番号

要因

予約処理の実行エラー情報の書き込みにエラーが発生しました。

処理

処理を続行します。

対処

既にある予約処理の実行エラー情報を削除したあと、再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB136DB-E

予約実行の初期化でエラーが発生しました。予約実行処理を中止します。

Reserved execution processing was canceled because an error occurred during initialization of reserved execution.

要因

予約実行の初期化でエラーが発生しました。

処理

予約実行を中止しました。

対処

再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB150DB-E

メモリ不足が発生したため、予約情報を取得できません。

Reserved information cannot be obtained because of insufficient memory.

要因

予約情報取得のために必要な領域を確保することができなかった。

処理

クライアントからの要求を受け付けません。

対処

実行中のほかの処理を終了してから、再実行してください。

KFDB151DB-E

予約情報取得のための作業ファイルが作成できなかったため、予約情報を取得できません。ErrInfo [XXXX, YYYY]

Reserved information cannot be obtained because an error occurred during creation of the work file used for reserved processing. ErrInfo[XXXX,YYYY]

XXXX：保守用情報

YYYY：エラー番号

要因

予約情報取得のための作業ファイルが作成できませんでした。

処理

クライアントからの要求を受け付けません。

対処

クライアントからの要求を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB152DB-E

予約処理用システムファイルでオープンエラーが発生したため、予約情報を取得できません。ErrInfo [XXXX]

Reserved information cannot be obtained because an error occurred during opening of the system file used for reserved processing. ErrInfo[XXXX]

XXXX：エラー番号

要因

予約処理用システムファイルでオープンエラーが発生しました。

処理

クライアントからの要求を受け付けません。

対処

クライアントからの要求を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB153DB-E

予約処理用システムファイルが破壊されているため、予約情報を取得できません。ErrInfo [XXXX]

Reserved information cannot be obtained because the system file used for reserved processing is corrupted. ErrInfo[XXXX]

XXXX：エラー番号

要因

予約処理用システムファイルが破壊されています。

処理

クライアントからの要求を受け付けません。

対処

クライアントからの要求を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB154DB-E

予約処理用システムファイルの読み込み LOCK にエラーが発生したため、予約情報を取得できません。ErrInfo [XXXX]

Reserved information cannot be obtained because an error occurred during a read lock of the system file used for reserved processing. ErrInfo[XXXX]

XXXX：エラー番号

要因

予約処理用システムファイルの読み込み LOCK にエラーが発生しました。

処理

クライアントからの要求を受け付けません。

対処

クライアントからの要求を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB155DB-E

予約処理用システムファイルのレコードの読み込みにエラーが発生したため、予約情報を取得できません。ErrInfo [XXXX]

Reserved information cannot be obtained because an error occurred during a read of a record in the system file used for reserved processing. ErrInfo[XXXX]

XXXX：エラー番号

要因

予約処理用システムファイルの読み込みにエラーが発生しました。

処理

クライアントからの要求を受け付けません。

対処

クライアントからの要求を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB156DB-E

予約情報取得用作業ファイルのレコードの書き込みエラーが発生したため、予約情報を取得できません。ErrInfo [XXXX, YYYY]

Reserved information cannot be obtained because an error occurred during the write of a record in the work file used for reserved processing. ErrInfo[XXXX,YYYY]

XXXX：保守用情報

YYYY：保守用情報

要因

予約情報取得用作業ファイルの書き込みにエラーが発生しました。

処理

クライアントからの要求を受け付けません。

対処

クライアントからの要求を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB157DB-E

予約処理の実行エラー情報格納ファイルでオープンエラーが発生したため、予約処理の実行エラー情報を取得できません。ErrInfo [XXXX]

The execution error information for reserved processing cannot be obtained because an error occurred during opening of the execution error information file used for reserved processing. ErrInfo[XXXX]

XXXX：エラー番号

要因

予約処理の実行エラー情報格納ファイルでオープンエラーが発生しました。

処理

処理を続行します。

対処

クライアントからの要求を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB158DB-E

予約処理の実行エラー情報格納ファイルのレコードの位置付けエラーが発生したため、予約処理の実行エラー情報を取得できません。ErrInfo [XXXX]

The execution error information for reserved processing cannot be obtained because an error occurred in the record position of the execution error information file used for reserved processing. ErrInfo[XXXX]

XXXX：エラー番号

要因

予約処理の実行エラー情報格納ファイルの位置付けエラーが発生しました。

処理

処理を続行します。

対処

クライアントからの要求を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB159DB-E

予約処理の実行エラー情報格納ファイルでレコードの読み込みエラーが発生したため、予約処理の実行エラー情報を取得できません。ErrInfo [XXXX]

The execution error information for reserved processing cannot be obtained because an error occurred during a read of a record in the execution error information file used for reserved processing. ErrInfo[XXXX]

XXXX：エラー番号

要因

予約処理の実行エラー情報格納ファイルで読み込みエラーが発生しました。

処理

処理を続行します。

対処

クライアントからの要求を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB160DB-E

予約情報取得用作業ファイルでレコードの位置付けエラーが発生したため、予約情報を取得できません。ErrInfo [XXXX, YYYY]

Reserved information cannot be obtained because an error occurred in the record position of the work file used for reserved processing. ErrInfo[XXXX,YYYY]

XXXX：保守用情報

YYYY：保守用情報

要因

予約情報取得用作業ファイルの位置付けエラーが発生しました。

処理

クライアントからの要求を受け付けません。

対処

クライアントからの要求を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB161DB-E

予約情報取得用作業ファイルでレコードの読み込みエラーが発生したため、予約情報を取得できません。ErrInfo [XXXX, YYYY]

Reserved information cannot be obtained because an error occurred during a read of a record in the work file used for reserved processing. ErrInfo[XXXX,YYYY]

XXXX：保守用情報

YYYY：保守用情報

要因

予約情報取得用作業ファイルの読み込みエラーが発生しました。

処理

クライアントからの要求を受け付けません。

対処

クライアントからの要求を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB162DB-E

予約情報取得用作業ファイルでオープンエラーが発生したため、予約処理の実行エラー情報を取得できません。ErrInfo [XXXX, YYYY]

Execution error information for reserved processing cannot be obtained because an error occurred during the opening of the work file used for reserved processing.
ErrInfo[XXXX,YYYY]

XXXX：保守用情報

YYYY：保守用情報

要因

予約情報取得用作業ファイルでオープンエラーが発生しました。

処理

処理を続行します。

対処

クライアントからの要求を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB163DB-E

予約情報取得用作業ファイルでレコードの読み込みエラーが発生したため、予約処理の実行エラー情報を取得できません。ErrInfo [XXXX, YYYY]

Execution error information for reserved processing cannot be obtained because an error occurred during a read of a record in the work file used for reserved processing.
ErrInfo[XXXX,YYYY]

XXXX：保守用情報

YYYY：保守用情報

要因

予約情報取得用作業ファイルの読み込みエラーが発生しました。

処理

処理を続行します。

対処

クライアントからの要求を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB164DB-E

予約情報取得用作業ファイルのレコード位置付けエラーが発生したため、予約処理の実行エラー情報を取得できません。ErrInfo [XXXX, YYYY]

Execution error information for reserved processing cannot be obtained because an error occurred in the record position of the work file used for reserved processing.
ErrInfo[XXXX,YYYY]

XXXX：保守用情報

YYYY：保守用情報

要因

予約情報取得用作業ファイルのレコード位置付けエラーが発生しました。

処理

処理を続行します。

対処

クライアントからの要求を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB170DB-E

メモリ不足が発生したため、予約処理の中止、及び予約処理の削除ができません。
Reserved processing cannot be canceled or deleted because of insufficient memory.

要因

予約処理の中止、及び予約処理の削除を行うための必要な領域を確保することができなかった。

処理

処理を終了します。

対処

実行中のほかの処理を終了してから、再実行してください。

KFDB171DB-E

予約処理用システムファイルでオープンエラーが発生したため、予約処理の中止、及び予約処理の削除ができません。ErrInfo [XXXX]
Reserved processing cannot be canceled or deleted because an error occurred in the system file used for reserved processing. ErrInfo[XXXX]

XXXX：エラー番号

要因

予約処理用システムファイルでオープンエラーが発生しました。

処理

クライアントからの要求を受け付けません。

対処

クライアントからの要求を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB172DB-E

予約処理用システムファイルが破壊されているため、予約処理の中止、及び予約処理の削除ができません。ErrInfo [XXXX]

Reserved processing cannot be canceled or deleted because the system file used for reserved processing is corrupted. ErrInfo[XXXX]

XXXX：エラー番号

要因

予約処理用システムファイルが破壊されています。

処理

クライアントからの要求を受け付けません。

対処

クライアントからの要求を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB173DB-E

予約処理用システムファイルの書き込み LOCK にエラーが発生したため、予約処理の中止、及び予約処理の削除ができません。ErrInfo [XXXX]

Reserved processing cannot be canceled or deleted because an error occurred during a write lock of the system file used for reserved processing. ErrInfo[XXXX]

XXXX：エラー番号

要因

予約処理用システムファイルの書き込み LOCK にエラーが発生しました。

処理

クライアントからの要求を受け付けません。

対処

クライアントからの要求を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB174DB-E

予約処理用システムファイルでレコードの読み込みエラーが発生したため、予約処理の中止、及び予約処理の削除ができません。ErrInfo [XXXX]

Reserved processing cannot be canceled or deleted because an error occurred during a read of a record in the system file used for reserved processing. ErrInfo[XXXX]

XXXX：エラー番号

要因

予約処理用システムファイルの読み込みエラーが発生しました。

処理

処理を中止します。

対処

予約情報の取得を実行し、予約情報を確認してください。必要なら、クライアントからの要求を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB175DB-E

予約処理用システムファイルのレコードの位置付けエラーが発生したため、予約処理の中止、及び予約処理の削除ができません。ErrInfo [XXXX]

Reserved processing cannot be canceled or deleted because an error occurred in the position of a record in the system file used for reserved processing. ErrInfo[XXXX]

XXXX：エラー番号

要因

予約処理用システムファイルのレコードの位置付けエラーが発生しました。

処理

処理を中止します。

対処

予約情報の取得を実行し、予約情報を確認してください。必要なら、クライアントからの要求を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB176DB-E

予約処理用システムファイルでレコードの書き込みエラーが発生したため、予約処理の中止、及び予約処理の削除ができません。ErrInfo [XXXX]

Reserved processing cannot be canceled or deleted because an error occurred during the write of a record in the system file used for reserved processing. ErrInfo[XXXX]

XXXX：エラー番号

要因

予約処理用システムファイルの書き込みエラーが発生しました。

処理

処理を中止します。

対処

予約情報の取得を実行し、予約情報を確認してください。必要なら、クライアントからの要求を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB177DB-E

```
予約処理の中止処理で、共有メモリのロックに失敗しました。ErrInfo [XXXX]  
DABroker failed to lock shared memory during cancellation of reserved processing.  
ErrInfo[XXXX]
```

XXXX：エラー番号

要因

予約処理の中止処理で、共有メモリのロックに失敗しました。

処理

処理を続行します。

対処

予約情報の取得を実行し、予約情報を確認してください。必要なら、クライアントからの要求を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB178DB-E

```
予約処理の中止に失敗しました。ErrInfo [XXXX]  
DABroker failed to cancel reserved processing. ErrInfo[XXXX]
```

XXXX：エラー番号

要因

予約処理の中止に失敗しました。

処理

処理を続行します。

対処

予約情報の取得を実行し、予約情報を確認してください。必要なら、クライアントからの要求を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB001EC-E

```
コマンドの引数に誤りがあります。使用法：dabstop [-p | -f | -c IP アドレス又はホスト名]  
Invalid argument. Usage:dabstop[-p | -f | -c IP-address_or_host-name]
```

要因

dabstop コマンドのオプションに誤りがあります。

処理

処理を終了します。

対処

コマンドを正しく指定して、再実行してください。

KFDB002EC-E

DABroker は起動していません。

DABroker is not running.

要因

DABroker を終了しようとしたますが、起動していません。

処理

処理を終了します。

KFDB003EC-E

DABroker は起動処理中です。コマンドを取り消します。

DABroker is starting. The command is canceled.

要因

dabstop コマンドを実行しましたが、DABroker の起動処理が完了していません。

処理

処理を終了します。

対処

しばらく時間を空けてから、再実行してください。

KFDB004EC-E

DABroker は終了処理中です。コマンドを取り消します。

DABroker is terminating. The command is canceled.

要因

dabstop コマンドを実行しましたが、DABroker は終了処理中です。

処理

処理を終了します。

KFDB005EC-E

DABroker は計画終了処理中です。コマンドを取り消します。

DABroker is terminating in the planned termination mode. The command is canceled.

要因

dabstop コマンドを実行しましたが、DABroker は計画終了処理中です。

処理

処理を終了します。

KFDB006EC-E

DABroker は強制終了処理中です。コマンドを取り消します。

DABroker is terminating in the forced termination mode. The command is canceled.

要因

dabstop コマンドを実行しましたが、DABroker は強制終了処理中です。

処理

処理を終了します。

KFDB007EC-E

ほかの端末から dabstop コマンドが実行されています。

dabstop has been executed from another terminal.

要因

dabstop コマンドは既にほかの端末から実行されています。

処理

処理を終了します。

KFDB008EC-E

DABroker を終了する権限がありません。

The user does not have authorization to stop DABroker.

要因

DABroker を終了しようとしたユーザが Administrators グループに属していないため、終了できません。

処理

処理を終了します。

対処

DABroker は Administrators グループに属しているユーザで終了してください。

KFDB009EC-E

実行中の処理があるため、終了できません。コマンドを取り消します。

DABroker cannot terminate because of current processing. The command is canceled.

要因

dabstop コマンドをオプションなしで指定しましたが、接続中のクライアント、実行中の予約処理、又は実行中のエージェント処理があるため、DABroker を終了できません。

処理

処理を終了します。

対処

1. 実行中の処理及びすべてのクライアントが終了してから、再実行してください。
2. 強制終了する場合はオプション-f を、計画終了する場合はオプション-P を指定して再実行してください。

KFDB010EC-E

-c オプションで指定したクライアントの処理はありません。

The client processing specified by the -c option was not found.

要因

dabstop コマンドに-c オプションを付け、指定したクライアントの処理を中断しようとしたのですが、該当する処理はありませんでした。

処理

処理を終了します。

対処

dabstat コマンドで、実行中の処理を確認してください。

KFDB011EC-E

-c オプションで指定した IP アドレス又はホスト名に誤りがあります。

An incorrect IP address or host name is specified with the -c option.

要因

IP アドレス又はホスト名に誤りがあります。

処理

処理を終了します。

対処

IP アドレス又はホスト名を正しく指定して、再実行してください。

KFDB012EC-E

DABroker は起動処理中です。
DABroker is starting.

要因

DABroker を終了しようとしたのですが、起動処理が完了していません。

処理

処理を終了します。

対処

しばらく時間を空けてから、再実行してください。

KFDB013EC-E

DABroker は終了処理中です。
DABroker is terminating.

要因

DABroker を終了しようとしたのですが、終了処理中です。

処理

処理を終了します。

KFDB014EC-E

DABroker は計画終了処理中です。
DABroker is terminating in the planned termination mode.

要因

DABroker を終了しようとしたのですが、計画終了処理中です。

処理

処理を終了します。

KFDB015EC-E

DABroker は強制終了処理中です。
DABroker is terminating in the forced termination mode.

要因

DABroker を終了しようとしたのですが、強制終了処理中です。

処理

処理を終了します。

KFDB016EC-E

実行中の処理があるため、終了できません。

DABroker cannot terminate because of current processing.

要因

DABroker を通常終了^{※1}しようとしたのですが、接続中のクライアント、実行中の予約処理、又は実行中のエージェント処理があるため、DABroker を終了できません。

処理

処理を終了します。

対処

1. 実行中の処理、及びすべてのクライアントが終了してから、再実行してください。
2. 強制終了^{※2}、又は計画終了^{※3}を選択してください。

注※1

[DABroker 終了] ダイアログボックスで通常終了を選択します。

注※2

[DABroker 終了] ダイアログボックスで強制終了を選択します。または、サービスを停止させます。

注※3

[DABroker 終了] ダイアログボックスで計画終了選択します。

KFDB017EC-E

指定したクライアントの処理はありません。

The specified client processing was not found.

要因

クライアントの処理を中断しようとしたのですが、該当する処理^{※1}はありませんでした。

処理

処理を終了します。

対処

クライアントの処理状態^{※2}を確認してください。

注※1

[DABroker 管理] ウィンドウで選択されたクライアントの処理です。

注※2

DABroker 管理ユーティリティを使用して確認してください。

KFDB101EC-E

```
コマンドの引数に誤りがあります。  使用法: dabcmstop [-p | -f]
Invalid argument. Usage:dabcmstop [-p | -f]
```

要因

dabcmstop コマンドのオプションに誤りがあります。

処理

処理を終了します。

対処

コマンドを正しく指定して、再実行してください。

KFDB102EC-E

```
DABroker コネクションマルチ機能は起動していません。
The DABroker Connection-Multi function is not running.
```

要因

DABroker コネクションマルチ機能を終了しようとしたますが、起動していません。

処理

処理を終了します。

KFDB103EC-E

```
DABroker コネクションマルチ機能は起動処理中です。コマンドを取り消します。
The DABroker Connection-Multi function is starting. The command is canceled.
```

要因

dabcmstop コマンドを実行しましたが、DABroker コネクションマルチ機能の起動処理が完了していません。

処理

処理を終了します。

対処

しばらく時間を空けてから、再実行してください。

KFDB104EC-E

DABroker コネクションマルチ機能は終了処理中です。コマンドを取り消します。
The DABroker Connection-Multi function is terminating. The command is canceled.

要因

dabcmstop コマンドを実行しましたが、DABroker コネクションマルチ機能は終了処理中です。

処理

処理を終了します。

対処

しばらく時間を空けてから、再実行してください。

KFDB105EC-E

DABroker コネクションマルチ機能は計画終了処理中です。コマンドを取り消します。
The DABroker Connection-Multi function is terminating in the planned termination mode. The command is canceled.

要因

dabcmstop コマンドを実行しましたが、DABroker コネクションマルチ機能は計画終了処理中です。

処理

処理を終了します。

KFDB106EC-E

DABroker コネクションマルチ機能は強制終了処理中です。コマンドを取り消します。
The DABroker Connection-Multi function is terminating in the forced termination mode. The command is canceled.

要因

dabcmstop コマンドを実行しましたが、DABroker コネクションマルチ機能は強制終了処理中です。

処理

処理を終了します。

KFDB107EC-E

ほかの端末から dabcmstop コマンドが実行されています。
dabcmstop has been executed from another terminal.

要因

dabmcstop コマンドは既にほかの端末から実行されています。

処理

処理を終了します。

KFDB108EC-E

DABroker コネクションマルチ機能を終了する権限がありません。

The user does not have authorization to stop the DABroker Connection-Multi function.

要因

DABroker コネクションマルチ機能を終了しようとしたユーザが Administrators グループに属していないため、終了できません。

処理

処理を終了します。

対処

DABroker コネクションマルチ機能は Administrators グループに属しているユーザが終了してください。

KFDB109EC-E

実行中の処理があるため、終了できません。コマンドを取り消します。

The DABroker Connection-Multi function cannot terminate because of current processing.

The command is canceled.

要因

dabcmstop コマンドをオプションなしで指定しましたが、接続中のユーザがあるため、DABroker コネクションマルチ機能を終了できません。

処理

処理を終了します。

対処

1. 実行中の処理が終了してから、再実行してください。
2. 強制終了する場合はオプション-f を、計画終了する場合はオプション-p を指定して、再実行してください。

KFDB110EC-E

-c オプションで指定した仮想コネクションはありません。

This virtual connection is not specified with the -c option.

要因

dabcmstop コマンドに-c オプションを付け、指定した仮想コネクションを中断しようとしたが、該当する処理はありませんでした。

処理

処理を終了します。

対処

dabcmstat コマンドで、実行中の処理を確認してください。

KFDB111EC-E

-d オプションで指定した通信デーモンに対応する処理はありません。
This communication demon is not specified with the -d option.

要因

dabcmstop コマンドに-d オプションを付け、指定した通信デーモンに対応した処理を中断しようとしたが、該当する処理はありませんでした。

処理

処理を終了します。

対処

dabcmstat コマンドで、実行中の処理を確認してください。

KFDB112EC-E

DABroker コネクションマルチ機能は起動処理中です。
The DABroker Connection-Multi function is starting.

要因

DABroker コネクションマルチ機能を終了しようとしたが、起動処理が完了していません。

処理

処理を終了します。

対処

しばらく時間を空けてから、再実行してください。

KFDB113EC-E

DABroker コネクションマルチ機能は終了処理中です。
The DABroker Connection-Multi function is terminating.

