

IBM i
バージョン 7.2

**データベース
データベース 分散データ
ベース・プログラミング**

IBM

IBM i
バージョン 7.2

**データベース
データベース 分散データ
ベース・プログラミング**

IBM

ご注意!

本書および本書で紹介する製品をご使用になる前に、445 ページの『特記事項』に記載されている情報をお読みください。

本製品およびオプションに付属の電源コードは、他の電気機器で使用しないでください。

本書は、IBM i 7.2 (製品番号 5770-SS1) に適用されます。また、改訂版で断りが無い限り、それ以降のすべてのリリースおよびモディフィケーションに適用されます。このバージョンは、すべての RISC モデルで稼働するとは限りません。また CISC モデルでは稼働しません。

本書にはライセンス内部コードについての参照が含まれている場合があります。ライセンス内部コードは機械コードであり、IBM 機械コードのご使用条件に基づいて使用権を許諾するものです。

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

原典： IBM i
Version 7.2
Database
Database Distributed database
programming

発行： 日本アイ・ビー・エム株式会社

担当： トランスレーション・サービス・センター

第1刷 2014.4

© Copyright IBM Corporation 1998, 2013.

目次

分散データベース・プログラミング	1
IBM i 7.2 の新機能	1
分散データベース・プログラミングの PDF ファイル	1
DRDA と DDM の概要	2
DRDA の概要	2
分散リレーショナル・データベースの処理	2
リモート作業単位	6
分散作業単位	7
分散要求	7
分散リレーショナル・データベースの他の用語および概念	10
分散リレーショナル・データベース体系サポート	10
DRDA および CDRA サポート	11
アプリケーション・リクエスター・ドライバー・プログラム	13
IBM i 上の分散リレーショナル・データベース例: Spiffy 社の分散リレーショナル・データベース	15
Spiffy 社の編成とシステム・プロファイル	15
Spiffy 社の自動車保守サービスの業務処理	16
Spiffy 社での分散リレーショナル・データベースの管理	17
DDM の概要	18
システムの互換性	20
DDM の機能の概要	20
DDM の基本概念	21
DDM の構成要素	22
DDM の構成要素: ソース DDM	23
DDM の構成要素: ターゲット DDM	23
DDM の構成要素: DDM ファイル	24
DDM の補足概念	30
DDM のクライアント・システムとしての IBM i	30
DDM のサーバー・システムとしての IBM i	33
DDM 関連ジョブおよび DDM 会話	35
例: DDM を使用した複数のリモート・ファイルへのアクセス	38
例: DDM を使用した複数のシステム上のファイルへのアクセス	38
例: DDM を使用したリモート・ファイルに対する複数の要求の処理	39
計画と設計	40
DRDA の計画と設計	40
分散リレーショナル・データベースの要件と要望事項の識別	41
分散リレーショナル・データベースのデータ要件	41
分散リレーショナル・データベースの能力	41
分散リレーショナル・データベースの目標と方向	42
アプリケーション、ネットワーク、および分散リレーショナル・データベースのデータの設計	43
ヒント: 分散リレーショナル・データベース・アプリケーションの設計	43
分散リレーショナル・データベースのネットワークの考慮事項	44
分散リレーショナル・データベースのデータの考慮事項	45
分散リレーショナル・データベースの管理方針の作成	46
分散リレーショナル・データベースの一般的な操作	46
分散リレーショナル・データベースのセキュリティの考慮事項	47
分散リレーショナル・データベースのアカウント	49
分散リレーショナル・データベースの問題分析	49
分散リレーショナル・データベースのバックアップおよびリカバリー	49
DDM の計画と設計	50
APPC ネットワークの DDM の通信要件	50
TCP/IP ネットワークの通信ネットワークの構成	51
DDM の機密保護要件	51
DDM のファイル要件	51
DDM のプログラム変更要件	52
DDM 体系に関連した制約事項	52
DDM の IBM i クライアントおよびサーバーの制約事項と考慮事項	53
DDM の IBM i 以外のターゲットに関する制約事項と考慮事項	54
初期セットアップ	55
IBM i 実行管理機能	56
実行管理機能環境のセットアップ	57
APPC サブシステム	58
TCP/IP サブシステム	60
独立補助記憶域プール上のユーザー・データベース	60
リレーショナル・データベース・ディレクトリーの使用	61
リレーショナル・データベース・ディレクトリーの処理	62
SNA 使用の場合の項目の追加	62
TCP/IP 使用の場合の項目の追加	64
リレーショナル・データベースの別名の指定	65
アプリケーション・リクエスター・ドライバーの項目の追加	66

WRKRDBDIRE コマンドの使用	66	インターネット・プロトコル・セキュリテ	
*LOCAL ディレクトリー項目	67	ィー・アーキテクチャー	105
独立補助記憶域プール上のユーザー・デー		特定のパスワードを平文で送信する場合の	
タベース用のディレクトリー項目	68	考慮事項	106
例: リレーショナル・データベースのディレク		ポートとポートの制限	106
トリーのセットアップ	69	ユーザー出口プログラムの使用による DRDA お	
セキュリティのセットアップ	72	よび DDM サーバーのアクセス制御	107
TCP/IP サーバーのセットアップ	73	サーバー・アクセス制御出口プログラムのパ	
対話式 SQL 用 SQL パッケージのセットアップ	74	ラメーター・リスト	109
DDM ファイルのセットアップ	74	例: ユーザー出口プログラムの使用による	
分散リレーショナル・データベースのテーブルへ		DRDA および DDM サーバーのアクセス制御	112
のデータのロード	75	機能使用 ID の使用による DRDA および DDM	
分散リレーショナル・データベースのテーブル		サーバーのアクセス制御	114
への新しいデータのロード	75	オブジェクト関連セキュリティー	115
SQL を使用したデータのテーブルへのロー		DRDA: 分散リレーショナル・データベース・オブ	
ド	75	ジェクトに対する権限	117
IBM i Query 管理機能の機能を使用したテ		DRDA: 分散リレーショナル・データベースの借	
ーブルおよびファイル内のデータ操作	76	用権限のもとで実行されるプログラム	118
データの入力、テーブルの更新、およびデー		分散リレーショナル・データベースの保護方針	119
タ・ファイル・ユーティリティーを使用		DRDA と DDM のアプリケーション開発	120
した照会の作成	77	DRDA のアプリケーション開発	121
システム相互間のデータの移動	77	分散リレーショナル・データベース・アプリ	
ユーザー作成のアプリケーション・プログ		ケーションのプログラミングの考慮事項	122
ラムの作成	78	分散リレーショナル・データベース・オブ	
対話式 SQL を使用したデータベースの		ジェクトの命名	122
Query	78	分散リレーショナル・データベースへの接	
DB2 for i Query 管理機能の機能を使用した		続	123
リモート・システムの Query	79	分散リレーショナル・データベースに固有	
テーブル相互間でのファイルのコピー	80	の SQL と SQL 呼び出し	133
ファイル・コピー・コマンドを使用したシ		DRDA 作業単位の終了	136
ステム間のデータの移動	80	ストアード・プロシージャー、ユーザー一定	
ネットワーク・ファイル・コマンドを使用		義関数、およびコミットメント制御	137
したネットワーク上のデータの転送	82	コード化文字セット識別子	137
オブジェクト保管および復元コマンドを使		他の DRDA データ変換	140
用したテーブルの移動	82	分散リレーショナル・データベース・プログ	
IBM i 以外のシステムから IBM i へのデータ		ラムの作成	141
ベースの移動	83	SQL ステートメントを含むプログラムの	
別の IBM システムからのデータ移動	83	プリコンパイル	142
IBM 以外のシステムからのデータ移動	85	アプリケーション・プログラムのコンパイ	
ル	86	アプリケーションのバインド	145
分散リレーショナル・データベースのセキュリテ		テストとデバッグ	146
ィー要素	86	SQL パッケージの処理	149
APPC ネットワークでのセキュリティーの要素	88	SQL パッケージの作成 (CRTSQLPKG) コ	
APPN 構成リスト	90	マンドの使用	150
会話レベルのセキュリティー	90	SQL パッケージの管理	150
APPC ネットワークでの DRDA サーバーの		DDM のアプリケーション開発	152
セキュリティー	91	DDM ファイルおよび SQL	152
TCP/IP ネットワークでのセキュリティーの要		DDM に対する言語、ユーティリティー、お	
素	94	よびアプリケーション・サポートの使用	153
TCP/IP ネットワークでのクライアントのセ		DDM のプログラミング言語に関する考慮	
キュリティー	94	事項	153
TCP/IP ネットワークでのサーバーのセキ		DDM のユーティリティーに関する考慮事	
ュリティー	102	項	163
接続セキュリティー・プロトコル	103	DDM の IBM i Access Family に関する考	
Secure Sockets Layer	104	慮事項	168

DDM の階層ファイル・システム API サポート	170	例: TCP/IP サーバーの終了	286
DDM での CL および DDS の使用	173	System i ナビゲーター のリスナーの開始	286
DDM 固有の CL コマンド	174	サーバー・ジョブ	286
DDM 関連の CL コマンドに関する考慮事項	187	サブシステム記述と DDM による事前開始	
DDM 関連の CL パラメーターに関する考慮事項	205	ジョブ項目	286
DDM 関連の CL コマンド・リスト	206	事前開始ジョブ	287
DDM データ記述仕様に関する考慮事項	216	サーバー・ジョブ・サブシステムの構成	290
DDM ユーザー・プロファイル権限	219	サーバー・ジョブの識別	291
DDM コマンドとパラメーター	219	IBM i ジョブ名	292
IBM i DDM でサポートされる DDM 体系のサブセット	219	サーバー・ジョブの表示	292
DDM コマンドとオブジェクト	222	ヒストリー・ログの表示	293
ユーザー・プロファイル権限	250	リレーショナル・データベース・ディレクトリーの監査	294
DDM での IBM i と CICS 間の考慮事項	251	DDM の操作に関する考慮事項	295
IBM i 言語、ユーティリティ、およびライセンス・プログラム	251	DDM でのファイル・アクセス	295
IBM i および CICS の言語に関する考慮事項	255	IBM i DDM でサポートされるファイルのタイプ	295
IBM i と他の IBM システムでの DDM 使用の比較	263	DDM ファイルとリモート・ファイルの存在	296
IBM i とシステム/36 の DDM の相違点	264	DDM のサーバー・システム・ファイル名の指定の規則	296
IBM i とシステム/38 の DDM の相違点	265	例: IBM i DDM リモート・ファイルへのアクセス (IBM i から IBM i)	299
DRDA と DDM の管理	266	例: システム/36 DDM リモート・ファイルへのアクセス (IBM i から IBM i)	300
リレーショナル・データベースの活動のモニター	266	DDM でのメンバーへのアクセス	301
分散リレーショナル・データベースにおけるジョブ処理	266	例: DDM リモート・メンバーへのアクセス (IBM i のみ)	301
分散リレーショナル・データベースにおけるユーザー・ジョブの処理	267	例: 特定メンバーをオープンする DDM ファイル	302
分散リレーショナル・データベースにおける活動ジョブの処理	269	DDM のアクセス方法の処理	302
分散リレーショナル・データベースにおけるコミットメント定義の処理	270	アクセス・インテント	303
分散リレーショナル・データベースのジョブ・ログによる要求情報の追跡	271	キー・フィールドの更新	303
分散リレーショナル・データベースのジョブの探索	273	削除済みレコード	303
リモート・システムの操作	275	ブロック化レコードの処理	304
プログラムで使用するオブジェクトの表示	277	可変長レコード	304
例: プログラム参照の表示	278	リモート・ファイルを対象とするその他の DDM 関連機能	304
分散リレーショナル・データベースからのコレクションの除去	279	リモート・システムでのファイル管理機能の実行	304
分散リレーショナル・データベースのジョブ会計	281	DDM のファイルとメンバーのロック	305
TCP/IP サーバーの管理	282	DDM 会話の制御	305
TCP/IP サーバーの用語	283	DDM リモート・ファイル情報の表示	307
TCP/IP 上での 接続の確立	283	DDM リモート・ファイル・レコードの表示	307
リスナー・プログラム	284	DDM でのコード化文字セット ID	307
TCP/IP サーバーの開始 (STRTCPSVR) CL コマンド	285	オブジェクト配布の使用	308
リスナーの制約事項	285	DDM でのオブジェクト配布の使用	308
例: TCP/IP サーバーの開始	285	分散データ管理作業の取り消し	308
TCP/IP サーバーの終了 (ENDTCPSVR) CL コマンド	285	ジョブの終了 (ENDJOB) コマンド	309
TCP/IP サーバーの制限の終了	286	要求の終了 (ENDRQS) コマンド	309
		DDM のシステム/36 クライアントとサーバーに関する考慮事項	309
		IBM i とシステム/36 ファイル間の DDM 関連の相違点	309

DDM の システム/36 クライアントから IBM i サーバーへの考慮事項	310
DDM の IBM i クライアントからシステム/36 サーバーへの考慮事項	310
DDM のシステム/36 に対する一時変更に関する考慮事項	313
DDM のパーソナル・コンピューター・クライアントから IBM i サーバーへの考慮事項	314
データの可用性および保護	315
分散リレーショナル・データベースの回復のサポート	315
分散リレーショナル・データベースのディスク障害後のデータの回復	316
補助記憶域プール	317
分散リレーショナル・データベースのチェックサム保護	317
分散リレーショナル・データベースのミラー保護	317
分散リレーショナル・データベースのジャーナル管理	317
索引の回復	319
索引再ビルド時間を短縮するためのテーブルの設計	320
システム管理アクセス・パス保護	320
コミットメント制御によるトランザクションの回復	321
分散リレーショナル・データベースの処理の保管と復元	325
分散リレーショナル・データベース環境での索引の保管と復元	326
分散リレーショナル・データベース環境でのセキュリティ情報の保管と復元	327
分散リレーショナル・データベース環境での SQL パッケージの保管と復元	327
分散リレーショナル・データベースのディレクトリーの保管と復元	328
分散リレーショナル・データベースのネットワーク冗長性の考慮事項	330
分散リレーショナル・データベース・ネットワーク内のデータの冗長性	332
DRDA と DDM のパフォーマンス	334
ネットワークによる分散リレーショナル・データベースのパフォーマンスの向上	334
システムによる分散リレーショナル・データベースのパフォーマンスの向上	335
DRDA のパフォーマンスに関する考慮事項	336
DRDA のブロック化に影響を与える要因	336
DRDA 照会ブロックのサイズに影響を与える要因	336
DDM のパフォーマンスに関する考慮事項	337
DDM でのバッチ・ファイル処理	341
DDM での対話式ファイル処理	342
DDM 会話の長さに関する考慮事項	343
例: アプリケーション・プログラミング	343
DRDA の例	343
例: プログラム定義	344

DRDA の例: RPG プログラム	348
DRDA の例: COBOL プログラム	357
DRDA の例: 組み込み SQL を使用する C プログラム	364
DRDA の例: Java プログラム	369
例: プログラム出力	374
DDM の例	375
DDM の通信のセットアップの例とタスク	375
DDM の例 1: 簡単な照会アプリケーション	376
DDM の例 2: ORDERENT アプリケーション	378
DDM の例 2: セントラル・システムの ORDERENT ファイル	378
DDM の例 2: ORDERENT プログラムの説明	379
DDM の例 2: リモート・システムの ORDERENT ファイル	380
DDM の例 2: サーバー・システムへのプログラム転送	381
DDM の例 2: ファイルのコピー	383
DDM の例 3: 複数の IBM i ファイルへのアクセス	383
DDM の例 4: システム/36 上のファイルへのアクセス	384
ユーザー FAQ	385
分散リレーショナル・データベースへの接続	385
IBM i システム値 QCCSID	386
DB2 for z/OS および DB2 Server for VM データベース・マネージャーの CCSID 変換の考慮事項	387
DB2 for Linux, UNIX, および Windows から接続しようとする SQL5048N メッセージが表示されるのはなぜですか?	387
IBM i ファイルはジャーナルする必要があるか?	388
パフォーマンスを向上させるために、照会データはどのような場合にブロック化されるのか?	388
SQL0969N エラー・メッセージで報告される SQLCODE と関連のトークンの解釈方法は?	389
WHERE 文節のホスト変数のタイプはパフォーマンスにどのような影響を与えるのか?	390
ライブラリー・リストを使用して、修飾されていないテーブルおよびビューの名前を解決できるか?	390
異種クライアントが NLSS ソート・シーケンス、システム命名、別の日時形式などのパッケージ・オプションをオーバーライドする方法	392
照会を実行するときに、戻される行がない理由は?	392
DB2 for IBM i との対話には、どのレベルの DB2 for Linux, UNIX, and Windows が必要ですか?	392
どのようにすれば IBM i プラットフォームでの DB2 for Linux, UNIX, and Windows のスクロール可能なカーソルのサポートを使用可能にできるか?	393
異種環境での相互運用に関するその他のヒント	393
DRDA と DDM のトラブルシューティング	396

IBM i問題処理の概要	397	DB2 for z/OS クライアント上でトランザク クション・プログラム名として QCNTSRVC を設定する	423
システムおよび通信の問題	398	DB2 for Linux, UNIX, and Windows クラ イアント上でトランザククション・プログラ ム名として QCNTSRVC を設定する	423
DRDA アプリケーション問題	399	TCP/IP サーバーのサービス・ジョブ	423
不正出力の問題の解決	399	QRWOPTIONS データ域	424
ループ、待機、またはパフォーマンスの問題 の解決	400	例: データ域を作成するための CL コマン ド	426
リスト作成	403	分散リレーショナル・データベース・ユーザーの 処理	427
プリコンパイラ・リストの作成	403	画面コピー	427
CRTSQPKG リスト	405	メッセージ	429
SQLCODE および SQLSTATE	406	メッセージ・タイプ	430
分散リレーショナル・データベース SQLCODE および SQLSTATEs	407	分散リレーショナル・データベースのメッ セージ	431
第 1 障害データ検知データの検索	410	APPC のプログラム開始要求障害の処理	434
障害の報告のためのデータの入手	411	TCP/IP の接続要求障害の処理	434
ジョブ・ログの印刷	411	サーバーが開始しないかポート ID が有効 でない	435
TCP/IP サーバー事前開始ジョブからのジョ ブ・ログの検出	412	DRDA 接続許可の障害	435
製品活動記録の印刷	413	システムが使用不可	436
ジョブ・トレース	413	対話式 SQL 固有の接続の障害	436
ジョブ・トレース	414	サーバーの事前開始ジョブが不十分	437
トレース開始	414	分散データベースのプログラミングに関連する情報	437
通信トレース	415	IBM i 情報	437
標準通信トレース	416	分散リレーショナル・データベース・ライブラリ ー	439
TCP/IP 通信トレース	418	他の IBM 分散リレーショナル・データベース・ プラットフォームのライブラリー	440
TCP/IP 通信トレースのフォーマット設定	419	アーキテクチャー・ブック	441
アプリケーション・サーバー問題の診断のための サービス・ジョブの開始	421	IBM Redbooks	442
APPC サーバーのサービス・ジョブ	421	特記事項 445	
独自のトランザククション・プログラム名の作 成と QCNTSRVC の設定	422	プログラミング・インターフェース情報	447
DB2 for i アプリケーション・リクエスト ー上でトランザククション・プログラム名と して QCNTSRVC を設定する	422	商標	447
DB2 for i アプリケーション・サーバー・ ジョブのデバッグに使用する独自のトラン ザククション・プログラム名の作成	422	使用条件	447
DB2 for VM クライアント上でトランザク クション・プログラム名として QCNTSRVC を設定する	423		

分散データベース・プログラミング

分散データベース・プログラミングでは、IBM® i ライセンス・プログラムの分散リレーショナル・データベース管理機能について説明します。分散リレーショナル・データベース管理により、アプリケーションは、そのアプリケーションの外部にあり、別のコンピューター・ネットワークに存在するデータへアクセスできるようになります。

本書は、IBM i の分散データ管理 (DDM) の概念、DDM 通信の準備、および DDM 関連のプログラミングについても説明しています。IBM i DDM を使用すると、リモート・ファイルにあるデータへアクセスしたり、リモート・システムによってローカル・ファイルへのアクセスを制御したりできるようにシステムを準備できます。

注: コードの例を使用することは、443 ページの『コードに関するライセンス情報および特記事項』の条件に同意したものとされます。



IBM i 7.2 の新機能

分散データベース・プログラミングのトピックに関する新情報や重要な変更情報についてお読みください。

- | IBM i 7.2 では、Distributed Relational Database Architecture™ (DRDA®) サポートが拡張され、以下の機能 | に関するサポートが含まれています。
- | • QDDMDRDASERVER サーバー認証項目の特殊値
- | • グループ・プロファイルの使用による DDM および DRDA 認証項目の簡素化された管理
- | • アクセス制限のための、DRDA/DDM ユーザー出口プログラムの代替となる QIBM_DB_DDMDRDA 機 | 能使用 ID
- | • 結合相互認証
- | • 3 部構成の SQL ステートメントでの RDB 別名の明示的な指定

新規情報または変更情報の見分け方

技術変更の加えられた箇所は、次のように使用されます。

-  イメージにより、新規または変更された情報の開始点を示します。
-  イメージにより、新規または変更された情報の終了点を示します。

PDF ファイルでは、左マージンに新規および変更情報のリビジョン・バー (I) があります。

このリリースの新機能および変更された機能に関する他の情報を見つけるには、プログラム資料説明書を参照してください。

分散データベース・プログラミングの PDF ファイル

この情報の PDF ファイルを表示または印刷できます。


本書の PDF 版を表示またはダウンロードするには、分散データベース・プログラミングを選択します。

PDF ファイルの保存

表示または印刷のために PDF をワークステーションに保存するには、以下のようにします。

1. ご使用のブラウザで PDF リンクを右クリックする。
2. PDF をローカルに保存するオプションをクリックする。
3. PDF を保存したいディレクトリーに進む。
4. 「保存」をクリックする。

Adobe Reader のダウンロード

これらの PDF を表示または印刷するには、Adobe Reader がご使用のシステムにインストールされている必要があります。このアプリケーションは、Adobe Web サイト (<http://get.adobe.com/reader/>)  から無償でダウンロードできます。

DRDA と DDM の概要

このトピックでは、IBM i 分散リレーショナル・データベース・サポートと IBM i 分散データ管理サポートの概念と処理について説明します。

DRDA の概要

IBM i 分散リレーショナル・データベース・サポートは、IBM 分散リレーショナル・データベース体系 (DRDA) の実装と、アプリケーション・リクエスト・ドライバ (ARD) プログラムによる他の SQL クライアントの統合で構成されます。

このトピックでは、分散リレーショナル・データベースと、IBM i オペレーティング・システム上での使用方法について説明します。

また、Spiffy Corporation という分散リレーショナル・データベースの例についても説明します。この架空の企業は、分散リレーショナル・データベース・アプリケーション・プログラムで、IBM i 製品を使用します。

分散リレーショナル・データベースの処理

リレーショナル・データベースとは、コンピューター内の 1 つ以上のテーブルに格納されたデータのセットです。

テーブルとは、データを 2 次元に配置したもので、次のテーブルに示されているように、横の行と縦の列から成っています。各行には、一連の値が含まれており、テーブルの列ごとに 1 つの値が入ります。列には名前が付けられており、特定のデータ・タイプ (文字、10 進数、整数など) が入ります。

表 1. 標準的なリレーショナル表

品目	品名	仕入れ先	数量
78476	野球ボール	ACME	650
78477	ラグビー・ボール	Imperial	228
78478	バスケット・ボール	ACME	105
78479	サッカー・ボール	ACME	307

システムでは、いろいろな方法でテーブルを定義しアクセスできます。システム上のテーブルに記述およびアクセスする 1 つの方法は、構造化照会言語 (SQL) のような言語を使うことです。SQL は標準的な IBM データベース言語であり、異なるシステムの間で分散データ処理を行えるようにするために必要な整合性を備えています。

システムのテーブルを記述しアクセスする別の方法は、データ記述仕様 (DDS) を使用して物理および論理ファイルに記述し、ファイル・インターフェースを使用してテーブルへアクセスすることです (たとえば、read や write の高水準言語のステートメントなど)。

SQL では、IBM i オペレーティング・システムで使われる用語とは異なる用語を使用します。ほとんどの SQL オブジェクトの場合、対応する IBM i システム・オブジェクトが存在します。次の表に、SQL リレーショナル・データベースの用語とシステム用語の関係を示します。

表 2. SQL 用語とシステム用語の関係

SQL 用語	システム用語
<p>リレーショナル・データベース。テーブルのセットと見なすことができ、データのリレーショナル・モデルに従って操作できるデータベース。ユーザーが IBM i 環境からアクセスできるリレーショナル・データベースは、システム用語の欄にリストされているように、3 つのタイプがあります。詳細については、リレーショナル・データベースのトピックを参照してください。</p>	<p>システム・リレーショナル・データベースまたはシステム・データベース。システムに接続されたディスク上に存在し、独立補助記憶域プール上には保管されないすべてのデータベース・オブジェクト。</p> <p>ユーザー・リレーショナル・データベースまたはユーザー・データベース。独立した補助記憶域プールには保管されていないデータベース・オブジェクトとともに、独立した単一の補助記憶域プールに存在するすべてのデータベース・オブジェクト。</p>
	<p>注:</p> <ul style="list-style-type: none"> • IBM i オペレーティング・システムは、独立補助記憶域プールがシステム上で構成されている場合、複数のリレーショナル・データベースに対するホストにすることができる。システム・リレーショナル・データベースは必ず 1 つ存在し、ユーザー・リレーショナル・データベースは 1 つ以上存在することができる。各ユーザー・データベースには、システム・データベース内のすべてのオブジェクトが入っている。 • システム・データベースは、コミットメント制御の観点では別のデータベースとして扱われ、SQL の観点でもユーザー・データベースに組み込まれていると見なされることを承知しておく必要がある。詳細については、『トランザクションおよびコミットメント制御のトラブルシューティング』のトピックを参照してください。 <p>リモート・リレーショナル・データベース、またはリモート・データベース。IBM i または別のシステム上に存在する、リモート・アクセス可能なデータベース。</p>
<p>スキーマ。ライブラリー、ジャーナル、ジャーナル・レシーバー、SQL カタログ、およびオプションのデータ・ディクショナリーで構成される。スキーマを使うと、関連するオブジェクトがグループ化され、名前で見つけられるようになる。</p> <p>注: スキーマは一般にコレクションとも呼ばれます。</p> <p>テーブル。列と行の集まり。</p>	<p>ライブラリー。これを使うと、関連するオブジェクトがグループ化され、名前で見つけられるようになります。</p> <p>物理ファイル。レコードの集まり。</p>

表 2. SQL 用語とシステム用語の関係 (続き)

SQL 用語	システム用語
行。一連の列を含む、表の横方向の部分。	レコード。フィールドの集まり。
列。1 つのデータ・タイプの、表の縦方向の部分。	フィールド。1 つのデータ・タイプの 1 バイト以上の関連情報。
ビュー。1 つ以上の表の列と行のサブセット。	論理ファイル。32 個までの物理ファイルの、フィールドまたはレコード (あるいはその両方) のサブセット。
索引。論理的に昇順または降順に配置されたテーブルの列内にあるデータのコレクション。	論理ファイルの 1 つのタイプ。
パッケージ。アプリケーション・サーバーで使う SQL ステートメントの制御構造を含むオブジェクト。	SQL パッケージ。SQL 用語での意味と同じ。
カタログ。テーブル、パッケージ、ビュー、索引、および制約についての情報を含む、一連のテーブルとビュー。QSYS2 のカタログ・ビューには、IBM i オペレーティング・システムにおける、すべてのテーブル、パッケージ、ビュー、索引、および制約についての情報が含まれています。さらに、SQL スキーマには、スキーマ内のテーブル、パッケージ、ビュー、索引、および制約についての情報だけを含む一連のビューがある。	類似のオブジェクトはありません。ただし、ファイル記述の表示 (DSPFD) コマンドおよびファイル・フィールド記述の表示 (DSPFFD) コマンドを使えば、SQL カatalogの照会で入手できる情報の一部を入手できます。

分散リレーショナル・データベースは、相互接続されたさまざまなデータベース間で分散する表や他のオブジェクトの集合で構成されます。データベース管理システムは DRDA プロトコルを使用して、これらの相互接続されたデータベースのどれにでもアクセスすることができます。そのような環境でのデータベースへのアプリケーション接続は、ローカルか DRDA のいずれかのタイプです。活動化グループごとに存在する最適化されたローカル・データベース接続は 1 つまでです。追加のデータベース接続については、DRDA を使用しなければなりません。

DRDA 用語では、アプリケーション・リクエスター (AR) とは、分散接続のアプリケーション側を処理するコードです。AR はデータを要求しているアプリケーションです。

アプリケーション・サーバー (AS) は、接続のデータベース側を処理するコードです。

分散リレーショナル・データベースの簡単な例が次の図に示されています。ここでは、アプリケーションを 1 つのシステムで実行し、データベースは別のシステム上に存在しています。

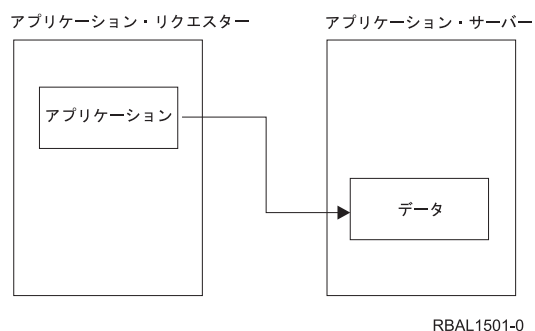


図 1. 単純な分散リレーショナル・データベース

DRDA は、アプリケーション・リクエスターとサーバーの間の複数層接続もサポートします。このトポロジーでは、アプリケーション・リクエスターの接続先のサーバーはアプリケーション・サーバーですが、さらに下流にある他のサーバーはアプリケーション・リクエスターと直接対話しないのでデータベース・サーバー (DS) と呼ばれます。また、データベース要求を発信するシステムとも要求に関するデータベース関数を実行するシステムとも異なる役割を強調するために、アプリケーション・リクエスターと最終データベース・サーバーの間の各アプリケーション・サーバーまたはデータベース・サーバーは、中間サーバーとも呼ばれます。

分散リレーショナル・データベースのより複雑な例が次の図に示されています。ここでは、アプリケーションを 1 つのシステムで実行していますが、2 つ目のシステム上で稼働しているデータベース管理システムは 3 つ目のシステム上に存在するデータベース・サーバーに要求を送ります。

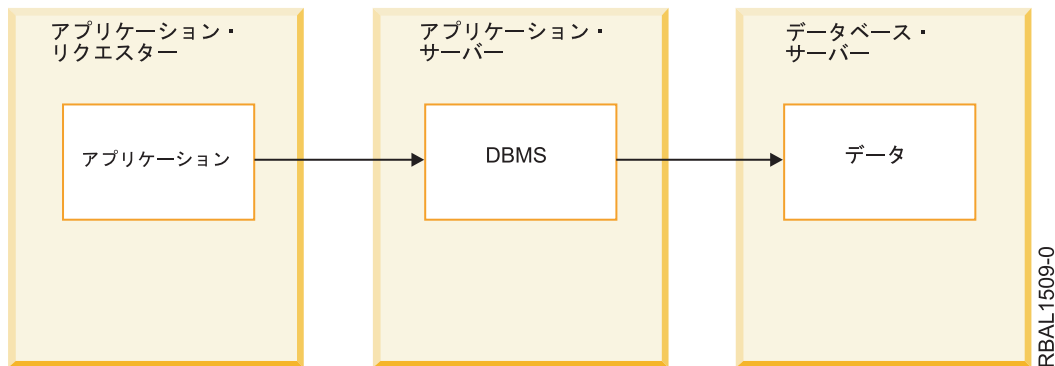


図 2. 複雑な分散リレーショナル・データベース

クライアントは AR と、サーバーは AS または DS と交換可能な単語としてしばしば使用されます。

作業単位とは、次の図に示されているように、完成した作業の一部を構成する、1 つ以上のデータベース要求および関連した処理のことです。簡単な例として、在庫管理アプリケーション・プログラムを使用して、在庫から部品を取り出すことが挙げられます。在庫プログラムでは、店の在庫アカウント・テーブルから項目を仮に削除し、その項目を、同じ場所の部品追加発注テーブルへ追加することができます。トランザクションという用語は、作業単位という概念を伝える別の表現です。

この例では、部品が店の在庫アカウント・テーブルから削除されて、追加発注テーブルへ追加されるまで、作業単位は完了しません。要求が完了したら、アプリケーション・プログラムは、作業単位をコミットできます。すなわち、作業単位に関連したデータベースの変更はすべて永続的なものになります。

作業単位サポートにより、アプリケーション・プログラムは、作業単位への変更をロールバックすることも可能です。作業単位をロールバックすると、最後のコミットあるいはロールバック操作後に加えられた変更は適用されません。このように、アプリケーション・プログラムは、データベースに対する一連の要求を 1 単位として扱います。

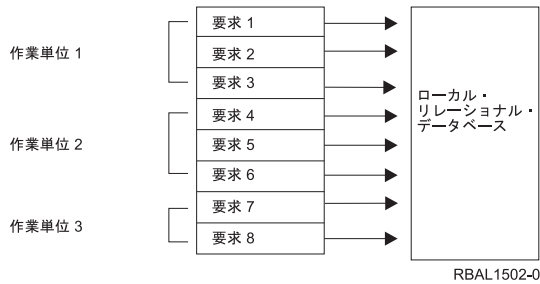


図3. ローカル・リレーショナル・データベースにおける作業単位

関連概念:

リレーショナル・データベース

トランザクションおよびコミットメント制御のトラブルシューティング

コミットメント制御に関する XA トランザクション・サポート

関連資料:

ファイル記述表示 (DSPFD) コマンド

ファイル・フィールド記述表示 (DSPFFD) コマンド

リモート作業単位:

リモート作業単位 (RUW) は、分散リレーショナル・データベース処理の一形態であり、アプリケーション・プログラムは、1 作業単位内でリモート・データベース上のデータにアクセスできます。

1 つのリモート作業単位には、複数のリレーショナル・データベース要求を含めることが可能ですが、すべての要求を同じリモート・データベースに対して発行する必要があります。リレーショナル・データベースに対する要求を別のリレーショナル・データベースへ送る前に、すべての要求を (コミットするかロールバックして) 完了しておかなければなりません。これを次の図に示します。

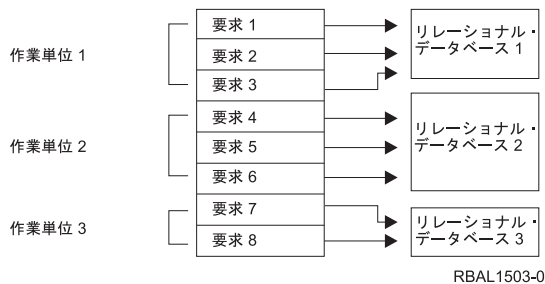


図4. 分散リレーショナル・データベースにおけるリモート作業単位

アプリケーション・プログラムは、要求を発行する前に、正しいリレーショナル・データベース・システムに接続する必要があるため、リモート作業単位はアプリケーション指示による分散処理です。しかし、アプリケーション・プログラム側では、正しい接続を確立するために、リモート・データベースの名前を知っているだけで構いません。

リモート作業単位サポートによって、アプリケーション・プログラムは、複数のロケーションでデータを読み取ったり更新することができます。ただし、作業単位内でプログラムがアクセスするすべてのデータを、同じリレーショナル・データベース管理システムで管理する必要があります。たとえば、店の在庫アプリケーション・プログラムは、別のロケーションに存在するテーブルを読み取ったり更新する前に、在庫および売掛管理作業単位をコミットしなければなりません。

リモート作業単位の処理では、コンピューターごとに、関連するリレーショナル・データベース管理システムと関連するアプリケーション・リクエスター・プログラムがあり、分散リレーショナル・データ要求の処理に役立てることができます。これにより、独自のアプリケーション・プログラムを使い、ローカル・リレーショナル・データを要求するときとほとんど同じ方法で、リモート・リレーショナル・データを要求できるようになります。

分散作業単位:

分散作業単位 (DUW) を使うと、ユーザーまたはアプリケーション・プログラムは、1 つの作業単位で、複数の場所にあるデータを読んだり更新することが可能になる。

1 つの作業単位内で、1 つのシステムで実行しているアプリケーションは、相手システムでサポートされている SQL を使うことにより、SQL 要求を複数のリモート・データベース管理システムへ送ることができます。たとえば、店の在庫プログラムでは、あるシステムにある在庫テーブルを更新すると同時に、同じ作業単位内の別のシステムにある売掛管理テーブルを更新することもできます。次の図に、このアイデアを示します。

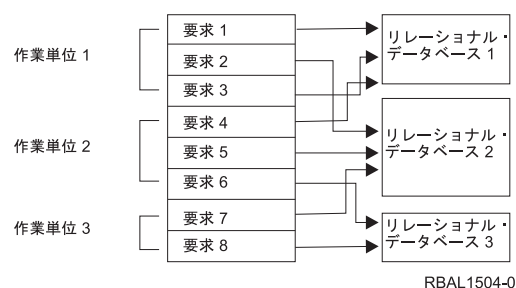


図 5. 分散リレーショナル・データベースにおける分散作業単位

要求のターゲットは、CONNECT TO や SET CONNECTION などの SQL ステートメントを使い、ユーザーまたはアプリケーションによって制御されます。各 SQL ステートメントは、それぞれ 1 つのロケーションのデータを参照しなければなりません。

アプリケーションが作業をコミットする準備ができると、アプリケーションはそのコミットを開始します。コミットメントの調整は、同期点管理プログラムまたはトランザクション・マネージャーによって実行されます。

DUW によって、1 つの作業単位の中で、複数のデータベース管理システムに対して、アクセスを更新したり、読み取り専用アクセスが可能になります。

1 つの作業単位の中で、アプリケーションが所定のデータベース管理システムを更新できるかどうかは、DRDA のレベル (リモート・リレーショナル・データベースへのアクセスに DRDA を使用する場合) と、接続と更新を行うときの順序に依存しています。

分散要求:

分散要求を使うと、ユーザーまたはアプリケーション・プログラムは、単一の SQL ステートメントで、1 つ以上のデータベースにあるデータを読んだり更新することが可能になります。

ローカル・データベースに対して実行しているアプリケーションは、1 つの SQL ステートメントで、1 つ以上のリモート・データベースに SQL 要求を宛てることができます。例えば、プログラムは 1 つの SQL ステートメントで、データベース 1 上の表 A とデータベース 2 上の表 B に対して更新を実行できま

す。次の図に、このアイデアを示します。

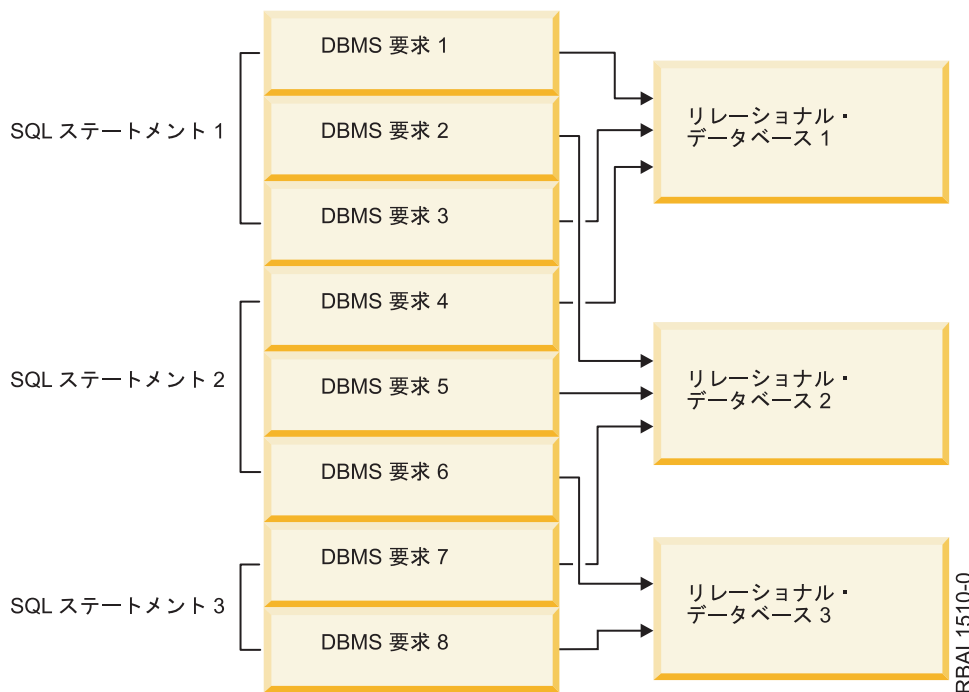


図 6. 分散リレーショナル・データベースの分散要求

IBM i 分散リレーショナル・データベースは、分散要求機能のサブセットをサポートします。この機能によって、1 つの SQL ステートメントで、単一のローカルまたはリモート・データベースに対する更新アクセスや読み取り専用アクセスが可能になります。

SQL ステートメントのターゲット・データベースは、参照されるオブジェクト上のデータベース修飾子を指定することにより、ユーザーまたはアプリケーションによって制御されます。オブジェクトの修飾について詳しくは、SQL 解説書を参照してください。

リモート・データベース接続は、データベース管理システムによって制御されます。ユーザーまたはアプリケーションは、これらのリモート・データベース接続に対して `CONNECT TO`、`SET CONNECTION`、`RELEASE` または `DISCONNECT` を使用できません。リモート・データベース接続は、ローカル・データベース接続に結び付いています。これらの接続は、必要に応じて開始され、ローカル・データベース接続の終了時に終了します。

特殊レジスタおよびグローバル変数は、リモート・データベース接続を介して伝搬されます。特殊レジスタまたはグローバル変数がリモート・データベース上にない場合は、無視されます。グローバル変数があるものの、同じ属性がない場合は、操作は失敗します。ローカル特殊レジスタまたはグローバル変数に対する更新は、必要に応じて既存のリモート・データベース接続を介して伝搬されます。

アプリケーションが作業をコミットする準備ができると、アプリケーションはそのコミットを開始します。コミットメントの調整は、同期点管理プログラムまたはトランザクション・マネージャーによって実行されます。

SQL ステートメントの中で、アプリケーションが所定のデータベースを更新できるかどうかは、DRDA のレベル (リモート・リレーショナル・データベースへのアクセスに DRDA を使用する場合) と、接続と更新を行うときの順序に依存しています。

3 部構成の名前:

3 部構成の名前は、リレーショナル・データベース、スキーマ (または命名方式によってはライブラリ一)、および SQL ステートメント内で使用するオブジェクトの名前を指定します。

ローカル・データベースに対して実行しているアプリケーションは、ローカル・データベースまたはリモート・データベースに SQL 要求を宛てることができます。SQL ステートメント内のデータベース修飾子は解決され、すべて同じデータベースを参照する場合は、SQL ステートメントはそのデータベースで処理されます。SQL ステートメント内のデータベース修飾子がすべて同じデータベースを参照していない場合は、SQL0512 がシグナル通知されます。

明示的な 3 部構成の名前

以下の例は、明示的リモート RDB 修飾子 (RemoteRDB) を使用する SQL ステートメントを示しています。

```
CRTSQLxxx PGM(MySchema/MyPgm) RDB(LocalRDB)

EXEC SQL SELECT * INTO :SERVICE FROM RemoteRDB/MySchema/MyTableA;

EXEC SQL COMMIT;
```

暗黙的な 3 部構成の名前

以下の例は、暗黙的ローカル RDB 修飾子を使用する 2 つの SQL ステートメントを示しています。

```
CRTSQLxxx PGM(MySchema/MyPgm) RDB(LocalRDB)

EXEC SQL SELECT * INTO :SERVICE FROM MyTableA;

EXEC SQL CALL MySchema/MyProcedureA;

EXEC SQL COMMIT;
```

暗黙的 RDB 修飾子について詳しくは、SQL 解説書を参照してください。

別名:

別名 とは、表またはビューの代替名のことで、別名は、現行データベースまたはリモート・データベースにある基礎となる表またはビューを参照できる点が他とは異なります。、ユニークです。

ローカル・データベースに対して実行しているアプリケーションは、ローカル・データベースまたはリモート・データベースに SQL 要求を宛てることができます。ステートメント内に別名がある場合、基礎となる表またはビューのデータベース修飾子を使用して、SQL ステートメントを処理する場所が判別されます。

以下の例は、基礎となる表のリモート・データベース修飾子 (RemoteRDB) を明示的に指定した別名を使用する SQL ステートメントを示しています。

```
CRTSQLxxx PGM(MySchema/MyPgm) RDB(LocalRDB)

EXEC SQL CREATE ALIAS MySchema/MyTableA FOR RemoteRDB/MySchema/MyTableA;

EXEC SQL SELECT * INTO :SERVICE FROM MySchema/MyTableA;

EXEC SQL COMMIT;
```

分散リレーショナル・データベースの他の用語および概念:

IBM システムでは、いくつかの分散リレーショナル・データベース・サポートは、DB2® for Linux, UNIX, and Windows および IBM DB2 DataPropagator for iSeries® V8.1 ライセンス・プログラムから提供されています。また、IBM i アプリケーション・プログラムの作成時に、これらの概念のいくつかを使うこともできます。

DB2 for i は、拡張プログラム間通信 (APPC) および TCP/IP 通信でのリモート作業単位と分散作業単位の両方をサポートしています。分散要求 とは、フェデレーテッド・データベース・システム中の複数のデータ・ソースに送られる SQL 照会のことです。このタイプの分散リレーショナル・データベース・アクセスを使うと、ユーザーまたはアプリケーション・プログラムは、1 つの SQL ステートメントを発行して、複数のロケーションにあるデータを読み取ったり更新できます。

分散リレーショナル・データベースのテーブルは、必ずしも相互に異ならなければならないわけではありません。テーブルによっては、他のテーブルのまったく同じコピーであるか、部分コピーである可能性があります。抽出、スナップショット、および複製は、分散処理を使用したコピーのタイプを示す用語です。

抽出 (Extract) は、ユーザーが要求したテーブルのコピーです。このコピーは、特定のデータベースから抽出され、ユーザー指定のデータベースへロードされます。アンロードとロードのプロセスを周期的に繰り返すことにより、更新したデータを取り出すことができます。抽出は、たまにしか変更されないデータの読み取り専用コピーが、1 回限り生じるあるいは低頻度の場合に最適です。

スナップショットは、システムによって自動的に生成される、テーブルの読み取り専用コピーです。システムは、ソース・テーブルからのこのようなコピーを、ユーザー指定の周期 (日ごと、週ごと、月ごとなど) でリフレッシュします。スナップショットは、更新された情報を周期的に受け取るための自動的な処理を必要とするようなロケーションで最適です。

データの複製 (replication) とは、システムがテーブルのコピーを自動的に更新することです。これは、テーブルのコピーが複数のロケーションに格納されるという点で、スナップショットと似ています。データ複製は、更新されることが少なく、高い信頼性と迅速な検索が求められる状況で、特に効果的です。

テーブルを、ネットワーク内のコンピューター・システム間で分けることも可能です。そのようなテーブルのことを、**分散テーブル**といいます。分散テーブルは、ローカルでの参照と格納を容易にするために、行ごとに横方向で分けることも、列ごとに縦方向で分けることもできます。分散テーブルを縦方向に分けた列は、分散テーブルを横方向に分けた行と同様、いろいろなロケーションに配置されます。さまざまなロケーションに分けられていても、ユーザー側からは、テーブルが 1 つのロケーションに存在するかのように見えます。分散テーブルは、テーブルの特定部分にアクセスし更新する要求が、テーブルの特定部分そのものと同じロケーションから発生する場合に、特に効果的です。

関連概念:

13 ページの『IBM i 上の分散リレーショナル・データベース』

DB2 for i は、IBM i に関するすべてのデータベース管理機能を提供します。システムの分散リレーショナル・データベースのサポートは、通信、実行管理機能、セキュリティー機能などの機能のサポートと同様に、オペレーティング・システムの不可欠な部分です。

分散リレーショナル・データベース体系サポート

分散リレーショナル・データベース処理のための分散リレーショナル・データベース体系 (DRDA) サポートは、IBM リレーショナル・データベース・プロダクトで使われます。DRDA サポートでは、アプリケーション・プログラムおよびリモート・リレーショナル・データベース間の通信プロトコルを定義しています。

DRDA サポートは、IBM および IBM 以外の環境の両方で分散リレーショナル・データベース管理機能を提供します。IBM 環境では、次のプログラムでリレーショナル・データが管理されます。

- DB2 for AIX®
- DB2 for HP-UX
- DB2 for i
- DB2 for Linux
- DB2 for Sun Solaris
- DB2 for VSE/VM
- DB2 for Windows
- DB2 for z/OS®

DRDA サポートは、リレーショナル・データベース管理機能が同種環境および異種環境で稼働するのに必要な、データベース情報にアクセスするための構造を提供します。例えば、DB2 for i データベースのある複数のシステム間でのリレーショナル・データのアクセスは、同種環境における分散です。一方、DB2 for i と別のタイプのシステム、または IBM i に組み込まれているものとは異なるクライアントとの間でのリレーショナル・データのアクセスは、異種環境での分散になります。1 つの具体的な例としては、DB2 for i と IBM DB2 Universal Driver for Structured Query Language for Java™ (SQLJ) and Java Database Connectivity (JDBC) の間で行われるリレーショナル・データのアクセスがあります。

SQL は、標準的な IBM データベース言語です。これを使用すると、同種操作環境と異種操作環境間で分散データ処理を行えるようにするために必要な整合性を備えています。DRDA サポートで SQL を使用すると、DRDA 実装をサポートする環境間で、データを定義、検索、および操作することが可能になります。

Distributed Relational Database Architecture は、分散データ管理 (DDM) アーキテクチャーの拡張アーキテクチャーです。ただし、データにアクセスするための DRDA と DDM メソッドは異なります。DRDA は、SQL の拡張で、DDM はネイティブ I/O の拡張機能です。

分散リレーショナル・データベース処理を使用すると、アプリケーションはローカル・システム上のリレーショナル・データベース・ディレクトリーを使用して、リモート・システムに接続できます。リレーショナル・データベース・ディレクトリーは、リレーショナル・データベース名とそのデータベースまでの通信パスの間の、必要なリンクを提供します。分散リレーショナル・データベースのもとで実行されるアプリケーションでは、データベース名を識別して、処理に必要な SQL ステートメントを実行するだけです。

DRDA および CDRA サポート

分散リレーショナル・データベースは、別の種類のコンピューター間で共有できるだけでなく、それぞれのコンピューターが別の国や地域にあっても構いません。

システムが同じでも、システムで使用する言語に応じて、データをコード化する方法は異なることがあります。システムが異なれば、データをコード化する方法も異なります。例えば、DB2 for Linux, UNIX, and Windows ライセンス・プログラムを実行している z/OS 製品、IBM i 製品、および Windows システムでは、数値データをそれぞれ固有の形式でコード化します。さらに、文字データをコード化する場合には、z/OS と IBM i 製品は EBCDIC コード化体系を使いますが、DB2 LUW を実行している Windows システムでは ASCII コード化体系を使います。

数値データでは、このような違いは問題になりません。分散リレーショナル・データベース体系 (DRDA) サポートを提供している異種システムでは、あるコンピューター・システムと別のコンピューター・システム間で数値を表示する方式が異なっている場合、その違いはすべて自動的に変換されます。例えば、IBM i

のアプリケーション・プログラムが DB2 for i データベースから数値データを読み取る場合には、DB2 for i は数値データを z/OS の形式で送り、IBM i データベース管理システムがそれを IBM i の数値形式に変換します。

これに対して、文字データの処理は複雑になりますが、それでも分散リレーショナル・データベース内で処理することができます。

CDRA による文字変換

コード化スキームに違いがある (Extended Binary Coded Decimal Interchange Code (EBCDIC) と American Standard Code for Information Interchange (ASCII) など) だけでなく、言語に関連した違いもあり得ます。

たとえば、システムが異なる言語用に構成されていれば、同じコードに割り当てる文字が異なったり、同じ文字に割り当てるコードが異なったりする可能性があります。たとえば、米国英語用に構成されているシステムでは、デンマーク語用に構成されたシステムで å に割り当てるのと同じコードを、文字 } に割り当てる場合があります。ただし、これら 2 つのシステムで異なるコードが、\$ などの同じ文字に割り当てられる場合もあります。

異なるシステム間でデータを共用する場合には、文字データの認識は、ユーザーおよびアプリケーションですべて同じである必要があります。すなわち、たとえ \$ のコード化文字が各システムで異なっている場合でも、ニューヨークの Windows とコペンハーゲンの IBM i ユーザーは、両方とも \$ を \$ として認識する必要がありますということです。さらに、コペンハーゲンのユーザーは、たとえデンマーク語の å と同じコードであっても、文字がニューヨークで保管された場合には a を } として表示する必要があります。このようなことを実現するためには、\$ は Windows システムでの適切な文字コード (すなわち、米国英語文字セットの ASCII) に変換され、さらにニューヨークからコペンハーゲンに渡る時点で、今度はデンマーク語のコード (すなわち、デンマーク語文字セットの EBCDIC) に変換されなければなりません。この種の文字変換は、IBM i でも、他の IBM 分散リレーショナル・データベース管理プログラムの場合と同じように用意されています。この変換は、文字データ表現体系 (CDRA) に従って、一貫した方法で行われます。

CDRA は、システムによって使用する文字セットおよびコード化体系が異なっている場合でも、システム相互間で文字データを理解することができるように、文字データの属性を識別する方法を指定するためのものです。システム相互間で変換が行われるためには、各システムは、それぞれ他のシステムから受信する文字データの属性を理解しなければなりません。CDRA では、これらの属性がコード化文字セット識別子 (CCSID) によって識別されるように指定します。DB2 for z/OS、DB2 for VM、および IBM i データベース管理システムの文字データには、すべて CCSID があり、これによってコード化体系、文字セット、およびコード・ページの特定の組み合わせを示します。Extended Services 環境の文字データは、いずれもコード・ページだけしか持っていません (ただし、他のデータベース管理プログラムは、そのコード・ページ識別子を CCSID として扱います)。コード・ページとは、文字と内部コードの間の割り当ての特定のセットです。

たとえば、CCSID 37 は、コード化体系 4352 (EBCDIC)、文字セット 697 (ラテン文字、1 バイト文字)、およびコード・ページ 37 (アメリカ/カナダ国別拡張コード・ページ) を意味します。CCSID 5026 は、コード化体系 4865 (拡張 EBCDIC)、コード・ページ 290 の文字セット 1172 (カタカナ/漢字の 1 バイト文字セット)、およびコード・ページ 300 の文字セット 370 (カタカナ/漢字の 2 バイト文字セット) を意味します。

DRDA が使用可能なシステムには、広範囲にわたる CCSID と CCSID の間、および CCSID とコード・ページの間で文字データを変換するためのメカニズムが組み込まれています。多くの CCSID およびコード・ページの場合の文字変換は、すでにこれらのプロダクトに組み込まれています。IBM i がサポートしている CCSID について詳しくは、『IBM i グローバリゼーション』トピックを参照してください。

関連概念:

IBM i グローバリゼーション

関連資料:

137 ページの『コード化文字セット識別子』

国別の言語サポートには、最小限の文字セットの適正な処理が必要です。

アプリケーション・リクエスター・ドライバー・プログラム

アプリケーション・リクエスター・ドライバー (ARD) プログラムは、出口プログラムの 1 つのタイプで、DB2 for i 以外のデータベース管理システムが管理するデータを SQL アプリケーションでアクセスできるようにするものです。

IBM i クライアントは、次の操作のときに、ARD プログラムを呼びます。

- 構造化照会言語パッケージの作成 (CRTSQLPKG) コマンドまたは CRTSQLxxx コマンドを使用して行われる、SQL プリコンパイルのパッケージ作成ステップにおいて、リレーショナル・データベース (RDB) パラメーターが ARD プログラムに対応する RDB 名に合致したとき。
- SQL ステートメントの処理時に、現行接続が ARD プログラムに対応する RDB 名であるとき。

このような呼び出しによって、ARD プログラムは、SQL ステートメントとそのステートメントについての情報をリモート・リレーショナル・データベースに渡し、その結果をアプリケーション・リクエスター (AR) に戻すことができます。すると、AR は、その結果をアプリケーションまたはユーザーに戻します。ARD プログラムがアクセスするリレーショナル・データベースをアクセスすることは、異環境で DRDA アプリケーション・サーバーにアクセスするのと似ています。

ARD プログラムは、RDB ディレクトリー項目の追加 (ADDRDBDIRE) コマンドを使用することにより、システムに登録されます。指定するパラメーターの 1 つは、プログラムが置かれているライブラリーです。補助記憶域プールで構成されているシステムの場合、ARD プログラムはシステム・データベース内のライブラリー (システム ASP または構成済みの基本 ASP の一部であるライブラリー) 内に置かれていなければなりません。

関連概念:

アプリケーション・プログラミング・インターフェース

関連資料:

リレーショナル・データベース・ディレクトリー項目の追加 (ADDRDBDIRE) コマンド

構造化照会言語パッケージの作成 (CRTSQLPKG) コマンド

IBM i 上の分散リレーショナル・データベース

DB2 for i は、IBM i に関するすべてのデータベース管理機能を提供します。システムの分散リレーショナル・データベースのサポートは、通信、実行管理機能、セキュリティ機能などの機能のサポートと同様に、オペレーティング・システムの不可欠な部分です。

IBM i オペレーティング・システムは、分散リレーショナル・データベース・アーキテクチャー (DRDA) の実装をサポートする他のシステムを持つ分散リレーショナル・データベース・ネットワークの一部になる場合があります。IBM i は、同種または異種のどちらの環境でも、アプリケーション・リクエスター (AR) またはアプリケーション・サーバー (AS) にすることができます。IBM i オペレーティング・システムでの分散リレーショナル・データベースの実装では、リモート作業単位 (RUW) と分散作業単位 (DUW) をサポートしています。RUW では、1 つの作業単位の中で、1 つのデータベースに対する複数の要求を出すことができます。一方、DUW では、1 つの作業単位の中に複数のデータベースに対する複数の要求を出すことができます。

DUW サポートでは、1 つの作業単位の中で、あるシステムで部品の在庫量を減らし、他のシステムで部品の在庫量を増やすことができます。そして、この作業単位の終わりで 2 フェーズ・コミット処理を使用し、これらの変更をそれぞれのリモート・データベースにコミットします。DB2 for i は、分散要求をサポートしないので、各 SQL ステートメントでは、1 つのデータベースにアクセスできるだけです。アプリケーション・プログラムに提供されるサポート・レベルは、アプリケーション・サーバー (AS) で利用できるサポート・レベルと、接続と更新が行われる順序によって決まります。

アプリケーション・リクエスター・ドライバー (ARD) プログラムは、DRDA アクセスの他に、DRDA をサポートしないデータベースのアクセスにも使用できます。ARD プログラムによってアクセスされるリレーショナル・データベースへの接続は、異種環境サーバーへの接続のように扱われます。このような接続は、DRDA アプリケーション・サーバーへの接続、ローカル・リレーショナル・データベースへの接続、および他の ARD プログラムへアクセスする接続と共存できます。

IBM i オペレーティング・システムでは、スナップショットと複製の分散機能は、システムでは自動的に実行されません。DB2 DataPropagator 製品を IBM i にインストールおよび構成することによって、これらの機能を実行できます。また、ユーザー作成のアプリケーション・プログラムでこれらの機能を使うこともできます。分散リレーショナル・データベースでこれらの機能を編成する方法の詳細については、『データの可用性および保護』のトピックで説明します。

IBM i オペレーティング・システムでは、分散要求機能は、直接にはサポートされていません。しかし、DataJoiner プロダクトを利用し、分散照会を実行し、さまざまなデータ・ソースからのテーブルを結合することができます。DataJoiner は、IBM Information Warehouse ファミリー・プロダクトの包括的な情報カタログである、DataGuide とともに機能します。DataGuide には、企業のデータ・リソースについての情報リストを完成させるために、グラフィカル・ユーザー・インターフェースが備えられています。

IBM i ライセンス・プログラムには、SQL の実行時サポートが組み込まれています。IBM i で分散リレーショナル・データベース要求を処理するため、または SQL コレクションを作成するために、DB2 for i アプリケーション・リクエスターまたはアプリケーション・サーバーに、DB2 for i Query Manager and SQL Development Kit ライセンス・プログラムをインストールする必要はありません。ただし、SQL ステートメントのあるプログラムをプリコンパイルする、対話式 SQL を実行する、または DB2 for i Query Manager を実行するためには、DB2 for i Query Manager and SQL Development Kit プログラムが必要です。

IBM i オペレーティング・システム上で DRDA を実装するための通信サポートが、TCP/IP または IBM システム・ネットワーク体系 (SNA) のいずれかのもとで、拡張対等通信ネットワーク機能 (APPN) の有無には関係なく、拡張プログラム間通信機能 (APPC) プロトコルを介して提供されます。

関連概念:

10 ページの『分散リレーショナル・データベースの他の用語および概念』

IBM システムでは、いくつかの分散リレーショナル・データベース・サポートは、DB2 for Linux, UNIX, and Windows および IBM DB2 DataPropagator for iSeries V8.1 ライセンス・プログラムから提供されています。また、IBM i アプリケーション・プログラムの作成時に、これらの概念のいくつかを使うこともできます。

123 ページの『分散リレーショナル・データベースへの接続』

分散リレーショナル・データベース・アプリケーションが分散されるのは、それが別のシステム上のリレーショナル・データベースへの接続可能性を備えているからです。

315 ページの『データの可用性および保護』

分散リレーショナル・データベース環境では、データ可用性の観点から、ネットワーク内の個々のシステムでのデータの保護だけでなく、ユーザーがネットワーク内のデータに確実にアクセスできるようにすることが必要です。

APPC、APPN、および HPR

TCP/IP の構成

OptiConnect

例: Spiffy 社の分散リレーショナル・データベース

IBM の資料には、分散リレーショナル・データベース・サポートの説明にあたって、Spiffy 社がしばしば登場しています。このトピック・コレクションでは、この架空の会社に若干の変更を加えて、IBM i 製品のネットワークでの DRDA の IBM i サポートを説明します。

このトピック・コレクションで使用する例は、特定の機能、接続、および処理を示すためのものです。したがって、本書以外の分散リレーショナル・データベースの資料で使用されている例とは正確に一致しない場合がありますが、なじみやすい例になるよう心掛けてあります。

Spiffy 社は架空の企業には違いありませんが、本書で説明する同社の業務は、同種の構造を持つ数社で行われている業務をモデルにしています。ただし、本書で取り上げる例は、たとえこの架空の会社の場合でも、分散リレーショナル・データベースを使用して行えることすべてを説明するものではありません。

Spiffy 社の編成とシステム・プロファイル:

Spiffy 社は、架空の全国的な規模を持つ製品卸売業者で、地域支社や地区販売店網を介して、得意先小売業者を対象に、自動車を中心とする製品の販売および保守サービスを行っています。

今日の自動車産業の競争の激しさを思えば、Spiffy 社のような企業の経営の成否は、質的に高いサービスを提供し、得意先に対する予備部品の納期を順守できるかどうかにかかっています。この競争に対処するために、Spiffy 社では、広大なサービス網を自社の販売店組織に組み込んで確立しました。

販売店組織を率いるのが、イリノイ州シカゴにある本社車両流通センターです。北米全域の数カ所に地域流通センターを置いています。そのうちの 2 つは、ミネソタ州のミネアポリスとミズーリ州のカンザス・シティーにあります。これらの流通センターでは、地域在庫を設けることによって、車両および予備部品の流通コストを最小限に抑制しています。ミネアポリス地域流通センターは、ほぼ 15 社の販売店を担当し、カンザス・シティー地域流通センターは、30 社にも及ぶ販売店を担当しています。

次の図に Spiffy 社のシステム編成図が示してあります。

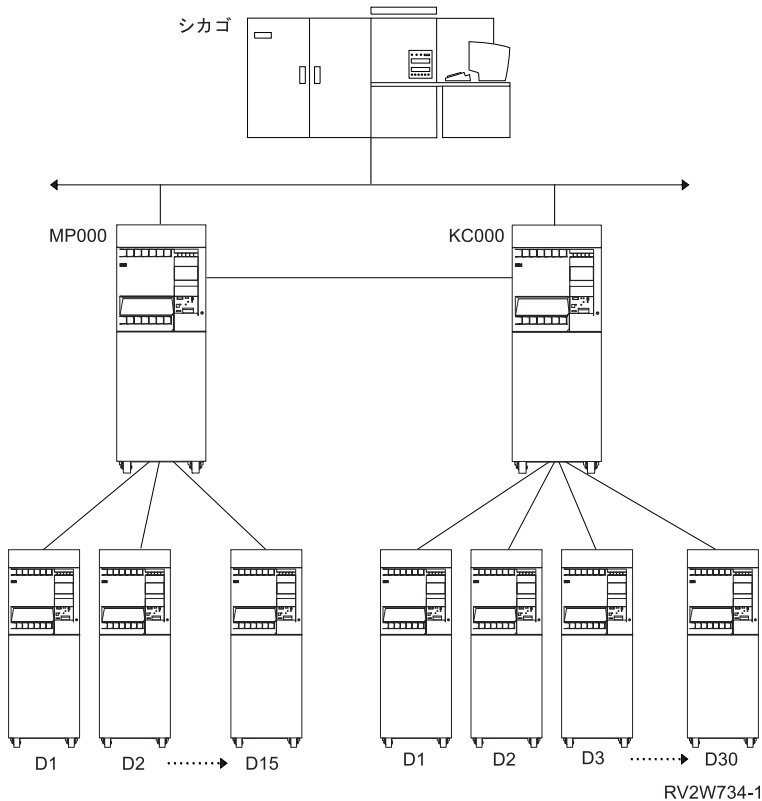


図 7. Spiffy 社のシステム編成

Spiffy 社では全国的な統合通信ネットワークを構築中です。自動車事業部では、地域流通センターおよび販売店を結ぶ IBM i 製品のネットワークを設けています。これを本社車両流通センターの System z[®] プラットフォームに接続します。このネットワークは、競争上の優位を維持するために不可欠の事業資産と見なされます。

本社の流通センターが関係のある意思決定支援ソフトウェアとともに、DB2 for z/OS を同センターの System z プラットフォームで稼働しています。このシステムを使用するのは、各種のアプリケーション・プログラムを使用して一度に処理しなければならないデータの量が多いからです。本社車両流通センターのシステムは、自動車事業部のデータ処理専用ではありません。会社の作業や処理の中には、まだ分散データベース環境での操作の対象になっていないものがあり、流通センターのシステムはそれら进行处理しなければならないからです。地域流通センターは、IBM i 製品を稼働しています。そこでは、SNADS および SDLC プロトコルを使用する 5250 表示装置パススルーで APPC/APPN を使用しています。

すべての販売店で IBM i 製品が使用されており、それらのサイズはそれぞれ異なります。これらのシステムは、SDLC プロトコルを使用して地域支社に接続されます。最大規模の販売店では、会社のデータ処理機能を担当する、パートタイムのプログラマー 1 名とシステム・オペレーター 1 名を置いています。しかし、ほとんどの販売店ではプログラミングの専門的技術を備えたものは雇用しておらず、小規模な販売店では一般的なコンピューターの知識以上のものを備えた従業員をまったく置かないところもあります。

Spiffy 社の自動車保守サービスの業務処理:

Spiffy 社の自動車事業部では、業務は本書で取り上げる分散リレーショナル・データベース環境で自動化されています。

例が必要以上に複雑になるのを避けるために、社内で車両保守サービスにかかわる機能だけを考えることにします。

販売店には、規模に応じて 2000 から 20 000 社の得意先リストがあります。つまり、保守サービス受注が、小規模販売店の場合で毎日 5 件、大規模販売店の場合で毎日 50 件の割合になります。これらの保守サービス受注には、定期保守、保証修理、定期修理、および部品受注があります。

販売店では、手持ち在庫は需要の多い予備部品に限っており、自社の在庫データベースを管理しています。前述した両地域流通センターでは、要求に応じて部品を提供しています。販売店在庫については、予測モデルで制御されるバッチ処理による定期的な調達も行われます。

Spiffy 社での分散リレーショナル・データベースの管理:

Spiffy 社では、各販売店に対して、それぞれが 1 つ以上の IBM i 製品を備え、それらのシステムを特定の時間にネットワークで使えるようにしています。

ただし、各販売店では、それぞれが独立した企業として、自社のデータ処理リソースおよび手順を管理します。システムのサイズおよびシステム上で自動化される業務処理数については、各販売店ごとの要件および使用可能なリソースに応じて決まります。

Spiffy 社では、販売店がすべて在庫の分散リレーショナル・データベースを使用できる状態になっている必要があります。Spiffy 社には系列販売店があるので、これらを完全に補うための販売店ソフトウェアを備えています。これには分散リレーショナル・データベース環境にアクセスできるものもできないものもあります。Spiffy 社の系列販売店では、このソフトウェア・ツールの全セットを使用します。フランチャイズ契約加盟販売店の場合も、特に Spiffy 社の営業方式に合わせているので、ほとんどがこのソフトウェア・ツールの全セットを使用します。

地域流通センターでは、担当する地域の在庫を管理します。また、地域で使用されるすべての分散データベース・リソースの、データベース管理者としての機能も果たします。担当する責任は、各販売店のデータ処理能力のレベルによって異なります。地域流通センターは、地域の販売店が援助を必要とする際に、まず接触する連絡先になります。

ミネアポリス地域流通センターは、システムおよびネットワークに関する広範な経験および知識を備えた IBM i プログラマーのスタッフを備えています。担当する販売店は他の地域流通センターの場合のほぼ半数ですが、これはミネアポリス地域流通センターがネットワーク全体の IBM i サポート機能に集中できるようにするためです。これらの機能には、アプリケーション・プログラム開発、プログラム保守、および問題処理などがあります。

次にリストするのは、ネットワークの各活動レベルごとのデータベースに関する責任範囲です。

販売店

- 基本システム操作および管理の実行。
- 地区ユーザーの登録。

地域流通センター

- 新規販売店のデータ処理のセットアップ。
- 解約販売店のデータベース・リソースの分散。
- 地域内ネットワーク・ユーザーの登録。
- 地域の在庫の管理。
- 販売店に関する保守サービス計画の開発。

- 販売店を対象とする援助の窓口の運営。

その他の活動

地域流通センター活動に加えて、ミネアポリス IBM i 専門センターでは、次の活動も行います。

- アプリケーションの開発。
- 地域流通センターを対象とする援助の窓口の運営。
- データベース・パフォーマンスの調整。
- データベース問題の解決。

このトピックでは、多くの例を挙げて、得意先の保守または修理の予定に合わせて、在庫から部品を調達する過程を示しています。その他の例では、Spiffy 社の分散リレーショナル・データベース・ネットワークのサーバーのセットアップ、保護、モニター、および問題解決に使われる分散リレーショナル・データベース管理作業を示しています。

DDM の概要

このトピックでは、分散データ管理機能 (DDM) の目的、DDM が提供する各種機能、および IBM i DDM の概念について説明します。

DDM は、IBM i ライセンス・プログラムの一部です。ソースとしての IBM i DDM は、DDM 体系のレベル 2.0 以下をサポートします。ターゲットとしての IBM i DDM は、レコード・ファイル (データをレコードに読み書きするディスク上のファイル) タイプの場合、レベル 2.0 以下をサポートし、ストリーム・ファイル (ドキュメント) およびディレクトリー (フォルダー) の場合は、DDM 体系のレベル 3.0 以下をサポートしています。

IBM i DDM がサポートされることによって、アプリケーション・プログラムやユーザーは、リモート・システムにあるデータ・ファイルにアクセスすることができ、リモート・システムもローカル IBM i オペレーティング・システムにあるデータ・ファイルにアクセスすることができます (19 ページの図 8 を参照)。クライアント・システムとして DDM 体系をサポートするどのシステムも、接続先の他のどのシステムのデータにでもアクセスする (その許可を受けていれば) ことができます。接続先のシステムは、サーバー・システム (そのシステム上の 1 つ以上のファイルを使用したいという別のシステムからの要求を受け取るシステム) として DDM をサポートすることが必要です。ただしこの場合、クライアント・システムとサーバー・システムは、互換性のある DDM 体系のサブセットとレベルをサポートしていなければなりません。

フォルダー管理サービス (FMS) サポートを利用することによりパーソナル・コンピューターのユーザーは、IBM i サーバー・システムにあるフォルダーと文書にアクセスすることができます。ストリーム・アクセス方法がレベル 3.0 またはレベル 2.0 の DDM 体系をサポートするリモート・システムは、ローカル・システムにあるフォルダーと文書にアクセスすることができます。

DDM は、IBM i データベース管理サポートのファイルへのアクセス能力を拡張します。本書では、データベース管理機能とはローカル・ファイル処理を制御するシステム機能のことを指します。すなわち、この機能はローカル・システムに保管されているファイル内のデータへのアクセスを制御するとともに、そのデータの同じシステムにある要求元プログラムへの転送を制御します。

分散データ管理は、リモート・ファイル処理を制御します。DDM を使用することにより、IBM i アプリケーション・プログラムは、DDM をサポートする別のシステムに保管されているデータ・ファイルへアクセスすることができます。同様に、DDM を持つ他のシステムは、ローカル・システムのデータベース内のファイルへアクセスすることもできます。DDM は複数のシステム間でのファイル処理の分散をより簡単なも

のにします。

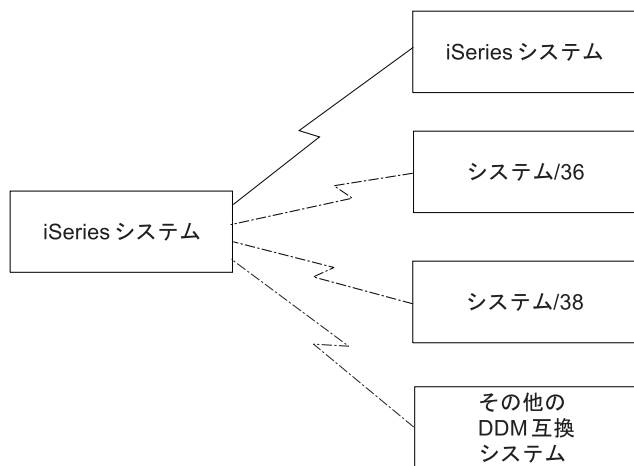


図8. クライアント・システムとサーバー・システム

DDM を使用するシステム相互間の通信は、拡張プログラム間通信機能 (APPC) サポート、拡張対等間通信ネットワーク機能 (APPN) サポート、または TCP/IP を使用して実行します。APPC と APPN を使用するために必要な情報については、IBM i PDF および資料の通信管理 マニュアルと APPC、APPN、および HPR (高性能経路指定) のトピックを参照してください。

フォルダー管理サービス (FMS) は、IBM i オペレーティング・システム上にある文書やフォルダーへのローカル・アクセスを可能にします。システム上のフォルダー管理機能へのパーソナル・コンピューターからのアクセスは、DDM を使用して行います。

注: IBM パーソナル・コンピューター用分散データ管理機能は、IBM i Access Family ライセンス・プログラムの IBM i 部分を使用します。

20 ページの図9 に示されているとおり、リモート・ファイルに関連した要求をユーザー・アプリケーションが出すシステムをクライアント・システムと呼びます。また、ファイル要求を受信するシステムをサーバー・システムと呼びます。別々の要求をシステムが同時に受信すると、そのシステムは、クライアント・システムであるとともにサーバー・システムにもなり得ます。

DDM を使用することにより、アプリケーション・プログラムは、サーバー・システムにあるファイル内のデータ・レコードを読み取り、追加、変更、および削除することができます。また、ファイルの作成、削除、名前の変更、またはサーバー・システムからクライアント・システムへのコピーなどのファイル関連操作を行うこともできます。

DDM を使用すれば、アプリケーション・プログラムやプログラム・ユーザーは、必要なファイルがローカル・システムまたはリモート・システムのどちらにあるかについて知っておく必要はありません。DDM では、ローカル・ファイルがローカル・システムで処理されるのと基本的には同じ方法でリモート・ファイルを処理するため、通常は、要求するファイルの所在場所の通知がアプリケーション・プログラムに送られることはありません。(ただし、エラー条件の場合、リモート・システムにアクセスがあったことを示すメッセージが必要に応じてユーザーに戻されます。) サーバー・システムのファイルの使用に関する通知メッセージは、クライアント・システムのジョブ・ログ内に入ります。

DDM を使用する場合、ファイルがどこにあるかはアプリケーション・プログラマーがわかっているだけであり、プログラマーは高水準言語 (HLL) プログラムの外部で制御言語 (CL) コマンドを使用すれば、ど

のファイルを使用するかを管理できます。なおプログラマーは、処理する通信障害に応じ、特定の回復機能を選択して使用できます。この場合は、該当する障害の処理を組み込むように、HLL プログラムを変更する必要が生じる場合もあります。

このため、ローカル・システム上でのデータベース・ファイル処理用にコンパイルされている BASIC、ILE COBOL、ILE RPG、ILE C、および IBM i などのプログラムの場合、リモート・システムへ移動された、またはリモート・システムにある同一ファイルを DDM で処理する際に、変更したり再コンパイルしたりする必要はありません。

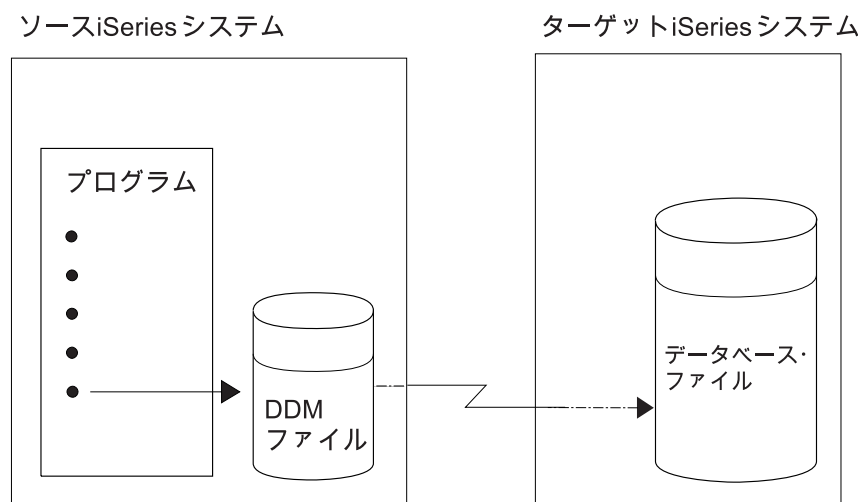


図9. クライアント・システムからサーバー・システムへのプログラムの移動

関連概念:

50 ページの『DDM の計画と設計』

分散データ管理機能 (DDM) を正しく使用するには、いくつかの要件を満たす必要があります。

システムの互換性

DDM は、アーキテクチャーの異なる複数のシステム間の通信の使用を可能にします。

IBM i オペレーティング・システムとシステム/36 のアーキテクチャーは異なりますが、いずれのシステムでも DDM を使うことにより互いに他方のデータベースのファイルにアクセスできます。正常に相互の通信を行うためには、いずれのシステムもレベル 2.0 またはそれ以下の IBM DDM 体系と互換性のある DDM を備えていなければなりません。またいずれのタイプのシステムも、IBM DDM 体系全体またはその一部を使用したり、アーキテクチャーを拡張したりできます。

IBM i 以外のシステムと通信する場合は、固有の機密保護上の考慮事項など、そのサーバーで提供されている DDM サポートのレベルについて考慮する必要があります。

レベル 3.0 またはそれ以下の IBM DDM 体系に関して詳述した DDM 体系資料のリストは、アーキテクチャー・ブックを参照してください。

関連資料:

441 ページの『アーキテクチャー・ブック』

DDM および DRDA のアーキテクチャー・ブックを以下にリストします。

DDM の機能の概要

以下に、サーバー・システム上の DDM 機能のタイプについて概説します。

通常は HLL プログラム内で指定する、以下のファイル 操作を、サーバー・システムのファイルに対して行うことができます。

- 1 つ以上のファイルの割り振り、オープン、またはクローズ。
- ファイル内のレコードの読み取り、書き込み、変更、または削除。

通常は CL プログラム内で指定するか、または CL コマンドで指定する、以下のファイル 操作と非ファイル 操作を、サーバー・システムのファイルに対して行うことができます。

- ファイルの内容のコピー。
- 物理ファイルまたは論理ファイルのメンバーに対する操作（メンバーの追加、消去、または除去など）の実行。ただし、ターゲット・システムが IBM i または システム/38 の場合に限りです。
- 次のような、非データ目的でのリモート・ファイル へのアクセス。
 - **ファイル記述表示 (DSPFD)** および **ファイル・フィールド記述表示 (DSPFFD)** などのコマンドを使用した、1 つ以上のファイルに関する情報の表示。これらのコマンドで、クライアント・システム上の DDM ファイルのファイル属性の表示、サーバー・システムにあるリモート・ファイルのファイル属性やフィールド属性の表示が可能になります。
 - **オブジェクト割り振り (ALCOBJ)** コマンドや **オブジェクト割り振り解除 (DLCOBJ)** コマンドを使用した、サーバー・システム上でのファイルのロックの制御。
 - **ファイル削除 (DLTF)**、**オブジェクト名変更 (RNMOBJ)**、**物理ファイル作成 (CRTPF)**、**ソース物理ファイル作成 (CRTSRCPF)**、**論理ファイル作成 (CRTLF)**、**物理ファイル変更 (CHGPF)**、**論理ファイル変更 (CHGLF)**、および **ソース物理ファイル変更 (CHGSRCF)** コマンドを使用した、ファイルの削除、名前変更、作成、および変更。
- 非データ目的でのリモート・システム へのアクセス。
 - クライアント・システムでの CL コマンド実行が適当でない場合、**リモート・コマンド投入 (SBMRMTCMD)** コマンドを使用して、CL コマンドをサーバー・システム (IBM i とシステム/38 に限る) に送って、そこで実行するようにする。**SBMRMTCMD** コマンドは、サーバー・システムでファイルを移動、保管、または復元するために使用する方法です。例えば、サーバー・システム上のデータベース・ファイルを移動するには、**オブジェクト移動 (MOV OBJ)** コマンドを送信できます。(**SBMRMTCMD** コマンドの標準的な使用法については、『DDM での CL および DDS の使用』の説明を参照し、さらに詳しい説明が必要な場合は、CL のトピックを参照してください。)

他のさまざまな非ファイル関連の操作を、サーバー・システムで実行できます。

関連概念:

制御言語

173 ページの『DDM での CL および DDS の使用』

このトピックでは、特定の制御言語 (CL) コマンドについての DDM 関連の情報、データ記述仕様 (DDS) に関する考慮事項、DDS キーワード、および DDM ユーザー・プロファイル権限について説明します。

DDM の基本概念

DDM は、リモート・ファイルのオープンとアクセス以外に、他の機能もサポートしますが、このトピックで説明する概念は主として、リモート・ファイルへのアクセスに関するものです。

ユーザーの立場から見れば、リモート・システム上のデータへのアクセスは、ローカル・システム上のデータへのアクセスとほぼ同じです。主な相違は、リモート・ファイルへアクセスが行われるごとに、データ・リンクを介してシステムからシステムへデータを渡す分だけ時間が余計にかかる点にあります。これ以外の点では、ユーザーもアプリケーション・プログラムも、アクセスするデータがローカル・ファイルまたは

リモート・ファイルのいずれのものであるかを知る必要はありません。その他の考慮事項については、『DDM のパフォーマンスに関する考慮事項』を参照してください。

IBM i 間の DDM ファイルの処理の場合、リモート・ファイルの処理とローカル・ファイルの処理は、ほぼ同様に行われます。このトピックの主旨は、DDM における相違点を説明することにあります。また、他のシステムも DDM を使用できるため、プログラマーが DDM を用いるためのシステムを正常に準備できるように、必要に応じて考慮事項や概念について述べています。

このトピックで説明している DDM の概念は、主として IBM i 間でのリモート・ファイル処理について述べたものです。分かりやすくするため、システム/36 およびシステム/38 に関連した概念を示している例もあります。システム/36 と IBM i の両方で DDM を使用する場合は、それぞれリモート・ファイルを指す方法が違うことを除いては、いずれのタイプも概念が類似していることを認識しておく必要があります。IBM i とシステム/38 は別々の DDM ファイルを使って、アクセスするそれぞれのリモート・ファイルを参照します。システム/36 は、アクセスするリモート・ファイルごとに 1 つのネットワーク資源ディレクトリ項目の入ったネットワーク資源ディレクトリを使用します。

関連概念:

30 ページの『DDM の補足概念』

残りのトピックの情報は、DDM の大半のユーザーにとっては必要ではありません。ここで述べる情報は、主として DDM に関してさらに詳細な情報を必要とする熟練したプログラマーを対象としています。

337 ページの『DDM のパフォーマンスに関する考慮事項』

このトピックには、DDM を使用するときパフォーマンスを向上させるための参考情報と、ある種の機能を実行するのに DDM 以外のものを使用する場合に関する情報が記述されています。

DDM の構成要素

DDM は、DDM を用いた複数のシステム間でのリモート・ファイル処理を行う複数の構成要素からなっています。

- ソース DDM (SDDM)
- ターゲット DDM (TDDM)
- DDM ファイル

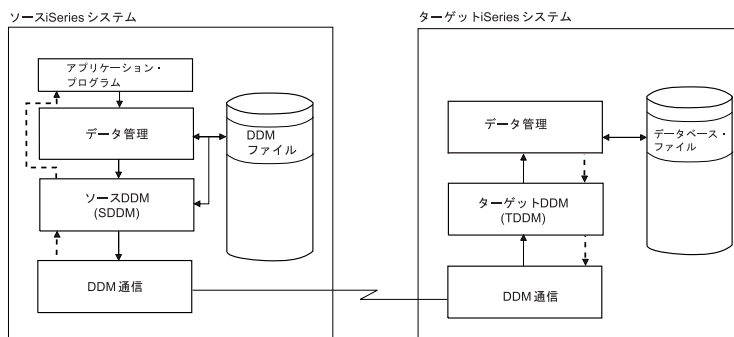


図 10. DDM を用いた通信

上の図は、両方のシステムでの DDM 通信に関与する基本構成要素がどのような相関関係を持っているかを示しています。

DDM ファイルがクライアント・システムのユーザーまたはプログラムからアクセスされると、そのプログラムまたはユーザーが作業しているジョブのために SDDM と TDDM との間で DDM 会話が開始されます。

DDM の構成要素: ソース DDM:

クライアント (または、ローカル) システム上のサポートは、DDM 機能を行うために必要に応じてソース・ジョブ内で開始されます。

ソース DDM (SDDM) は、クライアント・システムのアプリケーション・プログラムからのリモート・ファイルへのアクセス要求を DDM 要求に変換します。この要求は処理のためにサーバー・システムへ経路指定されます。SDDM サポートは、必要なりモート・ファイルを持つサーバー・システムとの DDM 会話を確立して管理します。

アプリケーション・プログラムが初めてリモート・ファイルにアクセスしようとしたとき、要求された DDM ファイルの探索がクライアント・システムで行われます。ローカル・ファイル処理の場合と同じように、ファイル名がライブラリー名で修飾されていない場合、プログラムが実行されているジョブの現行ライブラリー・リスト内で、その指定ファイルの探索が行われます。ファイルが見つかると、システムはファイルにアクセスし、それが DDM ファイルであることを判別したとき SDDM を開始します。

SDDM は開始時に、SDDM を開始したソース・ジョブと、DDM ファイル内でリモート・ロケーション値とモード値で識別されるサーバー・システムとの間で、すでに DDM 会話が活動化されているかどうかを検査します。使用可能な会話がすでに存在している場合、その会話が使用されます。存在していない場合には、適切なサーバー・システムに対してプログラム開始要求を出して、サーバー・システム上で TDDM (ターゲット・ジョブ) を開始し、SDDM と TDDM との間に DDM 会話を確立します。リモート・システムがプログラム開始要求を送ると、DDM ファイルのリモート・ファイルに関する情報によって自動的に作成されるパラメーターが渡されます。

TDDM が開始すると、SDDM は、処理すべき各プログラム要求をターゲット・ジョブに送ることができません。たとえば、リモート・ファイルの入出力 (I/O) 操作を行うことになっていれば、プログラムはそのファイルをオープンしてから、所定の操作要求を出します。つまり、SDDM がオープン要求を出し、TDDM がリモート・ファイルをオープンします。この後、SDDM は各ファイル操作要求を TDDM に送り、そしてこの両方が、アプリケーション・プログラムとリモート・ファイルとの間のデータのやりとりを処理します。DDM 機能を処理するときには、ローカル・ファイルの操作の場合と同じように、その機能が完了して結果を受け取るまで要求元プログラムは待機します。

関連概念:

30 ページの『DDM のクライアント・システムとしての IBM i』

これらの処置はすべて、サーバー・システム上で必要な処置も含めて、ソース・プログラムが要求する操作 (ファイルまたは非ファイル) を行う前に、正常に完了している必要があります。

DDM の構成要素: ターゲット DDM:

サーバー・システムのジョブは、着信 DDM 要求の結果としてターゲット (または、リモート) システム上で開始され、関連の DDM 会話が終了すると終了します。

ターゲット DDM (TDDM) は、リモート・ファイルへのアクセスのための DDM 要求を、サーバー・システム上でデータ管理要求に変換したうえで、クライアント・システムに情報を送り返すための処理を行います。

TDDM は、リモート・システムがプログラム開始要求を送ると開始します。TDDM は、サーバー・システムでのバッチ・ジョブとして始動します。TDDM が始動して DDM 会話が確立すると、TDDM は SDDM から送られてくる要求 (ファイル・オープンや読み取り操作、または非ファイル関連操作など) 待ちになります。

TDDM は、サーバー・システム上のオブジェクトへのアクセス要求を受け取ると、要求されたそのオブジェクトを探索します。そのオブジェクトがライブラリー名やパス名で修飾されていないと、ターゲット・ジョブの現行ライブラリー・リストまたは現行ディレクトリーを探索します。

要求されたオブジェクトが見つかり、TDDM は、要求のあった最初の操作をサーバー・システム上のデータベース管理機能またはフォルダー管理機能に渡し、渡された機能がそのオブジェクトに対する操作を行います。この操作が完了すると、データベース管理機能またはフォルダー管理サービスは操作結果を TDDM に戻し、次いで TDDM はそれを SDDM に渡します。SDDM は、その結果とそれに付随するすべてのデータ（読み取り命令で要求のあったレコードなど）をアプリケーション・プログラムに渡します。同じ処置が、受け取った残りの各入出力操作ごとに繰り返行われてから、オブジェクトがクローズされます。操作が正常に完了しない場合、SDDM はプログラムにエラー・メッセージを返し、エラーに関する情報を伝えます。

TDDM とターゲット・ジョブが活動する期間は、DDM 会話を開始したクライアント・システム・ジョブがその会話を終了するまでです。

関連概念:

33 ページの『DDM のサーバー・システムとしての IBM i』

IBM i ターゲット DDM (つまり TDDM) は、実際には DDM 関連のサーバー・システム・プログラムを実行するジョブです。これは、クライアント・システムがプログラム開始要求を送る (SDDM) と始動します。

DDM の構成要素: DDM ファイル:

DDM ファイルとは、サーバー・システムにあるデータ・ファイルにアクセスするための必要な情報を含むクライアント・システム上のファイルのことです。

DDM ファイルとは、クライアント・システム上にあり、リモート・ファイルを識別するシステム・オブジェクトのことです。このファイルは、装置ファイルとデータベース・ファイルの特性を結合しています。DDM ファイルは、装置ファイルとしてリモート・ロケーション名、ローカル・ロケーション名、装置名、モード、およびリモート・ネットワーク ID を参照し、リモート・システムがサーバー・システムかどうかを識別します。DDM ファイルは、アプリケーション・プログラムから見ればデータベース・ファイルであり、クライアント・システム・プログラムとリモート・ファイルとの間のアクセス装置の働きをします。

しかし、このファイルは、プログラムがデータベース操作のためにアクセスすることのできるデータ・ファイルではありません。クライアント・システムのプログラムが DDM のファイル名を指定すると、DDM はそのファイルの情報を使用して、アクセスしようとする データを持ったリモート・ファイルを探し出します。

DDM ファイル情報は、ロケーション に基づいています。リモート・ファイルが位置するリモート・ロケーションの指定は、**DDM ファイル作成 (CRTDDMF)** コマンドまたは **DDM ファイル変更 (CHGDDMF)** コマンドでリモート・ロケーション名 (RMTLOCNAME) パラメーターを使って行います。

CRTDDMF コマンドや **CHGDDMF** コマンド上で指定するリモート・ファイル名は、リモート・システムが使用する形式でなければなりません。

DDM ファイルの別の使用法として、制御言語 (CL) コマンドをサーバー・システムに実行依頼し、そのシステムで実行するという用法があります。この場合、通常は DDM ファイルに関連付けられるリモート・ファイルは無視されます。

関連資料:

175 ページの『リモート・コマンド投入 (SBMRMTCMD) コマンド』

リモート・コマンド投入 (SBMRMTCMD) コマンドは、コマンドをサーバー・システムで実行するよう、DDM を使って投入します。

SNA を使用した DDM ファイルの作成:

リモート・システムとの接続用の通信プロトコルとして SNA を使用する DDM ファイルを作成することができます。

SNA を使用する各 DDM ファイルには、以下のような情報が入っています。

DDM ファイル値および値の記述

DDM ファイル名

クライアント・システム上にあり、個々のリモート・ファイルを識別するために使用される DDM ファイルの名前。

リモート・ファイル名

リモート・ファイルの実際のファイル名。つまり、サーバー・システムで認識される時の名前。(受動側のシステム/36 では、この名前はリモート・ファイルのファイル・ラベルです。)

リモート・ロケーション名

リモート・ファイルのあるリモート・ロケーションの名前。このリモート・ロケーション名で、APPN/APPC を使用したサーバー・システム (リモート・ロケーション) へのデータ・リンクが設定され、該当する DDM ファイルにアクセスがあったときには、このリンクを経由して DDM 会話が確立されます。

装置 クライアント・システムにあり、リモート・ロケーションとの通信に使用される装置の名前。

ローカル・ロケーション名

ローカル・ロケーションの名前。これは、サーバー・システムが起動側のシステムと認識する名前です。起動側のシステムは、複数のローカル・ロケーションで構成できます。

モード ローカル・ロケーションとリモート・ロケーション間の通信に使用するモードの名前。

リモート・ネットワーク ID

リモート・ロケーションと合わせて使用されるリモート・ネットワーク ID。この値は、リモート・ロケーション名をさらに修飾します。リモート・ロケーション名が同じでもリモート・ネットワーク ID が異なる 2 つのロケーションは、まったく別々の 2 つのロケーションと見なされません。

タイプ リモート・システムと DDM 会話を確立するときに、リモート・ロケーションとの通信に使用される接続のタイプ。SNA 接続を使用する DDM ファイルを作成するには、*SNA を指定します。これは、デフォルト・タイプです。

TCP/IP を使用した DDM ファイルの作成:

リモート・システムとの接続用の通信プロトコルとして TCP/IP を使用する DDM ファイルを作成することができます。

TCP/IP を使用する各 DDM ファイルには、以下のような情報が入っています。

DDM ファイル値および値の記述

DDM ファイル名

クライアント・システム上にあり、個々のリモート・ファイルを識別するために使用される DDM ファイルの名前。

リモート・ファイル名

リモート・ファイルの実際のファイル名。つまり、サーバー・システムで認識されるときの名前。

リモート・ロケーション名

リモート・ファイルのあるリモート・ロケーションの名前。このリモート・ロケーション名で、TCP/IP を使用したサーバー・システム (リモート・ロケーション) へのデータ・リンクが設定され、該当する DDM ファイルにアクセスがあったときには、このリンクを経由して DDM 会話が確立されます。

タイプ リモート・システムと DDM 会話を確立するときに、リモート・ロケーションとの通信に使用される接続のタイプ。TCP/IP を使用する DDM ファイルを作成するには、*IP を指定します。

関連概念:

282 ページの『TCP/IP サーバーの管理』

DRDA および DDM TCP/IP サーバーでは、通常は、既存のシステム構成に変更を加える必要はありません。ただし、場合によっては、自分の要求によりよく合わせるため、問題を解決するため、システム・パフォーマンスを向上させるため、あるいはシステムでジョブを監視するために、システムによるサーバー・ジョブの管理方法を変更したいことがあります。

RDB ディレクトリー項目情報を使用した DDM ファイルの作成:

リレーショナル・データベース (RDB) ディレクトリー項目からのリモート・ロケーション情報を使用する DDM ファイルを作成することができます。

RDB ディレクトリー項目を使用する各 DDM ファイルには、以下のような情報が入っています。

DDM ファイル値および値の記述

DDM ファイル名

クライアント・システム上にあり、個々のリモート・ファイルを識別するために使用される DDM ファイルの名前。

リモート・ファイル名

リモート・ファイルの実際のファイル名。つまり、サーバー・システムで認識されるときの名前。

リモート・ロケーション名

*RDB を指定して、リモート・ロケーション情報を RDB ディレクトリー項目から取り出すことを指示します。

リレーショナル・データベース

リモート・ロケーション情報に使用されるリレーショナル・データベース項目の名前。RDB ディレクトリー項目のリモート・ロケーション情報を使用して、サーバー・システム (リモート・ロケーション) へのデータ・リンクを設定し、DDM ファイルにアクセスがあったときに DDM 会話を確立します。

ASP グループにアクセスするには、DDM ファイルのリモート・ロケーション情報の補助記憶域プール (ASP) グループに関連付けられた RDB ディレクトリー項目を指定する必要があります。

関連概念:

ディスク管理

ASP グループ選択におけるジョブ記述の影響:

ターゲットの DDM サーバーが ASP グループを使用するように構成され、DDM ファイルがリレーショナル・データベース名を指定すると、クライアント上の DDM ファイルに指定されたリレーショナル・データベース項目を使用して、ターゲット・ジョブの ASP グループを設定します。

リレーショナル・データベース名を指定しない DDM ファイルを使用するときには、ターゲット・ジョブの ASP グループは、ターゲット・ジョブが実行されているユーザー・プロファイルのジョブ記述にある初期 ASP グループ属性を使用して設定されます。

例: DDM の基本概念を APPC ネットワークで使用:

この例のアプリケーションは、DDM を使用してリモート・ファイルにアクセスします。このアプリケーションは、いくつかの都市に分けて倉庫を所有する企業で実行されるものです。

以下の図は、DDM ファイルに含まれる主要項目の相互関係を示しています。

シカゴの IBM i プラットフォームでは、データベース・ファイル・オープン (OPNDBF) コマンドにより、入力用としてファイル CUST021 のオープンを要求しています。このコマンドではファイル名が修飾されていないため、ファイルの検出にソース・ジョブ用のライブラリー・リストが使用されます。ファイルが検出されると、このファイルは NYCLIB ライブラリーに保管されます。

CUST021 は DDM ファイルであるため、ファイルがオープンされると、ソース・ジョブ内でシカゴのシステムの SDDM が開始されます。SDDM は、DDM ファイルにあるリモート・ロケーション名とモード名 (NEWYORK および MODENYC) を使って、該当するサーバー・システム (NEWYORK) との間の DDM 会話を確立し、そのサーバー・システム上でターゲット・ジョブ (TDDM) を開始します。クライアント・システム・プログラムがアクセスするリモート・ファイルは、ライブラリー XYZ 内の CUSTMAST です。

TDDM は SDDM からリモート・ファイル名を受け取ると、CUSTMAST という名前のファイルの割り振りとオープンを行います。このファイルは、クライアント・システム上の CUST021 という名前の DDM ファイルに対応するファイルです。

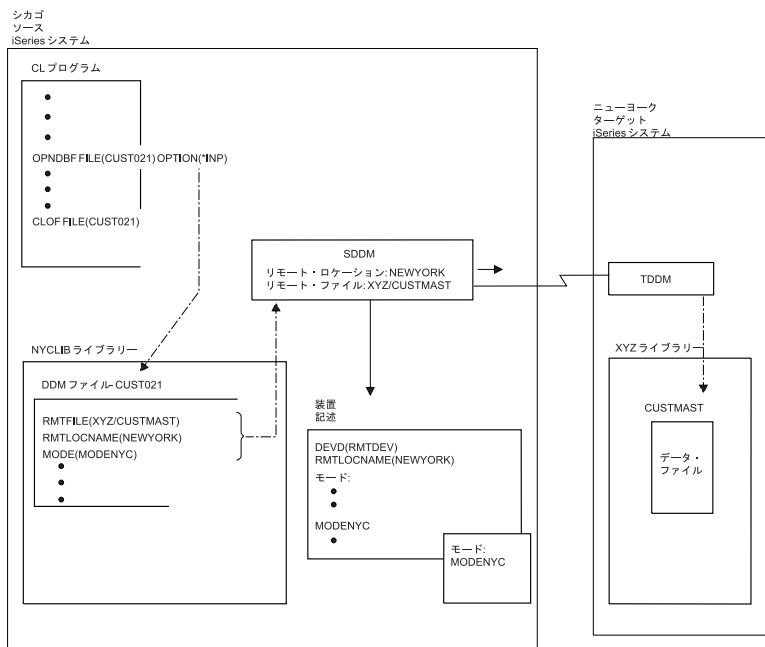


図 11. DDM ファイル・パラメーターとシステムとの関係

DDM ファイル内のリモート・ロケーション名は、そのファイルが所在するリモート・システムを識別します。ローカル・システムは、リモート・ロケーション名とあわせて、DDM ファイル内で指定された、その他の値を使用し、装置記述を選択します。装置記述の作成は、手作業で行うことも、また、APPN を使用するのであれば、システムが自動的に行った後その記述を活動化することもできます。SDDM は、APPN リモート・ロケーション名の中にある値 NEWYORK および MODENYC を使用して、サーバー・システムとの DDM 会話を確立します。SDDM から要求を出す前に、APPN 関連サポートをサーバー・システムで始動しておかなければなりません。(クライアント・システムでは、特にサポートは必要ありません。)

注: APPN 制御装置記述作成 (CRTCTLAPPN) および SNA ホスト制御装置記述作成 (CRTCTLHOST) コマンドの APPN パラメーターで、APPN サポートを使用するかどうかを決めます。

DDM の基本概念を APPN ネットワークで使用した例:

IBM i の拡張対等間通信ネットワーク機能 (APPN) サポートを使用すれば、ローカル・サーバーに直接接続されていないシステムへ DDM からアクセスすることができます。

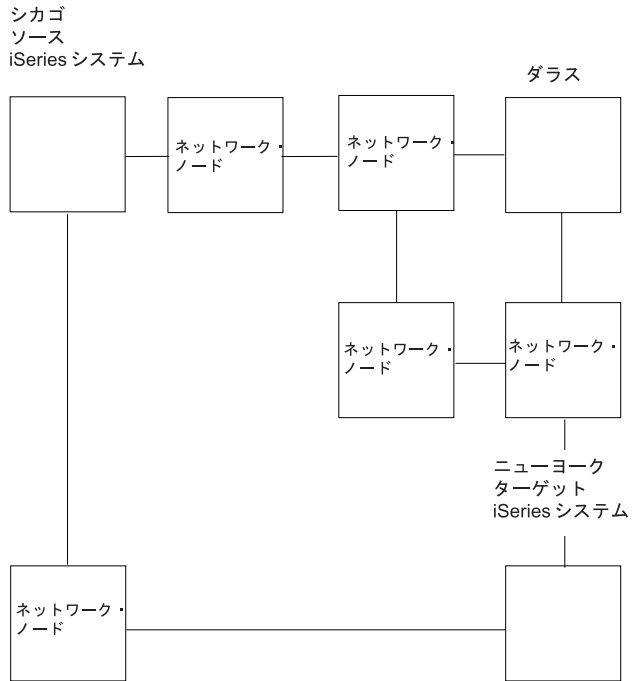


図 12. APPN ネットワークにおける DDM の使用

27 ページの『例: DDM の基本概念を APPC ネットワークで使用』の図 4 は、ニューヨークのシステム上のファイルにアクセスしようとするシカゴのシステムのプログラムを示しています。システムは直接接続されているものとして示されていますが、ネットワークが上の図のように構成されていても、同じ DDM 概念が適用されます。図の DDM ファイル CUST021 をシカゴのシステムでオープンすると、APPN サポートは NEWYORK という名前のリモート・ロケーションを検出し、ネットワーク経由の最適パスを決定して、そのロケーションとの DDM 会話を確立します。ソース DDM とターゲット DDM は、シカゴとニューヨークとの間でデータを転送する他のシステム (ネットワーク・ノード) がいくつあっても、この 2 つのシステムが直接接続されているかのように機能します。

ニューヨークからファイル CUSTMAST をネットワーク内のどこか他のシステム (例: ダラス) に移動すると、この例ではシカゴの DDM ファイルを変更する必要があります。リモート・ロケーション名は、NEWYORK から DALLAS に変更されます。ネットワーク内の多数のシステムがファイル CUSTMAST を参照する場合、このようにファイルを移動すると、それら各システムごとに DDM ファイルが変更されます。IBM i に備えられた複数のローカル・ロケーション名を設ける機能を使用すれば、これらのファイルの保守の作業が軽減されます。

上の図の場合、ニューヨークのシステムには、NEWYORK と FILELOC という 2 つのローカル・ロケーション名を付けることができます。シカゴの DDM ファイルでは、リモート・ロケーション名として FILELOC を使用します。ファイル CUSTMAST にアクセスする必要がある場合、APPN は NEWYORK という名前のシステムからロケーション FILELOC を検出し、従来の DDM 会話が確立されます。

ここでファイル CUSTMAST をニューヨークからダラスに移動した場合、ニューヨークのユーザーが自分のシステムからローカル・ロケーション FILELOC を削除し、それはダラスのシステムに追加されます。これを行うには、APPN のローカル・ロケーション・リストを使用します。ここでシカゴのプログラムからファイル CUSTMAST にアクセスすると、APPN サポートはダラスのシステムからリモート・ロケーション FILELOC を検出し、ダラスのシステムとの DDM 会話を確立します。CUSTMAST を移動しても、シカゴの DDM ファイルは変更されません。

この例は、複数のローカル・ロケーションの概念と、あるシステムから別のシステムへファイルを移動したときに、どのように保守の作業が軽減されるかについて示しています。この例は、DDM を介してアクセスする各ファイルごとに、固有ロケーション名の使用をお勧めするものではありません。別々のローカル・ロケーションに関連づけた方がよいと思われるファイルを判別する要因としては、それらのファイルの移動状況、およびそれらのファイルにアクセスするリモート・システムの数などがあります。これらの要因に基づいて判断してください。

DDM の補足概念

残りのトピックの情報は、DDM の大半のユーザーにとっては必要ではありません。ここで述べる情報は、主として DDM に関してさらに詳細な情報を必要とする熟練したプログラマーを対象としています。

以下で説明するのは、下記の各事項に関する詳細な概念と例です。

- TDDM (ターゲット・ジョブ) を始動するためのプログラム開始要求。
- ファイルへアクセスするために使用するオープン・データ・パス (ODP)。
- リモート・ロケーションの情報。
- ソース側とターゲット側の通信用に確立された DDM 会話。
- ソース・ジョブとターゲット・ジョブ。
- ジョブでの入出力操作。

DDM のクライアント・システムとしての IBM i:

これらの処置はすべて、サーバー・システム上で必要な処置も含めて、ソース・プログラムが要求する操作 (ファイルまたは非ファイル) を行う前に、正常に完了している必要があります。

DDM ファイルが参照されると、次のことが行われます。

- 要求がファイルのオープンであれば、そのファイルの情報を使用して、クライアント・システムでのオープン・データ・パス (ODP) の作成と、SDDM サポートの始動が同時に行われます。このサポートは、ソース・プログラムと同じジョブ内で稼働するものです。SDDM は、クライアント・システム要求を DDM 要求に変換したり、適切なサーバー・システムとの通信を実行したり、ソース・ジョブで使用する DDM 会話を確立するといった情報も使用します。(ODP の一部のみ、DDM ファイル情報を使って作成されます。したがって、DDM 会話が確立された後 SDDM が残りの情報の処理を終了しない場合は、ODP を使用することはできません。)
- DDM の通信部分は、サーバー・システムとの通信パスを確立します。サーバー・システム は、DDM ファイル内に指定されるリモート・ロケーション情報で識別され、ターゲット・ファイル は、リモート・ファイル名で識別されます。リモート・ロケーションに関するその他の情報は、DDM ファイルに保持されず、SDDM によって保管されます。この情報には、トランザクション・プログラム名、ユーザー ID、活動化グループ数、および会話の有効範囲が含まれます。TDDM はリモート・ロケーション情報を使ってサーバー・システムで始動され、リモート・システムがプログラム開始要求を受け取ったとき、DDM 会話が確立されます。会話はリモート・ファイルが初めてアクセスされたときに確立されますが、それは、そのサーバー・システムのための同じリモート・ロケーション値を用いる会話が、ソース・ジョブ用として存在しない場合に限られます。
- ファイル関連要求の場合、DDM 会話の確立が終了すると、SDDM (同一ソース・ジョブ内の複数のプログラムと複数の DDM ファイルによる使用が可能) は、DDM 体系コマンドを TDDM に送ります。このコマンドは、実行すべきファイル操作について記述し、アクセスすべきリモート・ファイル (DDM ファイル内で指定された) の名前を含んでいます。リモート・コマンド投入 (SBMRMTCMD) コマンドを使用するような、非ファイル関連要求の場合、リモート・ファイル名は TDDM には送られません。リモート・ファイル名は無視されます。

SDDM は、ファイル・オープンや入出力操作などの各プログラム要求 (DDM ファイルと ODP を使用して受け取る) を、それと同等の DDM コマンド要求に変換してから、サーバー・システムに送ります。

次の図は、リモート・ファイルへのアクセスに関係した、クライアント・システムの基本部分を示しています。

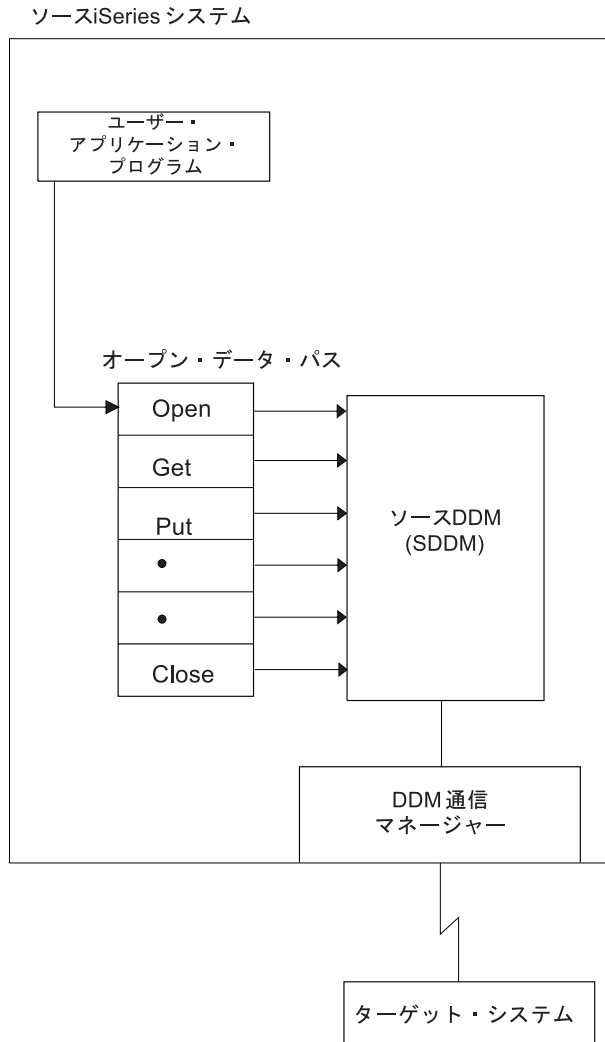


図 13. DDM クライアント・システムとしての IBM i

各要求をターゲット・ジョブが処理し終わると、サーバー・システムから DDM 応答が返されますが、この応答は SDDM によって適切な形式に変換されてから、ユーザーに返送されます。この応答には、データ (データを要求した場合) や状況を示す情報 (他のタイプのファイル・アクセスの場合) が入っていることがあります。ソース・プログラムは、要求した機能の完了および結果の受信待ちになります。

次の図は、リモート・ファイルへの標準的なアクセス要求の場合の、クライアント・システムとサーバー・システム間のデータ交換を単純化して表した例です。

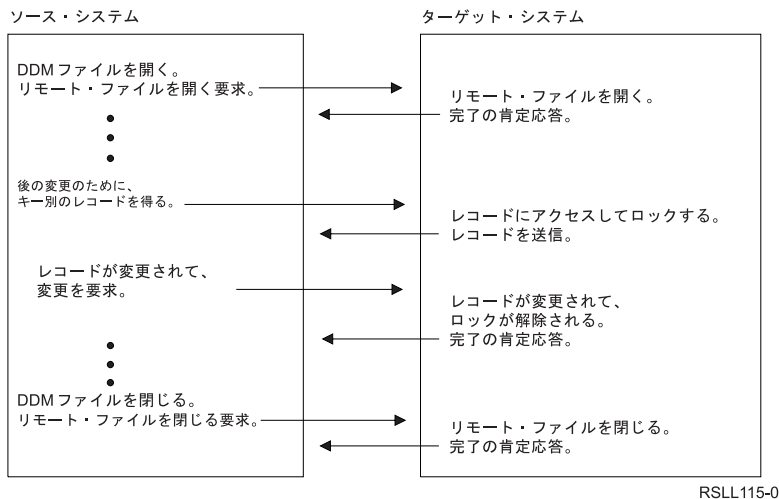


図 14. 入出力操作要求の一般的処理法

ジョブ内でオープンされた最初の DDM ファイルがクローズされても、通常はそのファイルが使用した DDM 会話は活動状態のままです。このため、その同じプログラムまたはそのジョブ内の別のプログラムは、別の DDM ファイルをオープンするときや、他の DDM 関連操作を実行するとき、その同じ会話を使用することができます。(たとえば、37 ページの図 16 にあるとおり、ソース・ジョブ 3A には、同一の会話を使用する 2 つの DDM ファイルがあります。) これで、同一のリモート・ロケーション情報を使用する別々の DDM ファイルをジョブ内で使用する場合は、時間と資源を費やして新たに会話を確立する手間が省けます。

DDM ファイルをクローズしても DDM 会話は活動状態のままですが、SDDM がプログラムからの次の DDM 関連の要求を処理するまでは、その会話では何も起きません。ただし、会話が使用中でなければ、それを除去することができます。除去されるのは、**ジョブ変更 (CHGJOB)** コマンドで DDMCNV ジョブ属性のデフォルト値 *KEEP を *DROP に変更した場合、または、ジョブの活動中に **DDM 会話再使用 (RCLDDMCNV)** コマンドや**資源再使用 (RCLRSC)** コマンドを使用した場合です。

関連概念:

23 ページの『DDM の構成要素: ソース DDM』

クライアント (または、ローカル) システム上のサポートは、DDM 機能を行うために必要に応じてソース・ジョブ内で開始されます。

統合言語環境と DDM:

Integrated Language Environment® (ILE) は、IBM i ジョブ内で実行される活動化グループの概念を導入します。活動化グループとは、実行時ジョブの副構造のことです。

活動化グループは、1 つ以上のプログラムに割り当てられるシステム資源 (プログラムまたはプロシージャの変数、コミットメント定義、およびオープン・ファイル用の記憶装置) によって構成されています。活動化グループは、ジョブ内の小さなジョブのようなものです。デフォルトでの DDM 会話の有効範囲はすべて、活動化グループ・レベルまでです。有効範囲によって、使用できるシステム資源の境界が指定されます。それぞれ異なる活動化グループ内で実行される複数のプログラムの場合、それらが同一 DDM ファイル情報や同一リモート・ロケーション情報を使用するとしても、DDM 会話は別々に始動します。既存 DDM 会話の共用が行われるのは、その活動化グループ内に限ります。OPNDBF コマンド上で OPNSCOPE (*JOB) を指定すれば、DDM 会話の有効範囲をジョブ・レベルにまで広げることができます。

サーバー・システムのタイプに従属するクライアント・システム・アクション:

クライアント・システムが IBM i でもシステム/38 でもない場合は、レベル 2.0 以前の DDM 体系で定義された DDM 体系コマンドのみが使用されます。

クライアント・システムが IBM i またはシステム/38 であれば、そのアーキテクチャーに対する IBM i やシステム/38 の拡張機能が使用され、レベル 2.0 DDM 体系で定義されていない特定の操作がサポートされるようになります。このアーキテクチャーに対するシステム/38 および IBM i 拡張機能の例には、リモート・コマンド投入 (**SBMRMTCMD**) やリモート・ファイルのファイル・メンバーの処理などがあります。クライアントが IBM i であり、サーバーも IBM i のときにファイルを作成する場合は、IBM i 拡張機能が使用されます。

サーバー・システムが IBM i でもシステム/38 でもない場合は、IBM i やシステム/38 で処理できる機能をすべて処理できるとは限りません。例えば、システム/36 では、1 つのオープン操作での相対レコード処理とキー付きレコード処理はサポートしないため、ファイル内のレコードに対して、キーまたは相対レコードによるアクセスが混在しているプログラムは、ファイルがシステム/36 にある場合は機能しません。さらに、レベル 2.0 の DDM 体系をサポートしていないサーバー・システムでは、そのシステムでサポートしているレベルで定義されている機能しか処理できません。

フォルダー管理オブジェクトへのアクセスは、システム/36 でもシステム/38 でもサポートされていません。

注: IBM i オペレーティング・システムでフォルダー管理サービス (FMS) オブジェクトへアクセスできるのは、ストリーム・ファイル (レコード境界のない連続フィールドにデータが読み取られ、また書き込まれるディスク上のファイル) 用とディレクトリー用のレベル 2.0 の DDM 体系 (例: DDM を使用する IBM パーソナル・コンピューター) をクライアント・システムがサポートする場合のみです。

IBM i クライアントは、ストリーム・ファイルとディレクトリーへのアクセスはサポートされません。

DDM のサーバー・システムとしての IBM i:

IBM i ターゲット DDM (つまり TDDM) は、実際には DDM 関連のサーバー・システム・プログラムを実行するジョブです。これは、クライアント・システムがプログラム開始要求を送る (SDDM) と始動します。

クライアント IBM i システムの場合、プログラム開始要求の始動は、IBM 提供の DDM 用のシステム間通信機能 (ICF) ファイルに入っている情報を使用して、クライアント・システムで行われます。アクセスされた DDM ファイル内のリモート・ロケーション情報を使用して、プログラム開始要求は該当するサーバー・システムへ送られます。

ターゲット・ジョブの属性は、**通信項目追加 (ADDCMNE)** コマンドで指定する値によって決定されます。このコマンドは、そのジョブで使用されるサブシステム記述に通信項目を追加するため、サーバー・システムで使用されるものです。またこのコマンドは、サブシステムが使用する装置記述、ジョブ記述 (ターゲット・ジョブのライブラリー・リストも含む)、およびデフォルト・ユーザー・プロファイルを識別します。

IBM i Access Family 接続の場合、DDM 用の QIWS サブシステム内の経路指定項目 (CMPVAL ('DDM')) は、パーソナル・コンピューターの接続先の装置記述も使用して、ターゲット・ジョブの属性を明らかにします。

TDDM は、始動後に以下を行います。

- データベース・ファイルの場合

- APPC、TCP/IP、または IBM i Access Family データ・リンクを使用して確立された DDM 会話を使用して、クライアント・システムとの通信を処理します。
- クライアント・システムからのアクセス要求をそれに相当するシステム機能に変換し、サーバー・システムで実行します。ターゲット・オブジェクトを見つけた後、サーバー・システム作成の ODP とターゲット・データベース管理サービスが使用されて、要求のあった操作のためにそのオブジェクトへのアクセスが行われます。たとえば TDDM は、オブジェクトをオープンした後、そのオブジェクトに対する所定の入出力操作の要求を渡すことができます。
- クライアント・システム (クライアント側が IBM iかシステム/38 の場合) から要求を受信すると、DDM レベル 2.0 アーキテクチャーに対する IBM i またはシステム/38 拡張機能が組み込まれ、これにより、ローカル・システムで機能するほとんどの IBM i 機能が、リモート側の IBM i システムでも機能するようになります。たとえば、現行ターゲット・ジョブ内のライブラリー・リストの置き換えにライブラリー・リスト置換 (RPLLIBL) CL コマンドを使用するなど、非ファイル関連操作を実行する SBMRMTCMD コマンドをクライアント・システム (IBM iまたはシステム/38) から受け取ることができます。
- ターゲット IBM i 応答を同等の DDM 応答に変換し、クライアント・システムに送り返します。クライアント・システムが IBM i またはシステム/38 の場合は、実際の IBM i またはシステム/38 のメッセージがクライアント・システムに送り返されます。
- フォルダー管理サービス・オブジェクトの場合

DDM ストリームとディレクトリーへのアクセス要求を、同等の IBM i フォルダー管理サービス機能に変換し、ターゲット・サーバーで実行します。以下のコマンドがサポートされます。

- 現行ディレクトリー変更 (CHGCD)
- ファイル属性変更 (CHGFAT)
- ディレクトリー・クローズ (CLSDRC)
- 文書クローズ (CLOSE)
- ファイル・コピー (CPYFIL)
- ディレクトリー作成 (CRTDRC)
- ストリーム・ファイル作成 (CRTSTRF)
- ディレクトリー削除 (DELDRC)
- ファイル削除 (DELFIL)
- バッファ強制 (FRCBFF)
- データ・ストリーム GET (GETSTR)
- ディレクトリー項目 GET (GETDRCEN)
- ファイル属性リスト (LSTFAT)
- ストリーム・ファイル・ロード (LODSTRF)
- データ・ストリーム・ロック (LCKSTR)
- ディレクトリー・オープン (OPNDRC)
- 文書オープン (OPEN)
- データ・ストリーム PUT (PUTSTR)
- 現行ディレクトリー QUERY (QRYCD)
- 使用可能スペース QUERY (QRYSPC)
- ディレクトリー名変更 (RNMDRC)
- ファイル名変更 (RNMFIL)

- ストリーム・ファイル・アンロード (ULDSTRF)
- データ・ストリーム・アンロック (UNLSTR)

次の図は、要求された宛先ファイルの処理に関する、IBM i クライアント上の基本部分を示しています。

他のすべてのユーザー APPC、TCP/IP、または IBM i Access Family ターゲット・アプリケーションの場合と同様に、TDDM は別個のバッチ・ジョブとして稼働します。サーバー・システムがソース・サーバー・プログラムから新たに開始要求を受け取るごとに、別のサーバー・システム資源を使って、新たに TDDM が始動されます。各 DDM 会話ごとに、1 つのターゲット・ジョブがあります。それぞれの TDDM は、DDM 会話内で複数のファイルへのアクセス要求を処理することができます。

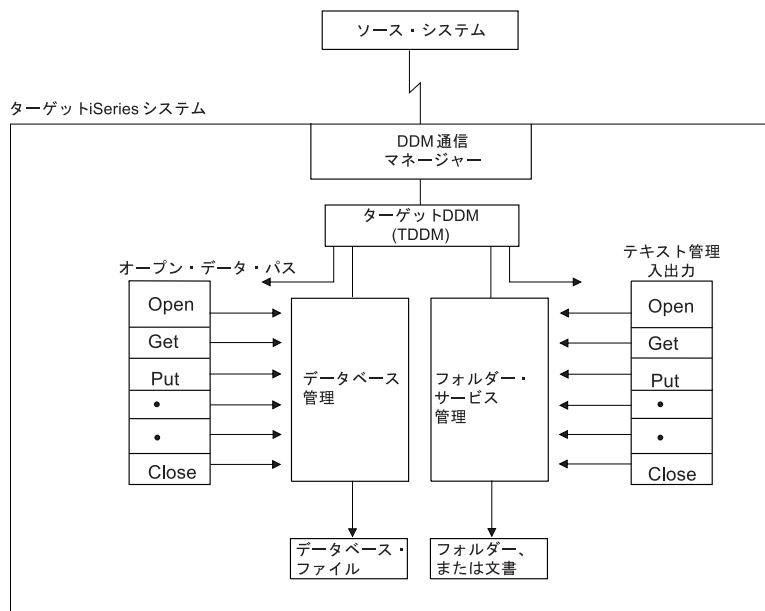


図 15. DDM サーバー・システムとしての IBM i

TDDM が使用するサブシステム、ユーザー・プロファイル、およびサーバー資源は、他のタイプのジョブの場合と同じように定義されます。

関連概念:

23 ページの『DDM の構成要素: ターゲット DDM』

サーバー・システムのジョブは、着信 DDM 要求の結果としてターゲット (または、リモート) システム上で開始され、関連の DDM 会話が終了すると終了します。

DDM 関連ジョブおよび DDM 会話:

このトピックでは、活動化グループ、クライアント・システム・ジョブ、サーバー・システム・ジョブ、およびそれらのジョブが使用する DDM 会話に関する補足説明を行います。

リモート・ファイルの処理を行う場合、少なくとも 2 つの異なるジョブが使用されることになります。それぞれのシステムで実行されるのは、クライアント・システム・ジョブとサーバー・システム・ジョブです。(クライアント・システム・ジョブは、ユーザー・アプリケーションを実行するジョブです。) 1 つのクライアント・システム・ジョブ内の異なる活動化グループにおいて複数のアプリケーション・プログラムを実行することができます。クライアント・システム・ジョブ内の活動化グループごとに、DDM ファイル

内で指定されたりリモート・ロケーション情報を処理するための別々の DDM 会話とサーバー・システム・ジョブがあります。次の条件が満たされると、複数の DDM ファイルは 1 つの会話を共有します。

- ファイルへのアクセスが、クライアント・ジョブ内の同一活動化グループ内で行われる。
- ファイルが、同一リモート・ロケーションの組み合わせを指定している。

DDM 会話ごとに、TDDM を含むサーバー・システム・ジョブが 1 つずつあります。

SDDM は、クライアント・システム・ジョブ内または、クライアント・システム上の活動化グループ内で稼働します。これは、1 つ以上のサーバー・システムとの間で、複数の DDM 会話を同時に扱うことができます。同一クライアント・ジョブや活動化グループの場合には、1 つの SDDM がすべてのリモート・ファイル・アクセス要求を処理します。これは、関与するサーバー・システムやリモート・ファイルの数には関係なく適用されます。システム内には、SDDM 用の別々のジョブはありません。

クライアント・システムの該当 DDM ファイルがすべて、同一のリモート・ロケーション情報を使用してサーバー・システムを識別している場合、TDDM ジョブはサーバー・システム上の 1 つ以上のファイルへのアクセスを要求するクライアント・システム・ジョブごとに、1 つずつ作成されます。

次の図は、6 つの DDM ファイルにアクセスする 5 つのプログラムを示しています。上段の枠内にある DDM ファイルを表す番号は、下段の枠内にある関連リモート・ファイルを表す同じ番号と対応しています。これらの DDM ファイルは、4 つのそれぞれ異なるリモート・ロケーション記述を使用して、すべて同一のサーバー・システム上にある 6 つのそれぞれ異なるリモート・ファイルにアクセスします。この処理を行うには、7 つの DDM 会話が必要です。以下に、DDM 会話について説明します。

- PGM1 と PGM2 はそれぞれ異なるクライアント・ジョブ内で稼働し、同一のリモート・ロケーション情報の入った DDM ファイル (2 および 3) を使用します。各クライアント・ジョブごとに、それぞれ別の会話が必要です。
- クライアント・ジョブ 3 内の PGM3 は、同じリモート・ロケーション情報を用いる 2 つの DDM ファイル (5 および 6) を使用します。これらのファイルは、同じ会話とサーバー・ジョブ (5B) を共有します。
- PGM4 と PGM5 は、クライアント・ジョブ 4 内のそれぞれ異なった活動化グループ内で稼働し、同じリモート・ロケーション情報を用いる 2 つの DDM ファイル (5 および 6) を使用します。活動化グループごとに、それぞれ別の会話が必要です。

次の図では、システム A 内のジョブ 1、2、および 3 に、それぞれ SDDM があります。ジョブ 4 内の各活動化グループには、独自の SDDM があります。ジョブ 1B から 7B までにはそれぞれ、独自の TDDM があります。

アプリケーション・プログラムまたはクライアント・ジョブがクライアント・システム上で DDM ファイルをクローズするとき、下記以外の場合に、DDM 会話および関連したサーバー・ジョブは終了します。

- クライアント・ジョブでの **ジョブ変更 (CHGJOB)** コマンドの DDMCNV 属性の値が、*KEEP (サーバー・デフォルト) の場合。
- ジョブ中に **オブジェクト割り振り (ALCOBJ)** コマンドによって設定されたロックがまだ存在する場合。

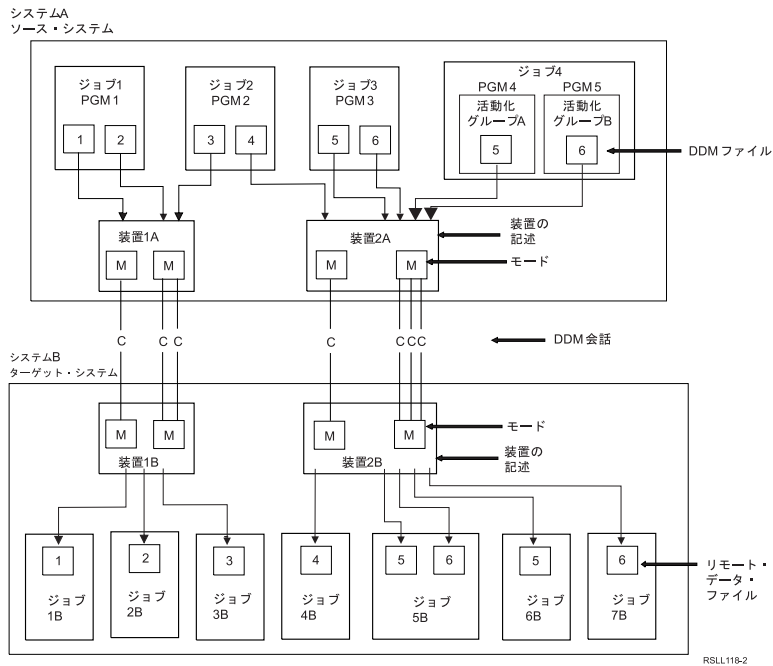


図 16. DDM クライアント・ジョブとサーバー・ジョブとの関係

CHGJOB および **ALCOBJ** コマンドについては、『DDM での CL および DDS の使用』で説明しています。DDMCNV (*KEEP) を指定すると、DDM 会話は活動中のままになり、別の DDM 要求の開始まで待機状態になります。

パフォーマンスの観点から見ると、DDM 会話をもう一度使用するようになっていれば、*KEEP の値を使用する必要があります。これにより、各 TDDM を始動して、会話とジョブを確立するためにサーバ・システムで費やす時間と資源を節約することができます。

次の図は、SDDM と、それぞれ異なったサーバー・システム上の 2 つの TDDM との関係を示しています。『DDM を使用した複数のサーバー上のファイルへのアクセス』のトピックで、SDDM と、1 つのサーバー・システム上の 2 つの TDDM との関係を示しています。

IBM i オペレーティング・システムは、クライアント・システムであると同時にサーバー・システムでもある場合があります。これらのシステムは、相互に相手側のファイルにアクセスすることができます。さらに、IBM i ジョブは、クライアント・ジョブにもサーバー・ジョブにもなります。DDM ファイルは、別の DDM ファイルであるリモート・ファイルを参照することができます。

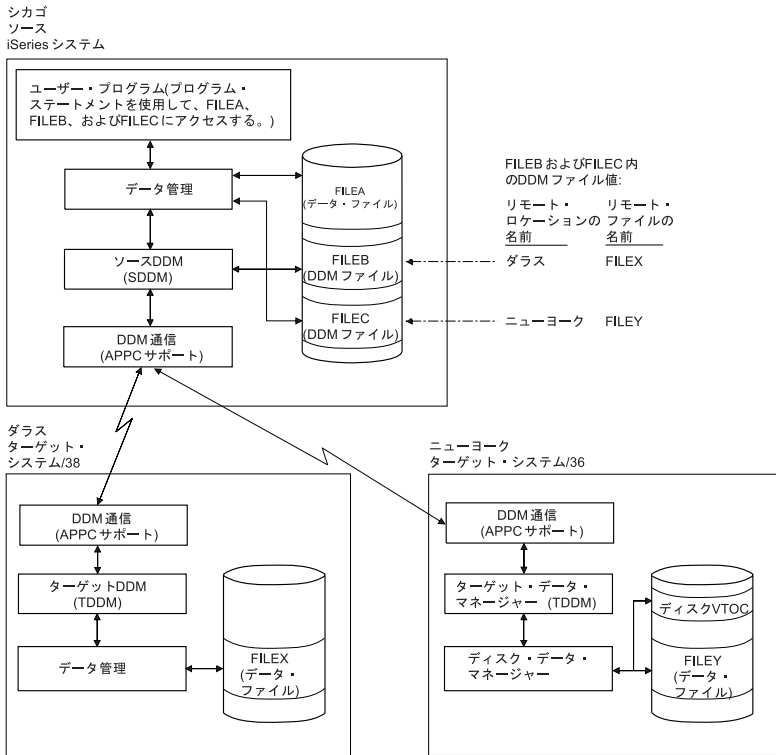


図 17. 例: 複数のローカルおよびリモート・ファイルへのアクセス: システム/38 およびシステム/36 との通信リンクを持った IBM i。

例: DDM を使用した複数のリモート・ファイルへのアクセス

以下に、DDM を使用して単一のアプリケーション・プログラムから複数のリモート・ファイルにアクセスする例を示します。最初の例は、それぞれ異なるサーバー・システムにあるリモート・ファイルを、2 番目の例は、同一サーバー・システムにあるリモート・ファイルを示しています。

例: DDM を使用した複数のシステム上のファイルへのアクセス:

このトピックでは、クライアント・システム、その DDM ファイル、および 2 つのサーバー・システムの相関関係を示した図を示します。

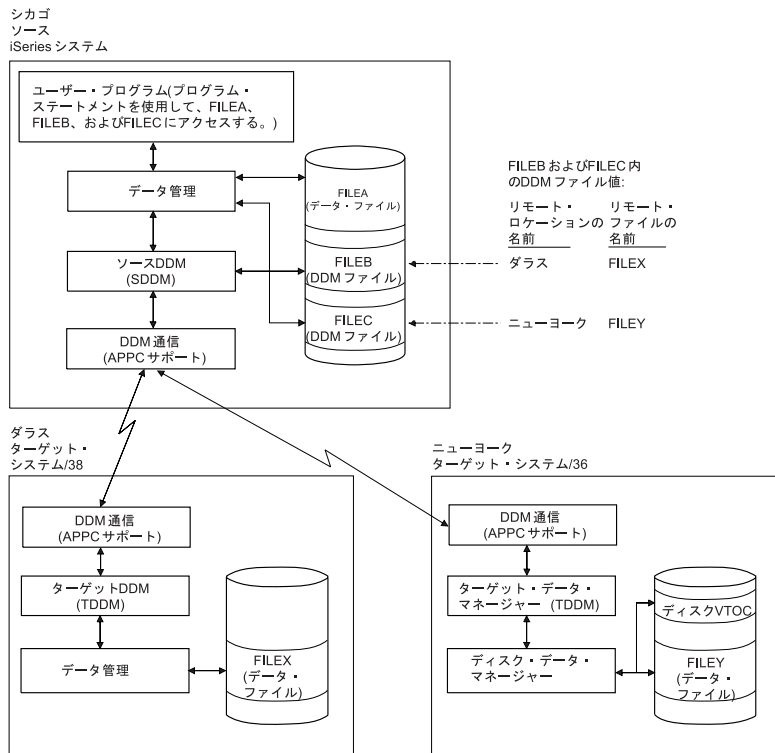


図 18. 例: 複数のローカルおよびリモート・ファイルへのアクセス: システム/38 およびシステム/36 との通信リンクを持った IBM i。

1 つのサーバー・システムはシステム/38 であり、もう 1 つはシステム/36 です。それぞれのシステムには DDM が導入されています。

クライアント・システムにおいて実行されるユーザー・プログラムが、3 つのファイル、つまり FILEA、FILEB、および FILEC にアクセスすることが示されています。クライアント・システムにある FILEA に対しては、ローカル・データ管理機能のみを使用してアクセスします。異なるサーバー・システム上では、DDM ファイル FILEB はリモート・ファイル FILEX に対応し、FILEC はリモート・ファイル FILEY に対応します。プログラムが FILEB と FILEC をオープンすると、プログラム DDM を使用して、クライアント・システムにあるファイルの場合と同様に、この対応するリモート・ファイルにアクセスすることができます。各ファイルのロケーションと、リモート・システムにおけるそのファイル名については、DDM ファイルを定義する担当者が知っていれば十分です。

例: DDM を使用したリモート・ファイルに対する複数の要求の処理:

この例は、複数のプログラムから、同一のサーバー・システムにある複数のファイルにアクセスする方法を示しています。

この例では、システム/36 サーバー・システムが示されています。この場合は、SDDM がそれぞれ異なるジョブで、2 つのファイルを求める、2 つのプログラムからの要求を処理し、2 つの TDDM (要求元プログラムごとに 1 つの TDDM) が、サーバー・システム上でそれらの要求を処理しています。プログラム B はサーバー・システムにある 2 つのファイルにアクセスしようとしていますが、すべての関連 DDM ファイルが、サーバー・システムを識別するのに同一のリモート・ロケーション情報を指定している場合、TDDM は 1 つしか作成されません。

プログラム A と B の両方が、FILEA を共有しています。しかし、これらのプログラムは示されたとおり別々のジョブ内にあるため、FILEA につながる同一のオープン・データ・パス (ODP) を共有することは

きません。プログラム A および B が同一ジョブ内であれば、クライアント・システム上の ODP および リモート・ファイルの両方とも共用することができます。同一ジョブ内の複数のプログラムが同時に 1 つのリモート・ファイルにアクセスする (各プログラムごとに 1 つの TDDM を使用して) ときのリモート・ファイルでのファイル共有の規則は、ローカル・ファイルの場合と同様です。これらの規則は、DDM ファイル作成 (CRTDDMF)、データベース・ファイル指定変更 (OVRDBF)、および DDM ファイル変更 (CHGDDMF) コマンドにどのように SHARE パラメーターが指定されるかに準拠します。

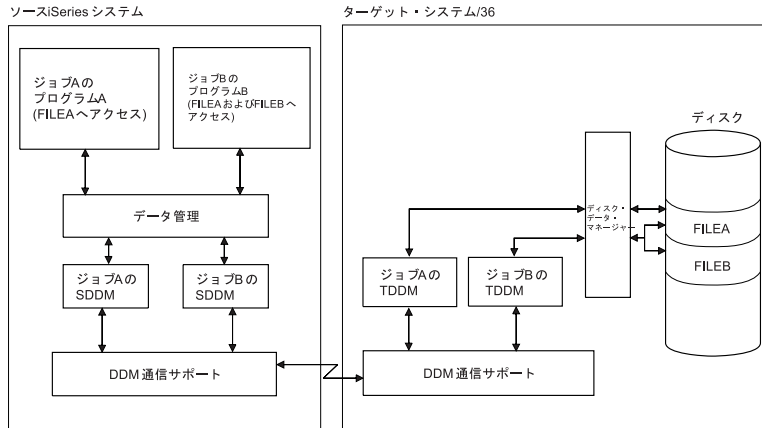


図 19. 例: 複数のプログラムおよびファイル要求の処理

計画と設計

分散リレーショナル・データベースを準備するにあたっては、業務上の要件とリレーショナル・データベース技術の両方を理解していなければなりません。DDM の使用を準備するにあたっては、通信要件、セキュリティ要件、および変更要件を含む複数の要件を満たしている必要があります。

関連概念:

336 ページの『DRDA のパフォーマンスに関する考慮事項』

分散リレーショナル・データベースのパフォーマンスは、データベースの総合的な設計の影響を受けます。分散データの位置、使用するコミットメント制御のレベル、および SQL 索引の設計など、すべてがパフォーマンスに影響します。

DRDA の計画と設計

分散リレーショナル・データベースを正しく操作するためには、計画がその第 1 要件となります。分散リレーショナル・データベースの使用を決定するにあたっては、企業としての必要性および目標を考慮する必要があります。

アプリケーション・プログラムのコーディング方法、データとの関連におけるアプリケーション・プログラムの所在場所、およびアプリケーション・プログラムをデータに接続するネットワーク設計は、いずれも設計上の重要な考慮事項です。

分散リレーショナル・データベースの場合は、単一の IBM i リレーショナル・データベースを扱う場合よりも、データベース設計が重要となります。複数の IBM i 製品を考慮しなければならないので、ネットワーク全般にわたって整合性を持つ管理指針を開発しなければなりません。以下の操作については、方針を作る際に特に注意が必要です。

- 一般的な操作
- ネットワーキング・プロトコル
- システム・セキュリティー
- 会計
- 問題分析
- バックアップおよび回復処理

分散リレーショナル・データベースの要件と要望事項の識別

分散リレーショナル・データベースの要件および要望事項の分析にあたっては、次の項目を考慮してください。

分散リレーショナル・データベースのデータ要件:

分析の最初のステップでは、どのような要因がどのようにデータに作用するかを判別します。

以下の質問項目を検討してください。

- 関与するロケーションはどこか?
- 想定するトランザクションの種類は?
- 各トランザクションでどのようなデータが必要か?
- データ項目の相互の依存関係、とくに参照限界は? たとえば、1つの表の情報を別の表の情報と突き合わせて調べる必要があるか? (その必要があれば、両方の表は同じロケーションに保持しなければなりません。)
- 現在データが存在しているか? 存在している場合は、そのロケーションはどこか? 誰がデータを「所有」しているのか、つまり、誰がデータの正確度の維持管理を担当しているのか?
- 必要なデータの可用性に対する優先度は? ロケーション間での保全本性は? 無許可アクセスからのデータの保護は?
- データに対して想定しているアクセス・パターンは? たとえば、データの読み取りか、更新か、あるいはその両方か? 頻度は? 典型的なアクセスで戻るデータは多いか、少ないか?
- 各トランザクションのパフォーマンスの期待値は? 受け入れることのできる応答時間は?

分散リレーショナル・データベースの能力:

分析の2番目のステップでは、データ要件が分散リレーショナル・データベースによる解決に適しているかどうかを判定します。

データベース処理がほとんどローカルで行われ、リモート・データにアクセスする必要はたまにしかないようなアプリケーションの場合は、分散リレーショナル・データベースの使用が概して適しています。

次のような要件を伴うアプリケーションの場合は、分散リレーショナル・データベースの使用は通常適していません。

- データを中央で保持し、しかも リモート・ユーザーが行う必要のある作業のほとんどが中央で行われる。
- 一貫して高いパフォーマンス、とくに一貫して高速の応答時間を必要とする。ネットワーク内でデータを移動すれば、それだけ時間がかかることとなります。
- 一貫して高い可用性、とくに 24 時間、週 7 日の可用性を必要とする。ネットワークは、それにかかわるシステム、および通信回線や通信制御装置など、中間的な構成要素が多くなり、それだけ故障の機会が増えます。

- 現在利用できないか、まだ発表されていない分散リレーショナル・データベース機能を必要とする。

分散リレーショナル・データベースの目標と方向:

分析の 3 番目のステップでは、短期目標および長期目標の評価を行います。

SQL は、標準的な IBM データベース言語です。目標および動向に、異環境システム間における可搬性またはリモート・データ・アクセスが含まれる場合には、IBM i オペレーティング・システムで分散リレーショナル・データベースを使うようにします。

分散作業単位の分散データベース機能は、DB2 DataPropagator の提供する追加データ・コピー機能と同様に、システムで実行できる処理範囲を広げます。ただし、分散データベース・アプリケーションで、IBM i で現在使用できない機能が必要な場合は、その機能がシステムで使用可能になるまで、他のオプションを使うことができます。たとえば、次のいずれかを行うことができます。

- 必要な機能を独自に用意する。
- 新しい機能が使用可能になるときに備えて、分散リレーショナル・データベースに関する計画を実施する。
- 目標および要件の再評価を行って、現在使用可能または発表済みの機能で満足できないか検討する、代替となる解決法のいくつかを次の表にリストします。これらの代替解決法を用いれば、使用可能な機能を補ったり、あるいは使用可能な機能に代えることができます。

表 3. 分散リレーショナル・データベースの代替解決法

解決法	説明	利点	欠点
分散データ管理機能 (DDM)	オペレーティング・システムの機能の 1 つで、これを使用すれば、1 つのシステムのアプリケーション・プログラムまたはユーザーは、リモート・システムに格納されているデータベース・ファイルを使用できます。この場合、システムは通信ネットワークによって接続されている必要があり、リモート・システムも DDM を使用している必要があります。	<ul style="list-style-type: none"> • 単純な読み取りアクセスおよび更新アクセスでは、パフォーマンスが SQL の場合より優れている。 • 既存のアプリケーションを作成し直す必要がない。 • これを使用すれば、S/38、S/36、および CICS® にアクセスすることができます。 	<ul style="list-style-type: none"> • 複雑な機能では、SQL の方が効率がよい。 • 他の分散リレーショナル・データベース・プラットフォームにアクセスできない。 • CCSID および数値データの変換を行うことができない。

表 3. 分散リレーショナル・データベースの代替解決法 (続き)

解決法	説明	利点	欠点
システム間通信機能/ 共通プログラミン グ・インターフェー ス (ICF/CPI 通信)	ICF は、オペレーティング・システムの機能の 1 つで、これを使用すれば、プログラムは別のプログラムまたはシステムと対話式に通信することができます。CPI 通信は、プログラム間通信を使用するアプリケーションのために、整合性のあるアプリケーション・インターフェースを提供する、呼び出しレベルのインターフェースです。これらのインターフェースは、SNA の論理装置 (LU) 6.2 アーキテクチャーを使い、リモート・システムでのプログラムとの会話を確立し、データを送受し、制御情報を交換し、会話を終了し、パートナー・プログラムにエラーを通知します。	<ul style="list-style-type: none"> • これを使用すれば、アプリケーションをカスタマイズして要件に適合させることができる。 • これによって、より高いパフォーマンスが得られる。 	分散リレーショナル・データベースおよび DDM に比較して、通信およびデータ交換要件をサポートするためにいくぶん複雑なプログラムが必要になる。
ディスプレイ・パス スルー	通信機能の 1 つで、これを使用すれば、ユーザーは IBM i 環境から別の IBM i 環境にサインオンし、そのシステムのプログラムやデータを使用することができる。	<ul style="list-style-type: none"> • リモート・システムのアプリケーションおよびデータがローカル・システムからアクセス可能である。 • データが一時的なものであり、しかも複数のシステムのユーザーが 1 つのシステムのデータを大量に必要とする場合に、迅速なアクセスが可能である。 	画面更新での応答時間がローカル接続装置の場合より遅い。

分散リレーショナル・データベースは、通常、業務上の要件が変化し、新製品が使用可能になるのに応じて、単純なものから複雑なものへと変遷します。要件および要望事項の分析にあたっては、このことを考慮してください。

アプリケーション、ネットワーク、および分散リレーショナル・データベースのデータの設計

分散リレーショナル・データベースの設計には、アプリケーション、ネットワークの考慮事項、およびデータの考慮事項に関する選択が伴います。

ヒント: 分散リレーショナル・データベース・アプリケーションの設計:

分散リレーショナル・データベース・アプリケーションと、ローカル・データベースだけに使用するために開発されたアプリケーションとは、要件に大きな違いがあります。

このような差異について適切に計画するためには、次の考慮事項を念頭に置いてアプリケーションを設計してください。

- 使用できる場合には、分散作業単位 (DUW) 機能を利用してください。
- 共通インターフェースを使用してプログラムをコーディングしてください。

- 複雑なアプリケーションは、それを細かく部分に分割し、分割した各部分をそれぞれの処理に最も適したロケーションに置くことを考慮します。処理するデータが入っているリモート・ロケーションで、ストアード・プロシージャを実行するために SQL CALL ステートメントを使用するのは、アプリケーションで分散処理を行う 1 つのよい方法です。このストアード・プロシージャは、DB2 for i アプリケーション・サーバーで実行するときは、SQL 操作である必要はありません。統合データベース入出力を使用してもよく、別のタイプの処理を実行することもできます。
- 初期のデータベース・アプリケーションの準備、テスト、および使用の方法を研究してください。
- 可能であれば、SQL セット処理機能を活用してください。これによって、アプリケーション・サーバーとの通信が最小限になります。たとえば、この処理が可能な場合、1 つの SQL ステートメントによって、複数の行を更新することができます。
- プログラムの準備時にリモート作業単位 (RUW) 接続方式を使用している場合、または分散アプリケーションの他のノードが DUW をサポートしていない場合は、1 作業単位内のデータベース更新は、単一の場所で行わなければならないことに注意してください。
- DUW 接続方式では、1 つのステートメントを複数のリレーショナル・データベースに送るのに制約があることを忘れてはなりません。
- 接続管理方式の選択によって、パフォーマンスは影響を受けます。異なるリモート・リレーショナル・データベースを切り替えながら使用する必要がない場合は、RUW 接続管理方式を使用するのがよいと思われます。これは、DUW 接続管理で使用される 2 フェーズ・コミットでは、負担が大きくなるからです。

ただし、複数のリモート・データベース管理システム間を頻繁に切り換える必要がある場合は、DUW 接続管理を使用してください。DUW 接続管理を使用して実行しているときは、あるデータベース管理システムとの通信会話は、もう 1 つのデータベース管理システムに接続を切り換えるときに、終了させる必要がありません。同種環境では、これは異環境ほど大きな影響はありません。同種環境での会話は、デフォルトの DDMCNV(*KEEP) ジョブ定義属性を使用することによって、活動状態のままにしておくことができるからです。ただし、同種環境でも、新しい接続を設定するために、カーソルをクローズし通信フローを送信するコストを除くために、DUW を使用してパフォーマンスを上げることができます。

- 接続管理方式は、CONNECT ステートメントの働きを決定します。RUW 接続管理方式では、CONNECT ステートメントは、リレーショナル・データベースへの新しい接続を設定する前に、既存の接続があればそれらを終了させます。DUW 接続管理方式では、CONNECT ステートメントは、既存の接続を終了させません。

分散リレーショナル・データベースのネットワークの考慮事項:

ネットワークの設計は、分散リレーショナル・データベースのパフォーマンスに直接影響します。

特定のネットワークで十分に機能する分散リレーショナル・データベースを適切に設計するには、以下のことを行います。

- アプリケーションのパフォーマンスにとって回線速度が非常に重要になることがあるので、ネットワーク内の適切な個所に十分な容量を用意して、主な分散リレーショナル・データベース・アプリケーションに効率的なパフォーマンスが達成できるようにしてください。
- 使用可能な通信ハードウェアおよびソフトウェアを評価し、必要に応じて、上位移行能力も評価を行ってください。
- 拡張プログラム間通信 (APPC) 接続の場合、ネットワークの定義時に指定するセッション限界および会話限界を考慮してください。

- 必要とされる (テスト環境および実稼働環境の両方で) ハードウェア、ソフトウェア、および通信機器、ならびに分散リレーショナル・データベース・ネットワークの機器の最も優れた構成を識別してください。
- TCP/IP をサポートするために必要なスキルは、APPC をサポートするために必要なスキルとは異なっていることを考慮してください。
- エンド・ユーザー・グループとの間の初期サービス・レベルに関する同意 (所定の分散リレーショナル・データベース・アプリケーションの場合に予測される応答時間など)、ならびに提供される実際のサービスの監視および調整についての方針を考慮に入れてください。
- 現在のスレッドで補助記憶域プール (ASP) グループに設定されているアプリケーション・リクエスト (AR) からデータベースにアクセスする際に、APPC 保護 DUW 会話は使用できないことを理解していなければなりません。
- 分散リレーショナル・データベースのデータベース・オブジェクト、および分散リレーショナル・データベースの各ロケーションでの命名についての方針を決めてください。ロケーションとは、分散リレーショナル・データベースに関与するリレーショナル・データベース管理システムの相互接続ネットワークの中にある、特定のリレーショナル・データベース管理システムのことです。この意味でのロケーションは、独立 ASP グループで構成されたシステム内のユーザー・データベースを指すこともあります。この方針を作成するにあたっては、次の項目を考慮してください。
 - 分散データベースのオブジェクトの完全修飾名には 3 つ (2 つではなく) の部分があり、最上位の修飾子でオブジェクトのロケーションを識別します。
 - 分散リレーショナル・データベースの各ロケーションには、それぞれ固有の識別名を与えてください。データベースの各オブジェクトにも、それぞれ固有の識別名が必要です。識別名が重複していると、重大な問題を生じることがあります。たとえば、ロケーション名およびオブジェクト名が重複していると、アプリケーションが意図したもの以外のリモート・データベースに接続される場合があります。しかもいったん接続された場合には、意図したもの以外のオブジェクトにアクセスすることになります。ネットワークの結合時には、命名に特別の注意を払ってください。
 - また、ユーザー・データベースでの各ロケーションには、固有の識別名を与えなければなりません。PAYROLL という名前のユーザー・データベースが 2 つのそれぞれ別個のシステムにある場合、アプリケーションが同じシステムからそれらの両方にアクセスしなければならないときに、名前に関する競合が発生します。独立 ASP 装置を構成するときに、ユーザーはその ASP 装置自体の名前とは異なる RDB 名をその装置に指定できます。これが、ASP グループ内で基本装置に関連付けられている RDB 名であり、これによって、そのユーザー・データベースは認識されます。

関連概念:



通信管理 (PDF)

分散リレーショナル・データベースのデータの考慮事項:

データを必要とするアプリケーションとの関連におけるデータの配置は、分散リレーショナル・データベースの設計時の重要な考慮事項になります。

配置に関する決定にあたっては、次の項目を考慮してください。

- アプリケーションで必要とするパフォーマンスのレベル。
- ロケーション間でのデータのセキュリティ、現行性、整合性、および可用性に関する要件。
- 必要とされるデータの量およびデータ・アクセスの予測パターン。
- 必要とされる分散リレーショナル・データベース機能が使用可能であるかどうか。
- システムをサポートするのに必要な技能と実際に使用可能な技能。

- 誰がデータを「所有」しているのか、つまり、誰がデータの正確度の維持管理を担当しているのか。
- システム間セキュリティー、会計、監視と調整、問題処理、データのバックアップと回復、および変更制御に関する管理方針。
- データをネットワーク内のどこに置くか、維持管理するデータのコピーは 1 つにするか複数にするかなど、分散データベース設計に関する決定事項。

分散リレーショナル・データベースの管理方針の作成

分散リレーショナル・データベースを管理する場合、以下の方針に留意してください。

分散リレーショナル・データベースの一般的な操作:

分散リレーショナル・データベースの一般的な操作の計画にあたっては、パフォーマンスと可用性の両方を考慮してください。

以下に設計上の考慮事項が挙げられてありますが、分散リレーショナル・データベースのパフォーマンスと可用性の両方の向上を図る上で役立てることができます。

- 実行頻度が高いか、またはデータの送受信量が多いトランザクションを伴うアプリケーションの場合には、データと同じロケーションにアプリケーションを保持するように努めてください。
- 異なるロケーションにあるアプリケーションによる共有の必要があるデータの場合は、活動の最も多いロケーションにデータを置いてください。
- 1 つのロケーションにあるアプリケーションと別のロケーションにあるアプリケーションが、同程度にデータを必要とする場合には、データのコピーを両方のロケーションに保持することを考慮してください。複数のロケーションにコピーを保持する場合には、管理方針に関する以下の質問事項を検討してください。
 - コピーに更新を行うことをユーザーに許可するか?
 - 最新のデータによるコピーの更新の方法および時期は?
 - コピーのバックアップは、すべてのコピーについて行わなければならないのか、コピーの 1 つについて行えば十分なのか?
 - 一般管理活動をすべてのコピーに関して一貫して行う方法は?
 - コピーの 1 つを削除することが許される時期は?
- 分散データベースの管理は中央ロケーションによるのか、各データベース・ロケーションによるのかを考慮してください。

以下によってもパフォーマンスを向上させることができます。

- データとアプリケーションを別々のロケーションに保持しなければならない場合には、次のようにして、パフォーマンスが許容限度内に収まるようにします。
 - アプリケーションで使用されるデータの列の検索だけに留めることによって、ネットワーク内のデータ通信量をできるだけ低く抑えてください。つまり、SELECT ステートメントの一部として列名のリストの代わりに * を使用することは避けてください。
 - リモート・ロケーションとの間で大量のデータの送信または受信を行うステートメントのコーディングを、プログラマーに行わせないようにしてください。つまり、SELECT ステートメントの WHERE 文節の使用を奨励して、データの行数を制限するようにしてください。
 - 参照保全、トリガー、およびストアド・プロシージャ (リモート・リレーショナル・データベース管理システムに対する CONNECT ステートメントの後の SQL CALL ステートメント) を使用してください。これによって、アプリケーション・サーバー (AS) への処理が分散され、したがって実質的な回線通信量が減り、パフォーマンスが改善されます。

- FOR FETCH ONLY 文節を指定することによって、読み取り専用照会を適宜使用してください。
- 照会をブロック化する規則に注意してください。たとえば、IBM i オペレーティング・システム間の照会では、COMMIT(*NONE) についてだけ、あるいは ALWBLK(*ALLREAD) が指定されている場合には COMMIT(*CHG) と COMMIT(*CS) についてだけ、読み取り専用データのブロック化が行われます。
- 可能な限りリモート・データではなくローカル・データを使用することによって、アクセス・データに対するアクセス数を低く抑えてください。
- SQL SET 命令を使用して、単一の SQL 要求でアプリケーション・リクエスターの複数行を処理してください。
- リモート作業単位 (RUW) 接続管理で実行する時か、または分散作業単位 (DUW) 接続管理で実行する場合は、DDMCNV(*KEEP) を使用することによって接続の停止を避けるようにしてください。
- 以下のようにして、十分なネットワーク容量を用意します。
 - 高速、高帯域幅の回線を導入することによって、あるいはネットワーク内の適当な点に回線を追加することによって、ネットワークの容量を増やしてください。
 - 競合を減らすか、または特定の処理装置の競合バランスを改良してください。たとえば、既存のアプリケーションをホスト・システムから部門システムに移動するか、または分散リレーショナル・データベースの作業をグループ化してバッチ処理します。
- 優れたテーブルの設計を奨励します。分散リレーショナル・データベース・ロケーションで、基本キー、テーブルの索引、および正規化技法の適切な使用を奨励してください。
- WHERE 文節で使用されるホスト変数のデータ・タイプが、対応するキー列データ・タイプの中のデータ・タイプと矛盾しないようにしてください。たとえば、浮動小数点のホスト変数は、異なるデータ・タイプの列に構築された索引では使用できません。

以下によっても可用性を向上させることができます。

- 一般的には、ネットワーク内のデータ通信量を制限するようにしてください。
- データとアプリケーションを別々のロケーションに保持しなければならない場合には、次のようにして、可用性が許容限度内に収まるようにします。
 - 代替ネットワーク経路を確立してください。
 - 時間帯の違いが可用性に及ぼす影響を考慮してください。
 - システムを始動する適格者が確保できるか?
 - 時間外のバッチ作業が処理の障害になるか?
 - 優れたバックアップおよび回復機能を確保してください。
 - 誰もがバックアップおよび回復に熟練するようにしてください。

分散リレーショナル・データベースのセキュリティーの考慮事項:

分散リレーショナル・データベース計画の一部として、分散データの保護について決定しなければならない事項があります。

そのような決定事項には以下のものがあります。

- 他のロケーションのユーザーにとってアクセス可能なシステムがどれで、それらのシステムに対してアクセスができる他のロケーションのユーザーが誰なのか。
- システムに対するアクセスを規制する厳しさの程度。たとえば、リモート・ユーザーによる会話の開始時には、ユーザー・パスワードを必須とするのか?
- パスワード入力を暗号化された形式で回線上に流すのか?

- クライアント・ジョブを実行するときのユーザー・プロファイルを、接続先のリレーショナル・データベースの名前に応じて、別のユーザー識別コードまたはパスワードへ割り当てる必要があるか?
- 他のロケーションのユーザーにとってアクセス可能なデータがどれで、そのデータに対してアクセスができる他のロケーションのユーザーが誰なのか。
- ユーザーがデータに対して講じることを許される処置。
- データに対する権限の認可は中央管理かローカル管理か。
- 複数のシステムがリンクされるため、特別の予防策を講じるべきかどうか。たとえば、名前変換を使用するのか?

以上の決定を行うにあたっては、ロケーションの選択時に次の項目を考慮してください。

- 物理的な保護。たとえば、利用制限を設けた部屋をロケーションに用意することができます。
- システム・セキュリティのレベル。システム・セキュリティのレベルは、ロケーションによって異なる場合があります。分散データベースのセキュリティ・レベルは、ネットワークで使用される最低レベルのセキュリティ以下になります。

拡張プログラム間通信機能 (APPC) によって接続されるすべてのシステムでは、以下の処理が可能です。

- 両方のシステムが IBM i 製品である場合、暗号化された形式のパスワードで通信する。
- 1 つのシステムがネットワーク内の別のシステムと通信するための要求を受信した場合に、要求側システムが実際に「自称どおり」であり、しかも受信側システムと通信する権限を認可されていることを検査する。

すべてのシステムで次の項目を行うことができます。

- リモート・データ・アクセスの許可に先立って、検査のために、ユーザーの識別名およびパスワードをローカル・システムからリモート・システムへ渡す。
- テーブルやビューなどの SQL オブジェクトをアクセスおよび操作する特権の認可および取り消しを行う。

IBM i オペレーティング・システムには、セキュリティ監査機能が組み込まれており、これを使用すれば、無許可でデータにアクセスしようとする試みの追跡、ならびにセキュリティにかかわるその他のイベントの追跡を行うことができます。さらに、システムでは、リモート・システムによる分散データベースへのアクセスをすべて防止できる機能も用意しています。

- セキュリティに関連したコスト。セキュリティのコストを考慮するにあたっては、セキュリティ関連製品の購入コストと、情報関係スタッフが次の活動を行うための時間コストの両方を考慮してください。
 - リモート・データにアクセスするユーザーの識別名を、ローカル・システムとリモート・システムの両方で維持管理する。
 - 監査機能をサイト間で調整する。

関連概念:

86 ページの『セキュリティ』

IBM i オペレーティング・システムには、サーバーのデータ資源に対するアクセスを制限するセキュリティ要素が組み込まれています。セキュリティ・オプションの範囲は、単純な物理的セキュリティから、コマンドおよびデータ・オブジェクトに対する許可と組み合わせられた、完全なパスワード・セキュリティにまで及んでいます。

分散リレーショナル・データベースのアカウント:

分散データの使用に関する会計および請求が可能でなければなりません。

以下の事項を考慮してください。

- 分散データの使用に関する会計では、1 つ以上のリモート・システムのリソースの使用、ローカル・システムのリソースの使用、およびシステムを接続するネットワーク・リソースの使用が対象になります。
- 会計情報は、システムごとに独立して累積されます。ネットワーク会計情報は、システムで累積されるデータとは別に累積されます。
- さまざまなシステムの時間帯については、会計情報を相関付けようとするときに、考慮に入れなければならないことがあります。各システム・クロックは、リモート・システム・クロックに同期化していません。
- 各システムの許容会計コード (番号) の間には、相違点が存在していることがあります。たとえば、IBM i オペレーティング・システムでは、会計コードを最大 15 文字に制限しています。

分散データの使用に関する会計では、次の機能が使用可能です。

- IBM i ジョブ会計ジャーナル。システムでは、各分散リレーショナル・データベース・アプリケーションごとに、ジョブ会計情報をジョブ会計ジャーナルに書き込みます。ジャーナル表示 (DSPJRN) コマンドを使用すると、ジャーナル項目をデータベース・ファイルに書き込みます。次に、ユーザー作成プログラムと照会機能のどちらかを使用すれば、会計データを分析することができます。
- NetView® 会計データ。NetView ライセンス・プログラムを使えば、ネットワーク・リソースの使用に関する会計データを記録することができます。

関連資料:

ジャーナル表示 (DSPJRN) コマンド

281 ページの『分散リレーショナル・データベースのジョブ会計』

IBM i ジョブ会計機能は、データを収集して、誰がシステムを使用し、どんなシステム・リソースを使用しているかを判別できるようにします。

分散リレーショナル・データベースの問題分析:

分散データベース環境で問題分析を管理する必要があります。問題分析には、システムのネットワーク内で処理されるアプリケーションに関する問題の識別と解決の両方が含まれます。

以下の事項を考慮してください。

- 分散データベース処理の問題はさまざまな仕方で現れます。たとえば、問題を検出したシステムによって、エラー戻りコードが分散データベース・アプリケーションに渡される場合があります。さらに、応答が遅かったり、間違っていたり、あるいは存在しない場合もあります。
- 分散データベース処理の問題を診断するためのツールが使用可能です。たとえば、各分散リレーショナル・データベース・プロダクトごとに、分散データ処理の問題を診断に役立つトレース機能が用意されています。
- IBM i オペレーティング・システムがシステム障害を検出した場合、このシステムは、障害の検出直後に、プログラム状況に関する情報をログ記録します。

分散リレーショナル・データベースのバックアップおよびリカバリー:

単一システム環境では、バックアップおよび回復はローカルで行われます。しかし、分散データベースでは、バックアップおよび回復はリモート・ロケーションにも影響します。

IBM i オペレーティング・システムでは、個々のテーブル、コレクション、またはコレクション・グループをバックアップおよび回復することができます。バックアップおよび回復が可能なのはローカルでだけです。十分なバックアップ・サポートを持っていないシステムには、重要性の低いデータを配置することもできます。バックアップおよび回復手順は、複数のアプリケーション・サーバー上に存在するデータに対し、一貫していなければなりません。ネットワークには複数のシステムがあるので、そのようなデータを 2 番目のシステムにも保管して、何らかの形式で常にネットワークで使えるようにしておくことができます。こうした方針は、データベースをネットワーク内で分散する前に、特に計画および展開する必要があります。

DDM の計画と設計

分散データ管理機能 (DDM) を正しく使用するには、いくつかの要件を満たす必要があります。

注:

- DDM を使用してアクセスするファイルを判別する前に、『DDM のパフォーマンスに関する考慮事項』を検討してください。
- 制御言語 (CL) コマンドとデータ記述仕様 (DDS) のプログラミングに関する要件および考慮事項については、『DDM での CL および DDS の使用』と『DDM の操作に関する考慮事項』で説明しています。

関連概念:

18 ページの『DDM の概要』

このトピックでは、分散データ管理機能 (DDM) の目的、DDM が提供する各種機能、および IBM i DDM の概念について説明します。

337 ページの『DDM のパフォーマンスに関する考慮事項』

このトピックには、DDM を使用するときパフォーマンスを向上させるための参考情報と、ある種の機能を実行するのに DDM 以外のものを使用する場合に関する情報が記述されています。

295 ページの『DDM の操作に関する考慮事項』

このトピックでは、IBM i オペレーティング・システムが、クライアントまたはサーバー・システムとして、どのように別の IBM i と通信してリモート・ファイル処理を行うかについて述べています。また、IBM i が、IBM i 以外のシステムと通信する場合の相違点についても述べます。

173 ページの『DDM での CL および DDS の使用』

このトピックでは、特定の制御言語 (CL) コマンドについての DDM 関連の情報、データ記述仕様 (DDS) に関する考慮事項、DDS キーワード、および DDM ユーザー・プロファイル権限について説明します。

APPC ネットワークの DDM の通信要件

OptiConnect を使用していない分散データ管理機能 (DDM) ネットワーク内の各 IBM i 製品は、以下の通信要件を満たしていなければなりません。

- APPC/APPN サポートまたは IBM i Access Family ライセンス・プログラムが、システムに導入され構成されていなければなりません。
- 少なくとも 1 つのシステム・ネットワーク体系 (SNA) 通信回線が、同期データ・リンク通信 (SDLC)、トークンリング・ネットワーク、イーサネット、または X.25 プロトコルを使用しなければなりません。

DDM 会話で使用できるセッション数に対する制限は、DDM にはありません。最大数は、他のすべての APPC 関連通信の場合と同じように決定されます。並列セッションでは、最大セッション数は、モード内で指定します。単一セッション装置の場合、最大セッション数は常に 1 です。

OptiConnect を使用している DDM ネットワーク内の IBM i 製品には、OptiConnect ソフトウェアおよびハードウェアが導入されていなければなりません。OptiConnect によって、SNA 通信回線接続は必要なくなります。

TCP/IP ネットワークの通信ネットワークの構成

このトピックでは、TCP/IP ネットワークをセットアップするために必要なステップの概要を示します。

1. 使用する IBM i プラットフォームをローカル・ネットワーク (システムと直接接続されたネットワーク) に識別します。
 - a. 回線記述がすでに存在しているかどうかを調べます。
 - b. 回線記述がまだない場合は、作成します。
 - c. TCP/IP インターフェースを定義して、システムに IP アドレスを与えます。
2. TCP/IP 経路を定義します。これにより、システムはリモート TCP/IP ネットワーク (IBM i プラットフォームと直接接続されていないネットワーク) 上のシステムと通信できるようになります。
3. ネットワーク内のサーバーの名前を識別します。
 - a. ローカル・ホスト・テーブルを作成します。
 - b. リモート・ネーム・サーバーを識別します。
4. TCP/IP を開始します。
5. TCP/IP が動いているかどうかを検査します。

DDM の機密保護要件

システムのデータ資源への分散データ管理機能 (DDM) ユーザーからの意図的なあるいは意図的でないアクセスを防止することができます。

DDM 環境のデータへのアクセスは、システム・レベルのネットワーク属性 (ネットワーク属性変更 (CHGNETA) コマンドの DDMACC パラメーター) を使用して、限定したり、まとめて防止したりすることができます。この属性を使用して、システム (サーバー・システム側) は、すべてのリモート・アクセスを防止できます。またこの属性を使用して、システムはファイルに対する標準権限を使用してファイル・アクセスを制御することができ、さらに、オプションのユーザー・出口プログラムを使用して、特定ユーザーのファイルに許される操作のタイプを限定したりすることができます。

十分な機密保護を講じるには、1 つ以上のサーバー・システム・ファイルへアクセスできる各クライアント・システム・ユーザーごとに 1 つずつ、その他のユーザー・プロファイル、サーバー・システム上で設定しなければなりません。そうでなければ、複数のクライアント・システム・ユーザー用に、デフォルト・ユーザー・プロファイルを設けなければなりません。デフォルト・ユーザー・プロファイルは、ターゲット・ジョブが実行されるサブサーバー内で使用する通信項目によって決められます。

IBM i を実行していないサーバー・システムにおけるユーザー・プロファイル (またはそれと同等のもの) については、そのシステムの資料を参照してください。

DDM のファイル要件

IBM i 製品からリモート・ファイルにアクセスするには、その前にクライアント・システムで分散データ管理機能 (DDM) ファイルを作成する必要があります。

APPN を使用していないのであれば、DDM ファイルを使用するときに、その DDM ファイルで指定された装置 (リモート・ロケーション名) とモード (APPC セッションの特性) が共にシステム上になければなりません。APPN を使用するのであれば、その装置がシステム上になくてもかまいません。ただし、リモート・ロケーション名で識別されるシステムは、APPN ネットワーク内に存在していなければなりません。

ん。APPC 制御装置記述作成 (CRTCTLAPPC) および SNA ホスト制御装置記述作成 (CRTCTLHOST) コマンドの APPN パラメーターで、APPN を使用するかどうかを制御します。

DDM のプログラム変更要件

リモート・ファイルは、高水準言語 (HLL) および制御言語で作成されている IBM i アプリケーション・プログラムからアクセスできます。

ほとんどの場合、この種のアプリケーションでは、プログラムを変更しなくても、ローカル・ファイルにも、またリモート・ファイルにもアクセスすることができます。ただし、考慮事項や制約事項によっては、プログラムを変更して再コンパイルすることが必要な場合があります。このような事項は、次のカテゴリにグループ化されています。

- DDM 体系ではサポートされていないが、そのアーキテクチャーに対してはシステム/38 の拡張機能が設けられている各種 IBM i 機能。このような機能を使用できるのは、クライアント・システムとサーバー・システムがシステム/38 または IBM i を実行している場合だけです。
- クライアント・システムまたはサーバー・システムが IBM i を実行しているときに関与する制約事項と考慮事項。
- すべてのターゲット・システムに関与する制約事項と考慮事項。ローカル・ファイルにアクセスするユーザー・プログラムの場合、「レコードが見つからない」、「ファイルの終わり」、および「更新のための読み取り中にレコード・ロックがタイムアウト」などの異常事態に備えてプログラムする必要があります。このような事態は、DDM を使用してリモート・ファイルにアクセスするときにも生じることがあります。また、DDM を使用すると、ディスク入出力命令の送信時に、プログラムで通信回線障害が生じることがあります。

通信障害が生じるとシステムは、該当するメッセージをジョブへ送りますが、これは、一般的なファイル・エラーとしてアプリケーション・プログラムへ返されます。いずれの高水準言語にも、ディスク操作の例外結果に対してユーザー制御処理またはデフォルト処理が行えるよう、固有のユーザー構文機能が備わっています。言語によっては、個々に DDM 通信障害を解明するためのジョブ・メッセージ識別コード (ID) を、ユーザーが検索することになっています。個々の機能に関しては、該当する言語の資料を参照してください。

SDLC 2 次回線の場合、クライアント・システムとサーバー・システムで SDLC回線記述作成 (CTRLNSDLC) コマンドで INACTTMR パラメーターを設定して、1 次システムによるポーリングの停止を検出できるようにすることをお勧めします。これによって、1 次システムで通信障害があっても、DDM の更新のための読み取りのレコード・ロックが無限に続かないようにすることができます。

DDM 体系に関連した制約事項:

このトピックでリストする項目は、DDM 体系に関連した制約事項です。したがって、これらの事項が関与するアプリケーション・プログラムが、リモート・ファイルへアクセスするときには、変更して再コンパイルしなければならないことがあります。

- DDM 体系は、多様式の IBM i 論理ファイルをサポートしません。しかし、多様式論理ファイルは、DDM 体系に対するシステム/38 拡張機能としてサポートされるため、DDM と一緒に使用することはできますが、これはソース・システムとサーバー・システムが IBM i またはシステム/38 を実行している場合に限られます。
- 外部記述データ (IBM i 上でデータ記述仕様 (DDS) を使用して記述されたもの) は、DDM 体系ではサポートされません。ただし、特に両方のシステムとも IBM i またはシステム/38 を実行している場合に

は、DDS を引き続き使用できます。サーバー・システムが IBM i またはシステム/38 を実行している場合は、リモート・ファイルをローカル・ファイルと同じように扱って、ほとんどの DDS サポートを使用できます。

- フォルダー管理サービス・オブジェクトにアクセスするには、クライアント・システムが、ストリーム・ファイル用のレベル 2.0 またはレベル 3.0 の DDM 体系およびストリーム・アクセス方法をサポートしていなければなりません。バイト・ストリーム・モデルの場合、以下の制約事項があります。
 - データ・ストリーム・ロック (LCKSTR) コマンドでは、待ち時間はフォルダー管理サービスによってサポートされません。ユーザーがクライアント・システムの待機機能を処理する必要があります。
 - IBM i オペレーティング・システム上で文書をコピーするのに使用されるファイル・コピー (CPYFIL) コマンドは、制限付きでサポートされます。見出しヘッダーのみがコピーされ、データはコピーされません。
 - DELDR COP (DRCALL) パラメーターは、ディレクトリー削除 (DELDR C) コマンドではサポートされません。
- ファイル、ライブラリー、またはメンバーなどのデータ管理オブジェクトを処理するとき、パーソナル・コンピューターの総称名を使用することはできません。ただし、文書やフォルダーなどのフォルダー管理サービス・オブジェクトを処理するときには、総称名を使うことができます。パーソナル・コンピューターが該当操作をサポートしていれば、パーソナル・コンピューターがその操作をサポートするように、総称名がサポートされます。たとえば、名前変更コマンドと削除コマンドを使用する時、フォルダーの総称名はサポートされませんが、それは、パーソナル・コンピューターが総称名をサポートしないためです。

DDM の IBM i クライアントおよびサーバーの制約事項と考慮事項:

クライアント・システムが IBM i を実行している場合、リモート・ファイルに対して IBM i データベース機能を使用できます。しかし、いくつかの制約事項があります。

制約事項は、次のとおりです。

- IBM i クライアント・システムは、システム/38 を実行するシステム上でファイルを作成することはできませんが、この場合は DDM 体系ファイル・モデルが使用されます。このため、システム/38 を含め、IBM i を実行していないサーバー・システムでは、多様式論理ファイルや結合論理ファイルを作成することはできません。
- 保管操作や復元操作を行っても、サーバー・システムにデータを保管したり復元することはできません。DDM ファイル・オブジェクトのみ、ローカルで保管または復元することができます。
- 操作上のある一定時間の遅延 (すなわち、ファイルまたはレコードの待ち時間) は、サーバー・システムで指定された時間値で決まります。この値は、各種 CL コマンドの WAITFILE パラメーターおよび WAITRCD パラメーターで指定します。このため、リモート・ファイルやレコードにアクセスするのに DDM を用いたときに、遅延時間が長くなってしまうこともあります。
- システム/38 に対する QUERY 要求 (OPNQRYF) に、グループ選択や結合処理を使用することはできません。
- IBM i との間でシステム/36 アプリケーション・プログラムを実行している場合、資源が使用可能になるのを待っている間に、これらのアプリケーション・プログラムがタイムアウトになる可能性があります。システム/36 アプリケーション・プログラムをもう 1 つ別のシステム/36 に対してまたはそのシステムから実行中の場合、アプリケーション・プログラムが資源が使用可能になるまで待機する時間に制限はありません。

ソースおよびターゲット DDM ジョブでは、DDM からの APPC 命令の送信の仕方に起因する、回線障害やその他の障害がリモート・システムで起きた後に、APPC 会話の 2 次側の DDM ジョブが際限なく待機することがあります。

無限待機を防ぐためには、以下の提案を参考にしてください。

- リモート・システムがレコード・ロック・タイムアウトをサポートしているなら、必ず妥当な時間値を指定してください。例えば、CRTPF コマンドに WAITRCD パラメーターを指定する場合には、ターゲットの IBM i またはシステム/38 データベース・ファイルに、使用可能な最大値または *NOMAX は使用しないでください。

WAITRCD は更新のための読み取り操作に対して使われますが、読み取り専用や追加などのその他のファイル操作には使用できません。

- SDLC 2 次回線を使用するときには、非アクティブ・タイマー (INACTTMR) 用の時間値を使用します。*NOMAX 値は使用しないでください。
- システム操作の担当者に、関連回線名、制御装置名、および装置名 (または、実行しようとする DDM ジョブのリスト) を知らせておきます。これにより、DDM ジョブが無限に待機していると思えるときには、その担当者がジョブ情報を表示させて、そのジョブの処理装置時間使用を調べれば (ジョブ表示 (DSPJOB) コマンドを使って、活動実行属性を表示して)、そのジョブが無限に待機しているかどうかを判別することができます。

サーバー・システムが IBM i を実行しているときには、IBM i データベース機能を使って、リモート・ファイルにアクセスする場合、以下の制約があります。

- 論理ファイルまたは結合論理ファイルの基礎となる物理ファイルが、同じ IBM i オペレーティング・システム上に存在する必要があります。
- IBM i クライアント・システムにある論理ファイルは、リモート・ファイルのアクセス・パス (どのサーバー・システムのものであっても) を共用することはできません。
- QUERY 要求 (OPNQRYF) は、システム/38 からのグループ選択および結合処理を必要とするため、機能しません。

DDM の IBM i 以外のターゲットに関する制約事項と考慮事項:

サーバー・システムが IBM i またはシステム/38 を実行していない場合、サーバー・システムが IBM i を実行しているときに適用される制約事項に加えて、このトピックの制約事項も適用されることがあります。

適用されるかどうかは、ターゲット・システムがサポートする内容によって決まります。詳細については、該当するシステムの資料を参照してください。

- 通常、ソース・システムとサーバー・システムに共通するフィールド・データ・タイプのみを、HLL アプリケーションが処理することができます。浮動小数点データは、共通にはならないデータ・タイプの一例です。浮動小数点の入ったレコードの伝送はできますが、システムから別のシステムへ送られた浮動小数点データは、異なる表現になることがあります。

複数のシステム間で送信されるバック記号は、異なることがあります。たとえば、あるシステムでは C を使用し、別のシステムでは F を使用することがあります。

注: DDM ファイルを介して処理するレコード用のバイト・ストリングを、ユーザーの望みどおりに解釈するよう、アプリケーション・プログラムを作成することは可能です。ただしそのように作成するときには、データが必ず正しく取り扱われるようにする必要があります。

- レコード・ロック待ち時間のような、応答の前に遅延期間が必要な操作の場合はすべて、サーバー・システムで拒否されたり、待ち時間がゼロに変更されたりすることがあります。
- サーバー・システムで、ロック要求は、さらに制約を受けるロックに変更されることがあります。このような変更のため、同時にローカル IBM i オペレーティング・システムで実行されたかもしれない特定の操作が、実行されなくなることがあります。詳細については、188 ページの『オブジェクト割り振り (ALCOBJ) コマンド』を参照してください。
- IBM i パラメーターによっては、IBM i が実行されていないサーバー・システムでリモート・ファイルの処理時に使用すると、無視されたりエラーが発生するものがあります。たとえば、特定のファイル・コマンドにおける FRCRATIO パラメーターと FMTSLR パラメーターです。詳しくは、OVRDBF (データベース・ファイル一時変更) コマンドおよび DDM のコピー・コマンドを参照してください。
- メンバー名は、DDM 体系ではサポートされません。サーバー・システムが IBM i またはシステム/38 を実行していない場合、物理ファイル・メンバー消去 (CLRPFM) コマンドなどの MBR パラメーターを持つ CL コマンドで、そのパラメーターにファイル名と異なるメンバー名が指定されている場合は、そのコマンドを変更する必要があります。メンバー名が異なっていると、IBM i オペレーティング・システム上にないリモート・ファイルに対してこのコマンドを使用したときに、エラーが生じます。コマンドによっては、MBR (*FIRST) や MBR (*LAST) も有効です。ファイル・メンバーに関連した全 CL コマンドのリストと、IBM i を実行しないサーバー・システムにあるファイルにアクセスするには無効であるコマンドについては、『DDM でのメンバー関連コマンド』を参照してください。