要因

DABroker コネクションマルチ機能を終了しようとしたが、終了処理中です。

処理

処理を終了します。

KFDB114EC-E

DABroker コネクションマルチ機能は計画終了処理中です。

The DABroker Connection-Multi function is terminating in the planned termination mode.

要因

DABroker コネクションマルチ機能を終了しようとしたのですが、計画終了処理中です。

処理

処理を終了します。

KFDB115EC-E

DABroker コネクションマルチ機能は強制終了処理中です。

The DABroker Connection-Multi function is terminating in the forced termination mode.

要因

DABroker コネクションマルチ機能を終了しようとしたのですが、強制終了処理中です。

処理

処理を終了します。

KFDB116EC-E

実行中の処理があるため、終了できません。

The DABroker Connection-Multi function cannot terminate because of current processing.

要因

DABroker コネクションマルチ機能を通常終了しようとしたのですが、接続中のユーザ処理があるため、DABroker コネクションマルチ機能を終了できません。

処理

処理を終了します。

対処

1. 実行中の処理が終了してから、再実行してください。
2. 強制終了又は計画終了を選択してください。終了形態については、「[4.10 コネクションマルチ機能の使用](#)」を参照してください。

KFDB117EC-E

指定した仮想コネクションはありません。

This specified virtual connection does not exist.

要因

処理を中断しようとしたのですが、([DABroker コネクションマルチ機能管理] ダイアログボックスで) 選択された仮想コネクションはありませんでした。

処理

処理を終了します。

対処

DABroker コネクションマルチ機能管理ユーティリティを使用して仮想コネクションの状態を確認してください。

KFDB002LC-E

コマンドの引数に誤りがあります。使用法:dabcltrc [-c IP アドレス又はホスト名]
Invalid argument. Usage:dabcltrc [-c IP-address_or_host-name]

要因

dabcltrc コマンドのオプションに誤りがあります。

処理

処理を終了します。

対処

コマンドを正しく指定して、再実行してください。

KFDB003LC-E

ユーティリティの実行中にメモリ不足が発生しました。
Memory became insufficient during execution of a utility program.

要因

ユーティリティの実行中にメモリ不足が発生しました。

処理

処理を終了します。

対処

しばらく時間を空けてから、再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB004LC-E

DABroker は動作していません。
DABroker is not starting.

要因

コマンドを実行しましたが、DABroker が動作していない状態でした。

処理

処理を終了します。

対処

DABroker が動作しているかどうか確認してください。DABroker が動作している状態でこのメッセージが出力された場合は、保守員に連絡してください。

KFDB005LC-E

指定された IP アドレス又はホスト名のクライアントとの通信トレースは見つかりません。
Communication information with the specified IP address or host name cannot be traced.

要因

指定した IP アドレス、又はホスト名のクライアントとの通信トレースが見つかりません。

処理

処理を終了します。

対処

指定した IP アドレス又はホスト名を確認してください。

KFDB001LM-E

システムコールでエラーが発生しました。[XXXX()error=YYYY]
An error occurred in a system call. [XXXX() errno=YYYY]

XXXX：システムコール名称

YYYY：エラー番号

要因

OS のシステムコール関数でエラーが発生しました。

処理

処理を終了します。

対処

操作を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB001MM-E

動作環境定義ファイルが存在しません。[XXXX]
The DABroker environment definition file was not found. [XXXX]

XXXX：DABroker 動作環境定義ファイル名称

要因

DABroker 動作環境定義ファイルが、DABroker 運用ディレクトリ/conf に存在しません。

処理

DABroker を終了します。

対処

DABroker 動作環境定義ファイルを作成してください。

KFDB002MM-E

```
動作環境定義に誤りがあります。 [XXXX]  
The environment definition is incorrect. [XXXX]
```

XXXX：環境変数名

要因

1. DABroker 動作環境定義ファイルに設定した環境変数の値に誤りがあります。
2. ユーザが設定した環境変数の値に誤りがあります。

処理

処理を終了します。

対処

1. テキストエディタを使用して、DABroker 動作環境定義ファイルの環境変数の値を修正してください。
2. ユーザが設定した環境変数の値を修正してください。

KFDB003MM-E

```
動作環境定義に誤りがあります。 LINE=XXXX [YYYY]  
The environment definition is incorrect. LINE=XXXX [YYYY]
```

XXXX：エラー発生行

YYYY：エラー発生定義

要因

- DABroker 動作環境定義ファイルの構文に誤りがあります。
- DABroker 動作環境定義ファイルに設定した環境変数の値に誤りがあります。

処理

処理を終了します。

対処

テキストエディタを使用して、DABroker 動作環境定義ファイルを修正してください。

KFDB004MM-E

```
DABroker is not installed.
```

要因

DABroker がインストールされていません。DABroker のインストール環境が不正です。

処理

処理を中断します。

対処

DABroker を再インストールしてください。

KFDB005MM-E

```
システムコールでエラーが発生しました。[XXXX errno=YYYY]  
An error occurred in a system call. [XXXX errno=YYYY]
```

XXXX：エラー発生関数の名称

YYYY：エラー番号

要因

XXXX 関数でエラーが発生しました。

処理

処理を中断します。

対処

使用する OS のマニュアルを参照しエラー原因を取り除いてから、再実行してください。なお、代表的なシステムコールのエラー番号については、「[7.5.3 システムコールのエラー番号](#)」を参照してください。

KFDB006MM-E

```
System call error occurred. [XXXX errno=YYYY]
```

XXXX：エラー発生関数の名称

YYYY：エラー番号

要因

XXXX 関数でエラーが発生しました。

処理

処理を終了します。

対処

使用する OS のマニュアルを参照しエラー原因を取り除いてから、再実行してください。なお、代表的なシステムコールのエラー番号については、「[7.5.3 システムコールのエラー番号](#)」を参照してください。

KFDB007MM-E

```
The message file could not be opened. [XXXX errno=YYYY]
```

XXXX：メッセージファイル名称

YYYY：エラー番号

要因

DABroker のメッセージファイルがオープンできません。

処理

処理を続行します。このメッセージ出力以降は、メッセージ ID だけを出力します。セットアップコマンドの場合は、処理を中止します。

対処

使用する OS のマニュアルを参照しエラー原因を取り除いてから、再実行してください。DABroker 動作環境の設定項目の LANG 環境変数 (DAB_LANG) に C を設定し、OS の locale を C で稼働した場合、このメッセージが出力されたときは、AIX の環境設定の LC_FASTMSG として false が設定されているかどうかを確認してください。なお、再実行しても繰り返し発生する場合は、DABroker の動作環境が破壊されていますので、DABroker を組み込み直してください。

KFDB008MM-E

```
An error occurred in a system call during output of XXXX. [YYYY errno=ZZZZ]
```

XXXX：メッセージログファイル名称

YYYY：エラー発生関数名称

ZZZZ：エラー番号

要因

メッセージログ出力中にシステムコールエラーが発生しました。

処理

処理を続行します。メッセージを syslog に出力します。

対処

使用する OS のマニュアルを参照しエラー原因を取り除いてから、再実行してください。なお、代表的なシステムコールのエラー番号については、「[7.5.3 システムコールのエラー番号](#)」を参照してください。

KFDB009MM-E

```
メモリを確保できませんでした。XXXX  
Required memory cannot be obtained. XXXX
```

XXXX：確保しようとしたメモリのサイズ，又はメモリ不足でエラーとなったシステムコール名称

要因

DABroker の動作に必要なメモリが不足しています。

処理

DABroker を終了します。

対処

しばらく時間を空けてから、再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB010MM-E

```
接続先データベース定義に誤りがあります。LINE=XXXX [YYYY]  
The connection database definition is incorrect. LINE=XXXX [YYYY]
```

XXXX：エラー発生行

YYYY：エラー発生定義

要因

- 接続先データベース定義ファイルの構文に誤りがあります。
- 接続先データベース定義ファイルに設定した値に誤りがあります。

処理

処理を終了します。

対処

テキストエディタを使用して、接続先データベース定義ファイルを修正してください。

KFDB011MM-E

```
動作環境定義ファイルに対するアクセス権限がありません。[XXXX]  
The user does not have authority to access the environment definition file. [XXXX]
```

XXXX：DABroker 動作環境定義ファイル名称

要因

DABroker 動作環境定義ファイルに対するアクセス権がありません。

処理

処理を終了します。

対処

DABroker 動作環境定義ファイルに読み込み権限があるかどうか調べてください。読み込み権限がなければ、読み込み権限を付加してください。

KFDB012MM-E

接続先データベース定義ファイルに対するアクセス権がありません。 [XXXX]

DABroker does not have authority to access the connection database definition file. [XXXX]

XXXX：接続先データベース定義ファイル名称

要因

接続先データベース定義ファイルに対するアクセス権がありません。

処理

処理を終了します。

対処

接続先データベース定義ファイルに読み込み権限があるかどうか調べてください。読み込み権限がなければ、読み込み権限を付加してください。

KFDB013MM-E

データベース種別名が重複しています。LINE=XXXX [YYYY]

The database-type-name is duplicated. LINE=XXXX [YYYY]

XXXX：エラー発生行

YYYY：エラー発生ブロック名称

要因

接続先データベース定義ファイル中のデータベース種別名が重複しています。

処理

処理を終了します。

対処

テキストエディタを使用して、接続先データベース定義ファイルの値を修正してください。

KFDB014MM-E

```
データベース名が重複しています。LINE=XXXX [YYYY]  
The database name is duplicated. LINE=XXXX [YYYY]
```

XXXX：エラー発生行

YYYY：エラー発生ブロック名称

要因

接続先データベース定義ファイル中のデータベース名が重複しています。

処理

処理を終了します。

対処

テキストエディタを使用して、接続先データベース定義ファイルの値を修正してください。

KFDB015MM-E

```
XXXX を定義していません。[YYYY]  
The [XXXX] was not found. [YYYY]
```

XXXX：必須定義名称

YYYY：定義ブロック名称

要因

接続先データベース定義ファイルのブロック中に必須定義がありません。

処理

処理を終了します。

対処

テキストエディタを使用して、接続先データベース定義ファイルの値を修正してください。

KFDB016MM-E

```
コネクションマルチ定義に誤りがあります。LINE=XXXX [YYYY]  
The Connection-Multi definition is incorrect. LINE=XXXX [YYYY]
```

XXXX:エラー発生行

YYYY:エラー発生定義

要因

1. コネクションマルチ定義ファイルの構文に誤りがあります。

2. コネクションマルチ定義ファイルに設定した定義値に誤りがあります。

処理

処理を終了します。

対処

テキストエディタなどを使用して、コネクションマルチ定義ファイルを修正してください。

KFDB017MM-E

```
コネクションマルチ定義ファイルに対するアクセス権限がありません。 [XXXX]
The user does not have authority to access the Connection-Multi definition file. [XXXX]
```

XXXX：コネクションマルチ定義ファイル名称

要因

コネクションマルチ定義ファイルに対するアクセス権限がありません。

処理

処理を終了します。

対処

DABroker コネクションマルチ定義ファイルに対するアクセス権限があるかどうか調べてください。アクセス権が無ければ、アクセス権限を付加してください。

なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB018MM-E

```
コネクションマルチ名が重複しています。LINE=XXXX [YYYY]
The Connection-Multi-name is duplicated. LINE=XXXX [YYYY]
```

XXXX:エラー発生行

YYYY:エラー発生ブロック名称

要因

コネクションマルチ定義ファイル中のコネクションマルチ名が重複しています。

処理

処理を終了します。

対処

テキストエディタなどを使用して、コネクションマルチ定義ファイルの値を修正してください。

KFDB019MM-E

```
ディレクトリが存在しません。ディフォルトディレクトリを使用します。 [XXXX]
```

The directory was not found. DABroker will use the default directory. [XXXX]

XXXX：環境変数名

要因

1. DABroker 動作環境定義ファイルで環境変数に設定したディレクトリが存在しないか、使用できません。
2. ユーザが環境変数に設定したディレクトリが存在しないか、使用できません。

処理

環境変数に指定したディレクトリを無視して、デフォルトのディレクトリを使用します。

対処

1. 環境変数に設定したディレクトリを作成してください。
2. テキストエディタなどを使用して、DABroker 動作環境定義ファイルの環境変数の値を修正してください。
3. ユーザが設定した環境変数の値を修正してください。
4. DABroker 環境設定ユーティリティを使用して、正しいディレクトリを指定してください。

KFDB020MM-E

ディレクトリが存在しません。 [XXXX]

The directory was not found. [XXXX]

XXXX：環境変数名

要因

1. DABroker 動作環境定義ファイルで環境変数に設定したディレクトリが存在しないか、使用できません。
2. ユーザが環境変数に設定したディレクトリが存在しないか、使用できません。

処理

処理を終了します。

対策

1. 環境変数に設定したディレクトリを作成してください。
2. テキストエディタなどを使用して、DABroker 動作環境定義ファイルの環境変数の値を修正してください。
3. ユーザが設定した環境変数の値を修正してください。

KFDB001NM-E

システムコールでエラーが発生しました。(エラー関数：XXXX, エラーコード：YYYY)

An error occurred in a system call. (Error function:XXXX, error code:YYYY)

XXXX：エラー発生関数名

YYYY：GetLastError 関数で返ってきたエラーコード

要因

システムコールでエラーが発生しました。

処理

処理を終了します。

対処

Windows をシャットダウンさせて再実行してください。なお、このエラーが繰り返し発生する場合は、保守員に連絡してください。

KFDB002NM-E

DABroker の実行中に異常が発生しました。DABroker を終了します。(ファイル名：XXXX, 行番号：YYYY)

An abnormality occurred during DABroker processing. DABroker terminates.
(File name:XXXX, Line number:YYYY)

XXXX：エラーが発生したソースファイル

YYYY：エラーが発生した行番号

要因

DABroker 実行中に異常が発生しました。

処理

処理を終了します。

対処

保守員に連絡してください。

KFDB003NM-E

DABroker の実行中にメモリ不足が発生しました。

Memory became insufficient during DABroker processing.

要因

DABroker の実行中にメモリ不足が発生しました。

処理

処理を終了します。

対処

ほかのアプリケーションプログラムを終了して、再実行してください。

KFDB004NM-E

動作環境が不正です。DABroker をインストールし直してください。
The environment is incorrect. Re-install DABroker.

要因

次の原因が考えられます。

- インストールが正しく完了していない。
- 環境変数（レジストリ）が不正である。
- DABroker のファイルがないか、又は破壊されている。

処理

処理を終了します。

対処

イベントログ（アプリケーション）にほかのメッセージが出力されている場合は、出力されているエラーメッセージを参照してください。

DABroker をインストールし直して、再実行してください。

KFDB005NM-E

DABroker の初期化中にエラーが発生しました。
An error occurred during DABroker initialization processing.

要因

DABroker の初期化処理でエラーが発生しました。

処理

処理を終了します。

対処

1. イベントログ（アプリケーション）にほかのメッセージが出力されている場合は、出力されているエラーメッセージを参照してください。
2. ログファイル(dabsv.log)にほかのメッセージが出力されている場合は、出力されているエラーメッセージを参照してください。
3. 定義エラーに対処できない場合は、該当する定義ファイルを削除してください。

KFDB101NM-E

サービス制御処理中にエラーが発生しました。（エラーコード 1：XXXX，エラーコード 2：YYYY）

An error occurred during service controlled processing. (Error code 1:XXXX, error code 2:YYYY)

XXXX：エラー発生関数種別

YYYY：GetLastError で返ってきたエラーコード

要因

サービス制御処理中にエラーが発生しました。

処理

処理を終了します。

対処

1. イベントログ (アプリケーション)、又はメッセージログにほかのメッセージが出力されている場合は、出力されているエラーメッセージを参照してください。
2. 定義エラーに対処できない場合は、該当する定義ファイルを削除してください。
3. DABroker をインストールし直して、再実行してください。
4. 1 から 3 の方法で対処できない場合は、Windows をシャットダウンさせて再実行してください。
なお、このエラーが繰り返し発生する場合は、保守員に連絡してください。

KFDB102NM-E

サービス制御処理中にメモリ不足が発生しました。

Memory became insufficient during service controlled processing.

要因

サービス制御処理中にメモリ不足が発生しました。

処理

DABroker のサービス制御処理を終了します。

対処

ほかのアプリケーションプログラムの終了などによって、動作環境のメモリを確保してから、再実行してください。

KFDB103NM-I

DABroker サービスを開始しました。

DABroker service has started.

要因

DABroker サービスを開始し、DABroker を起動しました。

処理

DABroker サービスを起動しました。

KFDB104NM-Q

DABroker サービスを停止してもよろしいですか？

Do you want to terminate the DABroker service?

要因

DABroker サービスを停止し、DABroker を終了しようとしてしました。

処理

ユーザの応答を待ちます。

対処

DABroker を停止する場合は [はい] を、停止しない場合は [いいえ] ボタンを選択してください。

KFDB105NM-E

DABroker サービスの情報が誤っているため起動できません。(エラーコード：XXXX)

DABroker cannot start because a DABroker service parameter is invalid. (Error code:XXXX)

XXXX：GetLastError 関数で返ってきたエラーコード

要因

DABroker サービスを開始しようとしてしましたが、DABroker サービスの情報が誤っていたためエラーが発生しました。

処理

処理を終了します。

対処

DABroker をインストールし直してください。その後、再実行してください。

KFDB151NM-E

サービス制御処理中にエラーが発生しました。(エラーコード 1：XXXX, エラーコード 2：YYYY)

An error occurred during service controlled processing. (Error code 1:XXXX, error code 2:YYYY)

XXXX：エラー発生関数種別

YYYY：GetLastError で返ってきたエラーコード

要因

DABroker コネクションマルチ機能のサービス制御処理中にエラーが発生しました。

処理

処理を終了します。

対処

1. イベントログ (アプリケーション), 又はメッセージログにほかのメッセージが出力されている場合は, 出力されているエラーメッセージを参照してください。
2. 定義エラーを対処できない場合は, 該当する定義ファイルを削除してください。
3. DABroker をインストールし直して, 再実行してください。
4. 1 から 3 の方法で対処できない場合は, Windows をシャットダウンさせて再実行してください。
なお, このエラーが繰り返し発生する場合は, 保守員に連絡してください。

KFDB152NM-E

サービス制御処理中にメモリ不足が発生しました。

Memory became insufficient during service controlled processing.

要因

DABroker コネクションマルチ機能のサービス制御処理中にメモリ不足が発生しました。

処理

DABroker コネクションマルチ機能のサービス制御処理を終了します。

対処

ほかのアプリケーションプログラムの終了などによって, 動作環境のメモリを確保してから, 再実行してください。

KFDB153NM-I

DABroker コネクションマルチ機能サービスを開始しました。

The DABroker Connection-Multi function service has started.

要因

DABroker コネクションマルチ機能サービスを開始し, DABroker コネクションマルチ機能を起動しました。

処理

DABroker コネクションマルチ機能サービスを起動しました。

KFDB154NM-Q

DABroker コネクションマルチ機能サービスを停止してもよろしいですか?

Do you want to terminate the DABroker Connection-Multi function service?

要因

DABroker コネクションマルチ機能サービスを停止し、DABroker コネクションマルチ機能を終了しようとした。

処理

ユーザの応答を待ちます。

対処

DABroker コネクションマルチ機能を停止する場合は [はい] を、停止しない場合は [いいえ] ボタンを選択してください。

KFDB155NM-E

DABroker コネクションマルチ機能サービスの情報が誤っているため起動できません。(エラーコード:XXXX)

The DABroker Connection-Multi function cannot start because a DABroker service parameter is invalid. (Error code:XXXX)

XXXX: GetLastError 関数で返ってきたエラーコード

要因

DABroker サービスを開始しようとしたが、DABroker サービスの情報が誤っていたためエラーが発生しました。

処理

処理を終了します。

対処

DABroker をインストールし直してください。その後、再実行してください。

KFDB201NM-E

ディレクトリが正しくありません。

The directory is invalid.

要因

次の原因が考えられます。

- ディレクトリ名の指定に誤りがある
- 存在しないディレクトリが指定されている

処理

処理を終了します。

対処

ディレクトリ名を正しく指定して、再実行してください。

KFDB202NM-E

範囲外の数値データが入力されました。
The entered numerical data is outside the range.

要因

次の原因が考えられます。

- 数値データ以外の文字が入力された
- 全角文字が入力された
- 指定できる範囲外の数値データ入力された

処理

処理を終了します。

対処

指定できるデータの範囲を確認してから、データを再入力してください。

KFDB203NM-Q

クライアントの処理を中断します。IP アドレス：XXXX， ホスト名：YYYY
The client processing was canceled. IP address:XXXX host name:YYYY

XXXX：中断するクライアント処理の IP アドレス

YYYY：中断するクライアント処理のホスト名

要因

クライアントプロセスを管理ユーティリティで中断しようとした。

処理

ユーザの応答を待ちます。

対処

クライアントプロセスを中断する場合は [はい] を、中断しない場合は [いいえ] ボタンを選択してください。

KFDB204NM-E

レジストリ名 XXXX の値が長過ぎます。
The registry name XXXX is too long.

XXXX：レジストリの値の名前

要因

レジストリ名 XXXX の値の文字列長が 1,024 バイトを超えました。

処理

処理を終了します。

対処

レジストリエディタで、レジストリ名 XXXX の値の文字列を 1,024 バイト以下に設定してください。

KFDB205NM-E

レジストリ名 XXXX で、DABroker がサポートしているデータタイプと異なるデータタイプが取得されました。

DABroker found a non-supported data type in the registry name XXXX.

XXXX：レジストリの値の名前

要因

レジストリ名 XXXX に DABroker がサポートしているデータタイプと異なるデータタイプが指定されました。

処理

処理を終了します。

対処

レジストリエディタでデータを修正してください。

KFDB206NM-E

レジストリ名 XXXX の値の取得に失敗しました。(エラー関数：YYYY, エラーコード：ZZZZ)

DABroker failed to get the value of registry name XXXX.

(Error function:YYYY, error code:ZZZZ)

XXXX：レジストリの値の名前

YYYY：エラー関数名

ZZZZ：GetLastError 関数に返されたエラーコード

要因

次の原因が考えられます。

- レジストリが破壊された
- DABroker, 又はユティリティの稼働環境が不安定になった
- DABroker, 又はユティリティが破壊された
- レジストリの読み取り権限がない

処理

処理を終了します。

対処

1. DABroker, 又はユティリティを一度終了して, 操作を再実行してください。
2. レジストリに読み取り権限を付与してください。
3. 1, 及び 2 の方法で対処できない場合は, Windows を再起動するか, 又は DABroker をインストールし直してください。

KFDB207NM-E

```
レジストリ名 XXXX の値の更新に失敗しました。(エラー関数:YYYY, エラーコード:ZZZZ)  
DABroker failed to update the value of registry name XXXX. (Error function:YYYY, error  
code:ZZZZ)
```

XXXX: レジストリの値の名前

YYYY: エラー関数名

ZZZZ: GetLastError 関数に返されたエラーコード

要因

次の原因が考えられます。

- レジストリが破壊された
- DABroker, 又はユティリティの稼働環境が不安定になった
- DABroker, 又はユティリティが破壊された
- レジストリに書き込み権限がない

処理

処理を続行します。

対処

1. DABroker, 又はユティリティを一度終了して, 操作を再実行してください。
2. レジストリに書き込み権限を付与してください。
3. 1, 及び 2 の方法で対処できない場合は, Windows を再起動するか, 又は DABroker をインストールし直してください。

KFDB211NM-E

```
DABroker が起動中のため, 動作環境定義の更新はできません。  
An environment definition cannot be updated because DABroker is running.
```

要因

環境設定ユティリティで, DABroker の動作環境を更新しようとしたますが, DABroker が起動中のためできませんでした。

処理

処理を終了します。

対処

DABroker を終了してから、再実行してください。

KFDB213NM-E

環境設定ユーティリティを実行する権限がありません。

The user does not have authority to execute the environment setting utility.

要因

環境設定ユーティリティを実行したユーザが Administrators グループに属していないため、実行できません。

処理

処理を終了します。

対処

環境設定ユーティリティは、Administrators グループに属しているユーザが実行してください。

KFDB214NM-E

取得する情報の種類が指定されていません。

The type of information to be acquired is not specified.

要因

データベースアクセストレースの取得が指示されたにもかかわらず、取得する情報の種類が一つも指定されていません。

処理

[設定値変更] ダイアログボックスを再表示し、ユーザの操作を待ちます。

対処

取得する情報の種類を一つ以上指定するか、[アクセストレースを取得する] チェックボックスをオフにして、再実行してください。

KFDB215NM-E

DABroker が起動中のため、更新できません。

Cannot update because DABroker is running.

要因

DABroker の定義を更新しようとしたのですが、DABroker が起動中のため更新できませんでした。

処理

処理を終了します。

対処

DABroker を終了してから、再実行してください。

KFDB216NM-W

DABroker が起動中のため、更新できません。
Cannot update because DABroker is running.

要因

DABroker の定義を更新しようとしたが、DABroker が起動中のためできませんでした。

処理

読み取り専用モードで実行します。

対処

DABroker を終了してから、再実行してください。

KFDB217NM-W

DABroker コネクションマルチ機能が起動中のため、更新できません。
Cannot update because the DABroker Connection-Multi function is running.

要因

DABroker の定義を更新しようとしたが、DABroker コネクションマルチ機能が起動中のためできませんでした。

処理

読み取り専用モードで実行します。

対処

DABroker コネクションマルチ機能を終了してから、再実行してください。

KFDB250NM-I

トレースを出力しました。
The trace has been output.

要因

トレース出力ユーティリティで指定したファイルにトレースを出力しました。

処理

処理を続行します。

KFDB251NM-I

出力するトレースがありません。
The trace to be output was not found.

要因

出力を要求された種別のトレースが 1 件も取得されていません。

処理

処理を続行します。

KFDB252NM-E

通信トレースの出力処理中にエラーが発生しました。
An error occurred during output processing for the communication-trace.

要因

次の原因が考えられます。

- 出力先のディスクに十分な容量がない
- 出力先のディスクに書き込む権限がない
- 出力先のディスクが書き込み禁止である
- 出力先のファイル指定に誤りがある

処理

処理を終了します。

対処

出力先のディスク、ファイル、及びファイルの指定を確認し、再実行してください。

KFDB253NM-E

アクセストレースの出力処理中にエラーが発生しました。
An error occurred during output processing for the access-trace.

要因

次の原因が考えられます。

- 出力先のディスクに十分な容量がない
- 出力先のディスクに書き込む権限がない
- 出力先のディスクが書き込み禁止である

処理

処理を終了します。

対処

出力先のディスク, 又はファイルを確認し, 再実行してください。

KFDB302NM-E

接続できないデータベース種別が指定されています。

You specified an invalid database type.

要因

クライアントからデータベースへの接続が要求されましたが, サポートしていないデータベース種別のため, 処理できません。

処理

データベースとの接続を中止し, コネクションを解放します。

対処

接続できるデータベース種別を指定してください。

KFDB002RC-E

コマンドの引数に誤りがあります。使用法:dabdbtrc [-u ユーザ ID] [-p プログラム名]

Invalid argument. Usage:dabdbtrc [-u user-ID][-p program-name]

要因

dabdbtrc コマンドに指定したオプションに誤りがあります。

処理

処理を終了します。

対処

コマンドを正しく指定して, 再実行してください。

KFDB003RC-E

ユーティリティの実行中にメモリ不足が発生しました。

Memory became insufficient during execution of a utility program.

要因

ユーティリティの実行中にメモリ不足が発生しました。

処理

処理を終了します。

対処

しばらく時間を空けてから, 再実行してください。なお, このエラーが繰り返し発生する場合は, システム管理者に連絡してください。

KFDB001RM-E

システムコールでエラーが発生しました。[XXXX()error=YYYY]

An error occurred in a system call. [XXXX() errno=YYYY]

XXXX：システムコール名称

YYYY：エラー番号

要因

OS のシステムコール関数でエラーが発生しました。

処理

処理を終了します。

対処

操作を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB001RR-E

指定されたユーザ ID のデータベースとのアクセストレースは見つかりません。

The access-trace for the database that has the specified user ID could not be found.

要因

指定したユーザ ID の、データベースアクセストレースが見つかりません。

処理

処理を終了します。

対処

指定したユーザ ID を確認してください。

KFDB002RR-E

指定されたプログラム名のデータベースとのアクセストレースは見つかりません。

The access-trace for the database that has the specified program name could not be found.

要因

指定したクライアントのプログラム名の、データベースアクセストレースが見つかりません。

処理

処理を終了します。

対処

指定したクライアントのプログラム名を確認してください。

KFDB003RR-E

指定されたユーザ ID, 又はプログラム名のデータベースとのアクセストレースは見つかりません。
The access-trace for the database that has the specified user ID or program name could not be found.

要因

指定したユーザ ID, 又はクライアントのプログラム名の, データベースアクセストレースが見つかりません。

処理

処理を終了します。

対処

指定したユーザ ID, 又はクライアントのプログラム名を確認してください。

KFDB001SC-E

コマンドの引数に誤りがあります。使用法:dabstat
Invalid argument. Usage:dabstat

要因

dabstat コマンドのオプションに誤りがあります。

処理

処理を終了します。

対処

コマンドを正しく指定して, 再実行してください。

KFDB002SC-E

DABroker は起動していません。
DABroker is not running.

要因

DABroker の動作状態を表示※しようとしたのですが, DABroker は起動していません。

処理

処理を終了します。

注※

Windows 対応の DABroker の場合は, 管理ユーティリティを実行したときです。UNIX 対応の DABroker の場合は, dabstat コマンドを実行したときです。

KFDB003SC-E

DABroker は起動処理中です。コマンドを取り消します。

DABroker is starting. The command is canceled.

要因

dabstat コマンドを実行しましたが、DABroker の起動処理が完了していません。

処理

処理を終了します。

対処

しばらく時間を空けて、再実行してください。

KFDB004SC-E

DABroker は終了処理中です。コマンドを取り消します。

DABroker is terminating. The command is canceled.

要因

dabstat コマンドを実行しましたが、DABroker は終了処理中です。

処理

処理を終了します。

KFDB005SC-I

DABroker は計画終了処理中です。

DABroker is terminating in the planned termination mode.

要因

DABroker の動作状態を表示[※]しようとしたのですが、DABroker は計画終了処理中です。

処理

DABroker の動作状態を表示し、計画終了を続行します。

注※

Windows 対応の DABroker の場合は、管理ユーティリティを実行したときです。UNIX 対応の DABroker の場合は、dabstat コマンドを実行したときです。

KFDB006SC-E

DABroker は強制終了処理中です。コマンドを取り消します。

DABroker is terminating in the forced termination mode. The command is canceled.

要因

dabstat コマンドを実行しましたが、DABroker は強制終了処理中です。

処理

処理を終了します。

KFDB007SC-E

DABroker は起動処理中です。

DABroker is starting.

要因

DABroker 管理ユーティリティを実行しましたが、DABroker の起動処理が完了していません。

処理

処理を終了します。

対処

しばらく時間を空けて、再実行してください。

KFDB008SC-E

DABroker は終了処理中です。

DABroker is terminating.

要因

DABroker 管理ユーティリティを実行しましたが、DABroker は終了処理中です。

処理

処理を終了します。

KFDB009SC-E

DABroker は強制終了処理中です。

DABroker is terminating in the forced termination mode.

要因

DABroker 管理ユーティリティを実行しましたが、DABroker は強制終了処理中です。

処理

処理を終了します。

KFDB101SC-E

コマンドの引数に誤りがあります。使用法：dabcmstat

Invalid argument. Usage:dabcmstat

要因

dabcmstat コマンドのオプションに誤りがあります。

処理

処理を終了します。

対処

コマンドを正しく指定して、再実行してください。

KFDB102SC-E

DABroker コネクションマルチ機能は起動していません。
The DABroker Connection-Multi function is not running.

要因

DABroker コネクションマルチ機能の動作状態を表示※しようとしたますが、DABroker コネクションマルチ機能は起動していません。

処理

処理を終了します。

注※

Windows 対応の DABroker の場合は、プログラムフォルダに登録された[DABroker コネクションマルチ機能管理]を選択したときです。UNIX 対応の DABroker の場合は、dabcmstat コマンドを実行したときです。

KFDB103SC-E

DABroker コネクションマルチ機能は起動処理中です。コマンドを取り消します。
The DABroker Connection-Multi function is starting. The command is canceled.

要因

dabcmstat コマンド実行しましたが、DABroker コネクションマルチ機能の起動処理が完了していません。

処理

処理を終了します。

対処

しばらく時間を空けて、再実行してください。

KFDB104SC-E

DABroker コネクションマルチ機能は終了処理中です。コマンドを取り消します。

The DABroker Connection-Multi function is terminating. The command is canceled.

要因

dabcmstat コマンドを実行しましたが、DABroker コネクションマルチ機能は終了処理中です。

処理

処理を終了します。

KFDB105SC-I

DABroker コネクションマルチ機能は計画終了処理中です。

The DABroker Connection-Multi function is terminating in the planned termination mode.

要因

DABroker コネクションマルチ機能の動作状態を表示[※]しようとしたのですが、DABroker コネクションマルチ機能は計画終了処理中です。

処理

DABroker コネクションマルチ機能の動作状態を表示し、計画終了を続行します。

注※

Windows 対応の DABroker の場合は、プログラムフォルダに登録された[DABroker コネクションマルチ機能管理]を選択したときです。UNIX 対応の DABroker の場合は、dabcmstat コマンドを実行したときです。

KFDB106SC-E

DABroker コネクションマルチ機能は強制終了処理中です。コマンドを取り消します。

The DABroker Connection-Multi function is terminating in the forced termination mode. The command is canceled.

要因

dabcmstat コマンドを実行しましたが、DABroker コネクションマルチ機能は強制終了処理中です。

処理

処理を終了します。

KFDB107SC-E

DABroker コネクションマルチ機能は起動処理中です。

The DABroker Connection-Multi function is starting.

要因

DABroker コネクションマルチ機能管理ユーティリティを実行しましたが、DABroker コネクションマルチ機能の起動処理が完了していません。

処理

処理を終了します。

対処

しばらく時間を空けて、再実行してください。

KFDB108SC-E

DABroker コネクションマルチ機能は終了処理中です。
The DABroker Connection-Multi function is terminating.

要因

DABroker コネクションマルチ機能管理ユーティリティを実行しましたが、DABroker コネクションマルチ機能は終了処理中です。

処理

処理を終了します。

KFDB109SC-E

DABroker コネクションマルチ機能は強制終了処理中です。
The DABroker Connection-Multi function is terminating in the forced termination mode.

要因

DABroker コネクションマルチ機能管理ユーティリティを実行しましたが、DABroker コネクションマルチ機能は強制終了処理中です。

処理

処理を終了します。

KFDB001SM-E

システムコールでエラーが発生しました。[XXXX()errno=YYYY]
An error occurred in a system call. [XXXX() errno=YYYY]

XXXX：システムコール名称

YYYY：エラー番号

要因

OS のシステムコール関数でエラーが発生しました。

処理

処理を終了します。

対処

操作を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB002SM-E

```
共有メモリを取得できませんでした。[shmget()errno=XXXX]  
Shared memory cannot be obtained. [shmget() errno=XXXX]
```

XXXX：エラー番号

要因

OS のシステムコール関数(shmget)でエラーが発生しました。

処理

処理を終了します。

対処

操作を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB003SM-E

```
共有メモリを削除できませんでした。[shmctl()errno=XXXX]  
Shared memory cannot be deleted. [shmctl() errno=XXXX]
```

XXXX：エラー番号

要因

OS のシステムコール関数(shmctl)でエラーが発生しました。

処理

処理を続行します。

対処

このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB004SM-E

```
セマフォを取得できませんでした。[semget()errno=XXXX]  
The semaphore cannot be obtained. [semget() errno=XXXX]
```

XXXX：エラー番号

要因

OS のシステムコール関数(semget)でエラーが発生しました。

処理

処理を終了します。

対処

操作を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB005SM-E

```
セマフォを削除できませんでした。[semctl()errno=XXXX]  
The semaphore cannot be deleted. [semctl() errno=XXXX]
```

XXXX：エラー番号

要因

OS のシステムコール関数(semctl)でエラーが発生しました。

処理

処理を続行します。

対処

このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB006SM-E

```
ユーティリティの実行中にメモリ不足が発生しました。  
Memory is insufficient during execution of a utility program.
```

要因

ユーティリティの実行中にメモリ不足が発生しました。

処理

処理を終了します。

対処

しばらく時間を空けてから、再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB002TA-E

```
エージェント処理の実行中にメモリ不足が発生したため、処理を中止しました。  
Agent processing was canceled because memory became insufficient.
```

要因

エージェント処理の実行中に、メモリ不足が発生しました。

処理

エージェント処理を終了します。

対処

実行中のほかの処理が終了してから、再実行してください。

KFDB006TA-E

```
不正なデータを受信しました。エラーコード= [XXXX], [YYYY]  
Invalid data was received. Error code = [XXXX],[YYYY]
```

XXXX：エラーコード 1(保守用情報)

YYYY：エラーコード 2(保守用情報)

要因

クライアントから不正なデータを受信しました。

処理

クライアントからの要求を受け付けません。

対処

保守員に連絡してください。

KFDB008TA-E

```
クライアントが指定した、エージェント情報の値に誤りがあります。Tag Name=XXXX, 詳細コード=YYYY  
The value in the agent information specified by the client is incorrect.  
Tag Name = XXXX, detail code = YYYY
```

XXXX：エラーコード 1(保守用情報)

YYYY：エラーコード 2(保守用情報)

要因

クライアントから指定されたエージェント情報の値に誤りがあります。

処理

クライアントからの要求を受け付けません。

対処

保守員に連絡してください。

KFDB009TA-E

エージェント処理の実行中に異常が発生しました。ErrKind=KKKK, ErrFileCnt=XXXX, SubRtnCode=YYYY, errno=ZZZZ

An abnormality occurred during agent processing.

ErrKind = KKKK, ErrFileCnt = XXXX, Sub RtnCode = YYYY, errno = ZZZZ

KKKK：エラーコード 1(保守用情報)

XXXX：エラーコード 2(保守用情報)

YYYY：エラーコード 3(保守用情報)

ZZZZ：エラーコード 4(保守用情報)

要因

エージェント処理の実行中に異常が発生しました。

処理

クライアントからの要求を受け付けません。

対処

保守員に連絡してください。

KFDB011TA-E

エージェント名に不正な文字が指定されています。

An invalid character is specified in the agent name.

要因

エージェント名に使用できない文字が使用されています。

処理

クライアントからの要求を受け付けません。

対処

エージェント名を正しく指定してから、再実行してください。

KFDB013TA-E

エージェント実行名に不正な文字が指定されています。

An invalid character is specified in the agent execution name.

要因

エージェント実行名に使用できない文字が使用されています。

処理

クライアントからの要求を受け付けません。

対処

エージェント実行名を正しく指定してから、再実行してください。

KFDB015TA-E

グループ ID に使用できない文字が指定されています。
An invalid character is specified in the group ID.

要因

エージェントグループ ID に使用できない文字が使用されています。

処理

クライアントからの要求を受け付けません。

対処

エージェントグループ ID を正しく指定してから、再実行してください。

KFDB017TA-E

グループ ID のパスワードに使用できない文字が指定されています。
An invalid character is specified in the group ID password.

要因

エージェントグループ ID のパスワードに使用できない文字が使用されています。

処理

クライアントからの要求を受け付けません。

対処

エージェントグループ ID のパスワードを正しく指定してから、再実行してください。

KFDB018TA-E

RDA Link for Gateway のロードモジュールが見つかりません。
There is no load module for RDA Link for Gateway.