注: システム/38 では、MBR(*LAST) はサポートされていません。

- CL コマンド上のパラメーターにソース・ファイル名が必要であったとしても、IBM i オペレーティング・システムにないターゲット・ファイルを参照する DDM ファイルの名前をソース・ファイル名として指定することはできません。それは、IBM i オペレーティング・システムは、IBM i にないターゲット・システム上のリモート・ファイルが実際にソース・ファイルかどうかを判別することはできないからです。(ソース・ファイルに関連した全 CL コマンドのリストは、『ソース・ファイル・コマンド』を参照してください。)
- IBM i コマンドの中には、IBM i またはシステム/38 サーバー・システムでは有効であっても、それ以外のターゲット・システムでは無効なものもあります。ターゲット・システムが IBM i またはシステム/38 でない場合にサポートされないコマンドのリストについては、『DDM 関連の CL コマンド・リスト』を参照してください。

初期セットアップ

IBM i オペレーティング・システムは、分散リレーショナル・データベースの実行時サポートを提供します。ただし、とりわけ拡張プログラム間通信 (APPC) 環境などにおいては、サーバーやクライアントで作業の送受信が行えるようにするために、いくらかのセットアップ作業が必要になる場合があります。

1 つ以上のサブシステムを使い、対話式ジョブ、バッチ・ジョブ、スプール・ジョブ、および通信ジョブを制御することができます。また、ネットワーク内のすべてのクライアントには、リレーショナル・データベース・ディレクトリーと、そこへの接続情報がセットアップされなければなりません。さらに、ネットワーク全体のサーバーのテーブルに、データを入れることも必要かもしれません。

リレーショナル・データベース・ディレクトリーには、データベース名および通信ネットワーク・パラメーターに変換される値が入ります。クライアントは、ローカル・データベースや、独立補助記憶域プール (独立 ASP または独立ディスク・プールともいう) 上に構成されるすべてのユーザー・データベースを含む、ネットワーク内の各データベースに、項目を持たなければなりません。これらのローカル項目は、システムに自動で追加させることもできますし、手動で追加することもできます。各ディレクトリー項目は、固有のリレーショナル・データベース名、および対応する通信パス情報で構成されています。アウトバウンド接続に

ついて、優先するパスワード・セキュリティーの情報を指定できます。ARD プログラムの提供するアクセスに対しては、リレーショナル・データベース・ディレクトリー項目に ARD プログラム名を追加する必要があります。

データベースにデータを入力する方法はいろいろあります。SQL アプリケーション・プログラム、その他の高水準言語アプリケーション・プログラム、または次の方法のいずれかを使用することができます。

- 対話式 SQL
- IBM i Query 管理機能
- データ・ファイル・ユーティリティー (DFU)
- ファイル・コピー (CPYF) コマンド

非類似システムの分散リレーショナル・データベース・ネットワークの接続とセットアップ情報は、『*Distributed Relational Database Cross-Platform Connectivity*』(SG24-4311) に説明されています。

関連概念:

独立補助記憶域プール

独立ディスク・プール

関連資料:

ファイル・コピー (CPYF) コマンド

IBM i 実行管理機能

IBM i オペレーティング・システム上の作業はすべて、実行管理機能を通して実行要求されます。システム上では、異なるタイプの作業を処理してシステム要件を充足するために、特別な操作環境を設計することができます。

ただし、オペレーティング・システムをインストールすると、そのオペレーティング・システムには、対話式処理、バッチ処理、通信、およびスプール処理をサポートする実行管理環境が含まれています。

システムでは、ユーザー・ジョブはすべてサブシステム記述で定義されたサブシステムと呼ばれる環境で動作します。システムはこの環境で処理やリソースを調整します。共通の特性を持つジョブが同一のサブシステム内に置かれている場合には、ユーザーは、それらのジョブのグループを他のジョブとは別に制御することができます。実行する作業をサポートし、所要のパフォーマンス特性を維持するために、必要に応じて、サブシステムの開始および終了を行うことができます。

システム上で実行されるジョブの基本タイプには、対話式、通信、バッチ、スプール、自動開始、および事前開始があります。

対話式ジョブは、ワークステーションにサインオンすると開始し、サインオフすると終了します。拡張プログラム間通信機能 (APPC) バッチ・ジョブは、別のシステムからプログラム開始要求によって開始されるジョブです。通信以外のバッチ・ジョブは、ジョブ待ち行列から開始されます。通信バッチ・ジョブを開始する時は、ジョブ待ち行列は使用されません。スプーリング機能は、入力と出力の両方で使用可能です。自動開始ジョブは、反復作業または 1 回限りの初期設定作業を行います。自動開始ジョブは、特定のサブシステムに対応しており、サブシステムが開始されると、それに対応する自動開始ジョブが開始されます。事前開始ジョブは、リモート・プログラムがプログラム開始要求を送る前に実行を開始するジョブです。

関連概念:

282 ページの『TCP/IP サーバーの管理』

DRDA および DDM TCP/IP サーバーでは、通常は、既存のシステム構成に変更を加える必要はありません。ただし、場合によっては、自分の要求によりよく合わせるため、問題を解決するため、システム・パフォーマンスを向上させるため、あるいはシステムでジョブを監視するために、システムによるサーバー・ジョブの管理方法を変更したいことがあります。

実行管理機能環境のセットアップ

システムをロードすると、**制御サブシステム**と呼ばれる 1 つのサブシステムが自動的に始動します。IBM では 2 つの制御サブシステム構成が用意されています。

第 1 の構成には、次のようなサブシステムがあります。

- QBASE は制御サブシステムで、対話式ジョブ、バッチ・ジョブ、および通信ジョブをサポートします。
- QSPL は、スプール読み取りプログラムおよびスプール書き出しプログラムの処理をサポートします。
- QSYSWRK は TCP/IP などの様々なシステム機能をサポートします。
- QUSRWRK は、ユーザーの作業のサブシステムです。これには、ユーザーの作業を代行処理するため、システムによって開始されるジョブも含まれます。

QBASE は、サーバーが開始されると、自動的に始動します。QBASE の中で自動的に開始されたジョブが QSPL を開始します。

提供されている 2 番目の制御サブシステム構成は、最初のものに比べて複雑です。この構成は、以下のサブシステムから成っています。

- QCTL は制御サブシステムで、コンソールで開始される対話式ジョブをサポートします。
- QINTER は他のワークステーションで開始された対話式ジョブをサポートします。
- QCMN は通信ジョブをサポートします。
- QBATCH はバッチ・ジョブをサポートします。
- QSPL は、スプール読み取りプログラムおよびスプール書き出しプログラムの処理をサポートします。
- QSYSWRK は TCP/IP などの様々なシステム機能をサポートします。
- QUSRWRK は、ユーザーの作業のサブシステムです。これには、ユーザーの作業を代行処理するため、システムによって開始されるジョブも含まれます。

構成を変更して QCTL 制御サブシステムを使用する場合には、このサブシステムはシステムの開始時に自動的に始動します。QCTL の中で自動的に開始されたジョブがその他のシステムを開始します。

システム値変更 (CHGSYSVAL) コマンドのシステム値 QCTLSBSD (制御サブシステム) を QCTL に変更し、再度システムを開始することによって、サブシステム構成を QBASE から QCTL に変更することができます。

サブシステム記述変更 (CHGSBSD) コマンドを使用することによって、IBM 提供のサブシステム記述またはユーザー作成のサブシステム記述を変更することができます。また、このコマンドを使用して、活動サブシステムの記憶域プール・サイズ、記憶域プール活動レベル、およびサブシステム記述の最大ジョブ数を変更することができます。

関連概念:



通信管理 (PDF)

作業の管理

関連資料:

サブシステム記述変更 (CHGSBSD) コマンド

システム値変更 (CHGSYSVAL) コマンド

APPC サブシステム

システム・ネットワーク体系 (SNA) ネットワークを使用する分散リレーショナル・データベースの場合、管理者が各システムで管理の計画を立てる必要のある主要な作業のタイプは、通信ジョブと対話式ジョブであるといえます。

ネットワーク内のシステムは、通信ジョブを開始して、クライアントからの要求を処理します。他のシステムに対するクライアントの通信要求は、通常、ローカル・システム上の対話式またはバッチ・ジョブから生じます。

分散リレーショナル・データベース・ネットワーク・システムに効率的な実行管理環境をセットアップするのなら、ネットワーク内の各サーバーやクライアントの特定の必要に合わせてシステム・リソースを割り振ることで、ネットワーク全体のパフォーマンスを高めることができます。

IBM i ライセンス・プログラムが最初にインストールされた時点では、QBASE がデフォルトの制御サブシステムです。制御サブシステムとして、QBASE は 2 つのサブシステム、QBASE と QSPL の間にシステム・リソースを割り振ります。対話式ジョブ、通信ジョブ、バッチ・ジョブなどが、QBASE サブシステム内でリソースを割り振ります。スプール・ジョブだけが別のサブシステム QSPL のもとで管理されます。つまり、通信ジョブを対話式ジョブと対比して処理する場合は、QCTL 制御サブシステムを使用する場合ほどには、システム・リソースの制御を行えないことを意味します。

QCTL サブシステム構成を使用すれば、システムが記憶域プールや他のシステム・リソースを割り振っている 4 つの追加サブシステムの制御を行うことができます。QCTL サブシステムの変更、または独自のサブシステムの作成を行えば、処理リソースの柔軟性および制御はさらに増します。

Spiffy 社の分散リレーショナル・データベース・ネットワークの中のシステムの一部でシステム要件が異なる場合、最高のネットワーク効率を得るためには、異なった実行管理環境が必要になることがあります。以下の説明では、分散リレーショナル・データベース管理担当者が実行管理サブシステムをどのように計画すれば、Spiffy 社の分散リレーショナル・データベース・ネットワークの中の各 IBM i 製品の要件に適合することができるかを示しています。

Spiffy 社のシステム編成では、販売店が小規模であれば、ユーザーがシステム上で処理する様々なジョブについて、QBASE レベルの制御で十分のことがあります。たとえば、小規模の販売店のリレーショナル・データベースに対する地域クライアントからの要求 (出荷に関する販売店在庫レベルの更新) は、通信ジョブとして処理されます。地区に現在在庫していない部品を要求するために、販売店ユーザーが地域サーバーに対して行う要求は、販売店システムで対話式ジョブとして処理されます。これらの活動は、販売店が小規模であり、処理する保守サービス受注件数および部品販売高が少ないので、両方とも相対的に小さいジョブです。QBASE サブシステム内のリソースの調整によって、この企業が対話式要件および通信要件で必要とする制御のレベルが提供されます。

これに対して、販売店の規模が大きくなると、ジョブのタイプが異なるのに対応して作業負荷が異なるので、QCTL サブシステムによって作業を管理することになると考えられます。

毎日の保守サービス受注件数が増える可能性があり、したがって、ローカル・リレーショナル・データベースに対する部品の照会、また販売店に在庫していない部品に関する地域センターのサーバーに対する照会が必要になります。このタイプの活動は、システム上で対話式ジョブを開始します。販売店では、企業人事記録の保管、営業および販売の計画および報告書作成など、分散リレーショナル・データベース関連ジョブではない多くの対話式ジョブも開始します。この販売店に対する地域センターからのパフォーマンス情報の

要求や在庫または作業計画の更新要求は、販売店が別の環境で管理したいと考える通信ジョブです。大規模販売店では、地域センターで在庫切れになっている部品に関する要求を、別の販売店から受信することもあります。

販売店が大規模な場合には、サブシステム管理を QINTER と QCMN とで別にした QCTL 構成によって、作業環境の管理に関する柔軟性と制御が増します。この例では、販売店システムでの対話式ジョブおよび通信ジョブには、他のタイプのジョブに比べて、多くのシステム・リソースを割り振ることができます。さらに、このシステムで、通信ジョブが対話式ジョブに比べて概して少ない場合には、QINTER と QCMN の両方のサブシステム記述を変更することによって、リソースを対話式ジョブに振り向けることができます。

Spiffy 社の地域センターの見地から実行管理環境を調整することも大切です。Spiffy ネットワークでは、地域センターは、定期部品出荷データによって販売店在庫表を更新したり、特定の修理ジョブに関する新規または更新保守サービス計画によって保守サービス計画表を更新する場合には、各販売店に対してクライアントになります。これらのジョブの中には、システム使用度が概して低い早朝または午後遅く、対話式ジョブとして実行する（地域システムで）か、または正規の営業時間後にバッチ・ジョブとして実行する（地域システムで）ことができるものがあります。管理担当者は、特定の処理時間およびリソース要件に合わせて、QINTER および QBATCH サブシステムを調整することができます。

販売店に在庫していない部品、特定の保守サービス・ジョブ（ステアリング・ラックの再組み立てなど）に関する保守サービス計画、または販売店リレーショナル・データベースに対する最後の更新以後の技術速報やリコール通知に関して、販売店が地域リレーショナル・データベースに照会する必要がある時には、地域センターは、各販売店に対してサーバーになることもあります。これらの通信ジョブは、すべて QCMN で管理されます。

しかし、KC000（カンザス・シティー）地域センター、およびその傘下にある販売店による分散リレーショナル・データベース・ネットワーク使用の特定の側面をさらに綿密に検討してみると、カンザス・シティーの分散リレーショナル・データベース管理担当者にとってもっと別の方法があることがわかります。

KC000 システムがサービスを提供している販売店には、毎日数百件の保守サービス受注を処理する大規模なものも数社あり、毎日の保守サービス受注処理件数が 20 件未満の小規模なものも少数あります。その他は中規模販売店で、それぞれが毎日 100 件前後の保守サービス受注を処理しています。分散リレーショナル・データベース管理担当者に提示される問題の 1 つは、他のシステムから KC000 システムに寄せられるすべての通信要求の公平な処理です。大規模販売店がその要求で QCMN リソースを支配する可能性があり、ネットワークの中の他のシステムにとっては、応答時間およびコストが不満足な結果になります。

分散リレーショナル・データベース管理担当者は、追加の通信サブシステムを作成することができるので、各クラスの販売店（小規模、中規模、または大規模）は、いずれもサーバーに対して支援を要求し、一般的によりよい応答を受け取ることができます。各サブシステム記述のサブシステム属性、事前開始ジョブ項目、通信実行処理項目、および経路指定項目を調整することによって、管理担当者は、あるサブシステム上で活動状態にできるジョブの数、およびそのサブシステムでジョブを処理する方法を制御します。

管理担当者は、経路指定項目を追加することができます。これにより、CMPVAL パラメーターに QCNTEDDM を指定し、また、ジョブの優先順位を制御するクラスを指定することによって、DRDA/DDM ジョブのクラス（したがって優先順位）を変更することができます。次は、その例です。

```
ADDRTGE SBSD(QCMN) SEQNBR(280) CLS(QINTER) CMPVAL('QCNTEDDM' 37)
```

また、管理担当者は、次の例のように、事前開始ジョブとして QCNTEDDM を指定することによって、DRDA/DDM ジョブのための事前開始ジョブを追加することもできます。

```
ADDPJE SBSD(QCMN) PGM(QCNTEDDM)
```

関連概念:



通信管理 (PDF)

作業の管理

TCP/IP サブシステム

デフォルトで、TCP/IP 接続に使用される TCP/IP サーバー事前開始ジョブは、QUSRWRK サブシステムで実行されます。

QUSRWRK は、ユーザーの作業のサブシステムです。これには、ユーザーの作業を代行処理するため、システムによって開始されるジョブも含まれます。QSYSWRK では、事前開始ジョブに作業をまわすリスナー・ジョブが実行されます。

独立補助記憶域プール上のユーザー・データベース

ユーザーは、システム上に独立した補助記憶域プール (独立 ASP) を構成することによって、別の IBM i リレーショナル・データベースを作ることができます。独立した補助記憶域プールの各グループが、リレーショナル・データベースを構成するのです。

このトピックでは、独立した補助記憶域プール・グループをユーザー・データベース といいます。これらのグループは、独立した補助記憶域プールのグループのディスク上に存在する、すべてのデータベース・オブジェクトを構成します。さらに、その独立した補助記憶域プールがオンにされた IBM i オペレーティング・システムのシステム・データベースにあるデータベース・オブジェクトも、論理的にはすべてユーザー・データベースに組み込まれます。ただし、コミットメント制御の視点から見た場合に、システム・データベースは扱われ方が異なります。

ユーザー・データベースの作成と使用には、先に述べたコミットメント制御の考慮事項に示される点以外にも、いくつかの規則が関係しています。たとえば、1 つの例として、現行スレッドのユーザー・データベース (補助記憶域プール (ASP) の 1 つのグループ) に設定されている、クライアントからデータベースへの接続には、拡張プログラム間通信機能 (APPC) で保護された分散作業単位 (DUW) 会話は使用できません。また、別の例では、ユーザー・データベースで作成する一切のスキーマに対して、そのユーザー・データベースや関連するシステム・データベースにすでに存在する名前を使用することはできません。このような制限の詳細については、SQL 解説書のトピックを参照してください。

- ユーザー・データベースには、組み込むことのできない特定の DRDA 関連オブジェクトというものがありません。DDM ユーザー出口プログラムは、システム・データベース内のライブラリーに常駐していなければなりません。すべてのアプリケーション・リクエスター・ドライバー・プログラムも、これと同様です。
- ユーザー・データベースをオンにする処理を行うと、リレーショナル・データベース (RDB) ディレクトリーがある期間使用できなくなるため、ディレクトリーを使おうとする DRDA のクライアントやサーバーによる試行が、遅延されたりタイムアウトになったりする可能性があります。それで、同時に複数のユーザー・データベースをオンにするなら、ディレクトリー操作をタイムアウトにするための障害は、かなり大きなものになります。ユーザー・データベースをはじめオンにするときには、システムによって、そのデータベースのディレクトリー項目の追加が試行されます。データベースをオンにする操作を並行して行うことにより、ディレクトリーが使用できなくなるなら、項目の追加は失敗し、項目を手動で追加しなければならなくなります。
- ユーザー・データベースを使用する際に考慮しなければならない別の点は、RDB ディレクトリー内の項目の構成に関するものです。ユーザー・データベースの命名に関する規則の 1 つには、ユーザー RDB 名を、ネットワーク属性で指定されているシステム名 (ネットワーク属性表示 (DSPNETA) コマンドで表示される) と一致させることはできない、というものがありません。

- RDB ディレクトリー内のローカル・ユーザー・データベース項目は、関連付けられているデータベースがはじめてオンにされるときに、自動的に追加されます。これらの項目の作成は、*IP プロトコル・タイプを使用し、LOOPBACK として指定されたリモート・ロケーションで行われます。LOOPBACK とは、データベースがディレクトリーと同じシステムにあることを示すものです。ですから、複数のシステム間で切り替えて使用することを意図したユーザー・データベースには、専用の IP アドレスを関連付けて構成することが、強く推奨されています。切り替え可能なデータベースに専用の IP アドレスがないと、切り替えを行う度に、そのデータベースを参照するすべてのシステムで、ディレクトリー項目を手動で更新しなければなりません。
- ユーザー・データベースがオンに変更されたら、システムは着信要求上の RDB 名を検証して、システム・データベースまたはユーザー・データベースと突き合わせなければなりません。無効な RDB 名に対してクライアントが稼働していた場合は、変更しなければなりません。

関連概念:

アプリケーション CRG テークオーバー IP アドレスの管理

トランザクションおよびコミットメント制御のトラブルシューティング

『リレーショナル・データベース・ディレクトリーの使用』

IBM i オペレーティング・システムは、リレーショナル・データベース・ディレクトリーを使用して、システム・アプリケーションでアクセスできるリレーショナル・データベース名を定義して、これらのリレーショナル・データベース名を対応するネットワーク・パラメーターに関連付けます。システムはこのディレクトリーを使用して、この接続でシステム・ネットワーク体系 (SNA) または IP が使用されるかどうかを指定することもできます。

関連資料:

ネットワーク属性表示 (DSPNETA) コマンド

SQL リファレンス

リレーショナル・データベース・ディレクトリーの使用

IBM i オペレーティング・システムは、リレーショナル・データベース・ディレクトリーを使用して、システム・アプリケーションでアクセスできるリレーショナル・データベース名を定義して、これらのリレーショナル・データベース名を対応するネットワーク・パラメーターに関連付けます。システムはこのディレクトリーを使用して、この接続でシステム・ネットワーク体系 (SNA) または IP が使用されるかどうかを指定することもできます。

リレーショナル・データベース・ディレクトリーを使用すると、クライアントはアプリケーションからのリレーショナル・データベース名を受け入れ、通信処理のために、この名前を該当するインターネット・プロトコル (IP) アドレスまたはホスト名とポート、あるいはシステム・ネットワーク体系ネットワーク識別コードおよび論理装置 (LU) 名の値に変換することができます。V5R2 と同じく、RDB ディレクトリーも、ユーザーの優先アウトバウンド接続セキュリティー・メカニズムを指定するために使用されます。リレーショナル・データベース・ディレクトリーを使用すると、また、ARD プログラムをリレーショナル・データベース名と関連付けることもできます。

分散リレーショナル・データベース・ネットワークの中の各 IBM i オペレーティング・システムごとに、リレーショナル・データベース・ディレクトリーを構成しなければなりません。システム上に存在するリレーショナル・データベース・ディレクトリーは 1 つだけです。分散リレーショナル・データベース・ネットワークの中の各クライアントは、それぞれのリレーショナル・データベース・ディレクトリーの中に、それ自体のローカル・リレーショナル・データベースの項目を 1 つと、そのクライアントがアクセスする各リモートおよびローカル・ユーザー・リレーショナル・データベースごとに項目を 1 つずつ持っていなければなりません。分散リレーショナル・データベース・ネットワークの中であって、サーバーとしてしか機能

することのないシステムの場合は、他のリモート・リレーショナル・データベースのリレーショナル・データベース名がそのディレクトリーに含まれている必要はありません。

ローカルのリレーショナル・データベースに割り当てられるリレーショナル・データベース名は、固有でなければなりません。つまり、ネットワークの中の他のリレーショナル・データベースとは異なっていなければなりません。ディレクトリーの中で他のリレーショナル・データベースに割り当てられる名前は、リモート・リレーショナル・データベースまたはローカル・ユーザー・データベースを識別しなければなりません。リモート RDB の名前は、サーバーがそのローカル・システム・データベースを識別するために使用する名前か、または構成済みである場合は、そのユーザー・データベースの 1 つを識別するために使用する名前と一致しなければなりません。サーバーのローカル・システム RDB 名前項目が必要なときに存在しないならば、ディレクトリーで自動的に作成されます。ディレクトリーで使用される名前は、ネットワーク属性の表示 (DSPNETA) コマンドによって表示される現行システム名です。

関連資料:

ネットワーク属性表示 (DSPNETA) コマンド

リレーショナル・データベース・ディレクトリーの処理

これらの説明に従って、リレーショナル・データベース・ディレクトリーを処理してください。

関連資料:

リレーショナル・データベース・ディレクトリー項目の追加 (ADDRDBDIRE) コマンド

リレーショナル・データベース・ディレクトリー項目の変更 (CHGRDBDIRE) コマンド

リレーショナル・データベース・ディレクトリー項目の表示 (DSPRDBDIRE) コマンド

リレーショナル・データベース・ディレクトリー項目の除去 (RMVRDBDIRE) コマンド

リレーショナル・データベース・ディレクトリー項目の処理 (WRKRDBDIRE) コマンド

SNA 使用の場合の項目の追加:

「RDB ディレクトリー項目の追加 (ADDRDBDIRE)」画面を以下に示します。リレーショナル・データベース・ディレクトリー項目の追加 (ADDRDBDIRE) コマンドを使用して、リレーショナル・データベース・ディレクトリーに項目を追加することができます。

RDB ディレクトリー項目の追加 (ADDRDBDIRE)

選択項目を入力して、実行キーを押してください。

項目:

リレーショナル・データベース	MP311	
リレーショナル・データベース別名	*NONE	
リモート・ロケーション:			
名前またはアドレス	MP311	
タイプ	*SNA	*SNA, *IP

この例では、MP311 というリモート・ロケーション名を持つシステムの MP311 という名前のリレーショナル・データベースを、ローカル・システム上のリレーショナル・データベース・ディレクトリーに追加するために、入力が行われます。SNA 接続の場合、リレーショナル・データベースの別名フィールドは、デフォルト値の *NONE のままにしておかなければなりません。リモート・ロケーション名は、それを使用するリレーショナル・データベース・ディレクトリー項目の作成前に、定義される必要はありません。ただし、リモート・ロケーション名は、リレーショナル・データベース・ディレクトリー項目がアプリケーションの中で使用される前に定義されなければなりません。リレーショナル・データベース名 (RDB) パラメーターおよびリモート・ロケーション名 (RMTLOCNAME) パラメーターは、RDB ディレクトリー項目の追加 (ADDRDBDIRE) コマンドでは必須です。デフォルトでは、RMTLOCNAME パラメーターの 2 番目の要素は *SNA です。テキスト (TEXT) パラメーターは任意指定です。この例に示してあるように、リレーシ

ヨナル・データベース名をシステム名、またはネットワーク構成でこのシステムに指定されているロケーション名と同じにするのはよい考えです。このようにすることにより、データベース名を識別するとともに、特にネットワークが複雑な場合に、そのデータベース名を、分散リレーショナル・データベース・ネットワークの中の特定のシステムに結び付けるのに役立ちます。

上記以外のこのコマンドの任意指定パラメーターを表示するには、「**RDB ディレクトリー項目の追加 (ADDRDBDIRE)**」画面で F9 を押してください。これらの任意指定パラメーターを下に示します。

```

RDB ディレクトリー項目の追加 (ADDRDBDIRE)

選択項目を入力して、実行キーを押してください。

項目:
リレーショナル・データベース . . . . . MP311
リレーショナル・データベース別名. . . . . *NONE
リモート・ロケーション:
名前またはアドレス . . . . . MP311
タイプ . . . . . *SNA          *SNA, *IP
...

装置:
APPC 装置記述 . . . . . *LOC          名前, *LOC
ローカル・ロケーション . . . . . *LOC          名前, *LOC, *NETATR
リモート・ネットワーク識別コード . . . . . *LOC          名前, *LOC, *NETATR, *NONE
モード . . . . . *NETATR        名前, *NETATR
トランザクション・プログラム. . . . . *DRDA         文字値, *DRDA
テキスト. . . . . 'Oak Street Dealership'

```

システムは、デフォルト値として *SNA を、以下の追加の **RDB ディレクトリー項目の追加 (ADDRDBDIRE)** コマンド・パラメーターに提供します。

- 装置 (DEV)
- ローカル・ロケーション (LCLLOCNAME)
- リモート・ネットワーク ID (RMTNETID)
- モード (MODE)
- トランザクション・プログラム (TNSPGM)

注:

1. SNA 接続の場合、リレーショナル・データベースの別名フィールドは、デフォルト値の *NONE のままにしておきます。
2. IBM i オペレーティング・システムのトランザクション・プログラム名パラメーターは TNSPGM です。SNA ではこれは TPN です。
3. デフォルトの値を拡張プログラム間通信 (APPC) で使用する場合には、システムは、使用される装置、ローカル・ロケーション、およびリモート・ネットワーク ID を決めます。ネットワーク属性に定義されているモード名が使用され、Distributed Relational Database Architecture (DRDA) サポートのトランザクション・プログラム名が使用されます。デフォルトの値を Advanced Peer-to-Peer Networking (APPN) で使用する場合には、システムは、装置 (DEV) パラメーターを無視し、ネットワーク属性に定義されているローカル・ロケーション名、リモート・ネットワーク ID、およびモード名を使用します。

ADDRDBDIRE コマンドの省略時の値は、いずれも変更することができます。たとえば、DB2 for VM サーバーと通信するためには、TNSPGM パラメーターを変更しなければならない場合があります。DB2 for VM サポートのデフォルト解釈では、トランザクション・プログラム名は接続する DB2 for VM データベ

ースの名前になります。DRDA (*DRDA) のデフォルトの TNSPGM パラメーターの値は、X'07F6C4C2' です。QCNTEDDM および DB2DRDA も X'07F6C4C2' にマップします。

関連タスク:

423 ページの『DB2 for VM クライアント上でトランザクション・プログラム名として QCNTSRVC を設定する』

UCOMDIR NAMES ファイルを変更して、TPN タグに QCNTSRVC を指定します。

関連資料:

リレーショナル・データベース・ディレクトリー項目の追加 (ADDRDBDIRE) コマンド

422 ページの『DB2 for i アプリケーション・リクエスター上でトランザクション・プログラム名として QCNTSRVC を設定する』

リレーショナル・データベース・ディレクトリー項目の追加 (ADDRDBDIRE) コマンドまたは リレーショナル・データベース・ディレクトリー項目の変更 (CHGRDBDIRE) コマンドの TNSPGM パラメーターに QCNTSRVC を指定します。

423 ページの『DB2 for z/OS クライアント上でトランザクション・プログラム名として QCNTSRVC を設定する』

SYSIBM.LOCATIONS テーブルを更新して、DB2 for i サーバーの RDB-NAME の入った行について、TPN 列に QCNTSRVC を指定します。

423 ページの『DB2 for Linux, UNIX, and Windows クライアント上でトランザクション・プログラム名として QCNTSRVC を設定する』

DB2 for Linux、UNIX、および Windows を操作し、このプロダクトのファミリーで TPN 設定の説明が必要な場合、Web ページがお役に立ちます。

TCP/IP 使用の場合の項目の追加:

「RDB ディレクトリー項目の追加 (ADDRDBDIRE)」画面では、RMTLOCNAME パラメーターの 2 番目の項目に *IP を入力した際のパネルの変更の様子と、TCP/IP を使用する RDB でどんな一般的な項目が表示されるかが示されています。

TCP/IP を使用する接続で「リレーショナル・データベース別名」フィールドの使用が可能になっていますが、この最初の TCP/IP の例では、別名を指定しません。

RDB ディレクトリー項目の追加 (ADDRDBDIRE)

選択項目を入力して、実行キーを押してください。

項目:

リレーショナル・データベース	> MP311	
リレーショナル・データベース別名 . . .	*NONE	

リモート・ロケーション:

名前またはアドレス	MP311.spiffy.com	
---------------------	------------------	--

タイプ	> *IP	*SNA, *IP
ポート番号またはサービス・プログラム	*DRDA	

リモート認証方式:

所望の方式	*USRENCPWD	*USRENCPWD, *USRID...
より低い認証の許可	*ALWLOWER	*ALWLOWER, *NOALWLOWER
暗号化アルゴリズム	*DES	*DES, *AES
保護接続	*NONE	*NONE, *SSL

続く...

F3= 終了 F4=プロンプト F5= 最新表示 F12= 取り消し
F13= この画面の使用法 F24= キーの続き

リレーショナル・データベースの別名の指定:

この例は、RDB 別名を指定するディレクトリー項目の追加を示しています。これにより、それぞれのネットワークに同じ名前のリレーショナル・データベースがあっても、分散リレーショナル・データベース・アーキテクチャー (DRDA) 環境でそれぞれを一意的に識別できるようになります。

また別名を使用する項目が RDB ディレクトリーに追加されている場合は、その別名で項目が識別されます。項目を表示または削除するときは、別名を指定する必要があります。

下の画面では、リレーショナル・データベースの別名として RDBALS が指定されています。

RDB ディレクトリー項目の追加 (ADDRDBDIRE)

選択項目を入力して、実行キーを押してください。

項目:
リレーショナル・データベース > TEST
リレーショナル・データベース別名 . . . RDBALS
リモート・ロケーション:
名前またはアドレス MP311.spiffy.com

タイプ	> *IP	*SNA, *IP
ポート番号またはサービス・プログラム	*DRDA	
リモート認証方式:		
所望の方式	*USRENCPWD	*USRENCPWD, *USRID...
より低い認証の許可	*ALLOWER	*ALLOWER, *NOALLOWER
暗号化アルゴリズム	*DES	*DES, *AES
保護接続	*NONE	*NONE, *SSL

続く...

F3= 終了 F4=プロンプト F5= 最新表示 F12= 取り消し
F13= この画面の使用法 F24= キーの続き

リレーショナル・データベース・ディレクトリー項目の処理 (WRKRDBDIRE) コマンドおよびオプション 1 を使用して別名の項目を追加する場合は、最初に「項目名 (Entry)」フィールドに実際の RDB 名を入力し、Enter を押します。次いで、「リレーショナル・データベース別名」フィールドに別名を入力し、その他のフィールドにも必要な値を入力すると、RDB 項目のリストの「項目名 (Entry)」フィールドで、実際の RDB 名が別名に置き換えられます。リモート・ロケーション名の「タイプ」は *SNA から *IP に変更する必要があります。リモート・ロケーション名として *LOCAL を指定しているリレーショナル・データベース・ディレクトリー項目に別名を追加することはできません。

リレーショナル・データベース項目を **リレーショナル・データベース・ディレクトリー項目の除去 (RMVRDBDIRE)** コマンドを使用して除去する場合、実際のリレーショナル・データベース名でなく別名を使用して、除去する項目を指定します。

別名によってリモート・データベースを識別する場合は、同一ディレクトリーの中で、実際の名前も使用してデータベースを参照することはできません。

RMTLOCNAME パラメーターで MP311.spiffy.com を指定しなくても、IP アドレス (例えば、9.5.25.176) を指定できます。別の IBM i 環境への IP 接続では、ポート 447 を使用する必要がなければ、PORT パラメーター値をデフォルトの値である *DRDA のままにしてください。たとえば、IP セキュリティー・アーキテクチャー (IPSec) を使用した伝送用に構成されたポート 447 があるとします。他のプラットフォーム上の IBM DB2 サーバーへの接続では、ポートを 50000 などの数字に設定する必要があるでしょう。詳しいことは、ご使用のシステムのプロダクト資料を参照してください。特定の場所の DRDA ポートで有効なサービス名を定義した場合には、数字の代わりにその名前を使用することもできます。ただし、IBM i オペレーティング・システム上では、*DRDA が DRDA サービス名の使用より優先されます。

アプリケーション・リクエスター・ドライバーの項目の追加:

リレーショナル・データベース・ディレクトリー項目の追加 (ADDRDBDIRE) コマンド・プロンプトで通信情報とアプリケーション・リクエスター・ドライバー (ARD) プログラムを指定するには、F9 (すべてのパラメーター) と次ページ・ボタンを押してください。

ARD プログラムで ADDRDBDIRE コマンドに指定された通信情報を使用しない場合は (通常は使用しない)、RMTLOCNAME パラメーターに特別な値 *ARDPGM を使用します。ARD プログラムは、システム・データベース内のライブラリー (ASP 番号 1-32) に置く必要があります。

関連資料:

リレーショナル・データベース・ディレクトリー項目の追加 (ADDRDBDIRE) コマンド

WRKRDBDIRE コマンドの使用:

次に示す RDB ディレクトリー項目の処理画面には、リレーショナル・データベース・ディレクトリー項目の追加、変更、表示、または除去を行うオプションが用意されています。

リレーショナル・データベース・ディレクトリー項目の処理 (WRKRDBDIRE)

位置指定

オプションを入力して、実行キーを押してください。
1= 追加 2= 変更 4= 除去 5= 明細の表示 6= 明細の印刷

OPT	項目	リモート ロケーション	テキスト
-	KC000	KC000	カンザスシティ地域のデータ・ベース
-	MP000	*LOCAL	ミネアポリス地域のデータ・ベース
-	MP101	MP101	販売店データ・ベース MP101
-	MP102	MP102	販売店データ・ベース MP102
-	MP211	MP211	販売店データ・ベース MP211
-	MP215	MP215	販売店データ・ベース MP215
4	MP311	MP311	販売店データ・ベース MP311
-	MP415	MP415	販売店データ・ベース MP415
-	MP515	MP515	販売店データ・ベース MP515
-	MP615	MP615	販売店データ・ベース MP615
-	MP715	MP715	販売店データ・ベース MP715

続く...

F3= 終了 F5= 最新表示 F6= リストの印刷 F12= 取り消し
F22=フィールド全体の表示

画面に示されているように、オプション 4 を使用すれば、ローカル・システムのリレーショナル・データベース・ディレクトリーから項目を除去することができます。項目を除去する場合には、別の画面が表示されるので、その画面で指定した項目の除去要求を確認するか、または別のリレーショナル・データベース・ディレクトリー項目を選択することができます。リレーショナル・データベース・ディレクトリー項目の除去 (RMVRDBDIRE) コマンドを使用する場合には、特定のリレーショナル・データベース名、総称名、すべてのディレクトリー項目、またはリモート項目だけのいずれかを指定することもできます。

「リレーショナル・データベース・ディレクトリー項目の処理」画面では、1 つの項目の明細を表示するオプションを選択できます。「リレーショナル・データベース・ディレクトリー項目の処理」画面の出力は画面表示に送信されます。ただし、リレーショナル・データベース・ディレクトリー項目の表示 (DSPRDBDIRE) コマンドを使用すれば、その出力をプリンターまたは出力ファイルに送ることができます。リレーショナル・データベース・ディレクトリーは IBM i オブジェクトではありません。したがって、出力ファイルを使用することによって、リレーショナル・データベース・ディレクトリーのバックアップ手段が得られます。リレーショナル・データベース・ディレクトリーのバックアップのために出力ファイルを用

いるリレーショナル・データベース・ディレクトリー項目表示 (DSPRDBDIRE) コマンドの使用に関する詳細については、「リレーショナル・データベース・ディレクトリーの保管とリストア」を参照してください。

「リレーショナル・データベース・ディレクトリー項目の処理」画面では、リレーショナル・データベース・ディレクトリーの中の 1 つの項目を変更するオプションを選択することができます。また、リレーショナル・データベース・ディレクトリー項目の変更 (CHGRDBDIRE) コマンドを使用しても、ディレクトリーの中の項目に変更を加えることができます。任意指定のコマンド・パラメーターおよびシステムのリモート・ロケーション名は、変更することができます。ディレクトリー項目のリレーショナル・データベース名を変更することはできません。ディレクトリーの中のリレーショナル・データベースの名前を変更するには、そのリレーショナル・データベースの項目を除去し、新しいリレーショナル・データベース名の項目を追加します。

注: リモート・ロケーションがリレーショナル・データベース・ディレクトリー項目で変更された場合、遠隔ジャーナルを、遠隔ジャーナルの除去 (RMVMTJRN) コマンドまたは QjoRemoveRemoteJournal API を使用して除去し、遠隔ジャーナルの追加 (ADDRMTJRN) コマンドまたは QjoAddRemoteJournal API を使用して再び追加する必要があります。リモート・ロケーションのタイプ、認証、または他の何かが変更された場合、遠隔ジャーナルは、遠隔ジャーナルの変更 (CHGRMTJRN) コマンドまたは QjoChangeJournalState API を使用して終了し、再び遠隔ジャーナルの変更 (CHGRMTJRN) コマンドまたは QjoChangeJournalState API を使用して再始動する必要があります。分散ファイル用に変更を使用するには、ノード・グループを削除して再作成し、それからファイルを再作成する必要があります。

関連タスク:

328 ページの『リレーショナル・データベースのディレクトリーの保管と復元』
リレーショナル・データベース・ディレクトリーは IBM i オブジェクトではありません。その代わりに、初期プログラム・ロード (IPL) 時にシステムによって開かれたファイルから構成されます。

関連資料:

遠隔ジャーナル追加 (ADDRMTJRN) コマンド

リレーショナル・データベース・ディレクトリー項目の変更 (CHGRDBDIRE) コマンド

遠隔ジャーナル変更 (CHGRMTJRN) コマンド

リレーショナル・データベース・ディレクトリー項目の表示 (DSPRDBDIRE) コマンド

遠隔ジャーナル除去 (RMVMTJRN) コマンド

リレーショナル・データベース・ディレクトリー項目の除去 (RMVRDBDIRE) コマンド

*LOCAL ディレクトリー項目:

*LOCAL を含むディレクトリー項目は、ディレクトリー内にそのような項目が 1 つしかないという点、また、それがローカル・システム・データベースの名前を指定しているという点で固有です。

関連した RDB 名は、SQL ステートメントの CONNECT TO xxx (xxx はローカル・システムの名前) で使用して、ローカル・データベースに接続できます。CONNECT TO xxx による結果は、SQL ステートメントの CONNECT RESET を使用した場合と同等です。

プログラム・テストなどのために、ローカル・システム・データベースへの DRDA 接続を作成したい場合は、この目的で使用できる ME および MYSELF という、2 つの特殊 RDB 名があります。例えば、ME という RDB 名、タイプ *IP、およびリモート・ロケーション名 LOOPBACK のディレクトリー項目をプログラマーが追加する場合があります。それからプログラマーは、プログラム内で SQL CONNECT TO ME ステートメントを実行して、ローカル・システムへのソケット DRDA 接続を確立できます。ただし、これらの RDB 名の一般的な使用は勧められておらず、ある状況ではこの使用の結果として予期しない動作が生じることを警告するためだけに記述しています。

ただし、ローカル RDB 項目の名前を変更する必要がある場合の手順には、除去と追加操作を行う必要があることが含まれています。ただし、その項目にはシステム全体に適用される DRDA 属性情報が入っているので、ローカル項目を除去する際には特別な注意が必要です。項目を除去しようとすると、CPA3E01 (*LOCAL ディレクトリー項目を除去すると、構成データ (C G) が失われることがあります。) メッセージが出され、その操作を取り消す (C) かまたは継続する (G) かを選択する機会が与えられます。そして、メッセージ・テキストは、その項目が **DDM TCP/IP 属性の変更 (CHGDDMTCPA)** コマンドで入力された構成データを保管するために使用されていることを知らせます。*LOCAL 項目が除去されると、構成データが破棄される可能性があり、デフォルトの構成値が使用されるようになります。このデフォルトの値が満足のいくものではない場合には、CHGDDMTCPA コマンドを使用して構成データを再入力する必要があります。項目を除去する前には CHGDDMTCPA コマンドで指定されている値を記録しておく、*LOCAL 項目が削除されて正確なローカル RDB 名で追加された後に復元することができます。

リモート・ロケーション名として *LOCAL を指定しているリレーショナル・データベース・ディレクトリー項目に別名を追加することはできません。追加しようとすると、メッセージ CPD3EC8 が表示されます。

サーバー・サイド RDB 別名は TCP/IP を介してサポートされます。サーバー・サイド RDB 別名を作成するには、*LOCAL 項目からの RDB 名、ご希望の別名、リモート・ロケーション LOOPBACK、およびタイプ *IP を使用します。その後 DRDA サーバー・ジョブはその別名に対する着信要求を受け入れます。

RDB ディレクトリー項目の追加 (ADDRDBDIRE)

選択項目を入力して、実行キーを押してください。

項目:

リレーショナル・データベース > LCLRDB

リレーショナル・データベース別名 . . . LCLALS

リモート・ロケーション:

名前またはアドレス LOOPBACK

タイプ > *IP *SNA, *IP

ポート番号またはサービス・プログラム *DRDA

リモート認証方式:

所望の方式 *USRENCPWD *USRENCPWD, *USRID...

より低い認証の許可 *ALWLOWER *ALWLOWER, *NOALWLOWER

暗号化アルゴリズム *DES *DES, *AES

保護接続 *NONE *NONE, *SSL

続く...

F3= 終了 F4=プロンプト F5= 最新表示 F12= 取り消し
F13= この画面の使用法 F24= キーの続き

関連資料:

DDM TCP/IP 属性変更 (CHGDDMTCPA) コマンド

独立補助記憶域プール上のユーザー・データベース用のディレクトリー項目:

データベースが 1 つだけのシステムの場合 (つまり、構成済みの独立補助記憶域プール (独立 ASP) が無い)、*LOCAL 項目とは、単一のローカル・データベースを指します。複数のデータベースがあるシステムの場合 (1 つのシステム・データベースと、1 つ以上のユーザー・データベース)、*LOCAL 項目とは、システム・データベースを指します。

ローカル・ユーザー・データベースは、リモート *IP 項目に似た項目によって表されます。主な相違点は、「リモート・ロケーション」フィールドです。データベースが別のシステムに切り替えられない場合に、このフィールドには通常は語 LOOPBACK が入ります。LOOPBACK は、ホスト・システムの IP ア

ドレスを表します。データベースを切り替えられる場合、接続先のシステムに関係なく、特定の IP アドレスがデータベースに関連付けられるようにシステムを構成することが勧められています。専用 IP アドレスを構成する方法についての説明は、『アプリケーション CRG テークオーバー IP アドレスの管理』トピックを参照してください。この場合、IP アドレスは「リモート・ロケーション」フィールドで使用されます。

LOOPBACK を交換可能データベースに使用すると、それがローカル・システムから切り替えられるときはいつも、ユーザーは手動でディレクトリー項目を変更して、LOOPBACK を接続先の新規システムの IP アドレスで置き換え、その後データベースが切り替えられて元に戻される場合には LOOPBACK に戻す必要があります。

関連資料:

アプリケーション CRG テークオーバー IP アドレスの管理

例: リレーショナル・データベースのディレクトリーのセットアップ

Spiffy 社のネットワークが示す例を見れば、分散リレーショナル・データベース・ネットワークの中のシステムでリレーショナル・データベース・ディレクトリーをセットアップして使用方法が分かります。

この例では、通信に TCP/IP を使用するのではなく、拡張プログラム間通信 (APPC) を使用することを想定しています。この方が、セットアップがより容易と考えられます。ただし、この例の中のある要素は、プロトコルに依存していません。APPC で使用される必要がある RDB ディレクトリー項目は、TCP/IP ネットワークでも必要ですが、パラメーターが異なります。この場合、論理単位 (LU) 名、装置記述、モード、TPN などが、ホスト名または IP アドレス、およびポート識別によって置き換えられます。

単純な関係を考察するために、次の図に示すような 2 つの地域支社間の関係を取り上げてみます。



図 20. システムが 2 つの場合のリレーショナル・データベース・ディレクトリーのセットアップ

各地域支社のリレーショナル・データベース・ディレクトリーには、ローカル・リレーショナル・データベースの項目とリモート・リレーショナル・データベースの項目が入っていなければなりません。その理由は、各システムがそれぞれクライアントとサーバーの両方になるからです。MP000 システムのリレーショナル・データベース・ディレクトリーを作成するためのコマンドは、次のようになります。

```
ADDRDBDIRE    RDB(MP000) RMTLOCNAME(*LOCAL) TEXT('Minneapolis region database')
```

```
ADDRDBDIRE    RDB(KC000) RMTLOCNAME(KC000) TEXT('Kansas City region database')
```

前述の例では、MP000 システムは、RMTLOCNAME パラメーターに *LOCAL を指定することによって、それ自体をローカル・リレーショナル・データベースとして識別します。リレーショナル・データベース

は、1 つの IBM i プラットフォームに 1 つだけあります。RDB ディレクトリーの中のリレーショナル・データベース名を、システム名と同じにすれば、ネットワーク・リレーショナル・データベースの識別を簡単にできます。ローカル・ロケーションの項目はローカル・システム名と同じ名前にすることができ、リモート・ロケーション名の項目は、リモート・システム名と同じ名前にできます。

注: システム名は、ネットワーク属性変更 (CHGNETA) コマンドの SYSNAME パラメーターで指定します。ローカル・システムは、通信構成時に、CHGNETA コマンドの LCLLOCNAME パラメーターで識別します。SNA (APPC) を使用するリモート・ロケーションは、通信構成時に、制御装置記述の作成 (APPC) (CRTCTLAPPC) コマンドの RMTCPNAME パラメーターで識別します。システム名、ネットワーク・ロケーション、およびデータベース名に同じ名前を使用すれば、特に複雑なネットワークの場合に、混乱を避ける上で役立ちます。

KC000 システムのリレーショナル・データベース・ディレクトリーの場合の対応する項目は、次の通りになります。

```
ADDRDBDIRE    RDB(KC000) RMTLOCNAME(*LOCAL) TEXT('Kansas City region database')
```

```
ADDRDBDIRE    RDB(MP000) RMTLOCNAME(MP000) TEXT('Minneapolis region database')
```

複雑な例を考察するために地域支社とその傘下の販売店の関係を取り上げます。たとえば、次の図に示すネットワーク内のリレーショナル・データベースをアクセスするには、MP000 システムのリレーショナル・データベース・ディレクトリーを拡張して、その傘下の各販売店の項目を含める必要があります。

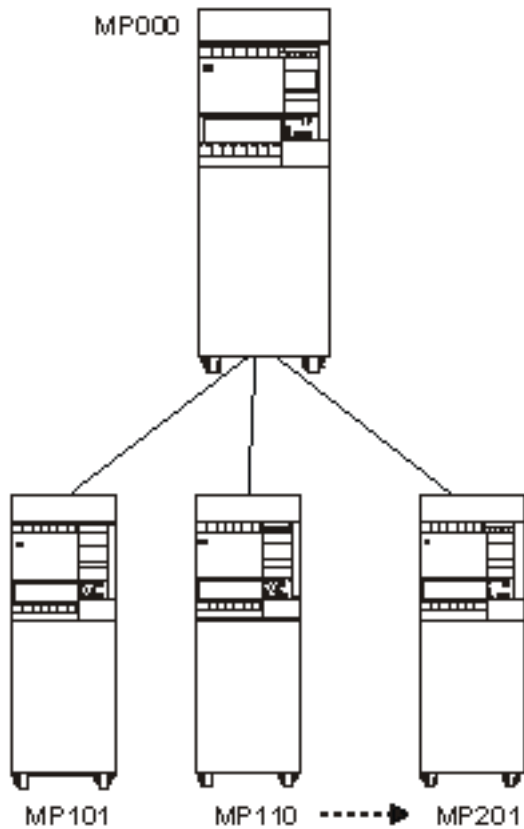


図 21. システムが多数の場合のリレーショナル・データベース・ディレクトリーのセットアップ

MP000 リレーショナル・データベース・ディレクトリーに、傘下の販売店データベースをすべて含めるために使用するコマンドの例を以下に示します。

```
PGM
ADDRDBDIRE   RDB(MP000) RMTLOCNAME(*LOCAL) +
TEXT('Minneapolis region database')
ADDRDBDIRE   RDB(KC000) RMTLOCNAME(KC000)
TEXT('Kansas City region database')
ADDRDBDIRE   RDB(MP101) RMTLOCNAME(MP101)
TEXT('Dealer database MP101')
ADDRDBDIRE   RDB(MP002) RMTLOCNAME(MP110)
TEXT('Dealer database MP110')
.
.
.
ADDRDBDIRE   RDB(MP215) RMTLOCNAME(MP201)
TEXT('Dealer database MP201')
ENDPGM
```

前述の例では、地域の各販売店は、リモート・リレーショナル・データベースとしてミネアポリスのリレーショナル・データベース・ディレクトリーに含められます。

各販売店は、MP000 および他の販売店サーバーに対してクライアントとして機能するので、各販売店は、それぞれ自体をローカル・リレーショナル・データベースとし、地域支社および他のすべての販売店をリモート・リレーショナル・データベースとする項目を持つリレーショナル・データベース・ディレクトリ

ーを持たなければなりません。データベース管理担当者は、各販売店のシステムのリレーショナル・データベース・ディレクトリーを作成するにあたって、複数のオプションから選択することができます。

最も時間を要し、しかもエラーを生じる可能性が最も高い方法は、MP000 分散リレーショナル・データベース・ネットワークの一部であるすべてのシステム上で各ディレクトリー項目を作成するためのリレーショナル・データベース・ディレクトリー項目の追加 (ADDRDBDIRE) コマンドを使用することによって、各システムにリレーショナル・データベース・ディレクトリーを作成する方法です。

それに代わる方法として優れているのは、前述の例で MP000 システムの場合に示してあるような制御言語 (CL) プログラムを作成する方法です。分散リレーショナル・データベース管理担当者は、各販売店のシステムごとにこの CL プログラムをコピーすることができます。このプログラムを各販売店に応じてカスタマイズするには、データベース管理担当者は、MP000 システムのリモート・ロケーション名を MP000 に変更し、地区販売店のリモート・ロケーション名を *LOCAL に変更します。分散リレーショナル・データベース管理担当者は、カスタマイズした CL プログラムを各販売店に配布し、そのシステムで実行させて、それぞれ固有のリレーショナル・データベース・ディレクトリーを作成させることができます。

3 番目の方法としては、リレーショナル・データベース・ディレクトリー項目の表示 (DSPRDBDIRE) コマンドを使用した結果として出力ファイルに送られるリレーショナル・データベース・ディレクトリー情報を読み取るプログラムを作成する方法があります。このプログラムは、MP000 システムのリレーショナル・データベース・ディレクトリー項目が入っている出力ファイルとともに、販売店に配布することができます。各システムでは、MP000 出力ファイルを読み取って、ローカル・リレーショナル・データベース・ディレクトリーを作成することができます。次にリレーショナル・データベース・ディレクトリー項目の変更 (CHGRDBDIRE) コマンドを使用して、MP000 システムのディレクトリーをローカル・システムに応じてカスタマイズできます。

関連タスク:

328 ページの『リレーショナル・データベースのディレクトリーの保管と復元』

リレーショナル・データベース・ディレクトリーは IBM i オブジェクトではありません。その代わりに、初期プログラム・ロード (IPL) 時にシステムによって開かれたファイルから構成されます。

関連資料:

リレーショナル・データベース・ディレクトリー項目の追加 (ADDRDBDIRE) コマンド

リレーショナル・データベース・ディレクトリー項目の変更 (CHGRDBDIRE) コマンド

制御装置記述の作成 (APPC) (CRTCTLAPPC) コマンド

ネットワーク属性表示 (DSPNETA) コマンド

リレーショナル・データベース・ディレクトリー項目の表示 (DSPRDBDIRE) コマンド

セキュリティのセットアップ

Distributed Relational Database Architecture (DRDA) セキュリティーについては、セキュリティのトピックで扱われています。しかし、その記述をより充実したものにするため、DRDA を使用する前、またはネットワークで拡張プログラム間通信 (APPC) の使用から TCP/IP の使用に変更する前の考慮事項としてここに記載します。

TCP/IP のセキュリティ・セットアップは、APPC で必要とされるものとはかなり異なっています。念頭に置くべき 1 つの点は、APPC の保護ロケーションという概念が存在しないということです。TCP/IP サーバーでは、クライアント・システムが主張しているとおりのものであることを信用することはできないので、接続要求でのパスワードの使用はより重要になります。接続要求でのパスワードの送信をより容易なものとするため、TCP/IP サポートでは特定のユーザー・プロファイルに関連したサーバー認証リストの使用が導入されました。サーバー認証リストの項目は、xxxSVRAUTE コマンド (xxx は ADD、CHG、および

RMV を表す) を使用して保守できます。このことは、86 ページの『セキュリティ』で説明されています。サーバー認証項目の使用に代わる別の方法は、SQL CONNECT ステートメントの USER/USING 形式を使用して、接続要求でパスワードを送信することです。

Kerberos のサポートによって、TCP/IP を使用していれば別のセキュリティ・オプションを使用できます。ネットワーク認証サービスは、Kerberos プロトコルをサポートします。またこのサービスを使用して、Kerberos を構成できます。

サーバー側のセットアップでは、着信する接続要求でどのレベルのセキュリティを必要とするかを決定して指定することも含まれています。たとえば、暗号化されていないパスワードを使用するかどうかなどです。デフォルトの設定では、パスワードは必要です。DDM TCP/IP 属性の変更 (CHGDDMTCPA) コマンドを使用すると、デフォルトの設定を変更することができます。

関連概念:

制御言語

関連タスク:

ネットワーク認証サービスの構成

関連資料:

DDM TCP/IP 属性変更 (CHGDDMTCPA) コマンド

TCP/IP サーバーのセットアップ

TCP/IP プロトコルを使用する Distributed Relational Database Architecture (DRDA) サーバーを所有する場合、DDM TCP/IP サーバーをセットアップする必要があります。

TCP/IP サーバーのセットアップは、必要な時にサーバーが開始されるようにすることによって簡単に行うことができます。そして、次のコマンドを実行して、サーバーがいつも活動状態であるようにできます。

```
CHGDDMTCPA AUTOSTART(*YES)
```

サーバーをご使用の環境に調整するために、他のパラメーターも変更することを望まれるかもしれません。これらには、開始される事前開始ジョブの最初の数、ジョブの最大数、ジョブをさらに開始する際の限界値などがあります。

すべてのクライアントが接続の際に使用する共通ユーザー・プロファイルを設定することもでき、リモート・ユーザーの様々なクラスで異なったセキュリティ・レベルを持つ様々なユーザー・プロファイルのセットを設定することもできます。それから、クライアントでサーバー認証項目の追加 (ADDSVRAUTE) コマンドを使用して、クライアントにある各ユーザーのプロファイル名を、サーバーでそれらのユーザーが実行されるユーザー・プロファイルに対応付けます。

関連概念:

94 ページの『TCP/IP ネットワークでのクライアントのセキュリティ』

接続性のシナリオが異なれば、異なるレベルの認証を使用することが求められます。したがって、管理者は、サーバーへ接続するときに、より望ましい認証方式フィールドを各 RDB ディレクトリー項目に設定することにより、クライアントで必要な最低のセキュリティ認証方式を設定できます。

282 ページの『TCP/IP サーバーの管理』

DRDA および DDM TCP/IP サーバーでは、通常は、既存のシステム構成に変更を加える必要はありません。ただし、場合によっては、自分の要求によりよく合わせるため、問題を解決するため、システム・パフォーマンスを向上させるため、あるいはシステムでジョブを監視するために、システムによるサーバー・ジョブの管理方法を変更したいことがあります。

関連資料:

サーバー認証項目追加 (ADDSVRAUTE) コマンド

対話式 SQL 用 SQL パッケージのセットアップ

このトピックは、IBM i を実行していないサーバー・システムのみにも適用されます。

以下の項目に該当する場合、SQL パッケージがシステムで作成されていることを確認する必要があります。

- IBM DB2 Query Manager and SQL Development Kit for i ライセンス・プログラムがあり、その製品の対話式 SQL 機能を使用する計画をしている場合。
- V5R1 より前の OS/400® クライアントの TCP/IP を使用するか、2 フェーズ・コミット機能がない、IBM i 以外の DRDA サーバーの接続を計画している場合

対話式 SQL は、IBM i 用の SQL パッケージを必要としません。通常の場合、IBM i を実行しないサーバー・システムでは、対話式 SQL のユーザー用に SQL パッケージが自動的に作成されます。しかし、対話式 SQL の最初の接続はローカル・システムに対して行われ、その接続は 2 フェーズ・コミット・プロトコルによって保護されているので、問題が生じる可能性があります。システムへの後続の接続が 1 フェーズ・コミットしか可能でなかったり、V5R1 より前の OS/400 クライアントから TCP/IP が使用されたりすると、その接続は読み取り専用になります。そのような接続でパッケージを自動的に作成しようとすると、パッケージの作成は更新と見なされ、読み取り専用の接続では行えないので失敗します。

この問題の解決策は、リモート・サーバー・システムに接続する前にローカル・データベースへの接続を終了することです。これは、RELEASE ALL コマンドの後に COMMIT を実行することによって行えます。それから、リモート・システムへの接続を行うことができ (最初の接続なので)、そこで更新を行うことができます。

対話式 SQL を開始する際には、*NONE 以外のコミットメント制御レベルを指定する必要があります。さらに、接続するために使用するユーザー ID は、サーバーで SQL パッケージを作成するための適切な権限がなければなりません。接続を試みる際に SQLSTATE 42501 を受け取る場合には、パッケージ作成権限を持っていないのかもしれません。

関連資料:

436 ページの『対話式 SQL 固有の接続の障害』

対話式 SQL から CONNECT ステートメントを実行しているときに、場合によっては、SQ30080 一般メッセージが出されます。

DDM ファイルのセットアップ

分散リレーショナル・データベース体系 (DRDA) サポートの IBM i 実装では、通信のために分散データ管理機能 (DDM) 会話を使用します。したがって、分散リレーショナル・データベース処理に関連して DDM を使用することができます。

DDM を使用して、サーバーにリモート・コマンドを投入し、システム間でテーブルをコピーし、非分散リレーショナル・データベース作業を別のシステムで処理することができます。

分散リレーショナル・データベースを使用すると、クライアントがデータベースとの接続に必要とする情報は、リレーショナル・データベース・ディレクトリーの中で提供されます。DDM を使用する時には、サーバー上で処理したい各ファイルごとに、別々の DDM ファイルを作成しなければなりません。DDM ファイルは、サーバー上のリモート・ファイルおよびサーバーへの通信経路を識別するために、クライアントのアプリケーションによって使用されます。

V5R2 の場合と同様、RDB ディレクトリー項目への参照がある DDM ファイルも作成することができます。「分散リレーショナル・データベースの管理」で説明するデータベース管理タスクは、DDM を使用してリモート・ファイルにアクセスします。DDM ファイルは、分散データ管理機能ファイルの作成 (CRTDDMF) コマンドを使用して作成します。DDM ファイルは、ファイルおよびファイルの中で名前を指定される通信経路が作成される前に、作成することができます。ただし、DDM ファイルの中で名前を指定されるファイル、および通信情報は、アプリケーションによる DDM ファイルの使用前に、作成されなければなりません。

以下に、DDM ファイルの作成方法の一例を示します。

```
CRTDDMF FILE (TEST/KC105TST) RMTLOCNAME(KC105)
        RMTFILE(SPIFFY/INVENT)
```

上記の例の DDM ファイル・アクセスが TCP/IP を介して行われる場合には、RMTLOCNAME パラメータの 2 番目の項目で *IP を指定する必要があります。

このコマンドは、KC105TST という名前の DDM ファイルを作成し、そのファイルをクライアントの TEST ライブラリーに保管するためのコマンドです。この DDM ファイルは、リモート・ロケーション KC105 を使用して、INVENT という名前のリモート・ファイルにアクセスします。このファイルは、サーバー・システムの SPIFFY ライブラリーに保管されています。

「DDM ファイルの処理」画面のオプションを使用して、DDM ファイルを変更、削除、表示、または作成することができます。

関連概念:

266 ページの『DRDA と DDM の管理』

分散リレーショナル・データベースの管理担当者は、複数のシステム上で行われる作業に責任を負うこととなります。

275 ページの『リモート・システムの操作』

分散リレーショナル・データベースでは、管理担当者は、リモート IBM i 製品を操作しなければならない場合があります。

関連資料:

分散データ管理ファイル作成 (CRTDDMF) コマンド

分散リレーショナル・データベースのテーブルへのデータのロード

分散リレーショナル・データベース環境のアプリケーションは、テーブルに保管されているデータに対して操作を行います。一般的に、アプリケーションを使用してテーブルに情報を照会し、1 つ以上のテーブルの行の挿入、更新、または削除を行い、あるいは新しいテーブルを作成します。その他の状況が生じて、1 つのシステムのデータを別のシステムに移動しなければならない場合もあります。

分散リレーショナル・データベースのテーブルへの新しいデータのロード

テーブルへのデータのロードは、各データ項目をテーブルに入力することによって行います。IBM i オペレーティング・システムでは、SQL、DB2 for i Query 管理機能の機能、または IBM i アプリケーション開発ツールのデータ・ファイル・ユーティリティー部分を使用して、データをテーブルに挿入するアプリケーションを作成することができます。

SQL を使用したデータのテーブルへのロード:

データをテーブルにロードする単純な方法としては、SQL アプリケーションおよび SQL INSERT 操作を使用する方法があります。

Spiffy 社では、地域センターから販売店へ定期的な在庫品出荷が行われているので、ある地域センターがある販売店の在庫テーブルに在庫品目を定期的に追加する必要がある状況を考えてみます。

```
INSERT INTO SPIFFY.INVENT
(PART, DESC, QTY, PRICE)
VALUES
('1234567', 'LUG NUT', 25, 1.15 )
```

前述のステートメントでは、SPIFFY という名前の SQL コレクションの中の INVENT という名前のテーブルに 1 行のデータを挿入します。

各定期出荷品目ごとに、SQL INSERT ステートメントによって、販売店の在庫テーブルに 1 行が挿入されます。前述の例で、15 の異なる品目が販売店に出荷される場合であれば、地域支社のアプリケーションには、15 の SQL INSERT ステートメントまたはホスト変数を使用する単一の SQL INSERT ステートメントを組み込むことができます。

この例では、地域センターは SQL アプリケーションを使用して、サーバーにあるテーブルにデータをロードしています。SQL の実行時のサポートは、IBM i ライセンス・プログラムの中で提供されているので、サーバーでは IBM DB2 Query Manager and SQL Development Kit for i ライセンス・プログラムは必要ありません。しかし、アプリケーションを作成するには、IBM DB2 Query Manager and SQL Development Kit for i が必要です。

関連概念:

SQL プログラミング

関連資料:

SQL リファレンス

IBM i Query 管理機能の機能を使用したテーブルおよびファイル内のデータ操作:

IBM i ライセンス・プログラムでは、DB2 for i Query 管理機能が提供されており、この機能を使用すれば、テーブルおよびファイルの中のデータを操作することができます。照会は、SQL 照会ステートメントを使用して作成します。

CL コマンドによって、あるいはアプリケーション・プログラムの中で照会呼び出し可能インターフェースを用いることによって、照会を実行することができます。Query 管理機能を使用すると、75 ページの『SQL を使用したデータのテーブルへのロード』で説明した在庫更新の場合に、次のようにして 1 行のデータをテーブルに挿入することができます。

ソース・メンバー INVLOAD をソース物理ファイル INVLOAD の中に作成し、次の SQL ステートメントを作成します。

```
INSERT INTO SPIFFY/INVENT
(PART, DESC, QTY, PRICE)
VALUES
(&PARTVALUE, &DESCVALUE, &QTYVALUE, &PRICEVALUE)
```

CL コマンドを使用して、Query 管理機能プログラム・オブジェクトを作成します。

```
CRTQMQR YQMQR(INVLOAD) SRCFILE(INVLOAD) SRCMBR(INVLOAD)
```

次の CL コマンドを実行すると、INSERT SQL ステートメントの結果が SPIFFY コレクションの INVENT テーブルに入ります。Query の中で変数 (&PARTVALUE、&DESCVALUE、など) を使用すると、所要の値を STRQMQR 呼び出しの一部として入力することができます。各行ごとに Query 管理機能プログラムを作成する必要はありません。

```
STRQMQRV QMQRV(INVLOAD) RDB(KC000)  
SETVAR((PARTVALUE '1134567')) (DESCVALUE ''Lug Nut'')  
(QTYVALUE 25) (PRICEVALUE 1.15))
```

Query 管理機能は動的です。つまり、そのアクセス・パスは、プログラムのコンパイル時ではなく、実行時に作成されることを意味します。したがって、DB2 for i Query 管理機能は、データをテーブルにロードする場合には、SQL アプリケーションほど効率的に機能しません。しかし、アプリケーションを作成するには、IBM DB2 Query Manager and SQL Development Kit for i 製品が必要です。SQL および Query 管理機能の実行時サポートは、IBM i ライセンス・プログラムの一部になっています。

関連概念:



Query 管理プログラミング (PDF)

データの入力、テーブルの更新、およびデータ・ファイル・ユーティリティーを使用した照会の作成:

データ・ファイル・ユーティリティー (DFU) は、IBM により提供される使用可能な IBM i アプリケーション開発ツール・パッケージの一部であり、データを入力し、テーブルを更新し、照会を行うためのプログラムの作成を援助するプログラム作成プログラムです。

DFU の使用には、プログラミング言語を必要としません。データ入力、保守、または照会プログラムは、一連の画面に対して応答することにより作成されます。DFU を使用する利点の 1 つは、それが本質的に汎用的であるために、SQL などのプログラミング言語を使用する場合に比べて、迅速にデータをテーブルにロードするためのデータベース更新プログラムを作成できるという点にあります。DDM ファイルと DFU を使用するか、または表示装置パススルーを使用してサーバーで DFU を実行することによって、リモート・システムにあるデータを処理することができます。

システム相互間のデータの移動

企業の運営においては、IBM i オペレーティング・システム相互間でデータを移動する必要のあるような状況が数多く発生します。

システム間でデータの移動が必要になる状況として、新しい販売店が地域内に開設され、既設の販売店の得意先の中には、その所在地によって、新設の販売店に移る場合があります。販売店が閉業したり、Spiffy 社の販売および保守サービスの代理業務を停止する場合もあるでしょう。そのような販売店の在庫および必要な保守サービス情報については、地域支社または他の域内販売店に割り振らなければなりません。販売店によっては、成長が著しく、システムのアップグレードが必要になり、データベース全体を新しいシステムに移動しなければならないことも考えられます。

システム相互間でデータを移動する方法としては、以下に挙げるものがあります。

- ユーザー作成のアプリケーション・プログラム
- 対話式 SQL (ISQL)
- DB2 for i Query 管理機能の機能
- テープ装置相互間でのコピー
- DDM の使用によるファイル・コピー・コマンド
- ネットワーク・ファイル・コマンド
- IBM i の保管および復元コマンド

ユーザー作成のアプリケーション・プログラムの作成:

分散作業単位 (DUW) 接続管理でコンパイルしたプログラムは、リモート・データベースとローカル・データベースに接続することができ、一方のシステムから FETCH し、もう一方に INSERT して、データを移動できます。

複数行 FETCH と複数行 INSERT を使用すると、レコードのブロックを一時に処理することができます。コミットメント制御を使用して、データの移動中にある時点でチェックポイントを取り、障害が起きたときに完全にコピーを再度開始しなくても済むようにすることができます。

対話式 SQL を使用したデータベースの Query:

SQL SELECT ステートメントおよび対話式 SQL を使用し、ローカル・システム上にあるテーブルの作成または更新を行うのに必要なデータに関し、別の IBM i オペレーティング・システム上にあるデータベースに照会することができます。

SELECT ステートメントでは、所要のデータが入っているテーブル名と列、およびどの行のデータを検索するのかが決める選択基準またはフィルターを指定することができます。SELECT ステートメントが正常に実行された場合には、指定したテーブルの 1 行または複数行が取り出される結果になります。

SQL を使用すると、1 つのテーブルからデータを取り出すだけでなく、結合操作を使用することによって、複数のテーブルに含まれている列から情報を取り出すことができます。SELECT ステートメントが正常に実行された場合には、指定したテーブル (複数可) の 1 行以上が取り出される結果になります。戻される行の列内のデータ値は、指定した表に含まれているデータ値を合成したものになります。

対話式 SQL 照会を使用すると、照会の結果は、ローカル・システムのデータベース・ファイルに入れることができます。コミットメント制御レベルが対話式 SQL 処理に指定された場合、それはサーバーに適用されます。ローカル・システムのデータベース・ファイルのコミットメント制御レベルは *NONE になります。

対話式 SQL を使用すると、次のことを行うことができます。

- 選択の結果を入れる新しいファイルを作成する。
- 既存のファイルを置き換える。
- ファイル内に新しいメンバーを作成する。
- メンバーを置き換える。
- 結果を既存のメンバーに追加する。

KC105 販売店が部品番号 1234567 の在庫全量を KC110 に転送する状況を考えてみます。KC110 では、KC105 から調達する部品について、KC105 データベースに照会します。この在庫照会の結果は、KC110 システムですでに存在しているデータベース・ファイルに戻されます。以下にこの作業を完了するために使用できる処理を示します。

SQL 対話式セッションの開始 (STRSQL) コマンドを使用して、「対話式 SQL」画面を表示します。新しいデータベースの SQL ステートメント (CONNECT 以外) を入力する前に、この操作の結果がローカル・システム上のデータベース・ファイルに送られるように指定します。これを行うためには、以下のステップに従ってください。

1. 「SQL ステートメントの入力」画面でサービスのオプションを選択してください。
2. 「サービス」画面でセッション属性変更のオプションを選択してください。
3. 「セッション属性」画面で出力装置選択のオプションを入力してください。

4. 出力装置フィールドにデータベース・ファイルを表す 3 を入力して、実行キーを押してください。次の画面が表示されます。

```

セッション属性変更

選択項目を入力して、実行キーを押してください。

ファイル . . . . . QSQLSELECT 名前
ライブラリー . . . . . QGPL 名前
メンバー . . . . . *FILE 名前、*FILE、*FIRST

オプション . . . . . 1 1= ファイルの作成
2= ファイルの置き換え
3= メンバーの作成
4= メンバーの置き換え
5= メンバーへの追加
権限 . . . . . *LIBCRTAUT *LIBCRTAUT、*CHANGE、*ALL
*EXCLUDE、*USE
権限リスト名:

テキスト . . . . .

F3= 終了 F5= 最新表示 F12= 取り消し

```

5. 結果を受け取るデータベース・ファイルの名前を指定してください。

データベース名が指定されたら、次の例のように対話式 SQL 処理を開始できます。

```

SQL ステートメントの入力

SQL ステートメントを入力して、実行キーを押してください。
現在の接続相手はリレーショナル・データベース KC000 である。
CONNECT TO KC105
現在の接続相手はリレーショナル・データベース KC105 である。
=====>SELECT * FROM INVENTORY
WHERE PART = '1234567'

____
____
____
____
____
____
____
____
____
____
____

終わり
F3= 終了 F4= プロンプト F6= 行の挿入 F9= コマンドの複写 F10= 行のコピー
F12= 取り消し F13= サービス F24= キーの続き

```

関連概念:

SQL プログラミング

関連資料:

SQL リファレンス

DB2 for i Query 管理機能の機能を使用したりモート・システムの Query:

リモート・システムに照会して、出力ファイルの中の結果をローカル・システムに戻す場合には、DB2 for i Query 管理機能の機能は、対話式 SQL とほとんど同じサポートを提供します。

対話式 SQL と Query 管理機能は、両方ともファイルまたはテーブルに関する INSERT、DELETE、SELECT などのデータ処理操作を実行することができ、そのためにテーブル (またはファイル) がコレクションの中にすでに存在している (ライブラリーの中に存在することができる) ことを必要としません。また、Query 管理機能では、新しいテーブルが照会の結果としてシステム上に作成される時には、SQL CREATE TABLE ステートメントを使用してデータ定義を提供します。Query 管理機能によって作成される表は、SQL を使用して作成される表に適用されるものと同じ指針および制約条件に従います。

ただし、Query 管理機能を使用する場合には、結果をファイルまたはテーブルに追加したい時に、メンバーを指定することはできません。Query 管理機能を使用した場合の結果は、**データベース・ファイルの一時変更 (OVRDBF)** コマンドを使用して、Query 管理機能を開始する前に別のメンバーを指定しない限り、最初のファイル・メンバーに入れられることとなります。

関連概念:



Query 管理プログラミング (PDF)

関連資料:

データベース・ファイルによるオーバーライド (OVRDBF) コマンド

テープ相互間でのファイルのコピー:

テープ相互間でファイルをコピーするために使用できるコマンドが、いくつかあります。

IBM i オペレーティング・システムの テープにコピー (CPYTOTAP) コマンドを使用すると、テープにテーブルまたはファイルをコピーできます。

テープ上のデータは、テープからのコピー (CPYFRMTAP) コマンドを使用して、別のシステムにロードできます。このコマンドの使用法の詳細については、ストレージ・ソリューションのトピックを参照してください。

さらに、ファイルのコピー (CPYF) コマンドを使用して、テープ上のデータを DB2 for i にロードできます。これは、DB2 for z/OS または DB2 Server for VM (SQL/DS) からアンロードされたデータをロードするときに特に便利です。これらのシステムからヌル値可能データをアンロードして、ヌル値可能フィールドに単一バイト・フラグを関連付けることができます。CPYF コマンドの FMTOPT パラメーターで *NULLFLAGS オプションが指定されていると、ヌル・フラグが識別されて、テープの隣接するフィールドのデータを無視します。そして DB2 for i のフィールドをヌルにします。IBM メインフレームからデータをインポートする際に役立つもう 1 つの FMTOPT パラメーター値は、*CVTFLOAT 値です。これは、z/OS 形式でテープ上に保管されている浮動小数点データを、DB2 for i で使用されている IEEE 形式に変換することを可能にします。

関連概念:

ストレージ・ソリューション

関連資料:

テープにコピー (CPYTOTAP) コマンド

テープからコピー (CPYFRMTAP) コマンド

ファイル・コピー (CPYF) コマンド

ファイル・コピー・コマンドを使用したシステム間のデータの移動:

IBM i オペレーティング・システム相互間でデータを移動する他の方法としては、ファイル・コマンドを DDM とともに使用して、データをコピーする方法があります。

ファイルのコピー (CPYF)、ソース・ファイルのコピー (CPYSRCF)、および照会ファイルからのコピー (CPYFRMQRYP) コマンドを使用して、クライアントとサーバーのファイル相互間でデータをコピーすることができます。リモート・データベース・ファイルとの間でローカル・リレーショナル・データベース・ファイルまたは装置ファイルをコピーすることができ、リモート・ファイルをリモート・ファイルへコピーすることもできます。

たとえば、ある販売店が閉業した場合には、分散リレーショナル・データベース管理担当者は、リモート・システムから地域のローカル・システムへ得意先テーブルおよび在庫テーブルをコピーすることができます。管理担当者は、これらのテーブルをアクセスおよびコピーするために、適正に許可されたユーザー・プロファイルをサーバー上で必要とし、コピーする各テーブルまたは各ファイルごとに、DDM ファイルをクライアント上に作成しなければなりません。次の例には、SPIFFY と呼ばれるコレクションの中の INVENT と呼ばれるテーブルを、KC105 というリモート・ロケーション名を持つシステムから、KC000 と呼ばれる地域中央システムへコピーする場合に、データベース管理担当者が使用するコマンドが示されています。KC000 クライアント上の TEST と呼ばれるライブラリーの中の INCOPY と呼ばれる DDM ファイルが、ファイル・アクセス用として使用されます。これらのコマンドは KC000 システム上で実行されます。

```
CRTDDMF FILE(TEST/INCOPI) RMTFILE(SPIFFY/INVENT)
      RMTLOCNAME(KC105)
CPYF FROMFILE(TEST/INCOPI) TOFILE(TEST/INVENTDDM)
      MBROPT(*ADD)
```

この例では、管理担当者は KC000 システム側でコマンドを実行します。管理担当者が KC000 システム側にいない場合には、パススルーを使用して、KC000 システムでこれらのコマンドを実行しなければなりません。SBMRMTCMD (リモート・コマンド投入) コマンドを使用して前述のコマンドを実行することはできません。IBM i オペレーティング・システムが、同一のジョブに関して、クライアントおよびサーバーになることはできないからです。

このコマンドを DDM で使用するにあたっては、次の事項を考慮してください。

- DDM ファイルは、ファイルのコピー (CPYF) コマンドおよびソース・ファイルのコピー (CPYSRCF) コマンドでは、FROMFILE パラメーターおよび TOFILE パラメーターで指定することができます。

注: 照会ファイルからのコピー (CPYFRMQRYP) およびテープからのコピー (CPYFRMTAP) コマンドでは、DDM ファイル名は、TOFILE パラメーターでしか指定することができません。また、テープへのコピー (CPYTOTAP) コマンドでは、DDM ファイル名を指定できるのは FROMFILE パラメーターだけです。

- 削除可能ファイルを削除不能ファイルにコピーする時には、COMPRESS(*YES) を指定しなければなりません。そうしないと、エラー・メッセージが送られて、ジョブが終了します。
- DDM ファイル上のリモート・ファイル名にメンバー名を指定する場合には、ファイルのコピー (CPYF) コマンドでそのファイルに指定されるメンバー名は、DDM ファイル上のリモート・ファイル名のメンバー名と同じでなければなりません。さらに、データベース・ファイルの一時変更 (OVRDBF) コマンドでは、DDM ファイル上のリモート・ファイル名のメンバー名と異なるメンバー名を指定することはできません。
- DDM ファイルがメンバー名を指定せず、データベース・ファイルの一時変更 (OVRDBF) コマンドでそのファイルのメンバー名を指定した場合には、ファイルのコピー (CPYF) コマンドでは、OVRDBF コマンドで指定したメンバー名を使用します。
- TOFILE パラメーターが存在していないファイルを参照する DDM ファイルである場合には、CPYF でそのファイルを作成します。ファイルのコピー (CPYF) コマンドで作成したリモート・ファイルについては、以下の特別な考慮事項に留意してください。

- ターゲット DDM ジョブのためのユーザー・プロファイルは、サーバー上の物理ファイルの作成 (CRTPF) コマンドに対して許可されなければなりません。
- IBM i サーバー・システムでは、TOFILE パラメーターは、『データベース・ファイル管理』のトピックに説明されているものを除いて、FROMFILE パラメーターの属性のすべてを備えています。
- TCP/IP を使用している場合には、分散データ管理ファイルの作成 (CRTDDMF) コマンドの RMTLOCNAME パラメーターの 2 番目の項目は *IP でなければなりません。

関連概念:

データベース・ファイル管理

関連資料:

ファイル・コピー (CPYF) コマンド

ソース・ファイル・コピー (CPYSRCF) コマンド

Query ファイルからコピー (CPYFRMQRYF) コマンド

テープからコピー (CPYFRMTAP) コマンド

テープにコピー (CPYTOTAP) コマンド

物理ファイル作成 (CRTPF) コマンド

分散データ管理ファイル作成 (CRTDDMF) コマンド

データベース・ファイルによるオーバーライド (OVRDBF) コマンド

リモート・コマンド投入 (SBMRMTCMD) コマンド

ネットワーク・ファイル・コマンドを使用したネットワーク上のデータの転送:

データは、システム・ネットワーク体系 (SNA) 配布サービス (SNADS) をサポートするネットワーク・プロトコルを用いて転送できます。分散リレーショナル・データベース処理で使用される APPC および APPN プロトコルに加えて、SNADS は、2 進データ同期通信同等リンク (BSCCL) および SNA アップライン機能 (SNUF) プロトコルを使用することができます。

SNADS がサポートする IBM i 環境は、ネットワーク・ファイル送信 (SNDNETF) コマンドを用いて別のシステムへデータを送信し、ネットワーク・ファイル受信 (RCVNETF) コマンドおよび ネットワーク・ファイルの処理 (WRKNETF) コマンドを用いて、別のシステムからネットワーク・ファイルを受信できます。

関連資料:

ネットワーク・ファイル受信 (RCVNETF) コマンド

ネットワーク・ファイル送信 (SNDNETF) コマンド

ネットワーク・ファイル処理 (WRKNETF) コマンド

オブジェクト保管および復元コマンドを使用したテーブルの移動:

オブジェクト保管 (SAVOBJ) コマンドおよびオブジェクト復元 (RSTOBJ) コマンドを使用して、別のシステムからテーブルを移動することができます。保管コマンドでは、テープまたは保管ファイルにデータベース・ファイルを保管します。保管ファイルは、通信によって別のシステムに配布することができます。

テーブルまたはファイルの保管および復元に使用される保管および復元コマンドには、次のものがあります。

- **ライブラリーの保管 (SAVLIB)** コマンドは、1 つ以上のコレクションまたはライブラリーを保管します。
- **オブジェクトの保管 (SAVOBJ)** コマンドでは、1 つ以上のオブジェクト (データベースのテーブルおよびビューを含む) を保管します。

- **変更されたオブジェクトの保管 (SAVCHGOBJ)** コマンドは、コレクションまたはライブラリーが最後に保管された時点以降、または指定された日付以後に変更されたオブジェクトを保管します。
- **ライブラリー復元 (RSTLIB)** コマンドは、コレクションまたはライブラリーを復元します。
- **オブジェクトの復元 (RSTOBJ)** コマンドでは、1 つ以上のオブジェクト (データベースのテーブルおよびビューを含む) を復元します。

例えば、2 つの販売店が合併している場合には、保管および復元コマンドを使用して、1 つのリレーショナル・データベースのコレクションおよびテーブルを保管し、その後で残っているシステムのリレーショナル・データベースに復元することができます。そのためには、管理担当者は次のことを行います。

1. システム A に対して**ライブラリーの保管 (SAVLIB)** コマンドを使用してコレクションを保管するか、またはシステム A に対して**オブジェクトの保管 (SAVOBJ)** コマンドを使用してテーブルを保管する。
2. データを保管ファイル (SNADS を使用して配布できるファイル) に保管するのか、あるいはテープに保管するのかを指定する。
3. 保管ファイルをシステム B に配布するか、あるいはテープをシステム B に送付する。
4. システム B に対して**ライブラリーの復元 (RSTLIB)** コマンドを使用してコレクションを復元するか、またはシステム B に対して**オブジェクトの復元 (RSTOBJ)** コマンドを使用してテーブルを復元する。

保管および復元コマンドの使用に際して考慮すべき事項の 1 つは、復元されるオブジェクトに対する所有権および許可です。現在のオブジェクト所有者の有効なユーザー・プロファイルが、オブジェクトが復元されるシステムに存在していません。現在の所有者のプロファイルがこのシステムに存在していない場合には、オブジェクトは QDFTOWN デフォルト・ユーザー・プロファイルのもとで復元されます。ユーザーのオブジェクトに対する許可は、デフォルト・ユーザー・プロファイル・パラメーターによって制限されます。QSECOFR 権限を持っているユーザーは、元の所有者のプロファイルをこのシステム上に作成して、復元されたオブジェクトに変更を加えるか、あるいはこのオブジェクトに対する新しい許可を、ローカル・ユーザーとリモート・ユーザーの両方に指定するか、どちらかを行わなければなりません。

関連概念:

バックアップおよび回復

関連資料:

ライブラリー復元 (RSTLIB) コマンド

オブジェクト復元 (RSTOBJ) コマンド

変更されたオブジェクトの保管 (SAVCHGOBJ) コマンド

ライブラリー保管 (SAVLIB) コマンド

オブジェクト保管 (SAVOBJ) コマンド

IBM i 以外のシステムから IBM i へのデータベースの移動

別の IBM システムから IBM i オペレーティング・システムへ、または IBM 以外のシステムから IBM i へ、ファイルを移動する必要がある場合があります。このトピックでは、IBM i 以外のシステムから IBM i にデータを移動する代替方法をリストします。ただし、それぞれの使用に関する特定の指示事項については、他のシステムとともに提供されているか、またはアプリケーションとして指定されている資料を参照する必要があります。

別の IBM システムからのデータ移動:

別の IBM システムから IBM i オペレーティング・システムへデータを移動する場合に使用できる方法はたくさんあります。



ここにリストされている方法を使用してデータを移動できます。

- 高水準言語プログラムを作成して、別のシステムからデータを取り出すことができます。これに対応するシステムのプログラムを使用して、データをロードすることができます。
- 他の Distributed Relational Database Architecture (DRDA) 実装をサポートするシステムの場合、SQL 機能を使用してデータを移動できます。分散作業単位を使用すれば、ソース・データに対して照会をオープンし、その同じ作業単位で、システム上のテーブルにデータを挿入することができます。パフォーマンスを最高にするためには、この照会でブロック化を使用し、システムでは複数行挿入を行ってください。
- データは、他のシステム上のテーブルおよびファイルから抽出して、テープにコピーして IBM i オペレーティング・システムに送付するか、あるいは通信回線を用いて送信することができます。
 - DB2 for z/OS データベースからの場合は、データベース・マネージャーに付属している、DSNTIAUL と呼ばれるサンプル・プログラムを使用して、ファイルまたはテーブルからデータを抽出することができます。
 - DB2 Server for VM (SQL/DS) データベースからの場合は、データベース・マネージャーのデータベース・サービス・ユーティリティ部分を使用して、データを抽出することができます。
 - DB2 for z/OS および DB2 Server for VM データベースの両方から、データを抽出するためにデータ抽出 (DXT™) を使用できます。ただし、DXT によるヌル・データの処理は、以下に説明するようにファイル・コピーによるヌル・データの処理と互換性がありません。したがって、IBM i への移行のためにリレーショナル・データをアンロードする目的で DXT を使用することはお勧めできません。
 - IMS/DB 階層データベースから、DXT を使用してデータを抽出することができます。
- DB2 for z/OS または DB2 Server for VM のデータベースからコピーする場合は、標準テープ管理技法を使用してデータをテープに書き出します。IBM i オペレーティング・システムでは、**テープからコピー (CPYFRMTAP)** コマンドを使用して、テープからデータをロードします。ただし、**ファイル・コピー (CPYF)** コマンドを使用すると、IBM メインフレーム・コンピューターからデータを移行するための特別なサポートを受けることができます。**テープ・ファイル一時変更 (OVRTAPF)** コマンドを使用して、テープ・データに **CPYF** を行うこともできます。**OVRTAPF** コマンドでは、特別なテープ固有のパラメーターを指定することができます。このパラメーターは IBM i 以外のシステムからデータをインポートする際に必要な場合があります。

特別な **CPYF** サポートにより、ヌル値可能データおよび浮動小数点データをインポートすることができます。メインフレームからヌル値可能データをアンロードして、ヌル値可能フィールドに単一バイト・フラグを関連付けることができます。FMTOPT パラメーターで *NULLFLAGS オプションが指定されていると、**ファイルのコピー (CPYF)** コマンドはヌル・フラグを識別して、テープの隣接するフィールドのデータを無視します。そして DB2 for i のフィールドをヌルにします。IBM メインフレームからデータをインポートする際に役に立つ別の FMTOPT パラメーター値は、*CVTFLOAT 値です。これは、z/OS 形式でテープ上に保管されている浮動小数点データを、DB2 for i で使用されている IEEE 形式に変換することを可能にします。

IBM i オペレーティング・システムでのテープ装置の使用に関する詳細については、『ストレージ・ソリューション』のトピックを参照してください。

- 通信回線を用いてデータを送信する場合は、IBM i オペレーティング・システム上の SNADS サポートによって処理することができます。SNADS サポートでは、分散リレーショナル・データベースの処理に使用される拡張プログラム間通信機能 (APPC) または 拡張対等通信ネットワーク機能 (APPN) プロトコルに加えて、BSCCEL および SNUF プロトコル用のネットワーク・ファイルを転送します。
 - MVS™ システムからの場合は、データは TSO XMIT 機能を使用して、IBM i オペレーティング・システムに送信することができます。システムは、**ネットワーク・ファイルの処理 (WRKNETF)** または **ネットワーク・ファイルの受信 (RCVNETF)** コマンドを使用して、ネットワーク・ファイルを受信します。

- VM システムからの場合は、データは SENDFILE 機能を使用して、IBM i オペレーティング・システムに送信することができます。システムは、**ネットワーク・ファイルの処理 (WRKNETF)** または **ネットワーク・ファイルの受信 (RCVNETF)** コマンドを使用して、ネットワーク・ファイルを受信します。
- Microsoft Windows からは、クライアント・データは IBM i Access (別途注文可能な IBM 製品) を使用して IBM i に送信することができます。
- 各種のワークステーション・クライアントからの場合は、DB2 for Linux, UNIX, and Windows の IMPORT および EXPORT ユーティリティを使用して IBM i との間でデータを相互にコピーできます。なお IMPORT では、既存のテーブルにのみデータをインポートすることができます。IMPORT ユーティリティと EXPORT ユーティリティの例については、Advanced Functions and Administration on DB2 Universal Database™ for iSeries, SG24-4249  Redbooks® を参照してください。この資料には、IMPORT ユーティリティと EXPORT ユーティリティで使用できるファイル・タイプとデータ・フォーマットに関する情報も含まれています。
- データはまた、非同期通信など、SNADS をサポートしない通信回線によって送信することもできます。IBM i ライセンス・プログラムの一部となっているユーティリティのファイル転送サポート (FTS) を使用して、データの送受信を行うことができます。通信および通信ファイルの処理に関する詳細は、ICF Programming  を参照してください。

関連概念:

ストレージ・ソリューション

関連資料:

テープからコピー (CPYFRMTAP) コマンド

ファイル・コピー (CPYF) コマンド

データベース・ファイルによるオーバーライド (OVRDBF) コマンド

ネットワーク・ファイル受信 (RCVNETF) コマンド

43 ページの『ヒント: 分散リレーショナル・データベース・アプリケーションの設計』

分散リレーショナル・データベース・アプリケーションと、ローカル・データベースだけに使用するために開発されたアプリケーションとは、要件に大きな違いがあります。

ネットワーク・ファイル処理 (WRKNETF) コマンド

IBM 以外のシステムからのデータ移動:

IBM 以外のシステムからファイルまたはテーブルをテープにコピーし、それらのファイルを IBM i オペレーティング・システムにロードすることができます。

コピー元インポート・ファイル (CPYFRMIMPF) コマンドを使って、これを行ってください。

販売会社独自の通信機能も、別の 2 つの IBM i ライセンス・プログラムを介してサポートされます。

ローカル・エリア・ネットワークと広域ネットワークの両方のための対等接続機能が、伝送制御プロトコル/インターネット・プロトコル (TCP/IP) によって提供されています。IBM i ライセンス・プログラム用 TCP/IP 接続ユーティリティのファイル転送プロトコル (FTP) を使用すれば、リモート・システムの機能に応じて、多くのタイプのファイルを受信することができます。

関連概念:

TCP/IP のセットアップ

関連資料:

インポート・ファイルからコピー (CPYFRMIMPF) コマンド

セキュリティー

IBM i オペレーティング・システムには、サーバーのデータ資源に対するアクセスを制限するセキュリティー要素が組み込まれています。セキュリティー・オプションの範囲は、単純な物理的セキュリティーから、コマンドおよびデータ・オブジェクトに対する許可と組み合わせられた、完全なパスワード・セキュリティーにまで及んでいます。

ユーザーは、データベースがローカルであれリモートであれ、データベースにアクセスするための適正な許可を与られていなければなりません。ユーザーはまた、アプリケーション・プログラムを実行するのに必要な、コレクション、テーブルなどのリレーショナル・データベース・オブジェクトに対する適正な許可も持っていなければなりません。つまり、分散データベースのユーザーは、ネットワーク内で使うデータベースについての、有効なユーザー・プロファイルを持っていなければならないということです。セキュリティー計画で、ネットワーク内におけるユーザーおよびアプリケーション・プログラムの要件を考慮しなければなりません。

分散リレーショナル・データベース管理担当者は、次の 2 つのセキュリティー問題に対処しなければなりません。

- システム間保護
- リモート・サイトのユーザーの識別

2 つ以上のシステムがセットアップされていて、相互のデータベースにアクセスする場合には、通信回線の向こう側が確かに通信先として意図したロケーションであり、侵入者ではないことを確認することが大切です。DRDA でリモート・リレーショナル・データベースにアクセスするとき、IBM i で拡張プログラム間通信 (APPC) および 拡張対等通信ネットワーク機能 (APPN) の通信構成機能を使用する場合は、この必要なネットワーク・レベル・セキュリティーを行うオプションが用意されています。

分散リレーショナル・データベース管理担当者にとって 2 番目に重要になるのは、データ・セキュリティーが、データを保管しているシステムによって維持されるということです。分散リレーショナル・データベースでは、データベースがローカルであるかリモートであるかにかかわらず、データベースにアクセスするには、ユーザーは適正な権限を (システムのセキュリティー・レベルに応じて) 認可されていなければなりません。分散リレーショナル・データベースのネットワーク・ユーザーは、サーバーでどんなジョブを実行する場合でも、サーバーでユーザー ID によって正しく識別されなければなりません。APPC/APPN および TCP/IP 通信プロトコルの両方を使用する分散リレーショナル・データベース・アーキテクチャー (DRDA) サポートでは、接続要求とともにユーザー ID とパスワードの送信を行います。

このトピック・コレクションでは、リモート・リレーショナル・データベースに対する通信および DRDA アクセスに関連するセキュリティー・トピックについて説明します。そして、APPC ネットワーク接続の会話レベルのセキュリティーと、それに対応する DRDA アプリケーションによって開始される TCP/IP 接続のセキュリティーのレベルの相違点を説明します。これから行われるセキュリティーの説明では、ユーザーという用語は、通信ジョブを開始しているリモート・ユーザーを含みます。

関連資料:

47 ページの『分散リレーショナル・データベースのセキュリティーの考慮事項』

分散リレーショナル・データベース計画の一部として、分散データの保護について決定しなければならない事項があります。

分散リレーショナル・データベースのセキュリティー要素

分散リレーショナル・データベース管理担当者は、ネットワーク内のクライアントによるデータのアクセスを不必要に制限せずに、ネットワーク内のサーバーのリソースを保護する必要があります。

クライアントは、分散リレーショナル・データベース・プログラムに対して許可されたユーザーだけがアクセスできるようにオブジェクトとリレーショナル・データベースを保護します。これは、通常の IBM i オブジェクト許可を使用してユーザーを識別してから、各ユーザー（またはユーザー・グループ）がオブジェクトに対して何を行うことができるか指定することによって可能になります。あるいは、SQL GRANT と REVOKE ステートメントを使用してテーブル、ビュー、および SQL パッケージに対する権限を許可したり取り消したりできます。クライアント上の SQL オブジェクトに対して権限レベルを提供すると、別のシステムのデータをアクセスする SQL アプリケーションに対して許可ユーザーだけがアクセスできるようにすることができます。

サーバーで使用されているシステム・セキュリティ・レベルにより、クライアントからの要求が受け入れられるかどうか、およびリモート・ユーザーがサーバー上のオブジェクトに対して権限を認可されているかどうかが決まります。

分散リレーショナル・データベース・ネットワーク内の IBM i 環境に関するセキュリティ計画には、以下に挙げるような側面があります。

- 機密のテーブル、プログラム、およびパッケージなど、特定のリソースに対するユーザーのアクセスを規制するオブジェクト関連セキュリティ。
- ネットワーク内の正当な他システムであることを検査するロケーション・セキュリティ。
- ローカル・システムおよびリモート・システム上の正当なユーザーであること、およびその権限を検査するためのユーザー関連セキュリティ。
- 回線記述で構成することができ、経路選択プロセスで使用される、システム、モデム、通信回線、および端末を囲むドアの施錠や建物の警備保障など、物理的なセキュリティ。

ロケーション・セキュリティ、ユーザー関連セキュリティ、およびオブジェクト関連セキュリティが使用できるのは、システム・セキュリティ・レベルが 20 またはそれ以上のレベルにセットされている場合だけです。

拡張プログラム間通信 (APPC) 会話では、システムがレベル 10 のセキュリティを使用している場合、無保護システムとして IBM i オペレーティング・システムはネットワークに接続します。システムはセッション確立時にリモート・システムの身元を妥当性検査することはなく、着信するプログラム開始要求では会話セキュリティを必要としません。レベル 10 では、APPC リモート・ロケーションで構成されたセキュリティ情報は無視され、セッションまたは会話の確立時には使用されません。システム上でユーザー・プロファイルが存在しない場合には、それは作成されます。

システムがレベル 20 以上のセキュリティを使用している場合、保護システムとして IBM i オペレーティング・システムはネットワークに接続します。このようにして、システムでは、会話レベル・セキュリティ機能を実現でき、APPC の場合にはセッション・レベル・セキュリティも実現できます。

ネットワーク内のシステムで同じレベルのシステム・セキュリティを設定すると、セキュリティの管理担当者の作業はより容易なものになります。サーバーは、クライアントで予期されているものを指定してセッションを確立することによって、セッションおよび会話が確立できるかどうかを制御します。たとえば、クライアントのセキュリティ・レベルが 10 に設定されており、サーバーのセキュリティ・レベルが 10 よりも大きい場合には、適切な情報が送信されない場合があり、これらシステムの中の 1 つのセキュリティ要素を変更しなければセッションは確立されないかもしれません。

DRDA アクセス用のパスワード

リモート・ユーザーにデータベース・アクセスを許可するための最も一般的な方法は、接続時にユーザー ID とパスワードを渡すことです。

組み込み SQL の使用時に、アプリケーション・プログラマーが使用できる 1 つの方法は、SQL CONNECT ステートメントに USER/USING 節をコーディングすることです。以下に組み込み SQL の例を示します。

```
EXEC SQL CONNECT TO :rdbname USER :userid USING :pwd
```

動的インターフェースの使用時には、これらは通常は接続 API 上で渡されます。以下に CLI の例を示します。

```
SQLConnect(rdbname, userid, pwd)
```

リモート・リレーショナル・データベースでの分散リレーショナル・データベース・アーキテクチャ (DRDA) アクセスでは、会話がいったん確立されると、パスワードを再度入力する必要はありません。リモート作業単位 (RUW) 接続管理方式で実行している際に RELEASE、DISCONNECT、または CONNECT ステートメントのいずれかを使用して接続を終了すると、最初のサーバーとの会話は除去される場合もあり、除去されない場合もあります。これは、接続しているサーバーの種類とクライアントのジョブ属性にかかっています (特定の規則については、「DDM 会話の制御」を参照してください)。最初のサーバーとの会話が除去されていない場合には、2 番目のサーバーと接続している間は未使用のまま残されます。最初のサーバーに再度接続した際に会話で使用されていない場合、ユーザー ID とパスワードを入力しなくても会話は再度活動状態になります。この会話を 2 度目に使用するときにも、パスワードが再度検査されることはありません。

ローカル接続が 3 部構成の名前を使用して接続を開始しようとする場合には、ローカル接続上で渡されるユーザー ID とパスワードを使用して、新しいリモート接続を開始します。ホスト・サーバーなど、ローカル接続からパスワードを検索できない環境もあります。したがって、パスワードの伝搬にサーバー認証項目を使用する必要があります。

関連概念:

APPC、APPN、および HPR

セキュリティ

関連資料:

305 ページの『DDM 会話の制御』

通常、クライアント・システム・ジョブに関連した DDM 会話は、このトピックに記述された条件のいずれかを満たすまでは、活動状態に保持されます。

APPC ネットワークでのセキュリティの要素

Distributed Relational Database Architecture (DRDA) を使用する場合、DRDA 環境の各システムのデータ・リソースを保護する必要があります。

DRDA 環境の各システムのデータ・リソースを保護するために、以下のパラメーターで制御される、3 つのグループのセキュリティ要素を使用できます。

- システム関連のセキュリティまたはセッションの場合、クライアント・システムとサーバー・システムの間で拡張プログラム間通信 (APPC) セッションが最初に確立される場合には、それらのシステム間で交換されるシステム検証パスワードを示すために、各システムで LOCPWD パラメーターが使用されます。どちらのシステムも、セッションを開始する前に同じパスワードを交換する必要があります。(システム/36 では、このパスワードのことをロケーション・パスワードといいます。) APPC ネットワークでは、装置記述の作成 (APPC) (CRTDEVAPPC) コマンドの LOCPWD パラメーターは、このパスワードを指定します。装置は APPN を使用して自動的に作成され、リモート・ロケーション・リストのロケーション・パスワードは、身元を検査するために 2 つのロケーションで使用するパスワードを指定します。構成リスト作成 (CRTCFGL) コマンドを使用して、リモート・ロケーション・リストのタイプ (*APPNRMT) を作成します。

- ユーザー関連またはロケーション・セキュリティーの場合、すでにクライアント・システムによってセキュリティーが確認された着信アクセス要求を (サーバー・システムとして) 受け入れるかどうか、あるいはユーザー ID と暗号化されたパスワードが必要かどうかを示すため、各システムで SECURELOC パラメーターが使用されます。APPC ネットワークでは、装置記述の作成 (APPC) (CRTDEVAPPC) コマンドの SECURELOC パラメーターによって、ローカル・システムがリモート・システムによるセキュリティーの検査を許可するかどうかを指定します。装置は APPN を使用して自動的に作成され、APPN リモート構成リストのロケーション保護を使用して、ローカル・システムがリモート・システムによるユーザー・セキュリティー情報の検査を許可するかどうかが決まります。SECURELOC 値は、リモート・ロケーションごとに、個別に指定できます。

SECURELOC パラメーターは、以下のセキュリティー要素と共に使用されます。

- クライアント・システムによって送信されたユーザー ID (このパラメーターで許可されている場合)。
- ユーザー ID と暗号化されたパスワード (このパラメーターで許可されている場合)。
- デフォルトユーザー・プロファイルを含む、サーバー・システムのユーザー・プロファイル。

詳細は、『APPC ネットワークでの DRDA サーバーのセキュリティー』のトピックを参照してください。

- オブジェクト関連のセキュリティーの場合、他のシステムによって IBM i オペレーティング・システムのファイルに完全にアクセスできるかどうか、そしてアクセスできる場合には、どのレベルのセキュリティーで着信要求を検査するかを示すため、ネットワーク属性変更 (CHGNETA) コマンドで DDMACC パラメーターが使用されます。
 - DDMACC パラメーター上で *REJECT が指定されている場合、サーバー・システムで受信される DRDA 要求はすべて拒否されます。
 - DDMACC パラメーターに *OBJAUT が指定されている場合には、通常のオブジェクト・レベルのセキュリティーがサーバー・システム上で使用されます。
 - DDMACC パラメーターで出口プログラム (またはアクセス制御プログラム) の名前が指定されると、追加のセキュリティーのレベルが使用されます。特定クライアント・システムの指定ユーザーが、サーバー・システムの特定のファイル (規則的に) アクセスするのに特定のコマンドを使用できるかどうかを制御するために、ユーザー出口プログラムを使用することができます。(詳細は、『追加機密保護のための DDM サーバー・アクセス制御出口プログラム』を参照してください。)
 - DRDA を使用してサーバー・システム上でファイルを作成する場合、指定されるライブラリー名にはそのファイルが含まれます。DRDA 要求でライブラリー名が指定されない場合、現行ライブラリー (*CURLIB) が使用されます。デフォルトのファイル権限を使用すると、ファイルを作成したユーザーがサーバー・システムのセキュリティーの責任担当者だけがファイルにアクセスできます。

リモート・ファイル・アクセスを制限するためのほとんどのセキュリティー管理は、サーバー・システムによって扱われます。クライアント・システムによって指定されるユーザー ID を除き、これらのすべての要素がサーバー・システム上で指定されて使用されます。しかし、クライアント・システムでは、クライアント・システム上の DRDA ファイルへのアクセスを制御し、必要な場合にユーザー ID をサーバー・システムに送信することにより、サーバー・システム・ファイルへのアクセスも制限します。

関連概念:

91 ページの『APPC ネットワークでの DRDA サーバーのセキュリティー』

サーバー・システムが IBM i オペレーティング・システムの場合、複数の要素を同時に使用して、リモート・ファイルへアクセスする要求が許可されているか判別します。

関連資料:

ネットワーク属性変更 (CHGNETA) コマンド
構成リスト作成 (CRTCFGL) コマンド
装置記述作成 (APPC) (CRTDEVAPPC) コマンド

APPN 構成リスト:

APPC ネットワークでは、2 つのロケーションが相互間で終端間 (エンドツーエンド) セッションを持つ場合、ロケーション・パスワードが指定されます。

中間ノードであるロケーションでは、ロケーション・パスワードを指定する必要はありません。

リモート・ロケーション・リストは**構成リスト作成 (CRTCFGL)** コマンドで作成され、これには、すべてのリモート・ロケーション、それぞれのロケーション・パスワード、およびリモート・ロケーションが保護されているかどうかを示すリストが含まれています。IBM i オペレーティング・システムには、システム全体のリモート・ロケーション構成リストが 1 つ存在します。中央側のシステムは、制御言語 (CL) プログラムを送ることによって、リモート・システムのロケーション・リストを作成できます。

構成リスト変更 (CHGCFGL) コマンドを使用して、リモート構成リストに変更を加えることができますが、変更が有効になるのは、そのロケーションのすべての装置がいずれもオフに構成変更された状態になってからです。

構成リスト表示 (DSPCFGL) コマンドが使用されたとき、パスワードの存在を示す表示はありません。パスワードが入力された場合には、**構成リスト変更 (CHGCFGL)** コマンドは *PASSWORD をフィールドに入れることによってパスワードの存在を示します。パスワードを表示する方法はありません。ロケーションのセキュリティーを設定する際に問題に直面する場合には、両方のシステムにパスワードを再度入力して、パスワードが一致することを確認する必要があります。

関連概念:

APPC、APPN、および HPR

関連資料:

構成リスト変更 (CHGCFGL) コマンド
構成リスト作成 (CRTCFGL) コマンド
構成リスト表示 (DSPCFGL) コマンド

会話レベルのセキュリティー:

システム・ネットワーク体系 (SNA) 論理装置 (LU) 6.2 の体系では、異なったシステムのネットワークで一貫した会話セキュリティーを提供するために、SNA ネットワークのさまざまなタイプのシステムが使用できる 3 つの会話セキュリティーの指定方法を示しています。

SNA セキュリティーのレベルには、以下のものがあります。

SECURITY(NONE)

通信を確立するのに、ユーザー ID もパスワードも送信されない。

SECURITY(SAME)

ローカル・サーバーと同じユーザー ID でユーザーをリモート・サーバーに署名させる。

SECURITY(PGM)

通信のために、ユーザー ID とパスワードの両方が送信される。

SECURITY(PROGRAM_STRONG)

パスワードが暗号化されずに送信されない場合のみ、通信のためにユーザー ID とパスワードの両方が送信され、そうでない場合には、エラーが報告される。これは IBM i の DRDA ではサポートされていません。

IBM i オペレーティング・システムは、会話セキュリティーの 4 つの SNA レベルをすべてサポートしますが、DRDA は最初の 3 つのレベルのみを使用します。ターゲットでは、会話で使用される SNA 会話レベルを制御します。

SECURITY(NONE) レベルでは、ターゲットはユーザー ID またはパスワードを要求しません。ターゲット上のデフォルトのユーザー・プロファイルを使用して会話することが許可されています。会話でデフォルトのユーザー・プロファイルが使用できるかどうかは、通信項目追加 (ADDCMNE) コマンドの DFTUSR パラメーターまたは当該サブシステムの通信項目変更 (CHGCMNE) コマンドで指定される値に依存しています。DFTUSR パラメーターの値が *NONE の場合は、サーバーでは、ターゲット上のデフォルトのユーザー・プロファイルを使用する会話が許可されていないことを示しています。SECURITY (NONE) は、パスワードとユーザー ID が提供されておらず、ターゲットで SECURELOC(*NO) が指定されているときに送信されます。

SECURITY(SAME) レベルでは、リモート・システムの SECURELOC 値によって、どんなセキュリティー情報が送信されるかが決まります (リモート・システムが IBM i 製品であることを想定)。SECURELOC 値が *NONE である場合、あたかも SECURITY(NONE) が要求されたかのように、ユーザー ID とパスワードは送信されません。SECURELOC 値が *YES である場合、ユーザー・プロファイルの名前が抽出され、ローカル・システムによってパスワードがすでに検査されたという旨の指示とともに送信されます。SECURELOC 値が *VFYENCPWD である場合、ユーザー・プロファイルとそれに関連したパスワードは、パスワードが暗号化されてその値が秘密にされた後にリモート・システムに送信されます。したがって、DRDA を使用するには、ユーザーは両方のシステムで同一のユーザー・プロファイル名とパスワードを持つ必要があります。

注: これら 3 つのオプションの中で SECURELOC(*VFYENCPWD) は一番安全なものであると言えます。なぜなら、SECURELOC(*VFYENCPWD) では最も多くの情報がリモート・サーバーによって検査されるからです。しかし、このオプションでは、ユーザーが複数のサーバー上で同一のパスワードを持つ必要があります。この場合、ユーザーが 1 つのサーバーのパスワードを変更しても、他のサーバーのパスワードを変更しないと問題が生じる可能性があります。

SECURITY(PGM) レベルでは、会話を行うために、ソースからのユーザー ID とパスワードの両方がターゲットによって要求されます。パスワードは会話が確立される際に検査され、それ以降の会話では無視されます。

関連資料:

通信項目追加 (ADDCMNE) コマンド

通信項目変更 (CHGCMNE) コマンド

APPC ネットワークでの DRDA サーバーのセキュリティー:

サーバー・システムが IBM i オペレーティング・システムの場合、複数の要素を同時に使用して、リモート・ファイルへアクセスする要求が許可されているか判別します。

ユーザー関連セキュリティー要素

ユーザー関連セキュリティー要素には、サーバー・システムの SECURELOC パラメーター、クライアント・システムによって送信されるユーザー ID (許可される場合)、クライアント・システムによって送信さ

れるユーザー ID のパスワード、およびユーザー・プロファイルかサーバー・システムでのデフォルト・ユーザー・プロファイルがあります。

オブジェクト関連セキュリティー要素

オブジェクト関連セキュリティー要素には、DDMACC パラメーター、および任意で、通常のオブジェクト権限制御を補足するための、ユーザー指定のユーザー出口プログラムがあります。

ユーザー関連サーバー・セキュリティーの要素

分散リレーショナル・データベースの作業を処理するには、有効なユーザー・プロファイルがサーバー上に存在する必要があります。IBM i オペレーティング・システム上で通信ジョブを処理するサブシステムに、デフォルトのユーザー・プロファイルを指定することができます。

サーバー上で通信項目追加 (ADDCMNE) コマンドの DFTUSR パラメーターに、デフォルトのユーザー・プロファイル名を指定できます。ADDCMNE コマンドは、通信ジョブで使用されるサブシステム記述に通信項目を追加します。

デフォルトのユーザー・プロファイルが通信サブシステムで指定された場合、サーバーが保護ロケーションであるかどうかによって、この要求にデフォルトのユーザー・プロファイルを使用するかどうかが決まります。装置記述の作成 (APPC) (CRTDEVAPPC) コマンド上の SECURELOC パラメーター、または APPN リモート・ロケーション・リスト上での保護ロケーションの指定によって、サーバーが保護ロケーションかどうかを指定します。

- SECURELOC またはサーバー上の保護ロケーションで *YES が指定されると、サーバーはクライアントが保護ロケーションであると見なします。クライアントからの要求とともにユーザー ID および検査済み標識が要求されます。クライアントによって送信されるユーザー ID と一致するユーザー・プロファイルがサーバー上に存在する場合、要求は許可されます。そうでない場合には、要求は拒否されます。
- サーバー上の SECURELOC パラメーターで *NO が指定されると、サーバーはクライアントを保護ロケーションであると見なしません。クライアントはそれでもユーザー ID を送信しますが、サーバーはそれをこの要求で使用することはありません。その代わりに、この要求ではサーバー上のデフォルトのユーザー・プロファイルが使用されます (存在する場合)。サーバー上でデフォルトのユーザー・プロファイルが存在しない場合には、要求は拒否されます。
- サーバー上の SECURELOC で *VFYENCPWD が指定されると、サーバーはクライアントを保護ロケーションであると見なしますが、現行のユーザーが正当なユーザーかどうかを検査するため、ユーザー ID とパスワードを (暗号化された形式で) 送信することが必要になります。クライアントによって送信されるユーザー ID と一致するユーザー・プロファイルがサーバー上で存在していて、そのクライアントが両方のシステムで同一のパスワードを持っている場合、要求は許可されます。そうでない場合には、要求は拒否されます。

次の表に、IBM i オペレーティング・システム上の SNA SECURITY(PGM) を制御する要素の可能な組み合わせすべてが示されています。列内の Y は、その要素が存在するか条件が満たされていることを示します。PWD 列の M は、セキュリティー管理機能がユーザーのパスワードを検索し、パスワード保護が活動状態ならば、保護 (暗号化された) パスワードが送信されることを示しています。保護パスワードが送信されない場合には、パスワードは送信されません。保護パスワードとは、会話が開始される際に APPC がユーザー・パスワードに置き換える文字ストリングのことです。保護パスワードはシステムの両方のパートナーがパスワード保護をサポートしていて、パスワードが IBM i V5R3 以降または OS/400 V2R2 以降を実行するシステムで作成されている場合にのみ使用できます。

表4. 分散リレーショナル・データベースへのリモート・アクセス

行	UID	PWD ¹	AVI	SEC(Y)	DFT	有効	アクセス
1	Y	Y		Y	Y	Y	UID を使用
2	Y	Y		Y	Y		拒否
3	Y	Y		Y		Y	UID を使用
4	Y	Y		Y			拒否
5	Y	Y			Y	Y	UID を使用
6	Y	Y			Y		拒否
7	Y	Y				Y	UID を使用
8	Y	Y					拒否
9	Y		Y	Y	Y	Y	UID を使用
10	Y		Y	Y	Y		拒否
11	Y		Y	Y		Y	UID を使用
12	Y		Y	Y			拒否
13	Y	M ³			Y	Y	DFT または UID ² を使用
14	Y	M ³			Y		DFT または UID ² を使用
15	Y	M ³				Y	拒否または UID ² を使用
16	Y	M ³					拒否または UID ² を使用
17				Y	Y		DFT を使用
18				Y			拒否
19					Y		DFT を使用
20							拒否

キー:

UID ユーザー ID の送信

PWD パスワードの送信

AVI 検査済み標識の設定

SEC(Y) SECURELOC(YES) の指定

DFT 通信サブシステムでのデフォルトのユーザー ID の指定

有効 ユーザー ID とパスワードが有効

UID を使用

提供されたユーザー ID を使用して接続する

DFT を使用

デフォルトのユーザー ID を使用して接続する

拒否 接続が行われなかった

1. パスワード保護が活動状態であれば、保護パスワードが送信される。
2. パスワード保護が活動状態であれば、「UID を使用」となる。
3. パスワード保護が活動状態であれば、セキュリティー管理機能がユーザーのパスワードを検索し、保護パスワードを送信する。活動状態でなければ、パスワードは送信されない。

デフォルトのユーザー・プロファイルを使用しないようにするには、分散リレーショナル・データベース・オブジェクトにアクセスする必要があるクライアント・ユーザーごとに、サーバー上にユーザー・プロファイルを作成してください。デフォルトのユーザー・プロファイルを使用することに決めた場合には、適切な権限のないユーザーにシステムの使用許可を与えないでください。たとえば、次のコマンドはデフォルトのユーザー・パラメーターを DFTUSER(QUSER) として指定します。これにより、システムは通信要求からユーザー ID またはパスワードを受け取らなくてもジョブ開始要求を受諾することができます。通信ジョブは、QUSER ユーザー・プロファイルを使用してサインオンされます。

```
ADDCMNE SBS(D(SAMPLE) DEV(*ALL) DFTUSER(QUSER)
```

TCP/IP ネットワークでのセキュリティーの要素

ネイティブ TCP/IP 上の DDM および DRDA は、IBM i 通信セキュリティー・サービスや通信装置、モード、保護ロケーション属性などの概念を使用しません。また、拡張プログラム間通信 (APPC) と関連した会話セキュリティー・レベルも使用しません。したがって、TCP/IP のセキュリティー設定はかなり異なったものです。

TCP/IP ネットワークでのクライアントのセキュリティー:

接続性のシナリオが異なれば、異なるレベルの認証を使用することが求められます。したがって、管理者は、サーバーへ接続するときに、より望ましい認証方式フィールドを各 RDB ディレクトリー項目に設定することにより、クライアントで必要な最低のセキュリティー認証方式を設定できます。

さらに管理者は、より低いセキュリティー認証方式を許可するよう選択することにより、認証方式についての決定をサーバーと折衝することを可能にできます。この場合、より望ましい認証方式は試行中になりますが、サーバーがより望ましい方式を受け入れられない場合、システムのセキュリティー設定や、暗号化サポートの可用性などの他の要因に応じ、より低い方式を使用できます。たとえば、2 つのシステムが物理的に保護されていない環境にある場合、管理者は、より低いセキュリティー認証方式を許可せずに Kerberos 認証を必須とするよう選択できます。

- | 暗号化サーバー認証方式を RDB ディレクトリー項目で使用する場合は、オプションで、ADDRDBDIRE
- | コマンドまたは CHGRDBDIRE コマンドで *DES または *AES 暗号化アルゴリズムを指定することがで
- | きます。 ENCALG(*DES) または ENCALG(*AES) は、ユーザー ID またはパスワードの暗号化のために
- | クライアントから要求される暗号化アルゴリズムを決定します。DDM ファイルでも、CRTDDMF コマン
- | ドまたは CHGDDMF コマンドの RMTLOCNAME キーワードに *RDB 値、RDB キーワードに RDB を
- | 指定して DDM ファイルを作成することで、*AES 暗号化を使用することができます。すべてのデータを
- | 暗号化する場合には、SSL を使用すると、ネットワークを介するすべてのデータが暗号化され、さまざま
- | な暗号化タイプがサポートされます。

- | 以下の設定は、暗号化認証方式のためにクライアントから要求される暗号化アルゴリズムを設定するのに使
- | 用できます。
- | • ENCALG(*AES)
- | Advanced Encryption Standard (AES) 暗号化アルゴリズムが要求されます。
- | • ENCALG(*DES)
- | Data Encryption Standard (DES) 暗号化アルゴリズムが要求されます。これはデフォルトです。

クライアント側では、DRDA TCP/IP 接続要求のユーザー ID とともにパスワードを送信するのに 2 つの方法のいずれかを使用できます。これらの方法のどちらも使用しない場合には、CONNECT コマンドはユーザー ID を送信できるだけです。

パスワードを送信する最初の方法は、対話式 SQL 環境からの以下の例のように、SQL CONNECT ステートメントの USER/USING 形式を使用することです。

```
CONNECT TO rdbname USER userid USING 'password'
```

組み込み SQL を使用したプログラムでは、ユーザー ID とパスワードの値は USER/USING データベースのホスト変数に含められます。

CLI を使用するプログラムで、DRDA クライアントに対し、ユーザー ID とパスワードをホスト変数で表す方法は、次の例で示します。

```
SQLConnect(hdbc,sysname,SQL_NTS, /*do the connect to the server */  
uid,SQL_NTS,pwd,SQL_NTS);
```

1 TCP/IP 経由の接続要求を介して送信するユーザー ID とパスワードを定義する 2 つ目の方法は、サーバ
1 ー認証項目です。サーバ認証リストは、システム上のすべてのユーザー・プロファイルまたはグループ・
1 プロファイルに関連付けられます。デフォルトでは、このリストは空です。しかし、サーバ権限項目の追
1 加 (ADDSVRAUTE) コマンドを使用して項目を追加できます。

権限の要件: ADDSVRAUTE コマンドを使用するには、以下の権限が必要です。

- 1 • セキュリティー管理者 (*SECADM) ユーザー特殊権限
- 1 • オブジェクト管理 (*OBJMGT) ユーザー特殊権限
- 1 • サーバ認証項目が追加される、またはサーバ認証のサインオンに使用される、ユーザー・プロフ
1 ールに対する使用 (*USE) 権限

サーバ許可項目の追加 (ADDSVRAUTE) コマンドを使用してパスワードを保管するには、QRETSVRSEC システム値を '1' に設定する必要があります。デフォルトでは、この値は '0' です。この値を変更するには、次のコマンドを入力してください。

```
CHGSYSVAL QRETSVRSEC VALUE('1')
```

次の例は、RDB ディレクトリー項目を使用するときの、サーバ許可項目の追加 (ADDSVRAUTE) コマンドの構文を示します。

```
ADDSVRAUTE USRPRF(user-profile) SERVER(rdbname) USRID(userid) PASSWORD(password)
```

1 USRPRF パラメーターは、クライアント・ジョブが実行されるユーザー・プロファイルを指定します。通
1 常、SERVER パラメーターには、RDB DDM ファイルによる接続または DRDA 接続のために接続する
1 RDB または QDDMDRDASERVER の名前を指定します。ただし、RDB ディレクトリーを使用するように
1 作成されていない非 RDB DDM ファイルを使用する場合は、SERVER パラメーターに
1 QDDMDRDASERVER または QDDMSERVER を指定する必要があります。SERVER パラメーターに指定
1 する値は、大文字でなければなりません。USRID パラメーターは、サーバ・ジョブが実行されるユー
1 ー・プロファイルを指定します。PASSWORD パラメーターは、ユーザー・プロファイルのパスワードを
1 指定します。

USRPRF パラメーターを省略すると、サーバ許可項目の追加 (ADDSVRAUTE) コマンドが実行されるところのユーザー・プロファイルがデフォルト値として使われます。USRID パラメーターを省略すると、USRPRF パラメーターの値がデフォルト値として使われます。PASSWORD パラメーターを省略したり、QRETSVRSEC 値を 0 に設定すると、項目にパスワードは保管されません。そして、その項目を使用した接続が試みられると、試行されるセキュリティー機構はユーザー ID だけになります。

サーバー認証リストにどの認証項目が追加されているかは、**サーバー認証項目の表示 (DSPSVRAUTE)** コマンドを使用して確認できます。ユーザー作成プログラムの **サーバー認証項目の取得 (QsyRetrieveServerEntries) (QSYRTVSE)** API も使用できます。

サーバー認証項目は、**サーバー権限項目の除去 (RMVSVRAUTE)** コマンドを使用して除去できます。サーバー認証項目は、**サーバー権限項目の変更 (CHGSVRAUTE)** コマンドを使用して変更できます。

リレーショナル・データベース (RDB) でサーバー認証項目が存在していて、接続要求でユーザー ID とパスワードが渡される場合には、渡されるユーザー ID とパスワードがサーバーの認証項目より優先されます。

DRDA と DDM でサポートされる、**サーバー認証項目追加 (ADDSVRAUTE)** コマンドの SERVER パラメーターの 2 つの特殊値は、以下のとおりです。

- **サーバー認証項目追加 (ADDSVRAUTE)** コマンドの SERVER パラメーターの QDDMDRDASERVER 特殊値を DDM および DRDA 接続に使用すると、管理者は、TCP/IP ネットワーク上のすべてのシステムに対して可能なすべての DDM または DRDA 接続を、共通のユーザー ID とパスワードを介してユーザーが操作できるように構成できます。この特殊値を特定のユーザーに構成すると、リレーショナル・データベース・ディレクトリーにシステムが追加されても、そのユーザーに対する追加の変更は必要ありません。ただし、以前と同様に、ユーザーがサーバー認証項目または CONNECT ステートメントに有効なユーザー ID とパスワードを指定しない限り、そのユーザーが DRDA/DDM 経由で接続することはできません。この特殊値は、QDDMSERVER と同じサポートをすべて提供するほか、RDB DDM ファイルによる接続と DRDA 接続もサポートします。

- **サーバー認証項目追加 (ADDSVRAUTE)** コマンドの SERVER パラメーターの QDDMSERVER 特殊値を非 RDB DDM 接続に使用すると、管理者は、TCP/IP ネットワーク上のすべてのシステムに対して可能なすべての非 RDB DDM 接続を、共通のユーザー ID とパスワードを介してユーザーが操作できるように構成できます。この特殊値を特定のユーザーに構成すると、新しい非 RDB DDM ファイルが作成されて使用されても、そのユーザーに対する追加の変更は必要ありません。QDDMSERVER の使用の欠点は、この値は、DDM ファイルを使用した接続や DRDA 接続をサポートしないことです。DDM ファイル、RDB DDM ファイル、および DRDA 接続で、コマンド・ユーザーのユーザー ID とパスワードを必要とするサーバー認証項目を利用する場合は、QDDMDRDASERVER を使用することを推奨します。

ユーザー ID とパスワードを指定せずに TCP/IP 経由で DRDA 接続を試行すると、DB2 for i クライアント (AR) は、クライアント・ジョブが実行されているユーザー・プロファイルのサーバー認証リストを検査します。CONNECT ステートメントの RDB 名または QDDMDRDASERVER と認証項目の SERVER 名 (どちらも大文字でなければなりません) が一致することが分かれば、項目内の関連する USRID パラメーターが接続ユーザー ID として使用されます。PASSWORD パラメーターが項目内で保管されている場合には、そのパスワードも接続要求で送信されます。

サーバー認証項目は、DDM ファイル入出力操作のために、TCP/IP 経由でパスワードを送信するときにも使用できます。TCP/IP 上で DDM 接続を試みる際には、DB2 for i は、クライアント・ジョブが実行されているところのユーザー・プロファイルを調べるために、サーバー認証リストを検査します。RDB 名 (RDB DDM ファイルが使用された場合) または QDDMSERVER (非 RDB DDM が使用された場合) のいずれかと、認証項目の SERVER 名 が一致することが検出された場合、項目内の関連する USRID パラメーターが接続ユーザー ID として使用されます。PASSWORD パラメーターが項目内で保管されている場合には、そのパスワードも接続要求で送信されます。

| DRDA 接続要求の場合、システム名を指定するサーバー認証項目が存在しており、CONNECT ステートメントでユーザー ID とパスワードが渡されていない場合は、サーバー認証項目に関連付けられたユーザー ID とパスワードが、QDDMDRDASERVER のサーバー認証項目よりも優先されます。

| DRDA 接続要求の場合、システム名を指定するサーバー認証項目が存在しており、CONNECT ステートメントでユーザー ID とパスワードが渡されている場合は、CONNECT ステートメントに関連付けられたユーザー ID とパスワードが、いずれのサーバー認証項目よりも優先されます。

| RDB DDM ファイル接続要求の場合、システム名を指定するサーバー認証項目が、QDDMDRDASERVER のサーバー認証項目よりも優先されます。

| 非 RDB DDM ファイル接続要求の場合、サーバー認証項目 QDDMSERVER が、QDDMDRDASERVER のサーバー認証項目よりも優先されます。

| 例 1: DRDA 接続に QDDMDRDASERVER を使用する

| 環境: 3 つのシステム (SYSA、SYSB、SYSC)

| SYSA はアプリケーション・リクエスター (AR) です。SYSB と SYSC は、アプリケーション・サーバー (AS) です。

| SYSA で、以下を指定します。

```
| ADDSVRAUTE USRPRF(YOURPRF) SERVER(QDDMDRDASERVER) USRID(youruid) PASSWORD(yourpwd)
| STRSQL
| CONNECT TO SYSB
| CONNECT TO SYSC
```

| この時点では、サーバー認証項目の特殊値 QDDMDRDASERVER で指定したユーザー ID 「youruid」とパスワード「yourpwd」を使用した、2 つの DRDA 接続が確立されます。

| 例 2: DRDA 接続に QDDMDRDASERVER ではなくシステム名を使用する

| 環境: 3 つのシステム (SYSA、SYSB、SYSC)

| SYSA はアプリケーション・リクエスター (AR) です。SYSB と SYSC は、アプリケーション・サーバー (AS) です。

| SYSA で、以下を指定します。

```
| ADDSVRAUTE USRPRF(YOURPRF) SERVER(QDDMDRDASERVER) USRID(youruid) PASSWORD(yourpwd)
| ADDSVRAUTE USRPRF(YOURPRF) SERVER(SYSB) USRID(yourotheruid) PASSWORD(yourotherpwd)
| ADDSVRAUTE USRPRF(YOURPRF) SERVER(SYSC) USRID(yourotheruid) PASSWORD(yourotherpwd)
| STRSQL
| CONNECT TO SYSB
| CONNECT TO SYSC
```

| この時点では、ユーザー ID 「yourotheruid」とパスワード「yourotherpwd」を使用した 2 つの DRDA 接続が確立されます。これは、システム名を指定するサーバー認証項目が、QDDMDRDASERVER を指定するサーバー認証項目よりも優先されるためです。

| 例 3: DRDA 接続にシステム名と QDDMDRDASERVER を混用する

| 環境: 3 つのシステム (SYSA、SYSB、SYSC)

| SYSA はアプリケーション・リクエスター (AR) です。SYSB と SYSC は、アプリケーション・サーバー (AS) です。

| SYSA で、以下を指定します。

```
| ADDSVRAUTE USRPRF(YOURPRF) SERVER(QDDMDRDASERVER) USRID(youruid) PASSWORD(yourpwd)
| ADDSVRAUTE USRPRF(YOURPRF) SERVER(SYSB) USRID(yourotheruid) PASSWORD(yourotherpwd)
| STRSQL
| CONNECT TO SYSB user testuserid using 'testpassword'
| CONNECT TO SYSC
```

| この時点では、2 つの DRDA 接続が確立されます。SYSB への接続は、ユーザー ID 「testuserid」とパスワード「testpassword」を使用して行われます。これは、CONNECT ステートメントに指定したユーザー ID とパスワードが、サーバー認証項目よりも優先されるためです。SYSC への接続には、システム名を指定するサーバー認証項目が他に存在しない場合に、QDDMDRDASERVER 認証項目が使用されるため、SYSC への接続は、ユーザー ID 「youruid」とパスワード「yourpwd」を使用して行われます。

| 例 4: RDB DDM 接続と非 RDB DDM 接続に QDDMDRDASERVER と QDDMSERVER を混用する

| 環境: 3 つのシステム (SYSA、SYSB、SYSC)

| SYSA はアプリケーション・リクエスター (AR) です。SYSB と SYSC は、アプリケーション・サーバー (AS) です。

| SYSA で、以下を指定します。

```
| ADDSVRAUTE USRPRF(YOURPRF) SERVER(QDDMDRDASERVER) USRID(youruid) PASSWORD(yourpwd)
| ADDSVRAUTE USRPRF(YOURPRF) SERVER(QDDMSERVER) USRID(youruid2) PASSWORD(yourpwd2)
| ADDSVRAUTE USRPRF(YOURPRF) SERVER(SYSC) USRID(yourotheruid) PASSWORD(yourotherpwd)
| CRTDDMF FILE(QTEMP/DDMF) RMTFILE(FILE) RMTLOCNAME(SYSB *IP)
| CRTDDMF FILE(QTEMP/DDMF2) RMTFILE(FILE) RMTLOCNAME(*RDB) RDB(SYSB)
| CRTDDMF FILE(QTEMP/DDMF3) RMTFILE(FILE) RMTLOCNAME(*RDB) RDB(SYSC)
|
| SBMRMTCMD CMD('DSPLIB YOURLIB') DDMFILE(QTEMP/DDMF)
```

| SYSB への接続は、ユーザー ID 「youruid2」とパスワード「yourpwd2」を使用して行われます。これは、非 RDB DDM ファイルが、接続時にユーザー ID とパスワードに QDDMSERVER を使用するためです。QDDMSERVER が存在しない場合は、QDDMDRDASERVER が使用されます。

```
| SBMRMTCMD CMD('DSPLIB YOURLIB') DDMFILE(QTEMP/DDMF2)
```

| SYSB への接続は、ユーザー ID 「youruid」とパスワード「yourpwd」を使用して行われます。これは、システム名を指定するサーバー認証項目が存在しない場合に、RDB DDM ファイルが接続時にユーザー ID とパスワードに QDDMDRDASERVER を使用するためです。

```
| SBMRMTCMD CMD('DSPLIB YOURLIB') DDMFILE(QTEMP/DDMF3)
```

| SYSC への接続は、ユーザー ID 「yourotheruid」とパスワード「yourotherpwd」を使用して行われます。これは、システム名を指定するサーバー認証項目が存在する場合に、RDB DDM ファイルが接続時にユーザー ID とパスワードに QDDMDRDASERVER を使用しないためです。

| グループ・プロファイルの使用による DDM および DRDA 認証項目の簡素化された管理

| DDM と DRDA では、グループ・プロファイル名または補足グループ・プロファイル名でサーバー認証項目に定義されている共通のユーザー ID とパスワードを利用することができるようになりました。グループ・プロファイル名または補足グループ・プロファイル名は、サーバー認証項目追加 (ADDSVRAUTE) コマンドの USRPRF パラメーターで指定します。この機能強化の前では、ユーザー ID とパスワードの認証を必要とするアプリケーション・サーバーへのアクセスにサーバー認証項目を使用する場合に、アクセスを必要とするシステム上のすべてのユーザーに対してサーバー認証項目が必要でした。

| サーバー認証項目のシステム管理は、グループ・プロファイルを使用することで、ずっとシンプルになります。グループ・プロファイル・ベースの DDM または DRDA 接続を確立するときは、共通のユーザー ID とパスワードのプロファイルを使用します。リモート接続の機能は、同じグループに属するユーザーを、これらのユーザーのグループ・プロファイルの権限を使用して制御することで管理されます。

| **例:**

```
| CRTUSRPRF USRPRF(TESTTEAM) PASSWORD(*NONE)
| CRTUSRPRF USRPRF(JIM) PASSWORD(yourpasswordA) GRPPRF(TESTTEAM)
| ADDSVRAUTE USRPRF(TESTTEAM) SERVER(QDDMDRDASERVER or <RDB-name>)
|   USRID(youruseridB) PASSWORD(yourpasswordB)
|
| Signon to <system-A> with Jim's user profile
| STRSQL
| CONNECT TO <system-B>
| (where <system-B> is defined in the Relational Database Directory Entries - WRKRDBDIRE)
|
| A connection is attempted to <system-B> with userid = youruseridB and password = yourpasswordB.
```

| **注:** 以下の進行の順序が示すとおり、ユーザー ID 認証項目は、グループ・プロファイル認証項目よりも優先されます。DDM および DRDA サーバー認証項目の検査の進行は、以下の順序で実行されます。接続は、最初に一致した認証項目の設定を使用して試行されます。

- | 1. USRPRF = ユーザー・プロファイル、SERVER = アプリケーション・サーバー名の認証項目を検索します。
- | 2. USRPRF = ユーザー・プロファイル、SERVER = 「QDDMDRDASERVER」の認証項目を検索します。
- | 3. USRPRF = グループ・プロファイル、SERVER = アプリケーション・サーバー名の認証項目を検索します。
- | 4. USRPRF = グループ・プロファイル、SERVER = 「QDDMDRDASERVER」の認証項目を検索します。
- | 5. USRPRF = 補足グループ・プロファイル、SERVER = アプリケーション・サーバー名の認証項目を検索します。
- | 6. USRPRF = 補足グループ・プロファイル、SERVER = 「QDDMDRDASERVER」の認証項目を検索します。
- | 7. 上記のすべてのステップで項目が見つからなかった場合は、USERID のみの認証が試行されます。

| **ユーザー・プロファイルのパスワードの使用による DRDA および DDM 認証の有効化**

| この機能拡張は、正式には DRDA/DDM 結合相互認証と呼ばれ、IBM i から IBM i への接続でのみサポートされ、環境変数 QIBM_CONJOINED_MUT_AUTH によって有効になります。

| この機能拡張の前は、分散リレーショナル・データベース体系 (DRDA) または分散データ管理 (DDM) 接続を試行するユーザーは、アプリケーション・サーバーがパスワードを必要とするよう構成されている場合には、SQL CONNECT ステートメント上でパスワードを指定するか、パスワードを含むサーバー認証項目を指定する必要がありました。これにより、ユーザーまたはシステム管理者は、すべてのターゲット・システムのサーバー認証項目を管理するために時間を費やしたり、リモートで接続するときに CONNECT 時間にパスワードを指定したりする必要がありました。SQL CONNECT ステートメントまたはアプリケーション・サーバーのサーバー認証項目でパスワードを指定しないと、パスワードを必要とするアプリケーション・サーバーへの DRDA/DDM 接続は失敗しました。

| この機能拡張を使用すると、アプリケーション・サーバーに対する IBM i から IBM i DRDA への接続または DDM 接続は、現在サインイン済みのユーザー・プロファイルのユーザー ID とパスワードを使用して試行されます。この接続は、環境変数の値が Y に設定されており、パスワードを必要とするよう構成されたアプリケーション・サーバーに対して、SQL CONNECT ステートメントまたはサーバー認証項目でユーザー ID とパスワードを明示的に指定していない場合のみ試行されます。

この機能拡張は、複数のシステム間で同じ組み合わせのユーザー ID とパスワードを使用しているネットワークで有益です。この機能拡張を使用すると、3 部構成の接続にサーバー認証項目を指定する必要はありません。

要件:

アプリケーション・リクエスターとアプリケーション・サーバーは、いずれも v7r2m0 の IBM i でないとサポートされません。この機能拡張は、アプリケーション・リクエスターのジョブによって値が Y の環境変数 QIBM_CONJOINED_MUT_AUTH を追加すると有効になります (環境変数を追加するには、環境変数の追加 (ADDENVVAR) コマンドを参照してください)。アプリケーション・サーバーは、パスワードを必要とするように設定する必要があります (CHGDDMTCPA コマンドの PWDRQD キーワードを参照してください)。いずれのシステムも、接続するには、それぞれのために同じユーザー ID とパスワードが定義されている必要があります。また、いずれのシステムも、システム値 QPWDLVL に同じパスワード・レベルが指定されている必要があります。ユーザーは、パスワードを必要とするよう構成されたアプリケーション・サーバーに対して、SQL CONNECT ステートメントまたはサーバー認証項目にユーザー ID とパスワードを明示的に指定してはなりません。

注: アプリケーション・リクエスターによってパスワードまたはサーバー認証項目が指定されておらず、サポートが有効な IBM i から IBM i の DRDA または DDM 接続の場合は、接続にパスワードを必要とするアプリケーション・サーバーに対して追加の接続が試行されます。この追加の接続は、暗号化ユーザー ID およびパスワード・セキュリティ・メカニズム (ユーザー ID は接続を行っているユーザー・プロファイルから取得され、パスワードはそのユーザー・プロファイルから取得される) を使用して試行されます。パスワードが一致すると判断されたら、接続は通常どおり許可されます。パスワードが一致しない場合は、セキュリティの障害があるか上記の要件が満たされておらず、CPF22E2 メッセージがシグナル通知され、PW 監査レコードが書き込まれます。この無効な接続の試行は、ユーザー・プロファイルの 1 回の無効なサインオンの試行としてカウントされます。

監査が有効で、監査レベルが許可の失敗を記録するように構成されている場合は、セキュリティ監査レコードが QSYS/QAUDJRN セキュリティ監査実行記録に書き込まれます。PW 監査レコードは、ユーザー ID とパスワードが接続に失敗した場合に表示されます。

Kerberos ソース構成

分散リレーショナル・データベース体系 (DRDA) と分散データ管理 (DDM) では、それぞれのシステムで Kerberos が構成されている場合は、Kerberos 認証を利用できます。

ジョブのユーザー・プロファイルに有効なチケット許可チケット (TGT) がある場合、DRDA クライアントはこの TGT を使用して、サービス・チケットを生成し、ユーザーをリモート・システムに認証します。有効な TGT を使用すると、パスワードを直接入力する必要はなくなるため、サーバー認証項目を行う必要性はなくなります。しかし、ジョブのユーザー・プロファイルに有効な TGT がない場合、サーバー認証項目からユーザー ID とパスワードを取り出して、必要な TGT およびサービス・チケットを生成できます。

Kerberos を使用する場合、リモート・ホスト名として、RDB ディレクトリー項目のリモート・ロケーション (RMTLOCNAME) を入力する必要があります。Kerberos 認証では、IP アドレスは役立ちません。

Kerberos レルム名が DNS 接尾部名と異なる場合には、正確なレルムにマップする必要があります。これを行うには、Kerberos 構成ファイル (krb5.conf) 内に、各リモート・ホスト名を正しいレルム名にマップする項目がなければなりません。この入力されるホスト名は、リモート・ロケーション名 (RMTLOCNAME) と正確に一致している必要があります。DSPRDBDIRE または DSPDDMF コマンドで表示されるリモート・ロケーション・パラメーターは、krb5.conf ファイルのドメイン名と一致していなければなりません。次の図に、DSPRDBDIRE 画面の一例を示します。

リレーショナル・データベース項目の詳細の表示

```
リレーショナル・データベース . . . . . : RCHASXXX
リモート・ロケーション:
  リモート・ロケーション . . . . . : rchasxxx.rchland.ibm.com
  タイプ . . . . . : *IP
  ポート番号またはサービス名 . . . . . : *DRDA
リモート認証方式 :
  所望の方式 . . . . . : *KERBEROS
  より低い認証の許可 . . . . . : *NOALLOWER
  保護接続 . . . . . : *NONE
  暗号化アルゴリズム . . . . . : *DES
テキスト . . . . . :

リレーショナル・データベース・タイプ . . . : *REMOTE
```

続行するには、実行キーを押してください。
F3= 終了 F12= 取り消し

次に示すのは、対応する `krb5.conf` ファイルの内容の一部です。ここでは、リモート・ロケーション名と一致するドメイン名が示されています (注: 構成ファイルの内容を表示するために、**ファイルの表示 (DSPF) コマンド**が使用されています)。

```
DSPF STMF('/QIBM/UserData/OS400/NetworkAuthentication/krb5.conf')
```

```
[domain_realm]
; Convert host names to realm names. Individual host names may be
; specified. Domain suffixes may be specified with a leading period
; and will apply to all host names ending in that suffix.
rchasxxx.rchland.ibm.com = REALM.RCHLAND.IBM.COM
```

Kerberos を使用するジョブは、`krb5.conf` ファイルに対して構成の変更が行われる場合に、再始動する必要があります。

関連概念:

エンタープライズ識別マッピング (EIM)

関連タスク:

73 ページの『TCP/IP サーバーのセットアップ』

TCP/IP プロトコルを使用する Distributed Relational Database Architecture (DRDA) サーバーを所有する場合、DDM TCP/IP サーバーをセットアップする必要があります。

ネットワーク認証サービスの構成

関連資料:

サーバー認証項目追加 (ADDSVRAUTE) コマンド

サーバー認証項目表示 (DSPSVRAUTE) コマンド

サーバー認証項目取得 (QsyRetrieveServerEntries) (QSYRTVSE) API

サーバー認証項目除去 (RMVSVRAUTE) コマンド

サーバー認証項目変更 (CHGSVRAUTE) コマンド

ファイル表示 (DSPF) コマンド

TCP/IP ネットワークでのサーバーのセキュリティー:

TCP/IP サーバーには、デフォルトのセキュリティーとして、クリア・テキストのパスワードを指定したユーザー ID が備えられています。これは、サーバーがインストールされると、着信 TCP/IP 接続要求ではサーバー・ジョブが実行されるユーザー ID とともに、最低でもクリア・テキストのパスワードが必要になることを表しています。

セキュリティーは、**DDM TCP/IP 属性の変更 (CHGDDMTCPA)** コマンドか、System i® ナビゲーター の「ネットワーク」 > 「サーバー」 > **TCP/IP** > 「**DDM サーバー (DDM server)** プロパティ」で変更できます。この設定を変更するには、*IOSYSCFG 特殊権限が必要です。

DDM TCP/IP属性変更 (CHGDDMTCPA)

選択項目を入力して、実行キーを押してください。

サーバーの自動開始	*YES	*SAME, *NO, *YES
最低認証方式	*VLDONLY	*SAME, *USRID, *VLDONLY...
最も低い暗号化アルゴリズム . . .	*DES	*SAME, *DES, *AES

終わり

F3= 終了 F4=プロンプト F5= 最新表示 F12= 取り消し
F13= この画面の使用法 F24= キーの続き

以下の設定を、サーバーで許可される最低認証方式の設定に使用できます。

- PWDRQD (*USRID)

パスワードは不要です。

- PWDRQD(*VLDONLY)

パスワードは必須ではないが、送信される場合には有効でなければならない。

*USRID と *VLDONLY の違う点を挙げると、パスワードがクライアント・システムから送信される場合に、*USRID オプションでは無視されます。それに対して *VLDONLY オプションでは、パスワードが送信されると、そのパスワードは付随するユーザー ID の妥当性が検査されて、間違っている場合にはアクセスが拒否されます。

より高いセキュリティー・レベルでは、暗号化されたユーザー ID とパスワード (つまり PWDRQD(*ENCUSRPWD)), 暗号化された必須のパスワード (つまり PWDRQD(*USRENCPWD)), および Kerberos (つまり PWDRQD(*KERBEROS)) を使用できます。Kerberos を使用する場合、エンタープライズ識別マッピング (EIM) を使用して、ユーザー・プロファイル Kerberos プリンシパルにマッピングする必要があります。

次の例では、**DDM TCP/IP 属性の変更 (CHGDDMTCPA)** コマンドを使用して、ユーザー ID に暗号化されたパスワードが付随しなければならないことを指定します。このオプションを設定するには、次のように入力します。

CHGDDMTCPA PWDRQD(*USRENCPWD)

以下の設定を、サーバーで許可される暗号化認証方式に関する最も低い暗号化アルゴリズムの設定に使用できます。

- ENCALG(*AES)

Advanced Encryption Standard (AES) 暗号化アルゴリズムのみ許可されます。

- ENCALG(*DES)

Data Encryption Standard (DES) 暗号化アルゴリズムまたはさらに強力な暗号化アルゴリズムが許可されます。

次の例では、**DDM TCP/IP 属性変更 (CHGDDMTCPA)** コマンドを使用して、暗号化認証方式を使用するサーバーに対する認証に AES 暗号化を使用しなければならないことを指定します。このオプションを設定するには、次のように入力します。

CHGDDMTCPA ENCALG(*AES)

ENCALG(*DES) または ENCALG(*AES) は、サーバーでサポートされる、ユーザー ID とパスワードの暗号化に関する最も低い暗号化アルゴリズム・レベルを判別します。すべてのデータを暗号化する場合には、SSL を使用すると、ネットワークを介するすべてのデータが暗号化され、さまざまな暗号化タイプがサポートされます。

注: DDM および DRDA TCP/IP サーバーは、以下の形式のパスワード暗号化をサポートします。

- パスワード置換アルゴリズム
- Diffie-Hellman 公開鍵アルゴリズム (56 ビット *DES または 256 ビット *AES)
- 強力なパスワード置換アルゴリズム

クライアントとサーバー間では使用されるセキュリティー・メカニズムの折衝が行われます。3 つあるどの暗号化方式を使用しても暗号化パスワード (PWDRQD(*USRENCPWD)) の要件が満たされ、Secure Socket Layer (SSL) のデータ・ストリームの使用にも十分です。

関連概念:

エンタープライズ識別マッピング

関連資料:

DDM TCP/IP 属性変更 (CHGDDMTCPA) コマンド

接続セキュリティー・プロトコル:

TCP/IP を使用した分散データ管理 (DDM) または Distributed Relational Database Architecture (DRDA) の現在の DB2 for i 実装によって、以下のいくつかの接続セキュリティー・プロトコルがサポートされています。

- ユーザー ID のみ
- クリア・テキスト・パスワード付きのユーザー ID
- 暗号化されたパスワード付きのユーザー ID
- 暗号化されたパスワード付きの暗号化ユーザー ID
- Kerberos

暗号化されたデータ・ストリーム・サポートでは、通常の通信トレース・サポートを使用する利点はあまりありません。 **TCP/IP アプリケーションのトレース (TRCTCPAPP)** コマンドは、暗号化の前の時点でアウトバウンド・データ・ストリームを記録し、インバウンド・データ・ストリームを暗号化解除の後で記録します。

関連概念:

415 ページの『通信トレース』

Distributed Relational Database Architecture (DRDA) を使って分散リレーショナル・データベースにアクセスしようとして、CPF3Exx の範囲または CPF91xx の範囲のメッセージが出されたら、通信トレースを実行する必要があります。

関連資料:

TCP/IP アプリケーション・トレース (TRCTCPAPP) コマンド

Secure Sockets Layer:

DB2 for i 分散リレーショナル・データベース体系 (DRDA) クライアントおよびサーバーは、Secure Sockets Layer (SSL) をサポートします。インターネット・プロトコル・セキュリティー・アーキテクチャー (IPSec) を使用して同様の機能が提供されます。

DDM TCP/IP クライアントおよびサーバーは、SSL データ暗号化プロトコルをサポートします。このプロトコルを使用して、IBM Toolbox for Java および IBM i Access Family OLE DB Provider などの、レコード・レベルのアクセス用の SSL をサポートする他のクライアントとサーバーで相互運用できます。このプロトコルを使用して、SSL をサポートする独立ソフトウェア販売会社が提供する DDM ファイル入出力クライアントも相互運用できます。

IBM i DDM TCP/IP サーバーで SSL を使用するには、クライアントを構成してサーバー上の SSL ポート 448 に接続する必要があります。

SSL を DDM TCP/IP クライアントとして使用するには、リレーショナル・データベース・ディレクトリー項目追加 (ADDRDBDIRE) コマンドまたはリレーショナル・データベース・ディレクトリー項目変更 (CHGRDBDIRE) コマンドに SECCNN(*SSL) を指定しなければなりません。

DDM TCP/IP 属性の変更 (CHGDDMTCPA) コマンドで PWDRQD(*USRENCPWD) を指定すると、SSL で有効な任意のパスワードを使用できます。これは、パスワードを含むデータ・ストリーム全体が暗号化されていることをシステムが認識するので可能です。

関連概念:

105 ページの『インターネット・プロトコル・セキュリティー・アーキテクチャー』

インターネット・プロトコル・セキュリティー・アーキテクチャー (IPSec) は、ネットワーク層のセキュリティー・プロトコルで暗号セキュリティー・サービスを提供します。これらのサービスは、インターネットまたはイントラネット上でデータの機密送達をサポートしています。

Secure Sockets Layer (SSL)

関連資料:

DDM TCP/IP 属性変更 (CHGDDMTCPA) コマンド

435 ページの『サーバーが開始しないかポート ID が有効でない』

DDM TCP/IP サーバーが開始されていない場合に出されるエラー・メッセージは CPE3425 です。

必須プログラム:

SSL サポートのセットアップとインストールが必要です。

関連概念:

IBM i Access for Windows: プログラミング

IBM i 要件:

IBM i 製品が Secure Sockets Layer (SSL) サーバー上で通信するには、システムは IBM i V5R3 以降または OS/400 V4R4 以降を実行していて、以下のアプリケーションがインストールされている必要があります。

- IBM i 用 TCP/IP 接続ユーティリティー、5761-TC1 (基本 TCP/IP サポート)。TCP/IP セットアップを参照してください。
- IBM HTTP Server for IBM i、5761-DG1 (デジタル証明書マネージャーへのアクセス用)。IBM HTTP Server for IBM i を参照してください。
- デジタル証明書マネージャー、5761-SS1 (オプション 34)。デジタル証明書マネージャーを参照してください。

IBM i 製品が Secure Sockets Layer (SSL) 上でクライアントとして通信するには、システムは IBM i V5R3 以降を実行していて、以下のアプリケーションおよび PTF がインストールされている必要があります。IBM i 7.1 以上では、この機能に PTF は必要ありません。

- IBM i 用 TCP/IP 接続ユーティリティー、5761-TC1 (基本 TCP/IP サポート)。TCP/IP セットアップを参照してください。
- IBM HTTP Server for IBM i、5761-DG1 (デジタル証明書マネージャーへのアクセス用)。IBM HTTP Server for IBM i を参照してください。
- デジタル証明書マネージャー、5761-SS1 (オプション 34)。デジタル証明書マネージャーを参照してください。
- V6R1M0 PTF: SI33570 (英語のみ)
- V5R4M0 PTF: SI33515 (英語のみ)
- V5R3M0 PTF: SI33470 (英語のみ)

インターネット・プロトコル・セキュリティ・アーキテクチャー:

インターネット・プロトコル・セキュリティ・アーキテクチャー (IPSec) は、ネットワーク層のセキュリティ・プロトコルで暗号セキュリティ・サービスを提供します。これらのサービスは、インターネットまたはイントラネット上でデータの機密送達をサポートしています。

IBM i オペレーティング・システム、IPSec では、仮想私設ネットワーク (VPN) サポートの構成要素は、アプリケーションが DRDA または DDM であるかどうかにはかかわりなく、2 つの IP アドレスまたはポート間のデータがすべて暗号化されるようにします。IPSec で使用されるアドレスおよびポートを構成することができます。IBM では DRDA アクセスまたは DDM アクセス用いずれかの IPSec にはポート 447 を使用することをお勧めしています。

IPSec とともに有効なパスワードを使用しても、普通は、DDM TCP/IP 属性の変更 (CHGDDMTCPA) コマンドの PWDRQD (*USRENCPWD) を指定する際に課せられる要件を満たせません。それは、アプリケーション (DRDA または DDM) は、IPSec が使用されているかどうかを判別することができないためです。そのため、IPSec で PWDRQD(*USRENCPWD) を使用するのを避けるべきです。

関連概念:

104 ページの『Secure Sockets Layer』

DB2 for i 分散リレーショナル・データベース体系 (DRDA) クライアントおよびサーバーは、Secure Sockets Layer (SSL) をサポートします。インターネット・プロトコル・セキュリティ・アーキテクチャー (IPSec) を使用して同様の機能が提供されます。

仮想プライベート・ネットワーク (VPN)

『特定のパスワードを平文で送信する場合の考慮事項』

IBM i オペレーティング・システムは接続パスワードの暗号化をサポートしているものの、RDB ディレクトリー項目をセットアップする際に指定できる接続セキュリティ・オプションの 1 つに、*USRIDPWD というオプションがあります。

関連資料:

DDM TCP/IP 属性変更 (CHGDDMTCPA) コマンド

特定のパスワードを平文で送信する場合の考慮事項:

IBM i オペレーティング・システムは接続パスワードの暗号化をサポートしているものの、RDB ディレクトリー項目をセットアップする際に指定できる接続セキュリティ・オプションの 1 つに、*USRIDPWD というオプションがあります。

詳細については、『リレーショナル・データベース・ディレクトリーの処理』の、リレーショナル・データベース・ディレクトリー項目追加コマンド (ADDRDBDIRE) およびリレーショナル・データベース・ディレクトリー項目変更コマンド (CHGRDBDIRE) を参照してください。

接続が行われるシステムでこの *USRIDPWD セキュリティ・オプションが許可されている場合、接続のパスワードは暗号化されずに流すことが可能です。SQL の SET ENCRYPTION PASSWORD ステートメントと ENCRYPT 関数でも、パスワードを暗号化されずにネットワーク上に流すことが可能です。現時点で、データ・ストリームの暗号化に使用できるソリューションは 2 つあります。1 つは IPSec を使用する方法です。もう 1 つは、SSL をサポートするクライアントを使用していれば、IBM i サーバー・システムとの間で伝送されるデータを暗号化するプロトコルが使用できます。

関連概念:

105 ページの『インターネット・プロトコル・セキュリティ・アーキテクチャー』

インターネット・プロトコル・セキュリティ・アーキテクチャー (IPSec) は、ネットワーク層のセキュリティ・プロトコルで暗号セキュリティ・サービスを提供します。これらのサービスは、インターネットまたはイントラネット上でデータの機密送達をサポートしています。

関連資料:

62 ページの『リレーショナル・データベース・ディレクトリーの処理』

これらの説明に従って、リレーショナル・データベース・ディレクトリーを処理してください。

SET ENCRYPTION PASSWORD ステートメント

ポートとポートの制限:

分散データ管理機能 (DDM) 通信のセキュリティのための新しい方法が提供されるようになったので、システムの管理担当者は、使用するポートをブロック化することによって特定の通信モードを制限できます。このトピックでは、これらの考慮事項の一部について説明します。

- | DDM または DRDA TCP/IP サーバーはポート 446 (予約済み DRDA ポート)、ポート 447 (予約済み
- | DDM ポート)、さらにポート 448 (予約済み SSL ポート) で listen します。DDM の DB2 for i 実装で
- | は、446 と 447 という 2 つのポートの判別がされないので、DDM および DRDA アクセスの両方は、
- | どちらのポートでも行えます。

IPSec で推奨される規則を使用すると、DDM の TCP/IP サーバーでは以下のようにポートが使用されま
す。

- 明示テキスト・データ・ストリームでは 446
- IPSec で暗号化されたデータ・ストリームでは 447 (推奨)
- SSL で暗号化されたデータ・ストリームでは 448 (必須)

1 **TCP/IP の構成 (CFGTCP)** コマンドを使用すると、サーバーでの 1 つ以上のポートの使用をブロックできま
1 す。これを行うには、そのコマンドの「TCP/IP ポート制約事項の処理」オプションを選んでください。制
1 限を追加して、QRWTLSTN が実行されているユーザー・プロファイル (通常は QUSER) 以外の特定のユ
1 ーザー・プロファイルだけが 446 などの特定のポートを使用するようにできます。これによって、446 は
1 ブロックされたこととなります。447 が IPSec 専用構成された場合、446 をブロックすると、暗号化さ
1 れたデータ・ストリームだけが、ネイティブ TCP/IP 上で DDM および DRDA アクセスで使用されるこ
1 ととなります。446 および 447 の両方をブロックして、SSL だけが使用できるように制限することもでき
1 ます。パフォーマンス上の問題または他の理由があって (SSL が使用できるクライアントが現時点では少
1 ないなど)、これらの例に従って設定を行うのは実際的ではないかもしれませんが、これらの例は、このよ
1 うな構成方法も可能であることを示すために挙げられています。

関連資料:

TCP/IP 構成 (CFGTCP) コマンド

1 ユーザー出口プログラムの使用による DRDA および DDM サーバーのアク 1 セス制御

1 拡張プログラム間通信機能 (APPC) と TCP/IP の両方で使用される Distributed Relational Database
1 Architecture (DRDA) サーバーのセキュリティー機能は、**ネットワーク属性変更 (CHGNETA)** コマンドの
1 DDMACC パラメーターの使用を DRDA に拡張します。

1 このパラメーターは、以前には DDM ファイルの入出力アクセスだけに適用されました。この機能の
1 DRDA での使用は、接続要求だけに限られているので、接続が行われた後にデータを要求するためには使
1 われません。

1 このセキュリティー機能を使用したくない場合には、通常は何も行う必要はありません。唯一の例外は、現
1 在 DDM ユーザー出口プログラムを使用していて、知られていない機能コードが受信された際に処理を拒
1 否するようにコーディングされており、そのシステムのデータをアクセスするのに DRDA を使用している
1 場合だけです。この場合には、ユーザー出口プログラムを変更して、機能コードが 'SQLCNN' である場合
1 に、'I' を戻して DRDA アクセスが行われるようにしなければなりません。

1 DRDA 接続をブロックまたはフィルター処理するためにユーザー出口プログラムを使用する場合、新しい
1 DRDA ユーザー出口プログラムを作成するか、既存のものを変更する必要があります。

1 **注:** システムが複数のデータベース (ASP グループ) で構成されている場合、ユーザー出口プログラムは
1 (範囲が 1 から 32 の補助記憶域プール上の) システム・データベース内のライブラリーに存在していなけ
1 ればなりません。

1 DRDA ユーザー出口プログラムを作成するための一般的な説明は、以下に記載します。

1 このセキュリティー機能は、DRDA 機能コードを、入力パラメーター構造でプログラムに入力できる要求
1 機能のリストに追加します。'SQLCNN' (SQL 接続要求) という名前の機能コードは、DRDA 接続要求が
1 処理されていることを示します (『例: ユーザー出口プログラムの使用による DRDA および DDM サーバ

のアクセス制御』の FUNC パラメーターを参照してください)。DRDA 接続要求呼び出しでは、APP (アプリケーション) 入力パラメーターは '*DDM' ではなく、'*DRDA' と設定されます。

DRDA のユーザー出口プログラムをコーディングする際には、パラメーター構造内の以下のフィールドが役立つかもしれません。

- USER フィールドによりプログラムは、ユーザー・プロファイル ID に基づいて DRDA アクセスを許可したり拒否したりします。
- RDBNAME フィールドには、ユーザーが接続する先の RDB の名前が示されます。これは、システム・データベースかユーザー・データベース (ASP グループ) である可能性があります。このフィールドは、複数のデータベースを構成する環境内にある、1 つ以上のデータベースへのアクセスを拒否する場合に役立ちます。
- トピック『例: ユーザー出口プログラムの使用による DRDA および DDM サーバーのアクセス制御』の SRVNAME パラメーターは、ユーザー出口プログラムの呼び出し元によって設定される場合も、されない場合もあります。このパラメーターが設定される場合、クライアント・システムの名前を表します。このパラメーターが設定されない場合には、パラメーターの値は *N になります。DRDA アプリケーション・リクエスターが IBM i 製品であるならばいつも設定されます。
- TYPDEFN パラメーターは接続しているクライアントのタイプについて追加の情報を提供します。IBM メインフレームでは、TYPEDEFN は QTDSQL370 です。IBM i 製品では、QTDSQL400 になります。Intel PC では、QTDSQLX86 です。RS/6000[®] クライアントでは、QTDSQLASC です。
- PRDID (プロダクト ID) パラメーターは、接続しようとしているプロダクトと、そのプロダクトのリリース・レベルを識別します。次に示すのは、これらのコードの最初の 3 文字の部分リストです (ユーザー出口プログラムで使用する前に IBM 以外のコードを検査する必要があります)。

QSQ IBM DB2 for i
DSN IBM DB2 for z/OS
SQL IBM DB2 for Linux、UNIX、および Windows (正式には DDCCS)
ARI IBM DB2 for VSE & VM
GTW Oracle 社のプロダクト
GVW Grandview DB/DC Systems のプロダクト
XDB XDB Systems のプロダクト
IFX Informix[®] Software のプロダクト
SIG StarQuest プロダクト
STH FileTek プロダクト
JCC IBM DB2 Universal Driver for SQLJ and JDBC

フィールドの残りは vvrrm として構造化されます。ここで、vv はバージョン、rr はリリース、および m はモディフィケーション・レベルを表します。

ユーザー出口プログラムが 0 という RTNCODE 値を戻して、接続要求が IBM i クライアントからのものである場合、ユーザーへの接続の失敗を示すメッセージは、SQ30060, ユーザーにはリレーショナル・データベース... が認可されていません になります。通常は、ユーザー出口プログラムによるアクセス拒否への応答は DRDA RDBATHRM 応答メッセージで行われます。この応答メッセージは、ユーザーがリレーショナル・データベースへの許可を持っていないことを示します。クライアントのプラットフォームが異なれば、ユーザーへのエラーも異なる場合があります。

制限:

- 機能チェックがユーザー出口プログラムで起きる場合、プログラムによって同様の応答メッセージが戻され、接続の試みは失敗します。ユーザー出口プログラムは、DB2 for i に対してコミット可能な更新を行うべきではありません。さもなければ、予期できない結果が起こる可能性があります。
- ユーザー出口プログラムを使用して、事前開始サーバー・ジョブの以前の呼び出しで開かれたファイルにアクセスしようとするべきではありません。
- V5R2 より前は、TCP/IP サーバーで使用される事前開始ジョブを後で使用するためにリサイクルする場合、さらに多くの制限がありました。次回にジョブを使用するよう準備するため、終結処理が行われます。この処理の一部として、値が *ELIGIBLE である ACTGRP パラメーターを指定した**活動化グループの再利用 (RCLACTGRP)** コマンドを使用することが関係します。その結果、事前開始サーバー・ジョブ内に残っているリンケージを、**RCLACTGRP** で破棄された活動化グループに対して使用しようとする、MCH3402 例外 (もはや存在していないすべてまたは一部のオブジェクトを参照しようとした) が起こり得ます。この制限の回避策の 1 つは、QRWTSRVR 事前開始ジョブの MAXUSE 値を 1 に設定することです。そうすると、CHGPJE SBS(D) (SYSWRK) PGM(QRWTSRVR) MAXUSE(1) のようになります。

関連概念:

- 115 ページの『オブジェクト関連セキュリティー』
- IBM i 製品がサーバー・システムである場合、セキュリティーを適用できる 2 つのオブジェクト関連レベルがあり、リレーショナル・データベース・テーブルへのアクセスを制御します。

関連資料:

- 活動化グループ再利用 (RCLACTGRP) コマンド
- ネットワーク属性変更 (CHGNETA) コマンド

サーバー・アクセス制御出口プログラムのパラメーター・リスト

ターゲット・サーバー上のユーザー出口プログラムは 2 つのパラメーター値を渡します。すなわち、文字戻りコード・フィールドと、さまざまなパラメーター値を含む文字データ構造です。

ターゲット・サーバー上のユーザー出口プログラムは、文字データ構造のパラメーター値を使用します。これらの値は TDDM によって渡され、ソース・サーバーからの要求を許可するかどうかの評価されます。ファイル・アクセス要求またはコマンド要求が TDDM に送信されるたびにパラメーター・リストが作成されます。「サブアプリケーション」フィールドに示されているいずれかの関数が要求されると、パラメーター・リストが作成されます。ファイル入出力操作の実行には、ファイル・オープン要求のみに対してこのパラメーター・リストが作成され、その後の入出力操作要求に関するリストは作成されません。

プログラムはパラメーター・リストを使用して、ソース・サーバー・ユーザーのファイル・アクセスまたはコマンド要求を受け入れるか拒否するかを判別します。リストには次のパラメーターと値が含まれます。

- ソース・サーバー・ユーザーの要求が実行されるユーザー・プロファイルまたはデフォルト・ユーザー・プロファイルの名前。
- 使用されているソース・サーバー上のアプリケーション・プログラムの名前。DDM を使用する場合、名前は *DDM になります。DRDA を使用する場合、名前は *DRDA になります。
- ターゲット・サーバーまたはそのいずれかのファイル上での使用が要求されているコマンドまたは関数 (サブアプリケーション) の名前。

以下の表にリストされているほとんどの関数は直接ファイルに影響を及ぼします。それには、ソース・サーバー・ユーザーが**ファイル記述表示 (DSPFD)** または**ファイル・フィールド記述表示 (DSPFFD)** などのコマンドを指定する際にファイルから情報を抽出する EXTRACT 関数が含まれています。メンバーの特性を変更できる CHGMBR 関数などのいくつかの関数は、メンバー関連の関数です。COMMAND 関

- 数は、ターゲット・サーバー上で実行するコマンド・ストリングを、リモート・コマンド投入 (SBMRMTCMD) コマンドによってサブミットするように指示します。SQLCNN 関数は DRDA 接続の試行を指定します。
- 前のパラメーターで指定されている方法でアクセスされるファイル (オブジェクト) の名前。コマンド・ストリング (COMMAND) かストリームおよびディレクトリー・アクセス・コマンドがサブミットされる場合や、DRDA コマンドの場合は、このフィールドは適用されません。
 - ストリームおよびディレクトリー・アクセス・コマンドを指定すると、オブジェクトおよびディレクトリー・フィールドは *SPC の値になります。ユーザーは、「その他」フィールドに進んで、代替オブジェクト名と代替パス名を入手しなければなりません。
 - ファイルにアクセスしようとしている場合、ファイルが入っているライブラリーの名前。
 - ファイル・メンバーにアクセスしようとしている場合、ファイル・メンバーの名前。ストリームおよびアクセス・コマンドは *N の値になります。
 - 形式フィールドは、DDM または DRDA の場合には適用されません。
 - 次のフィールドの使用法に応じて長さが変わります。
 - 以下の 6 つの値のうち最大 3 つに「その他」フィールドが使用されます。最初の 2 つは常に指定し (システム名を判別できない場合、2 つ目の値に *N を使用することもできます)、「サブアプリケーション」フィールドで指定されている関数のタイプに応じてその後の 4 つのうちいずれかを指定できます。
 - ソース・サーバーのロケーション名。APPC 通信を使用している場合、この名前は、ターゲット・サーバーの装置記述で指定されているソース・サーバーの RMTLOCNAME パラメーター値と一致します。
 - ソース・サーバーのシステム名。
 - ファイルが既に指定済みで、入出力操作作用に開かれる (OPEN) 場合、このフィールドは要求されている操作のタイプを示します。例えば、読み取り操作専用でファイルを開く場合は、入力要求値は 1 に設定され、残りの値は 0 に設定されます。
 - 代替オブジェクト名。
 - 代替ディレクトリー名。
 - iSeries コマンドの名前。コマンド・ストリングをサブミットする場合はその後に、すべてのサブミットされるパラメーターと値が続きます。

表 5. ターゲット・サーバー上のユーザー・出口プログラムのパラメーター・リスト

フィールド	タイプ	長さ	説明
ユーザー	文字	10	ターゲット DDM ジョブのユーザー・プロファイル名。
アプリケーション	文字	10	アプリケーション名: • 分散データ管理の場合は '*DDM '。

表 5. ターゲット・サーバー上のユーザー出口プログラムのパラメーター・リスト (続き)

フィールド	タイプ	長さ	説明
サブアプリケーション	文字	10	要求される関数: <ul style="list-style-type: none"> • 'ADDMBR ' 'DELETE ' 'RGZMBR ' • 'CHANGE ' 'EXTRACT ' 'RMVMBR ' • 'データ域変更 (CHGDTAARA) ' 'INITIALIZE' 'RNMMBR ' • 'CHGMBR ' 'LOAD ' 'データ域検索 (RTVDTAARA)' • 'CLEAR ' 'LOCK ' 'SNDDTAQ ' • 'CLRDTAQ ' 'オブジェクト移動 (MOVE) ' • 'COMMAND ' 'OPEN ' • 'オブジェクト・コピー (COPY) ' 'RCVDTAQ ' • 'CREATE ' 'RENAME ' • 'SQLCNN '
オブジェクト	文字	10	指定するファイル名。サブアプリケーション・フィールドが 'COMMAND ' の場合は *N が使用されます。ファイルがドキュメントまたはフォルダーの場合は *SPC が使用されます。
	文字	10	指定するライブラリー名。サブアプリケーション・フィールドが 'COMMAND ' の場合は *N が使用されます。ライブラリーがフォルダーの場合は *SPC が使用されます。
メンバー	文字	10	指定するメンバー名。メンバー名が適用外の場合は *N が使用されます。
形式	文字	10	DDM には適用しません。
長さ	10 進数	5,0	次のフィールドの長さ。
ソース・リモート・ロケーション	文字	10	ソース・システムのリモート・ロケーション・ユニット名 (SNA の場合)。
ソース・システム名	文字	10	リモート・サーバーのシステム名。この値が使用不可の場合、このフィールドには '*N ' が含まれます。

表 5. ターゲット・サーバー上のユーザー出口プログラムのパラメーター・リスト (続き)

フィールド	タイプ	長さ	説明
その他	文字	2000	<p>この 2000 バイト域の使用法は、要求関数に応じて異なります。SQLCNN の場合は、DRDA マッピングを使用する必要があります。他の関数の場合は、DDM マッピングを使用します。</p> <p>DDM を使用する場合:</p> <p>以下の内容は、関数に応じて変わります。OPEN を指定してファイルを開く場合は、以下のとおりです。</p> <p>1 入力要求 Char(1) 1=yes 0=no</p> <p>1 出力要求 Char(1) 1=yes 0=no</p> <p>1 更新要求 Char(1) 1=yes 0=no</p> <p>1 削除要求 Char(1) 1=yes 0=no</p> <p>12 代替オブジェクト名。</p> <p>63 代替ディレクトリー名。</p> <p>1921 COMMAND を指定してコマンドをサブミットする場合、コマンド・ストリング。</p> <p>DRDA を使用する場合:</p> <p>9 DRDA アプリケーション・リクエスターのタイプ定義名。DRDA アプリケーション・リクエスターの製品 ID。</p> <p>3 製品コード。</p> <p>2 バージョン ID。</p> <p>2 リリース ID。</p> <p>1 モディフィケーション・レベル。</p> <p>1983 予約済み</p>
<p>注:</p> <p>*N = NULL 値は、値が指定されていないパラメーター位置を示します。その後に定位置形式の他のパラメーターが続くことができます。</p>			

例: ユーザー出口プログラムの使用による DRDA および DDM サーバーのアクセス制御

このユーザー出口プログラムでは、すべての DRDA 操作と、ユーザー ID が 'ALIEN' 以外の DRDA 接続をすべて許可する PL/I ユーザー出口プログラムの例を示します。

注: コードの例を使用することは、443 ページの『コードに関するライセンス情報および特記事項』の条件に同意したものとされます。

```

| /*****
| /*
| /* PROGRAM NAME: UEPALIEN
| /*
| /* FUNCTION:      USER EXIT PROGRAM THAT IS DESIGNED TO
| /*                RETURN AN UNSUCCESSFUL RETURN CODE WHEN
| /*                USERID 'ALIEN' ATTEMPTS A DRDA CONNECTION.
| /*                IT ALLOWS ALL TYPES OF DDM OPERATIONS.
| /*
| /* EXECUTION:    CALLED WHEN ESTABLISHED AS THE USER EXIT
| /*                PROGRAM.
| /*
| /* ALL PARAMETER VARIABLES ARE PASSED IN EXCEPT:
| /*
| /* RTNCODE -     USER EXIT RETURN CODE ON WHETHER FUNCTION IS
| /*                ALLOWED: '1' INDICATES SUCCESS; '0' FAILURE.
| /*
| /*****
|
| UEPALIEN: PROCEDURE (RTNCODE,CHARFLD);
|
| DECLARE RTNCODE CHAR(1);          /* DECLARATION OF THE EXIT          */
|                                  /* PROGRAM RETURN CODE. IT          */
|                                  /* INFORMS REQUEST HANDLER          */
|                                  /* WHETHER REQUEST IS ALLOWED.      */
| DECLARE                          /* DECLARATION OF THE CHAR          */
| 1 CHARFLD,                       /* FIELD PASSED IN ON THE CALL.     */
|
| 2 USER      CHAR(10),            /* USER PROFILE OF DDM/DRDA USER   */
| 2 APP       CHAR(10),            /* APPLICATION NAME                  */
| 2 FUNC      CHAR(10),            /* REQUESTED FUNCTION                */
| 2 OBJECT    CHAR(10),            /* FILE NAME                          */
| 2 DIRECT    CHAR(10),            /* LIBRARY NAME                      */
| 2 MEMBER    CHAR(10),            /* MEMBER NAME                       */
| 2 RESERVED  CHAR(10),            /* RESERVED FIELD                    */
| 2 LENGTH    PIC '99999',         /* LENGTH OF USED SPACE IN REST     */
| 2 REST,                          /* REST OF SPACE = CHAR(2000)       */
|
| 3 LUNAME    CHAR(10),            /* REMOTE LU NAME (IF SNA)           */
| 3 SRVNAME   CHAR(10),            /* REMOTE SERVER NAME                */
| 3 TYPDEFN   CHAR(9),            /* TYPE DEF NAME OF DRDA AR          */
| 3 PRDID,    /* PRODUCT ID OF DRDA AR             */
|
| 5 PRODUCT   CHAR(3),            /* PRODUCT CODE                      */
| 5 VERSION   CHAR(2),            /* VERSION ID                        */
| 5 RELEASE   CHAR(2),            /* RELEASE ID                        */
| 5 MOD       CHAR(1),            /* MODIFICATION LEVEL                */
| 5 RDBNAME   CHAR(18),           /* RDB NAME                          */
| 5 REMAING   CHAR(1965),         /* REMAINING VARIABLE SPACE         */
|
| START:
| IF (USER = 'ALIEN' &          /* IF USER IS 'ALIEN' AND          */
|     FUNC = 'SQLCNN') THEN     /* FUNCTION IS DRDA CONNECT        */
|     RTNCODE = '0';           /* SET RETURN CODE TO UNSUCCESSFUL */
| ELSE                          /* IF ANY OTHER USER, OR DDM       */
|     RTNCODE = '1';           /* SET RETURN CODE TO SUCCESSFUL   */
|
| END UEPALIEN;

```

図 22. 例: PL/I ユーザー出口プログラム

機能使用 ID の使用による DRDA および DDM サーバーのアクセス制御

拡張プログラム間通信機能 (APPC) と TCP/IP の両方で使用される、分散データ管理機能 (DDM) と分散リレーショナル・データベース体系 (DRDA) サーバーのセキュリティ機能は、ユーザー出口プログラムを書かなくても、QIBM_DB_DDMDRDA 機能使用 ID を使用することで、DDM および DRDA 上でのユーザーやグループによるデータベース・サーバーへのアクセスをブロックすることができます。

機能使用 ID、QIBM_DB_DDMDRDA は、ユーザーやグループに対して DDM および DRDA アプリケーション・サーバーのインバウンド接続要求をブロックするのに使用され、通信プロトコルには基づいていません。機能使用 ID は、機能使用法の変更 (CHGFCNUSG) コマンドと機能使用法の処理 (WRKFNCUSG) コマンドを使用して構成することができます。機能使用 ID は、デフォルトの権限が *ALLOWED に設定された状態で出荷されます。セキュリティの責任担当者は、特定のユーザーやグループのアクセスを簡単に拒否することができます。この機能使用 ID は、ユーザー出口プログラムを使用する手法の代替となることを意図されています。コーディングは不要です。簡単に変更でき、監査も可能です。

以下は、ユーザーが DDM または DRDA へのアクセスを拒否された例です。

```
CHGFCNUSG FCNID(QIBM_DB_DDMDRDA) USER(user1) USAGE(*DENIED)
```

この機能使用 ID の権限の検査は、ユーザー出口プログラムが制御する前に行われます。ユーザーが機能使用の権限の検査に失敗すると、ユーザー出口プログラムは呼び出されません。

多くのアプリケーションと一部の IBM 製品は、QRWTSRVR ジョブに依存しています。アクセスを拒否する範囲が広すぎると、機能使用の管理が原因であることを明確に示さない形で、これらのアプリケーションに障害が発生する可能性があります。ほとんどのクライアントは、ユーザーがアクセスを拒否されたときにこの接続の障害を、接続がドロップしていたと報告します。

ユーザーは、IBM i リリース 6.1、6.1.1、および 7.1 の接続時に、CPF9898 メッセージ

```
「QIBM_DB_DDMDRDA FUNCTION USAGE IS DENIED FOR USER PROFILE XXXXXXXXX」
```

が出されてアクセスを拒否されます。このメッセージは、機能使用 ID QIBM_DB_DDMDRDA が、ユーザーのユーザー・プロファイルのアクセスを拒否するように設定されていることを示します。システム管理者がユーザーのユーザー・プロファイルにアクセス拒否を設定する前に、ユーザーが接続を確立した場合、そのユーザーは、既存の接続は引き続き使用することができますが、それ以降の接続ではアクセスを取り消されます。

特殊なケース:

- 1 つ目の特殊なケースは、機能の使用に対して、ユーザーのプロファイルが 2 回以上参照される場合です。ユーザーは 1 つ以上のグループ・プロファイルに属することができるため、機能の使用を指定すると、ユーザーは 1 つの機能の使用に対して複数の方法で参照される場合があります。機能の使用に対して、ユーザーが属する複数のグループ・プロファイルが参照され、ユーザー・プロファイルが参照されない場合、リストされているグループ・プロファイルのいずれかが USAGE(*ALLOWED) と構成されていれば、ユーザーは機能の使用が許可されます。1 つのグループ・プロファイルで機能の使用が *DENIED と構成されていても、別のグループ・プロファイルで *ALLOWED と構成されていれば、ユーザーは機能の使用が許可されます。必要なのは、グループ・プロファイルの 1 つで機能の使用が許可されていることです。

- 2 つ目の特殊なケースは、機能の使用に対して、ユーザーが属する 1 つ以上のグループ・プロファイルが参照され、ユーザー・プロファイルが明示的に参照される場合です。ユーザー・プロファイルの明示的な参照は、そのユーザーが機能の使用を許可されるかどうかを決定します。ここでは、ユーザーが 2 つのグループ・プロファイル GRPA と GRPB に属するとします。両方のグループ・プロファイルで機能の使用が *ALLOWED と構成されており、ユーザー・プロファイルで *DENIED と構成されている場合、ユーザーは機能の使用が許可されません。機能の使用に対するユーザー・プロファイルの参照は、グループ・プロファイルの参照よりも優先されます。

監査レコード:

監査が有効で、監査レベルが許可の失敗を記録するように構成されている場合は、セキュリティー監査レコードが QSYS/QAUDJRN セキュリティー監査実行記録に書き込まれます。GR-F 監査レコードは、機能使用の検査に失敗した場合に出される *USAGEFAILURE とともに表示されます。

例:

```
CHGSYSVAL SYSVAL(QAUDCTL) VALUE('*AUDLVL')
CHGSYSVAL SYSVAL(QAUDLVL) VALUE('*AUTFAIL')
DSPJRN JRN(QSYS/QAUDJRN) FROMTIME('01/25/2012' 080000)

Display Journal Entries

Journal . . . . . : QAUDJRN Library . . . . . : QSYS
Largest sequence number on this screen . . . . . : 00000000000000003036
Type options, press Enter.
5=Display entire entry

Opt Sequence Code Type Object Library Job Time
3035 T GR QRWTSRVR 8:24:29
3036 T GR QRWTSRVR 8:24:30

Display Journal Entry

Object . . . . . : Library . . . . . :
Member . . . . . :
Incomplete data . . : No Minimized entry data : *NONE
Sequence . . . . . : 3036
Code . . . . . : T - Audit trail entry
Type . . . . . : GR - General purpose audit record

Entry specific data
Column *...+...1...+...2...+...3...+...4...+...5
00001 'FZRSLFUSER *USAGEFAILURE '
00051 ' '
00101 ' QIBM_DB_DDMRDA '
00151 ' '
00201 ' '
00251 ' '
00301 ' '