要因

RDA Link for Gateway が組み込まれていません。

処理

クライアントからの要求を受け付けません。

対処

RDA Link for Gateway を組み込んでください。

KFDB001TD-E

```
システムコールでエラーが発生しました。[XXXX errno=YYYY]  
An error occurred in a system call. [XXXX errno=YYYY]
```

XXXX：システムコール名称

YYYY：システムコールのエラー番号

要因

XXXX システムコールでエラーが発生しました。

処理

クライアントからの要求を受け付けません。

対処

使用する OS のマニュアルを参照しエラー原因を取り除いてから、再実行してください。なお、代表的なシステムコールのエラー番号については、「7.5.3 システムコールのエラー番号」を参照してください。

KFDB002TD-E

```
メモリ不足が発生しました。  
Memory is insufficient.
```

要因

メモリ不足が発生しました。

処理

クライアントからの要求を受け付けません。

対処

しばらく時間を空けてから、再実行してください。なお、このエラーが繰り返し発生する場合は、実メモリ、又は仮想メモリの設定を見直してください。

KFDB003TD-E

```
指定されたカーソルは使用中、又は初期化処理がされていません。[CursorId=XXXX]  
The specified cursor is being used, or it is not initialized. [CursorId=XXXX]
```

XXXX：カーソル番号

要因

指定されたカーソルは、他の処理で使用中です。または、初期化処理がされていません。

処理

クライアントからの要求を受け付けません。

対処

未使用のカーソルを使用して、再実行してください。または、初期化処理をしてください。

KFDB004TD-E

カーソルの値が不正です。 [CursorId=XXXX]
The cursor value is invalid. [CursorId=XXXX]

XXXX：カーソルの値

要因

指定されたカーソルの値が不正です。

処理

クライアントからの要求を受け付けません。

対処

カーソルは 1 から 64 の値を指定してください。

KFDB005TD-E

不正なデータを受信しました。 [ErrorCode=XXXX]
Invalid data was received. [ErrorCode=XXXX]

XXXX：保守用情報

要因

クライアントから不正なデータを受信しました。

処理

クライアントからの要求を受け付けません。

対処

クライアントのバージョン、又は DABroker のバージョンを確認してください。

KFDB006TD-E

必要なパラメタがありません。または、パラメタの値が不正です。 [XXXX]
The required parameter was not found, or the parameter value is invalid. [XXXX]

XXXX：パラメタ名

USER ID：ユーザ ID

RD NODE NAME：RD ノード名称

TABLE TYPE：表属性

ABSTRACT KIND：データベース種別名

ABSTRACT NAME：データベース名

DATASOURCE：ODBC3.0 のデータソース名

要因

パラメタがありません。または、パラメタの値が不正です。

処理

クライアントからの要求を受け付けません。

対処

パラメタを正しく設定して、再実行してください。

KFDB007TD-E

指定された機能はサポートしていません。[RequestCode=XXXX]

DABroker does not support the specified function. [RequestCode=XXXX]

XXXX：保守用情報

要因

クライアントのバージョンが新しいため、サポートしていない機能を要求しました。又は、クライアントが不正なデータを送信しました。

処理

クライアントからの要求を受け付けません。

対処

クライアントのバージョン、又は DABroker のバージョンを確認してください。なお、クライアントからの要求コードについては、「[7.5.2 要求コード/応答コード](#)」を参照してください。

KFDB008TD-E

システムでエラーが発生しました。[ErrorCode=XXXX]

An error occurred in the system. [ErrorCode=XXXX]

XXXX：保守用情報

要因

DABroker システム内でエラーが発生しました。

処理

クライアントからの要求を受け付けません。

対処

保守員に連絡してください。

KFDB009TD-E

返信データの要求サイズが不正です。 [ResponseSize=XXXX]
The request size of reply data is invalid. [ResponseSize=XXXX]

XXXX：保守用情報

要因

クライアントが要求した返信データのサイズが不正です。

処理

クライアントからの要求を受け付けません。

対処

保守員に連絡してください。

KFDB010TD-E

要求コードのシーケンスエラーが発生しました。 [RequestCode=XXXX-YYYY]
A sequence error of the request code occurred. [RequestCode=XXXX-YYYY]

XXXX：前回の要求コード

YYYY：今回の要求コード

要因

クライアントからの要求コードのシーケンスエラーが発生しました。

処理

クライアントからの要求を受け付けません。

対処

出力された要求コードを基に要求シーケンスを見直してください。なお、要求コードについては、[「7.5.2 要求コード/応答コード」](#)を参照してください。

KFDB011TD-E

指定されたデータベースは未サポートです。
DABroker does not support the specified database.

要因

クライアントのバージョンが新しいため、サポートしていないデータベースを指定しました。または、クライアントが不正なデータを送信しました。

処理

クライアントからの要求を受け付けません。

対処

接続先のデータベースを見直してください。

KFDB012TD-E

指定された SQL に誤りがあります。

The specified SQL is incorrect.

要因

クライアントが指定した SQL に次に示す誤りがあります。

- 「:」の使用方法が不正です。

処理

クライアントからの要求を受け付けません。

対処

SQL の誤りを修正し、再実行してください。

KFDB013TD-E

データベースアクセスライブラリでエラーが発生しました。[ErrorCode=XXXX]

An error occurred in the database access library. [ErrorCode=XXXX]

XXXX : 詳細コード

要因

データベースアクセスライブラリでエラーが発生しました。

処理

クライアントからの要求を受け付けません。

対処

詳細コードを参照しエラー原因を取り除いてから、再実行してください。なお、詳細コードについては、「[7.5.1 DABroker 詳細コード](#)」を参照してください。

KFDB014TD-E

処理中、又は登録中により、エージェントの実行結果データはありません。

The agent execution result data was not found because the agent is executing or registering.

要因

処理中、又は登録中により、エージェントの実行結果データはありません。

処理

クライアントからの要求を受け付けません。

対処

正常終了のエージェントを指定してください。

KFDB015TD-E

タイムオーバーのため処理を中断しました。

Processing was canceled because a timeover occurred.

要因

タイムアウトが発生しました。

処理

クライアントからの要求を受け付けません。

対処

クライアントを再起動してから、操作を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB016TD-E

データベースアクセスでエラーが発生しました。[ErrorCode=XXXX]

A database access error occurred. [ErrorCode=XXXX]

XXXX : DBMS のエラーコード

要因

データベースアクセスでエラーが発生しました。

処理

クライアントからの要求を受け付けません。

対処

DBMS のマニュアルを参照してエラーとなる原因を取り除き、操作を再実行してください。

KFDB017TD-E

指定されたエージェントはありません。

The specified agent was not found.

要因

クライアントが指定したエージェントはありません。

処理

クライアントからの要求を受け付けません。

対処

正しいエージェントを指定して、操作を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB018TD-E

即時配信時のエージェント実行でタイムアウトが発生しました。

A timeout occurred during direct agent execution.

要因

即時配信時のエージェント実行でタイムアウトが発生しました。

処理

クライアントからの要求を受け付けません。

対処

しばらく時間を空けてから再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡し、DABroker 動作環境定義ファイルの「即時配信実行監視時間、DABAGTEXEDIRETIME」の値を大きくしてください。

KFDB020TD-E

データ変換でアンダーフローエラーが発生しました。

An underflow error occurred during data conversion.

要因

データ変換でアンダーフローエラーが発生しました。

処理

エージェント処理を中止します。

対処

エージェントの内容と、DBMS のテーブルの定義情報を確認してください。

KFDB021TD-E

データ変換でオーバフローエラーが発生しました。

An overflow error occurred during data conversion.

要因

データ変換でオーバフローエラーが発生しました。

処理

エージェント処理を中止します。

対処

エージェントの内容と、DBMS のテーブルの定義情報を確認してください。

KFDB022TD-E

データ変換でメモリ不足が発生しました。

Memory became insufficient during data conversion.

要因

データ変換に必要なメモリが不足しているため、データ変換処理ができません。

処理

エージェント処理を中止します。

対処

しばらく時間を空けてから、再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB023TD-E

データ変換で NULL を指定できないデータに NULL 値を入れようとしてしました。

An invalid null value was set in the data during data conversion.

要因

データ変換で NULL を指定できないデータに NULL 値を入れようとしてしました。

処理

エージェント処理を中止します。

対処

エージェントの内容と、DBMS のテーブルの定義情報を確認してください。

KFDB024TD-E

データ変換でデータ不正エラーが発生しました。

An invalid data error occurred during data conversion.

要因

データ変換でデータ不正エラーが発生しました。

処理

エージェント処理を中止します。

対処

エージェントの内容と、DBMS のテーブルの定義情報を確認してください。

KFDB025TD-E

データ変換でパラメタ不正エラーが発生しました。

An invalid parameter error occurred during data conversion.

要因

データ変換でパラメタ不正エラーが発生しました。

処理

エージェント処理を中止します。

対処

保守員に連絡してください。

KFDB026TD-E

データ変換で予期しないエラーが発生しました。

An unexpected error occurred during data conversion.

要因

データ変換で予期しないエラーが発生しました。

処理

エージェント処理を中止します。

対処

保守員に連絡してください。

KFDB027TD-E

指定されたエージェント実行結果を削除する権限がありません。

The user does not have authorization to delete the agent execution result.

要因

指定されたエージェント実行結果を削除する権限がありません。

処理

処理を終了します。

対処

削除オプションの指定を解除した後、再実行してください。

KFDB030TD-W

キャンセルされたため、処理を中断しました。

Processing was canceled.

要因

キャンセルされたため、処理を中断しました。

処理

処理を終了します。

KFDB031TD-W

処理が終了したため、キャンセルできません。

Processing cannot be canceled because it has already ended

要因

処理をキャンセルしようとしたのですが、処理が終了したため、キャンセルできませんでした。

処理

キャンセル処理を終了します。

KFDB032TD-E

クライアントでサポートされていないデータ型があります。(DataType=XXXX)

A data type that the client does not support was found. (DataType=XXXX)

XXXX：データ属性コード

要因

クライアントでサポートされていないデータ型があります。

処理

処理を終了します。

対処

表の定義、又は DABroker のバージョンを見直してください。また、データ属性コードについては、接続先のデータベースのマニュアルを参照してください。

KFDB033TD-E

データベースに接続されていません。

Not connected to the database.

要因

データベースに接続されていません。

処理

処理を終了します。

対処

データベースの接続を行ってから、処理を再実行してください。

KFDB034TD-E

指定されたファイルを削除する権限がありません。

ファイル名：XXXX

The user does not have authorization to delete the specified file. File:XXXX

XXXX：ファイル名

要因

指定されたファイルを削除する権限がありません。

処理

処理を終了します。

対処

ファイルの権限を確認し、再実行してください。

KFDB035TD-E

指定されたファイルがありません。ファイル名：XXXX

The specified file was not found. File:XXXX

XXXX：ファイル名

要因

指定されたファイルが見つかりません。

処理

処理を終了します。

対処

ファイルの名を確認し、再実行してください。

KFDB036TD-E

指定されたファイルの削除に失敗しました。ファイル名：XXXX [unlink () errno = YYYY]

DABroker failed to delete the specified file. File:XXXX [unlink() errno=YYYY]

XXXX：ファイル名

YYYY：unlink()のエラー番号

要因

指定されたファイルの削除に失敗しました。

処理

処理を終了します。

対処

unlink()のエラー番号を参照してエラーとなる原因を取り除いてください。なお、unlink()のエラー番号については、「7.5.3 システムコールのエラー番号」を参照してください。

KFDB037TD-E

クライアントへの情報の送信に使用する領域が不足しました。要因= XXXX

The area for sending information to the client was insufficient. Cause=XXXX

XXXX：処理種別

要因

クライアントへの情報の送信に使用する領域が不足しました。

処理

クライアントからの要求を受け付け、処理を行いました。

対処

クライアントプログラムで設定した受信データ領域長を見直してください。データベースに対して更新、又は表の定義を要求した場合、データベースにアクセスし、処理が正しく実行されているか確認してください。

なお、出力された処理種別によって次の計算式を使用して必要な受信データ領域長を算出し、受信データ領域長を見直してください。ただし、対策後もこのエラーが繰り返し発生する場合、及びこのほかの処理種別が出力された場合は、保守員に連絡してください。

1. 処理種別が EXT-CMD <Prepare> の場合

$$24 * n + \sum_{i=1}^n (\uparrow(\text{検索対象の列名称の長さ} / 4) \uparrow * 4) \text{ バイト} \quad (n: \text{検索する列数})$$

2. 処理種別が EXT-CMD <Fetch> の場合

検索する 1 レコードの長さ + 4 * 検索する列数 バイト

3. 処理種別が PROCEDURE PREPARE の場合

$$52+36 * n + \sum_{i=1}^n (\uparrow (\text{パラメタ列名称の長さ} / 4) \uparrow * 4) \text{ バイト}$$

(n : 入力, 出力, 及び入出力パラメタ数*2の和)

4. 処理種別が PROCEDURE EXECUTE の場合

$$44+20 * n + \sum_{i=1}^n (\uparrow (\text{出力, 又は入出力パラメタの列名称の長さ} / 4) \uparrow * 4) \text{ バイト}$$

(n : 出力, 及び入出力パラメタ数の和)

5. 処理種別が PROCEDURE-PARAMETER の場合

$$52+48 * n + \sum_{i=1}^n (\uparrow (\text{入力, 又は出力, 入出力パラメタの列名称の長さ} / 4) \uparrow * 4) \text{ バイト}$$

(n : 入力, 及び出力, 入出力パラメタ数の和)

6. 処理種別が PROCEDURE-RESULT の場合

$$52+48 * m + \sum_{i=1}^n (\uparrow (\text{入力, 又は出力, 入出力パラメタの列名称の長さ} / 4) \uparrow * 4) + 36 * n +$$

$$\sum_{i=1}^n (\uparrow (\text{ResultSetの列名称の長さ} / 4) \uparrow * 4) \text{ バイト}$$

(m : 入力, 及び出力, 入出力パラメタ数の和 n : ResultSet数)

7. 処理種別が EXT-CMD <EXECUTE> の場合

52+4 * 設定された可変値データのうち実行されたレコード数 バイト

KFDB040TD-E

指定されたデータベース種別定義がありません。データベース種別名：XXXX
The specified database type definition was not found. Database-type-name:XXXX

XXXX：データベース種別名

要因

指定されたデータベース種別名が見つかりません。

処理

クライアントからの要求を受け付けません。

対処

接続先データベース定義で指定したデータベース種別名を確認し、再実行してください。

KFDB041TD-E

指定されたデータベース名定義がありません。データベース種別名：XXXX, データベース名：YYYY
The specified database definition name was not found. Database-type-name:XXXX database-name:YYYY

XXXX：データベース種別名

YYYY：データベース名

要因

指定されたデータベース名が見つかりません。

処理

クライアントからの要求を受け付けません。

対処

接続先データベース定義で指定したデータベース名を確認し、再実行してください。

KFDB042TD-E

指定されたコネクションマルチ定義がありません。データベース種別名：XXXX，データベース名：YYYY

The specified Connection-Multi definition name was not found.

Database-type-name:XXXX database-name:YYYY

XXXX：データベース種別名

YYYY：データベース名

要因

指定されたコネクションマルチ名が見つかりません。

処理

クライアントからの要求を受け付けません。

対処

接続先データベース定義に指定したコネクションマルチ名を確認し、再実行してください。

KFDB043TD-E

指定されたデータベース名定義に分散定義名称がありません。データベース種別名：XXXX，データベース名：YYYY

The distributed-definition-name was not found in the specified database name definition.

Database-type-name:XXXX database-name:YYYY

XXXX：データベース種別名

YYYY：データベース名

要因

指定されたデータベース名定義に分散定義名称が定義されていません。

処理

クライアントからの要求を受け付けません。

対処

接続先データベース定義を確認し、再実行してください。

KFDB044TD-E

指定されたコネクションマルチ定義にホスト名、又は IP アドレスがありません。

データベース種別名：XXXX，データベース名：YYYY

The host name or the IP address was not found in the specified Connection-Multi definition.
Database-type-name:XXXX database-name:YYYY

XXXX：データベース種別名

YYYY：データベース名

要因

指定されたコネクションマルチ定義にホスト名、又は IP アドレスが定義されていません。

処理

クライアントからの要求を受け付けません。

対処

接続先データベース定義、及びコネクションマルチ定義を確認し、再実行してください。

KFDB045TD-E

指定されたコネクションマルチ定義にサービス名、又はポート番号がありません。

データベース種別名：XXXX，データベース名：YYYY

The service name or port number not was found in the specified Connection-Multi definition.
Database-type-name:XXXX database-name:YYYY

XXXX：データベース種別名

YYYY：データベース名

要因

指定されたコネクションマルチ定義にサービス名、又はポート番号が定義されていません。

処理

クライアントからの要求を受け付けません。

対処

接続先データベース定義、及びコネクションマルチ定義を確認し、再実行してください。

KFDB046TD-E

指定されたデータベース名定義にコネクションマルチ名がありません。

データベース種別名：XXXX，データベース名：YYYY

The Connection-Multi name was not found in the specified database name definition.

Database-type-name:XXXX database-name:YYYY

XXXX：データベース種別名

YYYY：データベース名

要因

指定されたデータベース名定義にコネクションマルチ名が定義されていません。

処理

クライアントからの要求を受け付けません。

対処

接続先データベース定義を確認し、再実行してください。

KFDB047TD-E

データベース種別定義に定義したデータベース種別と、クライアントから接続要求したデータベース種別が異なります。

データベース種別名：XXXX，データベース名：YYYY

The specified database type in the database type definition differs from the database type in the connection request from the client. Database-type-name:XXXX database-name:YYYY

XXXX：データベース種別名

YYYY：データベース名

要因

データベース種別定義に定義したデータベース種別とクライアントから接続要求したデータベース種別が異なります。

処理

クライアントからの要求を受け付けません。

対処

接続先データベース定義、及びクライアントの設定を確認し、再実行してください。

KFDB048TD-E

指定されたデータベース名定義にユーザ ID がありません。

データベース種別名：XXXX，データベース名：YYYY

The user ID was not found in the specified database name definition. Database-type-name:XXXX database-name:YYYY

XXXX：データベース種別名

YYYY：データベース名

要因

指定されたデータベース名定義にユーザ ID が定義されていません。

処理

クライアントからの要求を受け付けません。

対処

接続先データベース定義を確認し、再実行してください。

KFDB049TD-E

```
データ処理中にメモリ不足が発生しました。 ErrInfo[XXXX]  
Memory is insufficient during data processing. ErrInfo[XXXX]
```

XXXX：保守用情報

要因

処理に必要な領域を確保しようとした時、メモリ不足が発生しました。

処理

処理を終了します。

対処

しばらく時間を空けてから、再実行してください。なお、このエラーが繰り返し発生する場合は、実メモリ、又は仮想メモリの設定を見直してください。

KFDB001TJ-E

```
JNI function (XXXX) failed. LINE : YYYY
```

XXXX：JNI 関数名称

YYYY：保守用情報

要因

JNI 関数でエラーが発生しました。

処理

当メッセージを Throw します。

対処

保守員に連絡してください。

KFDB002TJ-E

Handle of dayd_Control is NULL. LINE : XXXX

XXXX : 保守用情報

要因

内部関数(dayd_Control())で使うハンドルが NULL です。

処理

当メッセージを Throw します。

対処

保守員に連絡してください。

KFDB003TJ-E

Invalid handle. LINE : XXXX

XXXX : 保守用情報

要因

Java I/F 関数に指定するハンドルに不正があります。

処理

当メッセージを Throw します。

対処

保守員に連絡してください。

KFDB004TJ-E

Local storage allocation failed.

要因

メモリの確保ができません。

処理

当メッセージを Throw します。

対処

しばらく時間を空けてから、再実行してください。なお、このエラーが繰り返し発生する場合は、実メモリ、又は仮想メモリの設定を見直してください。

KFDB005TJ-E

Invalid argument. LINE : XXXX

XXXX：保守用情報

要因

Java I/F 関数に指定する引数に不正があります。

処理

当メッセージを Throw します。

対処

保守員に連絡してください。

KFDB006TJ-E

```
Invalid calling sequence. LINE : XXXX
```

XXXX：保守用情報

要因

Java I/F 関数の呼び出し順序が不正です。

処理

当メッセージを Throw します。

対処

保守員に連絡してください。

KFDB007TJ-E

```
Storage allocation failed.
```

要因

メモリの確保ができないので DBMS 側が処理できません。

処理

当メッセージを Throw します。

対処

メモリの確保ができる状態にしてください。

しばらく時間を空けてから、再実行してください。なお、このエラーが繰り返し発生する場合は、実メモリ、又は仮想メモリの設定を見直してください。

KFDB008TJ-E

```
jbyteArray of send data is NULL. LINE : XXXX
```

XXXX：保守用情報

要因

Java I/F 関数に渡された送信データが NULL です。

処理

当メッセージを Throw します。

対処

保守員に連絡してください。

KFDB009TJ-E

```
Size of receive buffer is under 0. LINE : XXXX
```

XXXX : 保守用情報

要因

Java I/F 関数に渡された受信データバッファサイズが負の数です。

処理

当メッセージを Throw します。

対処

保守員に連絡してください。

KFDB010TJ-E

```
The size of the received data is over the buffer size. LINE:XXXX
```

XXXX : 保守用情報

要因

Java I/F 関数に渡された受信バッファサイズを受信データのサイズが超えました。

処理

当メッセージを Throw します。

対処

保守員に連絡してください。

KFDB011TJ-E

```
jbyteArray of receive buffer is NULL. LINE : XXXX
```

XXXX : 保守用情報

要因

Java I/F 関数に渡された受信バッファアドレスが NULL です。

処理

当メッセージを Throw します。

対処

保守員に連絡してください。

KFDB012TJ-E

```
Invalid response information. LINE : XXXX
```

XXXX : 保守用情報

要因

ハンドル毎に保持している応答情報に不正があります。

処理

当メッセージを Throw します。

対処

保守員に連絡してください。

KFDB013TJ-E

```
Invalid status. LINE : XXXX
```

XXXX : 保守用情報

要因

Java I/F 関数の状態不正です。

処理

当メッセージを Throw します。

対処

保守員に連絡してください。

KFDB014TJ-E

```
Error outbreak. LINE : XXXX
```

XXXX : 保守用情報

要因

Java I/F 関数がエラー発生中です。

処理

当メッセージを Throw します。

対処

保守員に連絡してください。

KFDB015TJ-E

```
Invalid response information status. LINE : XXXX
```

XXXX : 保守用情報

要因

Java I/F 関数でハンドル毎に保持している応答データ情報状態が不正です。

処理

当メッセージを Throw します。

対処

保守員に連絡してください。

KFDB016TJ-E

```
dayd_Control failed. LINE : XXXX
```

XXXX : 保守用情報

要因

dayd_Control 関数が予期しない return 値で失敗しました。

処理

当メッセージを Throw します。

対処

保守員に連絡してください。

KFDB017TJ-E

```
Invalid environment. XXXX
```

XXXX : Java I/F 関数

要因

DABroker の動作環境が不正です。

処理

当メッセージを Throw します。

対処

DABroker の動作環境を見直してください。

KFDB001TM-E

services ファイルに、DABroker のサービス名とポート番号(DABROKER_SV)が登録されていません。
The services name and port number(DABROKER_SV) of DABroker are not listed in the "services" file.

要因

システムの services ファイルに、DABroker のサービス名とポート番号が登録されていません。

処理

処理を終了します。

対処

services ファイルに、次の記述を追加してください。

DABROKER_SV XXXXX/tcp

XXXXX には DABroker がオープンするポート番号を指定します。推奨値は 40179 です。

NIS を使用している場合は、NIS のサーバに追加してください。

KFDB002TM-E

ソケットをオープンできません。[socket() errno=XXXX]
The socket cannot be opened. [socket() errno=XXXX]

XXXX：エラー番号

要因

OS のシステムコール関数(socket)でエラーが発生しました。

処理

処理を終了します。

対処

操作を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB003TM-E

ソケットへのアドレスの割り付けができません。[bind() errno=XXXX]
The address cannot be allocated to the socket. [bind() errno=XXXX]

XXXX：エラー番号

要因

OS のシステムコール関数(bind)でエラーが発生しました。DABroker の強制終了時に、LAN 上に通信データが残った可能性があります。

処理

処理を終了します。

対処

しばらく時間を空けてから、DABrokerを再起動してください。

KFDB004TM-E

```
コネクション要求の受付開始時にエラーが発生しました。[listen() errno=XXXX]  
An error occurred when a requested connection was accepted. [listen() errno=XXXX]
```

XXXX：エラー番号

要因

OSのシステムコール関数(listen)でエラーが発生しました。

処理

処理を終了します。

対処

保守員に連絡してください。

KFDB005TM-E

```
コネクションの確立時にエラーが発生しました。[accept() errno=XXXX]  
An error occurred when a connection was established. [accept() errno=XXXX]
```

XXXX：エラー番号

要因

OSのシステムコール関数(accept)でエラーが発生しました。

処理

処理を終了します。

対処

システムに負荷が掛かっていることが考えられます。時間を空けてから、再度処理を実行してください。なお、システムが出力するメッセージなどで対処できず、このエラーが繰り返し発生する場合は、保守員に連絡してください。

KFDB006TM-I

```
コネクションを確立します。[XXXX]  
A connection is established. [XXXX]
```

XXXX：クライアントのIPアドレス又はホスト名

要因

クライアントとの接続を開始しました。

処理

処理を続行します。

KFDB007TM-I

コネクションを解放します。 [XXXX]

A connection is released. [XXXX]

XXXX：クライアントの IP アドレス又はホスト名

要因

クライアントとの接続を終了しました。

対処

処理を続行します。

KFDB008TM-E

クライアントプログラムのバージョンが違います。

Incorrect version of the client program.

要因

クライアントプログラムは接続できるバージョンではありません。

処理

クライアントからの要求を受け付けません。

対処

クライアントプログラムのバージョンを確認してください。

KFDB009TM-E

メモリ不足が発生したため、データベースに接続できません。 IPADDR=XXXX

Cannot connect the database because of insufficient memory. IPADDR=XXXX

XXXX：IP アドレス又はホスト名 クライアントには「IPADDR=XXXX」は表示されません。

要因

メモリ不足が発生したため、データベースに接続できません。

処理

データベースとの接続を中止し、コネクションを解放します。

対処

DABroker で実行中のほかの処理が終了してから、再実行してください。

KFDB010TM-E

不正なデータを受信しました。エラーコード=XXXX, IPADDR=YYYY
Invalid data was received. Error code=XXXX IPADDR=YYYY

XXXX：保守用情報

YYYY：IP アドレス又はホスト名 クライアントには「IPADDR=YYYY」は表示されません。

要因

クライアントから不正なデータを受信しました。

処理

クライアントからの要求を無視します。または、処理を終了します。

対処

保守員に連絡してください。

KFDB011TM-E

通信処理でエラーが発生しました。エラーコード=XXXX, IPADDR=YYYY
An error occurred during communication processing. Error code=XXXX IPADDR=YYYY

XXXX：保守用情報

YYYY：IP アドレス又はホスト名 クライアントには「IPADDR=YYYY」は表示されません。

要因

データの送信時、通信管理プログラムでエラーが発生しました。

処理

処理を終了します。

対処

システム管理者に連絡してください。

KFDB012TM-E

データの送受信中にシーケンスエラーが発生しました。エラーコード=XXXX IPADDR=YYYY
A sequence error occurred during sending or receiving of data.
Error code=XXXX IPADDR=YYYY

XXXX：保守用情報

YYYY：IP アドレス又はホスト名 クライアントには「IPADDR=YYYY」は表示されません。

要因

クライアントから不正なシーケンスでデータを受信しました。

処理

クライアントからの要求を無視します。または、処理を終了します。

対処

保守員に連絡してください。

KFDB013TM-E

```
DABroker が計画終了中のため、処理要求を受け付けられません。IPADDR=XXXX  
The processing request cannot be accepted because DABroker is terminating in the planned  
termination mode. IPADDR=XXXX
```

XXXX：IP アドレス又はホスト名 クライアントには「IPADDR=XXXX」は表示されません。

要因

データベースと接続しようとしたのですが、DABroker が計画終了中です。計画終了中は、新たなクライアントからの処理要求を受け付けません。

処理

クライアントからの要求を無視します。

対処

DABroker の計画終了が終わってから、DABroker を再起動してください。その後、DABroker 及びデータベースへの接続を再実行してください。

KFDB014TM-I

```
TCP/IP 通信処理スレッドを終了します。要因：[XXXX]，IPADDR=YYYY  
The TCP/IP communication thread is terminated. Cause:[XXXX] IPADDR=YYYY
```

XXXX：終了要因

CANCEL_REQUEST：ユーザキャンセル要求

TERM_REQUEST：強制終了要求

YYYY：IP アドレス又はホスト名。クライアントには「IPADDR=XXXX」は表示されません。

要因

強制終了※¹，又はユーザキャンセル※²要求により TCP/IP 通信用の処理スレッドが終了しました。

処理

トランザクションのロールバック及びデータベースとの切り離しを行い、コネクションを解放します。

対処

システム管理者が強制終了、又はユーザキャンセル要求を行っていないか確認してください。

注※1

Windows 対応の DABroker の場合は、[DABroker 終了] ダイアログボックスで強制終了を選択、又はサービスを停止させたときです。UNIX 対応の DABroker の場合は、-f オプションを指定した dabstop コマンドを実行したときです。

注※2

Windows 対応の DABroker の場合は、[DABroker 管理] ウィンドウで選択されたクライアントの処理を中断したときです。UNIX 対応の DABroker の場合は、-c オプションで指定されたホスト名、又は IP アドレスに対応したクライアントの処理を dabstop コマンドで中断したときです。

KFDB015TM-E

エージェントサービスが開始されていないため、エージェント処理要求を受け付けられません。

IPADDR=XXXX

The agent processing request cannot be accepted because the agent service is not running.

IPADDR=XXXX

XXXX：IP アドレス又はホスト名。クライアントには「IPADDR=XXXX」は表示されません。

要因

エージェント処理を実行しようとしたのですが、エージェントサービスが開始されていないため、クライアントからのエージェント処理要求は受け付けません。

処理

クライアントからの要求を無視します。

対処

DABroker を終了させてください。その後、エージェントサービスが開始される指定※で DABroker を再起動してください。

注※

Windows 対応の DABroker の場合は、環境設定ユーティリティで[エージェント機能]を[使用する]を選択して起動します。UNIX 対応の DABroker の場合は、dabstart コマンドの-a オプションを指定して起動します。

KFDB016TM-E

クライアントからの要求待ちタイムアウトが発生しました。接続を解放します。

IPADDR=XXXX

A timeout occurred during the wait for the request from the client.

DABroker releases the connection. IPADDR=XXXX

XXXX：IP アドレス又はホスト名 クライアントには「IPADDR=XXXX」は表示されません。

要因

DABroker 動作環境定義ファイルで DABCLIENT_TIMEOUT に指定した値以内に、クライアントからの要求がありませんでした。

処理

トランザクションのロールバック及びデータベースとの切り離しを行い、コネクションを解放します。

対処

クライアントを再起動してから、操作を再実行してください。なお、このエラーが繰り返し発生する場合は、DABCLIENT_TIMEOUT の値を大きくしてください。

KFDB017TM-E

データベースアクセス処理終了待ちタイムアウトが発生しました。コネクションを解放します。

IPADDR=XXXX

A timeout occurred during the wait for termination of the database access.

DABroker releases the connection. IPADDR=XXXX

XXXX：IP アドレス又はホスト名 クライアントには「IPADDR=XXXX」は表示されません。

要因

DABroker 動作環境定義ファイルで DABDBA_TIMEOUT に指定した値以内に、データベースアクセス処理が終了しませんでした。

処理

実行中の処理を中断し、コネクションを解放します。

対処

クライアントを再起動してから、操作を再実行してください。なお、このエラーが繰り返し発生する場合は、DABDBA_TIMEOUT の値を大きくしてください。

KFDB018TM-E

コネクションが強制解放されました。IPADDR=XXXX

The connection was forcibly released. IPADDR=XXXX

XXXX：IP アドレス又はホスト名 クライアントには「IPADDR=XXXX」は表示されません。

要因

クライアントシステムが原因で、コネクションを強制解放しました。

処理

トランザクションのロールバック及びデータベースとの切り離しを行い、コネクションを解放します。

対処

クライアントでシステムダウンやタイムアウトなどが発生していないかどうか調査してください。また、ケーブルの接続や電源の状態などを確認してください。

KFDB019TM-E

```
コネクションが強制解放されました。IPADDR=XXXX  
The connection was forcibly released. IPADDR=XXXX
```

XXXX：IP アドレス又はホスト名 クライアントには「IPADDR=XXXX」は表示されません。

要因

TCP/IP 通信時のキープアライブ機能によって、障害が検知されました。

処理

トランザクションのロールバック及びデータベースとの切り離しを行い、コネクションを解放します。

対処

1. クライアントシステム又はクライアント通信システムが障害を起こしていないか確認してください。
2. 回線の接続状態を調査してください。

KFDB020TM-E

```
データベースアクセス処理が異常終了しました。エラーコード=XXXX IPADDR=YYYY  
The database access process terminated abnormally. Error code=XXXX IPADDR=YYYY
```

XXXX：保守用情報

YYYY：IP アドレス又はホスト名 クライアントには「IPADDR=YYYY」は表示されません。

要因

システムに負荷が掛かり過ぎているか、又は他の障害によって、データベースアクセス処理が異常終了しました。

処理

コネクションを解放します。

対処

システムに負荷が掛かり過ぎているか、又は他の障害が発生していないか確認してください。エラーの原因が分からない場合は、保守員に連絡してください。

KFDB021TM-E

```
TCP/IP 通信処理プロセスを終了します。要因：[XXXX]  
The TCP/IP communication process terminated. Cause:[XXXX]
```

XXXX：終了要因（保守用情報）

要因

TCP/IP 通信処理プロセスが異常終了しました。

処理

コネクションを解放します。

以降のクライアントからの要求を受け付けません。

対処

kill コマンドなどを入力していないか、又はシステムに負荷が掛かり過ぎていないか確認してください。
エラーの原因が分からない場合は、保守員に連絡してください。

KFDB022TM-W

```
キャンセル処理要求を受け付けられる状態ではありません。[XXXX] IPADDR=YYYY  
The cancel processing request cannot be accepted. [XXXX] IPADDR=YYYY
```

XXXX：保守用情報

YYYY：IP アドレス又はホスト名 クライアントには「IPADDR=YYYY」は表示されません。

要因

DABroker は、非同期キャンセル要求を受け付けましたが、キャンセル処理を実行できる状態ではありません。

処理

キャンセル処理を実行しませんでした。

対処

必要ならば、再度、キャンセル要求を実行してください。

KFDB023TM-E

```
指定された DBMS に対する環境変数に誤りがあります。[XXXX=YYYY] IPADDR=ZZZZ  
The specified DBMS environment variable is incorrect. [XXXX] IPADDR=YYYY
```

XXXX：環境変数名

YYYY：環境変数の値

ZZZZ：IP アドレス又はホスト名 クライアントには「IPADDR=ZZZZ」は表示されません。

要因

指定された DBMS に対して、動作環境定義ファイルに設定した環境変数の値に誤りがあります。

処理

クライアントからの要求を無視します。

対処

動作環境定義ファイルの環境変数の値を修正してください。

KFDB001TR-E

クライアントとの通信トレースの取得中に、メモリ不足が発生しました。

Memory became insufficient during collection of a trace of communication with a client.

要因

クライアントとの通信トレースの取得に必要なメモリが不足しました。

処理

処理を続行します。ただし、このエラーが発生したクライアントとの通信トレースは取得しません。

対処

このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB002TR-E

クライアントとの通信トレースの取得中に、ディスクの容量不足が発生しました。

Disk space became insufficient during collection of a trace of communication with a client.

要因

クライアントとの通信トレースの取得中に、ディスクの容量不足が発生しました。

処理

処理を続行します。ただし、このエラーが発生したクライアントとの通信トレースは取得しません。

対処

通信トレースを取得するディレクトリ下の不要なファイルを削除してください。

KFDB003TR-E

クライアントとの通信トレースの取得中に、システムコールでエラーが発生しました。

[XXXX()errno=YYYY]

An error occurred in a system call during collection of a trace of communication with a client.

[XXXX() errno=YYYY]

XXXX：システムコール名称

YYYY：エラー番号

要因

OS のシステムコール関数、あるいは DABroker の共通関数でエラーが発生しました。

処理

処理を続行します。ただし、このエラーが発生したクライアントとの通信トレースは取得しません。

対処

このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB001YM-E

```
コマンドの引数に誤りがあります。使用法：dabstart [-t] [-a]
Invalid argument. Usage:dabstart [-t] [-a]
```

要因

dabstart コマンドのオプションに誤りがあります。

処理

DABroker を終了します。

対処

コマンドを正しく指定して、再実行してください。

KFDB003YM-I

```
DABroker の初期化処理が完了しました。
DABroker initialization processing has finished.
```

要因

DABroker の起動要求※を受け付け、初期化処理が完了しました。

処理

DABroker を起動し、クライアントからのアクセス要求の受け付けを開始します。

注※

Windows 対応の DABroker の場合は、[DABroker 起動] アイコンをダブルクリック、又はサービスを開始したときです。UNIX 対応の DABroker の場合は、dabstart コマンドを実行したときです。

KFDB004YM-I

```
DABroker の終了処理を開始しました。mode=NORMAL
DABroker termination processing has started. mode=NORMAL
```

要因

DABroker の通常終了※処理を開始しました。

処理

DABroker を終了します。

注※

Windows 対応の DABroker の場合は、[DABroker 終了] ダイアログボックスで通常終了を選択したときです。UNIX 対応の DABroker の場合は、オプションを指定しないで dabstop コマンドを実行したときです。

KFDB005YM-I

```
DABroker の終了処理を開始しました。mode=FORCED  
DABroker termination processing has started. mode=FORCED
```

要因

DABroker の強制終了※処理を開始しました。

処理

データベースへのアクセスを中断します。実行中の処理のロールバック及びデータベースとの切り離しを行い、コネクションを解放します。

注※

Windows 対応の DABroker の場合は、[DABroker 終了] ダイアログボックスで強制終了を選択、又はサービスの停止を実行したときです。UNIX 対応の DABroker の場合は、-f オプションを指定した dabstop コマンドを実行したときです。

KFDB006YM-I

```
DABroker の終了処理を開始しました。mode=PLANNED  
DABroker termination processing has started. mode=PLANNED
```

要因

DABroker の計画終了※処理を開始しました。

処理

実行中のデータベースへのアクセス処理が終了し、接続中のクライアントがすべて終了した時点で DABroker を終了します。計画終了中は、クライアントからの処理要求や、新たなクライアントからの接続要求は受け付けません。

注※

Windows 対応の DABroker の場合は、[DABroker 終了] ダイアログボックスで計画終了を選択したときです。UNIX 対応の DABroker の場合は、-p オプションを指定した dabstop コマンドを実行したときです。

KFDB007YM-E

```
DABroker は計画終了中です。クライアントは接続できません。
```

DABroker is terminating in the planned termination mode. A client cannot be connected.

要因

DABroker に接続しようとしたのですが、DABroker が計画終了中です。計画終了中は、新たなクライアントからの接続要求は受け付けません。

処理

クライアントからの要求を受け付けません。

対処

DABroker の計画終了が終わってから、DABroker を再起動してください。その後、DABroker への接続を再実行してください。

KFDB008YM-E

接続できる最大クライアント数を超えました。クライアントは接続できません。

[DABMAXUSERS=XXX]

The number of clients exceeds the maximum that can be connected.

A client cannot be connected. [DABMAXUSERS=XXXX]

XXX：接続できる最大クライアント数

要因

DABroker 動作環境定義ファイルで DABMAXUSERS に設定した値を超えて、クライアントからの接続要求が発生しました。

処理

クライアントからの要求を受け付けません。

対処

ほかのクライアントが終了するまで待って、DABroker への接続を再実行してください。または、DABroker 動作環境定義ファイルの DABMAXUSERS の値を大きくしてから、再実行してください。

KFDB009YM-E

DABroker は既に起動しています。

DABroker is already running.

要因

DABroker が既に起動されているマシン上で、DABroker の起動要求※を受け付けました。

処理

処理を終了します。

対処

一つのマシン上では、複数の DABroker は起動できません。

注※

Windows 対応の DABroker の場合は、[DABroker 起動] アイコンをダブルクリック、又はサービスの開始を実行したときです。UNIX 対応の DABroker の場合は、dabstart コマンドを実行したときです。

KFDB010YM-E

DABroker のロードモジュールに実行権限がありません。

ロードモジュール名：[XXXX]

The user does not have authority to execute the load module of DABroker. Name:[XXXX]

XXXX：エラーとなったロードモジュールのファイル名称

要因

DABroker のロードモジュールに実行権限がないため、クライアントからの要求を処理できません。

処理

クライアントからの要求を受け付けません。

対処

DABroker を組み込み直してください。UNIX 対応 DABroker の場合は、その後 dabsetup コマンドで DABroker のセットアップを実行してください。

KFDB011YM-E

シンボリックリンクが多過ぎるため、DABroker を実行できません。

ロードモジュール名：[XXXX]

DABroker cannot start because there are too many symbolic links. Name:[XXXX]

XXXX：エラーとなったロードモジュールのファイル名称

要因

DABroker 組み込み先ディレクトリのシンボリックリンクが多過ぎるため、クライアントからの要求を処理できません。

処理

クライアントからの要求を受け付けません。

対処

データベース接続時に発生した場合は、DBMS ライブラリの組み込み先ディレクトリのシンボリックリンクが多過ぎることが考えられます。DBMS ライブラリの組み込み環境を確認してください。

それ以外の場合は、システム管理者に連絡し、シンボリックリンクの少ないディレクトリに DABroker を組み込み直してもらってください。

KFDB012YM-E

DABroker のロードモジュールがありません。

ロードモジュール名：[XXXX]

A load module of DABroker was not found. Name:[XXXX]

XXXX：エラーとなったロードモジュールのファイル名称

要因

DABroker の動作環境が正しく設定されていないため、クライアントからの要求を処理できません。

処理

クライアントからの要求を受け付けません。

対処

DABroker を組み込み直してください。

データベースのクライアント製品が正しくセットアップされているか確認してください。

UNIX 対応の DABroker の場合は、その後 dabsetup コマンドで DABroker のセットアップを実行してください。

HiRDB と接続する場合は、DABroker と HiRDB の双方が 64 ビット版であること、又は双方が 32 ビット版であることを確認してください。

KFDB013YM-E

メモリ不足のため、DABroker を実行できません。

ロードモジュール名：[XXXX]

DABroker cannot start because of insufficient memory. Name:[XXXX]

XXXX：エラーとなったロードモジュールのファイル名称

要因

DABroker の動作に必要なメモリが不足しています。

処理

クライアントからの要求を受け付けません。

対処

しばらく時間を空けてから、再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB014YM-E

DABroker を起動する権限がありません。

The user does not have authority to execute DABroker.