```

最後に、この機能使用 ID は、機能使用の構成「*ALLOBJ 特殊権限」が *USER に設定された状態で出荷されます。*ALLOBJ ユーザー特殊権限を持つユーザーには明示的な機能使用の権限が付与されているため、それらのユーザーは、この設定によって機能を使用することが許可されます。この *ALLOBJ 権限を許可しない場合は、機能使用法の変更 (CHGFCNUSG) コマンドを使用して、構成を *NOTUSED に変更します。

例:

```
CHGFCNUSG FCNID(QIBM_DB_DDMRDA) ALLOBJAUT(*NOTUSED)
```

オブジェクト関連セキュリティー

IBM i 製品がサーバー・システムである場合、セキュリティーを適用できる 2 つのオブジェクト関連レベルがあり、リレーショナル・データベース・テーブルへのアクセスを制御します。

ネットワーク属性変更 (CHGNETA) コマンド上では DDMACC パラメーターが使用され、このシステム上のテーブルが別のシステムでアクセスできるかどうかを示されます。アクセスできる場合には、着信 DRDA 要求がどのセキュリティー・レベルで検査されるかも示されます。

- DDMACC パラメーター上で *REJECT が指定されている場合、サーバーで受信される分散リレーショナル・データベース要求はすべて拒否されます。しかし、このシステム (クライアントとして) は SQL 要求を使用して、アクセスを許可する他のシステムにあるテーブルにアクセスすることができます。*REJECT を指定する IBM i 環境では、リモート・システムによるデータベースのアクセスは行えません。

SQL 要求がすでに使用中である時に *REJECT が指定された場合には、このシステムのデータベースに対するアクセスを要求するシステムからの新しい ジョブはすべて拒否され、それらのジョブにエラー・メッセージが戻されます。既存のジョブに影響はありません。

- DDMACC パラメーターに *OBJAUT が指定されている場合には、通常のオブジェクト・レベルのセキュリティがサーバー上で使用されます。

DDMACC パラメーターは最初 *OBJAUT に設定されています。*OBJAUT 値はリモート要求をすべて許可しますが、それらの要求はこのサーバー上のオブジェクト権限によって制御されます。DDMACC 値が *OBJAUT である場合には、このジョブで使用されるユーザー・プロファイルに、私用権限、共通権限、グループ権限、または借用権限を通して適切なオブジェクト権限が与えられているか、クライアント・ジョブによって必要とされるオブジェクトの権限リストにプロファイルが存在していなければなりません。システム上の各 SQL オブジェクトについて、そのオブジェクトにアクセスする権限を、すべてのユーザー、または特定のユーザーのみに (ユーザー ID によって) 与えることができます。また、誰にもその権限を与えないこともできます。

オブジェクトで許可されるユーザー ID は、サーバー・ジョブのユーザー ID です。どのユーザー・プロファイルの下でサーバー・ジョブが稼働するかについては、88 ページの『APPC ネットワークでのセキュリティの要素』のトピックを参照してください。

TCP/IP 接続の場合には、まず最初にサーバー・ジョブは QUSER で実行されます。ユーザー ID が検査された後に、接続要求で指定されたユーザー・プロファイルでジョブが実行されるようにユーザー・プロファイルの交換が行われます。ジョブは、そのユーザー・プロファイルの属性 (たとえば、ライブラリー・リスト) を継承します。

値 *OBJAUT が指定されると、それ以上の検査 (IBM i オブジェクト・レベルのセキュリティを超えるもの) は必要でないことを示します。

- DDM ジョブでは、DDMACC パラメーターで出口プログラム (またはアクセス制御プログラム) の名前が指定されると、追加のセキュリティのレベルが使用されます。DDM クライアントのユーザーが、IBM i オペレーティング・システムの特定のファイルにアクセスするのに特定のコマンドを使用できるかどうかを制御するために、出口プログラムを使用することができます。

DRDA ジョブでは、DDMACC パラメーターで出口プログラム (アクセス制御プログラム) の名前が指定されると、システムは *OBJAUT が指定されたかのようにその入力を扱います (1 つの例外を除いて)。出口プログラムが DRDA ジョブに対して行えるのは、接続要求を拒否することだけです。

最初は *OBJAUT に設定される DDMACC パラメーターは、ネットワーク属性変更 (CHGNETA) コマンドを使用すると前述の値のいずれかに変更できます。そして、ネットワーク属性表示 (DSPNETA) コマンドで現行値を表示できます。また、ネットワーク属性検索 (RTVNETA) コマンドを使用して、制御言語 (CL) プログラム内の値を入手することもできます。

DDMACC パラメーター値が変更されるとその変更はすぐに反映されますが、システムでサーバーとして開始された新しい分散リレーショナル・データベース・ジョブだけが影響を受けます。このサーバー上で変更前に実行されていたジョブは、古い値を使用し続けます。

関連概念:

通信管理 (PDF)

107 ページの『ユーザー出口プログラムの使用による DRDA および DDM サーバーのアクセス制御』拡張プログラム間通信機能 (APPC) と TCP/IP の両方で使用される Distributed Relational Database Architecture (DRDA) サーバーのセキュリティ機能は、ネットワーク属性変更 (CHGNETA) コマンドの DDMACC パラメーターの使用を DRDA に拡張します。

関連資料:

ネットワーク属性変更 (CHGNETA) コマンド

ネットワーク属性表示 (DSPNETA) コマンド

ネットワーク属性取得 (RTVNETA) コマンド

DRDA: 分散リレーショナル・データベース・オブジェクトに対する権限

SQL GRANT および REVOKE ステートメント、または制御言語 (CL) であるオブジェクト権限認可 (GRTOBJAUT) およびオブジェクト権限取り消し (RVKOBJAUT) コマンドのいずれかを使用して、リレーショナル・データベース・オブジェクトに対するユーザー権限の認可および取り消しを行うことができます。

SQL GRANT および REVOKE ステートメントは、パッケージ、テーブル、およびビューのみで動作します。ある場合には GRTOBJAUT および RVKOBJAUT を使用して、それ以外のオブジェクトに対する権限をユーザーに認可する必要があります。

SQL ステートメントの場合に検査の対象となる権限は、ステートメントが静的か、動的か、または対話式に実行されるかによって異なります。

対話式 SQL ステートメントの場合は、そのステートメントの処理を行う人の権限の検査が行われます。対話式の SQL ステートメントに対しては、借用権限は使用されません。

分散リレーショナル・データベース・アプリケーションを実行するユーザーは、サーバー上で SQL パッケージを実行するための権限を必要とします。GRANT EXECUTE ON PACKAGE ステートメントを使用すると、SQL パッケージの所有者、またはその SQL パッケージに対する管理特権を持っているユーザーは、指定したユーザーに SQL パッケージの中のステートメントを実行する特権を認可することができます。このステートメントを使用して、サーバーに対する権限を認可されているすべてのユーザー、またはサーバー上の 1 つ以上のユーザー・プロファイルのリストに、SQL パッケージの中のステートメントを実行する特権を与えることができます。

通常、ユーザーは、CRTSQLxxx コマンドを使用して作成された分散アプリケーション・プログラムに対する権限を認可されている場合には、パッケージに対して処理特権を持っています。パッケージが構造化照会言語パッケージの作成 (CRTSQLPKG) コマンドを使用して作成されている場合には、そのパッケージに対する処理特権をユーザーに授与しなければならないことがあります。このステートメントは、SQL プログラムに組み込むか、または対話式 SQL を使用して出すことができます。次に、ステートメントの例を示します。

```
GRANT EXECUTE
ON PACKAGE SPIFFY.PARTS1
TO PUBLIC
```

REVOKE EXECUTE ON PACKAGE ステートメントを使用すると、SQL パッケージの所有者、またはその SQL パッケージに対する管理特権を持っているユーザーは、指定したユーザーから SQL パッケージの

中のステートメントを実行する特権を取り去ることができます。サーバーに対する権限を認可されているすべてのユーザー、またはサーバー上の 1 つ以上のユーザー・プロファイルのリストに対する EXECUTE 特権を取り去ることができます。

同一のユーザーに同一の特権を複数回授与している場合には、そのユーザーからその特権を取り消すことによって、すべての認可がいずれも無効になります。あるユーザーに以前授与した SQL パッケージに対する EXECUTE 特権を取り消した場合には、その SQL パッケージに対する EXECUTE 特権の授与は、誰が授与したものであっても、すべて無効になります。次に、ステートメントの例を示します。

```
REVOKE EXECUTE  
ON PACKAGE SPIFFY.PARTS1  
FROM PUBLIC
```

SQL パッケージに対する権限は、オブジェクト権限認可 (GRTOBJAUT) コマンドを使用して授与し、オブジェクト権限取り消し (RVKOBJAUT) コマンドを使用して取り消すこともできます。

関連資料:

構造化照会言語パッケージの作成 (CRTSQLPKG) コマンド

オブジェクト権限の付与 (GRTOBJAUT) コマンド

オブジェクト権限取り消し (RVKOBJAUT) コマンド

SQL オブジェクトのセキュリティー

134 ページの『分散リレーショナル・データベースのステートメント』

SQL 言語を使用して組み込まれるステートメントは、特に分散リレーショナル・データベースをサポートするためのものです。

DRDA: 分散リレーショナル・データベースの借用権限のもとで実行されるプログラム

分散リレーショナル・データベース・プログラムは、借用権限のもとで実行することができます。このことは、ユーザーが、プログラムの実行時にプログラムで使用されるオブジェクトに対するプログラム所有者の権限を借用することを意味します。プログラムが命名のための *SQL プリコンパイラー・オプションを使用して作成された時は、そのプログラムはプログラム所有者のユーザー・プロファイルのもとで実行されません。

異環境システムからの SQL パッケージでは、そのパッケージの中の静的 SQL ステートメントのすべてについて、常にパッケージ所有者の権限を借用します。OPTION(*SQL) を指定した CRTSQLxxx コマンドを使用して IBM i オペレーティング・システム上で作成された SQL パッケージでも、そのパッケージの中の静的 SQL ステートメントのすべてについて、パッケージ所有者の権限を借用します。

分散リレーショナル・データベース管理担当者は、プログラム借用表示 (DSPPGMADP) コマンドを使用することによって、サーバーにおけるセキュリティーの問題を検査することができます。DSPPGMADP コマンドを実行すると、以下に示すように、指定したユーザー・プロファイルを使用するプログラムおよび SQL パッケージが表示されます。コマンドを実行した結果をプリンターまたは出力ファイルに送ることもできます。

採用するプログラムの表示

ユーザー・プロファイル. : MPSUP

オブジェクト	ライブラリー	タイプ	ASP 装置	属性	テキスト
INVENT	SPIFFY	*PGM	*SYSBAS		借用プログラム
CLIENT1	SPIFFY	*PGM	*SYSBAS		借用プログラム
TESTINV	TEST	*PGM	*SYSBAS	CLP	テスト在庫プログラム
INVENT1	SPIFFY	*SQLPKG	*SYSBAS		SQL パッケージ
CLIENT1	SPIFFY	*SQLPKG	*SYSBAS		SQL パッケージ
TESTINV	SPIFFY	*SQLPKG	*SYSBAS		SQL パッケージ

終わり

続行するには、Enter キーを押してください。

F3= 終了 F12= 取り消し F17= 最上部 F18= 最下部

分散リレーショナル・データベースの保護方針

サーバー上の重要なデータを無許可アクセスから保護するには、IBM i 分散リレーショナル・データベースでのネットワーク・セキュリティーを計画しなければなりません。ただし、リレーショナル・データベースの種類が分散型であるため、セキュリティー計画では、ネットワークの中におけるデータの可用性が不必要に制約されることがないようにしなければなりません。

分散リレーショナル・データベース管理担当者が行う必要のある決定の 1 つとして、ネットワークの中の各システムに適したシステム・セキュリティー・レベルの決定があります。システム・セキュリティー・レベルが 10 の場合は、サーバーに対するセキュリティーは、システム・サイトにおける物理的セキュリティー以外には何も提供されません。システム・セキュリティー・レベルが 20 の場合は、ネットワーク・セキュリティー検査が行われて、ローカル・システムおよびリモート・システムが正しく識別されるので、サーバーに対してある程度の保護が提供されます。ただし、このレベルのセキュリティーでは、重要なデータベース要素を無許可アクセスから保護するために必要なオブジェクト許可は提供されません。したがって、ネットワークの中のシステムが特定のシステム・オブジェクトの保護を必要とする場合には、30 かそれ以上のシステム・セキュリティー・レベルの選択をお勧めします。

分散リレーショナル・データベース管理担当者は、ネットワーク上のクライアントとサーバーとの間で通信を確立する方法についても考慮しなければなりません。そのためには、以下の質問事項を解決する必要があります。

- サーバー上にデフォルトのユーザー・プロファイルを存在させるかどうか。

多くのユーザー・プロファイルをネットワークの中で維持管理するのは難しい場合があります。これに対して、通信サブシステムの項目の中にデフォルトのユーザー・プロファイルを作成すると、サーバーが保護ロケーションでない場合には、サーバーは受信する通信要求に対して開放されることとなります。この状態は、場合によっては、受け入れ可能なこともあります。デフォルトのユーザー・プロファイルのために、システム保護機能が著しく低下して、セキュリティー要件を満たせない場合もあります。

たとえば、システムが多くのクライアントにサービスを提供する場合には、高いレベルのセキュリティーを必要とします。そのようなシステムのデータベースに脱落または損傷が生じた場合には、ネットワ

ーク全体が影響を被ることになります。アクセスを必要とする可能性のあるユーザーをすべて識別するユーザー・プロファイル、またはグループ・プロファイルをサーバー上に作成することは可能であるため、データベース管理担当者は、分散リレーショナル・データベースの作業を管理する通信サブシステムのために、デフォルトのユーザー・プロファイルを作成することを考える必要はありません。

これに対して、ネットワークの中で他のシステムに対してサーバーとして機能することがほとんどなく、機密性の高い重要なデータが入っていないシステムの場合には、分散リレーショナル・データベース作業を管理する通信サブシステムのデフォルトのユーザー・プロファイルを使用することができます。これが特に効果的であるのは、ネットワークの中の他のすべてのシステムで同じアプリケーションを使用して、このデータベースに対する作業を処理する場合です。

デフォルトのユーザーという概念は、厳密には APPC の使用の際にのみ適用されます。ただし、TCP/IP を使用しているシステムでは、これに類似した技法を使用することができます。ユーザー ID を 1 つ確立してから、そのもとでサーバー・ジョブを実行することができます。そして、すべてのクライアントでサーバー許可項目の追加 (ADDSVRAUTE) コマンドを使用して、接続するユーザーすべてがそのユーザー ID を使用する必要があることを指定できます。サーバーの DDM TCP/IP 属性の変更 (CHGDDMTCPA) コマンド上にある PWDRQD パラメーターの設定に応じて、サーバー認証項目でパスワードを指定するか、パスワードとして *NONE を指定することができます。この属性のデフォルトの値では、パスワードは必須です。

- データベース・オブジェクトに対するアクセスをどのように処理するか。

オブジェクトに対する権限は、私用権限、グループ権限、共通権限、借用権限、および認証リストによって認可することができます。通信要求が受諾されるためには、ユーザー・プロファイル (またはデフォルトのプロファイル) がサーバー上に存在していなければなりません。オブジェクトに対する権限をユーザーに認可する方法によっては、パフォーマンスに影響が生じる可能性があります。

分散リレーショナル・データベース・オブジェクトに対するアクセスの認可にあたっては、可能なかぎり、グループ権限または認証リストを使用してください。これらの検査の方が、すべての私用権限を検査する場合に比べて、時間およびシステム・リソースの節減を図ることができます。

TCP/IP 接続では、ユーザー ID のマッピングを行うことができるので、サーバーに接続できるユーザーごとに専用ユーザー ID を割り当てる必要はありません。

関連資料:

サーバー認証項目追加 (ADDSVRAUTE) コマンド

DDM TCP/IP 属性変更 (CHGDDMTCPA) コマンド

DRDA と DDM のアプリケーション開発

プログラマーは、IBM i 分散アプリケーション・プログラム用の SQL ステートメントを使用する、高水準言語プログラムを作成することができます。また DDM に対応している言語、ユーティリティ、およびアプリケーション・プログラムを示し、リモート・ファイルに正しくアクセスするための DDM 固有の情報について述べています。

関連概念:

SQL プログラミング

関連資料:

構造化照会言語パッケージの作成 (CRTSQLPKG) コマンド

DRDA のアプリケーション開発

プログラマーは、IBM i 分散アプリケーション・プログラム用の SQL ステートメントを使用する、高水準言語プログラムを作成することができます。

ローカル処理用に作成されるプログラムとの主な違いは、リモート・データベースに接続することができ、SQL パッケージを作成することができる点にあります。CONNECT SQL ステートメントを使用して、クライアントをサーバーに明示的に接続するか、またはプログラムの作成時に、リレーショナル・データベースの名前を指定して、暗黙接続することができます。また、SET CONNECTION、RELEASE、および DISCONNECT ステートメントを使用して、分散作業単位を使用するアプリケーションのために接続を管理できます。

SQL パッケージとは、分散リレーショナル・データベースでのみ使用される IBM i オブジェクトのことです。SQL のプリコンパイル処理の結果として作成するか、またはコンパイル済みプログラム・オブジェクトから作成することができます。SQL パッケージは、サーバー側に存在します。このパッケージには、SQL ステートメント、ホスト変数属性、およびサーバーがクライアントの要求を処理するのに使うアクセス・プランが入っている。

アプリケーション・プログラムは多くの異なるシステムに接続できるので、プログラマーは、システム間のデータ変換には十分に注意を払う必要があります。IBM i オペレーティング・システムでは、さまざまなタイプのデータ変換を用意しており、これには文字情報の管理用のコード化文字セット識別子 (CCSID) サポートも含まれています。

ローカル処理のアプリケーションに SQL 言語を使用するのと同じように、SQL 言語を使用してシステムで分散リレーショナル・データベース用のプログラムを作成し、保守することができます。以下の高水準言語の 1 つまたはいくつかを使用して、静的および動的構造化照会言語 (SQL) ステートメントを組み込むことができます。

- PL/I PRPQ
- ILE C
- COBOL/400
- ILE COBOL
- FORTRAN/400
- RPG/400®
- ILE RPG

分散アプリケーションを開発するためのプロセスは、ローカル処理用の SQL アプリケーションを開発するプロセスに類似しています。異なっているのは、分散処理のアプリケーションでは、接続されるリレーショナル・データベースの名前を指定しなければならないという点です。これはプログラムのプリコンパイル時、またはアプリケーション内で行うことができます。

使用される SQL オブジェクトは、1 つを除いて、ローカル・アプリケーションでも分散アプリケーションでも同じです。つまり、SQL パッケージというオブジェクトが 1 つだけ分散リレーショナル・データベースの専用になります。SQL プログラム作成 (CRTSQLxxx) コマンドを使用して、プログラムを作成します。このコマンドの xxx は、ホスト言語である、CI、CBL、CBLI、FTN、PLI、RPG、または RPGI を指します。SQL パッケージはこのプロセスにおけるプリコンパイルによって作り出される場合があります。既存の分散 SQL プログラムの場合は、構造化照会言語パッケージの作成 (CRTSQLPKG) コマンドによって SQL パッケージを作成します。

SQL ステートメントを組み込んだプログラムをプリコンパイルするためには、IBM DB2 Query Manager and SQL Development Kit for i ライセンス・プログラムをインストールしておかなければなりません。ただし、システムにインストールされているコンパイル済みプログラムだけを使用して、既存の分散 SQL プログラムから SQL パッケージを作成することができます。IBM DB2 Query Manager and SQL Development Kit for i ライセンス・プログラムを使用すると、対話式 SQL によって分散リレーショナル・データベースにアクセスすることができます。これはプログラムをデバッグするときに役立ちます。その理由は、プログラムをプリコンパイルしたりコンパイルしたりせずに、SQL ステートメントをテストすることができるからです。

分散リレーショナル・データベース・アプリケーションのプログラミングの考慮事項

IBM i オペレーティング・システム上の分散リレーショナル・データベース・アプリケーションのプログラミングに関する考慮事項は、次の 2 つのカテゴリーに分けられます。つまり、ローカル・システム上でサポートされる機能を扱うものと、他のシステムに接続しなければならないことに由来するものです。

関連資料:

43 ページの『ヒント: 分散リレーショナル・データベース・アプリケーションの設計』

分散リレーショナル・データベース・アプリケーションと、ローカル・データベースだけに使用するために開発されたアプリケーションとは、要件に大きな違いがあります。

分散リレーショナル・データベース・オブジェクトの命名:

SQL オブジェクトは、IBM i オブジェクトとして作成され、保守されます。DB2 for i プログラミングでは、次の命名方式のいずれかを使用することができます。すなわち、システムと (*SYS) と SQL (*SQL) のいずれかです。

使用する命名規則によって、ファイル名およびテーブル名を修飾する方法に影響が生じます。また、セキュリティおよび対話式 SQL 画面で使用される用語にも影響があります。分散リレーショナル・データベース・アプリケーションが別の IBM i プラットフォーム上のオブジェクトにアクセスするのであれば、命名規則は、2 つのうちどちらを使用することもできます。ただし、プログラムが DB2 for i でないシステムにアクセスする場合は、SQL 名だけしか使用できません。名前は、SQL の開始 (STRSQL) コマンドの NAMING パラメーター、CRTSQLxxx コマンドのいずれかの OPTION パラメーター、または呼び出しレベル・インターフェース (CLI) および Java Database Connectivity (JDBC) の命名接続プロパティーを使用して指定できます。

関連資料:

命名規則

システム (*SYS) 命名規則:

システム命名規則を使用するときは、ファイルは、ライブラリー/ファイル という形式によってライブラリー名で修飾されます。

テーブルがこの命名規則を使用して作成される場合には、テーブルが作成されるライブラリーの共通権限が前提となります。テーブル名が明示的に修飾されず、デフォルトのコレクション名が CRTSQLxxx コマンドまたは CRTSQLPKG コマンドの DFTRDBCOL パラメーターで使用される場合には、デフォルトのコレクション名が静的 SQL ステートメントで使用されます。ファイル名が明示的に修飾されず、デフォルトのコレクション名が指定されない場合には、以下の規則が適用されます。

- 特定の CREATE ステートメント以外のすべての SQL ステートメントで、SQL は、修飾されていないファイルを見つけるために、ライブラリー・リスト (*LIBL) を検索します。
- CREATE ステートメントでは、修飾されていないオブジェクトについて、以下のように解決します。

- CREATE TABLE: テーブル名は明示的に修飾されなければなりません。
- CREATE VIEW: ビューは、副選択で参照された最初のライブラリーの中に作成されます。
- CREATE INDEX: 索引は、その索引が作成されるテーブルが入っているコレクションまたはライブラリーの中に作成されます。

SQL (*SQL) 命名規則:

SQL 命名規則を使用するときは、テーブルは、コレクション.テーブル という形式によってコレクション名で修飾されます。

テーブル名が明示的に修飾されず、デフォルトのコレクション名が CRTSQLxxx コマンドまたは**構造化照会言語パッケージの作成 (CRTSQLPKG)** コマンドのデフォルトのリレーショナル・データベース・コレクション (DFTRDBCOL) パラメーターで指定される場合には、デフォルトのコレクション名が使用されます。テーブル名が明示的に修飾されず、デフォルトのコレクション名が指定されない場合には、以下の規則が適用されます。

- 静的 SQL では、デフォルトの修飾名はプログラム所有者のユーザー・プロファイルです。
- 動的 SQL または対話式 SQL では、デフォルトの修飾名はステートメントを実行するジョブのユーザー・プロファイルです。

関連資料:

構造化照会言語パッケージの作成 (CRTSQLPKG) コマンド

デフォルトのコレクション名:

プログラムのプリコンパイル時に、CRTSQLxxx コマンドの DFTRDBCOL パラメーターにデフォルトのコレクション名を指定することによって、デフォルトのコレクション名が SQL プログラムで使用されるように指定することができます。

DFTRDBCOL パラメーターは、*SYS 命名規則が使用される場合、修飾されていないファイルのライブラリーとして、あるいは、*SQL 命名規則が使用される場合、修飾されていないテーブルのコレクションとして、コレクション名をプログラムに提供します。プログラムのプリコンパイル時にデフォルトのコレクション名を指定しなかった場合には、それぞれの命名規則に応じて、修飾されていない名前に関する規則が適用されます。コレクション名が適用されるのは、静的 SQL ステートメントの場合だけです。

構造化照会言語パッケージの作成 (CRTSQLPKG) コマンドで DFTRDBCOL パラメーターを使用して、パッケージのデフォルトのコレクションを変更することもできます。また、SQL プログラムがコンパイルされた後で、新規 SQL パッケージを作成して、デフォルトのコレクションを変更することができます。

関連タスク:

150 ページの『SQL パッケージの作成 (CRTSQLPKG) コマンドの使用』

SQL パッケージの作成 (CRTSQLPKG) コマンドを入力して、コンパイル済みの分散リレーショナル・データベース・プログラムから SQL パッケージを作成することができます。また、このコマンドを使用すれば、前に作成されていた SQL パッケージを置き換えることもできます。

関連資料:

構造化照会言語パッケージの作成 (CRTSQLPKG) コマンド

分散リレーショナル・データベースへの接続:

分散リレーショナル・データベース・アプリケーションが分散されるのは、それが別のシステム上のリレーショナル・データベースへの接続可能性を備えているからです。

CONNECT ステートメントには、同じ構文でしかも働きの異なる 2 つのタイプがあります。

- CONNECT (タイプ 1) は、リモート作業単位用に使用されます。
- CONNECT (タイプ 2) は、分散作業単位用に使用されます。

組み込み SQL アプリケーションの場合、CRTSQLxxx コマンドの RDBCNNMTH パラメーターは CONNECT のタイプを判別します。CLI および Java アプリケーションは、言語の性質から、常に分散作業単位を使用します。

関連資料:

134 ページの『分散リレーショナル・データベースのステートメント』

SQL 言語を使用して組み込まれるステートメントは、特に分散リレーショナル・データベースをサポートするためのものです。

リモート作業単位:

リモート作業単位 機能は、SQL ステートメントのリモートでの準備および処理のために提供されています。

システム A の活動化グループは、システム B のサーバーに接続できます。その後、その活動化グループは、1 つ以上の作業単位の中で、B のオブジェクトを参照する静的または動的 SQL ステートメントを何回でも処理できます。B での 1 つの作業単位の終了後、活動化グループは、システム C などのサーバーに接続できます。

ほとんどの SQL ステートメントは、以下の制約付きでリモートで準備し、処理できます。

- 1 つの SQL ステートメントで参照するオブジェクトは、すべて、同じサーバーで管理しなければなりません。
- 1 つの作業単位の中の SQL ステートメントは、すべて、同じサーバーで実行しなければなりません。

活動化グループの状態:

活動化グループは、いつも 3 つの状態のいずれかになっています。

3 つの状態は次のとおりです。

- 接続可能で接続済み
- 接続不能で接続済み
- 接続可能で未接続

次の図は、状態遷移を示しています。

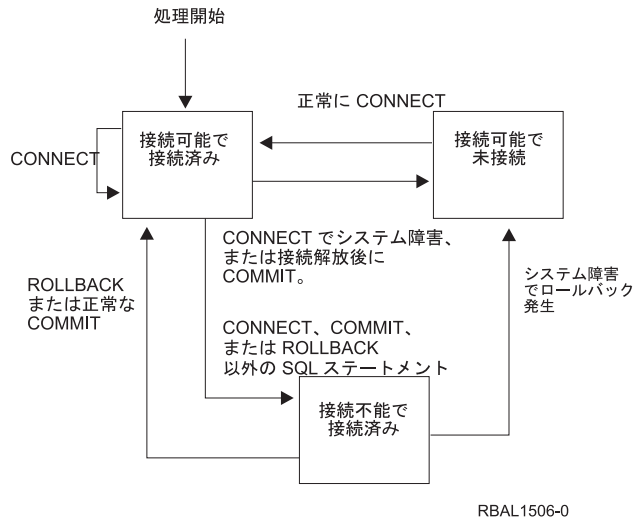


図 23. リモート作業単位の活動化グループの接続状態遷移

活動化グループの初期の状態は、**接続可能で接続済み** です。活動化グループが接続されるサーバーは、`CRTSQlxxx` コマンドと `STRSQL` コマンドの `RDB` パラメーターによって判別され、暗黙の `CONNECT` 操作を含むことができます。暗黙の `CONNECT` 操作は、暗黙または明示の `CONNECT` 操作が、正常であったかどうかは別として、すでに行われている場合には行われません。したがって、活動化グループは、2 度以上サーバーに暗黙に接続されることはありません。

接続可能で接続済み状態:

活動化グループはサーバーに接続され、`CONNECT` ステートメントを実行することができます。活動化グループがこの状態に入るのは、ロールバックまたは正常なコミットを接続不能で接続済みの状態から完了したとき、または接続可能で未接続の状態から `CONNECT` ステートメントが正常に実行されたときです。

接続不能で接続済みの状態:

活動化グループはサーバーに接続されていますが、サーバーを変更するための `CONNECT` ステートメントは正常に実行されませんでした。活動化グループが接続可能で接続済み状態からこの状態に入るのは、`CONNECT`、`COMMIT`、または `ROLLBACK` 以外の `SQL` ステートメントを実行したときです。

接続可能で未接続状態:

活動化グループがサーバーに接続していません。実行できる唯一の `SQL` ステートメントは、`CONNECT` です。

活動化グループがこの状態になるのは、以下の場合です。

- 接続は前に解放されており、`COMMIT` が正常に実行された。
- 接続が `SQL DISCONNECT` ステートメントを使用して切断されている。
- 接続は接続可能状態だったが、`CONNECT` ステートメントは正常に行われなかった。

活動化グループは、`CONNECT` によって接続可能状態でなくなることはないので、`CONNECT` ステートメントを連続して正常に実行することができます。活動化グループが現在接続されているサーバーに `CONNECT` を行うと、その他の場合の `CONNECT` ステートメントと同様に実行されます。

(COMMIT(*NONE) で実行しているのではない限り) CONNECT、 COMMIT、 DISCONNECT、 SET CONNECTION、 RELEASE、 または ROLLBACK 以外の SQL ステートメントに続く CONNECT は、正しく実行されません。エラーを避けるために、CONNECT ステートメントを実行する前にコミット、またはロールバックの操作を行ってください。

分散作業単位:

アプリケーション指示による分散作業単位でも、リモート作業単位と同じ方法で、SQL ステートメントをリモートで準備し、実行できます。

リモート作業単位の場合と同じように、システム A の活動化グループは、システム B のサーバーに接続することができ、その作業単位の終わる前に、B のオブジェクトを参照する静的または動的な SQL ステートメントをいくつでも実行することができます。1 つの SQL ステートメントで参照するオブジェクトは、すべて、同じサーバーで管理しなければなりません。ただし、リモート作業単位の場合とは異なり、同じ作業単位の中で、任意の数のサーバーと関係を持つことができます。コミットまたはロールバックの操作で作業単位が終了します。

活動化グループの状態:

活動化グループは、常に接続済み 状態であるか、または未接続 の状態です。すなわち接続がゼロであるか、1 つ以上の接続を持っています。活動化グループの各接続は、接続しているサーバーの名前によって唯一のものとして識別されます。

SQL 接続は、常に以下のいずれかの状態です。

- 現行で保持の状態
- 現行で解放の状態
- 休止で保持の状態
- 休止で解放の状態

活動化グループの初期状態:

活動化グループは、最初は接続済み状態で、接続を 1 つ持っています。接続の初期状態は、現行で保持 の状態です。

次の図は、状態遷移を示しています。

処理開始

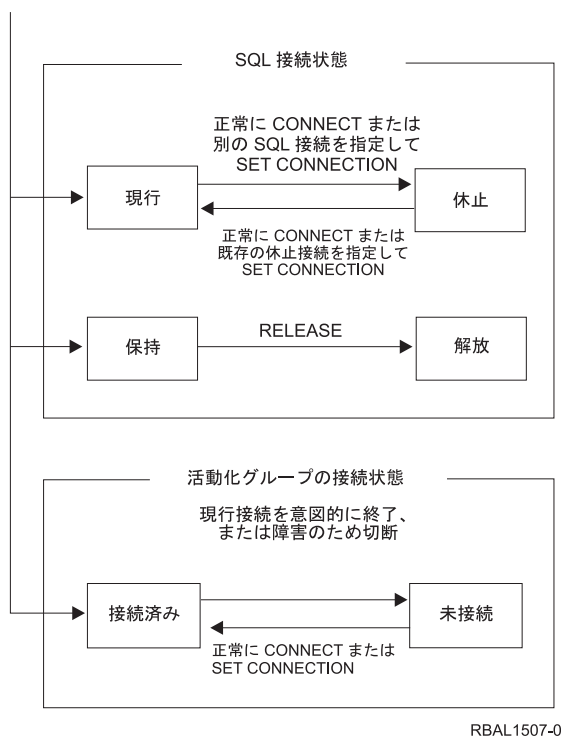


図 24. アプリケーションが指定する分散作業単位接続および活動化グループの接続の状態遷移

接続状態

このトピックでは、各種の接続状態とその変更方法を説明します。

アプリケーションが `CONNECT` ステートメントを処理したとき、そのサーバー名がアプリケーション・リクエスターに認識されていても、活動化グループの既存の接続セットの中に入っていない場合は、以下のようになります。

- 現行接続は休止状態で保持状態に置かれます。
- そのサーバー名は、接続セットに加えられ、新しい接続が現行状態で保持状態になります。

そのサーバー名がすでに活動化グループの既存の接続セットに入っている場合は、エラーとなります。

`SET CONNECTION` ステートメントを使用すると、休止状態の接続が現行状態になります。接続が現行状態になると、前の現行状態接続は (それがあった場合は)、休止状態になります。活動化グループの既存の接続セットの中で、現行状態になることができるのはいつでも 1 つだけです。接続状態が現行状態から休止状態に、または休止状態から現行状態に変わっても、その保持状態または解放状態には影響がありません。

`RELEASE` ステートメントを使用すると、接続状態は解放状態になります。活動化グループがコミット操作を実行すると、その活動化グループの各解放接続は終了します。接続状態が保持状態から解放状態に変わっても、その現行状態または休止状態には影響ありません。したがって、解放状態の接続は次のコミット操作までは使用することができます。接続を解放状態から保持状態に変える方法はありません。

活動化グループの接続状態:

CONNECT ステートメントを暗黙にまたは明示的に実行することによって、異なるサーバーを設定することができます。

以下の規則が適用されます。

- 活動化グループは、同時に同一のサーバーに複数の接続を持つことはできない。
- 活動化グループが SET CONNECTION ステートメントを実行するときの指定のロケーション名は、その活動化グループの接続セットの中の既存の接続でなければならない。
- 活動化グループが CONNECT ステートメントを実行するときの指定のサーバー名は、その活動化グループの接続セットの既存接続であってはならない。

活動化グループに現行接続がある場合は、その活動化グループは接続済み 状態です。

現行接続のサーバー名は、CURRENT SERVER という特殊レジスターに入っています。活動化グループは、そのサーバーが管理するオブジェクトを参照する SQL ステートメントを実行することができます。

未接続状態の活動化グループは、CONNECT または SET CONNECTION ステートメントの実行に成功すると接続済み状態になります。

活動化グループに現行接続がない場合は、その活動化グループは未接続 状態です。このときの CURRENT SERVER 特殊レジスターは、ブランクになっています。実行できる SQL ステートメントは、CONNECT、DISCONNECT、SET CONNECTION、RELEASE、COMMIT、および ROLLBACK だけです。

接続済み状態の活動化グループが未接続状態になるのは、その現行接続が意図的な終了となったか、または障害が発生し、サーバーでロールバックが行われて、接続が失われたために SQL ステートメントが正常に実行されなかった場合です。接続が意図的に終了となるのは、活動化グループがコミット操作を正常に実行し、その接続が解放状態になっているとき、またはアプリケーション・プロセスが正常に DISCONNECT ステートメントを実行したときです。

接続の終了時:

接続が終了すると、接続によって活動化グループが獲得していたすべてのリソースが割り振り解除されます。また、接続の作成と保守に使用されていたすべてのリソースも割り振り解除されます。

たとえば、活動化グループが RELEASE ステートメントを実行すると、次のコミット操作で接続が終了するときにすべてのオープン・カーソルはクローズされます。

また、通信障害が起きた場合も接続は終了することがあります。この場合は、その活動化グループは未接続状態になります。活動化グループのすべての接続は、その活動化グループが終了すると、終了します。

RWU および DUW の両方の接続管理による実行:

リモート作業単位 (RWU) 接続管理でコンパイルしたプログラムは、分散作業単位 (DUW) 接続管理でコンパイルしたプログラムで呼び出すことができます。RWU 接続管理でコンパイルしたプログラムは、SET CONNECTION、RELEASE、および DISCONNECT ステートメントを使用して、任意の活動接続を処理することができます。

ただし、DUW 接続管理でコンパイルしたプログラムが、RWU 接続管理でコンパイルしたプログラムを呼ぶときは、RWU 接続管理でコンパイルしたプログラムで実行される CONNECT が、CONNECT 機能の一部としてその活動化グループのすべての活動接続を終了しようとしています。

そのような CONNECT は、活動状態の接続が使用している会話が保護会話である場合、失敗します。さらに、非活動状態の接続のために保護会話が使用されていて、DDMCNV ジョブ属性が *KEEP であると、これらの未使用の DDM 会話も、RUW 接続管理でコンパイルされたプログラムの接続を失敗させます。このような事態を避けるためには、ジョブ属性は DDMCNV(*DROP) を使用して、RUW 接続管理でコンパイルしたプログラムを呼び出して CONNECT を実行する前に、RELEASE と COMMIT を実行してください。

同様に、RUW 接続管理でコンパイルしたプログラムのためのパッケージを作成した後で、DUW 接続管理でコンパイルしたプログラムのためのパッケージを作成するときは、DDMCNV(*DROP) を指定して実行するか、または DUW 接続管理でコンパイルしたプログラムのためのパッケージを作成した後で RCLDDMCNV を実行してください。

DUW 接続管理でコンパイルしたプログラムは、RUW 接続管理でコンパイルしたプログラムから呼び出すこともできます。DUW 接続管理でコンパイルしたプログラムが CONNECT を実行しても、RUW 接続管理でコンパイルしたプログラムによる接続は切断されません。この接続を、DUW 接続管理でコンパイルしたプログラムで使用することができます。

デフォルトの活動化グループのための暗黙の接続管理:

暗黙に接続されるのは、デフォルトの活動化グループのための最初の活動状態 SQL プログラムが最初の SQL ステートメントを発行したことをクライアントが検出し、以下の項目が該当する場合です。

- 発行された SQL ステートメントがパラメーターをもつ CONNECT ステートメントでない。
- デフォルトの活動化グループで SQL が活動状態になっていない。

分散プログラムの場合、暗黙の接続は RDB パラメーターに指定されたリレーショナル・データベースへの接続になります。非分散プログラムの場合、暗黙の接続はローカル・リレーショナル・データベースへの接続になります。

SQL が活動状態でなくなるとき、SQL はデフォルトの活動化グループのすべての活動状態の接続を終了します。クライアントが、そのプロセスの最初の活動状態 SQL プログラムが終了したことを検出し、さらに以下の条件がすべて該当する場合、SQL は活動状態になりません。

- 保留中の SQL 変更がない
- 保護会話を使用している接続がない
- SET TRANSACTION ステートメントが活動状態ではない
- CLOSQLCSR(*ENDJOB) でプリコンパイルされたプログラムを実行していなかった

保留中の変更、保護会話、または活動中の SET TRANSACTION があると、SQL は終了状態になります。CLOSQLCSR(*ENDJOB) を指定してプリコンパイルしたプログラムを実行する場合は、SQL はそのジョブが終わるまでデフォルトの活動化グループのために活動状態のままになっています。

- SQL が終了状態で、作業単位の終わりになっている場合。これは、SQL プログラムの外側で COMMIT または ROLLBACK コマンドが発行された場合に起きます。
- ジョブの終わり。

非デフォルトの活動化グループのための暗黙の接続管理:

暗黙に接続されるのは、その活動化グループのために最初の SQL ステートメントを発行したことをクライアントが検出し、それがパラメーター付きの CONNECT ステートメントではない場合です。

分散プログラムの場合、暗黙の接続は RDB パラメーターに指定されたりレシーショナル・データベースへの接続になります。非分散プログラムの場合、暗黙の接続はローカル・リレーショナル・データベースへの接続になります。

プロセスの以下の時点では、暗黙の切断が起きることがあります。

- 活動化グループが終了したときに、コミットメント制御が活動状態ではないか、活動化グループ・レベルのコミットメント制御が活動状態か、またはジョブ・レベルのコミットメント定義が作業単位境界にある場合。

ジョブ・レベルのコミットメント定義が活動中で、作業単位境界になっていないときは、SQL は終了状態になります。

- ジョブ・レベルのコミットメント定義がコミットされるか、ロールバックされるときに、SQL が終了状態の場合。
- ジョブの終わり。

以下のプログラム例は分散ではありません (接続は必要ありません)。これは Spiffy 社の地域支社で実行されるプログラムであり、ローカルの修理情報を報告書の形で収集するためのものです。

注: コードの例を使用することは、443 ページの『コードに関するライセンス情報および特記事項』の条件に同意したものとされます。

```
CRTSQLxxx PGM(SPIFFY/FIXTOTAL) COMMIT(*CHG) RDB(*NONE)
```

```
PROC: FIXTOTAL;
.
.
SELECT * INTO :SERVICE      A
      FROM REPAIRTOT;
EXEC SQL
COMMIT;
.
.
END FIXTOTAL;
```

A ローカル・リレーショナル・データベースに対して実行されるステートメント。

以下の例のような、別のプログラムでは、カンザス・シティー地域の Spiffy 社の販売店から同様の情報を収集することができます。これは、暗黙に接続および切断が行われる分散プログラムの例です。

```
CRTSQLxxx PGM(SPIFFY/FIXES) COMMIT(*CHG) RDB(KC101) RDBCNNMTH(*RUW)
```

```
PROC: FIXES;
.
.
EXEC SQL
SELECT * INTO :SERVICE      B
      FROM SPIFFY.REPAIR1;

EXEC SQL C
COMMIT;
.
.
END FIXES; D
```

B サーバーに対する暗黙接続。ステートメントはサーバーに対して実行されます。

- C 作業単位の終了。COMMIT が正常に行われた場合には、クライアントは接続可能で接続済みの状態に入ります。
- D SQL プログラムの終了時の暗黙切断。

明示接続管理:

CONNECT ステートメントを使用して、指定されたサーバーに明示的にクライアントを接続します。この SQL ステートメントはアプリケーション・プログラムに組み込むことも、対話式 SQL を使用して発行することもできます。

CONNECT ステートメントは、TO 文節または RESET 文節を指定して使用します。TO 文節を指定した CONNECT ステートメントを使用すると、特定の AS リレーショナル・データベースへの接続を指定することができます。RESET 文節を指定した CONNECT ステートメントでは、ローカル・リレーショナル・データベースへの接続を指定します。

TO 文節または RESET 文節を指定した CONNECT ステートメントを発行する (またはプログラムが発行する) ときは、指定された AS がリレーショナル・データベース・ディレクトリーに記述されていなければなりません。また、CONNECT ステートメントが正常に実行されるためには、AR が接続可能状態であればなりません。

CONNECT ステートメントの効果は、使用している接続管理によって異なります。RUW 接続管理では、CONNECT ステートメントの効果は以下の通りです。

- TO 文節または RESET 文節を指定した CONNECT ステートメントが正常に実行されると、以下のイベントが発生します。
 - アプリケーション・プロセスが、COMMIT HOLD または ROLLBACK HOLD SQL ステートメントの使用によって接続可能状態に入った場合、またはアプリケーション・プロセスが COMMIT(*NONE) を実行中である場合には、オープンカーソルはいずれもクローズされ、準備されていたステートメントはいずれも廃棄され、保持されていたリソースはいずれも前の AS から解放されます。
 - アプリケーション・プロセスが、前に接続されていた AS (もしあれば) から切断され、指定された AS に接続されます。
 - AS の名前が現行サーバー特殊レジスターに置かれます。
 - エラーを戻したシステム・モジュールを識別する情報は、SQL 通信域 (SQLCA) の SQLERRP フィールドか、SQL 診断域の DB2_MODULE_DETECTING_ERROR フィールドに置かれます。
- 何らかの理由で CONNECT が成功しないと、アプリケーションは接続可能で未接続の状態のままになります。接続可能で未接続の状態のアプリケーションでは、CONNECT ステートメントだけを実行できません。
- AR は、CONNECT によって接続可能状態でなくなることはないので、CONNECT ステートメントを連続して正常に実行することができます。AR が現在接続されている AS に CONNECT を行うと、その他の場合の CONNECT ステートメントと同様に実行されます。
- コミットメント制御で実行中に、CONNECT、SET CONNECTION、COMMIT、ROLLBACK、DISCONNECT、または RELEASE 以外の SQL ステートメントに続く CONNECT ステートメントは、正しく実行されません。エラーを避けるために、CONNECT ステートメントを実行する前に、COMMIT または ROLLBACK の操作を行ってください。コミットメント制御を使用しないで実行する場合には、CONNECT はいつでも使用することができます。

DUW 接続管理では、CONNECT ステートメントの効果は以下の通りです。

- TO 文節または RESET 文節を指定した CONNECT ステートメントが正常に実行されると、以下のイベントが発生します。

- AS の名前が現行サーバー特殊レジスターに置かれます。
- エラーを戻したシステム・モジュールを識別する情報は、SQL 通信域 (SQLCA) の SQLERRP フィールドか、SQL 診断域の DB2_MODULE_DETECTING_ERROR フィールドに置かれます。
- 接続のタイプに関する情報も、SQLCA および SQL 診断域に置かれます。これらのフィールドにエンコードされるのは、以下の情報です。
 - アプリケーションが接続状態か未接続状態かの情報は、SQLCA の SQLERRD(5) または SQL 診断域の DB2_CONNECTION_STATE にあります。
 - リモート接続が保護された会話を使用しているか無保護の会話を使用しているかの情報は、SQLCA の SQLERRD(4) または SQL 診断域の DB2_CONNECTION_TYPE にあります。
 - 接続が常に読み取り専用であるか、常に更新可能であるか、あるいは更新可能と更新不可能が各作業単位間で変更できるかに関する情報は、SQLCA の SQLERRD(4) または SQL 診断域の DB2_CONNECTION_STATUS にあります。

SQLCA の SQLERRD フィールドに関する詳細、および SQL 診断域の接続情報については、『SQL プログラミング』のトピックを参照してください。

- AR が接続可能状態でないか、またはサーバー名がローカル・リレーショナル・データベース・ディレクトリにないために、TO 文節または RESET 文節を指定した CONNECT ステートメントが正常に実行されなかった場合には、AR の接続状態は変更されません。
- 現行接続になっている AS への接続は、エラーとなります。
- TO 文節または RESET 文節を持たない接続を使用して、現行接続の情報を入手することができます。この情報には、以下のような情報が含まれます。
 - 状況を戻したシステム・モジュールを識別する情報は、SQL 通信域 (SQLCA) の SQLERRP フィールドか、SQL 診断域の DB2_MODULE_DETECTING_ERROR フィールドに置かれます。
 - 他の状況情報は、SQLERRD(4) と SQLERRD(5) の内容およびそれに対応する SQL 診断域の情報について扱ったセクションの前の段落で説明されています。

アプリケーション・プロセスによって実行される最初の SQL ステートメントを CONNECT ステートメントにするのは適切な例です。ただし、プログラムに CONNECT ステートメントを組み込んでいるときに、そのプログラムが複数の AS に接続するのであれば、AS 名を動的に変更できます。アプリケーションを複数のシステムで実行する場合には、以下に示すように、ホスト変数を指定した CONNECT ステートメントを指定して、プログラムにリレーショナル・データベース名が渡されるようにすることができます。

CONNECT TO : host-variable

CONNECT ステートメントを使用しない場合、AS を変更するには、新規リレーショナル・データベース名を使用してプログラムを再コンパイルするだけでよいことになります。

以下の例では、アプリケーション・プログラム中に組み込まれた 2 つの形式 (1 および 2) の CONNECT ステートメントが示されています。

```
CRTSQLxxx PGM(SPIFFY/FIXTOTAL) COMMIT(*CHG) RDB(KC105)
```

```
PROC: FIXTOTAL;
EXEC SQL CONNECT TO KC105; 1
```

```
EXEC SQL
  SELECT * INTO :SERVICE
  FROM REPAIRTOT;
```

```
EXEC SQL COMMIT;
```

```
EXEC SQL CONNECT TO MPLS03 USER :USERID USING :PW; 2
```

```
EXEC SQL SELECT ...  
  
EXEC SQL COMMIT;  
  
END FIXTOTAL;
```

例 (2) は、USER/USING 形式の CONNECT ステートメントの使用法を示しています。この形式の CONNECT ステートメントがプログラムに組み込まれているときは、ホスト変数でユーザー ID とパスワードを指定しなければなりません。TCP/IP を使用している場合、適切なパラメーターを指定した **サーバ一許可項目の追加 (ADDSVRAUTE)** コマンドを使用してユーザー ID とパスワードを保管してあれば、接続時にセキュリティー・オブジェクトからユーザー ID とパスワードを抽出できます。

以下の例は、対話式 SQL の中で両方の CONNECT ステートメント形式を示しています。パスワードは単一引用符で囲まなければならないことに注意してください。

```
SQL ステートメントを入力して、実行キーを押してください。  
現在の接続相手はリレーショナル・データベース (RDB) KC105 である。  
CONNECT TO KC000 _____
```

```
COMMIT _____  
====> CONNECT TO MPLS03 USER JOE USING 'X47K' _____  
_____  
_____
```

関連概念:

61 ページの『リレーショナル・データベース・ディレクトリーの使用』

IBM i オペレーティング・システムは、リレーショナル・データベース・ディレクトリーを使用して、システム・アプリケーションでアクセスできるリレーショナル・データベース名を定義して、これらのリレーショナル・データベース名を対応するネットワーク・パラメーターに関連付けます。システムはこのディレクトリーを使用して、この接続でシステム・ネットワーク体系 (SNA) または IP が使用されるかどうかを指定することもできます。

分散リレーショナル・データベースに固有の SQL と SQL 呼び出し:

分散 DB2 for i アプリケーションのプリコンパイル処理中に、IBM i ライセンス・プログラムは、サーバーで実行される SQL パッケージを作成できます。

アプリケーションのコンパイル後、分散 SQL プログラムとパッケージは、クライアントおよびサーバーとして使用するシステムと互換性がなければなりません。プリコンパイル・プロセスの変更および SQL パッケージの追加に関する詳細については、「分散リレーショナル・データベース・プログラムの作成」のトピックで説明しています。

これらのトピックでは、分散リレーショナル・データベース・サポートとともに使用される SQL ステートメントの概要、および他のシステムとの共存に関する考慮事項がいくつか示されています。

関連概念:

141 ページの『分散リレーショナル・データベース・プログラムの作成』

SQL を使用してプログラムを作成するときは、ホスト・プログラムに SQL ステートメントを組み込むことができます。

SQL プログラミング

関連資料:

SQL リファレンス

分散リレーショナル・データベースのステートメント:

SQL 言語を使用して組み込まれるステートメントは、特に分散リレーショナル・データベースをサポートするためのものです。

たとえば、次のステートメントがあります。

- CONNECT
- SET CONNECTION
- RELEASE
- DISCONNECT
- DROP PACKAGE
- GRANT EXECUTE ON PACKAGE
- REVOKE EXECUTE ON PACKAGE

SQL CALL ステートメントはローカルで使用することができます。ただし、その本来の目的はリモート・システムからプロシージャを呼び出せるようにすることです。

関連概念:

123 ページの『分散リレーショナル・データベースへの接続』

分散リレーショナル・データベース・アプリケーションが分散されるのは、それが別のシステム上のリレーショナル・データベースへの接続可能性を備えているからです。

117 ページの『DRDA: 分散リレーショナル・データベース・オブジェクトに対する権限』

SQL GRANT および REVOKE ステートメント、または制御言語 (CL) であるオブジェクト権限認可 (GRTOBJAUT) およびオブジェクト権限取り消し (RVKOBJAUT) コマンドのいずれかを使用して、リレーショナル・データベース・オブジェクトに対するユーザー権限の認可および取り消しを行うことができます。

149 ページの『SQL パッケージの処理』

SQL パッケージとは、特に分散リレーショナル・データベース・アプリケーションによって使用される SQL オブジェクトです。これには、サーバー上のデータにアクセスする各 SQL ステートメントの制御構造が含まれます。

SQL CALL ステートメント (ストアード・プロシージャ):

SQL CALL ステートメントは、実際は分散リレーショナル・データベースに特有のものではありませんが、その主な価値は分散アプリケーションの論理および処理の中にあるので、ここでその説明を行います。

結果セットは、ストアード・プロシージャで、SQL SELECT ステートメントに関連した 1 つ以上の SQL カーソルをオープンすることによって生成できます。さらに、最高で 1 つの配列結果セットを戻すことができます。結果セットを戻すストアード・プロシージャの作成について詳しくは、『SQL 解説書』トピックの SET RESULT SETS および CREATE PROCEDURE ステートメントの説明を参照してください。

リモート・プロシージャ呼び出し (RPC) メカニズムが Open Software Foundation (OSF) 分散コンピューター環境 (DCE) で提供しているものとはほぼ類似した機能を、CALL ステートメントは DRDA 環境で提供しています。事実、リモート・リレーショナル・データベースでのプログラムに対する SQL CALL は、実際にはリモート・プロシージャ呼び出しです。この RPC のタイプにはいくつかの利点があります。たとえば、インターフェース定義のコンパイル、またはスタブ・プログラムの作成は不要です。

SQL CALL (あるいは手法としてストアード・プロシージャとも呼ばれる)を使用するのは、以下の理由によります。

- クライアントとサーバー間のメッセージ・フローの数を減らし、与えられている機能を実行するため。SQL 操作のセットが実行される場合、サーバーのプログラムにステートメントおよび相互接続論理を含めたほうがより効率的です。
- リモート・ロケーションでローカル・データベース操作を行えるようにするため。
- SQL を使用して非データベース操作 (たとえば、メッセージの送信またはデータ待ち行列操作の実行) を行うため。

注: データベース操作の場合と異なり、これらの操作は、システムによるコミットメント制御で保護されません。

- リモート・システム上でシステム API にアクセスするため。

ストアード・プロシージャとアプリケーション・プログラムは、同じ活動化グループ内で稼働することも、別個の活動化グループ内で稼働することもできます。クライアント側のアプリケーション・プログラムとサーバー側のストアード・プロシージャとの間に一貫性を持たせるため、ACTGRP(*CALLER) を指定してストアード・プロシージャをコンパイルすることをお勧めします。ストアード・プロシージャが結果セットを戻すように設計されている場合は、それを *NEW 活動化グループで実行するように作成すべきではありません。そのようにすると、プロシージャが呼び出し元に戻るときに、結果セットに関連したカーソルが早くにクローズし、活動化グループは破棄される可能性があります。

照会メッセージを発行するストアード・プロシージャが呼び出されると、メッセージが QSYSOPR メッセージ待ち行列に送られます。ストアード・プロシージャは照会メッセージへの応答を待ちます。ストアード・プロシージャが照会メッセージに応答するようにするには、**応答リスト項目追加 (ADDRPYLE)** コマンドを使用し、ストアード・プロシージャ内の **ジョブ変更 (CHGJOB)** コマンドの INQMSGRPY パラメーターに *SYSRPYL を指定します。

ストアード・プロシージャおよびアプリケーション・プログラムが別々のコミットメント定義の下で実行されるときには、アプリケーション・プログラム中の COMMIT ステートメントおよび ROLLBACK ステートメントは、自らのコミットメント定義にのみ影響を及ぼします。他の手段を用いて、ストアード・プロシージャで変更をコミットしなければなりません。

関連資料:

SQL リファレンス

返信リスト項目追加 (ADDRPYLE) コマンド

ジョブ変更 (CHGJOB) コマンド

146 ページの『テストとデバッグ』

分散 SQL プログラムのテストとデバッグは、ローカル SQL プログラムのテストとデバッグに類似していますが、プロセスの面では、一部異なっているところがあります。

DB2 for i呼び出しの考慮事項:

一部のプラットフォームで呼び出される C で書き込まれたストアード・プロシージャは、パラメーターとして argc および argv を使用できません (つまり、main() タイプになることができません)。この点は IBM i ストアード・プロシージャとは異なります。そこでは argc および argv を使用しなければなりません。

DB2 プラットフォームのストアード・プロシージャの例については、\SQLLIB\SAMPLES (または /sqllib/samples) サブディレクトリーを参照してください。C サブディレクトリーで outsrv.sqc および outcli.sqc を見つけてください。

IBM i オペレーティング・システムで呼び出される DB2 ストアード・プロシージャでは、プロシージャ名が大文字になっているか確認してください。システムでは現在プロシージャ名を大文字に変換しています。つまり、DB2 サーバー上のプロシージャには、同一のプロシージャで小文字であるものはありません。IBM i ストアード・プロシージャの場合、プロシージャ名は大文字です。

IBM i オペレーティング・システム上のストアード・プロシージャは、同じ活動化グループの中で呼び出しプログラム (ストアード・プロシージャを作成するための適切な方法) として実行するために作成される場合、COMMIT ステートメントを含めることができません。DB2 では、ストアード・プロシージャに COMMIT ステートメントを含めることができますが、コミットが起こる DB2 for i の部分については認知できないということをアプリケーション設計者は理解する必要があります。

DB2 for i 共存:

SQL 言語を使用して、分散リレーショナル・データベースのためのプログラムを作成および保守する場合は、分散リレーショナル・データベース・ネットワーク内の他のシステムについて考慮する必要があります。

作成および保守するプログラムは、以下の項目との間で互換性が必要になる可能性があります。

- 他の IBM i 製品
- 以前の IBM i オペレーティング・システムのリリース
- IBM i 以外のシステム

分散 SQL プログラムの中の SQL ステートメントはサーバー上で実行されるということに注意してください。プログラムがクライアント上で実行される場合でも、SQL ステートメントはサーバー上で実行される SQL パッケージの中に入っています。それらのステートメントはサーバーによってサポートされ、サーバー上に存在しているコレクション、テーブル、およびビューとの間で互換性がなければなりません。また、クライアント側でプログラムを実行するユーザーは、サーバー上の SQL パッケージおよびその他の SQL オブジェクトに対して許可されていなければなりません。

CRTSQLxxx コマンドを使用し、サーバーに対してリレーショナル・データベース名 (RDB パラメーター) を指定して、プログラムを再度作成することによって、非分散組み込み SQL プログラムを分散組み込み SQL プログラムに変換することができます。この場合には、DB2 for i の分散リレーショナル・データベース・サポートを使用してプログラムを再度コンパイルし、サーバー上で必要とされる SQL パッケージを作成します。

IBM i を実行しないサーバー・システムで実行される DB2 for i プログラムを作成できます。これらの他のプラットフォームは、多少なりとも SQL 機能をサポートします。DB2 for i クライアントでサポートされていないステートメントは、サーバーがその機能をサポートするときには、サーバー上で使用およびコンパイルすることができます。このトピック・コレクションで説明しているレベルのサポートを提供するのは、IBM i サーバー・システム上で実行するために作成した SQL プログラムだけです。他のシステムが提供するレベルの機能を判別したい場合には、他のシステムのサポート資料を参照してください。

DRDA 作業単位の終了:

非コミット作業で SQL プログラムを終了する場合には注意が必要です。プログラムが非コミット作業で終了すると、リレーショナル・データベースへの接続は活動状態のままになります。

ただし、システム命名の活動化グループで実行されるプログラムを組み込んでいる場合は、システムはプログラムの終了時に自動コミットを行います。

この状態は他のシステムの場合とは異なります。つまり、IBM i オペレーティング・システムでは、COMMIT および ROLLBACK は、コマンド行から、または CL プログラムでコマンドとして使用することができるためです。ただし、上記のシナリオではその状態についてあらかじめ計画していた場合を除き、次の SQL プログラムの実行時に予期しない結果を招く場合があります。たとえば、次に対話式 SQL (STRSQL コマンド) を実行した場合、非コミット作業を持つ前のサーバーへ接続されている状態に対話式セッションが開始されます。別の例として、上記のシナリオの後で暗黙接続を行う 2 番目の SQL プログラムを開始した場合、最後に使用された AS 上で該当するパッケージを見つけて使用しようとする試みが行われます。このサーバーは予定していたサーバーではない場合があります。このような不測の事態を回避するために、アプリケーション・プログラムを終了する場合は、必ずその前に最後の作業単位をコミットするかまたはロールバックするようにしてください。

ストアド・プロシージャ、ユーザー定義関数、およびコミットメント制御:

DRDA 接続に対してコミットメント制御がアクティブになっていない状態 (COMMIT(*NONE)) で対話式 SQL などのアプリケーションが稼働している場合は、呼び出されたストアド・プロシージャやユーザー定義関数 (UDF) が、IBM i オペレーティング・システム上でコミットメント制御を開始する可能性があります。

これは、クライアントとサーバーの間でコミットメント制御の不一致を生み、それによってアプリケーション終了時に更新がコミットされなくなる場合があります。

このような状態は避けなければなりません。とはいえ、もしこのような状態が発生してしまった場合には、1 つの解決方法として、コミットメント制御下で稼働しているストアド・プロシージャや UDF に対し、そのデータベース更新すべてを明示的にコミットすることができます。また、これが行われない場合は、サーバーが切断プロセス中に保留中の更新を検出し、その保留中の作業を自動的にコミットします。

コード化文字セット識別子:

国別の言語サポートには、最小限の文字セットの適正な処理が必要です。

文字情報管理のためのシステム間サポートは、IBM 文字データ表現体系 (CDRA) によって提供されます。CDRA では、文字を表すために使用されるコード・ポイントを識別し、それらの意味を保持するための必要に応じて、それらのコード (文字データ) を変換するためのコード化文字セット識別コード (CCSID) を定義します。

CDRA などのアーキテクチャーおよび対応する変換プロトコルの使用が重要になるのは、次のような場合です。

- 複数の国別言語バージョンが IBM i オペレーティング・システムにインストールされている場合。
- 複数の IBM i オペレーティング・システムが、異なる各国別言語の基本バージョンを使用している異なる国のシステム間でデータを共有する場合。
- IBM i オペレーティング・システムおよび IBM i 以外のシステムが、異なる各国別言語の基本バージョンを使用している異なる国のシステム間でデータを共有する場合。

タグ付けは、コード化表示文字に意味を割り当てるための基本的な手段です。タグは、データ・オブジェクトに関連するデータ構造の中にある場合 (明示タグ付け) もあれば、ジョブまたはシステム事態など、オブジェクトから受け継がれる場合 (暗黙タグ付け) もあります。

DB2 for i は、CCSID 文字の列にタグ付けをします。CCSID は、コード化体系識別コード、文字セット識別コード、コード・ページ識別コード、および使用されるコード化表示文字の表示を固有に識別する追加のコード化関連情報から成る特定のセットを識別する 16 ビットの数です。アプリケーションの実行に際して、データは別のシステムに送信される時には変換されません。データはタグ付けされたまま CCSID とともに送信されます。受信側ジョブでは、そのデータのタグ付け方式が異なる場合、自動的にデータを独自の CCSID に変換します。

CDRA では、以下に挙げる範囲の値を CCSID に定義しています。

00000 次の階層の CCSID を使用する。

00001 から 28671

IBM 登録 CCSID

28672 から 65533

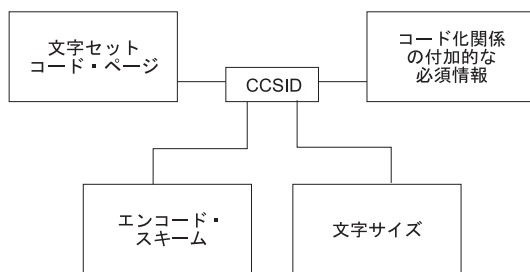
予約済み

65534 下位の階層の CCSID を参照する

65535 変換は行われぬ

CDRA CCSID の完全なリストについては、*Character Data Representation Architecture - Level 1, Registry* を参照してください。

次の図には、CCSID の各部分が示してあります。



RBAL1508-0

図 25. コード化文字セット識別子 (CCSID)

関連概念:

IBM i グローバリゼーション

SQL プログラミング

11 ページの『DRDA および CDRA サポート』

分散リレーショナル・データベースは、別の種類のコンピューター間で共有できるだけでなく、それぞれのコンピューターが別の国や地域にあっても構いません。

関連資料:

SQL リファレンス

IBM i サポート:

ジョブ変更 (CHGJOB) コマンドを使用することによって、IBM i ジョブの CCSID を変更することができます。

CCSID がこのようにして指定されない場合には、ジョブ CCSID は、ユーザー・プロファイルの CCSID 属性から得られます。CCSID が指定されていない場合には、システムはそれを QCCSID システム値から取得します。この QCCSID 値は最初は 65535 に設定されています。システムが異種システムとの分散リレーショナル・データベースにある場合は、CCSID 65535 を使用できない場合があります。

クライアントとサーバーの間を流れる制御情報は、すべて CCSID 500 (DRDA 標準) です。これは、コレクション名、テーブル名、および記述テキストなどの情報です。制御情報に可変レコード文字を使用すると、これらの名前が変換されることになり、パフォーマンスに影響を与える可能性があります。パッケージ名も CCSID 500 で送られます。パッケージ名の中で可変レコード文字を使用すると、パッケージ名が変換されることになります。つまり、そのパッケージが実行時に見つからないことになります。

ジョブが開始されてから、ジョブ変更 (CHGJOB) コマンドを使用することによって、ジョブの CCSID を変更できます。そのためには、以下のようになります。

1. ジョブ処理 (WRKJOB) コマンドを入力して、「ジョブの処理」画面を表示してください。
2. オプション 2 (ジョブ定義属性の表示) を選択してください。これで現行 CCSID 値が見つかるので、後でジョブを元の CCSID 値にセットし直すことができます。
3. 新しい CCSID 値を指定して、ジョブ変更 (CHGJOB) コマンド>を入力します。

新しい CCSID 値は即時にジョブに反映されます。ただし、変更するジョブ CCSID が AR ジョブである場合には、次の CONNECT までは新しい CCSID が実行中の作業に影響することはありません。

重要: サーバー・ジョブの CCSID を変更した場合には、結果は予想することができません。

CCSID がソース物理ファイルの作成 (CRTSRCPF) または物理ファイルの作成 (CRTPF) コマンドでソース・ファイルに明示的に指定されない場合には、ソース・ファイルはジョブの CCSID でタグ付けされます。CCSID がデータ記述仕様 (DDS)、対話式データ定義ユーティリティ (IDDU)、または CREATE TABLE SQL ステートメントで明示的に指定されない場合には、外部記述データベース・ファイルおよびテーブルはジョブの CCSID でタグ付けされます。

ソース・ファイルおよび外部記述ファイルでは、ジョブ CCSID が 65535 であれば、オペレーティング・システムの言語に基づいたデフォルトの CCSID が使用されます。プログラム記述ファイルは、CCSID 65535 でタグ付けされます。ビューは、対応するテーブルの CCSID または列レベル・タグでタグ付けされます。複数のテーブルにまたがって定義されているビューの場合には、列レベルでタグ付けされ、基礎を成している列のタグが想定されます。ビューは、CCSID で明示的にタグ付けすることはできません。CCSID が等しくなく、しかもどちらの CCSID も 65535 に等しくない場合には、システムはジョブとテーブルの間でデータを自動的に変換します。

タグ付けされたテーブルの CCSID を変更すると、そのテーブルは列レベルでタグ付けすることも、ビューを定義することもできません。タグ付けされた CCSID を変更するには、物理ファイルの変更 (CHGPF) コマンドを使用します。列レベルのタグ付けがあるテーブルを変更するには、そのテーブルを再度作成し、ファイル・コピー (CPYF) コマンドの FMT(*MAP) を使用して、データを新しいテーブルにコピーしなければなりません。テーブルに 1 つ以上のビューが定義されている場合、テーブルを変更するには以下のステップを行います。

1. ビューおよびテーブルをそれらのアクセス・パスとともに保管する。
2. ビューを削除する。
3. テーブルを変更する。
4. 作成したテーブルにビューおよびそのアクセス・パスを復元する。

DB2 for i に移行されるソース・ファイルおよび外部記述ファイルで、タグ付けされていないもの、または CCSID 65535 でタグ付けされているものは、インストールされているオペレーティング・システムの言語に基づいたデフォルトの CCSID でタグ付けされます。このようなファイルには、新しいリリースのインストール時にシステムにあるファイル、および DB2 for i に復元されるファイルが含まれます。

AR と AS の間で送信されるデータは、すべて変換されずに送信されます。さらに、CCSID も送信されます。受信側ジョブでは、そのデータのタグ付け方式が異なる場合、自動的にデータを独自の CCSID に変換します。たとえば、販売店システム KC105 で実行される次のアプリケーションを考慮します。

注: コードの例を使用することは、443 ページの『コードに関するライセンス情報および特記事項』の条件に同意したものとされます。

```
CRTSQLxxx PGM(PARTS1) COMMIT(*CHG) RDB(KC000)
```

```
PROC: PARTS1;
  .
  .
EXEC SQL
  SELECT * INTO :PARTAVAIL
         FROM INVENTORY
         WHERE ITEM = :PARTNO;
  .
  .
END PARTS1;
```

前述の例では、ローカル・システム (KC105) は QCCSID システム値が CCSID 37 に設定されています。リモート地域センター (KC000) では CCSID 937 を使用し、そのテーブルはすべて CCSID 937 でタグ付けされています。CCSID 処理は次のようにして行われます。

- KC105 システムが入力ホスト変数 (:PARTNO) を CCSID 37 で送信します。(ジョブの CCSID がホスト変数に適さない場合には、`DECLARE VARIABLE SQL` ステートメントが使用できます。)
- KC000 システムが :PARTNO を CCSID 937 に変換し、必要なデータを選択し、そのデータを CCSID 937 で KC105 に送り返します。
- KC105 では、そのデータを入手すると、CCSID 37 に変換し、ローカル使用に備えて :PARTAVAIL に入れます。

関連概念:

385 ページの『ユーザー FAQ』

他の特定の IBM 製品を処理するときには、以下の状況を考慮する必要があります。

関連資料:

ジョブ変更 (CHGJOB) コマンド

物理ファイル変更 (CHGPF) コマンド

ファイル・コピー (CPYF) コマンド

物理ファイル作成 (CRTPF) コマンド

ソース物理ファイル作成 (CRTSRCPF) コマンド

ジョブ処理 (WRKJOB) コマンド

他の DRDA データ変換:

リモート・システムで処理を行っている場合には、時として、プログラムがデータを 1 つのシステムから変換して、他方のシステムでも使用できるようにする必要があります。IBM i オペレーティング・システム上の Distributed Relational Database Architecture (DRDA) サポートは、DRDA サポートを使用する他の各システム間でデータを自動的に変換します。

DB2 for i クライアントがサーバーに接続する際に、そのタイプを識別する情報を送信します。同様にして、サーバーでは、その処理装置タイプ (たとえば、z/OS ホストまたは IBM i) を識別する情報をシステムに送り返します。次いで、2 つのシステムは、この接続で定義されているように、相互間でデータを自動的に変換します。つまり、システム間の体系的な違いについてはプログラムで考慮する必要はないことを意味します。

DRDA サポートを使用する IBM システム間でのデータ変換には、以下のようなデータ・タイプが含まれます。

- 浮動小数点表示
- ゾーン 10 進数表示
- バイト反転
- 混合データ・タイプ
- IBM i 固有のデータ・タイプ
 - DBCS 専用
 - DBCS 択一
 - 精度および位取り付き整数

データ・タイプと、必要な場合にマップされるデータ・タイプのリストを以下に示します。

表6. マップされるデータ・タイプ

ソース・タイプ	マップされるタイプ
XML スtring内部エンコード	BLOB
XML スtring外部エンコード	CLOB または DBCLOB
固定バイナリー	char(n) for BIT
可変バイナリー	varchar(n) for BIT
10 進浮動小数点	8 バイト浮動小数点

分散リレーショナル・データベース・プログラムの作成

SQL を使用してプログラムを作成するときは、ホスト・プログラムに SQL ステートメントを組み込むことができます。

ホスト・プログラム とは、次に挙げるホスト言語 のいずれか 1 つで書かれ、SQL ステートメントが含まれているプログラムのことです。すなわち、PL/I、PRPQ、ILE C、COBOL/400、ILE COBOL、FORTRAN/400、RPG/400、または ILE RPG プログラミング言語です。ホスト・プログラムの中では、ホスト変数と呼ばれる変数を使用します。これらは、SQL ステートメントの中で使用され、ホスト・プログラムに対して識別可能な変数のことです。RPG では、フィールド名 と呼ばれています。これらは FORTRAN、PL/I、および C では変数 といい、COBOL ではデータ項目 と呼ばれます。

分散 DB2 for i プログラムは、分散ではない DB2 for i プログラムの場合と似た方法でコーディングすることができます。ホスト言語を使用して、SQL ステートメントをホスト変数とともに組み込みます。また、分散でない DB2 for i プログラムの場合と同じように、分散 DB2 for i プログラムを特定のプロセスで作成します。

ただし、分散 DB2 for i プログラムでは、データにアクセスするために SQL パッケージもサーバー上で作成しなければなりません。

このトピック・コレクションでは、分散 DB2 for i プログラムの場合の違いを概説し、上記のプロセスのステップを説明します。

SQL ステートメントを含むプログラムのプリコンパイル:

組み込み SQL ステートメントが入っているアプリケーション・プログラムは、その実行に先立って、プリコンパイルとコンパイルが必要です。そのようなプログラムのプリコンパイルは、SQL コンパイラーによって行われます。

SQL プリコンパイラーは、アプリケーション・プログラム・ソースの各ステートメントを走査し、以下のことを行います。

- SQL ステートメント、およびホスト変数名の定義を検索する
- 各 SQL ステートメントが有効で、構文エラーがないかどうかを検査する
- データベースの中の記述を使用して、SQL ステートメントの妥当性検査をする
- 各 SQL ステートメントをホスト言語でのコンパイルができるように準備する
- 各プリコンパイル済み SQL ステートメントについての情報を作成する

プリコンパイラーへの主要な入力は、アプリケーション・プログラミング・ステートメントと組み込み SQL ステートメントです。SQL プリコンパイラーでは、ホスト言語ステートメントが構文上正しいことを前提としています。ホスト言語ステートメントが構文上正しくない場合には、プリコンパイラーは SQL ステートメントおよびホスト変数宣言を正しく識別できないことがあります。

SQL プリコンパイル処理では、リストと一時ソース・ファイル・メンバーが作られます。また、プリコンパイラー・コマンドの OPTION および RDB パラメーターに指定される値によっては、SQL パッケージが作られます。

関連資料:

145 ページの『アプリケーション・プログラムのコンパイル』
DB2 for i プリコンパイラーは、プログラムのプリコンパイルが正常に終了すると、*NOGEN プリコンパイラー・オプションが指定されていなければ、自動的にホスト言語コンパイラーを呼びます。

リスト作成:

出力リストは、CRTSQLxxx コマンドの PRTFILE パラメーターで指定された印刷装置ファイルに送られます。

以下の項目が印刷装置ファイルに書き込まれます。

- プリコンパイラー・オプション

これは、CRTSQLxxx コマンドで指定されたすべてのオプション、およびソース・メンバーが最後に変更された日付のリストです。

- プリコンパイラー・ソース

この出力が作成されるのは、非 ILE プリコンパイラーに *SOURCE オプションが、または ILE プリコンパイラーに OUTPUT(*PRINT) パラメーターが使用されている場合です。各プリコンパイラー・ソース・ステートメントが、プリコンパイラーによって割り当てられたそのレコード番号、原始ステートメント入力キューティリティー (SEU) の使用時に表示される順序番号 (SEQNBR)、およびレコードが最後に変更された日付とともに示されます。

- プリコンパイラー相互参照

この出力が作成されるのは、OPTION パラメーターに *XREF が指定された場合です。ホスト変数または SQL エンティティ (テーブルおよび列など) の名前、その名前が定義されているレコード番号、その名前が定義されていること、および名前が現れるレコード番号が示されます。

- プリコンパイラー診断リスト

この出力は診断メッセージを提供し、エラー状態のステートメントのプリコンパイラー・レコード番号を示します。

一時ソース・ファイル・メンバー:

プリコンパイラーが処理したソース・ステートメントは、QTEMP ライブラリーの QSQLTEMP (CRTSQLRPGI を使用して作成されたプログラムでは QTEMP ライブラリーの QSQLTEMP1) に書き込まれます。

プリコンパイラーによって変更されたソース・コードでは、SQL ステートメントは、SQL インターフェース・モジュール QSROUTE、QSQLOPEN、QSQCLOSE、および QSQCMTIT に対する注記および呼び出しに変換されています。一時ソース・ファイル・メンバーの名前は、CRTSQLxxx の PGM パラメーターに指定されている名前と同じです。このメンバーは、コンパイラーに対する入力として使用される前には変更できません。

コンパイルを後にしたい場合には、QSQLTEMP または QSQLTEMP1 をプリコンパイルの後で永続ライブラリーに移すことができます。一時ソース・ファイル・メンバーのレコードを変更する場合には、コンパイルを後に試みると、正常に行われません。

SQL パッケージの作成:

CRTSQLxxx コマンドのコンパイル時に、プリコンパイル・プロセスの一部として、SQL パッケージと呼ばれるオブジェクトを作成することができます。

関連概念:

149 ページの『SQL パッケージの処理』

SQL パッケージとは、特に分散リレーショナル・データベース・アプリケーションによって使用される SQL オブジェクトです。これには、サーバー上のデータにアクセスする各 SQL ステートメントの制御構造が含まれます。

関連資料:

145 ページの『アプリケーション・プログラムのコンパイル』

DB2 for i プリコンパイラーは、プログラムのプリコンパイルが正常に終了すると、*NOGEN プリコンパイラー・オプションが指定されていないければ、自動的にホスト言語コンパイラーを呼びます。

145 ページの『アプリケーションのバインド』

アプリケーション・プログラムで実行できるようにするには、その前にプログラムと参照されるテーブルおよびビューとの間の関係が確立されていなければなりません。この処理がバインドと呼ばれるものです。

プリコンパイラー・コマンド:

IBM DB2 Query Manager and SQL Development Kit for i ライセンス・プログラムには、各ホスト言語に対して 1 つずつ、7 つのプリコンパイラーがあります。

ホスト言語	コマンド
PL/I PRPQ	CRTSQLPLI
ILE C 言語	CRTSQLCI
COBOL/400 言語	CRTSQLCBL

ホスト言語	コマンド
ILE COBOL 言語	CRTSQLCBLI
FORTRAN/400 言語	CRTSQLFTN
RPG III (RPG/400 言語の一部)	CRTSQLRPG
ILE RPG 言語	CRTSQLRPGI

各言語ごとに別のコマンドが存在しているので、各言語はその言語だけに適用されるパラメーターを持つことができます。たとえば、オプション ***APOST** および ***QUOTE** は **COBOL** に固有のもので、これらは他の言語用のコマンドには含まれていません。プリコンパイラーは、**SQL** プリコンパイラー・コマンドの1つが呼び出すときに指定するパラメーターによって制御されます。これらのパラメーターは、入力の処理方法および出力の提示方法を指定します。

プログラムのプリコンパイルには、プログラム・ソース・ステートメントを含んでいるメンバーの名前を **CRTSQlxxx** コマンドの **PGM** パラメーター (非 **ILE** プリコンパイラーの場合)、または **OBJ** パラメーター (**ILE** プリコンパイラーの場合) として指定する以外は、何も指定する必要がありません。**SQL** は、すべてのプリコンパイラー・パラメーターに対してデフォルト値を割り当てます (ただし、値を明示的に指定すれば、デフォルト値を指定変更できます)。

以下のリストで、分散リレーショナル・データベースをサポートするために使用される、すべての **CRTSQlxxx** コマンドに共通なパラメーターを簡単に説明します。

RDB

SQL パッケージ・オプションを作成するリレーショナル・データベースの名前を指定します。 ***NONE** を指定すると、プログラムまたはモジュールは非分散オブジェクトになり、**構造化照会言語パッケージの作成 (CRTSQlPKG)** コマンドは使用できなくなります。リレーショナル・データベースの名前は、ローカル・データベースの名前とすることができます。

RDBCNMTH

CONNECT ステートメントの働き方のタイプ、すなわち、リモート作業単位 (**RWU**) または分散作業単位 (**DUW**) の別を指定します。

SQLPKG

SQL パッケージの名前とライブラリーを指定します。

USER

会話の開始時にリモート・システムに送られるユーザー名を指定します。このパラメーターは、会話プリコンパイル・プロセスの一部として開始される場合だけ使用します。

PASSWORD

会話を開始するときに、リモート・システムで使用されるパスワードを指定します。このパラメーターは、会話プリコンパイル・プロセスの一部として開始される場合だけ使用します。

REPLACE

プリコンパイルの一部として作成されるオブジェクトが、既存のオブジェクトを置き換えられるようにする場合に指定します。

以下の例では、**INVENT** という名前の **COBOL** プログラムを作成し、それを **SPIFFY** という名前のライブラリーに保管します。 **SQL** 命名規則が選択され、指定されたテーブルから選択された行は、すべてが回復単位の終了までロックされます。プログラムと同じ名前の **SQL** パッケージが、**KC000** という名前のリモート・リレーショナル・データベースに作成されます。

```
CRTSQLCBL PGM(SPIFFY/INVENT) OPTION(*SRC *XREF *SQL)
          COMMIT(*ALL) RDB(KC000)
```

関連資料:

構造化照会言語パッケージの作成 (CRTSQLPKG) コマンド

アプリケーション・プログラムのコンパイル:

DB2 for i プリコンパイラーは、プログラムのプリコンパイルが正常に終了すると、*NOGEN プリコンパイラー・オプションが指定されていない場合は、自動的にホスト言語コンパイラーを呼びます。

このコンパイラー・コマンドは、プログラム名、ソース・ファイル名、プリコンパイラー作成のソース・メンバー名、テキスト、およびユーザー・プロファイルを指定して、実行されます。その他のパラメーターも、ホスト言語に応じて、コンパイラーに渡されます。

関連概念:

142 ページの『SQL ステートメントを含むプログラムのプリコンパイル』

組み込み SQL ステートメントが入っているアプリケーション・プログラムは、その実行に先立って、プリコンパイルとコンパイルが必要です。そのようなプログラムのプリコンパイルは、SQL コンパイラーによって行われます。

組み込み SQL プログラミング

関連資料:

143 ページの『SQL パッケージの作成』

CRTSQLxxx コマンドのコンパイル時に、プリコンパイル・プロセスの一部として、SQL パッケージと呼ばれるオブジェクトを作成することができます。

アプリケーションのバインド:

アプリケーション・プログラムで実行できるようにするには、その前にプログラムと参照されるテーブルおよびビューとの間の関係が確立されていなければなりません。この処理がバインドと呼ばれるものです。

バインドの結果がアクセス・プランです。アクセス・プランは、各 SQL 要求を満たすのに必要なアクションを記述する制御構造です。アクセス・プランには、プログラムおよびプログラムで使用するデータに関する情報が入ります。分散リレーショナル・データベース作業では、アクセス・プランは SQL パッケージの中に保管され、SQL パッケージとともにシステムによって管理されます。

コンパイルが成功して、プログラムまたはサービス・プログラムのオブジェクトが作成されると、SQL は自動的にアクセス・プランのバインドと作成を試みます。コンパイルが正常に行われなかった場合、またはコンパイルの結果がモジュール・オブジェクトでない場合は、アクセス・プランは作成されません。実行時に、アクセス・プランが有効でないこと、またはパフォーマンスを向上させるような変更 (たとえば、索引の追加) がデータベースに加えられたことをデータベース・マネージャーが検出した場合には、新規アクセス・プランが自動的に作成されます。サーバーが IBM i 製品ではない場合には、構造化照会言語パッケージの作成 (CRTSQLPKG) コマンドを使用して、バインドをやり直さなければなりません。バインドにより以下のタスクが実行されます。

- データベースの中の記述を使用して、SQL ステートメントの妥当性を再検査する。

バインド・プロセス中に、SQL ステートメントは、テーブル、ビューおよび列名が有効かどうかを検査されます。参照されたテーブルまたはビューがプリコンパイル時またはコンパイル時に存在していない場合には、妥当性検査は実行時に行われます。テーブルまたはビューが実行時に存在していない場合には、負の SQLCODE が戻されます。

- プログラムで処理したいデータにアクセスするために必要なアクセス・パスを選択する。

アクセス・パスの選択にあたっては、SQL がアクセス・プランを作成する際に、索引、テーブル・サイズ、およびその他の要因が考慮されます。バインド処理では、データにアクセスするのに使用できるすべての索引を考慮し、データへの経路の選択にあたって使用する索引を (ある場合) 決定します。

- アクセス・プランを作成しようと試みる。

すべての SQL ステートメントが有効である場合には、バインド処理はアクセス・プランを作成し、それをプログラムの中に保管します。

プログラムがアクセスするテーブルまたはビューの特性が変更された場合には、アクセス・プランは有効ではなくなる場合があります。有効でないアクセス・プランを使用しようとすると、システムは自動的にアクセス・プランの再作成を試みます。アクセス・プランが再作成できない場合には、負の SQLCODE が戻されます。この場合には、プログラムの SQL ステートメントを変更し、CRTSQLxxx コマンドを再度発行して、状況を修正しなければならないことがあります。

たとえば、プログラムの中に TABLEA 中の COLUMN A を参照する SQL ステートメントが入っていて、しかもユーザーが TABLEA を削除および再作成したために COLUMN A が存在していない場合には、プログラムを呼び出したときに、COLUMN A が存在していないために、自動再作成は正常に行われません。プログラムのソースを変更し、CRTSQLxxx コマンドを再度発行しなければなりません。

関連概念:

149 ページの『SQL パッケージの処理』

SQL パッケージとは、特に分散リレーショナル・データベース・アプリケーションによって使用される SQL オブジェクトです。これには、サーバー上のデータにアクセスする各 SQL ステートメントの制御構造が含まれます。

関連資料:

143 ページの『SQL パッケージの作成』

CRTSQLxxx コマンドのコンパイル時に、プリコンパイル・プロセスの一部として、SQL パッケージと呼ばれるオブジェクトを作成することができます。

構造化照会言語パッケージの作成 (CRTSQLPKG) コマンド

テストとデバッグ:

分散 SQL プログラムのテストとデバッグは、ローカル SQL プログラムのテストとデバッグに類似していますが、プロセスの面では、一部異なっているところがあります。

プログラムの再コンパイル、プログラムに対する入力パラメーターの変更、プログラムのソースの小さな修正を行うことによって、リレーショナル・データベースが容易に変更できるようにアプリケーションがコーディングされている場合には、ほとんどのテストは単一のシステムを使用して実施することができます。

プログラムはローカル・データでテストした後で、分散リレーショナル・データベース・ネットワーク上での最終的なテストを行えるようにします。アプリケーションをリモート接続でテストするときにサーバーとして使用するシステム上で、アプリケーションをローカルにテストすれば、テストを分散環境に移すときにプログラムを移すだけですみます。

分散 SQL プログラムのデバッグは、ローカル SQL プログラムのデバッグと同じ技法を使用します。デバッグの開始 (STRDBG) コマンドを使用して、デバッガーを始動させ、アプリケーションをデバッグ・モードにします。ブレークポイントの追加、ステートメントのトレース、および変数の内容の表示を行うことができます。

ただし、分散 SQL プログラムをデバッグする場合には、UPDPROD パラメーターに *YES の値を指定しなければなりません。その理由は、IBM i 分散リレーショナル・データベース・サポートではライブラリー QSYS の中のファイルを使用して、QSYS は実動ライブラリーだからです。これにより、実動ライブラリーの中のデータをクライアント上で変更することができます。クライアント上でデバッグの開始 (STRDBG) コマンドを発行しても、デバッグ・モードになるのはクライアント・ジョブだけであり、サーバー上でのデータ操作能力が変化することはありません。

クライアント上でのデバッグ・モード中は、実行された各 SQL ステートメントごとに、通知メッセージがジョブ・ログに入られます。これらのメッセージは、各 SQL ステートメントの結果について情報を与えるものです。分散リレーショナル・データベースでの SQL 戻りコードのリストおよびエラー・メッセージのリストは、『トラブルシューティング』のトピックに挙げられています。

システムが SQL ステートメントの処理効率を最大にする方法を示す通知メッセージも、デバッグ・モードの結果として発行されます。効率の最大化はサーバー側で行われるので、これらのタイプのメッセージはクライアント・ジョブ・ログの中には現れません。この情報を得るためには、サーバー・ジョブがデバッグ・モードでなければなりません。

TCP/IP を使用している場合、システムでデバッグ・モードを開始する比較的簡単な方法は、QRWOPTIONS データ域を使用する方法です。しかし、この機構では、特定のプログラムを指定してデバッグすることはできません。セットアップについて詳しくは、QRWOPTIONS データ域の使用法を参照してください。データ域は、デバッグの開始だけでなく、ジョブ・トレースの開始、ジョブ・ログの要求、ジョブ出力の表示などにも使用できます。IBM i クライアント上で QRWOPTIONS セットアップを行って、IBM i プラットフォーム上ではこのオプションを隠すこともできます。

クライアントおよびサーバーが IBM i 製品であり、それらが APPC に接続している場合には、リモート・コマンド投入 (SBMRMTCMD) コマンドを使用して、サーバー・ジョブでのデバッグ・モードを開始できます。「DDM ファイルのセットアップ」のトピックで説明しているように、DDM ファイルを作成してください。DDM ファイルの中の通信情報は、アクセスされるリレーショナル・データベースに関するリレーショナル・データベース・ディレクトリー項目の中の情報に一致しなければなりません。そこで、次のコマンドを入力してください。

```
SBMRMTCMD CMD('STRDBG UPDPROD(*YES)') DDMFILE(ddmfile name)
```

(SBMRMTCMD) コマンドは、サーバーがまだ存在していない場合には、サーバー・ジョブを始動させ、そのジョブでデバッグ・モードを開始させます。リレーショナル・データベースの活動をモニターするいずれかの方法を使用して、サーバー・ジョブ・ログを調べ、ジョブを見つけてください。

サーバー・ジョブをデバッグ・モードにする以下の方法は、どのクライアントおよび DB2 for i サーバーでも使用できますが、特定の制限があります。この方法は、アプリケーションがセットアップをするために接続を行った後で、一時停止できることを前提としています。また、この方法では、トレースあるいはデバッグする事柄が、接続が確立された後に発生することを想定しています。

- クライアントにサインオンして、サーバー・ジョブを検索してください。
- 下に示すように、サービス・ジョブの開始 (STRSRVJOB) コマンドを対話式ジョブ (サーバー・ジョブを見つけるために使用するジョブ) から出してください。

```
STRSRVJOB (job-number/user-ID/job-name)
```

(STRSRVJOB) コマンドのジョブ名はサーバー・ジョブの名前です。このコマンドを出せば、サーバー・ジョブに影響する特定のコマンドを対話式ジョブから出すことができます。そのようなコマンドの 1 つにデバッグの開始 (STRDBG) コマンドがあります。