要因

DABroker を起動しようとしたユーザが、DABroker の管理者ユーザ[※]ではないため、DABroker を起動できません。

処理

処理を終了します。

対処

DABroker の起動は、DABroker の管理ユーザが入力してください。

注※

Windows 対応の DABroker の場合は、Administrators グループに属しているユーザ
UNIX 対応の DABroker の場合は、DABroker の管理ユーザ

KFDB015YM-E

```
TCP/IP 通信処理プロセスが異常終了しました。要因：[XXXX]  
The TCP/IP communication process terminated abnormally. Cause:[XXXX]
```

XXXX：終了要因（保守用情報）

要因

TCP/IP 通信処理プロセスが異常終了しました。

処理

コネクションを解放します。以降のクライアントからの要求を受け付けません。

対処

kill コマンドなどを入力していないか、又はシステムに負荷が掛かり過ぎていないか確認してください。
エラーの原因が分からない場合は、システム管理者に連絡してください。

KFDB016YM-E

```
DABroker が異常終了しました。要因：[XXXX]  
DABroker terminated abnormally. Cause:[XXXX]
```

XXXX：終了要因（保守用情報）

要因

kill コマンドを入力したか、又はシステムに負荷が掛かり過ぎて、DABroker が異常終了しました。

処理

DABroker を終了します。

対処

DABroker を再起動する場合は、強制終了を実行後、再起動してください。
このエラーが繰り返し発生する場合は、システム管理者に連絡してください。UNIX 対応の DABroker の場合、保守員に連絡する場合は、次のファイルを退避しておいてください。

- \$DABROKER/spool 以下のファイル（\$DABROKER：DABroker 運用ディレクトリ）
- システムダンプ（(DABroker を起動したカレントディレクトリ下の core ファイル）

KFDB017YM-E

エージェントサービスを開始できません。
Agent services cannot start.

要因

エージェントサービスの開始に失敗しました。

処理

エージェントサービスを終了します。以後、エージェントサービスを使用できません。

対処

エージェント機能を使用する場合は、DABrokerを終了させ、直前に出力されたメッセージの指示に従ってください。

KFDB018YM-E

エージェント管理プロセスが異常終了しました。要因：XXXX
The agent management process terminated abnormally. Cause:XXXX

XXXX：終了要因（保守用情報）

要因

エージェント管理プロセスが異常終了しました。

処理

処理を続行します。

対処

kill コマンドなどを入力していないか、又はシステムに負荷が掛かり過ぎていないか確認してください。エラーの原因が分からない場合は、保守員に連絡してください。

UNIX 対応の DABroker の場合、保守員に連絡する場合は、次のファイルを退避しておいてください。

- \$DABROKER/spool 以下のファイル（\$DABROKER：DABroker 運用ディレクトリ）
- システムダンプ（DABroker を起動したカレントディレクトリ下の core ファイル）

KFDB019YM-E

実行可能な最大ユーザ処理プロセス数を超えました。クライアントは接続できません。
[DABMAXPROCESS=XXXX]
The number of users exceeds the maximum that can be executed.
A client cannot be connected. [DABMAXPROCESS=XXXX]

XXXX：DABroker が実行できる最大処理プロセス数

要因

DABroker 動作環境定義ファイルで DABMAXPROCESS に設定した値を超えて、クライアントからの接続要求が発生しました。

処理

クライアントからの要求を受け付けません。

対処

他のクライアントが終了するまで待って、DABroker への接続を再実行してください。または、DABroker 動作環境定義ファイルの DABMAXPROCESS の値を大きくしてから、再実行してください。

KFDB020YM-E

DABroker を起動できません。

DABroker cannot start.

要因

DABroker の起動に失敗しました。

処理

DABroker を終了します。

対処

直前に出力されたメッセージの指示に従って対処してください。対処後、DABroker を再起動してください。

KFDB021YM-E

同時実行可能な最大データベースアクセス数を超えました。データベースアクセスできません。

[DABDBANUMBER=XXX]

The number of database accesses exceeds the maximum that can be executed. Database access is not possible. [DABDBANUMBER=XXXX]

XXX：同時実行可能なデータベースアクセス数

要因

DABroker 動作環境定義ファイルで DABDBANUMBER に設定した値を超えて、データベースアクセス要求を受け付けました。

処理

クライアントからの要求を受け付けません。

対処

他のデータベースアクセスが終了するまで待って、再実行してください。または、DABroker 動作環境定義ファイルの DABDBANUMBER の値を大きくしてから、再実行してください。

KFDB022YM-I

DABroker の終了処理が完了しました。
DABroker has terminated.

要因

DABroker が終了しました。

KFDB023YM-E

TCP/IP 通信処理プロセスを起動できません。
The TCP/IP communication process cannot start.

要因

TCP/IP 通信処理プロセスの起動に失敗しました。

処理

DABroker 初期化処理を続行します。

対処

DABroker を終了させ、直前に出力されたメッセージに対する指示に従ってください。

KFDB100YM-E

システムコールでエラーが発生しました。[XXXX() errno=YYYY]
An error occurred in a system call. [XXXX() errno=YYYY]

XXXX：関数名称(保守用情報)

YYYY：エラー番号(保守用情報)

要因

OS のシステムコール関数、又は DABroker の共通関数でエラーが発生しました。

処理

処理を終了します。

対処

操作を再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

KFDB301YM-I

Clean up. {xxx,yyy,zzz}, {sss}

要因

クライアントとの通信サーバデーモンのクリーンアップを実行しました。

処理

クライアントとの通信サーバデーモンのクリーンアップを実行した。

対処

なし

備考

xxx：クリーンアップ前のデーモンを数。

yyy：クリーンアップしたデーモンを数。

zzz：クリーンアップ実行後のデーモンを数。

sss：DABroker の状態。

0：正常実行中。

1：正常終了中。

2：計画終了中。

3：強制終了中。

KFDB302YM-I

```
dabtcpdmn {xxx} start. [yyyy]
```

要因

クライアントとの通信サーバデーモンを起動しました。

処理

クライアントとの通信サーバデーモンを起動した。

対処

なし

備考

xxx：dabtcpdmn の引数。

yyyy：起動したプロセス ID。

KFDB303YM-I

```
dabtcpdmn end. [xxxx]
```

要因

クライアントとの通信サーバデーモンを終了しました。

処理

クライアントとの通信サーバデーモンを終了した。

対処

なし

備考

yyyy：起動したしたプロセス ID。

KFDB310YM-I

```
Thread stack size = xxxx (yyyy)
```

要因

スレッドのスタックサイズを設定した。

処理

スレッドのスタックサイズを xxxx バイトに設定した。

対処

なし

備考

xxx：スレッドのスタックサイズ。

yyyy：環境変数で指定されたスレッドのスタックサイズ。

```
DABP_THREAD_STACK_SIZE is undefined. :
```

環境変数 DABP_THREAD_STACK_SIZE が定義されていない。

```
DABP_THREAD_STACK_SIZE=aaa :
```

aaa：環境変数

DABP_THREAD_STACK_SIZE

に指定されている値。

(最大 55 バイト)

KFDB0101YM-E

```
メモリを確保できません。
```

```
Required memory cannot be obtained.
```

要因

DABroker の動作に必要なメモリが不足しています。

処理

DABroker を終了します。

対処

しばらく時間を空けてから、再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

7.4 エラーコードとメッセージの対応

DABroker 関連製品で開発するアプリケーションに、DABroker のメッセージ ID と対応するエラーコードを返す場合があります。アプリケーションに返すエラーコードと DABroker のメッセージ ID の対応を次に示します。メッセージの詳細は、「7.3 メッセージ一覧」を参照してください。

KFDBxxxzy-E → aabbxxx

aa

メッセージ ID の y の桁を表 7-2 の変換表により変換した y の変換結果の値です。

bb

メッセージ ID の z の桁を表 7-2 の変換表により変換した z の変換結果の値です。

xxx

メッセージ ID の xxx

(例)

KFDB001DB-E → 402001

表 7-2 メッセージ ID の変換表

メッセージ ID のコード	y の変換結果	z の変換結果
A	1	01
B	2	02
C	3	03
D	4	04
E	5	05
F	6	06
G	7	07
H	8	08
I	9	09
J	10	10
K	11	11
L	12	12
M	13	13
N	14	14
O	15	15
P	16	16

メッセージIDのコード	yの変換結果	zの変換結果
Q	17	17
R	18	18
S	19	19
T	20	20
U	21	21
V	22	22
W	23	23
X	24	24
Y	25	25
Z	26	26

7.5 メッセージ中に出力されるコード

ここでは、メッセージ中に出力される詳細コードなどについて説明します。

7.5.1 DABroker 詳細コード

DABroker が出力するエラーメッセージの詳細コードを表 7-3 に示します

表 7-3 DABroker 詳細コード

詳細コード	要因	対策
-12001	本タイミングでは CANCEL 要求は受け付けられません。	CANCEL 要求が受け付けられるタイミングになってから、要求してください。CANCEL 要求が受け付けられるのは、データベース接続完了から、データベース切り離し要求受け付けまでです。
-16002	本バージョン、又は DBMS でサポートされていない要求を受け付けました。	DABroker、又は接続先の DBMS を確認してください。
-16004	DABroker が動作できる環境ではありません。	DABroker をインストールした環境を見直して、エラー要因を取り除いてください。
その他負のコード	DABroker システムで異常が発生しました。	保守員に連絡してください。
その他正のコード	DABroker でエラーが発生しました。	コードの詳細については、「 7.4 エラーコードとメッセージの対応 」を参照し、対策してください。

7.5.2 要求コード/応答コード

クライアントからの要求コードと応答コードを表 7-4 に示します。

表 7-4 要求コードと応答コード

要求コード/応答コード	種別
1001	コネクション解放
1002	データベース接続
1003	データベース切り離し
1004	Commit
1005	Rollback
1006	表一覧取得準備
1007	表一覧取得
1008	表一覧取得後処理

要求コード/応答コード	種別
1009	列一覧取得準備
1010	列一覧取得
1011	列一覧取得後処理
1012	索引一覧取得準備
1013	索引一覧取得
1014	索引一覧取得後処理
1015	表権限一覧取得準備
1016	表権限一覧取得
1017	表権限一覧取得後処理
1018	列権限一覧取得準備
1019	列権限一覧取得
1020	列権限一覧取得後処理
1021	SQL 実行準備
1022	SQL 実行
1023	SQL 実行結果取得
1024	SQL 実行後処理
1035	SQL 実行結果取得オプション
1036	BLOB ファイル削除
1037	プライマリキー一覧取得準備
1038	プライマリキー一覧取得
1039	プライマリキー一覧取得後処理
1040	BLOB, CLOB, 及び BFILE 分割取得
1041	自動 Commit 設定
1042	自動 Commit 解除
1050	XA トランザクション制御
1501	ストアプロシジャー一覧取得準備
1502	ストアプロシジャー一覧取得
1503	ストアプロシジャー一覧取得後処理
1504	ストアプロシジャー列一覧取得準備
1505	ストアプロシジャー列一覧取得
1506	ストアプロシジャー列一覧取得後処理

要求コード/応答コード	種別
1507	ストアドプロシジャパラメタ定義情報取得
1508	ストアドプロシジャ実行準備
1509	ストアドプロシジャ実行
1510	ストアドプロシジャ拡張実行
2001	正常応答
2002	異常応答

7.5.3 システムコールのエラー番号

代表的なシステムコールのエラー番号に対する原因と対策を表 7-5 に示します。表 7-5 にないエラーについては、使用している OS のマニュアルを参照してください。

ここでは、主なシステムコールについてだけ記述しています。さらに、詳細に調べる場合は、メッセージで表示されたシステムコールのエラー番号の値に該当するニモニックを、使用している OS の `errno` 定義ファイルで調べてください。そして、該当するニモニックのエラーが返った原因を、OS のマニュアルで特定してください。

表 7-5 システムコールのエラー番号に対する原因と対策

システムコール	errno	ニモニック	考えられる主な原因	対策
<code>dlopen</code>	2	<code>ENOENT</code>	DABroker のライブラリ格納ディレクトリの設定が漏れている可能性があります。	DABroker のライブラリ格納ディレクトリの設定を確認してください。
<code>dup</code>	23	<code>ENFILE</code>	オープンできるファイルの最大数に達しました。	不要なファイルをクローズしてください。
	24	<code>EMFILE</code>	オープンできるファイルの最大数に達しました。	不要なファイルをクローズしてください。
<code>fcntl</code>	23	<code>ENFILE</code>	オープンできるファイルの最大数に達しました。	不要なファイルをクローズしてください。
	24	<code>EMFILE</code>	オープンできるファイルの最大数に達しました。	不要なファイルをクローズしてください。
<code>fork</code>	11	<code>EAGAIN</code>	プロセスの数が多過ぎるか、一時的なメモリ不足が発生しています。	再度実行しても繰り返し発生する場合は、不要なプロセスを停止させてください。
	12	<code>ENOMEM</code>	次のどちらかの要因が考えられます。 <ul style="list-style-type: none"> 仮想メモリ不足のため、プロセスを新しく生成できません。 	仮想メモリが足りない場合は、拡張してください。これらを拡張できない場合は、不要なプロセスを停止させてください。

システムコール	errno	二モニック	考えられる主な原因	対策
			<ul style="list-style-type: none"> プロセスの数が多過ぎるか、又は一部のプロセスが大量のメモリを消費しています。 	一部のプロセスが大量のメモリを消費している場合は、該当するプロセスをいったん停止できないか検討してください。
malloc	12	ENOMEM	<p>次のどちらかの要因が考えられます。</p> <ul style="list-style-type: none"> 仮想メモリ不足のため、プロセスを新しく生成できません。 プロセスの数が多過ぎるか、又は一部のプロセスが大量のメモリを消費しています。 	<p>仮想メモリが足りない場合は、拡張してください。これらを拡張できない場合は、不要なプロセスを停止させてください。</p> <p>一部のプロセスが大量のメモリを消費している場合は、該当するプロセスをいったん停止できないか検討してください。</p>
mkdir	13	EACCESS	アクセス権限が不正です。	DABroker 運用ディレクトリ下の、ディレクトリに対するアクセス権限が正しいか確認してください。
	28	ENOSPC	ファイルシステムに十分な領域がありません。	ファイルシステム内の不要なファイルを削除してください。
open	2	ENOENT	ファイル、又はディレクトリが見つかりません。	環境変数が正しく設定されているかどうかを見直してください。
	13	EACCES	ファイルに対するアクセス権がありません。	ファイルに対するアクセス権を確認してください。
	17	EEXIST	作成しようとしたファイルは既に存在します。	ファイル名を変更して再度実行するか、既存のファイルが不要であれば、削除してから再度実行してください。
	23	ENFILE	オープンできるファイルの最大数に達しました。	不要なファイルをクローズしてください。
	24	EMFILE	オープンできるファイルの最大数に達しました。	不要なファイルをクローズしてください。
semget	28	ENOSPC	セマフォ識別子を生成しようとしたが、システムの上限值を超えました。	すべてのプログラムの所要量を見積り、設定値を変更した後に、マシンを再起動してください。
semop	22	EINVAL	DABroker 稼働中に、DABroker のインストールディレクトリ下の作業ファイルを削除した可能性があります。	DABroker を強制終了してください。その後、DABroker のサービスを再起動してください。
shl_load	8	ENOEXEC	DBMS の設定が誤っている可能性があります。	DBMS の設定を確認してください。
shmat	22	EINVAL	DABroker 稼働中に、DABroker のインストールディレクトリ下の作業ファイルを削除した可能性が	DABroker が起動完了していない場合は、DABroker の起動が完了してからコマンドを実行してくだ

システムコール	errno	ニモニック	考えられる主な原因	対策
			あります。または、DABrokerの起動が完了していない状態でコマンドを実行したことが考えられます。	さい。DABrokerが異常終了してなければ、強制終了してください。その後、DABrokerのサービスを再起動してください。
shmget	22	EINVAL	DABroker稼働中に、DABrokerのインストールディレクトリ下の作業ファイルを削除した可能性があります。	DABrokerを強制終了してください。その後、DABrokerのサービスを再起動してください。
	28	ENOSPC	ディスクの容量が不足しています。	ディスクに、共用メモリサイズ以上の空き容量を確保してください。
write	5	EIO	入出力エラーが発生しました。	OSやハードウェアの情報に従って対策してください。
	14	EFAULT	アクセスできない領域に書き込みしようとした。書き込みしようとしたディスクが切り離された可能性があります。	保守員に連絡してください。
	27	EFBIG	ファイルの大きさがシステムの制限値を超えました。	DABroker管理ユーザでのファイルサイズの制限値をDABrokerで使用するファイルより大きくするか、無制限に変更してください。または、使用するファイルサイズを小さくしてください。
	28	ENOSPC	ファイルシステムに十分な領域がありません。	ファイルシステム内の不要なファイルを削除するか、領域に余裕のあるファイルシステムに移行してください。
chmod, closecreat, dup, execv, fork, gethostbyaddr, gethostbyname, lseek, lockf, open, read, select, semctl, semget, semop, shmat, shmctl,	22	EINVAL	メモリ不足が考えられます。	システムの仮想メモリの初期サイズと最大サイズを同じ値（固定値）にしてください。

システムコール	errno	ニモニック	考えられる主な原因	対策
shmdt, shmget, signal, sigsetmask, stat, write など				

7.5.4 X A 要求時のエラー要因

X A 要求時のエラー要因を表 7-6 に示します。

表 7-6 X A 要求時のエラー要因

項番	設定値	要因	対処
1	LoadLibrary(pdtxa_switch_y) errno=XXXX	dlopen, LoadLibrary, 又は shl_findsym でエラーが発生しました。 XXXX : dlopen, LoadLibrary, 又は shl_findsym のエラー番号	使用する OS のマニュアルを参照し、エラー原因を取り除いてから、再実行してください。
2	GetProcAddress(pdtxa_switch_y) errno=XXXX	dlsym, 又は GetProcAddress でエラーが発生しました。 XXXX : dlsym, 又は GetProcAddress のエラー番号	使用する OS のマニュアルを参照し、エラー原因を取り除いてから、再実行してください。
3	XXXX rtncode=YYYY	XA 関連要求でエラーが発生しました。 XXXX : "xa_open" "xa_close" "xa_start" "xa_end" "xa_prepare" "xa_commit" "xa_rollback" "xa_forget" "xa_recover" YYYY : リターン値	接続している DBMS のマニュアルを参照し、エラー原因を取り除いてから、再実行してください。
4	LoadLibrary(xaosw) errno=XXXX	dlopen, LoadLibrary, 又は shl_findsym でエラーが発生しました。 XXXX : dlopen, LoadLibrary, 又は shl_findsym のエラー番号	使用する OS のマニュアルを参照し、エラー原因を取り除いてから、再実行してください。
5	GetProcAddress(xaosw) errno=XXXX	dlsym, 又は GetProcAddress でエラーが発生しました。 XXXX : dlsym, 又は GetProcAddress のエラー番号	使用する OS のマニュアルを参照し、エラー原因を取り除いてから、再実行してください。

付録

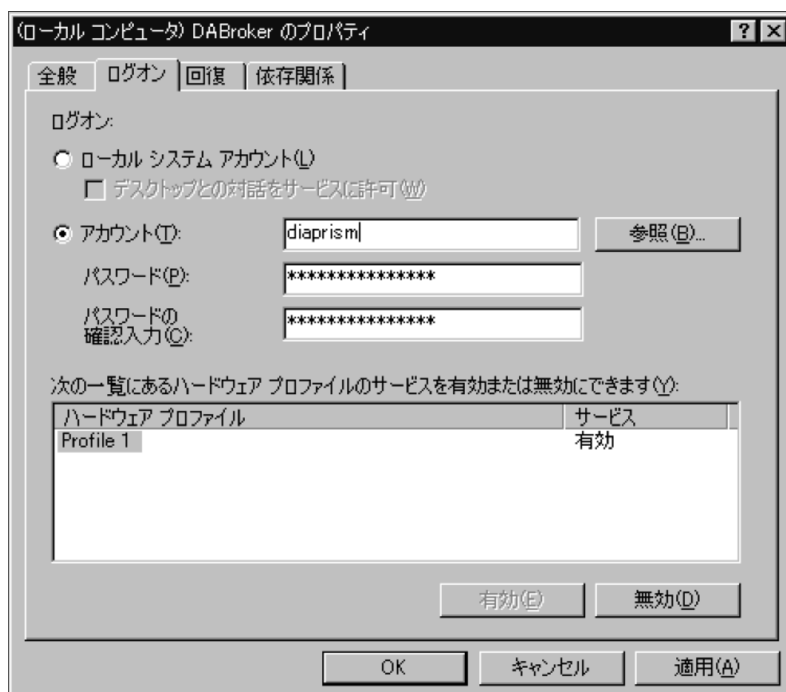
付録 A DIAPRISM アクセス時の環境設定方法 (Windows)

ここでは、HITSENER5 から DIAPRISM へアクセスする場合、DABroker の環境設定方法について説明します。

付録 A.1 DABroker の環境設定

HITSENER5 から DIAPRISM へアクセスする場合、DABroker のサービスにログオンアカウントを設定する必要があります。設定するアカウントは、DIAPRISM のログオンアカウントで Administrator 権限を持つユーザである必要があります。

[コントロールパネル] ダイアログボックスで [管理ツール] を選択し、さらに [サービス] を選択します。[サービス] ダイアログボックスで [DABroker] を選択後、[操作] メニューの [プロパティ] を選択すると、[DABroker のプロパティ] ダイアログボックスが表示されます。



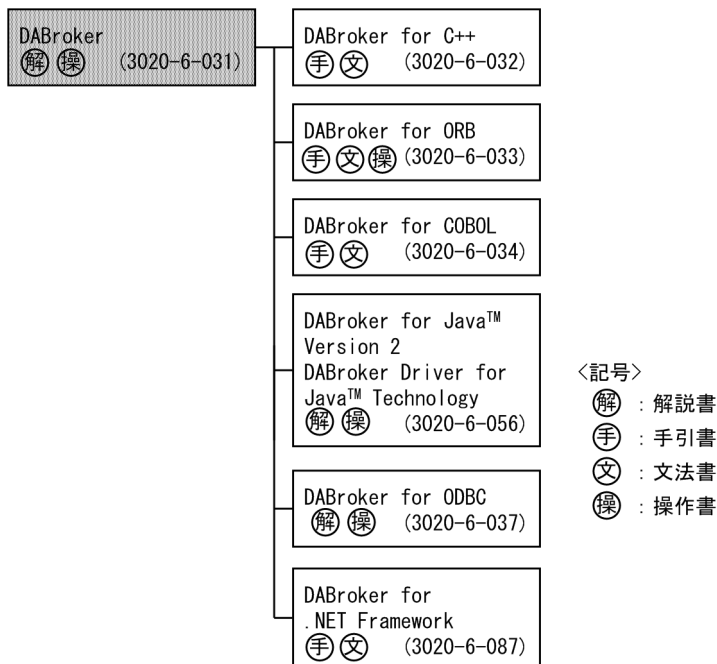
[ログオン] の [アカウント] を選択し、[アカウント]、[パスワード]、[パスワードの確認入力] を入力し、[適用] ボタンをクリックします。その後、[OK] ボタンをクリックして設定を終了します。

付録 B このマニュアルの参考情報

このマニュアルを読むに当たっての参考情報を示します。

付録 B.1 関連マニュアル

このマニュアルの関連マニュアルを次に示します。必要に応じてお読みください。



そのほか、このマニュアルに関連するマニュアルを次に示します。必要に応じてお読みください。

- ・「DBPARTNER/Client」(3020-3-714)
- ・「DBPARTNER/Client」(3020-3-743)
- ・「DBPARTNER2 Client DBPARTNER/Client32 操作ガイド」(3020-6-026)
- ・「DBPARTNER2 Client 操作ガイド」(3020-6-027)
- ・「DBPARTNER2 Client 操作ガイド」(3020-6-151)
- ・「DBPARTNER2 Client(64) 操作ガイド」(3020-6-153)

クライアントからのデータベースのデータを使用する場合に参照してください。なお、DBPARTNER/Client の形名に応じたマニュアルを参照してください。

- ・「DBPARTNER/Client プログラマーズガイド」(3020-3-715)
- ・「DBPARTNER/Client プログラマーズガイド」(3020-3-744)
- ・「DBPARTNER2 Client プログラマーズガイド」(3020-6-028)
- ・「DBPARTNER2 Client プログラマーズガイド」(3020-6-152)

クライアントからのデータベースのデータを使用する場合に参照してください。なお、DBPARTNER/Client の形名に応じたマニュアルを参照してください。

- 「DBPARTNER Mobile」 (3020-3-716)

モバイル環境下でデータベースのデータを使用する場合に参照してください。

- 「DBPARTNER/Web」 (3000-6-041)
- 「DBPARTNER2 Web」 (3020-6-029)

インターネット又はイントラネット環境下で、WWW ブラウザからデータベースのデータを使用する場合に参照してください。

- 「HITSENER5」 (3020-6-070)

データベースのデータを使用して、クライアント/サーバ環境で多次元分析 (OLAP) をする場合に参照してください。

- 「HITSENER5 Web」 (3020-6-080)

データベースのデータを使用して、WWW ブラウザから多次元分析 (OLAP) をする場合に参照してください。

- 「スケーラブルデータベースサーバ HiRDB Version 7 メッセージ (UNIX(R)/Windows(R)用)」 (3000-6-278)
- 「スケーラブルデータベースサーバ HiRDB Version 8 メッセージ」 (3020-6-358)
- 「HiRDB Version 9 メッセージ」 (3020-6-458)
- 「HiRDB Version 10 メッセージ」 (3020-6-562)

HiRDB からのエラーメッセージが出力された場合に参照してください。なお、HiRDB の形名に応じたマニュアルを参照してください。

- 「スケーラブルデータベースサーバ HiRDB Version 7 UAP 開発ガイド (UNIX(R)/Windows(R)用)」 (3000-6-276)
- 「スケーラブルデータベースサーバ HiRDB Version 8 UAP 開発ガイド」 (3020-6-356)
- 「HiRDB Version 9 UAP 開発ガイド」 (3020-6-456)
- 「HiRDB Version 10 UAP 開発ガイド」 (3020-6-560)

HiRDB の環境変数を設定する場合に参照してください。なお、HiRDB の形名に応じたマニュアルを参照してください。

- 「スケーラブルデータベースサーバ HiRDB Version 7 SQL リファレンス (UNIX(R)/Windows(R)用)」 (3000-6-277)
- 「スケーラブルデータベースサーバ HiRDB Version 8 SQL リファレンス」 (3020-6-357)

- 「HiRDB Version 9 SQL リファレンス」(3020-6-457)
- 「HiRDB Version 10 SQL リファレンス」(3020-6-561)

HiRDB とのアクセストレースの RETCODE の内容を知る場合に参照してください。なお、HiRDB の形名に応じたマニュアルを参照してください。

- 「リモートデータベースアクセス RDA Link for Gateway」(3000-6-042)

RDA Link for Gateway の機能概要、及び環境設定方法などを知る場合に参照してください。

- 「Database Connection Server」(6190-6-648)
- 「Database Connection Server」(650-3-243)
- 「Database Connection Server」(6150-6-802)

Database Connection Server の機能概要、及び環境設定方法などを知る場合に参照してください。

- 「データマネジメントシステム XDM E2 系 XDM/RD E2 解説」(6190-6-637)
- 「データマネジメントシステム XDM E2 系 プログラム作成の手引 (XDM/RD E2)」(6190-6-638)
- 「データマネジメントシステム XDM E2 系 XDM/RD E2 SQL 入門」(6190-6-655)
- 「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」(6190-6-656)
- 「データマネジメントシステム XDM E2 系 XDM/RD E2 性能設計」(6190-6-657)
- 「データマネジメントシステム XDM E2 系 ユティリティ (XDM/RD E2 第1分冊)」(6190-6-647)
- 「データマネジメントシステム XDM E2 系 ユティリティ (XDM/RD E2 第2分冊)」(6190-6-639)
- 「データマネジメントシステム XDM E2 系 ユティリティ (XDM/RD E2 第3分冊)」(6190-6-659)
- 「データマネジメントシステム XDM E2 系 メッセージ (XDM/RD E2)」(6190-6-643)
- 「データマネジメントシステム XDM E2 系 XDM/RD E2 使用の手引—システム作成・維持編—」(6190-6-644)
- 「データマネジメントシステム XDM E2 系 XDM/RD IDP 支援 E2 使用の手引」(6180-6-632)
- 「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—」(6190-6-660)
- 「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—」(6190-6-661)

VOS3 XDM/RD を使用する場合は運用方法などを知る場合に参照してください。

- 「データマネジメントシステム XDM E2 系 プログラム作成の手引 (XDM/SD)」(6190-6-652)
- 「エンドユーザ言語 ACE3 E3 操作編」(6190-6-705)

- 「エンドユーザ言語 ACE3 E3 定義編」(6190-6-706)

VOS3 XDM/SD を使用する場合は運用方法などを知る場合に参照してください。

- 「データマネジメントシステム RDB1 E2 解説」(6150-6-560)
- 「データマネジメントシステム RDB1 E2 使用の手引」(6150-6-561)
- 「データマネジメントシステム RDB1 E2 EQL 文法」(6150-6-562)

VOS1 RDB1 E2 を使用する場合は運用方法などを知る場合に参照してください。

- 「データベースマネジメントシステム PDM II E2 使用の手引」(6150-6-514)

VOS1 PDM II E2 を使用する場合は運用方法などを知る場合に参照してください。

- 「VOSK DBPARTNER Library」(650-3-240)

VOSK DBPARTNER Library を使用する場合は運用方法などを知る場合に参照してください。

- 「SQL/K」(650-3-241)

VOSK RDB 編成ファイル (VOSK SQL/K) を使用する場合は運用方法などを知る場合に参照してください。

- 「トランザクショナル分散オブジェクト基盤 TPBroker ユーザーズガイド」(3000-3-660)
- 「TPBroker Developer's Kit - Extension for C++ プログラマーズガイド」(3000-3-680)
- 「トランザクショナル分散オブジェクト基盤 TPBroker Object Transaction Monitor ユーザーズガイド」(3000-3-689)

TPBroker の機能概要、環境設定方法などを知る場合に参照してください。

付録 B.2 このマニュアルでの表記

このマニュアルでは、製品名称、及び名称について次のように表記します。

製品名	表記	
AIX 5L	AIX	UNIX
AIX V6.1		
AIX V7.1		
HP-UX 11.0	HP-UX	
HP-UX 11i		
HP-UX 11i V2 (PA-RISC)		

製品名	表記
HP-UX 11i V3 (PA-RISC)	
Solaris 2.6	Solaris
Solaris 7	
Solaris 8	
Solaris 9	
Red Hat Enterprise Linux AS3.0	Red Hat Linux
Red Hat Enterprise Linux ES3.0	
Red Hat Enterprise Linux AS4.0	
Red Hat Enterprise Linux ES4.0	
Red Hat Enterprise Linux 5	
Red Hat Enterprise Linux 5 Advanced Platform	
Red Hat Enterprise Linux Server 6(64-Bit x86_64)	
Red Hat Enterprise Linux Server 7(64-Bit x86_64)	
Red Hat Enterprise Linux Server 8(64-Bit x86_64)	
Application Programming Interface	
Common Object Request Broker Architecture	CORBA
DABroker	DABroker
DABroker for Java	DABroker for Java
DABroker for Java Version 2	
DABroker Driver for Java Technology	
DABroker for Java Version 2	DABroker for Java Version 2
DABroker Driver for Java Technology	
DABroker for Java Version 2 Extension	Extension
DABroker Driver Optional Package for Java Technology	
Graphical User Interface	GUI
HiRDB Version 7	HiRDB
HiRDB Version 8	
HiRDB Version 9	
HiRDB Version 10	
HITSENER5 Standard for Cosmicube	HITSENER5
HITSENER5 Professional for Cosmicube	

製品名	表記
HITSENER5 Standard for RDB	
HITSENER5 Professional for RDB	
HITSENER5 Standard	
HITSENER5 Professional	
On-Line Analytical Processing	OLAP
Open Database Connectivity	ODBC
Oracle8	ORACLE
Oracle8i	
Oracle9i	
Oracle10g	
Oracle11g	
Oracle12c	
Oracle19c	
Sybase Adaptive Server Anywhere	
Sybase SQL Anywhere	SQL Anywhere
User Application Program	UAP
VOS3 XDM/RD E2	VOS3 XDM/RD
VOS3 XDM/SD E2	VOS3 XDM/SD
VOSK RDB 編成ファイル	VOSK SQL/K
World Wide Web	WWW

このマニュアルでは、DBPARTNER 関連の製品名称について、次のように表記します。

名称	マニュアルでの表記	
DBPARTNER/Client	DBPARTNER/Client	DBPARTNER
DBPARTNER2 Client		
DBPARTNER2 Client(64)		
DBPARTNER/Client32		
DBPARTNER/Web	DBPARTNER/Web	
DBPARTNER2 Web		
DBPARTNER/Web for Windows NT		

名称	マニュアルでの表記		
DBPARTNER Mobile - Server for Windows NT	DBPARTNER Mobile - Server	DBPARTNER Mobile	
DBPARTNER Mobile - Client	DBPARTNER Mobile - Client		

特に、それぞれのプログラムについての記述が必要な場合は、そのまま表記します。

付録 B.3 KB (キロバイト) などの単位表記について

1KB (キロバイト), 1MB (メガバイト), 1GB (ギガバイト), 1TB (テラバイト) はそれぞれ $1,024$ バイト, $1,024^2$ バイト, $1,024^3$ バイト, $1,024^4$ バイトです。

(英字)

API (Application Programming Interface)

このマニュアルでは、DABroker 関連製品で提供するインタフェースを指します。

BLOB

バイナリ長大オブジェクト(Binary Large Object)を指します。LOB(Large Object)データの一種で、画像、音声などのマルチメディアデータ、UDT などの長大なバイナリデータの固まりです。このマニュアルでは、ORACLE の LONG 系バイナリ型、ODBC の長大バイナリ型も BLOB と呼んでいます。

CORBA (Common Object Request Broker Architecture)

分散オブジェクト環境での、分散オブジェクト間の通信などを規定したインタフェースのことです。OMG (Object Management Group) によって提唱されたインタフェースです。CORBA には、主に次の機能があります。

- ORB (Object Request Broker) 機能
分散オブジェクト環境でのオブジェクトの管理、ネットワーク上のオブジェクトの相互作用 (オブジェクト間の通信など) の管理をします。
- IIOP (Internet Inter-ORB Protocol)
インターネット上での、分散オブジェクト環境の通信プロトコルです。
- OTS (Object Transaction Service) 機能
分散オブジェクト環境でのトランザクションを制御します。

Database Connection Server

メインフレーム系データベースにアクセスする場合に必要なプログラムです。Database Connection Server を使うと、クライアントからサーバ上の DABroker を介して、メインフレーム系データベース (VOS3 XDM/RD, VOS3 XDM/SD, VOSK SQL/K, VOS1 PDMII E2, VOSK スプールファイル) にアクセスできます。

DBMS (Data Base Management System)

データベースを管理するソフトウェアです。このマニュアルでは、HiRDB, ORACLE, SQL Anywhere, Adaptive Server Anywhere, SQL Server, 又はメインフレーム系データベース (VOS3 XDM/RD, VOS1 RDB1 E2, VOSK SQL/K, VOS3 XDM/SD, VOS1 PDMII E2, VOSK スプールファイル, VOS1 RDB1 E2) を指します。

DBPARTNER/Client

パーソナルコンピュータに組み込んで使用し、ワークステーションに組み込んだ DABroker とともにクライアント/サーバシステムを構成するプログラムです。データベースのデータの抽出、更新、及びデータベースへの表定義を行う GUI プログラムです。

DBPARTNER/Web

インターネット又はイントラネット上で DABroker を介してデータベースにアクセスするために必要なプログラムです。

DBPARTNER Mobile

モバイル環境下で DABroker を介してデータベースにアクセスするために必要なプログラムです。このマニュアルでは、DBPARTNER Mobile-Client、及び DBPARTNER Mobile-Server を総称して DBPARTNER Mobile と呼びます。

DTP モデル (Distributed Transaction Processing)

オープンシステムの標準化を目的とした団体である X/Open が規定する、分散処理システムモデルのことです。DTP モデルは、トランザクション処理を管理、実行するトランザクションマネージャ (TM: Transaction Manager)、各種資源を管理するリソースマネージャ (RM: Resource Manager)、及び業務処理をする UAP (AP: Application Program) から構成されます。

HITSENER5 for Cosmicube

Cosmicube で構築した多次元データベースのデータを検索し、分析ができる MOLAP ツールです。

HITSENER5 for RDB

リレーショナルデータベースで構築したデータベースのデータを検索し、分析ができる ROLAP ツールです。

HITSENER5 Professional

HITSENER5 動作環境などを管理するシステム管理者が使います。「マネージャ」と「アナライザ」の二つが含まれています。「マネージャ」は、HITSENER5 の動作環境の設定、ユーザ管理、分析情報ファイルの管理などをするためのツールです。「アナライザ」は、データベースから表やグラフを作成して、分析作業をするためのツールです。

HITSENER5 Standard

HITSENER5 を使って多次元分析をする分析者、及び分析者が登録した分析状態を参照するユーザが使います。「アナライザ」だけが含まれています。「アナライザ」は、データベースから表やグラフを作成して、分析作業をするためのツールです。

IP アドレス

データを転送する経路を確立するためのアドレスです。

JDBC インタフェース

Java の標準的なデータベースアクセスインタフェースのことです。

JDBC ドライバ

DABroker for Java で提供するクラスライブラリです。JDBC インタフェースに従って開発したプログラムからの要求を DABroker に転送します。常に、JDBC インタフェースで開発したプログラムと同一マシンに組み込んで使用します。

Mobile Message Queue

DBPARTNER Mobile 同様、モバイル環境下で DABroker を介したデータベースにアクセスするために必要なプログラムです。DBPARTNER Mobile と連携し、DBPARTNER Mobile から実行したデータの受け渡しをキュー機能を使って実現しています。

ODBC

Open DataBase Connectivity の略で、米国 Microsoft Corp. が提唱するデータベースアクセス機構です。ODBC が規定する API を使ったアプリケーションからは、OS、DBMS、ネットワークなどの違いを意識せずにデータベースにアクセスできます。アクセスする相手の DBMS も、ODBC ドライバと呼ばれるソフトウェアドライバに対応している必要があります。

OTS

CORBA 上のオブジェクトに対して、トランザクション処理をするサービスです。

RDA Link for Gateway

分散データベースにアクセスする場合に必要なプログラムです。RDA Link for Gateway を使うと、クライアントから DABroker を介して、メインフレーム系データベース (VOS3 XDM/RD, VOS1 RDB1 E2* VOSK SQL/K) にアクセスできます。

注※ HP-UX の場合、VOS1 RDB1 E2 は使用できません。

Simple Interface

DABroker for ORB で提供する、複雑なデータベースのアクセスを七種類の関数に統合した CORBA オブジェクトのことです。

TCP/IP (Transmission Control Protocol/Internet Protocol)

米国国防省 (DOD) のネットワーク (ARPANET) 用に開発された通信プロトコル (ネットワーク上で通信を行う場合の規約) です。クライアントとのデータ交換は、TCP/IP のライブラリを用いて行われます。

TPBroker

CORBA インタフェースに対応し、分散オブジェクト環境でのオブジェクトの管理、オブジェクト間の通信の管理 (ORB 機能)、及びトランザクションの制御 (OTS 機能) をするためのプログラムです。DABroker では、DABroker for ORB で提供するオブジェクトを呼び出すために TPBroker を利用します。

Web Page Generator

CORBA インタフェースを含む HTML ファイルを動的に生成するプログラムです。CORBA インタフェースを記述できるため、分散オブジェクトに対応します。

X/Open DTP モデル XA インタフェース

X/Open XA インタフェースは、分散トランザクション処理 (DTP : Distributed Transaction Processing) システムで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) の接続インタフェースを規定した X/Open の標準仕様です。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。

(ア行)

アクセストレース

DABroker からデータベースにアクセスした内容についての情報です。Windows の場合は [DABroker トレース出力] ユティリティを、UNIX の場合は dabdbtrc コマンドを使用して取得できます。アクセストレースの種類は任意に選択できます。

アプリケーション

このマニュアルでは、DABroker 関連製品で提供するインタフェースを使用して開発するユーザアプリケーションプログラム (UAP) のことです。

エージェント

データベースアクセス処理の実行時間、アクセス条件などの定型の処理手順を登録しておき、必要に応じて実行できる機能をエージェント機能といいます。ネットワーク環境に関係することなく処理を実行できます。

(カ行)

コネクションの解放

コネクションの解放では、TCP/IP 通信メインプロセス又はスレッドを終了し、クライアントとの通信路を解放します。コネクションが解放されるのは、次のような場合です。

- クライアントからのコネクション解放要求時

- サーバの強制終了時
- サーバの異常終了時
- 通信エラー発生時
- タイムアウト発生時
- DBMS の異常終了時

コネクションの確立

コネクションの確立では、クライアントとの通信路を確立し、TCP/IP 通信メインプロセス又はスレッドを生成、起動します。コネクションが確立されるのは、クライアントからデータベースの最初のアクセスが発生した時です。

コミット

トランザクションによって発生した更新をデータベースに反映することです。

(サ行)

サーバ

使用する DBMS 及び DABroker を組み込んだマシンを指します。ネットワーク上では、データベースにアクセスし、クライアントから要求された処理を実行する側です。

(ナ行)