- 対話式ジョブで UPDPROD パラメーターに *YES の値を使用して、(STRDBG) コマンドを発行してください。これでサーバー・ジョブはデバッグ・モードになり、サーバー・ジョブ・ログにデバッグ・メッセージが作られます。

このデバッグ・セッションを終了させるには、サインオフすることによって対話式ジョブを終了するか、またはデバッグの終了 (ENDDBG) コマンドに続いてサービス・ジョブの終了 (ENDSRVJOB) コマンドを使用してください。

SQL ステートメントを実行する前に、サーバー・ジョブをデバッグにしなければならないので、アプリケーションを変更してサーバー上でデバッグをセットアップする時間を確保しなければなりません。サーバー・ジョブは、サーバーへのアプリケーションの接続の結果として開始します。アプリケーションはサーバーへの接続後、デバッグがサーバーで開始されるまで、待ち状態に入るようにコーディングすることができます。

TCP/IP 接続で使用される事前開始ジョブが起きる前、たとえば作業の待ちが 1 つしかない場合や他のクライアントからの妨害がない場合などに、それを予想できる場合、遅延を行う必要はありません。

関連概念:

266 ページの『リレーショナル・データベースの活動のモニター』

次に挙げる制御言語 (CL) コマンド (すべてのコマンドがさまざまな方法で同じ情報を提供する) を使用すれば、IBM i オペレーティング・システムでの作業の概要を把握することができます。

396 ページの『DRDA と DDM のトラブルシューティング』

分散リレーショナル・データベースへのアクセスで問題が生じた場合、問題の性質を判別し、問題がアプリケーションによるものか、ローカル・システムまたはリモート・システムによるものかを判別してください。

424 ページの『QRWOPTIONS データ域』

DDM または DRDA TCP/IP サーバー・ジョブは、開始されると、ユーザーが診断や他のオプションを指定できるデータ域を探します。このデータ域の名前は QRWOPTIONS で、このデータ域は QGPL ライブラリーに常駐している場合にのみ有効になります。このデータ域は、48 文字のストリングから成っています。

関連タスク:

74 ページの『DDM ファイルのセットアップ』

分散リレーショナル・データベース体系 (DRDA) サポートの IBM i 実装では、通信のために分散データ管理機能 (DDM) 会話を使用します。したがって、分散リレーショナル・データベース処理に関連して DDM を使用することができます。

関連資料:

デバッグ終了 (ENDDBG) コマンド

サービス・ジョブ終了 (ENDSRVJOB) コマンド

デバッグの開始 (STRDBG) コマンド

サービス・ジョブの開始 (STRSRVJOB) コマンド

134 ページの『SQL CALL ステートメント (ストアード・プロシージャ)』

SQL CALL ステートメントは、実際は分散リレーショナル・データベースに特有のものではありませんが、その主な価値は分散アプリケーションの論理および処理の中にあるので、ここでその説明を行います。

リモート・コマンド投入 (SBMRMTCMD) コマンド

プログラム参照:

プログラムが作成されると、IBM i ライセンス・プログラムは、SQL ステートメントの中で参照されるすべてのコレクション、テーブル、ビュー、SQL パッケージおよび索引についての情報を SQL プログラムに保管します。

プログラム参照表示 (DSPPGMREF) コマンドを使用して、プログラムの中のすべてのオブジェクト参照を表示することができます。SQL 命名規則が使用されている場合には、ライブラリー名は以下のいずれかの方法で保管されます。

- SQL 名が完全修飾されている場合には、コレクション名が名前修飾子として保管されます。
- SQL 名が完全修飾されておらず、DFTRDBCOL パラメーターが指定されていない場合には、ステートメントの権限 ID が名前修飾子として保管されます。
- SQL 名が完全修飾されておらず、DFTRDBCOL パラメーターが指定されている場合には、DFTRDBCOL パラメーターで指定されたコレクション名が名前修飾子として保管されます。

システムの命名規則が使用されている場合には、ライブラリー名は以下のいずれかの方法で保管されます。

- オブジェクト名が完全修飾されている場合には、ライブラリー名が名前修飾子として保管されます。
- オブジェクトが完全修飾されておらず、DFTRDBCOL パラメーターが指定されていない場合には、*LIBL が保管されます。
- SQL 名が完全修飾されておらず、DFTRDBCOL パラメーターが指定されている場合には、DFTRDBCOL パラメーターで指定されたコレクション名が名前修飾子として保管されます。

関連資料:

プログラム参照表示 (DSPPGMREF) コマンド

SQL パッケージの処理

SQL パッケージとは、特に分散リレーショナル・データベース・アプリケーションによって使用される SQL オブジェクトです。これには、サーバー上のデータにアクセスする各 SQL ステートメントの制御構造が含まれます。

これらの制御構造は、アプリケーションが SQL ステートメントを使用してデータを要求する実行時に、サーバーによって使用されます。

SQL パッケージ作成のための SQL ステートメントはありません。SQL パッケージは、以下の 2 つの方法で作成することができます。

- RDB パラメーターにリレーショナル・データベース名を指定した CRTSQLxxx コマンドを使用する
- **SQL パッケージの作成 (CRTSQLPKG)** コマンドの使用

ユーザーが上記のいずれかのオプションを使用して SQL パッケージを作成することが推奨されていますが、パッケージが見つからない場合は暗黙的なパッケージ作成が試行されます。パッケージの所有者は、アプリケーションの所有者に設定されます。

関連資料:

134 ページの『分散リレーショナル・データベースのステートメント』

SQL 言語を使用して組み込まれるステートメントは、特に分散リレーショナル・データベースをサポートするためのものです。

143 ページの『SQL パッケージの作成』

CRTSQLxxx コマンドのコンパイル時に、プリコンパイル・プロセスの一部として、SQL パッケージと呼ばれるオブジェクトを作成することができます。

145 ページの『アプリケーションのバインド』

アプリケーション・プログラムで実行できるようにするには、その前にプログラムと参照されるテーブルおよびビューとの間の関係が確立されていなければなりません。この処理がバインドと呼ばれるものです。

SQL パッケージの作成 (CRTSQLPKG) コマンドの使用:

SQL パッケージの作成 (CRTSQLPKG) コマンドを入力して、コンパイル済みの分散リレーショナル・データベース・プログラムから SQL パッケージを作成することができます。また、このコマンドを使用すれば、前に作成されていた SQL パッケージを置き換えることもできます。

新規 SQL パッケージは、RDB パラメーターによって定義されたりレーショナル・データベース上に作成されます。新規 SQL パッケージは、CRTSQLxxx コマンドの PKG パラメーターで指定されているものと同じ名前を持ち、同じライブラリーに入れられます。

サーバー・システムに SQL パッケージを作成するのに、IBM DB2 Query Manager and SQL Development Kit for i ライセンス・プログラムは必要ありません。

関連資料:

123 ページの『デフォルトのコレクション名』

プログラムのプリコンパイル時に、CRTSQLxxx コマンドの DFTRDBCOL パラメーターにデフォルトのコレクション名を指定することによって、デフォルトのコレクション名が SQL プログラムで使用されるように指定することができます。

構造化照会言語パッケージの作成 (CRTSQLPKG) コマンド

SQL パッケージの管理:

SQL パッケージの作成後は、IBM i オペレーティング・システム上の他のオブジェクトの管理と同じ要領で、SQL パッケージを管理することができます。

SQL パッケージを保管および復元し、他のシステムへ送り、パッケージに対するユーザーの権限の認可および取り消しを行うことができます。また、**構造化照会言語パッケージの削除 (DLTSQLPKG)** コマンドまたは DROP PACKAGE SQL ステートメントを入力することによって、SQL パッケージを削除することもできます。

分散 SQL プログラムが作成されると、SQL パッケージの名前および内部整合性トークンがプログラムに保管されます。これらは、SQL パッケージを見つけ、SQL パッケージがそのプログラムにとって正しいかどうかを検査するために実行時に使用されます。SQL パッケージの名前は SQL プログラムの実行にとって重要なので、SQL パッケージを別のライブラリーに移動、名前変更、複写、または復元することはできません。

SQL パッケージに関する情報の表示:

SQL 情報印刷 (PRTSQLINF) コマンドを使用して、SQL パッケージに保管されている情報を表示できます。

表示される情報には、パッケージ中のコミットメント制御レベル、オプション、静的ステートメントが含まれます。例えば、以下のコマンドは、SPIFFY コレクションの中の SQL パッケージ PARTS1 を表示します。

```
PRTSQLINF OBJ(SPIFFY/PARTS1) OBJTYPE(*SQLPKG)
```


SQL パッケージの削除 (DLTSQLPKG) コマンドを使用した SQL パッケージの削除:

構造化照会言語パッケージの削除 (DLTSQLPKG) コマンドを使用して、1 つまたは複数の SQL パッケージを削除することができます。削除する SQL パッケージが存在しているシステムで、DLTSQLPKG コマンドを実行しなければなりません。

SQL パッケージに対しては *OBJEXIST 権限が、また SQL パッケージが入っているコレクションに対しては、少なくとも *EXECUTE 権限が必要です。

パッケージを除去するためには、SQL を使用する方法も何種類かあります。

- IBM DB2 Query Manager and SQL Development Kit for i ライセンス・プログラムをインストールしている場合は、対話式 SQL を使用してサーバーを接続し、次に SQL DROP PACKAGE ステートメントを使用してパッケージを除去します。
- 接続する SQL プログラムを実行し、次にパッケージを除去します。
- Query 管理機能を使用してパッケージを接続し、除去します。

以下のコマンドは、SPIFFY コレクションの中の SQL パッケージ PARTS1 を削除します。

```
DLTSQLPKG SQLPKG(SPIFFY/PARTS1)
```

リモート IBM i オペレーティング・システム上の SQL パッケージを削除するには、SBMRMTCMD (リモート・コマンド投入) コマンドを使用して、リモート・システムに対して構造化照会言語パッケージの削除 (DLTSQLPKG) コマンドを実行してください。また、ディスプレイ・パススルーを使用して、リモート・システムにサインオンし、SQL パッケージを削除することもできます。リモート・システムが IBM i オペレーティング・システムではない場合には、リモート・ワークステーション・プログラムを使用してそのシステムにパススルーしてから、SQL パッケージ削除コマンドをそのシステムにローカルで投入してください。

関連資料:

構造化照会言語パッケージ削除 (DLTSQLPKG) コマンド

リモート・コマンド投入 (SBMRMTCMD) コマンド

パッケージ権限の変更:

IBM i 上で作成されたどのプログラムでも、そのパッケージの使用が許可されたユーザーを変更できます。

これには、次のように SQL の GRANT および REVOKE ステートメントを使用します。

- GRANT ALL PRIVILEGES ON TABLE テーブル名 TO ユーザー (ユーザーは PUBLIC でもよい)
- GRANT EXECUTE ON PACKAGE パッケージ名 (通常は IBM i プログラム名) TO ユーザー (ユーザーは PUBLIC でもよい)

またコマンド行に GRTOBJAUT および RVKOBJAUT コマンドを入力しても実行できます。

SQL DROP PACKAGE ステートメントの使用:

DROP PACKAGE ステートメントには、分散リレーショナル・データベースに関する PACKAGE パラメーターが組み込まれています。DROP PACKAGE ステートメントは、プログラムに組み込んで発行することも、対話式 SQL を使用して発行することもできます。

DROP PACKAGE を発行すると、SQL パッケージとその記述がサーバーから削除されます。この結果は、構造化照会言語パッケージの作成 (DLTSQPKG) コマンドをローカル・システムに入力した場合と同じになります。SQL パッケージに依存している他のオブジェクトが、このステートメントの結果として削除されることはありません。

SQL パッケージを正常に削除するには、その SQL パッケージに対して次の特権を持っていない限りなりません。

- 参照コレクションに対するシステム権限 *EXECUTE
- SQL パッケージに対するシステム権限 *OBJEXIST

以下の例には、DROP PACKAGE ステートメントを発行する方法が示してあります。

```
DROP PACKAGE SPIFFY.PARTS1
```

プログラムは、現在使用している SQL パッケージに対して DROP PACKAGE ステートメントを出すことはできません。

関連資料:

SQL DROP ステートメント

DDM のアプリケーション開発

このトピックでは、DDM に関するプログラミングの考慮事項をリストします。DDM で使用できる CL コマンド、言語、およびユーティリティーが含まれます。

DDM ファイルおよび SQL

IBM i 分散データ管理 (DDM) サポートを使用すると、SQL 分散リレーショナル・データベース・サポートも使用するプログラム内の分散リレーショナル・データベース・タスクの実行に役立ちます。

たとえば、大量のレコードを取り出すには、DDM およびファイル・コピー (CPYF) コマンドを使用する方が、SQL FETCH ステートメントを使用するよりも迅速にできる場合があります。また、DDM を使用すると、コンパイル時に取り込まれたリモート・システム・データの外部ファイル記述を、分散リレーショナル・データベース・アプリケーションで使用するために取り出すことができます。そのためには、55 ページの『初期セットアップ』で説明してあるように DDM を使用する必要があります。

次の例には、リレーショナル・データベース・ディレクトリー項目を追加し、DDM ファイルを作成して、サーバーとクライアントで同じジョブが使用できるようにする方法を示してあります。

注:

- 共用される会話に関しては、両方の接続が保護であるか、または両方の接続が非保護であるかどうかでなければなりません。
- コードの例を使用することは、443 ページの『コードに関するライセンス情報および特記事項』の条件に同意したものとされます。

Relational Database Directory:

```
ADDRDBDIRE      RDB(KC000) +  
                 RMTLOCNAME(KC000)  
                 TEXT('Kansas City regional database')
```

DDM File:

```
CRTDDMF FILE(SPIFFY/UPDATE)
        RMTFILE(SPIFFY/INVENTORY)
        RMTLOCNAME(KC000)
        TEXT('DDM file to update local orders')
```

以下は、リモート・システム上の同じジョブの中で、リレーショナル・データベース・ディレクトリー項目と DDM ファイルの両方を使用するプログラム例です。

```
CRTSQLxxx PGM(PARTS1) COMMIT(*CHG) RDB(KC000) RDBCNNMTH(*RUW)
```

```
PROC :PARTS1;
OPEN SPIFFY/UPDATE;
.
.
.
CLOSE SPIFFY/UPDATE;
.
.
EXEC SQL
SELECT * INTO :PARTAVAIL
FROM INVENTORY
WHERE ITEM = :PARTNO;
EXEC SQL
COMMIT;
.
.
END PARTS1;
```

関連資料:

ファイル・コピー (CPYF) コマンド

DDM に対する言語、ユーティリティー、およびアプリケーション・サポートの使用

このトピックでは、IBM i DDM のために用意されている言語、ユーティリティー、およびアプリケーション・プログラムのサポートについて説明します。

VS 顧客情報管理システム (CICS) ファイルへのアクセスに関する言語固有の情報は、トピック『DDM での IBM i と CICS 間の考慮事項』にあります。

DDM のプログラミング言語に関する考慮事項:

DDM は、ILE C、ILE COBOL、ILE RPG、PL/I PRPQ および制御言語 (対話式およびコンパイル済み形式) でサポートされます。

DDM のすべての言語に関する考慮事項:

高水準言語 (HLL) プログラムにおいて DDM ファイルは、データ・ファイルとしても、またはソース・ファイルとしても使用することができます。

ただし、CL、データ記述仕様 (DDS)、および BASIC では、DDM ファイルをソース・ファイルとして使用する場合は、サーバー・システムは IBM i またはシステム/38 でなければならず、DDM ファイルによって参照されるファイルは、ターゲット IBM i またはシステム/38 上でソース・ファイルとして定義されていなければなりません。つまり、リモート・ファイルを前もってソース物理ファイル作成 (CRTSRCPF) コマンドで作成するか、または物理ファイル作成 (CRTPF) コマンドで FILETYPE (*SRC) として作成しておく必

要があります。これらの制約事項は ILE RPG、ILE COBOL、および ILE C コンパイラーには適用されず、IBM i および非 IBM i のどちらのサーバー・システムからでもソース・ファイルを使用できます。

サーバー・システムが IBM i またはシステム/38 ではないときに、ソース・ファイル・メンバー 名を指定した場合、SRCMBR パラメーターに指定されたソース・メンバーの名前が SRCFILE パラメーターに指定された DDM ファイルの名前と異なっていると、すべての HLL コンパイラーはコンパイルを終了します。

ローカル・ファイルにアクセスしたプログラムが、リモート・ファイルにアクセスすることになる場合、そのプログラムを変更して再コンパイルすることを義務づけるある種の制約を受けることがあります。さらに、サーバー・システムが IBM i でもシステム/38 でもない場合は、外部記述データをローカル (クライアント) システムに常駐させておかなければならない場合もあります。これらの制約事項に関してはすべて、『DDM のプログラム変更要件』で説明しています。

サーバー・システムが IBM i でもシステム/38 でもない場合、オープン・フィードバックで返されるレコード数は有効でないことがあります。

SRCFILE パラメーターにライブラリー名を指定しなかった場合、SRCFILE パラメーターに指定したファイルと同じ名前を持ち、ライブラリー・リストで最初に見つかったユーザーのファイルが、ソース・ファイルとして使用されます。

IBM i DDM による HLL プログラムの入出力命令:

高水準言語命令は、キー順または非キー順操作のいずれも DDM によってサポートされます。

次の表の情報を参照してください。

表 7. キー順または非キー順操作の場合に DDM がサポートする高水準言語命令

	高水準言語			
	ILE RPG プログラミング 言語	ILE COBOL プログラミング 言語	BASIC	PL/I
IBM i データベース操作 ファイル・オープン ファイルの照会	OPEN	OPEN	OPEN	OPEN
読み取り (キー順アクセス)	CHAIN (キー)	READ INVALID KEY	READ KEY	READ EQUAL
最初/最終レコードの読み取り ¹	*LOVAL *HIVAL	READ FIRST LAST	READ FIRST LAST	READ FIRST LAST
次のレコードの読み取り	READ READE ²	READ <NEXT> AT END	READ	READ NEXT
直前のレコードの読み取り	READP	READ PRIOR AT END	READ PRIOR	READ PRV
次のレコードまたは直前の レコードの読み取り ³ 、次の 等しいレコード、直前の等 しいレコード、次の固有レ コード、直前の固有レコー ド			READ =、 PRIOR	READ NXTEQL PRVEQL NXTUNQ PRVUNQ
読み取り (開始点を基準と した) ⁴	CHAIN (rm)	READ RELATIVE KEY	READ REC=	READ KEY
レコード・アンロック	EXCPT または次の入出力 命令	(次の入出力命令)	(次の入出力 命令)	(次の入出力 命令)
強制データ終了	FEOD			

表7. キー順または非キー順操作の場合に DDM がサポートする高水準言語命令 (続き)

IBM i データベース操作	高水準言語			
	ILE RPG プログラミング 言語	ILE COBOL プログラミン グ言語	BASIC	PL/I
ファイル位置付け ⁵	SETGT SETLL	START KEY GREATER KEY NOT LESS KEY EQUAL	RESTORE	
レコード更新	UPDAT	REWRITE ⁶	REWRITE	REWRITE
レコード書き込み	WRITE/ EXCPT	WRITE ⁶	WRITE	WRITE
レコード削除	DELET	DELETE ⁶	DELETE	DELETE
ファイル・クローズ	CLOSE	CLOSE	CLOSE	CLOSE

¹ ILE RPG 言語の場合、ファイルのキー順アクセス・パスで DESCENDING を指定すると、*LOVAL ではファイルの最終レコード、*HIVAL ではファイルの最初のレコードが取り出されます。

² 重複キー付きファイルの場合、ILE RPG 言語は READ NEXT 操作を実行してから、戻ってきたレコードのキーを対比して、そのレコードが修飾されているかどうかを判別します。修飾されていれば、そのレコードはプログラムに戻されます。修飾されていないならば、ファイルの終わりの標識が返されます。

³ リモート・ファイルが非 IBM i にあれば、上記の操作を DDM を使用して行うことはできません。

⁴ IBM i アプリケーション・プログラムは、1 つのファイルにつながるキー順 アクセス・オープン・データ・パスをオープンしてから、キー順アクセス方法と相対レコード・アクセス方法の両方を使用してそのファイルのレコードにアクセスすることができます。DDM で混合アクセス・アクセス方法がサポートされていても、サーバー・システム (システム/36 など) ではこの方式がサポートされていない場合があります。この場合、サーバー・システムがレコード番号別混合 アクセス方法をサポートしており、DDM ファイルでその方式が指定されていれば、IBM i で、非 IBM i ターゲット・サーバーにあるキー付きファイルへの相対レコード・アクセスを実行できます。IBM i では、レコード番号別混合アクセス方法は、**DDM ファイル作成 (CRTDDMF)** コマンドで ACCMTH(*ARRIVAL *BOTH) を指定します。これらの値を DDM ファイルに指定せず、さらにサーバー・システムが混合アクセス・アクセス方法のサポートもしない場合には、キー付きファイルに対する相対レコード操作は拒否されます。

⁵ 位置決め操作 (ILE RPG 言語の場合は SETxx ILE COBOL 言語の場合は START) では、レコード・データはアプリケーション・プログラムに戻されません。また、このような操作を行うと、ランダム処理のためにファイルがオープンされます。

⁶ 索引付きファイルまたは相対ファイルを変更する ILE COBOL 操作では、レコードを処理対象にするための操作の前に、レコードをロックすることができます。

表8. キー順または非キー順操作の場合に DDM がサポートする高水準言語命令

IBM i データベース操作	高水準言語	
	CL	ILE C プログラミング言語
ファイル・オープン	OPNDBF	FOPEN、FREOPEN
ファイルの照会	OPNQRYF	
読み取り (キー順アクセス)		
最初/最終レコード読み取り		
次のレコードの読み取り	RCVF	FREAD、FGETC
直前のレコードの読み取り		
次のレコードまたは直前のレコードの読み取り：次の等しいレコード、直前の等しいレコード、次の固有レコード、直前の固有レコード		

表 8. キー順または非キー順操作の場合に DDM がサポートする高水準言語命令 (続き)

IBM i データベース操作	高水準言語	
	CL	ILE C プログラミング言語
読み取り (開始点を基準とした)		
レコード・アンロック		(次の入出力命令)
強制データ終了		FFLUSH
ファイル位置付け	POSDBF	FSEEK、FSETPOS
レコード更新		FWRITE、FPUTC、FFLUSH
レコード書き込み		FWRITE、FPUTC、FFLUSH
レコード削除		
ファイル・クローズ	CLOF	FCLOSE

DDM のコミットメント制御サポート:

IBM i アプリケーション・プログラムでは、リモート IBM i オペレーティング・システムでトランザクションをコミットしたりロールバックすることができます。

ただし、DDM は、IBM i ジャーナル・コマンド (CRTJRN、CRTJRNRCV、および STRJRNPF) をサポートしません。アプリケーションを実行するには、あらかじめターゲット IBM i オペレーティング・システムで、コミットメント制御の下で使用する回復可能資源用のジャーナルを作成してから、コミットメント制御の下でオープンする物理ファイルのジャーナル処理を開始し、クライアント・システムで、**コミットメント制御開始 (STRCMTCTL)** コマンドを発行する必要があります。**STRCMTCTL** コマンドは、DDM ファイルに対して **通知オブジェクト (NTFOBJ)** コマンドをサポートしません。リモート・システムにジャーナル処理を設定する別の方法は、DDM がサポートする **SBMRMTCMD** を使用し、リモート・ファイルをジャーナルするためにサーバー・システムに対するジャーナル・コマンドを実行要求します。

- 1 DDM 会話が 2 フェーズ・コミットメント制御を使用できるようにするには、DDM 会話を保護すること
- 1 が必要です。DDM 会話を保護するには、該当する DDM ファイルを作成するときに、保護会話
- 1 (PTCCNV) パラメーターを *YES に設定しておく必要があります。保護会話 (PTCCNV) パラメーター
- 1 は、アドレス・タイプが *SNA および *IP の場合に、*YES の値を許可します。*IP アドレス・タイプに
- 1 PTCCNV(*YES)) を設定するには、リモート・ロケーション (RMTLOCNAME) パラメーターに *RDB を
- 1 指定して、リレーショナル・データベース (RDB) パラメーターに RDB を指定します。RDB リレーシ
- 1 ナル・データベース・ディレクトリー項目には、*IP または *SNA のいずれかを指定できます。RDB ディ
- 1 レクトリー項目を追加するには、RDB ディレクトリー項目の追加 (ADDRDBDIRE) コマンドとリレーシ
- 1 ナル・データベース・ディレクトリー項目の処理 (WRKRDBDIRE) コマンドを参照してください。

コミットメント制御を使用した DDM ファイルの用法:

コミットメント制御の下でオープンされる DDM ファイルを処理する際には、以下の制約事項を考慮する必要があります。

- コミットメント制御の下で複数の DDM ファイル (PTCCNV(*NO) 指定で) をオープンする場合は、各ファイルとも次の項目は同じになるようにしてください。
 - リモート・ロケーション名
 - ローカル・ロケーション名
 - 装置
 - モード

- リモート・ネットワーク ID
- トランザクション・プログラム名 (TPN)
- ユーザー ID
- 活動化グループ番号
- オープンの有効範囲

この規則の例外として、コミットメント制御の下にオープンする DDM ファイルの有効範囲がすべて、ジョブ・レベルまでになっている場合があります。このような場合、活動化グループ番号は無視されるため、一致している必要はありません。

- DDM ファイルとリモート SQL オブジェクト (分散リレーショナル・データベース体系、DRDA) をコミットメント制御 (PTCCNV(*NO) を指定) の下で実行するときは、ファイルとオブジェクトの次の項目が同じになるようにしてください。
 - リモート・ロケーション名
 - ローカル・ロケーション名
 - 装置
 - モード
 - リモート・ネットワーク ID
 - TPN
 - ユーザー ID
 - 活動化グループ番号
 - オープンの有効範囲
- DDM ファイル (PTCCNV(*YES)) を出力、更新、または削除用にオープンしている (入力専用でオープンしているのではない) 場合、1 フェーズ DDM または DRDA 会話を活動化することはできません。
- DDM ファイル (PTCCNV(*YES) 指定で) を使用している場合、そのファイルは、2 フェーズ・コミットメント制御プロトコルをサポートするターゲット IBM i を指している必要があります。
- DDM ファイル (PTCCNV(*NO) 指定で) とローカル・データベース・ファイルを、同一活動化グループ内のコミットメント制御の下で同時にオープンすることはできません。
- コミットメント制御の有効範囲がジョブ・レベルまでのときには、DDM ファイル (PTCCNV(*NO) 指定で) とローカル・データベース・ファイルを、同一ジョブ内で同時にコミットメント制御の下でオープンすることはできません。
- コミットメント制御の下で DDM ファイルをオープンし、その有効範囲をジョブ・レベルまでに制御する場合、あらかじめ **コミットメント制御開始 (STRCMTCTL)** コマンドで **CMTSCOPE (*JOB)** を指定しておかなければなりません。
- コミットメント制御の有効範囲がジョブ・レベルまでであることを前提としたプログラムを呼び出すために、**リモート・コマンド投入 (SBMRMTCMD)** コマンドを使用することはできません。この場合、DDM ターゲット・ジョブでのコミットメント制御の有効範囲は常に活動化グループ・レベルまでであるため、そのプログラムは失敗します。
- コミットメント制御を開始したり終了したりするのに、**SBMRMTCMD** コマンドを使用することはできません。
- コミットメント制御の下で稼働するシステムで指定するサーバー・システムも、同じく IBM i でなければなりません。

注: COMMIT 操作の途中で通信回線障害が起こったときには、クライアント・システムとサーバー・システムが ROLLBACK 操作を行います。ただし、回線障害の起きる前にサーバー・システムが正常に COMMIT 操作を完了していたとしても、クライアント・システムは常に ROLLBACK 操作を行います。

表9. 高水準言語コミットおよびロールバック・コマンド

操作	ILE RPG プログラミング言語	ILE COBOL プログラミング言語	PL/I	CL	ILE C プログラミング言語
トランザクション内の変更のコミット	COMMIT	COMMIT	PLICOMMIT	COMMIT	_Rcommit
トランザクション全体の取り消し	ROLBK	ROLLBACK	PLIROLLBACK	ROLLBACK	_Rrollback

DDM の ILE RPG に関する考慮事項:

ILE RPG プログラムと報告書簡易作成機能プログラムは、いずれも DDM ファイルを参照することができます。DDM ファイル名は、一般に、サーバー・システムが IBM i であっても IBM i 以外であっても、データベース・ファイル名を指定できる場所ならどこにでも ILE RPG で指定することができます。

- DDM ファイル名の指定は、**RPG プログラム作成 (CRTRPGPGM)** および**報告書簡易作成プログラム作成 (CRTRPTPGM)** コマンドで行うことができます。
 - IBM i の場合も IBM i 以外の場合も、ソース・ステートメントを含むリモート・ファイルにアクセスするのに、DDM ファイル名を SRCFILE パラメーターで指定し、そしてメンバー名を SRCMBR パラメーターで指定することができます。
 - IBM i またはシステム/38 サーバー・システムの場合、リモート IBM i またはシステム/38 ソース・ファイル (または、任意指定で、メンバー) へのアクセスは、ローカル・ソース・ファイルおよびメンバーにアクセスする場合と同様の方法で実行できます。
 - IBM i 以外のサーバー・システムの場合、いずれかのコマンドに PGM パラメーターと SRCMBR パラメーターの両方のデフォルトを使用すれば、リモート・ソース・ファイルにアクセスすることができます。また、メンバー名を指定する場合、その名前は、SRCFILE パラメーターで指定された DDM ファイル名と同じでなければなりません。(報告書簡易作成機能プログラムを作成するのに使用する入力仕様の /COPY ステートメントで指定されるメンバー名や、ソースを組み込むためにコンパイラーが使用するメンバー名の場合も、同様です。)
 - サーバー・システム上のデータベース・ファイル内にコンパイラー・リストを入れるため、上記のいずれかのコマンドの PRTFILE パラメーターに DDM ファイル名を指定することができます。
- DDM ファイル名とメンバー名は、**CRTRPTPGM** コマンドの OUTFILE パラメーターと OUTMBR パラメーターで指定することができます。ただし、このコマンドで作成される出力を、その DDM ファイルが参照するリモート・ファイル内に保管するには、先にそのリモート・ファイルがすでに存在していなければなりません。また、ローカル・ファイルの場合と同様、リモート・ファイルのレコード様式は、必要な OUTFILE パラメーター様式と一致していなければなりません。一般的に、このことは、サーバー・システムが IBM i オペレーティング・システムまたはシステム/38 製品でなければならないことを意味します。

サーバー・システムが IBM i であっても IBM i 以外であっても、ILE RPG プログラムがクライアント・システム上で DDM ファイルをオープンすると、ターゲット・サーバーのリモート・ファイルで CHAIN、CLOSE、DELET、EXCPT、FEOD、OPEN、READ、READE、READP、SETGT、SETLL、UPDAT、および WRITE タイプの入出力操作を実行できます。

その他の考慮事項を次に示します。

- プログラム内で外部記述される DDM ファイルが宣言されると、ILE RPG コンパイラーによるコンパイル時に、参照されるリモート・ファイルの外部記述がプログラムにコピーされます。ただし、そのリモート・ファイルが IBM i または システム/38 以外のシステムにある場合、レコード記述に対して宣言されるフィールドの名前には意味がなくなります。その代わりに、すべてのフィールド名は Fnnnnn と宣言され、キー・フィールドは Knnnnn と宣言されます。

ターゲット・システムが IBM i でも システム/38 でもない場合、リモート・ファイルを記述するための推奨される方法は、データ記述仕様 (DDS) をローカル・システムに設定し、ローカル・システム上で **物理ファイル作成 (CRTPF)** コマンドまたは **論理ファイル作成 (CRTLF)** コマンドを入力するという方法です。ローカル・ファイル名を使用してプログラムをコンパイルします。リモート・システムのファイルのフィールド・タイプとフィールド長は、必ずローカル・ファイルのものと対応するように注意してください。

リモート・ファイルにアクセスするためには、次に示すとおり、プログラムの前に **データベース・ファイル一時変更 (OVRDBF)** コマンドを使用します。

```
OVRDBF FILE(PGMFIL) TOFILE(DDMFIL) LVLCHK(*NO)
```

- また DDM ファイルは、ILE RPG プログラムに指定するファイルとしても有効で、ILE RPG の論理サイクルで暗黙に使用されます。
- サーバー・システムが IBM i でも システム/38 でもない場合にレコード様式名を使用するときは、DDM ファイル名と一致する名前を付けるようにしてください。
- システム/36 で作成した ADDROUT ファイルを IBM i 上で使用することはできません。IBM i システム/36 互換 RPG II では、3 バイトの ADDROUT ファイルが使用され、IBM i オペレーティング・システムおよびシステム/38 で ILE RPG プログラミング言語を使用する場合は、4 バイトの ADDROUT ファイルが使用されます。

DDM の ILE COBOL に関する考慮事項:

ILE COBOL プログラムは DDM ファイルを参照することができます。DDM ファイル名は、一般に、サーバー・システムが IBM i を実行しても、IBM i を実行しなくても、データベース・ファイル名を指定できるところならどこにでも ILE COBOL で指定することができます。

- DDM ファイル名は、COBOL プログラム作成 (CRTCLPLGM) コマンドで指定することができます。
 - ソース・ステートメントを含まリモート・ファイルにアクセスするのに、DDM ファイル名を SRCFILE パラメーターで指定し、そしてメンバー名を SRCMBR パラメーターで指定することができます。
 - IBM i または システム/38 サーバー・システムの場合、リモート IBM i または システム/38 ソース・ファイル (または、任意指定で、メンバー) へのアクセスは、ローカル・ソース・ファイルおよびメンバーにアクセスする場合と同様の方法で実行できます。
 - IBM i を実行しないサーバー・システムの場合、CRTCLPLGM コマンドで PGM パラメーターと SRCMBR パラメーターの両方にデフォルトを使用すれば、リモート・ソース・ファイルにアクセスすることができます。また、メンバー名を指定する場合、その名前は、SRCFILE パラメーターで指定された DDM ファイル名と同じでなければなりません。
 - サーバー・システム上のデータベース・ファイル内にコンパイラー・リストを入れるため、CRTCLPLGM コマンドの PRTRFILE パラメーターに DDM ファイル名を指定することができます。
- ILE COBOL の SORT 操作と MERGE 操作では、入力ファイルおよび出力ファイルとして DDM ファイル名を指定することができます。(この操作での作業ファイルは、DDM ファイルであってはなりません。)

- ILE COBOL の COPY ステートメントの DDS オプションで、1 つまたは全部の外部記述レコード様式を、DDM ファイルが参照するリモート・ファイルから、コンパイル中のプログラムにコピーする場合は、この COPY ステートメントで DDM ファイルを使用することができます。リモート・ファイルが IBM iにもシステム/38 にもないときにこのコピー操作を行うと、そのレコード記述のために宣言されるフィールドの名前には意味がなくなります。その代わりに、すべてのフィールド名は Fnnnnn と宣言され、キー・フィールドは Knnnnn と宣言されます。

ターゲット・システムが IBM i でもシステム/38 でもない場合、リモート・ファイルを記述するための推奨される方法は、データ記述仕様 (DDS) をローカル・システムに設定し、ローカル・システム上で物理ファイル作成 (CRTPF) コマンドまたは論理ファイル作成 (CRTLF) コマンドを入力するという方法です。ローカル・ファイル名を使用してプログラムをコンパイルします。リモート・システムのファイルのフィールド・タイプとフィールド長は、必ずローカル・ファイルのものと対応するように注意してください。

リモート・ファイルにアクセスするためには、次に示すとおり、プログラムの前にデータベース・ファイル一時変更 (OVRDBF) コマンドを使用します。

```
OVRDBF FILE(PGMFIL) TOFILE(DDMFIL) LVLCHK(*NO)
```

- DDM ファイル名は COPY ステートメントで指定できます。
 - ファイル名と一緒にライブラリー名を指定しない場合、ユーザーのライブラリー・リスト内で最初に見つかったそのファイル名の付いたファイルが、組み込みファイルとして使用されます。
 - サーバー・システムが IBM iでもシステム/38 でもない場合、DDM ファイル名を、COPY ステートメント上で組み込みファイルとして指定することはできますが、そのメンバー名は DDM ファイル名と同じになっている必要があります。
- サーバー・システムがシステム/36 であり、関連リモート・ファイルに、作成された論理ファイルがある場合、ILE COBOL プログラミング言語を使用して DDM ファイルを出力用にオープンすることはできません。論理ファイルが作成されているシステム/36 ファイルの場合、ILE COBOL プログラミング言語がこのファイルを使用する前に消去しようとするため、オープン操作 (オープン出力) は失敗します。

ILE COBOL プログラムがクライアント・システム上で DDM ファイルをオープンすると、CLOSE、DELETE、OPEN、READ、REWRITE、START、および WRITE ステートメントを使用して、サーバー・システムのリモート・ファイルで入出力操作を実行できるようになります。

ILE COBOL による直接ファイル・サポート:

直接ファイルは、IBM i オペレーティング・システムがサポートするファイル・タイプには含まれていません。ただし、IBM i上の ILE COBOL プログラムは、直接ファイルとしてファイルにアクセスすることを指定できます。

通常 IBM i オペレーティング・システムは直接ファイルを順次ファイルとして作成します。IBM i上の ILE COBOL プログラムは、SELECT ステートメントで RELATIVE を指定することによって、ファイルを直接ファイルとして定義します。そのプログラムが出力専用でファイルを開く (OPEN ステートメント上で OUTPUT を指定して) 場合、そのファイルは、削除済みレコードで作成し、活動レコードが含まれないようにする必要があります。これは、IBM i を実行しないクライアント・システム (システム/36 など) が DDM を使用して IBM i 上に直接ファイルを作成または消去する場合のファイルの条件でもありません (以下の説明どおりにファイルが作成されると想定して)。

IBM i と システム/38 では、順次ファイルとキー付きファイル・タイプがサポートされています。DDM は、順次ファイル・タイプ、キー付きファイル・タイプ、および直接ファイル・タイプを認識します。IBM

i 以外のシステムが DDM を使って、IBM i 上で直接ファイルを作成しようとしている場合、DDM 体系コマンドの直接ファイル作成 (CRTDIRF) が使用されます。

IBM i 以外のオペレーティング・システムから CRTDIRF アーキテクチャー・コマンドを発行してファイルを作成するときには、そのファイルは物理ファイルとして作成され、直接ファイルとして指定されます。これは、その後に IBM i を実行しないクライアント・システムから直接ファイルへのアクセスがあった場合に、そのファイルを相手側のシステムも直接ファイルとして識別できるようにするためです。ファイルをこのようにして作成しないと、その後で IBM i はそのファイルが直接ファイルまたは順次ファイルのどちらであるかを判別することができません。なぜなら、前述のとおり、IBM i のファイル・タイプには、直接ファイルは含まれていないためです。

このため、IBM i またはシステム/38 以外のサーバーで、ILE COBOL プログラムが直接モードで (つまり、相対レコード番号を使用して) 出力用に IBM i またはシステム/38 ファイルにアクセスする必要がある場合、ファイルは CRTDIRF アーキテクチャー・コマンドで作成されていなければなりません。

IBM i オペレーティング・システムで出力専用の直接ファイルをサポートするために、ILE COBOL OPEN ステートメントは、オープンするファイルのメンバーを消去し、準備します。このため、他の IBM i オペレーティング・システムやシステム/38 プラットフォームの ILE COBOL プログラムからでも、DDM ファイルを使用すれば、既存の IBM i またはシステム/38 ファイルにアクセスできます。IBM i を実行しないサーバー・システムの場合、エラーが生じないようにするには、出力用にオープンする相対ファイルを直接ファイルとして定義しておく必要があります。

これまでの説明を要約しておきます。

- IBM i 以外のシステムのプログラムまたはユーザーがローカル IBM i にあるファイルを直接ファイルとして作成する場合、IBM i を実行しないリモート・クライアント・システムの ILE COBOL プログラムは直接ファイルとしてこのファイルにアクセスできます。
- プログラムまたはユーザーがローカル IBM i で、同じ IBM i に作成したファイルに、IBM i 以外のシステムが直接ファイルとしてアクセスすることはできません。その理由は、このような場合、そのファイルが直接ファイルまたは順次ファイルのどちらであるかを、IBM i を実行するサーバー・システムが判別できないからです。
- リモート・システムが作成するすべてのファイルを、ローカルで使用することができます。

DDM の PL/I に関する考慮事項:

コンパイル済みのプログラムは DDM ファイルを参照することができます。さらに DDM ファイル名を、**PL/I プログラム作成 (CRTPLIPGM)** コマンドで指定することもできます。

- リモート・ソース・ファイルが IBM i オペレーティング・システムまたはシステム/38 プラットフォームにある場合に限り、DDM ファイル名を SRCFILE パラメーター、メンバー名を SRCMBR パラメーターに指定できます。%INCLUDE ソース・ディレクティブ・ステートメントに DDM ファイルおよびメンバー名を指定するときも、同じ条件です。DDM ファイルが参照するリモート・ファイルが IBM i オペレーティング・システムまたはシステム/38 プラットフォーム以外のシステムにある場合、CRTPLIPGM コマンドや %INCLUDE ステートメントに DDM ファイル名が指定されていると、エラーが発生します。
- DDM ファイルがプログラムのソース・ファイルとしてアクセスされる場合、ソースのコンパイルで使用されるマージンは、デフォルト値の 2 と 72 になります。その他のマージン値を指定することはできません。
- %INCLUDE DDS 指示ステートメントで DDM ファイル名を指定すると、リモート・ファイルのレコード記述がコンパイル済みプログラムに取り込まれます。ただし、そのリモート・ファイルが IBM i オペレーティング・システムまたはシステム/38 プラットフォーム以外のシステムにある場合、レコード記述

に対して宣言されるフィールドの名前には意味がなくなります。その代わりに、すべてのフィールド名は *Fnnnnn* と宣言され、キー・フィールドは *Knnnnn* と宣言されます。

DDM ファイルを使用して、リモート・レコード・ファイルまたはリモート・ストリーム・ファイルを参照することが可能です。プログラムがクライアント・システムの DDM ファイルをオープンする場合、IBM i および非 IBM i ターゲットの両方とも、以下のタイプのステートメントを使用して、サーバー・システムのリモート・ファイルに対して入出力操作を行うことができます。つまり、レコード・ファイル処理には OPEN、CLOSE、READ、WRITE、REWRITE、および DELETE ステートメントを使用し、ストリーム・ファイル処理には GET および PUT ステートメントを使用できます。

サーバー・システムが IBM i オペレーティング・システムでもシステム/38 製品でもない場合は、NXTEQL、PRVEQL、NXTUNQ、または PRVUNQ の値を POSITION パラメーターに指定すると、リモート・ファイルからの読み取り時にキー付き READ ステートメントで POSITION パラメーターを指定しても機能しないという点も考慮する必要があります。(NEXT、PREVIOUS、FIRST、および LAST の値を使用できます。) サーバー・システムが IBM i オペレーティング・システムまたはシステム/38 製品であれば、これらの値はすべて有効です。

DDM の CL コマンドに関する考慮事項:

コンパイル済み制御言語 (CL) プログラムと対話式に入力した CL コマンドのいずれも、DDM ファイルを参照することができます。

一般的に、IBM i および非 IBM i サーバー・システムの両方とも、DDM ファイル名は、CL コマンド内で、データベース・ファイル名を指定できるどの場所にも指定することができます。ただし、いくつかの制限があります。

次に、DDM ファイル名をどこに指定できるかについての例をいくつか示します。

- DDM ファイル名は、ファイルのコピー、表示、および一時変更コマンドなどの、ほとんどのデータベース・ファイル関連コマンドで指定することができます。
- サーバー・システムが IBM i オペレーティング・システムまたはシステム/38 製品の場合に限り、リモート・ソース・ファイルにアクセスする際、DDM ファイル名をファイル作成コマンド上に指定できません。DDM ファイル名は SRCFILE パラメーター、メンバー名は SRCMBR パラメーターに指定できません。DDM ファイルが参照するリモート・ソース・ファイルが IBM i オペレーティング・システムやシステム/38 製品上にないと、エラーが発生します。リモート IBM i またはシステム/38 のソース・メンバーに関する考慮事項は、ローカル・ソース・メンバーの場合と同じです。
- DDM ファイル名は、**ファイル宣言 (DCLF)** コマンドの FILE パラメーターに指定することができます。

コマンドによっては、DDM ファイル名を指定することによって、クライアント・システム上のファイルを処理したり、ターゲット・ファイルを処理することができ、また特定のパラメーター値を使用して、ソース・ファイルやターゲット・ファイルを指定することができます。

DDM の ILE C に関する考慮事項:

ILE C プログラムは DDM ファイルを参照することができます。

DDM ファイル名は、一般に、サーバー・システムが IBM i であっても IBM i 以外であっても、データベース・ファイル名を指定できる場所ならどこにでも ILE C プログラミング言語で指定することができます。

C プログラム作成 (CRTCPGM) コマンドで DDM ファイル名を指定すると、以下を行うことができます。

- IBM i オペレーティング・システムまたは非 IBM i にある、ソース・ステートメントを含むリモート・ファイルへアクセスします。アクセスするためには、SRCFILE パラメーター上に DDM ファイル名を指定し、SRCMBR パラメーターにメンバー名を指定します。

注:

1. IBM i またはシステム/38 サーバー・システムの場合、リモートの IBM i またはシステム/38 ソース・ファイル (またはメンバー) へのアクセスは、ローカル・ソース・ファイルおよびメンバーにアクセスする場合と同様の方法でできます。
 2. IBM i 以外のサーバー・システムの場合、SRCMBR パラメーターと SRCFILE パラメーターに同じファイル名を使用して、リモート・ソース・ファイルにアクセスします。
- サーバー・システムにあるデータベース・ファイル内に、コンパイラー・リストを入れます。そのためには、CRTCPGM コマンドの PRTPFILE パラメーターで DDM ファイル名を指定します。

ILE C プログラミング言語を使用する場合は、以下の点を考慮してください。

- サーバー・システムが IBM i オペレーティング・システムでもシステム/38 でもない場合、DDM ファイル名を、#INCLUDE ソース・ディレクティブ・ステートメントで組み込みファイルとして指定することはできますが、メンバー名は DDM ファイル名と同じになっている必要があります。
- ILE C プログラミング言語でサポートされているのは、順次入出力操作だけです。
- ILE C プログラミング言語ではキー付きファイルは直接サポートされていませんが、キー付きファイルを使用した場合は、キー例外が発生する場合があります。

DDM のユーティリティに関する考慮事項:

以下の IBM i ユーティリティは、リモート・ファイルにアクセスするために DDM をサポートします。

注:

1. 次のユーティリティは、DDM をサポートしません。IBM i Query、原始ステートメント入力ユーティリティ (SEU)、画面設計機能 (SDA)、および拡張印刷機能ユーティリティ。
2. システム/38 互換データベース・ツールまたは DFU/400 を使用している場合は別として、DDM ではリモート・ファイルのメンバーのリスト表示はサポートしていません。ただし、サーバー・システムが IBM i オペレーティング・システムまたはシステム/38 であれば、表示装置パススルーで、この機能を実行することができます。
3. IBM i ライセンス・プログラムの一部である SQL/400 ライセンス・プログラムと QUERY 管理機能は、DDM をサポートしません。ただし、いずれも、分散ネットワークにおいて分散リレーショナル・データベース体系 (DRDA) をサポートします。

システム/38 互換のデータベース・ツール:

このトピックでは、システム/38 と互換性のあるデータ・ファイル・ユーティリティ (DFU/38) およびシステム/38 と互換性のある QUERY ユーティリティ (Query/38) について説明します。

システム/38 互換データ・ファイル・ユーティリティ (DFU/38):

DFU/38 データ入力アプリケーションを作成して DDM とともに使用すれば、ローカル・ファイルの場合と同じようにして、リモート・ファイルを処理することができます。

リモート・ファイルが IBM i オペレーティング・システムまたはシステム/38 製品にある場合は、ローカル・ファイルと同様の方法でリモート・ファイルにより、ほとんどの DFU/38 機能を実行できます。DFU/38 アプリケーションを作成または変更し、さらにリモート・ファイルが論理ファイルである場合に

は、以下の考慮事項のいずれかが適用されます。(1) 基礎となるリモート・ファイルのそれぞれを参照する DDM ファイルが、クライアント・システム上に存在し、さらに、DDM ファイル名とライブラリー名が基礎となるリモート・ファイルのファイル名やライブラリー名と一致している必要がある。または、(2) 基礎となるリモート・ファイルと同一のファイル名とライブラリー名、および同一のレコード様式を持つ物理ファイルが、クライアント・システム上に存在している必要がある。物理ファイルのレコード様式のみが必要なため、そのファイルにはデータが入っていてもかまいません。いずれの考慮事項でも、基礎となるリモート・ファイルのレコード様式を変更したときには、クライアント・システムでのレコード様式も変更して、レコード様式が一致するようにする必要があります。

ただし DFU/38 は、非 IBM i または非システム/38 サーバー・システムをサポートしません。非 IBM i または非システム/38 リモート・ファイルで DFU/38 を使用しようとする、そのファイル内のレコードを変更または削除する際に、処理上の問題が発生する可能性があります。IBM i オペレーティング・システムで、ユーザーがこのようなアプリケーションを作成して使用することはできますが、非 IBM i または非システム/38 システムのリモート・ファイル用にソース IBM i 上で作成されたデフォルト・フィールド記述は、具体的でないため役に立たないものになると考えられます。(このようなファイルは、ファイル名と同じメンバー名を持ったメンバーを 1 つ含む物理ファイルとして扱われます。このファイルには 1 つのレコード様式があり、その様式では、ファイルがキーなしファイルであれば、レコード全体に対して 1 つのフィールドがあります。そして、キー付きファイルであれば、2 つのフィールドがあり、1 つのフィールドがキー用で、もう 1 つがレコードの残りの部分用であるか、または、複数のフィールドがあり、キー付きファイル用に別々のキー・フィールドを持っています。)

すべての DFU/38 コマンドは、ローカル・ファイルまたは DDM ファイルにアクセスするアプリケーションで使用することができます。また、他に制約事項があってもそれを満たしてさえいれば、任意の DFU コマンドのパラメーターで、ローカル・データベース・ファイル名を指定できる場所にはどこにでも、DDM ファイルを指定することができます。

DDM ファイル名の指定は、**DFU アプリケーション作成 (CRTDFUAPP)** コマンドや **DFU ソース検索 (RTVDFUSRC)** コマンドの SRCFILE パラメーターで行うことができますが、これは、サーバー・システムが IBM i オペレーティング・システムまたはシステム/38 であり、ターゲット・ファイルがソース物理ファイルである場合に限られます。

システム/38 互換 QUERY ユーティリティ (QUERY/38):

システム/38 互換 QUERY ユーティリティ (QUERY/38) を DDM で使用すると、対話式またはバッチ式の QUERY アプリケーションを作成し、使用することができます。

サーバー・システムが IBM i またはシステム/38 であれば、ほとんどの機能において、リモート・ファイルをローカル・ファイルと同じように扱うことができます。Query/38 アプリケーションを作成または変更し、さらにリモート・ファイルが論理ファイルである場合には、以下の考慮事項のいずれかが適用されます。(1) 基礎となるリモート・ファイルのそれぞれを参照する DDM ファイルが、クライアント・システム上に存在し、さらに、DDM ファイル名とライブラリー名が基礎となるリモート・ファイルのファイル名やライブラリー名と一致している必要がある。または、(2) 基礎となるリモート・ファイルと同一のファイル名とライブラリー名、および同一のレコード様式を持つ物理ファイルが、クライアント・システム上に存在している必要がある。物理ファイルのレコード様式のみが必要なため、そのファイルにはデータが入っていてもかまいません。いずれの考慮事項でも、基礎となるリモート・ファイルのレコード様式を変更したときには、クライアント・システムでのレコード様式も変更して、レコード様式が一致するようにする必要があります。

サーバー・システムが IBM i でもシステム/38 でもない場合、QUERY/38 アプリケーションを実行するときには、リモート・ファイル内のデータ記述形式およびフィールドに関するローカル・ファイルを参照して

から、データベース・ファイル一時変更 (OVRDBF) コマンドを使用して、ローカル・ファイルを DDM ファイルに一時変更する必要があります。QUERY を作成 (または再作成) するのに使用するローカル・ファイルは、IBM i またはシステム/38 以外のターゲット・ファイルのソース記述と同じレコード様式名を持たなければなりません。デフォルトのレコード様式名はソース DDM ファイルの名前になります。

QUERY/38 は、IBM i やシステム/38 以外のシステム上のファイルを使用するアプリケーションの作成はできますが、IBM i 以外のリモート・ファイル用にソース IBM i 上で作成されたデフォルト・フィールド記述は、具体的でないため役に立たないものになると考えられます。(このようなファイルは、ファイル名と同じメンバー名を持ったメンバーを 1 つ含む物理ファイルとして扱われます。このファイルには 1 つのレコード様式があり、その様式では、ファイルがキーなしファイルであれば、レコード全体に対して 1 つのフィールドがあります。そして、キー付きファイルであれば、2 つのフィールドがあり、1 つのフィールドがキー用で、もう 1 つがレコードの残りの部分用であるか、または、複数のフィールドがあり、キー付きファイル用に別々のキー・フィールドを持っています。)

非 IBM i または非システム/38 の Query/38 の例:

この例は、ローカル・ファイルを作成し、このファイルを使用して、非 IBM i または非システム/38 システムのリモート・ファイルで照会されるデータを定義する方法を示します。

RMTS36FILE という名前の DDM ファイルが IBM i オペレーティング・システム上にあり、このファイルが照会したいリモート・システム/36 ファイルを参照しているとします。次のステップを実行して、リモートのシステム/36 ファイルの属性を判別し、このリモート・ファイルと同じ属性の物理ファイルをローカルに作成し、このリモート・ファイルに対して QUERY/38 を定義し、作成し、実行することができます。

1. ファイル・フィールド記述表示 (DSPFFD) コマンドを使用し、SYSTEM (*RMT) を指定して、RMTS36FILE DDM ファイルに関連したリモート・ファイルの属性を表示します。

```
DSPFFD FILE(RMTS36FILE) SYSTEM(*RMT)
```

この例で表示されている結果は、そのリモート・ファイルについて、レコード長が 80 文字、レコード様式名は RMTS36FILE、そして、K00001 (12 文字、位置 1 で開始) と F00001 (68 文字、位置 13 で開始) との 2 つのフィールドを持っていることを示しています。フィールド K00001 内の K は、この形式でのキー・フィールドを示しています。

2. DDS と上記の情報を使用して、QUERY/38 アプリケーションを定義する前に、ローカル物理ファイルを作成し、それを LCLS36FILE と名付けます。その DDS は次のようなものになります。

```
A      R RMTS36FILE
A      CUSNO          6A
A      BILLCODE      6A
A      ADDR1         15A
A      ADDR2         15A
A      ADDR3         15A
A      ZIP            5A
A      AMTOWE        7S 2
A      OUTBAL        7S 2
A      MISC           4A
A      K CUSNO
A      K BILLCODE
```

ローカル・ファイルを定義するときには、次の 3 つの主要規則を守らなければなりません。

- レコード様式名は、ファイル・フィールド記述表示 (DSPFFD) コマンドが表示したレコード様式名と同じでなければなりません。
- キーの整合性を保たなければなりません。この場合、キーは 12 文字の長さで、ファイルの先頭の 1 桁目から始まらなければなりません。
- 合計レコード長は、DSPFFD コマンドで表示されるレコード長と同じでなければなりません。

3. ステップ 2 で作成したローカル・ファイルを使用して、QUERY/38 アプリケーションを定義します。リモート・ファイルは非 IBM i ファイルであるため、QUERY コマンドに OPTIMIZE (*NO) を指定する必要があります。
4. QUERY/38 アプリケーションを実行する前に、次のようなデータベース・ファイル一時変更 (OVRDBF) コマンドを発行します。

```
OVRDBF FILE(LCLS36FILE) TOFILE(RMTS36FILE)
```

QUERY/38 アプリケーションを実行すると、このコマンドで、ユーザーが作成したローカル・ファイルを、所定のターゲット・ファイルと関連した DDM ファイルに指定変更します。

5. データ照会 (QRYDTA) コマンドを使用して、QUERY/38 アプリケーションを実行します。結果として、実際には、ローカル・ファイル記述を使用したリモート・ファイルの照会が行われます。

DDM の Query/38 出力に関する考慮事項:

IBM i でもシステム/38 でもない既存のターゲット・ファイルへの QUERY/38 出力は可能ですが、これは特定の状況下に限られます。

QUERY/38 の出力を任意のローカル・ファイルまたはリモート・ファイルに対して行うことができるのは、そのファイルが順次ファイルであり、そのフィールド属性が、QUERY/38 アプリケーションで定められた属性と一致する場合のみです。いずれの条件も満たされなければ、QUERY/38 は、QUERY/38 アプリケーションが稼働する前に、指定出力ファイルを拒否します。

IBM i やシステム/38 以外のターゲット・ファイルのクライアント・システム記述は非常に一般的であるため、この記述のフィールド属性は、QUERY/38 アプリケーション要件の属性とは一致しない可能性があります。したがって多くの場合、出力用に指定されたファイルを、QUERY/38 は拒否します。ただし、QUERY/38 出力が 1 つの英数字フィールドのみで構成されていて、また、ターゲット・ファイルのレコード長がそのフィールドを収容するのに十分な長さであれば、そのファイルを出力用にすることができます。

DDM の Query/38 コマンドに関する考慮事項:

すべての QUERY/38 コマンドは、ローカル・ファイルまたは DDM ファイルにアクセスするアプリケーション内で使用することができます。また、他に制約事項があってもそれを満たしてさえいれば、任意の QUERY/38 コマンドのパラメーター上で、ローカル・データベース・ファイル名を指定できるところにはどこにでも、DDM ファイルを指定することができます。

注: IBM i でもシステム/38 でもないサーバー・システム上のリモート・ファイルと関連した DDM ファイルを QUERY/38 コマンドが使用する場合、その DDM ファイルで LVLCHK (*NO) を指定するか、または、OVRDBF コマンドを使用して、そのパラメーターを *NO に指定変更する必要があります。これはターゲット・ファイルに関するレベル・チェックの問題を避けるために推奨されます。

DDM ファイル名の指定は、**QUERY アプリケーション作成 (CRTQRYAPP)** または **QUERY ソース検索 (RTVQRYSRC)** コマンドの SRCFILE パラメーターで行うことができますが、これは、サーバー・システムが IBM i またはシステム/38 であり、ターゲット・ファイルがソース物理ファイルである場合に限られます。

DDM の Query/38 最適化:

QUERY/38 には最適化機能が備えられていますが、この機能では IBM i データベース QUERY が使用されることになるため、IBM i にもシステム/38 にもないリモート・ファイルに対して照会を実行する場合、この機能は使用できません。

IBM i データベース QUERY は、非 IBM i または非システム/38 には存在しないため、非 IBM i または非システム/38 リモート・ファイルに対して照会を実行する場合、ソース IBM i は最適化機能を使用できません。

このため、非 IBM i または非システム/38 システム上のリモート・ファイルにアクセスするために QUERY/38 アプリケーションを作成または変更する場合は、**QUERY アプリケーション作成 (CRTQRYAPP)**、**QUERY 定義作成 (CRTQRYDEF)**、または **QUERY 定義変更 (CHGQRYDEF)** コマンドの OPTIMIZE パラメーターを *NO に指定変更する必要があります。OPTIMIZE(*NO) の指定により、非 IBM i ターゲット・ファイルに対して行うことのできるファイルの順次読み取りを QUERY/38 に強制的に行わせることができます。デフォルトの *YES を使用すると、QUERY/38 アプリケーションを実行したときにエラーが生じます。

同様に、**QUERY アプリケーション設計 (DSNQRYAPP)** コマンドを使用して、非 IBM i ターゲット・ファイルで実行しようとする照会を作成および実行するときには、アプリケーション作成画面上の「照会の最適化」プロンプトを Y から N へ変更する必要があります。

DDM の既存の Query/38 アプリケーションに関する考慮事項:

既存の Query/38 アプリケーションを使用する場合は、以下の考慮事項に留意する必要があります。

既存の QUERY/38 アプリケーションでリモート・ファイルを照会しようとするときには、サーバー・システムが IBM i またはシステム/38 であっても、必ずこれらのアプリケーションを再作成してください。サーバー・システムが IBM i またはシステム/38 である場合、DDM ファイルを使用する再作成されたアプリケーションはあたかもリモート・ファイルがローカル・ファイルであるかのように定義され、実行されます。最適化機能は、ターゲット IBM i やターゲット・システム/38 にあるレコードを取り出すために使用することができます。

IBM i のデータ・ファイル・ユーティリティー:

データ・ファイル・ユーティリティー (DFU) データ入力アプリケーションを作成して DDM を始動すれば、ローカル・ファイルの場合と同じようにリモート・ファイルを処理することができます。ほとんどの DFU 機能は、リモート・ファイルに対しても、ローカル・ファイルに対するのと同様に実行されます。

アプリケーション開発ツールの DFU 機能を作成または変更し、さらにリモート・ファイルが IBM i またはシステム/38 の論理ファイルである場合には、以下の考慮事項のいずれかが適用されます。(1) 基礎となるリモート・ファイルのそれぞれを参照する DDM ファイルが、クライアント・システムに存在し、さらに、DDM ファイル名とライブラリー名が基礎となるリモート・ファイルのファイル名やライブラリー名と一致している必要があります。または、(2) 基礎となるリモート・ファイルと同一のファイル名とライブラリー名、および同一のレコード様式を持つ物理ファイルが、クライアント・システム上に存在している必要があります。物理ファイルのレコード様式のみが必要なため、そのファイルにはデータが入ってなくてもかまいません。いずれの考慮事項でも、基礎となるリモート・ファイルのレコード様式を変更したときには、クライアント・システムでのレコード様式も変更して、レコード様式が一致するようにする必要があります。リモート・ファイルがシステム/36 論理ファイルのときにも、同様の考慮事項が適用されます。

DFU は、IBM i、システム/38、およびシステム/36 のリモート・ファイルをサポートしています。IBM i、システム/38、またはシステム/36 以外のリモート・ファイルが使用できないわけではありませんが、DFU でそのようなファイルを使うと、問題が発生する場合があります。

IBM i 以外またはシステム/36 ファイルはプログラム記述ファイルです。DFU では、ILE RPG を含むローカルまたはリモート・ファイルやこれらのデータ・ファイルを定義するための入力仕様を使用できます。

IBM i データベース QUERY:

IBM i ライセンス・プログラムに備えられている対話式データベース照会機能は、DDM ファイルをサポートします。

このサポートは、OPTIMIZE(*YES) を指定したときに、IBM i Access Family およびシステム/38 互換 QUERY ユーティリティによって使用されます。リモート・ファイルは、QUERY ファイル・オープン (OPNQRYF) コマンドを使って照会することができます。ただし、そのリモート・ファイルがターゲット IBM i またはターゲット・システム/38 にある場合のみです。

システム/38 上で QUERY ユーティリティを使用すると、IBM i オペレーティング・システムに属していないリモート・ファイルを照会することができます。

データベース QUERY を使用すると、複数のローカル・ファイルまたは複数のリモート・ファイルのいずれかに (DDM ファイルを使用して) 同時にアクセスすることができます (両方に同時にはアクセスできません)。

すべてがリモート・ファイルであるファイルの場合、それらのファイルは、同一のサーバー・システムになければなりません。また、それらのリモート・ファイルを参照する DDM ファイルはすべて、同一のリモート・ロケーション情報を指定していなければなりません。この制約事項が守られないと、照会を要求した IBM i Access Family ライセンス・プログラムのユーザーまたは QUERY ファイル・オープン (OPNQRYF) コマンドのユーザーに対して、エラー・メッセージが表示されます。

分類ユーティリティ:

分類ユーティリティは、IBM i および非 IBM i サーバー・システムの両方とも、ローカル・ファイル処理をサポートするところであればどこでも、DDM を用いたリモート・ファイル処理をサポートします。

一般的に、**データ形式設定 (FMTDTA)** コマンドには、データベース・ファイル名を指定できる任意の場所に、DDM ファイル名を指定することができます。

- IBM i またはシステム/38 サーバー・システムの場合、DDM ファイル名は SRCFILE パラメーター、メンバー名は SRCMBR パラメーターに指定できます。DDM ファイルによって参照されるリモート・ファイルが IBM i にもシステム/38 にもなければ、メンバー名を指定することはできません。
- DDM ファイル名の指定は、INFILE パラメーター (変換用の入力ファイルとしてリモート・ファイルにアクセスするため) か、または OUTFILE パラメーター (変換の出力ファイルとしてリモート・ファイルにアクセスするため) にも行うことができます。両方のパラメーターに、DDM ファイル名を同時に指定することはできません。

DDM の IBM i Access Family に関する考慮事項:

IBM i Access Family は、DDM によるリモート・ファイルへのアクセスに対応していますが、使える機能に制限があります。

注: IBM Business Graphics Utility for IBM i では、DDM はサポートされません。

IBM i Access Family にある転送機能を DDM とともに使用すれば、ローカル IBM i 製品に接続されたパーソナル・コンピューターと、別のリモート・システムとの間でデータを転送することができます。

転送機能を使用する場合、リモート・システムは IBM i またはシステム/38 を実行していなければなりません。IBM i Access Family のコピー・コマンド、**PC 文書へのコピー (CPYTOPCD)** コマンド、および **PC**

文書からのコピー (CPYFRMPCD) コマンドを使えば、1 つのホスト・サーバーにあるデータをコピーしたり、複数のホスト・サーバー間でデータをコピーしたりできます。

次の図は、ローカル IBM i プラットフォームに接続されたパーソナル・コンピューターを示しています。IBM i Access Family のユーザーは、ローカル・システムで定義された DDM ファイルを介して、リモート・システムにあるデータにアクセスすることができます。パーソナル・コンピューターが接続された IBM i プラットフォームは、クライアント・システムにしかなり得ません。

- IBM i Access Family の転送機能をパーソナル・コンピューターのユーザーが使えば、リモート・ファイルからパーソナル・コンピューターへ、またはパーソナル・コンピューターからリモート・ファイルへ、データを転送することができます。このような要求を出すことができるのは、パーソナル・コンピューターのユーザーだけであり、IBM i ユーザーから出すことはできません。

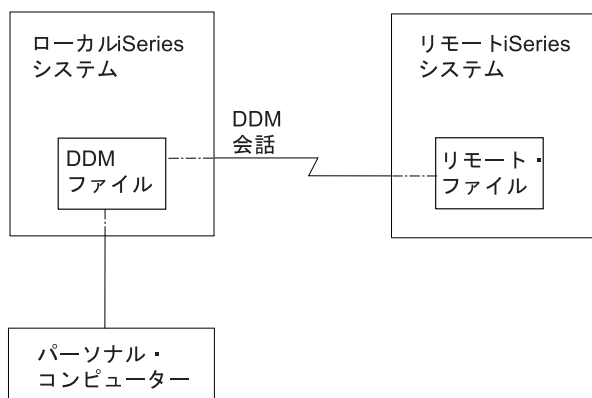


図 26. IBM i Access Family での DDM の使用

- IBM i Access Family のコピー・コマンドを DDM とともに使用すれば、ローカル IBM i プラットフォームにあるパーソナル・コンピューター文書から、リモート IBM i プラットフォームにあるデータベース・ファイルへデータをコピーしたり、リモート IBM i プラットフォームのデータベース・ファイルから、ローカル IBM i プラットフォームにあるパーソナル・コンピューター文書へデータをコピーしたりすることができます。

注: IBM i Access Family の場合、データベース QUERY を使用すれば、複数のリモート・ファイルに同時にアクセスする (DDM ファイルを使用して) ことができます。

IBM i Access Family の転送機能に関する考慮事項:

パーソナル・コンピューターのユーザーは、IBM i Access Family にある転送機能と、パーソナル・コンピューターが接続されているローカル IBM i プラットフォームにある DDM サポートを使用して、パーソナル・コンピューターからリモート・ファイルへ、またはリモート・ファイルからパーソナル・コンピューターへ、データを転送することができます。

リモート・ファイルは、IBM i または システム/38 プラットフォームにあるファイルでなければなりません。

DDM を使用して、リモート・システムから、接続されたパーソナル・コンピューターへファイルやデータを転送するときに、ローカル IBM i プラットフォームにある DDM ファイル (リモート・ファイルを参照するファイル) をローカル・ファイルと結合すると、パーソナル・コンピューターにデータを転送することはできません。(すなわち、リモート・システムとローカル・システムの両方のファイルのデータを結合することはできないということです。) ただし DDM ファイルは、複数の物理ファイルをもとに作成された

論理結合ファイルであるリモート・ファイルを指定することができます。同一のサーバー・システムを参照し、そして同一のリモート・ロケーション情報を使用する複数の DDM ファイルを結合することができます。

ローカル・システムがシステム/38 製品でリモート・システムが IBM i 製品の場合、またはローカル・システムが IBM i 製品でリモート・システムがシステム/38 製品の場合には、グループ処理を必要とする転送要求を出しても、機能しません。

DDM を使用してファイルやデータを、接続されたパーソナル・コンピューターからリモート・システムへ転送するときに、サーバー・システムでリモート・ファイルを作成することはできません。それは、パーソナル・コンピューターからデータを転送するには、リモート・ファイルがすでに存在していなければならないからです。なお、ターゲット・システムは IBM i またはシステム/38 製品でなければならないため、パーソナル・コンピューターのデータをファイル・メンバーに転送する前であれば、新しいメンバーをリモート・ファイルに追加できます。

IBM i Access Family コピー・コマンドに関する考慮事項:

IBM i Access Family で使用される IBM i CL コマンドの「パーソナル・コンピューター文書からのコピー (CPYFRMPCD)」を使用すれば、IBM i にある文書から、DDM を使用して同一の IBM i 上またはリモート IBM i 上にあるデータベース・ファイル・メンバーへ、データをコピーすることができます。

CL コマンドの「パーソナル・コンピューター文書へのコピー (CPYTOPCD)」を使用すれば、(DDM を使って) ローカル IBM i またはリモート IBM i にあるデータベース・ファイル・メンバーから、ローカル IBM i にある文書へ、データをコピーすることができます。リモート・ファイルは、ターゲット IBM i にあっても、または非 IBM i にあってもかまいません。上記のコマンドを使用するには、以下の個所で DDM ファイル名を指定します。

- パーソナル・コンピューター文書を IBM i 物理ファイルにコピーする場合は、**PC 文書からのコピー (CPYFRMPCD)** コマンドの TOFILE パラメーター。
- IBM i データベース・ファイルのメンバーを、フォルダー内のパーソナル・コンピューター文書にコピーする場合は、**PC 文書へのコピー (CPYTOPCD)** コマンドの FROMFILE パラメーター。

IBM i Access Family での CL コピー・コマンドには、次の制約事項があります。

- **CPYFRMPCD** コマンドの場合、リモート・ファイルをサーバー・システムで作成することはできません (ターゲット・サーバーが IBM i か、非 IBM i かに関係なく)。パーソナル・コンピューター文書のデータをコピーする先のリモート・ファイルは、あらかじめ存在していなければなりません。なお、ターゲット・システムが IBM i やシステム/38 の場合は、そのリモート・ファイル用のメンバーを新たに作成してから、パーソナル・コンピューター文書のデータを、そのファイル・メンバーへコピーすることができます。
- **CPYFRMPCD** および **CPYTOPCD** コマンドは IBM i CL コマンドであるため、パーソナル・コンピューターの DOS プロンプトから入力することはできません。

CPYTOPCD および **CPYFRMPCD** コマンドの詳細については、オンライン・ヘルプ情報を参照してください。

DDM の階層ファイル・システム API サポート:

階層ファイル・システム (HFS) API とこれらの API がサポートする機能は、IBM i オペレーティング・システムの一部です。

これらの API によってアプリケーションが、IBM iで利用できるすべての階層ファイル・システムに、単一、かつ一貫性のあるインターフェースを持つアプリケーションを提供します。これらは、自動的に文書ライブラリー・サービス (DLS) ファイル・システムをサポートし、さらにユーザー作成ファイル・システムもサポートすることができます。

DDM は、ユーザー作成のファイル・システムの 1 つとして HFS の下で登録できます。ただし、DDM は、ストリーム・ファイル・コピー (QHFCPYSF) HFS API をサポートするだけです。HFS の下で DDM を登録するには、IBM i クライアント・システム上でコマンド CALL QTSREGFS を実行する必要があります。エラーが発生しなければ、HFS に DDM が正常に登録されています。

HFS QHFCPYSF API を使用して DDM を呼び出すと、2 つの DDM 体系コマンド、つまり **LODSTRF (ストリーム・ファイル・ロード)** コマンドまたは **ULDSTRF (ストリーム・ファイル・アンロード)** コマンドのうち、1 つが生成されます。これらの DDM コマンドの両方ともストリーム・ファイル DDM モデル (STRFIL) の一部です。作業中の DDM サーバー・システムが STRFIL DDM モデルをサポートしていない場合、このサポートを使おうとする時にエラーが発生します。DDM は、ストリーム・ファイル・データを発信 (ULDSTRF の場合)、または受信 (LODSTRF の場合) のいずれかでコピーするためサーバー上で文書とフォルダー (DLS) を使用します。

DDM HFS ストリーム・ファイルのコピー・サポートを使用するには、以下に注意してください。

- DDM がストリーム・ファイルのコピー機能処理するファイル・システムであることを HFS に示すため、ソースおよび宛先ファイル・パス名の両方とも文字列 '/QDDM/' で始めなければなりません。
- コピー情報の HFS パラメーターを DDM は無視しますが、ユーザーは有効な HFS の値をさらに渡さなければなりません。
- ソースまたは宛先ファイル・パス名のパラメーターのいずれかが (両方ではなく) DDM ファイルの名前でなければなりません。使用する DDM ファイルは STRFIL DDM ファイル・モデルをサポートするターゲット・サーバーを指していなければならず、DDM ファイルが別の IBM iを指している場合は、リモート・ファイル名の値が文字列 'FMS' で終わっていなければなりません。
- DDM ファイルではない他のソースまたは宛先ファイルのパス名のパラメーターは、既存の DLS オブジェクト (フォルダー内の文書) の名前ではなければならず、文字列 'FMS' が後に続かなければなりません。
- DDM によってサポートされるソースまたはターゲット・パス名の最大長は、63 文字です。63 文字に '/QDDM/' を含めたり、'FMS' を付加したりすることはできません。
- LODSTRF の場合 (ソース・ファイル・パス名はローカル DLS オブジェクトで、ターゲット・ファイル・パス名は DDM ファイル)、ローカル DLS 文書は、ファイルの相対位置ゼロから終わりまで読み取られます。宛先ファイル (DDM ファイルによって示される) が存在するかどうかは、サーバー・システムのストリーム・ファイル・サポートによって左右されます。
- ULDSTRF の場合 (ソース・ファイル・パス名は DDM ファイルで、宛先ファイル・パス名はローカル DLS オブジェクト)、ローカルまたはターゲット DLS 文書は IBM i 上に存在していなければならず、その内容が消去された後、相対位置ゼロから作成されます。

以下に、リモート・サーバーに **LODSTRF** DDM コマンドを生成する、ストリーム・ファイルのコピーの例を示します。

```
CRTDDMF FILE(DDMLIB/DDMFILE) +  
RMTFILE(*NONSTD 'TARGET/SYSTEM/  
SYNTAX/PATHNAME FMS') RMTLOCNAME(RMTSYSNM)
```

この例では、ローカル DLS オブジェクトは 'PATH1/PATH2/FOLDER1/DOC1' です。

次のパラメーター・リストを使って QHFCPYSF を呼び出します。

```
1 Source file path name = '/QDDM/PATH1/PATH2/FOLDER1/DOC1 FMS'  
2 Source file path name length = 34  
3 Copy information = valid HFS value that is ignored by DDM  
4 Target file path name = '/QDDM/DDMLIB/DDMFILE'  
5 Target file path name length = 20
```

ULDSTRF DDM コマンドを生成するには、ソースおよび宛先ファイルのパス名と長さを逆に指定するだけです。

以下のプログラム例は、DDM HFS API を呼び出します。

注: コードの例を使用することは、443 ページの『コードに関するライセンス情報および特記事項』 の条件に同意したものとされます。

```

/*****
/*****
/* FUNCTION: This program copies a stream file using the QHFCPYSF */
/*           HFS API.                                          */
/*                                                     */
/* LANGUAGE: PL/I                                           */
/*                                                     */
/* APIs USED: QHFCPYSF                                       */
/*                                                     */
/*****
/*****
TRANSFER: PROCEDURE(SRCFIL,TRGFIL) OPTIONS(MAIN);

/* parameter declarations                                     */
DCL SRCFIL CHARACTER (73);
DCL TRGFIL CHARACTER (73);

/* API entry declarations                                    */
/*                                                     */
/* The last parameter, the error code, is declared as FIXED BIN(31) */
/* for the API. This always has a value of zero, specifying that */
/* exceptions should be returned.                               */
DCL QHFCPYSF ENTRY(CHAR(73),FIXED BIN(31),CHAR(6),CHAR(73),
                  FIXED BIN(31),FIXED BIN(31))
                  OPTIONS(ASSEMBLER);

/*****
/* Parameters for QHFCPYSF                                   */
/*****
DCL srclen FIXED BIN(31);
DCL trglen FIXED BIN(31);
DCL cpyinfo CHAR(6);
DCL error_code FIXED BIN(31);

/*****
/* Mainline routine                                         */
/*****

srclen = INDEX(SRCFIL,' ') - 1;
trglen = INDEX(TRGFIL,' ') - 1;
cpyinfo = '1  ';
error_code = 0;
/* Copy the stream file                                     */
Call QHFCPYSF(SRCFIL,srclen,cpyinfo,TRGFIL,trglen,
             error_code);

END TRANSFER;

```

図 27. プログラムの例

前のプログラムで使用できるサンプル・コマンド・ソースを以下に示します。

```

CMD
  PARM  KWD(SRCFIL) TYPE(*CHAR) LEN(73) +
        PROMPT('SOURCE FILE NAME')
  PARM  KWD(TRGFIL) TYPE(*CHAR) LEN(73) +
        PROMPT('TARGET FILE NAME')

```

DDM での CL および DDS の使用

このトピックでは、特定の制御言語 (CL) コマンドについての DDM 関連の情報、データ記述仕様 (DDS) に関する考慮事項、DDS キーワード、および DDM ユーザー・プロファイル権限について説明します。

関連概念:

50 ページの『DDM の計画と設計』

分散データ管理機能 (DDM) を正しく使用するには、いくつかの要件を満たす必要があります。

DDM 固有の CL コマンド:

このトピックでは、DDM 固有の CL コマンドについて説明します。

DDM ファイル変更 (CHGDDMF) コマンド:

DDM ファイル変更 (CHGDDMF) コマンドは、ローカル (クライアント) システムにある DDM ファイルの 1 つ以上の属性を変更します。

DDM ファイルは、IBM i の DDM ネットワーク内の任意のサーバー・システムにあるファイルにアクセスするために、IBM i クライアント・システム上のプログラムによって参照ファイルとして使用されます。

このコマンドを使用するには、次の例に示したコマンドを入力するか、または、DDM ファイル処理の画面でオプション 2 (DDM ファイル変更) を選択します。

例: CHGDDMF コマンド

このコマンドは、クライアント・システムの SOURCE ライブラリーに保管されている SALES という名前の DDM ファイルの通信モードを変更するコマンドです。モードは、MODEX に変更されます。

```
CHGDDMF FILE(SOURCE/SALES) MODE(MODEX)
```

DDM ファイル作成 (CRTDDMF) コマンド:

DDM ファイル作成 (CRTDDMF) コマンドは、ローカル (クライアント) システムで DDM ファイルを作成します。

DDM ファイルは、IBM i の DDM ネットワーク内の任意のリモート (サーバー) システムにあるファイルにアクセスするのに、IBM i のプログラムで参照ファイルとして使用されます。ローカル IBM i のプログラムは、リモート・ファイルの実際の名前ではなく、DDM ファイルの名前でしかリモート・ファイルを認識しません。(ただし DDM ファイル名は、リモート・ファイル名と同じであってもかまいません。)

DDM ファイルは、リモート・サーバーへ CL コマンドを投入するときにも使用します。(リモート・コマンド投入 (SBMRMTCMD) コマンドは、CL コマンドを投入するのに使用しますが、リモート・サーバーは IBM i またはシステム/38 でなければなりません)。SBMRMTCMD コマンドを使用すると、DDM ファイルと通常関連したリモート・ファイルは、無視されます。

DDM ファイルには、アクセスしようとするリモート・ファイルの名前と、そのリモート・ファイルがあるリモート (サーバー) システムを識別するリモート・ロケーション情報が入っています。リモート・ファイル内のレコードにアクセスするのに使用するその他の属性を指定することもできます。

このコマンドを使用するには、次の例に示したコマンドを入力するか、または、DDM ファイル処理の画面で F6 (DDM ファイル作成) を選択します。

例: システム/38 上のファイルにアクセスするための DDM ファイルの作成

```
CRTDDMF FILE(SOURCE/SALES) RMTFILE(*NONSTD 'SALES.REMOTE')  
RMTLOCNAME(NEWYORK)
```

このコマンドは、SALES という名前の DDM ファイルを作成し、そのファイルをクライアント・システムの SOURCE ライブラリーに保管するためのコマンドです。この DDM ファイルは、リモート・ロケーション NEWYORK で、ニューヨークのシステム/38 にある REMOTE ライブラリーに保管されている SALES という名前のリモート・ファイルにアクセスします。

例: IBM i上のファイル・メンバーにアクセスするための DDM ファイルの作成

```
CRTDDMF FILE(SOURCE/SALES) RMTLOCNAME(NEWYORK)
RMTFILE(*NONSTD 'REMOTE/SALES(APRIL)')
```

このコマンドは、前の例と同様の DDM ファイルを作成しますが、ここでは IBM i上の REMOTE ライブラリーに保管されているリモート SALES ファイル内の APRIL という名前のメンバーにアクセスする点が異なります。

例: システム/36 上のファイルにアクセスするための DDM ファイルの作成

```
CRTDDMF FILE(OTHER/SALES) RMTFILE(*NONSTD 'PAYROLL')
RMTLOCNAME(DENVER) LVLCHK(*NO)
```

このコマンドは、SALES という名前の DDM ファイルを作成し、そのファイルをクライアント・システムの OTHER ライブラリーに保管するためのコマンドです。リモート・ロケーション DENVER は、デンバーのシステム/36 にある PAYROLL という名前のリモート・ファイルにアクセスする場合に DDM ファイルが使用します。PAYROLL ファイルと、このファイルへアクセスするアプリケーション・プログラムとの間では、レベル検査は行われません。ACCMTH パラメーターが指定されていないため、サーバー・システムのアクセス方法は、リモート・ファイルにアクセスするために DDM ファイルをオープンする際に、ソース IBM iによって選択されます。

DDM ファイル表示 (DSPDDMF) コマンド:

分散データ管理 (DDM) ファイル表示 (DSPDDMF) コマンドは、DDM ファイルの詳細を表示するためのコマンドです。このコマンドを使用するには、コマンド名をタイプするか、または DDM ファイル作業の画面でオプション 5 (詳細情報の表示) を選択します。

DDM 会話再使用 (RCLDDMCNV) コマンド:

DDM 会話再使用 (RCLDDMCNV) コマンドは、ソース・ジョブで現在使用されていないすべての DDM クライアント・システム会話を再使用するのに使用します。

ジョブの DDMCNV 属性の値が *KEEP であっても、または、このコマンドを活動化グループ内で入力しても、上記のような会話を再使用します。このコマンドを使用すれば、オープン・ファイルをすべてクローズすることなく、また、資源再使用 (RCLRSC) コマンドで実行している他のいずれかの機能を行わなくても、使用されていない DDM 会話を再使用することができます。

RCLDDMCNV コマンドを使用できるのは、このコマンドが入力されたソース・サーバー上のジョブの DDM 会話に対してのみです。ソース・ジョブが使用する各 DDM 会話ごとに、サーバー・システム上にはそれと関連したジョブが 1 つずつあります。該当する関連 DDM 会話が終了すると、ターゲット・ジョブ側も自動的に終了します。

このコマンドは 1 つのジョブで使用されるすべての DDM 会話に適用されますが、このコマンドを使用しても、そのような DDM 会話のすべてが再利用されることにはなりません。ある会話が再使用されるのは、その会話が実際に使用されていないときのみです。

リモート・コマンド投入 (SBMRMTCMD) コマンド:

リモート・コマンド投入 (SBMRMTCMD) コマンドは、コマンドをサーバー・システムで実行するよう、DDM を使って投入します。

DDM ファイル内のリモート・ロケーション情報をもとに、使用すべき通信回線が判別されるので、その投入されたコマンドを受け取ることになるターゲット・サーバーが間接的に識別されることとなります。

SBMRMTCMD コマンドは、以下のうちのどのサーバー・システムへでもコマンドを送るのに使用することができます。

- IBM i
- システム/38
- システム・コマンド投入 (**SBMSYSCMD**) DDM コマンドをサポートするすべてのサーバー

SBMRMTCMD コマンドは、IBM iまたはシステム/38 に、CL コマンド (ただし CL のみ) を送る場合に使用できます。また、サーバー・システムが DDM 体系のシステム投入コマンドをサポートしている場合は、このコマンドを使用して、IBM i またはシステム/38 サーバー以外のサーバー・システムにコマンドを送ることもできます。このコマンドは、サーバー・システムの構文になっていなければなりません。システム/36 サーバーはこの機能をサポートしないため、**SBMRMTCMD** コマンドを使用して システム/36 ターゲットに操作制御言語 (OCL) コマンドを送信することはできません。

このコマンドの第一の目的は、クライアント・システムを使用するユーザーまたはプログラムがファイル管理操作を行ったり、サーバー・システムにあるファイル上のファイル権限認可活動を行えるようにすることにあります。なおユーザーは、このコマンドで処理しようとするサーバー・システム・オブジェクトでの所定の権限を持つユーザーでなければなりません。以下に、**SBMRMTCMD** コマンドを使用してリモート・ファイル上で実行できる操作の例を示します。

- 装置ファイルの作成または削除
- リモート・ファイルに対するオブジェクト権限の認可または取り消し
- ファイルまたはその他のオブジェクトの検査
- ファイルまたはその他のオブジェクトの保管または復元

このコマンドは、ファイルまたはオブジェクトのさまざまな処理に使用できますが、いくつかの処理は他と比較して有用ではありません。たとえば、このコマンドを使用して、リモート・ファイルのファイル記述やフィールド属性を表示したり、または、ファイルやその他のオブジェクトをダンプしたりすることができますが、出力はサーバー・システムに残ります。クライアント・システムでリモート・ファイルのファイル記述やフィールド属性を表示する別の方法として、**ファイル記述表示 (DSPFD)** コマンドや**ファイル・フィールド記述表示 (DSPFFD)** コマンドを用いる方法があります。この場合、SYSTEM (*RMT) パラメーターと、リモート・ファイルに関連した DDM ファイル名を指定します。これで、ユーザーが必要とする情報は、ローカル・サーバーに直接戻ります。

このコマンドの第 2 の目的は、サーバー・システムで、非ファイル操作 (メッセージ待ち行列の作成など) を行ったり、実行しようとするユーザー作成のコマンドを投入したりできるようにすることにあります。CMD パラメーターを使えば、サーバー・システムで実行したいコマンドを表すのに、最大 2000 文字の文字ストリングを指定できます。

関連概念:

24 ページの『DDM の構成要素: DDM ファイル』

DDM ファイルとは、サーバー・システムにあるデータ・ファイルにアクセスするための必要な情報を含むクライアント・システム上のファイルのことです。

SBMRMTCMD コマンドの IBM i およびシステム/38 サーバー・システム:

SBMRMTCMD コマンドは、バッチ環境でも QCAEXEC サーバー・プログラムを使用しても実行可能な、任意の CL コマンドを投入することができます。

つまり、ALLOW 属性に以下の値を両方とも持っているコマンドであれば、**SBMRMTCMD** コマンドを使用して投入できます。

*BPGM

このコマンドは、バッチ入力から呼び出されたコンパイル済み CL プログラム内で処理することができます。

*EXEC

このコマンドを、CALL コマンドのパラメーターとして使用し、文字ストリングとしてサーバー・プログラムに送信し、そこで処理することができます。

これらの可能な値は、**コマンド表示 (DSPCMD)** コマンドを使用して調べることができます。(SBMRMTCMD コマンドは、QCAEXEC または QCMDEXEC システム・プログラムを使用して、投入されたコマンドをサーバー・システム上で実行します。) ただし、これらの許容されるコマンドの中にはサーバー・システムでの介入を必要とするものがあり、予想通りの結果が得られないことがあるため、使用にあたっては、最初に『SBMRMTCMD の制約事項』トピックにリストされた項目を考慮する必要があります。

ユーザーは、投入しようとする CL コマンドと、コマンドで処理しようとするサーバー・システム・オブジェクトの両方で、規定の権限を持ったユーザーでなければなりません。

SBMRMTCMD コマンドの制約事項:

このトピックでは、SBMRMTCMD コマンドの制約事項について説明します。

- リモート・ファイル処理は、2 つの別々のジョブ (各サーバーごとに 1 つずつ) を含むユーザー・ジョブ内では同期処理ですが、サーバー・システムでのファイル処理は、クライアント・システムとは関係なく行われます。**データベース・ファイル一時変更 (OVRDBF)**、**メッセージ・ファイル一時変更 (OVRMSGF)**、および**一時変更削除 (DLTOVR)** などのコマンドは、プログラム・スタック内のプログラムの特定の位置 (反復レベル) または要求レベルに依存するため、予想どおりに機能しないことがあります。

たとえば、クライアント・システムで、各レベルの一時変更を含む複数の反復レベルが発生し、特定レベルの 1 つ以上の一時変更が SBMRMTCMD コマンドでサーバー・システムに投入された場合、サーバー・システム・ジョブはクライアント・システム・ジョブのレベルを知る方法がありません。すなわち、ソース・サーバーで特定の反復レベルの一時変更が終了しても、サーバー・システムの一時変更はまだ有効である場合があります。

- 投入されたコマンドが作成する出力 (スプール・ファイルなど) は、サーバー・システム上にのみ存在します。この出力が、クライアント・システムに送り返されることはありません。
- 一部のタイプの CL コマンドは、ターゲット IBM i に投入してはなりません。以下の項目は、SBMRMTCMD コマンドの本来の目的ではないため、望ましくない結果になる可能性のあるタイプの例です。
 - データベース・ファイル、メッセージ・ファイル、および装置ファイル (通信ファイルと保管ファイルを含む) を参照するすべての OVR_{xxx}F コマンド。
 - すべての DSP_{xxx} コマンド。このコマンドは、出力がサーバー・システムに残ってしまうためです。
 - サーバー・システムのジョブを制御するのに使用する**ジョブ経路再指定 (RRTJOB)** などのジョブ関連コマンド。ただし、**ジョブ変更 (CHGJOB)** コマンドは使用できます。
 - **ジョブ・サービス (SRVJOB)**、**ジョブ追跡 (TRCJOB)**、**内部追跡 (TRCINT)**、**ジョブ・ダンプ (DMPJOB)** などのサービス・プログラムに使用されるコマンド。
 - **印刷装置書出プログラム開始 (STRPRTWTR)** のような、システム・オペレーターに照会メッセージを送る可能性があるコマンド。(代わりに、パススルーを使用できます。)
 - ターゲット・ジョブの独立補助記憶域プール (ASP) を変更しようとするコマンド (例えば、SETASPGRP) は、**リモート・コマンド投入**を使用して発行してはなりません。

- サーバー・システムで作成されるすべての即時メッセージは、そのサーバー上に保管されないため、変換は行われません。すなわち即時メッセージのテキストは、クライアント・システムへ直接送られて表示されます。(他のタイプのメッセージの場合はすべて、サーバー・システムは、メッセージ ID を送り返します。クライアント・システムにあって、そのメッセージ ID に該当するメッセージ・テキストが、表示されるテキストです。このメッセージ・テキストは、もとはクライアント・システム・テキストで変換されたものです。)
- 投入されたコマンドを実行して作成された、最大 10 個のメッセージを、サーバー・システムからクライアント・システムへ送ることができます。10 個を超えるメッセージが作成されると、サーバー・システムのどこ(ジョブ・ログ内など)にそれらのメッセージがあるかを示すため、通知メッセージが追加して送られます。このメッセージのうちの 1 つがエスケープ・メッセージであるときには、最初に他のタイプのメッセージが 9 個送信され、続いて、通知メッセージとエスケープ・メッセージが送信されます。
- サーバー・システムから送られるメッセージのタイプは、完了、通知、診断、およびエスケープの各メッセージのみです。

例: SBMRMTCMD コマンド:

以下の例は、SBMRMTCMD コマンドのさまざまな使用法を示しています。

例: リモート・サーバー上に別の DDM ファイルを作成するためのコマンドの投入

```
SBMRMTCMD CMD('CRTDDMF FILE(SALES/MONTHLY)
RMTFILE(*NONSTD 'SALES/CAR(JULY)')
RMTLOCNAME(DALLAS)') DDMFILE(CHICAGO)
```

上記で投入されたコマンドは、CHICAGO という名前の DDM ファイル内の情報で識別されるサーバー・システムで、MONTHLY という名前の別の DDM ファイルを作成し、この新規 DDM ファイルは、DDMFILE CHICAGO で定義されるサーバー上の SALES という名前のライブラリーに保管されます。CHICAGO サーバー上の新規 DDM ファイルは、DALLAS という名前の別のサーバー上のファイルとメンバーにアクセスするのに使用します。アクセスされるファイルは、ライブラリー SALES 内で CAR という名前が付けられていて、そのファイル内のメンバー名は、JULY です。

この CRTDDMF コマンド・ストリングには、3 組の単一引用符が含まれていることに注意してください。1 組は、投入されるコマンド全体を囲み (SBMRMTCMD コマンドの CMD パラメーターに必要)、2 重の組は、RMTFILE パラメーターで指定されたファイルとメンバーを囲んでいます。*NONSTD を使用すると、1 組の単一引用符で標準外ファイル名を囲む必要があるため、2 番目の組の単一引用符は、2 重にする必要があります。2 番目のは最初の組の単一引用符の中にあるからです。

例: 表示装置ファイルのテキストを変更するためのコマンドの投入

```
SBMRMTCMD CMD('CHGDSPF FILE(LIBX/STANLEY)
TEXT('Don''t forget to pair apostrophes.')
```

このコマンドは、ライブラリー LIBX 内に保管されている STANLEY という名前の表示装置ファイルの記述内のテキストを変更するためのコマンドです。投入されるコマンドは、外側を 1 組の単一引用符で囲む必要があるため (CMD パラメーター用)、ローカル・サーバー処理の TEXT パラメーターに通常必要な単一引用符または 2 重引用符は、リモート・サーバー処理の場合はさらに 2 倍にする必要があります。上記のコーディングでは、リモート・サーバー上で表示または印刷すると、テキスト内に単一引用符が表示されます。

例: リモート・サーバー上のライブラリー・リストを置き換えるためのコマンドの投入

```
SBMRMTCMD CMD('CHGLIBL LIBL(QGPL QTEMP SALES EVANS)')
DDMFILE(EVANS)
```

このコマンドは、EVANS という名前の DDM ファイルに関連したターゲット・ジョブによって使用されるライブラリー・リストのユーザー部分を変更するものですが、その DDM ファイルは、この SBMRMTCMD コマンドを投入しているソース・ジョブが使用しています。そのソース・ジョブでは、リモート・ロケーション情報を指定した DDM ファイル・オープン命令が他にもあれば、その命令にもこのライブラリー・リストが使用されます。

例: 一時変更の使用

SBMRMTCMD コマンドの DDMFILE パラメーターは、コマンド (CMD パラメーター) の送信先のサーバー・システムを決定するのに使用します。この機能では、DDM ファイル (リモート・ファイルではない) に適用される一時変更が考慮されます。たとえば、次のようなコマンドにより DDM ファイルに対するファイル一時変更が指定された場合 (FILEA を FILEX に一時変更)、ファイル削除 (DLTF) コマンドの送付先サーバー・システムは、DDM FILEX で指定されたリモート・ロケーション情報に関連したシステム (この例の場合、値は DENVER システムを指します) になります。

```
CRTDDMF FILE(SRCLIB/FILEA) RMTFILE(SALES/CAR)
RMTLOCNAME(CHICAGO)
CRTDDMF FILE(SRCLIB/FILEX) RMTFILE(SALES/CAR)
RMTLOCNAME(DENVER)
OVRDBF FILE(FILEA) TOFILE(SRCLIB/FILEX)
SBMRMTCMD CMD('DLTF RMTLIB/FRED') DDMFILE(SRCLIB/FILEA)
```

この SBMRMTCMD コマンドは、DENVER サーバーから FRED という名前のファイルを削除します。

その他の考慮事項: SBMRMTCMD コマンド:

このトピックでは、SBMRMTCMD コマンドの追加の考慮事項について説明します。

DDM 会話

コマンドがサーバー・システムで実行される場合、サーバー・システムのジョブが関連付けられます。同一の DDM ファイルと DDM 会話を使用して投入される後続の SBMRMTCMD コマンドは、DDMCNV ジョブ属性の値に応じて、同一または異なるサーバー・システム・ジョブで実行されます。DDMCNV ジョブ属性の値によって、投入された機能の完了後に DDM 会話が停止されるか、活動状態のままになるかが決まります。会話が停止された場合は、次の SBMRMTCMD コマンドは、別のターゲット・ジョブを使用して実行されます。複数のコマンドを投入するには、DDMCNV (*KEEP) が指定されているか、または、DDM の代わりに、表示装置パススルーが使用されていなければなりません。

コマンド構文の検査

クライアント・システムでは、CMD パラメーターで投入されたコマンド文字ストリングの構文の検査は行われません。たとえば、ユーザー定義コマンドの場合、コマンド定義オブジェクトがクライアント・システム上にあることもないこともあります。

コマンド実行の結果

投入されたコマンドは、サーバー・システムのジョブの一部として実行されるため、そのジョブの属性 (ライブラリー検索リスト、ユーザー・プロファイル、待ち時間、および実行優先順位など) が原因で、そのコマンドをローカルで実行したときとは異なる結果が出る場合があります。コマンドを投入するのに困難があり、例えば、その理由がサーバー・システムが異なるライブラリー・リストを使用していることであると分

かった場合、SBMRMTCMD コマンドを使用してライブラリー・リストを編集することができます。

エラー・メッセージの処理

投入されたコマンドの処理中にサーバー・システムがエラーを検出した場合、クライアント・システムは、そのサーバー・システムで作成されたのと同じエラー情報をユーザーに送ることになっています。ただし、サーバー・システムで作成されたものと同等のメッセージがクライアント・システムにない場合には、クライアント・システム・ユーザーに送られるメッセージは、サーバー・システムで作成されたとおりのメッセージ ID を持ち、タイプと重大度も同じになります。エラーのときに送られるメッセージ・テキストは、デフォルト・メッセージ・テキストです。

サーバー・システムが IBM i または システム/36 以外のシステムである場合、クライアント・システムに送られるメッセージには、メッセージ ID もメッセージ・タイプもありません。この場合、そのサーバー・システムから受け取る情報は、メッセージ・テキストと重大度コードのみです。ターゲット・サーバーから大きい値の重大度コードが戻されると、クライアント・システムのユーザーは、SBMRMTCMD コマンドが異常終了したことを示すメッセージを受け取ります。サーバー・システムから送られるその他のメッセージは、メッセージ ID のない通知メッセージとして受信されます。

例えば、ソースとターゲットの両方とも IBM i の場合、ジョブ・ログには、次のようなステートメントが入ります。

```
INFO CPI9155 '以下のメッセージがターゲット・システム上で作成されました。'  
DIAG CPD0028 'ライブラリー ZZZZ が見つかりません。'  
ESCP CPF0006 'コマンドでエラーが発生しました。'
```

IBM i 以外のサーバー・システムから IBM i クライアント・システムにこれと同じメッセージが戻されると、ジョブ・ログは次のようになります。

```
INFO CPI9155 '以下のメッセージがターゲット・システム上で作成されました。'  
INFO nomsgid 'ライブラリー ZZZZ が見つかりません。'  
INFO nomsgid 'コマンドでエラーが発生しました。'ESCP CPF9172 'SBMRMTCMD コマンドが異常終了しました。'
```

サーバー・システムのメッセージをクライアント・システムで表示するには、パススルーと、ジョブ処理 (WRKJOB) またはジョブ・ログ処理 (WRKJOBLOG) コマンドのいずれかを使用します。ターゲット・ジョブが終了すると、メッセージはサーバー・システムの出力待ち行列に入るので、出力待ち行列処理 (WRKOUTQ) コマンドを使用して表示できます。

SBMRMTCMD コマンドを使用してサーバー・システム上の CL プログラムを呼び出す場合、モニターされないエスケープ・メッセージがプログラムによって作成されると、メッセージは照会メッセージに変更されて、システム・オペレーターに送られます。サーバー・システムのオペレーターがこの照会メッセージに応答しなくてもジョブを継続できるようにしたい場合は、サーバー・システムで以下のいずれかを行います。

- 特定ジョブ のデフォルト応答を指定する場合は、ジョブ記述作成 (CRTJOB) またはジョブ記述変更 (CHGJOB) コマンドの INQMSGRPY パラメーターを使用して、ターゲット・ジョブのジョブ記述に *DFT または *SYSRPYL を指定します。SBMRMTCMD コマンドを使用してジョブ変更 (CHGJOB) コマンドをターゲット・サーバーに投入しても、同じことが行えます。
- ジョブ内の特定の照会メッセージ のデフォルト応答メッセージを指定する場合は、応答リスト項目追加 (ADDRPYLE) コマンドを使用して (サーバー・システム上で)、システム全体の自動メッセージ応答リスト (SYSRPYL) に、そのメッセージの項目を追加します。この後、ジョブ記述内で INQMSGRPY (*SYSRPYL) を指定すれば、その照会メッセージがジョブ内で出されるごとに、このデフォルト応答が送られることになります。

独立補助記憶域プール (ASP)

サーバー・システムにオンライン独立 ASP がある場合、ターゲット・ジョブの独立 ASP グループは会話の開始時に確立され、これを変更することはできません。ターゲット・ジョブの独立 ASP グループを変更しようとするユーザー定義コマンドまたは CL コマンド (例えば、SETASPGRP または DLTUSRPRF) は、オンライン独立 ASP を持つサーバー・システムに投入されると失敗します。

DDM ファイル処理 (WRKDDMF) コマンド:

DDM ファイル処理 (WRKDDMF) コマンドを使用すると、リスト画面にある既存 DDM ファイルを処理することができます。リスト画面で、DDM ファイルの変更、削除、表示、または作成を行うことができます。

次の画面では、DDM ファイル作成 (CRTDDMF) コマンドを使用して DDM ファイルを作成済みであると仮定しています。WRKDDMF コマンドを入力して、ライブラリー WILSON とファイル A を指定すると、次のような画面が表示されます。

```

                                DDM ファイル処理

位置指定 . . . . . _____

オプションを入力して、実行キーを押してください。
  1= DDM ファイルの作成   2= DDM ファイルの変更   4= 削除   5= 明細の表示
  6= 明細の印刷

OPT      ローカル・ファイル      リモート・ファイル      リモート・ロケーション
  -      _____              A                          S36
  -

                                終わり

F3= 終了   F5= 最新表示   F9= リストの印刷   F12= 取り消し
```

この画面で DDM ファイルを作成するには、オプション欄に 1 と入力してから、作成したいライブラリーとファイルの名前を入力し、そして実行キーを押します。たとえば、オプション・フィールドに 1 (DDM ファイル作成) と入力し、最初のリスト記入項目のローカル・ファイル欄に WILSON/TEST (次の画面に示されているとおりに) と入力してから、実行キーを押します。DDM ファイル作成画面が表示されます。

DDM ファイル処理

位置指定 _____

オプションを入力して、実行キーを押してください。

- 1= DDM ファイルの作成 2= DDM ファイルの変更 4= 削除 5= 明細の表示
- 6= 明細の印刷

OPT	ローカル・ファイル	リモート・ファイル	リモート・ロケーション
1	WILSON/TEST_____		
-	WILSON/A	A	S36

終わり

- F3= 終了 F5= 最新表示 F9= リストの印刷 F12= 取り消し

DDM ファイル作成 (CRTDDMF)

選択項目を入力して、実行キーを押してください。

DDM ファイル	TEST	名前
ライブラリー	WILSON	名前, *CURLIB
リモート・ファイル:		
ファイル		名前, *NONSTD
ライブラリー		名前, *LIBL, *CURLIB
標準外ファイル ' 名 '		

リモート・ロケーション:
名前またはアドレス

タイプ *SNA *SNA, *IP

続く...

- F3= 終了 F4=プロンプト F5= 最新表示 F10= 追加のパラメーター
- F12= 取り消し F13= この画面の使用法 F24= キーの続き

DDM ファイル作成画面で、必要な値をタイプしてから、表示されているデフォルト値を変更するか、またはそのまま使用します。F10 (追加のパラメーター) を押すと、2つの画面に表示されるコマンド・パラメーターを次々と表示することができます。Page Down (次ページ) キーを押すと、次のような追加パラメーターが表示されます。

DDM ファイル作成 (CRTDDMF)

選択項目を入力して、実行キーを押してください。

テキスト ' 記述 ' *BLANK

追加のパラメーター

装置 :

APPC 装置記述	*LOC	名前, *LOC . .
ローカル・ロケーション	*LOC	名前, *LOC, *NETATR
モード	*NETATR	名前, *NETATR
リモート・ネットワーク識別コード	*LOC	名前, *LOC, *NETATR, *NONE
ポート番号	*DRDA	*DRDA, 1-65535
アクセス方法 :		
リモート・ファイル属性	*RMTFILE	*RMTFILE, *COMBINED...
ローカル・アクセス方法		*BOTH, *RANDOM, *SEQUENTIAL
オープン・データ・パス共用	*NO	*NO, *YES
保護された会話	*NO	*NO, *YES

続く...

F3= 終了 F4=プロンプト F5= 最新表示 F12= 取り消し F13= この画面の使用法
F24= キーの続き

DDM ファイル作成 (CRTDDMF)

選択項目を入力して、実行キーを押してください。

レコード様式レベルの検査	*RMTFILE	*RMTFILE, *NO
権限	*LIBCRTAUT	名前, *LIBCRTAUT, *ALL...
ファイルの置き換え	*YES	*YES, *NO

終わり

F3= 終了 F4=プロンプト F5= 最新表示 F12= 取り消し F13= この画面の使用法
F24= キーの続き

それぞれの値をタイプし終わったら、実行キーを押して、コマンドを処理し、DDM ファイル処理画面に戻ります。

DDM ファイルを変更したい場合には、DDM ファイル処理画面で、変更したいファイルの左側に 2 (DDM ファイルの変更) とタイプするか、または、オプション欄の最初のリスト項目内にオプション番号を入力して、変更したいローカル・ファイルを指定します。たとえば、WILSON/TEST という名前のローカル・ファイルのオプション 欄に、2 (DDM ファイルの変更) をタイプします。

DDM ファイル処理

位置指定 _____

オプションを入力して、実行キーを押してください。
 1= DDM ファイルの作成 2= DDM ファイルの変更 4= 削除 5= 明細の表示
 6= 明細の印刷

OPT	ローカル・ファイル	リモート・ファイル	リモート・ロケーション
-	_____	A	S36
2	WILSON/A WILSON/TEST	TESTFILE/TESTLIB	S38

終わり

F3= 終了 F5= 最新表示 F9= リストの印刷 F12= 取り消し

実行キーを押すと、DDM ファイル変更画面が表示されます。

たとえば、テキスト記述の追加だけ行いたいのであれば、その記述をタイプしてから、実行キーを押します。しかし、他にも変更を行いたい場合には、F10 (追加のパラメーター) を押すと、2 つの画面にわたって表示されるコマンド・パラメーターを次々と表示することができます。

DDM ファイルの変更 (CHGDMMF)

選択項目を入力して、実行キーを押してください。

DDM ファイル	TEST	名前
ライブラリー	WILSON	名前, *LIBL, *CURLIB
リモート・ファイル:		
ファイル	*SAME	名前, *SAME, *NONSTD
ライブラリー		名前, *LIBL, *CURLIB
標準外ファイル ' 名 '		
リモート・ロケーション:		
名前またはアドレス	*SAME	
タイプ	*SAME	*SAME, *SNA, *IP
レコード様式レベルの検査	*SAME	*SAME, *RMTFILE, *NO
		続く...

F3= 終了 F4=プロンプト F5= 最新表示 F12= 取り消し F13= この画面の使用法
 F24= キーの続き

モード・パラメーターを変更したい場合には、その値をタイプしてから、実行キーを押します。

DDM ファイルの変更 (CHGDDMF)

選択項目を入力して、実行キーを押してください。

テキスト ' 記述 ' *SAME

追加のパラメーター

装置 :

APPC 装置記述	*SAME	名前, *SAME, *LOC
ローカル・ロケーション	*SAME	名前, *SAME, *LOC, *NETATR
モード	*SAME	名前, *SAME, *NETATR
リモート・ネットワーク 識別コード	*SAME	名前, *SAME, *LOC, *NETATR...
ポート番号	*SAME	*SAME, *DRDA, 1-65535
アクセス方法 :		
リモート・ファイル属性	*SAME	*SAME, *RMTFILE, *COMBINED...
ローカル・アクセス方法		*BOTH, *RANDOM, *SEQUENTIAL
オープン・データ・パス共用	*SAME	*SAME, *NO, *YES
保護された会話	*SAME	*SAME, *NO, *YES

終わり

F9= すべてのパラメーター F11= キーワード F14= コマンド・ストリング
F24= キーの続き

実行キーを押すと、DDM ファイル処理画面に戻ります。

DDM ファイルの明細を表示したい場合には、DDM ファイル処理画面で、表示したいファイルの左側に 5 (明細の表示) とタイプするか、または、オプション欄の最初のリスト項目内にオプション番号を入力して、表示したいローカル・ファイルを指定します。たとえば、オプション 欄に、5 (明細の表示) を入力してから、最初のリスト記入項目のローカル・ファイル 内に、WILSON/TEST と入力します。

また、DDM ファイル表示 (DSPDDMF) コマンドを使用する方法でも、ファイルの明細を表示できます。

DDM ファイル処理

位置指定 _____

オプションを入力して、実行キーを押してください。

1= DDM ファイルの作成 2= DDM ファイルの変更 4= 削除 5= 明細の表示
6= 明細の印刷

OPT	ローカル・ファイル	リモート・ファイル	リモート・ロケーション
5	WILSON/TEST_____		
-	WILSON/A	A	S36
-	WILSON/TEST	TESTFILE/TESTLIB	S38

終わり

F3= 終了 F5= 最新表示 F9= リストの印刷 F12= 取り消し

実行キーを押すと、DDM ファイルの明細表示画面が表示されます。

DDM ファイルの明細表示

```
ローカル・ファイル :
  ファイル . . . . . : TEST
  ライブラリー . . . . . : WILSON

リモート・ファイル . . . . . : TESTFILE/TESTLIB

リモート・ロケーション:
  名前またはアドレス . . . . . : S38
  タイプ . . . . . : *SNA
  装置記述 . . . . . : *LOC
  ローカル・ロケーション . . . . . : *LOC
  リモート・ロケーション・ネットワーク ID . . . . . : *LOC
  モード . . . . . : S38MODE1
```

続行するためには、実行キーを押してください。

続く...

F3= 終了 F12= 取り消し

ページ送りをして、2 番目の画面を表示します。

DDM ファイルの明細表示

```
アクセス方法
  リモート・ファイル属性 . . . . . : *RMTFILE
  ローカル・アクセス方法 . . . . . :

オープン・データ・パス共用 . . . . . : *NO
レコード様式レベル ID の検査 . . . . . : *RMTFILE
テキスト . . . . . : TEST VERSION FOR DDM
```

続行するためには、実行キーを押してください。

終わり

F3= 終了 F12= 取り消し

実行キーを押すと、DDM ファイル処理画面に戻ります。

DDM ファイルの明細を表示できるだけでなく、オプション 欄で 6 (明細の印刷) をタイプすれば、詳細情報を印刷することもできます。

また、F9 (リストの印刷) を押すと、DDM ファイルのリストを印刷することもできます。

1 つ以上のファイルを削除 したい場合には、削除したいファイルの左側のオプション 欄に 4 (削除) とタイプするか、最初のリスト記入項目内にこのオプション番号をタイプして、削除したいローカル・ファイルを指定します。

DDM ファイル処理

位置指定 _____

オプションを入力して、実行キーを押してください。

1= DDM ファイルの作成 2= DDM ファイルの変更 4= 削除 5= 明細の表示
6= 明細の印刷

OPT	ローカル・ファイル	リモート・ファイル	リモート・ロケーション
-	_____	A	S36
4	WILSON/TEST	TESTFILE/TESTLIB	S38

終わり

F3= 終了 F5= 最新表示 F9= リストの印刷 F12= 取り消し

実行キーを押します。ファイルの削除確認画面が表示されます。

ファイル削除の確認

「4= 削除」の選択項目が正しい場合には、実行キーを押してください。
選択項目を変更するためには、F12 キーを押して戻ってください。

OPT	ローカル・ファイル	リモート・ファイル	リモート・ロケーション
4	WILSON/TEST	TESTFILE/TESTLIB	S38

終わり

F12= 取り消し

画面に示された処置のいずれかを選択して、実行キーを押します。DDM ファイル処理画面に戻ります。

DDM 関連の CL コマンドに関する考慮事項:

これらのトピックでは、DDM ファイルと一緒に使用される場合の CL コマンドについて、DDM 関連の詳細を説明します。以下の各セクションは、クライアント・システムでのコマンドの実行方法について説明したもので、これらのクライアント・システム・コマンドをリモート・コマンド投入 (SBMRMTCMD) コマンドで投入し、サーバー・システム上で実行する場合については述べていません。

注: 以下の事項が DDM ファイルに該当する場合、メッセージ CPF9810 が表示されます。

- ファイルは、ライブラリー QTEMP 内で作成されている。
- ファイルは、CL コマンド (CPYF 等) を使用する。

- リモート・ファイルとライブラリーは、CL コマンド内で指定されたものであり、そのライブラリーは、リモート・サーバーにはない。

メッセージ CPF9810 は、QTEMP ライブラリーが見つからなかったことを示します。ただし、見つからなかったライブラリーは、DDM ファイル内で指定されたりリモート・ライブラリーです。

DDM ファイルのファイル管理機能による処理:

DDM ファイルをデータ管理機能で処理する場合には、コマンド上でメンバー名を指定する際に注意する必要があります。メンバー名を指定すると、データ管理機能はまず、その指定メンバーの入ったローカル・データベース・ファイルを探してから、DDM ファイルを探します。

たとえば、以下を前提とします。

- DDM ファイル CUST021 は、ライブラリー NYCLIB 内にある。
- データベース・ファイル CUST021 は、ライブラリー CUBSLIB 内にある。

ユーザーのライブラリー・リスト内で NYCLIB は、CUBSLIB より前に入っています。CUBSLIB/CUST021 には、メンバー NO1 が入っています。DDM ファイルが指し示すリモート・ファイルには、メンバー NO1 が入っています。次のような一時変更が、データベース・ファイル一時変更 (OVRDBF) コマンドで使用されたとします。

```
OVRDBF FILE(CUST021) MBR(NO1)
```

データ管理機能は、DDM ファイル NYCLIB/CUST021 ではなく、データベース・ファイル CUBSLIB/CUST021 を探索します。

これを防ぐには、以下のうちのいずれかを行います。

- 一時変更において TOFILE を修飾します。
OVRDBF FILE(CUST021) TOFILE(NYCLIB/CUST021) MBR(NO1)
- データベース・ファイルの入ったライブラリーをライブラリー・リストから除去します。
RMVLIBLE LIB(CUBSLIB)
- 一時変更を取り除いてから、DDM ファイル内のリモート・ファイル名を、次のメンバー名を含むよう変更します。
CHGDMMF FILE(NYCLIB/CUST021)
RMTFILE(*NONSTD 'XYZ/CUSTMAST(NO1)')

オブジェクト割り振り (ALCOBJ) コマンド:

クライアント・システムで、オブジェクト割り振り (ALCOBJ) コマンドに DDM ファイル名を指定すると、このコマンドは、クライアント・システムで DDM ファイルを割り振り、また、サーバー・システムでそれに関連したファイルまたはファイル・メンバーを割り振ります。

このコマンドは、DDM ファイルとリモート・ファイルを 1 つずつペアにしてロックします。(これらのファイルは、両方のサーバーでロックされるため、ファイルやメンバーがロックされている間に、変更されたり削除されたりしないようになっています。) ファイル (クライアント・システムにある DDM ファイルと、1 つ以上のサーバー・システムにあるリモート・ファイル) の 1 つ以上のペアを同時に割り振ることができます。

各 DDM ファイルとも、常に共用読み取り (*SHRRD) ロックでロックされます。他のローカル・ファイルをロックするためにコマンド上で同時に指定されているロック・タイプには関係なく、DDM ファイルには共用読み取りが使用されます。

リモート・ファイル にかけるロックは、以下のようにサーバー・システムのタイプによって異なります。

- ターゲット・システムが IBM i やシステム/38 である場合、リモート・ファイルにかけるロックは、ローカル・データベース・ファイルにかけるものと同じになります。つまり、IBM i またはシステム/38 リモート・ファイルも共用読み取りロックでロックされ、そのメンバー (指定された、または最初の) もこのコマンドで指定されたロック・タイプでロックされます。
- ターゲット・システムが IBM i またはシステム/38 でもなければ、リモート・ファイルは指定どおりのロック・タイプでロックされます。例外として、特定の IBM i 以外のターゲット・サーバーでは、コマンドで指定されるものより強力なロックを使用する場合があります。ALCOBJ コマンドに複数の DDM ファイルを指定したときに、1 つ以上のリモート・ファイルが IBM i 以外のターゲット・サーバー上であれば、それらのリモート・ファイルは、コマンドに指定されたロック・タイプでロックされます。メンバーをサポートしないリモート・サーバー用にメンバー名を指定すると、そのメンバー名が DDM ファイル名と同じでない限り、ロック要求は拒否され、エラー・メッセージが出されます。

ALCOBJ コマンドで指定するメンバー名と IBM i ターゲット・サーバー:

ALCOBJ コマンド上で DDM ファイル名とともにメンバー名を指定すると、そのメンバー (リモート・ファイルの) は、コマンド上で指定されたロック・タイプでロックされます。

メンバー名を DDM ファイル自体にも指定するときには、両方のコマンド (ALCOBJ と CRTDDMF) 上でメンバー名が同じでなければなりません。メンバー名が違えば、ロック要求は拒否され、プログラムのユーザーにはエラー・メッセージが送られます。該当メンバーを含むリモート・ファイルの場合、そのメンバー用によどのようなロック・タイプを指定していても、共用読み取りロックでロックされます。

IBM i またはシステム/38 上のリモート・ファイルに対して ALCOBJ コマンドで DDM ファイル名を指定する際にメンバー名を指定しないと、*FIRST がデフォルトになります。サーバー・システムは、名前指定した場合と同様に、リモート・ファイル内の最初のメンバーを探し出してロックしようとします。リモート・ファイルにメンバーがない場合、ロック要求は拒否され、エラー・メッセージが出されます。

ALCOBJ コマンドによる複数の DDM ファイルのロック:

1 つの ALCOBJ コマンドを使って、複数のターゲット・サーバーにあるリモート・ファイルと関連した複数の DDM ファイルを指定することができます。全サーバー上のすべてのファイルをロックできない限り、どのファイルもロックされません。

DDM での ALCOBJ コマンド完了までの時間:

DDM 関連のファイルを割り振るときには、このコマンドを完了するのに要する時間がより長くなります。クライアント・システムとサーバー・システムとの間の通信に余分な時間がかかるためです。

ただし、オブジェクト割り振り (ALCOBJ) コマンドの WAIT パラメーターに指定する待ち時間を長くしてはなりません。通信時間と WAIT パラメーター値には、関連性はないからです。

注: ALCOBJ コマンドでかけたロックを先に解除 (DLCOBJ コマンドを使って) しないで、DLTF コマンドを使ってリモート・ファイルを削除すると、ソース・ジョブが終了しない限り、該当 DDM 会話を再使用することはできません。

ジョブ変更 (CHGJOB) コマンド:

ジョブ変更 (CHGJOB) コマンドを使用して、DDMCNV パラメーターを変更できます。このパラメーターは、DDM 用に割り振られた拡張プログラム間通信機能 (APPC) または IBM i Access Family の会話を、ジョブが使用中でないときに活動状態のまま保つか、自動的に停止するかを制御します。新しい値は、指定したジョブでただちに効力を持ちます。

DDMCNV ジョブ属性の現行値を表示するには、**ジョブ処理 (WRKJOB)** コマンドを使用します。

論理ファイル変更 (CHGLF) コマンド:

論理ファイル変更 (CHGLF) コマンドを使用すれば、SYSTEM パラメーターを介して、クライアント・システムとサーバー・システムにあるファイルを変更することができます。

SYSTEM パラメーター値を使用する場合は、以下の事項を考慮してください。

- *LCL を指定すると、ローカル・サーバーにある論理ファイルが変更されます。
- *RMT を指定すると、リモート・サーバーにある論理ファイルが変更されます。 DDM ファイルの指定は、FILE パラメーター上で行わなければなりません。
- *FILETYPE を指定した場合、FILE パラメーター上に DDM ファイルが指定されている限り、リモート・ファイルが変更されます。DDM ファイルを指定していない場合には、ローカル論理ファイルが変更されます。

このコマンドを DDM で使用するにあたっては、次の事項を考慮してください。

- FILE パラメーターは、変更しようとするリモート論理ファイルを表す DDM ファイル名です。DDM ファイル上で指定されるリモート・ファイルが、リモート・サーバー上で変更される論理ファイル (これも、DDM ファイルに指定されます) です。
- IBM i以外のサーバー・システムの場合:
 - TEXT 以外のすべてのパラメーターは、無視されます。
 - リモート・ファイルが論理ファイルかどうかの検査は行われません。

物理ファイル変更 (CHGPF) コマンド:

物理ファイル変更 (CHGPF) コマンドを使用すれば、SYSTEM パラメーターを介して、クライアント・システムとサーバー・システムにあるファイルを変更することができます。

SYSTEM パラメーター値を使用する場合は、以下の事項を考慮してください。

- *LCL を指定すると、ローカル・システムにある物理ファイルが変更されます。
- *RMT を指定すると、リモート・システムにある物理ファイルが変更されます。 DDM ファイルの指定は、FILE パラメーター上で行わなければなりません。
- *FILETYPE を指定したときに、FILE パラメーターで DDM ファイルが指定されていれば、リモート・ファイルが変更されます。DDM ファイルを指定していない場合は、ローカル物理ファイルが変更されません。

このコマンドを DDM で使用するにあたっては、次の事項を考慮してください。

- FILE パラメーターは、変更しようとするリモート物理ファイルを表す DDM ファイル名です。DDM ファイルに指定するリモート・ファイルが、リモート・システム上で変更される物理ファイル (これも、DDM ファイルに指定されます) です。
- IBM i以外のサーバー・システムの場合:

- EXPDATE、SIZE、および TEXT 以外のすべてのパラメーターが、無視されます。
- リモート・ファイルが物理ファイルかどうかの検査は行われません。

ソース物理ファイル変更 (CHGSRCPF) コマンド:

ソース物理ファイル変更 (CHGSRCPF) コマンドを使えば、SYSTEM パラメーターを介して、クライアント・システムとサーバー・システムにあるファイルを変更することができます。

SYSTEM パラメーター値を使用する場合は、以下の事項を考慮してください。

- *LCL を指定すると、ローカル・サーバーにあるソース物理ファイルが変更されます。
- *RMT を指定すると、リモート・サーバーにあるソース物理ファイルが変更されます。DDM ファイルの指定は、FILE パラメーター上で行わなければなりません。
- *FILETYPE を指定したときに、FILE パラメーターで DDM ファイルが指定されていれば、リモート・ファイルが変更されます。DDM ファイルを指定していなければ、ローカル・ソース物理ファイルが変更されます。

このコマンドを DDM で使用するにあたっては、次の事項を考慮してください。

- FILE パラメーターは、変更しようとするリモート・ソース物理ファイルを表す DDM ファイル名です。DDM ファイルに指定するリモート・ファイルが、リモート・サーバー (これも、DDM ファイルに指定されます) 上で変更されるソース物理ファイルです。
- CCSID パラメーターは、システム/38 のターゲット・サーバーでは無視されます。
- IBM i以外のサーバー・システムでは、ファイルを変更するのに CHGSRCPF コマンドを使用することはできません。

物理ファイル・メンバー消去 (CLRPFM) コマンド:

DDM で物理ファイル・メンバー消去 (CLRPFM) コマンドを使用すると、IBM i サーバー・システム上の物理ファイル・メンバーまたは非 IBM i サーバー・システム上のファイルから、すべてのレコードを消去することができます。このコマンドは、ローカル・ファイルでの場合と同様の働きをします (すべてのデータ・レコードと削除済みレコードを消去します)。

DDM でのコピー・コマンド:

このトピックでは、以下の CL コマンドの DDM での影響について説明します。

- ファイル・コピー (CPYF)
- QUERY ファイルからのコピー (CPYFRMQRYP)
- テープからのコピー (CPYFRMTAP)
- ソース・ファイル・コピー (CPYSRCF)
- テープへのコピー (CPYTOTAP)

上記のコマンドを使用すれば、ローカル・システムとリモート・サーバーにある各ファイル間で、データやソースを相互にコピーすることができます。これらのコマンドを使用して、コピー元とコピー先のファイルを指定します。次の表では、ローカル・サーバーとリモート・サーバー間でコピー可能な、データベース・ファイルと装置ファイルを示します。

表 10. データベースおよび装置ファイルのコピー

コピー元ファイル	コピー先ファイル
ローカルまたはリモート・データベース・ファイル	ローカルまたはリモート・データベース・ファイル

表 10. データベースおよび装置ファイルのコピー (続き)

コピー元ファイル	コピー先ファイル
ローカルまたはリモート・データベース・ファイル	ローカル装置ファイル
ローカル装置ファイル	ローカルまたはリモート・データベース・ファイル

リモート・データベース・ファイルを参照する装置ファイルは、DDM ファイルと見なされます。これらのコピー・コマンドを DDM と一緒に使用する場合は、以下の事項を考慮に入れます。

- コピー・コマンドでエラーになった場合、DDM 会話は、ジョブのために再使用されません。

注: バージョン 3 リリース 2 より前のリリースでは、コピーでエラーになると、**資源再使用 (RCLRSC)** コマンドが実行され、これが**分散データ管理会話再使用 (RCLDDMCNV)** コマンドも実行しました。

RCLRSC コマンドは引き続き実行されますが、コピー・エラーが発生したときに **RCLDDMCNV** コマンドは実行しなくなりました。エラーになったコピー・コマンドの後に明示的に **RCLDDMCNV** が指定されていない限り、DDM 会話は残ります。

- **CPYF** または **CPYSRCF** コマンドで DDM ファイルとローカル・ファイルを指定した場合、サーバーはリモート・ファイルとローカル・ファイルがクライアント・システム上の同じファイルではないことをチェックしません。1 つの DDM ファイルを指定した場合、ユーザーは同じファイルとの間でコピーすることになる可能性があります。
- **CPYF** および **CPYSRCF** コマンドの場合、DDM ファイルは、FROMFILE および TOFILE パラメーターで指定できます。

注: **QUERY** ファイルからのコピー (**CPYFRMQRYP**) およびテープからのコピー (**CPYFRMTAP**) コマンドの場合、DDM ファイル名は、TOFILE パラメーターでのみ指定できます。テープへのコピー (**CPYTOTAP**) コマンドの場合、DDM ファイルは FROMFILE パラメーターでのみ指定できます。

- サーバー・システムが、IBM i または システム/38 でない 場合:
 - ローカル IBM i にあるファイルをリモート・ファイルへコピーする (またはこの逆) には、通常は FMTOPT (*NOCHK) が必須です。
 - ローカル IBM i 上のソース・ファイルをリモート・ファイルにコピーする (または、その逆の) 場合、FMTOPT (*CVTSRC) を指定する必要があります。
- 作成された代替索引のあるターゲット・システム/36 ファイルヘデータをコピーする場合は、MBROPT (*REPLACE) を指定することはできません。指定すると、コピー・コマンドがリモート・ファイルを消去しようとした場合、代替索引があるためその操作は失敗します。
- 削除されたレコードを含む可能性がある IBM i ファイルを削除されたレコードを含む可能性のない IBM i ファイルへコピーするときには、COMPRESS (*YES) を指定しないと、エラー・メッセージが送られ、そのジョブは終了します。
- DDM ファイルにあるリモート・ファイルの名前にメンバー名を指定する場合、コピー・コマンド上でそのファイル用に指定するメンバー名は、DDM ファイル上のリモート・ファイル名にあるメンバー名と同じになっている必要があります。さらに、**データベース・ファイル一時変更 (OVRDBF)** コマンドには、DDM ファイル上のリモート・ファイル名にあるメンバー名とは別のメンバー名を指定することはできません。
- DDM ファイルはメンバー名を指定せず、**OVRDBF** コマンドがファイルのメンバー名を指定している場合、コピー・コマンドは **OVRDBF** コマンドで指定されたメンバー名を使用します。

TOFILE パラメーターで指定された DDM ファイルが、存在しないファイルを参照している場合、CRTFILE (*YES) が指定されていれば、**CPYF** はファイルを作成します。以下に、**CPYF** または **CPYFRMQRYP** コマンドで作成されるリモート・ファイルに対する特別な考慮事項を示します。

- サーバー・システムが IBM i または システム/38 の場合、ターゲット DDM ジョブのユーザー・プロフィールは、サーバー・システムでの **CRTPF** コマンドの使用が許可されていなければなりません。
- サーバー・システムが IBM i 以外のサーバーであれば、**FROMFILE** パラメーターに指定するファイルは、*HEX または ソース・ジョブの **CCSID** 以外のファイルまたはフィールドの **CCSID** を持つことはできません。
- **CPYF** コマンドは、サーバー・システムが IBM i 以外のシステムの場合、**FROMFILE** パラメーターにソース・ファイルを指定することはできません。
- サーバー・システムがシステム/38 である場合にこのコピーを正常実行するには、**TOMBR** パラメーターはリモート・ファイル名と同じか、*FIRST になっている必要があります。このコピーでは、リモート・ファイル名と同じ名前を持ったメンバーが作成されます。
- サーバー・システムがシステム/38 または IBM i 以外である場合にコピーを正常実行するには、**TOMBR** パラメーターを *FIRST にするか、**DDM** ファイル名を指定してください。**DDM** でのリモート・ファイル・アクセスの場合、そのファイルには、**DDM** ファイルと同じ名前を持つメンバーがあるように見えます。
- IBM i サーバー・システムの場合は、**TOFILE** パラメーターは **FROMFILE** パラメーターのすべての属性を持っています。
- IBM i 以外のサーバー・システムの場合、**CRTPF** コマンド上の無視される属性は、コピー・コマンドがファイルを作成する際にも無視されます。
- サーバー・システムがシステム/38 の場合に、**FROMFILE** パラメーターに削除済みレコードを許容しない直接ファイルを指定すると、これらのレコードが最終レコードのうしろにコピーされ、このコピーはファイルの最大サイズに達するまで試行されます。この場合、システム/38 のシステム・オペレーターは、レコードを追加するか、またはコピー命令を取り消すようサーバーに指示します。
- **CRTFILE(*YES)** を指定した **CPYF** または **CPYFRMQRYP** コマンドは、サーバー・システムが許容する大きさのサイズ記述を持つファイルをサーバー・システム上に作成します。
- 全ファイルのコピーの場合、コピーするレコード数がコピー先で許容された最大数を超過していると、最大数に達した時点でコピー機能は終了します。
- バージョン 2 リリース 3 またはそれ以前のシステムがバージョン 3 リリース 1 上のリモート・ファイルを参照するコピー・コマンドは制約関係があるので、制約関係違反により **ERRLVL** パラメーターは有効ではありません。コピーは、**ERRLVL** が指定されたかどうかに関係なく終了します。
- コピー・コマンドによって、リモート分散ファイルを参照する **DDM** ファイルへコピーしたり、**DDM** ファイルからコピーすることができます。

データ域作成 (CRTDTAARA) コマンド:

データ域作成 (CRTDTAARA) コマンドは、データ域を作成し、ライブラリーを指定してそこに作成したデータ域を保管します。データの属性も指定します。任意指定により、このデータ域を特定の値に初期設定することもできます。

DDM データ域を作成するには、**TYPE** パラメーターに *DDM を指定します。**DDM** データ域は、**DDM** ネットワークのリモート (サーバー) システムに保管されているデータ域にアクセスするとき、参照データ域としてプログラムが使用するものです。ローカル (クライアント) システムのプログラムは、リモート・データ域名ではなく、**DDM** データ域名でリモート・データ域を参照します。(DDM データ域名は、リモート・データ域名と同じにすることもできます。)

クライアント・システムの **DDM** データ域には、リモート・データ域の名前とリモート・データ域が保管されるリモート (サーバー) システムの名前が保管されます。

データ域検索 (RTVDTAARA) コマンドおよびデータ域変更 (CHGDTAARA) コマンドで DDM データ域を使用すると、リモート・サーバーのデータ域を検索および更新することができます。DDM データ域は、データ域検索 (QWCRDTAA) API でも使用できます。

このコマンドを DDM で使用するにあたっては、次の事項を考慮してください。

- RMTDTAARA パラメーターは、ターゲット・サーバーのリモート・データ域の名前です。DDM データ域を作成する時点では、データ域を指定しなくてもかまいません。
- RMTLOCNAME パラメーターは、このオブジェクトで使用されるリモート・ロケーションの名前です。複数の DDM データ域が、サーバー・システムに関連した同一のリモート・ロケーションを使用できます。RMTLOCNAME は、リモート・データ域をサポートする IBM i のリリースで稼働している IBM i 製品をサーバー・システムとして指すようにしてください。RMTLOCNAME の可能な値は次のとおりです。
 - *remote-location-name*: サーバー・システムに関連したリモート・ロケーションの名前を指定します。サーバー・システムへのアクセスで使用されるリモート・ロケーションは、DDM データ域が作成される時点では存在しなくてもかまいませんが、DDM データ域がアクセスされる時点では存在しなければなりません。
 - **RDB*: リレーショナル・データベース (RDB) パラメーターに指定されたリレーショナル・データベース・エントリーのリモート・ロケーション情報が、リモート・システムの決定に使用されます。
- DEV パラメーターは、この DDM データ域で使用される、ソース・サーバーの APPC 装置記述の名前です。DDM データ域を作成する時点では、装置記述は指定しなくてもかまいません。
- LCLLOCNAME パラメーターは、ローカル・ロケーション名です。
- MODE パラメーターは、サーバー・システムとの通信を行うときに、リモート・ロケーション名で使用するモード名です。
- RMTNETID パラメーターは、リモート・ネットワーク ID で、この中にはリモート・ロケーションがあり、サーバー・システムとの通信の際に使用されます。

このコマンドを DDM で使用するときには、以下の事項を考慮に入れなければなりません。

- DDM データ域を作成するときの名前として、*LDA、*GDA、または *PDA を指定することはできません。
- データ域をリモートに作成することはできません。リモート・コマンド投入 (SBMRMTCMD) コマンドを使用すれば、この機能をリモート実行できます。
- SBMRMTCMD コマンドを使用すると、データ域をリモート表示することができます。
- データ域表示 (DSPDTAARA) コマンドを使用すると、リモート・データ域の内容を表示することができます。この場合、SYSTEM パラメーターに *RMT を指定します。データ域内のデータは、ローカル・データ域に使用されているのと同じ形式で表示されます。ただし、TEXT フィールドは例外で、これは DDM データ域が作成されたときに提供されるテキスト記述となっています。DDM データ域で SYSTEM パラメーターに *LCL を指定すると、以下に示すものと同様の出力が得られます。

```
Data area . . . . . : DDMDTAARA
Library . . . . . : DDMLIB
Type . . . . . : *DDM
Length . . . . . : 62
Text . . . . . : 'This is a DDM data area'
```

```
Value
Offset *...+...1....+...2....+...3....+...4....+...5
0      '*LOC *NETATR SYSTEMA *LOC *LOC LCLDTAAR'
50     'A LCLLIB '
```

上記の値を解釈するには、次の表を参考にしてください。

表 11. オフセット値

オフセット	DDMDTAARA フィールド
1 から 10	DEV
11 から 18	MODE
19 から 26	RMTLOCNAME
27 から 34	LCLLOCNAME
35 から 42	RMTNETID
43 から 52	RMTDTAARA (名前)
53 から 62	RMTDTAARA (ライブラリー)

データ待ち行列作成 (CRTDTAQ) コマンド:

データ待ち行列作成 (CRTDTAQ) コマンドは、データ待ち行列を作成し、ライブラリーを指定してそこに作成したデータ待ち行列を保管します。データ待ち行列は、ジョブの内部または複数のジョブの間で通信を行ってデータを保管する場合に、複数のプログラムが使用するものです。複数のジョブと単一待ち行列との間でデータの送受信を行うことができます。

任意指定により、CRTDTAQ コマンドで分散データ管理機能 (DDM) データ待ち行列を作成できます。TYPE パラメーターで *DDM を指定すれば、作成できます。DDM データ待ち行列は、ネットワークのリモート (サーバー) システムに保管されているデータ待ち行列にアクセスするときに、参照データ待ち行列としてプログラムが使用するものです。ローカル (クライアント) システムのプログラムは、リモート・データ待ち行列名ではなく、DDM データ待ち行列名でリモート・データ待ち行列を参照します。(なお、DDM データ待ち行列名は、リモート・データ待ち行列名と同じにすることもできます。)

DDM データ待ち行列 (クライアント・システムの) には、リモート・データ待ち行列の名前とリモート・データ待ち行列が保管されるリモート (サーバー) システムの名前が保管されます。

このコマンドを DDM で使用するにあたっては、次の事項を考慮してください。

- TYPE パラメーターは、作成するデータ待ち行列のタイプを指定します。標準のデータ待ち行列または DDM データ待ち行列を作成できます。
- RMTDTAQ パラメーターは、サーバー・システムのリモート・データ待ち行列の名前です。DDM データ待ち行列を作成する時点では、データ待ち行列を指定しなくてもかまいません。
- RMTLOCNAME パラメーターは、このオブジェクトで使用されるリモート・ロケーションの名前です。複数の DDM データ域が、サーバー・システムに関連した同一のリモート・ロケーションを使用できます。RMTLOCNAME は、リモート・データ域をサポートする IBM i のリリースで稼働している IBM i 製品をサーバー・システムとして指すようにしてください。RMTLOCNAME の可能な値は次のとおりです。
 - *remote-location-name*: サーバー・システムに関連したリモート・ロケーションの名前を指定します。サーバー・システムへのアクセスで使用されるリモート・ロケーションは、DDM データ域が作成される時点では存在しなくてもかまいませんが、DDM データ域がアクセスされる時点では存在しなければなりません。
 - *RDB: リレーショナル・データベース (RDB) パラメーターに指定されたリレーショナル・データベース・エントリーのリモート・ロケーション情報が、リモート・システムの決定に使用されます。
- DEV パラメーターは、この DDM データ待ち行列で使用される、クライアント・システムの APPC 装置記述の名前です。DDM データ待ち行列を作成する時点では、装置記述は指定しなくてもかまいません。
- LCLLOCNAME パラメーターは、ローカル・ロケーション名です。

- **MODE** パラメーターは、サーバー・システムとの通信を行うときに、リモート・ロケーション名で使用するモード名です。
- **RMTNETID** パラメーターは、リモート・ネットワーク ID で、この中にはリモート・ロケーションがあり、サーバー・システムとの通信の際に使用されます。

このコマンドを **DDM** で使用するときには、以下の事項を考慮に入れなければなりません。

- **DDM** データ待ち行列の使用時にサポートされるのは、データ待ち行列に対する **API** インターフェースだけです。サポートされる **API** は以下のとおりです。
 - データ待ち行列へ送信 (**QSNDDTAQ**)
 - データ待ち行列から受信 (**QRCVDTAQ**)
 - データ待ち行列の消去 (**QCLRDTAQ**)

データ待ち行列記述検索 (**QMHQRDQD**) **API** およびデータ待ち行列メッセージ検索 (**QMHRDQM**) **API** は、**DDM** データ待ち行列ではサポートされていません。

データ待ち行列へ送信 **API** で ***ASYNCR** パラメーターを使用している場合、リモート・データ待ち行列へのアクセス中に検出したエラーの結果として出されるメッセージは、サーバー・システムのジョブ・ログに置かれ、**DDM** プロトコル・エラー (**CPF9173 - Error detected in DDM data stream by server system**) はクライアント・システムのジョブ・ログに通知されます。エラーの原因をサーバー・システムのジョブ・ログ内で探し、リモート・データ待ち行列を使用する前にその問題を訂正してください。このエラー・メッセージの受信後、最初に問題を訂正せずにリモート・データ待ち行列にアクセスしようとすると、予測不可能な結果が起こります。

- データ待ち行列をリモートに作成することはできません。リモート・コマンド投入 (**SBMRMTCMD**) コマンドを使用すれば、この機能をリモート実行できます。

論理ファイル作成 (**CRTLF**) コマンド:

論理ファイル作成 (CRTLF) コマンドを使用すれば、**SYSTEM** パラメーターを介して、クライアント・システムとサーバー・システムでファイルを作成することができます。

SYSTEM パラメーター値を使用する場合は、以下の事項を考慮してください。

- ***LCL** を指定すると、ローカル・システムでファイルが作成されます。
- ***RMT** を指定すると、リモート・システムでファイルが作成されます。 **DDM** ファイルの指定は、**FILE** パラメーター上で行わなければなりません。
- ***FILETYPE** を指定したときに、**FILE** パラメーターで **DDM** ファイルが指定されていれば、リモート・ファイルが作成されます。 **DDM** ファイルを指定していない場合は、ローカル・ファイルが作成されます。

このコマンドを **DDM** で使用するにあたっては、次の事項を考慮してください。

- **FILE** パラメーターは、作成しようとするリモート論理ファイルを表す **DDM** ファイル名です。 **DDM** ファイルに指定するリモート・ファイルが、リモート・システム上 (これも、**DDM** ファイルに指定されます) で作成される論理ファイルです。
- **OPTION** パラメーターと **GENLVL** パラメーターは、送信されるリモート・コマンドには影響を与えません。
- 論理ファイルの **DDS** 内で **PFILE** キーワードまたは **JFILE** キーワードに指定するファイルは、作成される論理ファイルと同じロケーションになければなりません。

- *JOB がパラメーター値として指定されたり、またはファイル用のデータ記述仕様 (DDS) の中にあるときには、そのソース・ジョブの属性が、ファイル属性およびフィールド属性に使用されます。ファイル属性またはフィールド属性のデフォルトがジョブ属性であるときにも、ソース・ジョブの属性が使用されます。
- IBM i以外のサーバー・システムの場合:
 - 形式名は無視されます。
 - DTAMBR5 パラメーターでは、*ALL の値のみがサポートされます。
 - 以下のパラメーターは無視されます。
 - AUT
 - FRCRATIO
 - FRCACCP5H
 - LVLCHK
 - MAINT
 - MBR
 - RECOVER
 - SHARE
 - UNIT
 - WAITFILE
 - WAITRCD

注: ターゲット・システムがシステム/38 の場合は、これらの属性の変更に **SBMRMTCMD** コマンドを使用できます。

- FMTSLR パラメーターでは、*NONE の値のみがサポートされます。
- FILETYPE は *DATA でなければなりません。
- メンバー名を指定するのであれば、その名前は、DDM ファイル名に一致していなければなりません。
- IBM i サーバー・システムの場合:
 - 権限リストを AUT (共通認可) パラメーターに使用することはできないという制約事項はありますが、**CRTL5** コマンドのすべてのパラメーターがサポートされます。DDM では、サーバー・システムに権限リストが必ずあるとは限らず、また、ユーザー ID リストがある場合でも、同じユーザー ID がそのリストにあるとも、限りません。権限リストを **CRTL5** コマンドの AUT パラメーターの値として使用すると、共通認可は *EXCLUDE に変更されます。
 - DTAMBR5 パラメーターに指定するファイル名は、基礎となるリモート物理ファイルを表す DDM ファイルの名前でなければなりません。DDM ファイルのリモート・ファイル名の一部としてメンバー名を指定していた場合は、そのメンバー名しか指定することはできません。そのメンバー名は、実際のリモート・ファイルのメンバー名でなければなりません。

物理ファイル作成 (CRTPF) コマンド:

物理ファイル作成 (CRTPF) コマンドを使用すれば、SYSTEM パラメーターを介して、クライアント・システムとサーバー・システムでファイルを作成することができます。

SYSTEM パラメーター値を使用する場合は、以下の事項を考慮してください。

- *LCL を指定すると、ローカル・サーバーでファイルが作成されます。

- *RMT を指定すると、リモート・サーバーでファイルが作成されます。DDM ファイルの指定は、FILE パラメーター上で行わなければなりません。
- *FILETYPE を指定したときに、FILE パラメーターで DDM ファイルが指定されていれば、リモート・ファイルが作成されます。DDM ファイルを指定していない場合は、ローカル・ファイルが作成されません。

このコマンドを DDM で使用するにあたっては、次の事項を考慮してください。

- FILE パラメーターは、作成しようとするリモート・ファイルを表す DDM ファイル名です。DDM ファイルに指定するリモート・ファイルが、リモート・サーバー上 (これも、DDM ファイルに指定されず) で作成されるファイルです。
- OPTION パラメーターと GENLVL パラメーターは、ローカル処理の場合と同じ働きをします。これらのパラメーターは、送信されるリモート・コマンドには影響を与えません。
- *JOB がパラメーター値として指定されたり、またはファイル用のデータ記述仕様 (DDS) の中にあるときには、そのソース・ジョブの属性が、ファイル属性およびフィールド属性に使用されます。ファイル属性またはフィールド属性のデフォルトがジョブ属性であるときにも、ソース・ジョブの属性が使用されます。
- IBM i以外のサーバー・システムの場合:
 - 形式名は無視されます。
 - 以下のパラメーターは無視されます。
 - AUT
 - CONTIG
 - DLTPCT
 - FRCRATIO
 - FRCACCPH
 - LVLCHK
 - MAINT
 - MAXMBRS2
 - MBR
 - RECOVER
 - REUSEDLT
 - SHARE
 - UNIT
 - WAITFILE
 - WAITRCD

注: ターゲット・システムがシステム/38 の場合は、これらの属性の変更に **SBMRMTCMD** コマンドを使用できます。

- FILETYPE は *DATA でなければなりません。
- その他のパラメーターは、すべてサポートされます。
- メンバー名を指定するのであれば、その名前は、DDM ファイル名に一致していなければなりません。
- 以下の CCSID 値のみがサポートされます。
 - *HEX

- 65535
- *JOB
- ソース・ジョブのプロセス CCSID

これ以外の CCSID 値を指定すると、ファイルは作成されません。

- DDS キーワードの VARLEN を使用すると、DDM は、サーバー・システムで可変長レコード・ファイルを作成しようとします。このキーワードには、いくつかの特定の規則があります。
- IBM i サーバー・システムでは、CRTPF コマンドのすべてのパラメーターがサポートされますが、次の 1 つの制御事項があります。権限リストを、AUT (共通認可) パラメーターに使用することはできません。DDM では、サーバー・システムに権限リストが必ずあるとは限らず、また、ユーザー ID リストがある場合でも、同じユーザー ID がそのリストにあるとも、限りません。権限リストを CRTPF コマンドの AUT パラメーターの値として使用すると、共通認可は *EXCLUDE に変更されます。

ソース物理ファイル作成 (CRTSRCPF) コマンド:

ソース物理ファイル作成 (CRTSRCPF) コマンドを使用すれば、SYSTEM パラメーターを介して、IBM i のソース・システムとサーバー・システムでファイルを作成することができます。

SYSTEM パラメーター値を使用する場合は、以下の事項を考慮してください。

- *LCL を指定すると、ローカル・システムでファイルが作成されます。
- *RMT を指定すると、リモート・システムでファイルが作成されます。DDM ファイルの指定は、FILE パラメーター上で行わなければなりません。
- *FILETYPE を指定したときに、FILE パラメーターで DDM ファイルが指定されていれば、リモート・ファイルが作成されます。DDM ファイルを指定していない場合は、ローカル・ファイルが作成されます。

このコマンドを DDM で使用するにあたっては、次の事項を考慮してください。

- FILE パラメーターは、作成しようとするリモート・ファイルを表す DDM ファイル名です。DDM ファイルに指定するリモート・ファイルが、リモート・システム上 (これも、DDM ファイルに指定されず) で作成されるファイルです。
- OPTION パラメーターと GENLVL パラメーターは、ローカル処理の場合と同じ働きをします。これらのパラメーターは、送信されるリモート・コマンドには影響を与えません。
- *JOB がパラメーター値として指定されたり、またはファイル用のデータ記述仕様 (DDS) の中にあるときには、そのソース・ジョブの属性が、ファイル属性およびフィールド属性に使用されます。ファイル属性またはフィールド属性のデフォルトがジョブ属性であるときにも、ソース・ジョブの属性が使用されます。

権限リストを AUT (共通認可) パラメーターに使用することはできないという制約事項はありますが、CRTSRCPF コマンドのすべてのパラメーターがサポートされます。DDM では、サーバー・システムに権限リストが必ずあるとは限らず、また、ユーザー ID リストがある場合でも、同じユーザー ID がそのリストにあるとも、限りません。権限リストを CRTSRCPF コマンドの AUT パラメーターの値として使用すると、共通認可は *EXCLUDE に変更されます。

オブジェクト割り振り解除 (DLCOBJ) コマンド:

クライアント・システムで、オブジェクト割り振り解除 (DLCOBJ) コマンドに DDM ファイル名を指定すると、このコマンドは、クライアント・システムで DDM ファイルを割り振り解除し、また、サーバー・システムでそれに関連したファイルまたはファイル・メンバーを割り振り解除します。

このコマンドは、クライアント・システムとサーバー・システムの両方において、**オブジェクト割り振り (ALCOBJ)** コマンドでかけられた各ファイル・ペアのロックを解放します。1 つ以上のファイルの対 (クライアント・システムにある **DDM** ファイルと、1 つ以上のサーバー・システムにあるリモート・ファイル) を、同時に割り振り解除することができます。

DLCOBJ コマンドのメンバー名と *IBM i* ターゲット・サーバー:

前述の **ALCOBJ** コマンドの説明でメンバー名に関して述べた事項が、すべて **DLCOBJ** コマンドにも適用されます。

詳細については、**ALCOBJ** コマンドの説明を参照してください。

DLCOBJ コマンドによる複数の **DDM** ファイルのアンロック:

1 つの **DLCOBJ** コマンドを使って、複数のサーバー・システムにあるリモート・ファイルと関連した複数の **DDM** ファイルを指定することができます。たいいていの場合、このコマンドは、できる限り多数の指定ロックを解放しようとします。

以下にその例を示します。

- **DLCOBJ** コマンドに指定した **DDM** ファイルのうちのいずれかが、データベース・ファイルではないリモート・ファイルを参照していると、該当ロックは解放されません。しかし、その他の指定 **DDM** ファイルに関連したリモート・ファイルにかけられた指定ロックは、当然のことながら、有効でありさえすれば解放されます。
- あるロックを解放しようとしたときに、それが以前に **ALCOBJ** コマンドを使ってファイルにかけたものではなかった場合、要求のうちのその部分は拒否され、通知メッセージがユーザーに戻されます。

ファイル削除 (**DLTF**) コマンド:

ファイル削除 (DLTF) コマンドを使用すれば、クライアント・システムとサーバー・システムにあるファイルを削除することができます。

SYSTEM パラメーター値を用いるときには、以下の事項を考慮に入れてください。

- ***LCL** を指定すると、ローカル・ファイルのみが削除されます。その中には、**DDM** ファイルが含まれることもあります。
- ***RMT** を指定すると、リモート・サーバーにあるファイルが削除されます。**DDM** ファイルの指定は、**FILE** パラメーター上で行わなければなりません。総称名を指定すると、その総称名に一致するすべての **DDM** ファイルに対応したリモート・ファイルが削除されます。(ローカル **DDM** ファイルは削除されません。)
- ***FILETYPE** を指定したときに、**DDM** ファイルが指定されていれば、そのリモート・ファイルが削除されます。**DDM** ファイルを指定していない場合は、ローカル・ファイルが削除されます。総称名を指定すると、**DDM** ファイルでないローカル・ファイルから先に削除されます。その後で、総称名に一致した **DDM** ファイルに対応するリモート・ファイルが削除されます。ローカル **DDM** ファイルは削除されません。

注:

1. 構造化照会言語 /400 (SQL/400) の **DROP TABLE** ステートメントと **DROP VIEW** ステートメントは、ローカル・ファイルに対してのみ有効です。
2. **ALCOBJ** コマンドでかけたロックを先に解除 (**DLCOBJ** コマンドを使って) しないで、**DLTF** コマンドを使ってリモート・ファイルを削除すると、ソース・ジョブが終了しない限り、該当 **DDM** 会話を再使用することはできません。

ファイル記述表示 (DSPFD) コマンド:

ファイル記述表示 (DSPFD) コマンドを使用すれば、クライアント・システムにある DDM ファイルや、サーバー・システムにあるリモート・ファイル、または DDM ファイルとリモート・ファイルの両方の属性を表示する (クライアント・システムで) ことができます。ローカル・ファイルの場合と同様に、複数の DDM ファイル、複数のリモート・ファイル、またはその両方の属性を 1 つの同じコマンドで表示することができます。

注: この説明は、1 つだけのサーバー・システムについて述べていますが、複数のサーバー・システムのファイルを同時に表示することができます。

SYSTEM パラメーターで、どの属性グループを表示するかが決まります。

- DDM ファイルがローカル・ファイル のときに、そのファイルの属性を表示するには、SYSTEM パラメーターに *LCL (デフォルト) を指定する必要があります。SYSTEM (*LCL) を指定する場合、
 - FILEATR パラメーターには、*DDM (DDM ファイル属性のみを表示) か、またはデフォルトの *ALL (DDM ファイルを含め、すべてのファイル・タイプを表示) のいずれかを指定する必要があります。DDM ファイル (ローカル・システムにあるもの) の場合も、ローカル・サーバーにある他のすべてのファイル・タイプの場合と同じ種類の情報が表示されます。
 - FILEATR (*DDM) を指定したときに、OUTFILE パラメーターにファイル名を指定すると、ローカル DDM ファイル情報のみが表示されます。
- リモート・ファイルの属性を表示するには、SYSTEM パラメーターに *RMT を指定する必要があります。SYSTEM (*RMT) を指定する場合、
 - FILEATR パラメーターには、*ALL、*PHY、または *LGL を指定する必要があります。
 - リモート・ファイルの場合に表示される情報のタイプは、どのタイプのサーバー・システムにそのファイルがあるかによって異なります。ターゲット・システムが IBM i または システム/38 である場合は、IBM i または システム/38 にあるローカル・ファイルの場合に表示されるのと同じタイプの情報を表示することができます。ターゲット・システムが IBM i でも システム/38 でもない 場合には、IBM i のインプリメンテーションとの互換性がある DDM 体系のサーバー・インプリメンテーションを介して入手できるすべての情報が表示されます。
- DDM ファイルとリモート・ファイルの両方の属性を表示するには、SYSTEM パラメーターに *ALL を指定する必要があります。

ファイル・フィールド記述表示 (DSPFFD) コマンド:

ファイル・フィールド記述表示 (DSPFFD) コマンドを使用すれば、リモート・ファイルのファイル、レコード様式、およびフィールドの各属性を表示することができます。ただし、リモート・ファイルの属性を表示するには、そのリモート・ファイル名ではなく、そのリモート・ファイルに関連した DDM ファイル名を入力する必要があります。

注: DDM ファイルにはフィールド属性はないため、ローカル DDM ファイル情報を表示するのに、DSPFFD コマンドに SYSTEM (*LCL) を指定することはできません。

FILE パラメーターに *ALL または総称ファイル名を指定すれば、DSPFFD コマンドで、ローカル・ファイルとリモート・ファイルの両方のグループに関する情報を表示したり、ローカル・ファイル・グループだけの情報を表示することもできます。この場合、SYSTEM パラメーターで、いずれの表示になるかが決まります。

- DDM ファイル以外のローカル・ファイルのみの属性を表示するには、*LCL がデフォルトであるため、SYSTEM パラメーターを指定する必要はありません。

- リモート・ファイルの属性を表示するには、SYSTEM パラメーターに *RMT を指定する必要があります。SYSTEM (*RMT) を指定すると、表示されるリモート・ファイルのフィールド様式とレコード様式の情報は、そのファイルがあるサーバー・システムのタイプによって異なります。
 - ターゲット・システムが IBM i または システム/38 の場合、IBM i にあるローカル・ファイルの場合に表示されるのと同じ情報が表示されます。
 - ターゲット・システムがシステム/38 でも IBM i でもない場合、
 - フィールドは、ファイルがキー付きファイルかどうかに応じて、Fnnnnn または Knnnnn (nnnnn は任意の数) になります。
 - レコード様式名は、DDM ファイル名です。

リモート・ファイルのレコード長クラスが、可変長レコード、または最初は可変長になっている場合は、固定長フィールド記述が表示されます。

- ローカル非 DDM ファイルおよびリモート・ファイルの両方の属性を表示するには、SYSTEM パラメーターに *ALL を指定する必要があります。リモート物理ファイルとリモート論理ファイルだけを表示することができます。

Query ファイル・オープン (OPNQRYP) コマンド:

リモート・ファイルは、QUERY ファイル・オープン (OPNQRYP) コマンドを使って照会することができます。ただし、そのリモート・ファイルがターゲット IBM i またはターゲット・システム/38 にある場合のみです。

複数のリモート・ファイルを 1 つの OPNQRYP コマンドに指定するときには、それらのファイルはすべて、同一のサーバー・システムになければならず、また、同一のリモート・ロケーション情報を使用するものでなければなりません。

サーバー・システムが IBM i または システム/38 の場合、QUERY 要求は、その QUERY が参照する DDM ファイルを使用して作成されサーバー・システムに送信されます。サーバー・システムが IBM i または システム/38 以外の場合、QUERY 要求は処理できず、エラー・メッセージが出されます。なお、システム/38 の照会ユーティリティーを使用すれば、IBM i ファイル以外のリモート・ファイルを照会できます。

サーバー・システムがシステム/38 でソース・システムが IBM i の場合、またはサーバー・システムが IBM i でソース・システムがシステム/38 の場合、OPNQRYP でグループ別および結合機能を使用することはできません。使用した場合は、エラーになります。

データベース・ファイル一時変更 (OVRDBF) コマンド:

データベース・ファイル一時変更 (OVRDBF) コマンドを DDM とともに用いれば、プログラム内で指名したローカル・データベース・ファイルを、DDM ファイルに一時変更 (置換) することができます。DDM ファイルによって、その関連リモート・ファイルが、ローカル・データベース・ファイルの代わりに、プログラムによって使用されることとなります。

TOFILE パラメーターに DDM ファイルを指定したときに、ファイルの属性を変更するような他のパラメーターを指定すると、実際にプログラムが使用するリモート・ファイルは、OVRDBF コマンドのパラメーター値で変更された属性とともに使用されることとなります。

サーバー・システムが IBM i または システム/38 の場合、既存のプログラムが OVRDBF コマンドでリモート・ファイルにアクセスすると、ローカル・ファイルにアクセスするときと同様に機能します。OVRDBF パラメーターはすべて、IBM i のソース・システムとターゲット・システムで同様に処理されます。

ファイル終わり遅延 (EOFDLY) を使用する場合は、ファイルの終わりレコードを使ってジョブを終了することをお勧めしますが、その理由は、ソース・ジョブが取り消されても、ターゲット・ジョブには知らされないからです。ユーザーは、ターゲット・ジョブも終了する必要があります。

サーバー・システムが IBM iでもシステム/38 でもない場合:

- TOFILE、POSITION、RCDFMTLCK、WAITFILE、WAITRCD、LVLCHK、EXPCHK、INHWRT、SECURE、SHARE、および SEQONLY パラメーターは依然有効です。
 - TOFILE パラメーターは常に、クライアント・システムで処理されます。このパラメーター上で DDM ファイル名を指定すると、プログラムは、プログラム内で指定されたローカル・データベース・ファイルの代わりに、DDM ファイルに関連したリモート・ファイルを使用します。
 - RCDFMTLCK パラメーターを指定した場合、使用するリモート・ファイルに関して以下の両方の条件が満たされる場合に限り、このパラメーターは有効です。それらの条件とは、(1) リモート・ファイル用には 1 つのタイプのロック状態しか要求することができない、(2) リモート・ファイル内のレコード様式名は、DDM ファイル名と同じになっている必要がある、というものです。
 - WAITFILE パラメーターと WAITRCD パラメーターは、リモート・ファイルの処理には効力はありません。
- メンバー名を使って MBR パラメーターを指定した場合、そのメンバー名がメンバーの入ったファイル名と異なっていれば、エラーの原因になります。
- FRCRATIO パラメーターと NBRRCDS パラメーターは、指定しても無視されます。
- FMSTLR パラメーターを指定したときに、オープンしようとするファイルが DDM ファイルであればエラーとなります。
- SEQONLY パラメーターを使用すると、起動側にあるレコードがブロック化されることとなります。ブロックがいっぱいになる前にソース・ジョブが取り消されると、レコードが失われることがあります。

リソース再利用 (RCLRSC) コマンド:

資源再使用 (RCLRSC) コマンドを使用すれば、DDM 会話再使用 (RCLDDMCNV) コマンドの場合と同様に、ジョブ内でユーザーが現在使用していないすべての DDM 会話を再使用することができます。

再使用は、DDMCNV ジョブ属性が *KEEP であっても可能です。ただし、RCLRSC コマンドでは、ローカル・ファイルの場合と同様に、適切な反復レベルごとに未使用ファイルがあれば、まずそれがクローズされることとなります。この処置で、DDM に割り振られた会話のうち特定のものは、そのジョブで使用できなくなる場合があります。例えば、データベース・ファイル・オープン (OPNDBF) コマンドを使って DDM ファイルがオープンされていると、RCLRSC コマンドは、そのファイルをクローズしてから会話を再使用します。

ファイルがクローズされた後に未使用 DDM 会話があれば、それはすべて停止されます。ある会話を再使用できるかどうかは、RCLRSC コマンドが出されたときの反復レベルや活動化グループによって影響されることはありません。

オブジェクト名変更 (RNMOBJ) コマンド:

オブジェクト名変更 (RNMOBJ) コマンドを使用すれば、リモート・ファイル名を変更することができます。

SYSTEM パラメーター値を用いるときには、以下の事項を考慮に入れてください。

- *LCL を指定すると、ローカル・オブジェクトが名前変更されます。その中には、DDM ファイルが含まれることもあります。

- *RMT を指定した場合、その値を用いることができるのは、OBJTYPE (*FILE) に対してのみです。名前を変更しようとするリモート・ファイルの入った DDM ファイルは、OBJ パラメーター上で指定します。

そのリモート・ファイルの新しい名前の入った DDM ファイルは、NEWOBJ パラメーター上で指定します。いずれの DDM ファイルも、あらかじめ同一のライブラリー (クライアント・システムにある) 内になければなりません。この 2 つの DDM ファイルは、同じサーバー・システムを参照し、同じリモート・ロケーション情報を持っていないければなりません。この 2 つのローカル DDM ファイル名も、2 つの DDM ファイル内の RMTFILE 名も変更されません。DDM ファイル名を変更するには *LCL を指定し、DDM ファイル内の RMTFILE 名を変更するには **DDM ファイル変更 (CHGDMMF)** コマンドを使用します。

- *FILETYPE を指定した場合、その値が使用されるのは OBJTYPE (*FILE) の場合に限られます。OBJ パラメーターに指定したファイルが DDM ファイルであるときには、*RMT の指定に関して定められた規則が適用されます。そのファイルが DDM ファイルではない場合、*LCL の指定に関して定められた規則が適用されます。

IBM i およびシステム/38 ターゲット・システムのリモート・ファイル名を変更する場合、2 つの DDM ファイルのライブラリー名を RMTFILE パラメーター内で指定したのであれば、それらのライブラリー名は同じになっている必要がありますが、ファイル名は異ならないければなりません。

ジョブ処理 (WRKJOB) コマンド:

ジョブ処理 (WRKJOB) コマンドを使用すれば、以下の 2 つの DDM 関連項目を表示することができます。

項目は、以下のものです。

- ソース・ジョブの DDMCNV ジョブ属性。
- クライアント・システム・ジョブで使用されている DDM ファイル用のオブジェクト・ロック要求 (現在有効なロックと保留中ロック)。これは、ジョブ処理メニューでオプション 12 (活動中の場合のロック処理) を選択すれば表示されます。

ジョブ・ロック画面には、ローカル DDM ファイルに対して保持されているロックだけが表示され、リモート・ファイルに対するロックは表示されません。また、DDM ファイルにはメンバーはないため、この画面にもメンバー・ロック画面にもメンバーが示されることはありません。

IBM i オペレーティング・システムは、リモート・ファイルに対するロックは表示しません。リモート・ファイルに対するロック、そのメンバーに対するロック、またはそのレコードに対するロックを、クライアント・システムで表示することはできません。なお、このようなリモート・ロックでも、パススルーを使用すれば表示できます。

表示される DDM ファイルのロック状態は、その関連リモート・ファイルやメンバーに使用されるロック状態がどのようなものであっても、常に共用読み取り (*SHRRD) になります。

オブジェクト・ロック処理 (WRKOBJLCK) コマンド:

オブジェクト・ロック処理 (WRKOBJLCK) コマンドを使用すれば、DDM ファイルに関するオブジェクト・ロック要求 (現在有効なロックと保留中ロック) を表示することができます。このコマンドは、ローカル DDM ファイルの関連リモート・ファイルに対して保持されているロックではなく、その DDM ファイルに対して保持されているロックのみを表示します。

IBM iは、リモート・ファイルに対するロックは表示しません。リモート・ファイルに対するロック、そのメンバーに対するロック、またはそのレコードに対するロックを、クライアント・システムで表示することはできません。

表示される DDM ファイルのロック状態は、その関連リモート・ファイルやメンバーに使用されるロック状態がどのようなものであっても、常に共用読み取り (*SHRRD) になります。

DDM 関連の CL パラメーターに関する考慮事項:

このトピックでは、DDM 関連の CL コマンドに適用されるパラメーターに関する考慮事項を示します。

注: DDM ファイルを作成するには、**DDM ファイル作成 (CRTDDMF)** コマンドを使用します。他のファイル作成コマンド **CRTPF** や **CRTxxxF** 等を使用しても、DDM ファイルを作成することはできません。

DDMACC パラメーターに関する考慮事項:

DDMACC パラメーターは、サーバー・システムとしての IBM iが、他のサーバーから出された DDM 要求をどのように処理するかを制御します。

DDMACC パラメーターは、**ネットワーク属性変更 (CHGNETA)** コマンド、**ネットワーク属性表示 (DSPNETA)** コマンド、および**ネットワーク属性検索 (RTVNETA)** コマンド上で使用するパラメーターです。このサーバー・レベル・パラメーターの値で、この IBM iが、他のサーバーから DDM 要求を受け入れられるかどうかが決まります。

DDMCNV パラメーターに関する考慮事項:

DDMCNV パラメーターはジョブ関連のパラメーターであり、これは、DDM で使用するためにジョブ内で割り振られた拡張プログラム間通信機能 (APPC) 会話や、IBM i 会話 (つまり、DDM 会話) を、ソース・ジョブで自動的に停止するか、または活動状態のままにするかを制御するものです。

デフォルトでは、会話は活動状態のまま保たれます。

このパラメーターは、活動ユーザーのいない会話を停止することができます。以下のような場合に、会話は使用されていません。

1. 会話で使用する DDM ファイルとリモート・ファイルがすべてクローズされ、アンロックされた (割り振り解除された) 場合。
2. 他の DDM 関連機能 (サーバー・システムにアクセスするための**リモート・コマンド投入 (SBRMTCMD)** コマンドや**ファイル記述表示 (DSPFD)** コマンドなど) が実行されていない場合。
3. DDM 関連機能が実行中に中断 (たとえば、中断プログラムで) されていない場合。

DDMCNV パラメーター値は以下のとおりです。

***KEEP**

ソース・ジョブ用に開始された各 DDM 会話を、ソース・プログラム要求が完了しても活動状態のままにし、同じプログラム、またはそのジョブ内で実行される別のプログラムから別の要求が来るまで待機させることを指定します。これはデフォルト値です。

***DROP**

以下の 2 つの項目が両方とも満たされたときには、ソース・ジョブ内で開始された各 DDM 会話を自動的に停止するよう指定します。(1) そのジョブ内で実行されているクライアント・システ

ム・プログラムからの要求で、その会話内で処理中のものがなく、また、(2) すべての DDM ファイルがクローズされ、割り振られていたすべてのオブジェクトはすでに割り振り解除されている場合。

DDMCNV パラメーターは、**ジョブ変更 (CHGJOB)** コマンドで変更し、**ジョブ処理 (WRKJOB)** コマンドで表示します。また、**ジョブ属性検索 (RTVJOB)** コマンドを使用すれば、このパラメーターの値を得て、それを CL プログラム内で使用することができます。

DDM の OUTFILE パラメーターに関する考慮事項:

OUTFILE パラメーターは、**ファイル記述表示 (DSPFD)**、**ファイル・フィールド記述表示 (DSPFFD)**、**オブジェクト記述表示 (DSPOBJD)**、および**報告書簡易作成プログラム作成 (CRTTRPTPGM)** などのコマンド上で使用します。このパラメーターは、このコマンドで作成される出力データが保管されるデータベース・ファイルを識別します。

DDM ファイル名を上記のコマンドの OUTFILE パラメーターに指定するときには、以下の 2 とおりの制約を受けます。

- リモート・サーバーは、IBM i または システム/38 でなければなりません。この条件は、関連したリモート・ファイルの形式が、必ず出力データに合った正しい形式になるようにするためのものです。
- DDM ファイルに関連したリモート・ファイルが、あらかじめ存在していなければなりません。リモート・ファイルが存在していないと、メッセージがユーザーに返され、この機能を実行するには、リモート・ファイルが存在していなければならないことが示されます。

OUTFILE パラメーターに指定されたリモート・ファイルが確かに存在し、それが IBM i または システム/38 にあっても、以下の 3 つの条件を満たしているかどうかを検査してからでなければ、このファイルを、表示された出力を保管するための出力データベース・ファイルとして使用することはできません。

- リモート・ファイルは、物理ファイルでなければなりません。
- リモート・ファイルは、モデル出力ファイルであってはなりません。つまり、このファイルは、IBM i オペレーティング・システムで提供された、規定どおりの様式ではあるが、データの入っていないモデル出力ファイルのうちのいずれかであってはなりません。
- リモート・ファイル内のレコード様式名は、モデル出力ファイルのレコード様式名と一致していなければなりません。(この条件を満たすには、リモート・システムが IBM i または システム/38 である必要があります。)

上記の条件をすべて満たせば、リモート・ファイル・メンバーが消去されます。(出力ファイル・メンバーをもう一度使用できるようにするには、まずそのメンバーを消去する必要があります。) リモート・ファイル・メンバーが存在していない場合には、そのメンバーが作成されてその中に出力が保管されます。

DDM 関連の CL コマンド・リスト:

このトピックでは、DDM と特に関連する制御言語 (CL) コマンドを、グループに分けて表にしてあります。表の中には、DDM で使用できるコマンド機能、DDM で使うと共通の制約が課せられるコマンド機能、および DDM とともに使用できないコマンド機能を示しています。

注:

1. このトピックでは、IBM i の CL コマンドをすべて示すわけではありません。ここでは、IBM が DDM の使用対象とする (または、推奨する) コマンド、または DDM に対して特別に非対象とするコマンドについてのみ示します。使用対象となるのは、クライアント・システム上で実行してサーバー・シス

テムのリモート・ファイルに影響を与えるコマンド、またはリモート・コマンド投入 (SBMRMTCMD) コマンドを使用してサーバー・システムに実行依頼するコマンドのいずれかです。

2. 複数の表に重複して出てくるコマンドもあります。

このトピックの表は、以下のものを示しています。

- DDM ファイルにのみ影響を与えるコマンド。

DDM ファイルとともに使用することはできるが、ただしそのファイルと関連したリモート・ファイルには影響を与えないオブジェクト指向コマンド。このコマンド・グループには、DDM ファイル作成 (CRTDDMF) コマンド、DDM ファイル変更 (CHGDDMF) コマンド、DDM 会話再使用 (RCLDDMCNV) コマンドなどがあります。

- DDM ファイルとリモート・ファイルの両方に影響を与えるコマンドは以下のものです。
 - サーバー・システムが IBM iまたはシステム/38 であることが必要なファイル管理コマンド。SBMRMTCMD コマンドは、このグループ内のコマンドです。
 - 特定の方法でリモート・ファイルを処理するためのメンバー関連のコマンド。
 - DDM を同時に使用して、ソース・ファイルを処理できるソース・ファイル・コマンド。

上記のコマンドは、通常はローカル・ファイルを処理するのに使用しますが、そのパラメーターのうちのいずれかに DDM ファイル名を指定すれば、リモート・ファイルを処理する (プログラムに対して透過的に) こともできます。

- DDM とともに使用できないコマンド。

これらのコマンドの多くは、表に示したとおりの制限を受けるときにも、SBMRMTCMD コマンドを使ってターゲット・サーバー (IBM iまたはシステム/38 のみ) へ実行を依頼することはできますが、そのようにしてもあまり有効でない場合があります。

DDM でのオブジェクト指向コマンド:

ソース IBM i上の DDM ファイル・オブジェクトには、以下のようなオブジェクト指向 CL コマンドでアクセスすることができます。これらのコマンドを DDM ファイルとともに実行すれば、ローカル・サーバーの他のすべてのファイルに対する通常の働きと同じ働きをします。

これらのコマンドのあるものは、複数のオブジェクトを処理することができ、たとえば、総称ファイル名を指定すれば、そのオブジェクトのうちの 1 つ以上のものを、DDM ファイルとすることもできます。

表に注記したものを除き、これらのコマンドは、DDM ファイルと関連したリモート・ファイルに対しては効果がありません。すなわち、このコマンドのいずれかに DDM ファイルを指定しても、通信回線を経由してサーバー・システムに対する参照は行われません。

なお、どうしてもこのコマンドの 1 つを使ってリモート・ファイル (DDM ファイルではなく) を処理したい場合、サーバー・システムが IBM iまたはシステム/38 であれば、リモート・コマンド投入 (SBMRMTCMD) コマンドを使って、サーバー・システムで、そのコマンドを実行するよう依頼することができます。このような場合、実行依頼したコマンドの実行結果は、ソース・サーバーには送り返されません。ただし、クライアント・システム・ユーザーには、該当機能が正常に実行されたかどうかを示す何らかの表示 (通常はメッセージ) があります。

コマンド名	記述名
CHGDDMF	DDM ファイル変更
CHGLF ^{1, 2, 3, 4}	論理ファイル変更

コマンド名	記述名
CHGOBJOWN	オブジェクト所有者変更
CHGPF ^{1, 2, 3, 4}	物理ファイル変更
CHGSRCPF ^{1, 2, 3, 4}	ソース物理ファイル変更
CHKOBJ	オブジェクト検査
CRTDDMF	DDM ファイル作成
CRTDUPOBJ	複製オブジェクト作成
CRTLFL ^{1, 2, 3}	論理ファイル作成
CRTPF ^{1, 2, 3}	物理ファイル作成
CRTSRCPF ^{1, 2, 3}	ソース物理ファイル作成
CRTS36CBL ⁶	S/36 COBOL プログラム作成
CRTS36DSPF ⁷	S/36 表示装置ファイル作成
CRTS36MNU ⁷	S/36 メニュー作成
CRTS36MSGF ⁷	S/36 メッセージ・ファイル作成
CRTS36RPG ⁶	S/36 RPG II プログラム
CRTS36RPGR ⁷	コンソール表示ファイル作成
CRTS36RPT ⁶	S/36 RPG II 報告書自動作成
DLTF ^{1, 2, 3}	ファイル削除
DMPOBJ	オブジェクト・ダンプ
DMPYSOBY	システム・オブジェクト・ダンプ
DSPFD ^{1, 2, 3}	ファイル記述表示
DSPFFD ^{1, 2, 3}	ファイル・フィールド記述表示
DSPOBJAUT	オブジェクト権限表示
DSPOBJD	オブジェクト記述表示
GRTOBJAUT	オブジェクト権限認可
MOVOBJ	オブジェクト移動
RCLDDMCNV	分散データ管理 (DDM) 会話再使用
RNMOBJ ^{1, 2, 3}	オブジェクト名変更
RSTLIB	ライブラリー復元
RSTOBJ	オブジェクト復元
RVKOBJAUT	オブジェクト権限取り消し
SAVCHGOBJ	変更済みオブジェクト保管
SAVLIB	ライブラリー保管
SAVOBJ	オブジェクト保管
WRKJOB ⁵	ジョブ処理
WRKOBJLCK ⁵	オブジェクト・ロック処理

- ¹ クライアント・システムで実行された場合のこのコマンドは、SYSTEM (*LCL) が使用されていると、リモート・ファイルを参照しません。
- ² SYSTEM (*RMT) が指定してあったり、SYSTEM (*FILETYPE) を指定していてファイルが DDM ファイルであったりすると、リモート操作が行われます。
- ³ これらのコマンドには DDM ファイル名を指定することができるため、ターゲット IBM i またはターゲット・システム/38 でこれらの機能を実行する場合、**SBMRMTCMD** コマンドは不要です。
- ⁴ ターゲット・システムは、リリース 3.0 およびそれ以上の IBM i か、またはサポート・レベル 2.0 の DDM 体系でなければなりません。
- ⁵ クライアント・システムでこのコマンドを実行すると、リモート・ファイルではなく、DDM ファイルでのすべてのロックが表示されます。
- ⁶ DDM では、このシステム/36 環境コマンドをサポートしています。
- ⁷ DDM では、このシステム/36 環境コマンドをサポートしていません。

IBM i ターゲット・システムで必要なファイル管理コマンド:

以下の CL コマンドは、サーバー・システムも同じく IBM i またはシステム/38 プラットフォームである場合にのみ使用できます。

コマンド名	記述名
ADDLFM ¹	論理ファイル・メンバー追加
ADDPFM	物理ファイル・メンバー追加
CHGLFM	論理ファイル・メンバー変更
CHGPFM	物理ファイル・メンバー変更
CPYSRCF	ソース・ファイル・コピー
INZPFM	物理ファイル・メンバー初期設定
OPNQRYF	QUERY ファイル・オープン
RGZPFM	物理ファイル・メンバー再編成
RMVM	メンバー除去
RNMM	メンバー名変更

¹ サーバー・システムは IBM i オペレーティング・システムを実行していなければなりません。

これらのコマンドには DDM ファイル名を指定することができるため、IBM i またはシステム/38 サーバー・システムでこれらの機能を実行する場合、リモート・コマンド投入 (SBMRMTCMD) コマンドは不要です。

DDM でのメンバー関連コマンド:

メンバーに対して使用できるデータベース・ファイル操作であれば、DDM でも使用することができます。

DDM ファイル名を以下の CL コマンドのいずれかに指定すると、IBM i の DDM は、その DDM ファイルが参照するリモート・ファイル (およびメンバー) にアクセスします。ただし、表にもあるとおり、一部のコマンドは、リモート・ファイルが IBM i またはシステム/38 にある場合しか有効ではありません。

コマンド名	記述名
ADDPFM ¹	物理ファイル・メンバー追加
ADDLFM ⁶	論理ファイル・メンバー追加
ALCOBJ	オブジェクト割り振り
CHGLFM ¹	論理ファイル・メンバー変更
CHGPFM ¹	物理ファイル・メンバー変更
CLOF	ファイル・クローズ
CLRPFM	物理ファイル・メンバー消去
CPYF ²	ファイル・コピー
CPYFRMTAP	テープからのコピー
CPYSPLF	スプール・ファイル・コピー
CPYSRCF ¹	ソース・ファイル・コピー
CPYTOTAP	テープへのコピー
DCLF	ファイル宣言
DLCOBJ	オブジェクト割り振り解除

コマンド名	記述名
DSPFD ³	ファイル記述表示
DSPFFD ³	ファイル・フィールド記述表示
DSPPFM	物理ファイル・メンバー表示
INZPFM ¹	物理ファイル・メンバー初期設定
OPNDBF ⁴	データベース・ファイル・オープン
OPNQRYF ¹	QUERY ファイル・オープン
OVRDBF ⁵	データベース・ファイル一時変更
POSDBF	データベース・ファイル位置決め
RCVF	ファイル受け取り
RCVNETF	ネットワーク・ファイル受け取り
RGZPFM ¹	物理ファイル・メンバー再編成
RMVM ¹	メンバー除去
RNMM ¹	メンバー名変更
SNDNETF	ネットワーク・ファイル送信

¹ サーバー・システムは、IBM iまたはシステム/38 でなければなりません。

² このコマンドについての他の DDM 関連の考慮事項については、『DDM でのコピー・コマンド』を参照してください。

³ SYSTEM パラメーターに *RMT または *ALL を指定していると、これらのコマンドは、リモート・ファイル情報を表示します。

⁴ コミットメント制御については、『DDM のコミットメント制御サポート』を参照してください。

⁵ このコマンドでは、リモート・ファイルにはアクセスしません。

⁶ サーバー・システムは IBM iでなければなりません。

コマンドによっては、リモート・コマンド投入 (SBMRMTCMD) コマンドを使用して、サーバー・システムへ実行依頼することもできます。

ネットワーク・ファイル送信 (SNDNETF) コマンドや ネットワーク・ファイル受信 (RCVNETF) コマンドを実行するときには、DDM ファイルを使ってリモート・ファイルへアクセスするのではなく、できる限り、データのあるサーバーで実行する必要があります。

DDM をサポートしないコマンド:

以下の CL コマンドは、DDM ファイルに対してはサポートされません。ただし、リモート・コマンド投入 (SBMRMTCMD) コマンドを使ってサーバー・システムで実行するよう依頼すれば、ターゲット IBM iまたはシステム/38 で DDM を使用して有用な結果が得られる場合があります。

コマンド名	記述名
DSNFMT	様式设计
DSPCHT	図表表示
DSPDBR	データベース関連表示
DSPRCDLCK	レコード・ロック表示
MNGDEVTBL	装置テーブル管理
MNGPGMTBL	プログラム・テーブル管理