ネイティブライブラリ

DABroker が提供するライブラリです。サーバで動作する Java Application から DABroker に接続するときには使用します。ネイティブライブラリは、Java の標準インタフェースである JNI (Java Native Interface) を使用し、JNI の引数にプロトコルデータを渡して、Java Application と DABroker を接続します。

(ハ行)

分散オブジェクト技術

複数の分散オブジェクトを組み合わせてシステムを構築する技術のことです。

この技術を利用すると、既存システムに付加価値サービスを追加した新規システムを、既存システムを変更することなく構築できます。分散オブジェクト間のインタフェースには、CORBA インタフェースを使用します。

- 分散オブジェクト

分散オブジェクトとは、ネットワーク上に分散していて、ネットワーク上のどこからでも利用できるオブジェクトのことです。なお、オブジェクトとは、データ、アプリケーション、プロセスなどを部品化したものです。

- **分散オブジェクト環境**

分散オブジェクト環境は、複数のオブジェクト（分散オブジェクト）を組み合わせで構築する環境のことです。

(マ行)

モバイル環境

ノート型パーソナルコンピュータなどから公衆回線網を使って、移動先から LAN 内の各種サーバにあるデータを利用する環境のことです。

(ラ行)

ロールバック

トランザクションによって発生したデータベースの更新を無効にすることです。

索引

記号

- [DABroker 動作環境設定] ダイアログボックス 184
- [アクセストレース設定] ダイアログボックス 314

数字

- 1 プロセスあたりの実行ユーザ数 89, 195

A

- API 549
- Applet 33
- ASP ファイル 39
- ATTACH 318
- ATTACH 情報出力要求 293
- ATTRGET 318
- ATTRGET 情報出力要求 293
- ATTRSET 319
- ATTRSET 情報出力要求 293
- AUTO COMMIT OFF 327
- AUTO COMMIT ON 328
- AUTO COMMIT ON 出力要求 301

B

- BIND 319
- BIND 情報出力要求 294
- BLOB 549

C

- CANCEL 314
- CANCEL 情報出力要求 292
- CHARFORM 319
- CHARFORM 情報出力要求 294
- CHARID 320
- CHARID 情報出力要求 294
- CLOSE 314
- CLOSE 情報出力要求 292
- COMMIT 315
- COMMIT 情報出力要求 290

- CONNECT 315
- CONNECT 情報出力要求 289
- CORBA 549

D

- DAB_DB_HOST 99
- DAB_DB_PORT 99
- DAB_DB_RECEIVE_BUFFER_SIZE 100
- DAB_DB_SEND_BUFFER_SIZE 99
- DAB_HIRDB_DBINF_ESC 85
- DAB_LANG 78
- DABAGT 101, 200
- DABAGT_LOG_ERROR 105
- DABAGT_LOG_INFO 104
- DABAGT_LOG_SIZE 104
- DABAGTEXEDIRETIME 106
- dabagtexeinfo 129
- DABAGTFILMAX 103
- DABAGTGRP 105
- dabagtreginfo 128
- DABAGTREGMAX 103
- DABAGTTIMMAX 103
- DABAGTWCHALDAY 105
- DABAGTWCHALTIME 106
- DABBATCH 89, 192
- DABCLIENT_TIMEOUT 86
- dabcltrc 124
- dabcmstart 139
- dabcmstat 140
- dabcmstop 140
- DABCOBOL_EVTTRC_LEVEL 95
- DABCOBOL_EVTTRC_nn 95
- DABCOBOL_EVTTRC_SIZE 94
- DABCPP_EVTTRC_LEVEL 93
- DABCPP_EVTTRC_nn 94
- DABCPP_EVTTRC_SIZE 93
- DABDBA_TIMEOUT 87

DABDBANUMBER 88
DABDBSDBA_BLOBBUFSIZE 91
dabdbtrc 303
DABEXSQL_TRC_LINE 84, 191
DABHIRDBA_BLOBBUFSIZE 91
DABIPCONVERT 92
DABKEEPALIVE 87
DABMAXBLOCKUSERS 89
DABMAXPROCESS 89
DABMAXUSERS 88
DABORACLE_TYPE 82
DABORADBA_LONGBUFSIZE 90
DABORB_EVTTRC_LEVEL 96
DABORB_EVTTRC_nn 97
DABORB_EVTTRC_SIZE 96
DABroker 17
DABroker for .NET Framework で開発するアプリケーション 41
DABroker for C++で開発するアプリケーション 30
DABroker for C++のイベントトレース取得サイズ 93
DABroker for C++のイベントトレース取得種別 94
DABroker for C++のイベントトレース取得レベル 93
DABroker for COBOL で開発するアプリケーション 31
DABroker for COBOL のイベントトレース取得サイズ 94
DABroker for COBOL のイベントトレース取得種別 95
DABroker for COBOL のイベントトレース取得レベル 95
DABroker for Java で開発するアプリケーション 32
DABroker for ODBC で開発するアプリケーション 38
DABroker for ORB で開発するアプリケーション 37
DABroker for ORB のイベントトレース取得サイズ 96
DABroker for ORB のイベントトレース取得種別 97
DABroker for ORB のイベントトレース取得レベル 96
DABroker 運用ディレクトリ 67
DABroker 環境設定 184
DABroker 管理ユーティリティの実行 233
DABroker 関連製品を使用する場合の環境設定 63, 177
DABroker コネクションマルチ定義 131, 221
DABroker 作業ディレクトリ 79
DABroker 接続先データベース定義 213
DABroker 接続先データベースの定義 110
DABroker セットアップコマンド 69
DABroker 動作環境定義ファイル 74
DABroker 動作環境定義ファイルの作成 74
DABroker との接続 60
DABroker トレース出力ユーティリティの実行 235
DABroker のインストール 66, 180
DABroker のインストールとセットアップ 66
DABroker の管理ユーザ 68
DABroker の起動と終了 119, 230
DABroker の終了 120
DABroker の処理内容の表示 123
DABroker のセットアップ 67
DABroker の動作環境の設定 74
DABroker の動作環境の設定項目 76
DABroker のログ取得ファイルサイズ 83
DABroker をインストールするマシン 66, 180
DABroker を運用する前に 253
DABSELECTCOLCNT 80
DABSELECTINFCNT 81
dabsetup 69
DABSQL_HIRDB_DICTIONARY_DATATYPE 83
DABSQL_TRC_ATTACH 293
DABSQL_TRC_ATTRGET 293
DABSQL_TRC_ATTRSET 293
DABSQL_TRC_AUTO_COMMIT_OFF 301
DABSQL_TRC_AUTO_COMMIT_ON 301
DABSQL_TRC_BIND 294
DABSQL_TRC_CANCEL 292
DABSQL_TRC_CHARFORM 294
DABSQL_TRC_CHARID 294
DABSQL_TRC_CLOSE 292
DABSQL_TRC_COMMIT 290
DABSQL_TRC_CONNECT 289

DABSQL_TRC_DECLARE 291
DABSQL_TRC_DEFINE 295
DABSQL_TRC_DESALLOC 295
DABSQL_TRC_DESCANY 295
DABSQL_TRC_DESCRIBE 291
DABSQL_TRC_DESFREE 295
DABSQL_TRC_DETACH 296
DABSQL_TRC_DISCONNECT 290
DABSQL_TRC_ENVCREATE 296
DABSQL_TRC_ERRGET 296
DABSQL_TRC_EXECUTE 292
DABSQL_TRC_EXECUTE_DIRECT 302
DABSQL_TRC_FETCH 291
DABSQL_TRC_GET_INFO 302
DABSQL_TRC_GETDIAG 301
DABSQL_TRC_GETPIECE 297
DABSQL_TRC_HDLALLOC 297
DABSQL_TRC_HDLFREE 297
DABSQL_TRC_LOBCLOSE 298
DABSQL_TRC_LOBLEN 298
DABSQL_TRC_LOBOPEN 299
DABSQL_TRC_LOBREAD 299
DABSQL_TRC_LOBWRITE 299
DABSQL_TRC_LOGGING 292
DABSQL_TRC_LOGLESS 293
DABSQL_TRC_OPEN 291
DABSQL_TRC_PARAMGET 300
DABSQL_TRC_PREPARE 290
DABSQL_TRC_ROLLBACK 290
DABSQL_TRC_SESBEGIN 300
DABSQL_TRC_SESEND 300
DABSQL_TRC_SET_INFO 302
DABSQL_TRC_SETPIECE 301
DABSQL_TRCSIZE 80
dabstart 119
dabstat 123
dabstop 121
DABSVLOGSIZE 83
DABTCP_TRCSIZE 87
DABTMP 79
DABXA_LOG_ERROR 84
dasysconf 184
Database Connection Server 549
Database Connection Server からの応答監視時間 137
Database Connection Server 接続時のタイマー監視 278
Database Connection Server 接続設定 209
Database Connection Server 接続設定項目 98
Database Connection Server に受信するときの最大バッファサイズ 100
Database Connection Server に送信するときの最大バッファサイズ 99
Database Connection Server の BLOB 型データ受取バッファサイズ 91
Database Connection Server の BLOB 型データ受取バッファサイズ 197
Database Connection Server のデータベース識別子 115
Database Connection Server のポート番号又はサービス名 137
Database Connection Server のホスト名又は IP アドレス 136
DBHOST 113
DBMS 549
DBNAME 114, 115
DBPARTNER 22, 44, 61
DBPARTNER [環境設定] 64, 178
DBPARTNER/Client 45, 550
DBPARTNER/Web 45, 550
DBPARTNER Library 22, 49, 61
DBPARTNER Library [環境設定] 64, 178
DBPARTNER Mobile 46, 550
DBPARTNER から実行するマルチスレッド 54
DBPARTNER と DABroker の接続時のタイマー監視 278
DBSHOST 136
DBSPORT 137
DBSTIMER 137
DECLARE 315

DECLARE 情報出力要求 291
DEFINE 320
DEFINE 情報出力要求 295
DESALLOC 320
DESALLOC 情報出力要求 295
DESCANY 321
DESCANY 情報出力要求 295
DESCRIBE 315
DESCRIBE 情報出力要求 291
DESFREE 321
DESFREE 情報出力要求 295
DETACH 321
DETACH 情報出力要求 296
DISCONNECT 316
DISCONNECT 情報出力要求 290
DTP モデル 550

E

ENVCREATE 322
ENVCREATE 情報出力要求 296
ERRGET 322
ERRGET 情報出力要求 296
EXECUTE 316
EXECUTE DIRECT 329
EXECUTE DIRECT 出力要求 302
EXECUTE 情報出力要求 292

F

FETCH 316
FETCH 情報出力要求 291
for C++設定 206
for COBOL 設定 207
for ORB 設定 208

G

GETDIAG 327
GETDIAG 情報出力要求 301
GET INFO 328
GET INFO 出力要求 302

GETPIECE 322
GETPIECE 情報出力要求 297

H

HDLALLOC 323
HDLALLOC 情報出力要求 297
HDLFREE 323
HDLFREE 情報出力要求 297
HiRDB 環境変数区切り文字 190
HiRDB 環境変数区切り文字 [UNIX] 85
HiRDB でのディクショナリ表のデータ属性 83
HiRDB の BLOB 型データ受取バッファサイズ 195
HiRDB の BLOB 型データ受取バッファサイズ 91
HiRDB の設定 97
HiRDB のディクショナリ表のデータ属性 189
HITSENER5 22, 44, 48, 61
HITSENER5 [環境設定] 64, 178
HITSENER5 for Cosmicube 550
HITSENER5 for RDB 550
HITSENER5 Professional 550
HITSENER5 Standard 550
HITSENER5 Web 49

I

IP アドレス 209, 551
IP アドレスからホスト名への変換オプション 92, 199

J

Java JTS を使った分散トランザクション処理 27
Java JTS を利用するまでの手順 28
Java TM 24, 27
JDBC インタフェース 551
JDBC ドライバ 551

L

LANG 環境変数 78
LOBCLOSE 323
LOBCLOSE 情報出力要求 298
LOBLEN 324

LOBLEN 情報出力要求 298
LOBOPEN 324
LOBOPEN 情報出力要求 299
LOBREAD 325
LOBREAD 情報出力要求 299
LOBWRIT 325
LOBWRITE 情報出力要求 299
LOGGING 317
LOGGING 情報出力要求 292
LOGLESS 317
LOGLESS 情報出力要求 293

M

MCONMAX 137
MCONPOOL 137
MCONPROC 137
Mobile Message Queue 551
MRSVPOOL 138

N

NLS_LANG 98

O

ODBC 551
ODBC3.0 の LONG 型データ受取バッファサイズ 196
OPEN 318
OPEN 情報出力要求 291
ORACLE_HOME 98
ORACLE_SID 98
ORACLE アクセスライブラリ再作成方法 142
ORACLE の LONG 型及び LONGRAW 型のデータの受け取りバッファサイズ 90
ORACLE の LONG 型データ受取バッファサイズ 196
ORACLE の設定 98
ORACLE のリスナー名 114
OTS 24, 551
OTS を使った分散トランザクション処理 25
OTS を利用するまでの手順 25

P

PARAMGET 326
PARAMGET 情報出力要求 300
PDCLTPATH 97
PDHOST 97
PDNAMEPORT 97
PDSQLTRACE 97
PDSWAITTIME 97
PDUAPERLOG 97
PREPARE 317
PREPARE 情報出力要求 290

R

RDA Link for Gateway 551
RDA Link for Gateway の分散定義ファイル名称 115
ROLLBACK 317
ROLLBACK 情報出力要求 290

S

services 72
SESBEGIN 326
SESBEGIN 情報出力要求 300
SESEND 326
SESEND 情報出力要求 300
SET INFO 328
SET INFO 出力要求 302
SETPIECE 327
SETPIECE 情報出力要求 301
Simple Interface 551

T

TCP/IP 551
TCP/IP 通信時キープアライブ 87, 198
TPBroker 552

V

VOSK RDB 編成ファイル 24

W

Web Page Generator 552

WWW 環境 20

X

X/Open DTP モデル XA インタフェース 552

XA トレースの取得形式 285

XA トレースを取得する 192

XA トレースを取得する〔UNIX〕 84

XDM/RD E2 の外部表 50

XDM/RD 外部表 22, 61

XDM/RD 外部表〔環境設定〕 64, 178

あ

アクセストレース 552

アクセストレース設定 314

アプリケーション 552

アプリケーション開発用インタフェース 21, 30

アプリケーションから実行するマルチスレッド 53

アプリケーションと DABroker との接続 60

い

イベントトレースの取得形式 280

イベントトレースの設定 92

え

エージェント 58, 552

エージェント機能で使用するファイルの運用 262

エージェント機能の実行 58

エージェント機能の使用 200

エージェント機能のパラメータ一覧 265

エージェント機能を使用するユーザのグループ管理
59, 254

エージェント実行結果格納ファイルの削除 264

エージェント実行情報の表示 129

エージェント実行情報ファイルの削除 263

エージェント実行名の割り当て規則 264

エージェント処理機能の設定 100

エージェントタイマー監視 278

エージェント登録情報の表示 128

エージェント登録情報ファイルの削除 263

エージェントの実行 58

エージェントの実行権限 258

エージェントログ選択オプション 104

エージェントログにインフォメーション情報を取得
する 104

エージェントログにインフォメーションメッセージを
取得する 203

エージェントログにエラー情報を取得する 105

エージェントログにエラーメッセージを取得する 204

エージェントログの取得形式 281

エージェントログの取得形式と取得内容 281

エージェントログの取得内容 283

エージェントログの選択 203

エージェントログファイル容量 104, 202

エラーコードとメッセージの対応 532

エラー時に DABroker が実行する処理 272

エラートレース機能 337

か

拡張データベースアクセストレース機能 306, 329

拡張データベースアクセストレースを取得する
〔UNIX〕 84

拡張データベースアクセストレースを取得する
〔Windows〕 191

仮想コネクション 133

環境設定の手順 63, 177

環境設定方法 179

環境設定ユティリティ 184

漢字コードセット 197

き

起動 230

強制終了 140, 231, 249

共通項目 76

共通設定〔Windows〕 186

<

クライアント／サーバ環境 19

クライアントアプリケーション 33
クライアントからの要求待ちタイムアウト 86, 193
クライアントとの通信トレースの表示 124
繰り返し実行 59
グループ管理情報ファイル 105
グループ管理情報ファイルの作成 255
グループ管理用ディレクトリ 105, 204

け

計画終了 140, 231, 249

こ

更新・検索項目数 81, 188
コネクションの解放 552
コネクションの確立 553
コネクションマルチ 134
コネクションマルチ運用ユティリティ 140
コネクションマルチ機能起動コマンド 139
コネクションマルチ機能終了コマンド 140
コネクションマルチ機能の使用 139, 248
コネクションマルチ定義ファイル 131
コネクションマルチ定義ファイルの作成 131
コネクションマルチ定義ユティリティ 226
コネクションマルチ名 132, 136, 221
コミット 553

さ

サーバ 553
サーバアプリケーション 33
サービス名 72
最大仮想コネクション数 134, 137, 224
最大仮想コネクション数と通信処理プロセス数の決定 134, 224
最大処理プロセス数 89, 195
最大接続ユーザ数 88, 194
作業ディレクトリ 101
作業ディレクトリ名 186

し

時間指定による実行 59
システムリソース消費量の比較 52
実行結果格納ファイル容量 103, 203
実行予約可能最大エージェント数 103, 202
実行予約時刻を経過しても有効にする秒数 106
実行予約日を経過しても有効にする日数 105, 205
実行予約日を経過しても有効にする秒数 205
自動起動 120
終了 230, 248
受信バッファサイズ 210
障害情報の取得 272
障害対策 272
使用する ORACLE のバージョン 82, 189
処理時間の比較 53
シングルコネクション 134
シングルスレッド 52

す

ステータス表示コマンド 140
スレッドの生成例 54
スレッドのチューニング 257

せ

正常終了 140
接続先サービス名 209
接続先データベース定義ファイル 110
接続先データベース定義ファイルの作成 110
接続先データベース定義ファイルの作成例 116
接続先データベース定義ユティリティ 213
接続先の指定 253
接続先ホスト名 209
接続する Database Connection Server のサービス名, 又はポート番号 99
接続する Database Connection Server のホスト名, 又は IP アドレス 99
セットアップコマンドの実行手順 68

そ

- 送信バッファサイズ 210
- 即時実行 59
- 即時配信実行 59
- 即時配信実行監視時間 106, 205
- その他の設定 107, 210

た

- タイマー監視 278

つ

- 通常の終了 231, 248
- 通信処理プロセス数 135, 137, 225
- 通信トレース 235
- 通信トレースの取得 87, 199

て

- データベースアクセス 57
- データベースアクセス処理終了待ちタイムアウト 187
- データベースアクセストレース 235
- データベースアクセストレース (UNIX) 289
- データベースアクセストレース (Windows) 314
- データベースアクセストレース選択オプション 289
- データベースアクセストレースの取得 80, 190
- データベースアクセストレースの表示 303
- データベースアクセストレースのファイル容量の目安 306
- データベースからの実行終了待ちタイムアウト 87
- データベース種別定義ブロック 111
- データベース種別名 112
- データベースの種類 22
- データベースのポート番号 114
- データベースのホスト名 113
- データベース名 112
- データベース名定義ブロック 112
- デーモンの生成・消滅・クリーンアップ実行結果ログ (共通) 287

と

- 同期アクセス処理 57
- 動作状態の表示 123
- 同時実行可能なデータベースアクセス数 88, 194
- 登録可能エージェント数 103, 202
- トラブルシュート情報 272, 275, 276
- トレースの取得形式 280
- トレースファイルサイズ 191

ね

- ネイティブライブラリ 553
- ネイティブライブラリ (libdaJavaC.sl) 67
- ネットワーク環境の設定 72, 182

ひ

- 非同期アクセス処理 57
- 表検索カラム数 80, 187

ふ

- ファイルの運用方法 [データベースアクセストレース] 312
- ファイル容量の見積もり 267
- プールコネクション数 136, 137, 226
- 複数 DBMS での共通インタフェース 17
- 分散オブジェクト技術 553
- 分散トランザクション処理 24, 59

ほ

- ポート番号 72, 209

ま

- マルチスレッド 17, 52
- マルチスレッドの実行 52

め

- メインフレーム系データベース 24
- メッセージ 344
- メッセージ一覧 352
- メッセージ中に出力されるコード 534

メッセージの形式 346
メッセージの出力先 345

も

モバイル環境 20, 554

ゆ

ユーザ ID の設定 253

よ

予約処理 57
予約処理作業ディレクトリ 89
予約処理作業ディレクトリ名 192
予約プールコネクション数 136, 138, 226

り

リモートアクセス 19
リモートアクセス設定 [Windows] 192
リモートアクセス設定項目 85

ろ

ローカルアクセス 19
ロールバック 554
ログファイルサイズ 189