コマンド名	記述名
MNGUSRTBL	ユーザー・テーブル管理
RTVQRYSRC	QUERY ソース検索
SBMFNCJOB	金融機関ジョブ投入

ソース・ファイル・コマンド:

サーバー・システムが IBM i または システム/38 の場合、以下の CL コマンドは DDM ファイルをソース・ファイルとしてサポート (SRCFILE パラメーターで) できます。

サーバー・システムが IBM i でも システム/38 でもない場合、リモート・ファイルは IBM i のソース・ファイルにも システム/38 のソース・ファイルにもならないので、DDM ファイル名を SRCFILE パラメーターに指定しないようにしてください。

これらのコマンドは、ファイルの一時変更 (データベース・ファイル一時変更 [OVRDBF] コマンドを使用して) によって影響を受ける場合もあります。

注: これらのコマンドをクライアント・システムで実行しても、任意のサーバー・システムでファイルを作成することはできません。ただし、リモート・コマンド投入 (SBMRMTCMD) コマンドを使用すれば、ターゲット・サーバーでの実行を依頼することができます。

コマンド名	記述名
CRTBASPGM	BASIC プログラム作成
CRTBSCF ¹	BSC ファイル作成
CRTCBLPGM	COBOL プログラム作成
CRTCLPGM	CL プログラム作成
CRTCMD	コマンド作成
CRTCMNF ¹	通信ファイル作成
CRTCPGM	C プログラム作成
CRTDSPF	表示装置ファイル作成
CRTICFF	システム間通信機能ファイル作成 ¹
CRTMXDF ²	混合ファイル作成
CRTPLIPGM	PL/I プログラム作成
CRTPRTF	印刷装置ファイル作成
CRTPRIMG ²	印刷イメージ作成
CRTRPGLPGM	RPG プログラム作成
CRTRPPTGM	報告書簡易作成プログラム作成
CRTTBL	テーブル作成
FMTDTA	データ形式設定
STRBAS	BASIC 開始
STRBASPRC	BASIC プロシージャラー開始

¹ CRTICFF は、IBM i で有効です。CRTCMNF、CRTBSCF、および CRTMXDF コマンドは、システム/38、または IBM i のシステム/38 環境において有効です。

² SBMRMTCMD コマンドで使用する場合、ターゲット・システムはシステム/38 でなければなりません。

DDM 関連の CL コマンド一覧表:

このトピックでは、DDM で使用される制御言語プログラム (CL) コマンドのほとんどが含まれた一覧表を示します。

これらの CL コマンドは、DDM ジョブ環境を判別してリモート・ファイル処理を実行したり (CL コマンドのファイル関連パラメーターに DDM ファイル名を指定して)、リモート・サーバー上でその他のアクションを実行したり (リモート・コマンド投入 (**SBMRMTCMD**) コマンドでサーバー・システムへ CL コマンドを投入 (実行依頼) して) するのに使用します。

この一覧表では、次のようにコマンドを分けて示しています。

- ファイル関連 (ファイル・オブジェクトに対して使用される) コマンド。
- オブジェクト関連 (ファイル・オブジェクトのほかに、ファイル以外のオブジェクトに対して使用される) コマンド。
- ソース側またはターゲット側で実行できるコマンド。
- データベース・ファイル一時変更 (**OVRDBF**) コマンドを使用したファイル一時変更から影響を受けるコマンド。
- 特定の役立つ目的があって、クライアント・システムで実行する代わりに、ターゲット IBM i に実行依頼できる (**SBMRMTCMD** コマンドを使用) コマンド。

この一覧表には、DDM ユーザーに役立つように、注が付いています。

以下に、この一覧表に記載された情報の種類を説明します。

- 最初の欄には、DDM で使用できるすべての CL コマンドを列記しています。これらの CL コマンドは、(a) DDM ファイル内で識別されるリモート・ファイルを処理するものと、(b) DDM ファイルを使用して **SBMRMTCMD** コマンドで投入するものに分かれます。
- 2 番目の欄の F は、コマンドがファイル関連のものであることを意味し、O は、ファイル以外の IBM i オブジェクト関連であることを意味し、そして空白は、そのどちらでもないことを意味します。
- 3 番目の欄の S は、コマンドはソース側のオブジェクトを処理するものであることを意味し、そして T は、ターゲット側のオブジェクトを処理するものであることを意味します。たとえば、ソース・ファイルとして DDM ファイルを使用してファイルまたはプログラムを作成する作成コマンドの場合、T は、サーバー・システムにあるソース・ファイルがその作成に使用されることを示します。このコマンドは、クライアント・システム上で稼働し、クライアント・システム上でファイルまたはプログラムを作成しますが、そのとき、ターゲット・サーバーにあるソース・ファイルを使用します。

S も T も示されていない場合には、DDM ファイル名をそのコマンドに指定してはなりません。このコマンドは、DDM 機能としてソース・サーバー上で実行してはなりません。ただしこのコマンドは、**SBMRMTCMD** コマンドで投入して、ターゲット・サーバー上で実行すれば有益です (最終欄を参照)。

- 最後の 2 つの欄の X は、この欄の上に示されているコマンド (**OVRDBF** または **SBMRMTCMD**) とともに使用すれば、このコマンドは有効であり、有益であることを示します。空白は、そのコマンドは無効であることを示します。

一般的に、サーバー・システムが IBM i またはシステム/38 のときには、バッチ・ジョブまたはバッチ・プログラムのいずれかで使用できるものであれば任意の CL コマンドを、**SBMRMTCMD** コマンド上に指定できます。コマンドが、その ALLOW 属性に、コマンド表示 (**DSPCMD**) コマンドを使用して表示できる *BPGM および *EXEC の値を指定されていれば、そのコマンドを、**SBMRMTCMD** コマンドを使用して投入することができます。(**SBMRMTCMD** コマンドは、QCAEXEC サーバー・プログラムを使用して、投入されたコマンドをターゲット・サーバーで実行します。)

注:

1. **SBMRMTCMD** コマンドを使用すれば、IBM i、システム/38、またはリモート・コマンド機能の実行依頼をサポートする他のどのサーバー・システムにでも、コマンドを送ることができます。実行依頼するコマンドは、サーバー・システムの構文になっている必要があります。
2. この図に示すコマンドの多くは、**SBMRMTCMD** コマンドを使ってリモート・サーバーへ実行依頼することができますが、そのうちのいくつかは、**DDM** ファイル名を指定して、クライアント・システム上で簡単に実行することもできます。
3. IBM は、お客様に、すべてのプログラム・コードのサンプルを使用することができる非独占的な著作権使用権を許諾します。お客様は、このサンプル・コードから、お客様独自の特別のニーズに合わせた類似のプログラムを作成することができます。
4. すべてのサンプル・コードは、例として示す目的でのみ、IBM により提供されます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。
5. ここに含まれるすべてのプログラムは、現存するままの状態を提供され、いかなる保証も適用されません。商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任の保証の適用も一切ありません。

表 12. DDM 関連の CL コマンド

コマンド名	ファイル、オブジェクト、または両方に関連	ソース、ターゲット、または両方のオブジェクトに影響	OVRDBF コマンド	SBMRMTCMD コマンド ¹
ADDLFM				
ADDPFM	F	T ²		X
ALCOBJ	F	T ³		X
CHGDFUDEF	F O	S T		X
CHGDTA		T		X
		T		
CHGJOB				
CHGLF	F	S T		X
CHGLFM	F	T ³		X
CHGNETA				X
CHGOBJOWN	F O	S		X
				X
CHGPF				
CHGPFM	F	S T		X
CHGQRYDEF	F	T ³		X
CHGSRCPF		T		
CHKOBJ	F	S T		X
	F O	S		X
CLOF				
CLRPFM	F	T	X	X
COMMIT	F	T		X
CPYF	F	S T		X ¹¹
CPYFRMDKT	F	S T	X	X
CPYFRMQRYF	F	S T	X	X ⁴
CPYFRMTAP	F	S T	X	X
	F	S T	X	X ⁴

表 12. DDM 関連の CL コマンド (続き)

コマンド名	ファイル、オブジェクト、または両方に関連	ソース、ターゲット、または両方のオブジェクトに影響	OVRRDBF コマンド	SBMRMTCMD コマンド ¹
CPYSPLF		T	X	X
CPYSRCF	F	T	X	X
CPYTODKT	F	S T	X	X
CPYTOTAP	F	S T	X	X ⁴
CRTBASPGM	F	S T		X ⁴
		T		X
CRTCBLPGM		T		X
CRTCLPGM		T		X
CRTCMD		T		X
CRTDFUAPP		T		X
CRTDFUDEF		T		X
CRTDSPF		T		X
CRTDUPOBJ	F	T		X
CRTICFF	O	S	X	X
	F	T		X
CRTLFL		T		X
CRTPLF	F	S T		X
CRTPLIPGM	F	S T	X	X
CRTPRTF		T		X
CRTPRIMG	F	T		X
		T		X
CRTQRYAPP		T		X
CRTQRYDEF		T		X
CRTRPGPGM		T		X
CRTRPTPGM		T		X
CRTSRCPF	F	S T	X	X
CRTTBL		T		X
DCLF		T		X
DLCOBJ	F	T		X
DLTDFUAPP	F O	S T		X
DLTF		T		X
	F	S T		X
DLTQRYAPP		T		X
DMPOBJ		T		X
DMPSYSOBJ	F O	S		X ⁵
DSNDFUAPP	O	S		X ⁵
DSNQRYAPP		T		X
		T		X
DSPDTA		T		X
DSPFD		T		X
DSPFFD	F	S T		X ⁵
DSPNETA	F	S T		X ⁵
DSPOBJAUT		T		X
	F O	S		X ⁵

表 12. DDM 関連の CL コマンド (続き)

コマンド名	ファイル、オブジェクト、または両方に関連	ソース、ターゲット、または両方のオブジェクトに影響	OVRDBF コマンド	SBMRMTCMD コマンド ¹
DSPOBJD				
DSPPFM	F O	S		X ⁵
ENDCMTCTL	F	T		
FMTDTA	F	S T		X ¹¹
GRTOBJAUT		T		X
INZPFM	F O	S		X
	F	T ²		X
MOVOBJ				
OPNDBF ⁶	O	S		X
OPNQRYF	F	T	X	
OVRDBF	F	T	X	X ⁷
POSDBF	F	S		X
	F	T		
QRYDTA				
RCVF		T		X
RCVNETF	F	T		
RGZPFM	F			X
RMVM	F	T		X
	F	T		X
RNMM				
RNMOBJ	F	T		X
ROLLBACK	F O	S T ⁸		X
RSTLIB	F	S T		X ¹¹
RSTOBJ		S		X ⁹
RTVDFUSRC	F O	S		X ⁹
		T		X
RTVQRYSRC				
RVKOBJAUT		T		X
SAVCHGOBJ	F O	S		X
SAVLIB	O	S		X ⁹
SAVOBJ		S		X ⁹
	F O	S		X ⁹
SBMDBJOB				
SNDNETF		T		X
STRBAS	F	T		X
STRBASPRC		T		X
STRCMTCTL				
STRDBRDR	O	T		X
WRKJOB	F	S T		X ¹¹
WRKOBJLCK ¹⁰		T		X
	O			X ⁵
	F O	S		X ⁵

注:

¹ サーバー・システムが IBM i サーバーまたはシステム/38 でない限り、上記の表のどの コマンドも、**SBMRMTCMD** コマンドで使用することはできません。

² このメンバー関連のコマンドは、サーバー・システムが IBM i のときしか使用できません。

- 3 このメンバー関連のコマンドは、サーバー・システムが IBM i またはシステム/38 のときだけ使用できます。
- 4 これらのコマンドでは、テープをロードするのにサーバー・システムでの介入を必要とし、所定どおりの結果を生じないこともあります。
- 5 サーバー・システムへ実行依頼した場合、これらのコマンドは、サーバー・システムにおいてのみ出力を作成します。その出力は、クライアント・システムへは送られません。
- 6 **OPNDBF** コマンド：コミットメント制御に関する制約事項についての詳細は、『DDM のコミットメント制御サポート』を参照してください。
- 7 **OVRDBF** コマンド：このコマンドは、**SBMRMTCMD** コマンドでターゲット IBM i またはシステム/38 に実行依頼することが可能ですが、その使用はお勧めできません。
- 8 **RNMOBJ** コマンド：OBJTYPE*FILE の指定は必須になっています。
- 9 これらのコマンドをサーバー・システムへ実行依頼し、テープを使用して出力を生成するには、サーバー・システムの資源が必要になります。
- 10 **WRKOBJLCK** コマンド：このコマンドは、リモート・ファイルではなく、DDM ファイルにかけられたロックがあれば、それを表示します。
- 11 このコマンドは使用可能ですが、その使用はお勧めできません。

DDM データ記述仕様に関する考慮事項:

データ記述仕様 (DDS) は、フィールド様式とレコード様式を外部で記述するのに使用するものですが、DDM で使用すれば、リモート・ファイルのファイル・フォーマットとレコード様式を記述することもできます。

DDM の IBM i ターゲットに関する考慮事項:

このトピックでは、DDM のターゲットに関する考慮事項を説明します。

すべてのデータベース・ファイルと同様に、リモート IBM i でリモート・ファイルを作成するときに、DDS を使用して、そのファイルの属性を外部で記述することもできないこともあります。DDS を使用すると、ソース・サーバー・プログラムは、その DDS にある属性を使用してリモート・ファイルへアクセスします (DDM ファイルを使用介して)。DDS を使用しないのであれば、ファイルの属性をプログラムで記述する必要があります。

ターゲット IBM i にあるファイルにアクセスするクライアント・システム・プログラムをコンパイル (または再コンパイル) するときには、既存 DDM ファイルを使用してターゲット・サーバーとの通信が確立されます。コンパイル中にリモート・ファイルが実際にアクセスされて、そのファイル属性とレコード属性が取り出されます。ファイルを記述するのに DDS を使用してもしなくても、コンパイル時にレベル検査識別コードが作成され、コンパイル済みプログラム内に入れられます。この値は次に、LVLCHK (*RMTFILE) がその DDM ファイルに対して効力を持っていれば、プログラムを実行するときに使用されます。

リモート IBM i ファイルを記述するのに DDS を使用してもしなくても、クライアント・システム・プログラムは、プログラム内に独自のフィールド様式やレコード様式の定義を備えたり、または DDS を使用して作成された別のクライアント・システム・ファイルの定義を置き換えたりすることができます。LVLCHK (*NO) が、DDM ファイル内で使用されている場合か、プログラムの実行時に使用されるデータベース・ファイル一時変更 (**OVRDBF**) コマンド内で指定されている場合は、このいずれも行うことができます。LVLCHK (*NO) を使用する必要が生ずるのは、クライアント・システムで使用するレコード様式が、リモート IBM i ファイルのレコード様式と異なっているときだけです。

DDM の IBM i 以外のターゲットに関する考慮事項:

DDS を非 IBM i ファイルで使用できるのは、使用している DDM ファイルと同じレコード様式名を持ったローカル IBM i ファイルを使ってローカル IBM i プログラムをコンパイルするときだけです。

プログラムをコンパイルし終われば、リモート・ファイルにアクセスするための DDM ファイルを使用してローカル・ファイルを一時変更することができます。LVLCHK (*NO) は、DDM ファイル内かまたは OVRDBF コマンド内で指定する必要があります。

リモート・ファイルを記述するための DDS がローカル・サーバーにない場合には、プログラムでフィールドを記述する必要があります。ファイル・フィールド記述表示 (DSPFFD) コマンドを使用すれば、リモート・ファイルのフィールド属性を表示することができます。LVLCHK (*NO) は、DDM ファイル内かまたは OVRDBF コマンド内に指定する必要があります。

LVLCHK (*RMTFILE) が指定されているか、デフォルトになっている場合、プログラムのコンパイル (または再コンパイル) には、リモート・ファイルにアクセスする DDM ファイルを使用する必要があります。この後、IBM iは、リモート・ファイル用のレコード様式とフィールドを作成します。作成されるフィールド名は、キー付きフィールドの場合はタイプ *Knnnnn* に、キーなしフィールドの場合は *Fnnnnn* になります。

DDM 関連の DDS キーワードおよび情報:

このトピックでは、特に DDM に関連した DDS キーワードについて述べます。

- ローカル・ファイルを作成するときの考慮事項は以下のとおりです。
 - DDS キーワードのうち、REFACCPATH および FORMAT には、DDM ファイル名を指定することはできません。
 - DDS キーワードの REF および REFFLD には、リモート・ファイルを参照するための DDM ファイル名を指定することができます。ただし、そのリモート・ファイルは、IBM iまたはシステム/38 にあるファイルでなければなりません。いずれかのキーワードで、データベース・ファイル名として DDM ファイル名を指定すると、そのファイル名はクライアント・システムにある DDM ファイルを参照し、その中で参照されるフィールド名とレコード様式名は、サーバー・システムにあるリモート・ファイル内で使用されるフィールド様式とレコード様式を参照します。
- リモート・サーバーが IBM iでない場合の論理ファイルの作成に関する考慮事項は以下のとおりです。
 - その論理ファイルのレコード様式に、少なくとも 1 つのキー・フィールドを指定する必要があります。
 - PFILE キーワードには、1 つのファイルしか指定することはできません。
 - SELECT 機能と OMIT 機能は、サポートされません。
 - 論理結合ファイルは、サポートされません。
 - リモート物理ファイルのフィールド名には、キーなしフィールドの場合は、F00001、F00002、F00003 など (*Fnnnnn*) の命名規則が、キー付きフィールドの場合は、K00001、K00002、K00003 など (*Knnnnn*) の命名規則があります。

この命名規則の例外として、サーバー・システムがシステム/38 であって、物理ファイルがローカルで作成されている場合があります。この場合、フィールド名は、物理ファイルを作成したときに指定したフィールド名と同じになります。

- 論理ファイル用に定義するフィールドはすべて、物理ファイル内で定義したとおりの順序で指定する必要があります。デフォルトを使用すれば、そのとおりの順序になります。

- SST キーワードを使用すれば、物理ファイルの一部にアクセスすることができます。物理フィールド全体を定義するには、複数のサブストリング・フィールドが必要です。さらに、このフィールドの一部は、物理ファイルのサブストリング・フィールド内で定義されたとおりの順序になっていなければなりません。
- CONCAT キーワードを使用すれば、各種物理ファイル・フィールドをグループにまとめて、1 つの論理フィールドにすることができます。各フィールドの連結順序は、物理ファイルに定義されたとおりの順序になっていなければなりません。
- 物理ファイルの各フィールドは、物理ファイルに定義されたとおりの順序で指定する必要があります。
- IBM i 以外のサーバー・システムでファイルを作成する場合の VARLEN DDS キーワードの用法に関する考慮事項は以下のとおりです。
 - ターゲット・サーバーは、可変長レコード・ファイルをサポートしていなければなりません。
 - ファイル・フォーマットには 1 つの可変長フィールドしか使用することはできず、しかもそれは、最後のフィールドでなければなりません。
 - VARLEN キーワードを指定するフィールドは、キー・フィールドであってはなりません。
- リモート・ファイルを作成する場合の PFILE および JFILE に関する考慮事項は以下のとおりです。
 - DDM ファイル内の物理ファイル用に JFILE キーワードや PFILE キーワード上で指定するレコード様式名は、リモート物理ファイルを表す DDM ファイルと同じ名前ではなければなりません。
 - 論理ファイルを作成するときには、PFILE または JFILE に指定するファイルは DDM ファイルでなければならず、JFILE キーワードや PFILE キーワード上の DDM ファイル内の各物理ファイルのロケーションは、その論理ファイル用の DDM ファイルのロケーションと同じでなければなりません。つまり、物理ファイルと論理ファイルは、同じリモート・サーバー上になければならないということです。

リモート・サーバーがリリース 1.0 または 1.2 の IBM iであれば、FCFO キーワードを使ってファイルを作成しようとしても失敗します。

- サーバーが IBM iでないときには、論理 ファイルに関しては、下記のいずれのキーワードも無視されるか、または、サポートされません。

ABSVAL	EDTCDE	LIFO
ACCPATH	EDTWRD	NOALTSEQ
ALIAS	FCFO	RANGE
ALL	FLTPCN	REFSHIFT
ALTSEQ	FORMAT	RENAME
CHECK	JDFTVAL	SIGNED
CMP	JDUPSEQ	TEXT
COLHDG	JFILE	TRNTBL
COMP	JFLD	VALUES
DIGIT	JOIN	ZONE
DYNSLT	JREF	

- サーバーが IBM iでないときには、物理 ファイルに関しては、下記のいずれのキーワードも無視されるか、または、サポートされません。

ABSVAL	EDTCDE	RANGE
ALTSEQ	EDTWRD	REFSHIFT
CHECK	FCFO	SIGNED
CMP	FLTPCN	TEXT
COLHDG	FORMAT	VALUES
COMP	LIFO	ZONE
DIGIT	NOALTSEQ	

DDM ユーザー・プロファイル権限:

IBM i ユーザーは、適切なコマンド権限を持っていない限り、DDM を使用して、リモート IBM i 上で CL コマンドと同等の機能を実行することはできません。

リモート IBM i で同等の操作を開始するには、ターゲット・ジョブに関連したユーザー・プロファイルは、以下の CL コマンドに対する *OBJOPR 権限を持っていない限りなりません。

コマンド名	記述名
ADDLFM	論理ファイル・メンバー追加
ADDPFM	物理ファイル・メンバー追加
ALCOBJ	オブジェクト割り振り
CHGLF	論理ファイル変更
CHGLFM	論理ファイル・メンバー変更
CHGPF	物理ファイル変更
CHGPFM	物理ファイル・メンバー変更
CRTLFL	論理ファイル作成
CRTPLF	物理ファイル作成
DLTF	ファイル削除
INZPFM	物理ファイル・メンバー初期設定
RGZPFM	物理ファイル・メンバー再編成
RMVM	メンバー除去
RNMM	メンバー名変更
RNMOBJ	オブジェクト名変更

DDM コマンドとパラメーター

このトピックでは、DDM のコマンドとパラメーターを分類します。

DDM のサブセットについての詳細は、「*DDM Architecture: Implementation Planner's Guide*」または「*DDM Architecture: Reference*」を参照してください。

注: このトピックの表では、省略語 *KB* を使用しています。これは、1024 バイトに相当する記憶容量を表します。

IBM i DDM でサポートされる DDM 体系のサブセット:

IBM i は、以下の DDM 体系のサブセットをサポートします。

サポートされる DDM ファイル・モデル:

IBM i DDM は、以下の DDM ファイル・モデルをサポートします。

- 代替索引ファイル (ALTINDF)
- 直接ファイル (DIRFIL)
- ディレクトリー・ファイル (DIRECTORY)
- キー付きファイル (KEYFIL)
- 順次ファイル (SEQFIL)
- ストリーム・ファイル (STRFIL)

IBM i では、上記の 5 つのファイル・モデルを使用した IBM i 物理ファイルと論理ファイルへのアクセスがサポートされます。次の表は、どのように DDM ファイル・モデルと IBM i データ・ファイルが対応するかを示しています。

表 13. IBM i データ・ファイル

DDM ファイル・モデル	対応する IBM i データ・ファイル
代替索引ファイル (ALTINDF)	単一形式の論理ファイル
直接ファイル (DIRFIL)	キーなし物理ファイル
ディレクトリー・ファイル (DIRECTORY)	フォルダー管理サービス (FMS) フォルダーまたはデータ管理ライブラリー
キー付きファイル (KEYFIL)	キー付き物理ファイル
順次ファイル (SEQFIL)	キーなし物理ファイル
ストリーム・ファイル (STRFIL)	フォルダー管理サービス (FMS) 文書

代替索引ファイル (ALTINDF):

IBM i DDM は、DDM 代替索引ファイル・モデルを使用した論理ファイルへのアクセスをサポートします。

論理ファイルを使用すれば、物理ファイルをもとに定義された代替索引を使用して、物理ファイル内に保管されているデータ・レコードへアクセスすることができます。IBM i DDM を使用した場合は、単一形式論理ファイルへのみアクセスすることができます。選択/除外論理を備えた論理ファイルへアクセスすることはできませんが、挿入されたレコードが選択/除外論理で除外されていると、そのレコードを検索できないことがあります。

サポートされるレコード・クラス

IBM i 代替索引ファイルは、固定長レコード (RECFIX) にも可変長レコード (RECVAR) にも保管することができます。

IBM i 以外のソース・サーバーが、可変長レコード・アクセスを使用して IBM i ターゲット・システム上でファイルをオープンした後は、IBM i ターゲット・システムはその後行われるすべての入出力操作において、常に可変長レコードを送受信することになります。

注: IBM i DDM は、代替索引以外のすべてのファイル・モデルに関して、DDM ファイル転送コマンドのレコード・ファイル・ロード (LODRECFIL) コマンドとレコード・ファイル・アンロード (ULDRECFIL) コマンドをサポートします。

直接ファイル (DIRFIL):

IBM i DDM は、DDM 直接ファイル・モデルを使用したキーなし物理ファイルへのアクセスをサポートします。

このサポートには、次のような特徴があります。

- 削除可能性: IBM i 直接ファイルは、削除可能なものと削除不可能なものがあります。削除不可能なファイルには、活動状態にあるデフォルトのレコードがなければなりません。
- サポートされるレコード・クラス: IBM i 直接ファイルは、固定長レコード (RECFIX) にも、可変長レコード (RECVAR) にも保管することができます。IBM i 以外のソース・サーバーが、可変長レコード・アクセスを使用して IBM i ターゲット・システム上でファイルをオープンした後は、IBM i ターゲット・システムはその後行われるすべての入出力操作において、常に可変長レコードを送受信することになります。

注: IBM i は、直接ファイルの概念をサポートしません。IBM i DDM が直接ファイルを作成するときには、削除済みレコードまたは活動デフォルト・レコードを使用してキーなし物理ファイルを作成し、要求どおりの最大サイズに初期設定します。このファイルに対して拡張を行うことはできません。

ディレクトリー・ファイル (DIRECTORY):

IBM i DDM は、DDM ディレクトリー・ファイル・モデルを使用したフォルダー管理サービス・フォルダーへのアクセスや、データ管理ライブラリーへのアクセスをサポートします。フォルダーを、作成、オープン、名前変更、クローズ、または削除することができます。ライブラリーは、作成、名前変更、または削除することができます。

キー付きファイル (KEYFIL):

IBM i DDM は、DDM キー付きファイル・モデルを使用したキー付き物理ファイルへのアクセスをサポートします。

IBM i キー付きファイルは、固定長レコード (RECFIX) にも、可変長レコード (RECVAR) にも保管することができます。IBM i 以外のソース・サーバーが、可変長レコード・アクセスを使用して IBM i ターゲット・システム上でファイルをオープンした後は、IBM i ターゲット・システムはその後行われるすべての入出力操作において、常に可変長レコードを送受信することになります。

順次ファイル (SEQFIL):

IBM i は、DDM 順次ファイル・モデルを使用したキーなし物理ファイルへのアクセスをサポートします。

このサポートには、次のような特徴があります。

- IBM i の順次ファイルは、削除可能であることも、削除不可能であることもあります。
- IBM i の順次ファイルは、固定長レコード (RECFIX) にも、可変長レコード (RECVAR) にも保管することができます。IBM i 以外のソース・サーバーが、可変長レコード・アクセスを使用して IBM i ターゲット・システム上でファイルをオープンした後は、IBM i ターゲット・システムはその後行われるすべての入出力操作において、常に可変長レコードを送受信することになります。

ストリーム・ファイル (STRFIL):

IBM i DDM は、DDM ストリーム・ファイル・モデルを使用したフォルダー管理サービス文書へのアクセスをサポートします。

サポートされる DDM アクセス方法:

IBM i DDM は、次の DDM アクセス方法をサポートします。括弧で囲んであるのは、DDM でのアクセス方法の省略形です。

- 混合アクセス方法 (CMBACCAM)
- 混合キー順アクセス方法 (CMBKEYAM)
- 混合レコード番号アクセス方法 (CMBRNBAM)
- ディレクトリー・アクセス方法 (DRCAM)
- キーによるランダム・アクセス方法 (RNDKEYAM)
- レコード番号によるランダム・アクセス方法 (RNDRNBAM)
- キーによる相対アクセス方法 (RELKEYAM)
- レコード番号による相対アクセス方法 (RELRNBAM)
- ストリーム・アクセス方法 (STRAM)

IBM i DDM がサポートする各 DDM ファイル・モデル別のアクセス方法の一覧表は、次の図を参照してください。これらのアクセス方法に関する説明は、「*DDM Architecture: Implementation Planner's Guide*」(GC21-9528) を参照してください。

表 14. 各 DDM ファイル・モデルでサポートされるアクセス方法

項目	アクセス方法	DDM ファイル・モデル					
		順次ファイル	直接ファイル	キー付きファイル	代替索引ファイル	ストリーム・ファイル	ディレクトリー・ファイル
CMBACCAM	混合アクセス	N	T	T	N		
CMBKEYAM	混合キー付き			T	T		
CMBRNBAM	混合レコード番号	T	T	T	N		
DRCAM	ディレクトリー						T
RELKEYAM	キーによる相対			T	T		
RELRNBAM	レコード番号による相対	T	T	T	N		
RNDKEYAM	キーによるランダム			T	T		
RNDRNBAM	レコード番号によるランダム	T	T	T	N		
STRAM	ストリーム					T	

注:

N = 非サポート

T = サポートされるターゲット DDM

ブランク = 適用外

DDM コマンドとオブジェクト:

このトピックでは、各 DDM 体系コマンド別に、IBM iがサポートする DDM コマンド・パラメーターについて説明します。

これらのパラメーターの詳細については、「*DDM Architecture: Reference*」(SC21-9526)を参照してください。

コマンドの説明には、以下のものが含まれます。

- 各コマンドの用法における制約事項
- クライアント・システムがサーバー・システムへ送ることのできるオブジェクト
- サーバー・システムがクライアント・システムへ送り返すことのできるオブジェクト
- IBM iがサポートするコマンドの DDM パラメーター、および各パラメーターに対する IBM iの応答

以下のコマンドがサポートされます。レベル 1.0、レベル 2.0 およびレベル 3.0 は、それぞれのコマンドがどのレベルの DDM 体系をサポートするかを示します。

CHGCD (現行ディレクトリー変更) レベル 2.0:

このコマンドは、現行パスを変更するためのコマンドです。パスは、一連のフォルダーです。現行パスは、斜線で始まっていないファイル名やディレクトリー名の前に付けます。

このコマンドは、ソース IBM iから送られることはありません。

パラメーター名	ソース	ターゲット
AGNNAM	該当せず	無視
DRCNAM ¹	該当せず	IBM i名

¹ 名前の形式はサーバーが定義します。現行ディレクトリー変更コマンドの場合には、このアーキテクチャーでは、ディレクトリー名の長さゼロは、ルート・ディレクトリーを示すと定められています。他のコマンドの場合、ディレクトリー名長がゼロの場合は、現行ディレクトリーであることを示します。このディレクトリーは、コマンドが出された時点でルートのディレクトリーであることも、ないこともあります。

CHGEOF (ファイルの終わり変更) レベル 2.0 およびレベル 3.0:

このコマンドは、文書のファイル終了マークを変更するためのものです。ファイルの終わりは、切り捨てることも拡張することもできます。

ソース IBM iから、このコマンドが送られることはありません。

パラメーター名	ソース	ターゲット
DCLNAM	該当せず	プログラムで定義
EOFNBR	該当せず	サポート
EOFOFF	該当せず	サポート

CHGFAT (ファイル属性変更) レベル 2.0:

このコマンドは、ファイル、文書、またはフォルダーの属性を変更するためのものです。

パラメーター名	ストリーム・ファイル	ディレクトリー	順次、直接、およびキ ー付きファイル	代替索引ファイル
DTAFMT	T			
FILCHGDT	T	T	N	N
FILCHGFL	T	N	N	
FILINISZ	N		S、T	

パラメーター名	ストリーム・ファイル	ディレクトリー	順次、直接、およびキー付きファイル	代替索引ファイル
FILEXNSZ	N		S、T	
FILEXPDT			S、T	
FILHDD	T	T	N	N
FILMAXEX	N		S、T	
FILPRT	T	N		
FILSYS	T	T	N	N
DELCP			N	N
GETCP	T		N	
INSCP			N	
MODCP	T		N	
TITLE	T	T	S、T	S、T

N = 非サポート
T = ターゲット DDM サポート
S = ソース DDM サポート
ブランク = 適用外

CLOSE (ファイル・クローズ) レベル 1.0 およびレベル 2.0:

このコマンドは、クライアント・システムと、サーバー・システム上にあるアクセスされたデータ・セットとの間の論理接続を終了します。ターゲット DDM は、このコマンドを実行し始めると、返された応答メッセージがどのようなものであっても、データ・セットをクローズする必要があります。

パラメーター名	ソース	ターゲット
DCLNAM	プログラムで定義	プログラムで定義
SHDPRC	送信不可	サポート

注: 名前は導入時に定義します。

CLRFIL (ファイル消去) レベル 1.0 およびレベル 2.0:

このコマンドは、既存ファイルを消去して、初めて作成されたものとして再初期設定します。

パラメーター名	ソース	ターゲット
FILNAM	受動側で定義	IBM i
OVRDTA	送信不可	FALSE (偽) のみ

名前の形式はサーバーで定義します。

CLSDRC (ディレクトリー・クローズ) レベル 2.0:

このコマンドは、フォルダーをクローズします。このコマンドは、ソース IBM i から送られることはありません。

パラメーター名	ソース	ターゲット
DCLNAM	該当せず	プログラムで定義

パラメーター名	ソース	ターゲット
名前は導入時に定義します。		

CPYFIL (ファイルのコピー) レベル 2.0:

このコマンドは、ある文書を別の文書へコピーするためのコマンドです。新規の文書が存在していない場合には、作成されます。このコマンドは、ソース IBM iから送られることはありません。

パラメーター名	ソース	ターゲット
ACCORD	該当せず	無視
BYPDMG	該当せず	無視
BYPINA	該当せず	無視
CPYNEW ¹	該当せず	サポート
CPYOLD ²	該当せず	サポート
DCLNAM ³	該当せず	プログラムで定義
FILNAM ⁴	該当せず	IBM i名
NEWFILNM ⁴	該当せず	IBM i名

1. CPYNDT は、サポートされる唯一のパラメーター値です。他のものはすべて拒否され、VALNSPRM が出ます。
2. CPYERR は、サポートされる唯一のパラメーター値です。他のものはすべて拒否され、VALNSPRM が出ます。
3. 名前は導入時に定義します。
4. 名前の形式はサーバーで定義します。

CRTAIF (代替索引ファイル作成) レベル 1.0 およびレベル 2.0:

このコマンドは、サーバー・システム上で代替索引ファイルを作成します。

パラメーター名	ソース	ターゲット
BASFILNM ¹	プログラムで定義	IBM i名
DUPFILOP	送信不可	サポート
FILCLS ²	送信不可	無視
FILHDD	送信不可	無視
FILNAM ³	プログラムで定義	IBM i名
FILSYS	送信不可	無視
KEYDEF ⁴	送信可	サポート
KEYDUPCP	送信可	サポート
RTNCLS ⁵	送信不可	サポート
TITLE	送信可	サポート

1. 名前の形式はサーバーで定義します。
2. ALTINDF コマンドには、CRTAIF のみが有効です。
3. 名前の形式はサーバーで定義します。
4. IBM i の最大キー長は 2000 です。
5. ライブラリー QTEMP が一時データに使用されます。

CRTDIRF (直接ファイル作成) レベル 1.0 およびレベル 2.0:

このコマンドは、サーバー・システム上で直接ファイルを作成するためのものです。

パラメーター名	ソース	ターゲット
ALCINIEX	送信可	無視
DCLNAM ¹	送信不可	サポート
DELCP ²	送信可	サポート
DFTREC	送信可	サポート
DFTRECOP	送信可	サポート
DUPFILOP	送信不可	サポート
FILCLS ³	送信不可	無視
FILEXNSZ ⁴	送信可	サポート
FILEXPDT ⁵	送信可	サポート
FILHDD	送信不可	無視
FILINISZ ⁴	送信可	サポート
FILMAXEX ⁶	送信可	サポート
FILNAM ⁷	プログラムで定義	IBM i名
FILSYS	送信不可	無視
GETCP	送信可	サポート
INSCP ⁸	送信可	サポート
MODCP	送信可	サポート
RECLN ⁹	送信可	サポート
RECLNCL	送信可	サポート
RTNCLS ¹⁰	送信不可	送信不可
サポート		
TITLE	送信可	サポート

¹ 名前は導入時に定義します。

² DFTRECOP (DFTSRCIN) を指定しない限り、値は TRUE でなければなりません。

³ **CRTDIRF** コマンドには DIRFIL のみが有効です。

⁴ IBM i のデフォルト・レコード数は、1,000 個です。

⁵ IBM i のデフォルトは *NONE です。

⁶ IBM i のデフォルトは 3 です。

⁷ 名前の形式はサーバーで定義します。

⁸ TRUE のみ有効です。

⁹ IBM i 最大レコード長 = 2**15-2 (2 の 15 乗マイナス 2)

¹⁰ ライブラリー QTEMP が一時データに使用されます。

CRTDRC (ディレクトリー作成) レベル 2.0:

このコマンドは、受け取った名前に基づいて、サーバー・システム上でフォルダーまたはライブラリーを作成します。このコマンドは、ソース IBM i から送られることはありません。

パラメーター名	ソース	ターゲット
DCLNAM ¹	該当せず	プログラムで定義
DRCNAM ²	該当せず	IBM i名
FILCLS ³	該当せず	無視
FILPRT ⁴	該当せず	サポート
RTNCLS	該当せず	PRMFIL のみ

パラメーター名	ソース	ターゲット
TITLE	該当せず	サポート
¹	名前は導入時に定義します。	
²	名前の形式はサーバーで定義します。	
³	CRTDRC コマンドには DIRECTORY のみが有効です。	
⁴	ライブラリーの場合は FALSE のみ。	

CRKEYF (キー付きファイル作成) レベル 1.0 およびレベル 2.0:

このコマンドは、サーバー・システム上でキー付きファイルを作成します。

パラメーター名	ソース	ターゲット
ALCINIEX	送信可	無視
DCLNAM ¹	使用不可	サポート
DELCP	送信可	サポート
DFTREC	送信不可	サポート
DFTRECOP	送信不可	サポート
DUPFILOP	送信不可	サポート
FILCLS ²	送信不可	無視
FILEXNSZ ³	送信可	サポート
FILEXPDT ⁴	送信可	サポート
FILHDD	送信不可	無視
FILINISZ ³	送信可	サポート
FILMAXEX ⁵	送信可	サポート
FILNAM ⁶	プログラムで定義	IBM i名
FILSYS	送信不可	無視
GETCP	送信可	サポート
INSCP	送信可	サポート
KEYDEF ⁷	送信可	サポート
KEYDUPCP	送信可	サポート
MODCP	送信可	サポート
RECLEN ⁸	送信可	サポート
RECLENCL	送信可	サポート
RTNCLS ⁹	送信不可	サポート
TITLE	送信可	サポート
¹	名前は導入時に定義します。	
²	CRKEYF コマンドには KEYFIL のみが有効です。	
³	IBM i のデフォルト・レコード数は、1,000 個です。	
⁴	IBM i のデフォルトは *NONE です。	
⁵	IBM i のデフォルトは 3 です。	
⁶	名前の形式はサーバーで定義します。	
⁷	IBM i の最大キー長は 2000 です。	
⁸	IBM i 最大レコード長 = 2**15-2 (2 の 15 乗マイナス 2)	
⁹	ライブラリー QTEMP が一時データに使用されます。	

注: CRTKEYF データが IBM i ターゲット・サーバーから送られると、新キー付きファイルは、作成時に削除されたレコードを再使用します。重複キーを使用でき (KEYDUPCP=TRUE 送信可) ても、重複キーの順序は保証されません。

CRTSEQF (順次ファイル作成) レベル 1.0 およびレベル 2.0:

このコマンドは、サーバー・システム上で順次ファイルを作成するためのコマンドです。

パラメーター名	ソース	ターゲット
ALCINIEX	送信可	無視
DCLNAM ¹	送信不可	サポート
DELCP	送信可	サポート
DFTREC	送信不可	サポート
DFTRECOP	送信不可	サポート
DUPFILOP	送信不可	サポート
FILCLS ²	送信不可	無視
FILEXNSZ ³	送信可	サポート
FILEXPDT ⁴	送信可	サポート
FILHDD	送信不可	無視
FILINISZ ³	送信可	サポート
FILMAXEX ⁵	送信可	サポート
FILNAM ⁶	プログラムで定義	IBM i名
FILSYS	送信不可	無視
GETCP	送信可	サポート
INSCP	送信可	サポート
MODCP	送信可	サポート
RECLEN ⁷	送信可	サポート
RECLENCL	送信可	サポート
RTNCLS ⁸	送信不可	サポート
TITLE	送信可	サポート

¹ 名前は導入時に定義します。

² **CRTSEQF** コマンドには SEQFIL のみが有効です。

³ IBM i のデフォルト・レコード数は、1,000 個です。

⁴ IBM i のデフォルトは *NONE です。

⁵ IBM i のデフォルトは 3 です。

⁶ 名前の形式はサーバーで定義します。

⁷ IBM i 最大レコード長 = 2**15-2 (2 の 15 乗マイナス 2)

⁸ ライブラリー QTEMP が一時データに使用されます。

CRTSTRF (ストリーム・ファイル作成) レベル 2.0:

このコマンドは、サーバー・システム上でストリーム・ファイルを作成するためのコマンドです。このコマンドは、ソース IBM i から送られることはありません。

パラメーター名	ソース	ターゲット
ALCINIEX	該当せず	無視
DCLNAM ¹	該当せず	プログラムで定義
DTAFMT	該当せず	サポート
DUPFILOP	該当せず	サポート
FILCLS ²	該当せず	無視

パラメーター名	ソース	ターゲット
FILEXNSZ	該当せず	無視
FILEXPDT	該当せず	無視
FILHDD	該当せず	サポート
FILINISZ	該当せず	無視
FILMAXEX	該当せず	無視
FILNAM ³	該当せず	IBM i名
FILPRT	該当せず	サポート
FILSYS	該当せず	サポート
GETCP	該当せず	サポート
MODCP	該当せず	サポート
RTNCLS	該当せず	サポート
TITLE	該当せず	サポート
¹	名前は導入時に定義します。	
²	CRSTSTRF コマンドには STRFIL のみが有効です。	
³	名前の形式はサーバーで定義します。	

DCLFIL (ファイル宣言) レベル 1.0 およびレベル 2.0:

このコマンドは、宣言名 (DCLNAM) を、ターゲット・エージェント内のオブジェクト指向パラメーターの集まり (グループ) に関連付けます。このパラメーター・グループは、後で使用する場合に備えて受信エージェントが保管します。受信時点でこのコマンドが、受信側エージェントが現在オープンしているオブジェクトに影響を与えることはありません。DCLFIL パラメーター・グループに主にアクセスするのは、DCLNAM パラメーターです。

パラメーター名	ソース	ターゲット
AGNNAM ¹	送信不可	無視
DCLNAM ²	プログラムで定義	プログラムで定義
DRCNAM ³	送信不可	IBM i名
FILEXNSZ ⁴	送信不可	無視
FILMAXEX ⁴	送信不可	無視
FILNAM ³	プログラムで定義	IBM i名
¹	IBM iでは唯一のエージェントです。	
²	名前は導入時に定義します。	
³	名前の形式はサーバーで定義します。	
⁴	作成値が使用されます。	

DELDCL (宣言名削除) レベル 1.0:

このコマンドは、宣言されたエージェント名を削除するためのものです。

パラメーター名	ソース	ターゲット
AGNNAM	送信不可	無視
DCLNAM ¹	プログラムで定義	プログラムで定義
¹	名前は導入時に定義します。	

DELDRC (ディレクトリー削除) レベル 2.0:

このコマンドは、フォルダーまたはライブラリーを削除するためのコマンドです。このコマンドは、ソース IBM i から送られることはありません。

パラメーター名	ソース	ターゲット
DELDRCOP ¹	該当せず	DRCEMP または DRCANY
DRCNAM ²	該当せず	IBM i名
OVRDTA	該当せず	FALSE (偽) のみ
¹	DRCALL 非サポート	
²	名前の形式はサーバーで定義します。総称名はサポートしません。	

DELFIL (ファイル削除) レベル 1.0 およびレベル 2.0:

このコマンドは、ファイルまたは文書を削除するためのコマンドです。

パラメーター名	ソース	ターゲット
FILNAM ¹	受動側定義の総称が使用可能	IBM i名
OVRDTA ²	送信不可	FALSE (偽) のみ
SHDONL ³	送信不可	サポート
¹	名前の形式はサーバーで定義します。総称名は、文書の場合のみ使用できます。	
²	IBM iは、上書きをサポートしません。	
³	ファイルの場合のみ FALSE。	

DELREC (レコード削除) レベル 1.0:

このコマンドは、現在更新Intentの置かれているレコードを削除します。削除の際、現行カーソル位置に影響を与えることはありません。

パラメーター名	ソース	ターゲット
DCLNAM ¹	プログラムで定義	プログラムで定義
¹	名前は導入時に定義します。	

EXCSAT (サーバー属性交換) レベル 1.0 およびレベル 2.0:

このコマンドは、各サーバー間で情報を交換するのに使用します。その情報には、サーバーのクラス名、サーバーがサポートする管理プログラムの各クラスのアーキテクチャー・レベル、サーバーのプロダクト・リリース・レベル、サーバーの外部名、およびサーバー名などがあります。

パラメーター名	ソース	ターゲット
EXTNAM	送信可	サポート
MGRLVLS	送信可	サポート
SPVNAM	送信不可	無視
SRVCLSNM	送信可	サポート
SRVNAM	送信可	サポート
SRVRLSLV	送信可	サポート

次のような応答オブジェクトが返されます。

EXCSATRD

サーバー属性応答データ

FILAL および FILATTRL (ファイル属性リスト) レベル 1.0、レベル 2.0、およびレベル 3.0:

これは、DDM が LSTFAT、OPEN、または GETDRCEN 上に要求することのできるファイル属性のリストです。パラメーターによっては、特定のファイル・タイプにしか有効ではありません。

表 15. ファイル属性リスト

パラメーター名	ソース	ターゲット
ACCMTHLS	要求可	サポート
BASFILNM ¹	要求可	IBM i名
DELCP	要求可	サポート
DFTREC	要求可	サポート
DTAFMT	要求不可	サポート
EOFNBR	要求可	サポート
EOFOFF	要求不可	サポート
FILBYTCN	要求不可	サポート
FILCHGDT	要求可	サポート
FILCHGFL	要求不可	サポート
FILCLS	要求可	サポート
FILCRTDT	要求可	サポート
FILEXNCN	要求可	サポート
FILEXNSZ	要求可	サポート
FILEXPDT	要求可	サポート
FILHDD	要求不可	サポート
FILINISZ	要求可	サポート
FILMAXEX	要求可	サポート
FILNAM	要求可	サポート
FILPRT	要求不可	サポート
FILSIZ	要求可	サポート
FILSYS	要求不可	サポート
GETCP	要求可	サポート
INSCP	要求可	サポート
KEYDEF	要求可	サポート
KEYDUPCP	要求可	サポート
LSTACCDT	要求不可	非サポート
LSTARCDT	要求可	サポート
MAXARNB	要求可	非サポート
MODCP	要求可	サポート
RECLN	要求可	サポート
RECLNCL	要求可	サポート
RTNCLS ²	要求不可	PRMFIL
SHDEXS	要求不可	サポート
STRSIZ	要求不可	サポート
TITLE ³	要求可	サポート

¹ 名前の形式はサーバーで定義します。FILCLS が ALTINDF であれば、修飾名です。

² ライブラリーが QTEMP でない場合のみ。

³ テキストの最大長は、データ・ファイルの場合は 50 文字、文書またはフォルダーの場合は 44 文字です。

FRCBFF (バッファ強制) レベル 2.0:

このコマンドは、参照済みオブジェクトのデータを不揮発性記憶装置に強制保管するためのコマンドです。

パラメーター名	ソース	ターゲット
DCLNAM ¹	要求可	プログラムで定義
¹ 名前は導入時に定義します。		

GETDRCEN (ディレクトリー項目 GET) レベル 2.0:

このコマンドは、フォルダー、文書、または両方のリストを取り出します。このコマンドは、ソース IBM i から送られることはありません。

パラメーター名	ソース	ターゲット
BGNNAM ¹	該当せず	IBM i名
DCLNAM ²	該当せず	プログラムで定義
FILATTRL	該当せず	サポート
FILCLS	該当せず	DIRECTORY または STRFIL のみ
FILHDD	該当せず	サポート
FILSYS	該当せず	サポート
MAXGETCN	該当せず	サポート
NAME ¹	該当せず	IBM i名
¹ 名前の形式はサーバーで定義します。		
² 名前は導入時に定義します。		

次の応答オブジェクトを使用することができます。

FILAL

ファイル属性リスト

GETREC (カーソル位置のレコードの GET) レベル 1.0:

このコマンドは、現行カーソル位置と示されているレコードを取り出してそれを返します。

パラメーター名	ソース	ターゲット
DCLNAM ¹	プログラムで定義	プログラムで定義
KEYVALFB	要求可	サポート
RECNRFB	要求可	サポート
RTNINA ²	必要時	サポート
UPDINT	送信不可	サポート
¹ 名前は導入時に定義します。		
² アプリケーションにより決まります。		

次の応答オブジェクトを使用することができます。

RECAL

レコード属性リスト

RECINA

非活動レコード (-1 は非サポート、最大長 = 2**15-2) (2 の 15 乗マイナス 2)

RECORD

固定長レコード (最大長 = 2**15-2)

GETSTR (サブストリーム GET) レベル 2.0 およびレベル 3.0:

このコマンドは、文書にあるストリーム・データを取り出すためのものです。このコマンドは、ソース IBM i から送られることはありません。

パラメーター名	ソース	ターゲット
DCLNAM ¹	該当せず	プログラムで定義
STRLEN	該当せず	サポート
STROFF	該当せず	サポート
STRPOS	該当せず	サポート

¹ 名前は導入時に定義します。

INSRECEF (EOF への挿入) レベル 1.0:

このコマンドは、ファイルの終わりにレコードを挿入します。

パラメーター名	ソース	ターゲット
DCLNAM ¹	プログラムで定義	プログラムで定義
KEYVALFB	要求可	サポート
RECCNT ²	必要時	サポート
RECNBRFB	要求可	サポート
RLSUPD	常に FALSE	サポート
UPDCSR	送信不可	サポート

¹ 名前は導入時に定義します。
² アプリケーションにより決まります。

次のコマンド・オブジェクトを使用することができます。

RECINA

非活動レコード (-1 は非サポート、最大長 = 2**15-2) (2 の 15 乗マイナス 2)

RECORD

固定長レコード (最大長 = 2**15-2)

次の応答オブジェクトを使用することができます。

KEYVAL

キー値

RECAL

レコード属性リスト

RECNBR

レコード番号

INSRECKY (キー値別のレコードの挿入) レベル 1.0:

このコマンドは、ファイル内に使用可能なスペースさえあれば、キー値に則して 1 つ以上のレコードを挿入します。

パラメーター名	ソース	ターゲット
DCLNAM ¹	プログラムで定義	プログラムで定義
RECCNT	必要時	サポート
RECNBRFB	要求可	サポート
RLSUPD	常に FALSE	サポート
UPDCSR	送信不可	サポート

¹ 名前は導入時に定義します。

次のコマンド・オブジェクトを使用することができます。

RECORD

固定長レコード (最大長 = 2**15-2) (2 の 15 乗マイナス 2)

IBM iは可変長レコードをサポートしないため、次の応答オブジェクトのみを返すことができます。

RECNBR

レコード番号

INSRECNB (番号別レコード挿入) レベル 1.0:

このコマンドは、レコード番号パラメーターに指定された位置に、1 つまたは複数のレコードを挿入します。

パラメーター名	ソース	ターゲット
DCLNAM ¹	プログラムで定義	プログラムで定義
KEYVALFB	要求可	サポート
RECCNT	必要時	サポート
RECNBR	送信可	サポート
UPDCSR	送信不可	サポート

¹ 名前は導入時に定義します。

次のコマンド・オブジェクトを使用することができます。

RECINA

非活動レコード (-1 は非サポート、最大長 = 2**15-2) (2 の 15 乗マイナス 2)

RECORD

固定長レコード (最大長 = 2**15-2)

次の応答オブジェクトを使用することができます。

KEYVAL

キー値

LCKFIL (ファイル・ロック) レベル 1.0 およびレベル 2.0:

このコマンドは、要求元が後で使用できるようファイルをロックします。

パラメーター名	ソース	ターゲット
FILNAM ¹	ターゲット・システム名	IBM i名
LCKMGRNM	使用不可	無視
RQSFILLK	送信可	サポート
WAIT	送信可	サポート

¹ 名前の形式はサーバーが定義します。

LCKSTR (サブストリーム・ロック) レベル 2.0 およびレベル 3.0:

このコマンドは、ストリーム・ファイルのサブストリームをロックするのに使用します。このコマンドは、ソース IBM iから送られることはありません。

パラメーター名	ソース	ターゲット
DCLNAM ¹	該当せず	プログラムで定義
RQSSTRLK	該当せず	EXCSTRLK および SHRSTRLK のみ
STRLOC	該当せず	サポート
STROFF	該当せず	サポート
WAIT ²	該当せず	サポート

¹ 名前は導入時に定義します。

² WAIT パラメーターは、拒否も実行もされません。

LODREC (レコード・ファイル・ロード) レベル 1.0 およびレベル 2.0:

このコマンドは、レコード・ファイル全体をサーバー・システムに入れます。

パラメーター名	ソース	ターゲット
FILNAM ¹	送信可	IBM i名

¹ 名前の形式はサーバーが定義します。

次のコマンド・オブジェクトを使用することができます。

RECAL

レコード属性リスト

RECCNT

レコード・カウント

RECINA

非活動レコード (-1 は非サポート、最大長 = 2**15-2) (2 の 15 乗マイナス 2)

RECORD

固定長レコード (最大長 = 2**15-2)

LODSTRF (ストリーム・ファイル・ロード) レベル 2.0:

このコマンドは、クライアント・システムにあるストリーム・ファイル全体をサーバー・システムへ入れるのに使用します。このコマンドは、ストリーム・ファイル・コピー HPS API の使用時にソース IBM iによって送られます。

パラメーター名	ソース	ターゲット
FILNAM ¹	送信可	IBM i名

¹ 名前の形式はサーバーが定義します。

次のコマンド・オブジェクトを使用することができます。

STREAM

ストリーム

STRSIZ

ストリーム・サイズ

LSTFAT (ファイル属性リスト) レベル 1.0、レベル 2.0、およびレベル 3.0:

このコマンドは、ファイル、文書、またはフォルダーの属性を抜粋して検索します。

パラメーター名	ソース	ターゲット
FILATTRL	送信可	サポート
FILNAM ¹	ターゲット・システム名	IBM i名
DCLNAM ²	送信不可	サポート

¹ 名前の形式はサーバーで定義します。
² 名前は導入時に定義します。

次の応答オブジェクトを使用することができます。

FILAL

ファイル属性リスト応答データ

MODREC (更新インテント付きレコード変更) レベル 1.0:

このコマンドは、現行カーソル位置に影響を与えずに、現在更新インテントを指定されているレコードを変更するのに使用します。

パラメーター名	ソース	ターゲット
ALWMODKY	送信可	サポート
DCLNAM ¹	送信可	プログラムで定義

¹ 名前は導入時に定義します。

次のコマンド・オブジェクトを使用することができます。

RECORD

固定長レコード (最大長 = 2**15-2) (2 の 15 乗マイナス 2)

OPEN (ファイル・オープン) レベル 1.0 およびレベル 2.0:

このコマンドは、クライアント・システムにある使用プログラムと、サーバー・システムにあるオブジェクトとの間で論理接続を確立します。

パラメーター名	ソース	ターゲット
ACCINTLS	送信可	サポート
ACCMTHCL	送信可	サポート
DCLNAM ¹	プログラムで定義	プログラムで定義
FILATTRL	送信不可	サポート
FILSHR	送信可	サポート
PRPSHD	送信不可	ストリーム・ファイルの場合のみサポート

¹ 名前は導入時に定義します。

OPNDRC (ディレクトリー・オープン) レベル 2.0:

このコマンドは、サーバー・システムにあるフォルダーをオープンするためのコマンドです。このコマンドは、ソース IBM i から送られることはありません。

パラメーター名	ソース	ターゲット
ACCMTHCL	該当せず	DRCAM のみ
DCLNAM ¹	該当せず	プログラムで定義

¹ 名前は導入時に定義します。

PUTSTR (サブストリーム PUT) レベル 2.0 およびレベル 3.0:

このコマンドは、ストリーム・データを文書内に入れるのに使用します。このコマンドは、ソース IBM i から送られることはありません。

パラメーター名	ソース	ターゲット
DCLNAM ¹	該当せず	プログラムで定義
STROFF	該当せず	サポート
STRPOS	該当せず	サポート

¹ 名前は導入時に定義します。

次のコマンド・オブジェクトを使用することができます。

STREAM

ストリーム

QRYCD (現行ディレクトリー QUERY) レベル 2.0:

このコマンドは、現行ディレクトリーを調べるのに使用します。このコマンドは、ソース IBM i から送られることはありません。

パラメーター名	ソース	ターゲット
AGNNAM	該当せず	無視

次の応答オブジェクトを使用することができます。

DRCNAM

ディレクトリー名

注: ゼロのディレクトリー名長は、ルート・ディレクトリーが現行 ディレクトリーであることを示します。

QRYSPC (スペース QUERY) レベル 2.0:

このコマンドは、ユーザーが使用できる記憶域スペースの大きさを調べるのに使用します。このコマンドは、ソース IBM i から送られることはありません。

パラメーター名	ソース	ターゲット
AGNNAM	該当せず	無視

次の応答オブジェクトを使用することができます。

QRYSPCRD

スペース QUERY 応答データ

RNMDRC (ディレクトリー名変更) レベル 2.0:

このコマンドは、フォルダーまたはデータベース・ライブラリーの名前を変更します。コマンドは移動フォルダーをサポートせず、また、ソース IBM iによって送信されません。

パラメーター名	ソース	ターゲット
DRCNAM	該当せず	IBM i名
NEWDRCNM	該当せず	IBM i名

注: 名前の形式はサーバーで定義します。 総称名は使用できません。

RNMFIL (ファイル名変更) レベル 1.0 およびレベル 2.0:

このコマンドは、既存のデータベース・ファイルまたは文書の名前を変更するのに使用しますが、転送文書にも使用することができます。

パラメーター名	ソース	ターゲット
FILNAM ¹	送信可	IBM i名
NEWFILNM ²	送信可	IBM i名

¹ 名前の形式はサーバーで定義します。 総称名は、文書の場合のみ使用できます。
² 名前の形式はサーバーで定義します。

SBMSYSCMD (サーバー・コマンド投入) レベル 4.0:

このコマンドは、ターゲット・サーバーの制御言語構文のサーバー・コマンドを、サーバー・システムへ実行依頼するためのものです。

パラメーター名	ソース	ターゲット
SYSCMD ¹	送信可	サポート

¹ 実行するコマンド・STRING

SETBOF (ファイル始めへのカーソル設定) レベル 1.0:

このコマンドは、ファイルの始めの位置にカーソルを設定します。

パラメーター名	ソース	ターゲット
DCLNAM ¹	プログラムで定義	プログラムで定義

¹ 名前は導入時に定義します。

SETEOF (ファイル終わりへのカーソル設定) レベル 1.0:

このコマンドは、ファイルの終わりの位置にカーソルを設定します。

パラメーター名	ソース	ターゲット
DCLNAM ¹	プログラムで定義	プログラムで定義

¹ 名前は導入時に定義します。

SETFRS (第 1 レコードへのカーソルの設定) レベル 1.0:

このコマンドは、ファイルの第 1 レコードにカーソルを設定します。

パラメーター名	ソース	ターゲット
BYPINA ¹	必要時	サポート
DCLNAM ²	プログラムで定義	プログラムで定義
HLDCSR	要求可	サポート
KEYVALFB	要求可	サポート
RECNBRFB	要求可	サポート
RTNREC ³	送信可	サポート
UPDINT ³	送信可	サポート

¹ アプリケーションにより決まります。
² 名前は導入時に定義します。
³ IBM i優先の導入。

次の応答オブジェクトを使用することができます。

KEYVAL

キー値

RECAL

レコード属性リスト

RECINA

非活動レコード (-1 は非サポート、最大長 = 2**15-2) (2 の 15 乗マイナス 2)

RECNBR

レコード番号

RECORD

レコード

SETKEY (キーによるカーソル設定) レベル 1.0:

このコマンドは、指定されたキー値と、RELOPR に指定された関係演算子に基づいてカーソルの位置を決めます。

パラメーター名	ソース	ターゲット
DCLNAM ¹	プログラムで定義	プログラムで定義
HLDCSR	要求可	サポート
KEYVAL ²	最大 = 2000	最大 = 2000
KEYVALFB	要求可	サポート
RECNBRFB	要求可	サポート
RELOPR	送信可	サポート
RTNREC ³	送信可	サポート
UPDINT ³	送信可	サポート

¹ 名前は導入時に定義します。
² IBM iで使用可能な最大キー・サイズです。
³ IBM i優先の導入。

次の応答オブジェクトを使用することができます。

KEYVAL

キー値

RECAL

レコード属性リスト

RECNR

レコード番号

RECORD

レコード

SETKEYFR (キー順第 1 レコードへのカーソル設定) レベル 1.0:

このコマンドは、キー順ファイル内の第 1 レコードにカーソルを設定します。

パラメーター名	ソース	ターゲット
DCLNAM ¹	プログラムで定義	プログラムで定義
HLDCSR	要求可	サポート
KEYVALFB	要求可	サポート
RECNRFB	要求可	サポート
RTNREC ²	送信可	サポート
UPDINT ²	送信可	サポート
¹	名前は導入時に定義します。	
²	IBM i優先の導入。	

次の応答オブジェクトを使用することができます。

KEYVAL

キー値

RECAL

レコード属性リスト

RECNR

レコード番号

RECORD

レコード

SETKEYLM (キー限界設定) レベル 1.0:

このコマンドは、後続の **SETKEYNX** および **SETNXTKE** コマンドのキー値の限界を設定します。このコマンドは、ソース IBM iから送られることはありません。

パラメーター名	ソース	ターゲット
DCLNAM ¹	該当せず	プログラムで定義
KEYHLM ²	該当せず	サポート
KEYLLM ²	該当せず	サポート
¹	名前は導入時に定義します。	
²	アプリケーションにより決まります。	

SETKEYLS (キー順最終レコードへのカーソルの設定) レベル 1.0:

このコマンドは、キー順配列のファイルの最後にあるレコードにカーソルを設定します。

パラメーター名	ソース	ターゲット
DCLNAM ¹	プログラムで定義	プログラムで定義
HLDCSR	要求可	サポート
KEYVALFB	要求可	サポート
RECNBRFB	要求可	サポート
RTNREC ²	送信可	サポート
UPDINT ²	送信可	サポート
¹	名前は導入時に定義します。	
²	IBM i優先の導入。	

次の応答オブジェクトを使用することができます。

KEYVAL

キー値

RECAL

レコード属性リスト

RECNBR

レコード番号

RECORD

レコード

SETKEYNX (キー順の次のレコードへのカーソル設定) レベル 1.0:

このコマンドは、現在カーソルが示しているレコードに続くレコード、つまりファイルのキー順配列の次のレコードにカーソルを設定するためのコマンドです。

パラメーター名	ソース	ターゲット
BYPDMG ¹	送信不可	サポート
DCLNAM ²	プログラムで定義	プログラムで定義
HLDCSR	要求可	サポート
KEYVALFB	要求可	サポート
RECCNT ¹	必要時	サポート
RECNBRFB	要求可	サポート
RTNREC ³	送信可	サポート
UPDINT ³	送信可	サポート
¹	アプリケーションにより決まります。	
²	名前は導入時に定義します。	
³	IBM i優先の導入。	

次の応答オブジェクトを使用することができます。

KEYVAL

キー値

RECAL

レコード属性リスト

RECNR

レコード番号

RECORD

レコード

SETKEYPR (キー順の前のレコードへのカーソル設定) レベル 1.0:

このコマンドは、現在カーソルが示しているレコードより前にある、ファイルのキー順配列の直前のレコードにカーソルを設定するためのコマンドです。

パラメーター名	ソース	ターゲット
DCLNAM ¹	プログラムで定義	プログラムで定義
HLDCSR	要求可	サポート
KEYVALFB	要求可	サポート
RECCNT ²	必要時	サポート
RECNRFB	要求可	サポート
RTNREC ³	送信可	サポート
UPDINT ³	送信可	サポート
¹	名前は導入時に定義します。	
²	アプリケーションにより決まります。	
³	IBM i優先の導入。	

次の応答オブジェクトを使用することができます。

KEYVAL

キー値

RECAL

レコード属性リスト

RECNR

レコード番号

RECORD

レコード

SETLST (最終レコードへのカーソル設定) レベル 1.0:

このコマンドは、ファイルの最終レコードにカーソルを設定するのに使用します。

パラメーター名	ソース	ターゲット
BYPINA ¹	必要時	サポート
DCLNAM ²	プログラムで定義	プログラムで定義
HLDCSR	要求可	サポート
KEYVALFB	要求可	サポート
RECNRFB	要求可	サポート
RTNREC ³	送信可	サポート
UPDINT ³	送信可	サポート
¹	アプリケーションにより決まります。	
²	名前は導入時に定義します。	
³	IBM i優先の導入。	

次の応答オブジェクトを使用することができます。

KEYVAL

キー値

RECAL

レコード属性リスト

RECINA

非活動レコード (-1 は非サポート、最大長 = 2**15-2)

RECNBR

レコード番号

RECORD

レコード

SETMNS (カーソル・マイナス設定) レベル 1.0:

このコマンドは、カーソルで示されたファイル・レコード番号から、CSRDSP で指定されたレコード位置数を引いた位置にカーソルを設定するコマンドです。

パラメーター名	ソース	ターゲット
ALWINA ¹	必要時	サポート
CSRDSP ¹	送信可	サポート
DCLNAM ²	プログラムで定義	プログラムで定義
HLDCSR	要求可	サポート
KEYVALFB	要求可	サポート
RECNBRFB	要求可	サポート
RTNINA ¹	必要時	サポート
RTNREC ³	送信可	サポート
UPDINT ³	送信可	サポート

¹ アプリケーションにより決まります。
² 名前は導入時に定義します。
³ IBM i優先の導入。

次の応答オブジェクトを使用することができます。

KEYVAL

キー値

RECAL

レコード属性リスト

RECINA

非活動レコード (-1 は非サポート、最大長 = 2**15-2) (2 の 15 乗マイナス 2)

RECNBR

レコード番号

RECORD

レコード

SETNBR (レコード番号へのカーソル設定) レベル 1.0:

このコマンドは、**RECNR** で指定されたレコード番号によって示された、ファイル内のレコードにカーソルを設定します。

パラメーター名	ソース	ターゲット
ALWINA ¹	必要時	サポート
DCLNAM ²	プログラムで定義	プログラムで定義
HLDCSR	要求可	サポート
KEYVALFB	要求可	サポート
RECNR	送信可	サポート
RTNINA ¹	必要時	サポート
RTNREC ³	送信可	サポート
UPDINT ³	送信可	サポート

¹ アプリケーションにより決まります。
² 名前は導入時に定義します。
³ IBM i優先の導入。

次の応答オブジェクトを使用することができます。

KEYVAL

キー値

RECAL

レコード属性リスト

RECINA

非活動レコード (-1 は非サポート、最大長 = 2**15-2) (2 の 15 乗マイナス 2)

RECORD

レコード

SETNXT (次の番号へのカーソル設定) レベル 1.0:

このコマンドは、ファイル内の現行カーソル位置より 1 大きいレコード番号を持つ、次のレコードにカーソルを設定します。

パラメーター名	ソース	ターゲット
BYPDMG ¹	送信不可	サポート
BYPINA ¹	必要時	サポート
DCLNAM ²	プログラムで定義	プログラムで定義
HLDCSR	要求可	サポート
KEYVALFB	要求可	サポート
RECCNT ¹	必要時	サポート
RECNRFB ¹	必要時	サポート
RTNREC ³	送信可	サポート
UPDINT ³	送信可	サポート

¹ アプリケーションにより決まります。
² 名前は導入時に定義します。
³ IBM i優先の導入。

次の応答オブジェクトを使用することができます。

KEYVAL

キー値

RECAL

レコード属性リスト

RECINA

非活動レコード (-1 は非サポート、最大長 = 2**15-2) (2 の 15 乗マイナス 2)

RECNBR

レコード番号

RECORD

レコード

SETNXTKE (指定値と等しいキーをもつキー順内の次のレコードへのカーソルの設定) レベル 1.0:

このコマンドは、キー順の次のレコードのキー・フィールドの値が、KEYVAL パラメーターに指定された値と等しいときに、そのレコードにカーソルを設定します。このコマンドは、ソース IBM i から送られることはありません。

パラメーター名	ソース	ターゲット
DCLNAM ¹	該当せず	プログラムで定義
HLDCSR	該当せず	サポート
KEYVAL ²	該当せず	最大 = 2000
KEYVALFB	該当せず	サポート
RECNBRFB	該当せず	サポート
RTNREC ³	該当せず	サポート
UPDINT ³	該当せず	サポート
¹	名前は導入時に定義します。	
²	IBM i で使用可能な最大キー・サイズです。	
³	IBM i 優先の導入。	

次の応答オブジェクトを使用することができます。

KEYVAL

キー値

RECAL

レコード属性リスト

RECNBR

レコード番号

RECORD

レコード

SETPLS (カーソル・プラス設定) レベル 1.0:

このコマンドは、ファイル内でカーソルで示されたレコード番号に、**CSRDSP** で指定された整数のレコード数を加えた位置にカーソルを設定します。

パラメーター名	ソース	ターゲット
ALWINA ¹	必要時	サポート

パラメーター名	ソース	ターゲット
CSRDSP ¹	送信可	サポート
DCLNAM ²	プログラムで定義	プログラムで定義
HLDCSR	要求可	サポート
KEYVALFB	要求可	サポート
RECNBRFB	要求可	サポート
RTNINA ¹	必要時	サポート
RTNREC ³	送信可	サポート
UPDINT ³	送信可	サポート
¹	アプリケーションにより決まります。	
²	名前は導入時に定義します。	
³	IBM i優先の導入。	

次の応答オブジェクトを使用することができます。

KEYVAL

キー値

RECAL

レコード属性リスト

RECINA

非活動レコード (-1 は非サポート、最大長 = 2**15-2) (2 の 15 乗マイナス 2)

RECNBR

レコード番号

RECORD

レコード

SETPRV (前のレコードへのカーソルの設定) レベル 1.0:

このコマンドは、ファイル内で現行カーソル位置より 1 小さいレコード番号を持ったレコードにカーソルを設定します。

パラメーター名	ソース	ターゲット
BYPINA ¹	必要時	サポート
DCLNAM ²	プログラムで定義	プログラムで定義
HLDCSR	要求可	サポート
KEYVALFB	要求可	サポート
RECCNT ¹	必要時	サポート
RECNBRFB	要求可	サポート
RTNREC ³	送信可	サポート
UPDINT ³	送信可	サポート
¹	アプリケーションにより決まります。	
²	名前は導入時に定義します。	
³	IBM i優先の導入。	

次の応答オブジェクトを使用することができます。

RECAL

レコード属性リスト

RECINA

非活動レコード (-1 は非サポート、最大長 = 2**15-2)

RECNR

レコード番号

RECORD

レコード

SETUPDKY (キー値による更新インテント設定) レベル 1.0:

このコマンドは、KEYVAL で指定されたキー値に等しいキー値を持ったレコードに更新インテントを付けます。

パラメーター名	ソース	ターゲット
DCLNAM ¹	プログラムで定義	プログラムで定義
KEYVAL ²	最大 = 2000	最大 = 2000
KEYVALFB	要求可	サポート
RECNRFB	要求可	サポート
RTNREC ³	送信可	サポート
¹	名前は導入時に定義します。	
²	IBM iで使用可能な最大キー・サイズです。	
³	RTNREC (FALSE) のみがサポートされています。	

次の応答オブジェクトを使用することができます。

KEYVAL

キー値

RECAL

レコード属性リスト

RECNR

レコード番号

RECORD

レコード

SETUPDNB (レコード番号による更新インテント設定) レベル 1.0:

このコマンドは、ファイル内で **RECNR** で指定されたレコード番号で示されるレコードに更新インテントを付けるためのコマンドです。

パラメーター名	ソース	ターゲット
ALWINA ¹	必要時	サポート
DCLNAM ²	プログラムで定義	プログラムで定義
KEYVALFB	要求可	サポート
RECNR	送信可	サポート
RTNINA ¹	必要時	サポート
RTNREC ³	送信可	サポート

パラメーター名	ソース	ターゲット
¹	アプリケーションにより決まります。	
²	名前は導入時に定義します。	
³	RTNREC (FALSE) のみがサポートされています。	

次の応答オブジェクトを使用することができます。

KEYVAL

キー値

RECAL

レコード属性リスト

RECINA

非活動レコード (-1 は非サポート、最大長 = 2**15-2) (2 の 15 乗マイナス 2)

RECORD

レコード

ULDRECF (レコード・ファイル・アンロード) レベル 1.0:

このコマンドは、ターゲット・システムのレコード・ファイルからソース・システムへレコードを送るのに使用します。

パラメーター名	ソース	ターゲット
ACCDORD ¹	送信可	サポート
BYPDMG ¹	送信可	サポート
FILNAM ²	送信可	IBM i名
RTNINA ¹	送信可	サポート
¹	アプリケーションにより決まります。	
²	名前の形式はサーバーで定義します。	

次の応答オブジェクトを使用することができます。

RECAL

レコード属性リスト

RECCNT

レコード・カウント

RECINA

非活動レコード (-1 は非サポート、最大長 = 2**15-2) (2 の 15 乗マイナス 2)

RECORD

レコード

ULDSTRF (ストリーム・ファイル・アンロード) レベル 2.0:

このコマンドは、ターゲット・システムからソース・システムへ文書を送るためのコマンドです。このコマンドは、ストリーム・ファイル・コピー (QHFCPYSF) HFS API の使用時にソース IBM iによって送られます。

パラメーター名	ソース	ターゲット
BYPDGM	送信不可	FALSE (偽) のみ
FILNAM ¹	送信可	IBM i名
STRORD	送信不可	サポート

¹ 名前の形式はサーバーが定義します。

次の応答オブジェクトを使用することができます。

STREAM

ストリーム

STRPOS

ストリーム位置

STRSIZ

ストリーム・サイズ

UNLFIL (ファイル・アンロック) レベル 1.0 およびレベル 2.0:

このコマンドは、ファイル上にあり、要求元によってかけられている明示ファイル・ロックを解除するのに使用します。

パラメーター名	ソース	ターゲット
FILNAM ¹	ターゲット・システム名	IBM i名
LCKMGRNM	送信不可	無視
RLSFILLK	送信可	サポート

¹ 名前の形式はサーバーが定義します。

UNLIMPLK (暗黙レコード・アンロック) レベル 1.0:

このコマンドは、現在カーソルによってかけられているすべての暗黙レコード・ロックを解除するためのコマンドです。

パラメーター名	ソース	ターゲット
DCLNAM ¹	プログラムで定義	プログラムで定義

¹ 名前は導入時に定義します。

UNLSTR (サブストリーム・アンロック) レベル 2.0 およびレベル 3.0:

このコマンドは、ストリーム・ファイルのサブストリームをアンロックします。このコマンドは、ソース IBM iから送られることはありません。

パラメーター名	ソース	ターゲット
DCLNAM ¹	該当せず	プログラムで定義
STRLOC	該当せず	サポート

¹ 名前は導入時に定義します。

ユーザー・プロファイル権限:

ターゲット IBM i・ジョブと関連したユーザー・プロファイルが、DDM コマンドを処理するには、対応する CL コマンドの使用許可のために権限を認可されていなければなりません。ターゲット・システム・ジョブのユーザー・プロファイルが、DDM 要求を処理するには、ここにリストした CL コマンドの使用のために権限を認可されていなければなりません。

表 16. ユーザー・プロファイル権限 CL コマンド

受信する DDM コマンド	DDM コマンド記述	オブジェクト・タイプ	権限認可 CL コマンド
CHGDRC	現行ディレクトリー変更	FLR	なし
CHGFAT	ファイル属性変更	PFILE LF DOC/FLR	CHGPF CHGLF なし
CLOSE	ファイル・クローズ	FILE DOC	なし ¹ なし
CLRFIL	ファイル消去	FILE DOC	なし なし
CLSDRC	ディレクトリー・クローズ	FLR	なし
CPYFIL	ファイル・コピー	DOC	なし
CRTAIF	代替索引ファイル作成	LF	CRTLF
CRTDIRF	直接ファイル作成	PF	CRTPF
CRTKEYF	キー付きファイル作成	PF	CRTPF
CRTSEQF	順次ファイル作成	PF	CRTPF
CRTSTRF	ストリーム・ファイル作成	DOC	なし
CRTDRC	ディレクトリー作成	LIB FLR	CRTLIB CRTFLR
DELFIL	ファイル削除	FILE DOC	DLTF なし
DELDRC	ディレクトリー削除	LIB FLR	DLTLIB なし
GETDRCEN	ディレクトリー項目 GET	DOC/FLR	なし
LCKFIL	ファイル・ロック	FILE	ALCOBJ
LODRECF	レコード・ファイル・ロード (PUT)	FILE	なし ²
LSTFAT	ファイル属性リスト	FILE DOC/FLR	なし ³ なし
OPEN	ファイル・オープン	FILE DOC	なし ¹ なし
OPENDRC	ディレクトリー・オープン	FLR	なし
QRYSPP	QUERY スペース	USRPRF	なし ⁴
RNMDRC	ディレクトリー名変更	FLR LIB	RNMOBJ なし
RNMFIL	ファイル名変更	FILE DOC MBR	RNMOBJ なし
			RNMM
UNLFIL	ファイル・アンロック	FILE	なし ⁵
ULDRECF	ファイルからのレコードのアンロード	FILE	なし ²

¹	コマンドの権限認可があるかどうかは検査されません。IBM i ユーザーがファイルをオープンしたりクローズするときには、コマンド・インターフェースを用いる以外にも手段があるからです。
²	コマンドの権限認可があるかどうかは検査されません。それは、CL コマンドと DDM LODRECF/ULDRECF コマンドとの間には、直接の 1 対 1 のマッピングはないからです。
³	DSPFD コマンドと DSPFFD コマンドの権限認可があるかどうかは検査されません。それは、いずれのコマンドを検査すべきかを判別できないからです。しかも、クライアント・システムで DDM コマンドが出されるときに条件が分からないからです。
⁴	ユーザーが使用できるスペースは、 DSPUSRPF コマンドを出せば得ることができますが、これは、このコマンドを使って使用できるデータのうちの一部にすぎません。
⁵	CL DLCOBJ コマンドの権限認可を受けているかどうかは検査されません。それは、リモート・ユーザーがファイルを割り振ることができたときには、DDM でそのファイルの割り振りを解除できなければならないからです。

次の表は、前の表で使用されているオブジェクト・タイプ・コードを説明したものです。

表 17. オブジェクト・タイプ・コードの定義

オブジェクト・タイプ	オブジェクト・タイプの定義
DOC	文書
FLR	フォルダー
PF	物理ファイル
LF	論理ファイル
LIB	ライブラリー
MBR	メンバー
SRCF	ソース物理ファイル
USRPRF	ユーザー・プロファイル

DDM での IBM i と CICS 間の考慮事項

このトピックでは、IBM i DDM を使用して CICS リモート・ファイルにアクセスする場合のプログラミングに関する考慮事項について説明します。

注: システム/370 ホスト・システムには、バージョン 1.7 またはそれ以降の CICS/OS/VS と、バージョン 1.1 の CICS/DDM が導入されていなければなりません。

IBM i 言語、ユーティリティー、およびライセンス・プログラム:

このトピックで説明されている IBM i 言語、ユーティリティー、およびライセンス・プログラムは、リモート CICS ファイルにアクセスできます。

- IBM i 上で以下の言語を使用して作成したプログラムであれば、リモート CICS ファイルへアクセスすることができます。

ILE C プログラミング言語

259 ページの『ILE C に関する考慮事項』を参照してください。

CL 253 ページの『IBM i の CL に関する考慮事項』を参照してください。

ILE COBOL プログラミング言語

257 ページの『ILE COBOL に関する考慮事項』を参照してください。

255 ページの『PL/I に関する考慮事項』を参照してください。

ILE RPG プログラミング言語

260 ページの『ILE RPG に関する考慮事項』を参照してください。

- BASIC で作成したプログラムを実行すると、リモート CICS ファイルへアクセスしたときに予測できない結果になることがあります。
- IBM i QUERY を使用すれば、リモートの入力順データ・セット (ESDS)、相対レコード・データ・セット (RRDS)、およびキー順データ・セット (KSDS) へアクセスすることができます。ただし IBM i QUERY では、DDM を介して仮想記憶アクセス方式 (VSAM) ファイルへアクセスすることはできません。
- IBM i 言語、ユーティリティー、およびライセンス・プログラムの代わりに、IBM i Access Family ライセンス・プログラムを実行すると、リモート CICS ファイルにアクセスしたときに予測できない結果になることがあります。

注: 高水準言語の中には、サーバー・データベースの入出力フィードバック域へアクセスできるものもあります。リモート VSAM RRDS へアクセスすれば、この区域には相対レコード番号が入っています。ただし、他のタイプの VSAM データ・セットへアクセスした場合には、相対レコードは認識されず、相対レコード番号として -1 の値が返されます。

システム/370 ホストでのデータの保管法が異なるために、システム/370 ホストが読み取ったり書き込んだりする CICS ファイルへアクセスする場合には、これ以外の配慮が必要なこともあります。たとえば、システム/370 ホストでの浮動小数点の表示法は、IBM i での浮動小数点数の表示法とは異なります。

DDM ファイル作成 (CRTDDMF) の考慮事項:

IBM i オペレーティング・システム上で稼働するアプリケーションからリモート・ファイルへアクセスするには、プログラマーは、CRTDDMF コマンドを使用して DDM ファイルという名前のオブジェクトを作成する必要があります。

このコマンドの ACCMTH パラメーターを使うと、リモート・ファイルをオープンするときに、どの DDM アクセス方法を使用すべきかが示されます。*RMTFILE を使用すると、IBM i DDM は、次のものと互換性のあるアクセス方法を選択します。

- アクセスする VSAM データ・セットのタイプ
- その VSAM データ・セットの場合に CICS/DDM がサポートするアクセス方法

下の表に、ACCMTH パラメーターに指定できる値が、どのように VSAM データ・セットと対応しているかを示します。

表 18. IBM i CRTDDMF コマンドの ACCMTH パラメーター

ACCMTH パラメーター値	VSAM データ・セット編成			
	ESDS	RRDS	KSDS	VSAM パス
*ARRIVAL	R	R	E	E
*KEYED	E	E	R	R
*BOTH	E	O	O	O
*RANDOM	E	O	O	O
*SEQUENTIAL	R	O	O	O
*COMBINED	E	O	E	E

図中の記号の意味は以下のとおりです。

R VSAM データ・セットにアクセスするのに、このパラメーターが必須であることを示します。

O VSAM データ・セットへアクセスするとき、このパラメーターはオプションであることを示します。

E このパラメーターで、IBM i メッセージが出ることを示します。

パフォーマンスを向上させるのに IBM i ユーザーは、ACCMTH パラメーターに *RMTFILE 以外の値を指定することができます。リモート VSAM データ・セットへアクセスするとき、サーバー・メッセージが出ないようにするには、表 18 に示した値を使用します。

VSAM データ・セットが CICS システムに対して定義されている場合、RMTFILE ファイル・パラメーターに指定する値は、CICS DFHFCT マクロの DATASET パラメーターに指定した値と同じになっている必要があります。

IBM i の CL に関する考慮事項:

IBM i の CL コマンドを使用してリモート CICS システム上の VSAM データ・セットにアクセスするときには、本書の説明以外に、以下の事項も考慮する必要があります。

注: 下記のトピックに含まれていないコマンドの場合、本書に述べた説明以外の考慮事項はありません。

オブジェクト割り振り (ALCOBJ):

CICS システム・プログラマーが、CICS/DDM 提供の排他的ファイル・ロック・プログラムを、同プログラムの特殊バージョンに差し替えた場合を除き、オブジェクト割り振り (ALCOBJ) コマンドを使用してリモート VSAM データ・セットを割り振るときには、*SHRRD または *SHRUPD のロック状態値を使用する必要があります。他のロック状態値ではすべて、IBM i システム・メッセージが出ることとなります。

物理ファイル・メンバー消去 (CLRPFM):

物理ファイル・メンバー消去 (CLRPFM) コマンドでは、リモート CICS システムにある VSAM データ・セットを消去することはできません。

ファイル・コピー (CPYF):

ファイル・コピー (CPYF) コマンドで、CICS システムに対して定義されたりリモート VSAM データ・セットへアクセスすることができます。

ただし、以下の事項を考慮する必要があります。

- TOFILE パラメーターがリモート VSAM データ・セットである場合:
 - CRTFILE パラメーターには、*NO の値を指定する必要があります。
 - MBROPT パラメーターには、*ADD の値を指定する必要があります。
 - FMTOPT パラメーターには、*NOCHK の値を指定する必要があります。
- TOFILE パラメーターがリモート VSAM ESDS または KSDS であれば、COMPRESS パラメーターには *YES の値を指定する必要があります。

CPYTOTAP、CPYFRMTAP、および CPYSPLF コマンド:

テープへのコピー (CPYTOTAP)、およびテープからのコピー (CPYFRMTAP) コマンドは、CICS システムに対して定義されたりリモート VSAM データ・セットにアクセスします。

ただし、MBROPT パラメーターに *ADD を指定する必要があります。スプール・ファイル・コピー (CPYSPLF) コマンドは、リモート VSAM データ・セットにアクセスできません。

オブジェクト割り振り解除 (DLCOBJ):

オブジェクト割り振り解除 (DLCOBJ) コマンドを使えば、リモート VSAM データ・セットに正常にかけられたすべてのロックを解除することができます。

DSPFD および DSPFFD コマンド:

ファイル記述表示 (DSPFD) コマンドとファイル・フィールド記述表示 (DSPFFD) コマンドを用いれば、リモート VSAM データ・セットに関する詳細を表示することができます。

ただし、フィールドに関する詳細のうち多くのものは、CICS/DDM で取り出すことはできず、IBM i DDM へ返されることもありません。CICS/DDM から返されないフィールドについては、次の表を参照してください。

表 19. CICS/DDM のファイル記述およびファイル・フィールド記述

フィールド	値
作成日	ゼロ
現行レコード数	ゼロ
データ・スペースのバイト数	ゼロ
ファイル・レベル識別コード	ゼロ
ファイル・テキスト記述	ゼロ
様式レベル識別コード	ブランク
最終変更日時	ゼロ
メンバー作成日	ゼロ
メンバー満了日	*NONE
メンバー・レベル識別コード	ゼロ
メンバー・サイズ	*NOMAX
削除済みレコード数	ゼロ
テキスト記述	ブランク
合計削除済みレコード数	ゼロ
総メンバー・サイズ	ゼロ
合計レコード数	ゼロ

注: ここに示した値は、そのファイルの実際のデータを表すものではありません。これらの値は、CICS/DDM から返されない情報のデフォルト値になります。

ファイル・タイプが論理ファイルのときには、ここに示した情報は、固有キーは不要であることを表しています。実際、CICS/DDM には、固有キーが必要かどうかは分かりません。

場合によっては IBM i ユーザーは、アクセスしようとする VSAM データ・セットのタイプを明らかにしなければならぬことがあります。DSPFD コマンドを使用して表示できる次のような情報を使用すれば、IBM i ユーザーは、以下の事項について判断することができます。

- ファイル・タイプが論理ファイルであれば、VSAM データ・セットは VSAM パスです。
- ファイル・タイプが物理ファイルであって、アクセス・パスがキー順であれば、VSAM データ・セットは KSDS です。
- 他の場合はすべて、VSAM データ・セットは RRDS または ESDS のいずれかです。IBM i ユーザーが RRDS または ESDS を判別しなければならない場合、CICS システム・プログラマーに問い合わせなければなりません。

物理ファイル・メンバー表示 (DSPPFM):

物理ファイル・メンバー表示 (DSPPFM) コマンドを使用すれば、リモートの相対レコード・データ・セット (RRDS) にアクセスすることができます。このコマンドは、他のタイプの VSAM データ・セットには使用できません。

データベース・ファイル・オープン (OPNDBF):

データベース・ファイル・オープン (OPNDBF) コマンドを使用すれば、リモート VSAM データ・セットをオープンすることができます。

ただし、ACCPH パラメーターに *ARRIVAL を指定し、リモート・データ・セットが VSAM キー順データ・セット (KSDS) または VSAM パスであれば、IBM i からシステム・メッセージが出ます。

データベース・ファイル一時変更 (OVRDBF):

データベース・ファイル一時変更 (OVRDBF) コマンドを用いれば、ローカル・データベース・ファイルをリモート VSAM データ・セットに一時変更することができます。

ただし、以下のことを考慮に入れなければなりません。

- *RRN の POSITION 値は、リモート VSAM データ・セットが RRDS の場合に有効です。他のどのタイプの VSAM データ・セットの場合も、*RRN を指定すると、システム・メッセージが出される原因になります。
- *KEYB または *KEYBE の POSITION 値では、リモート・ファイルが VSAM パスのときには、IBM i システム・メッセージが出される原因となります。
- CICS システム・プログラマーが、CICS/DDM 提供の排他的ファイル・ロック・プログラムを差し替えているのでない限り、RCDFMTLCK パラメーターには *SHRRD または *SHRUPD のロック状態値を指定する必要があります。他のロック状態値ではすべて、システム・メッセージが出ることとなります。
- CICS/DDM から、IBM i DDM へファイルの実際の日返されません。その代わりに、満了日は不明であることを示す特殊値が返されます。EXPCHK パラメーターに *YES の値を指定していても、同じことです。

ネットワーク・ファイル受け取り (RCVNETF):

ネットワーク・ファイル受け取り (RCVNETF) コマンドを使えば、リモート CICS システムに対して定義された VSAM データ・セットにアクセスすることができます。ただし、MBROPT パラメーターには、*ADD の値を指定する必要があります。

IBM i および CICS の言語に関する考慮事項:

ILE COBOL、ILE C、IBM i システム/36 互換 RPG II、または ILE RPG 言語を使用する IBM i アプリケーション・プログラマーの場合、以下のトピックに示す事項に注意してください。

PL/I に関する考慮事項:

以下のトピックでは、PL/I を使って IBM i からリモート VSAM データ・セットにアクセスするときに、守らなければならない制約事項を要約します。これらの制約事項は、本書で前述した制約事項を補足するものです。

PL/I ファイル・オープン要求:

IBM i DDM ユーザーは、PL/I プログラムでリモート CICS ファイルへアクセスできます。

RECORD ファイル属性を使用してファイルをオープンするときには、以下の表に指定されたファイル属性をプログラムで使用する必要があります。この表にある値を確認すれば、IBM i データベース・ファイルへのアクセスとリモート VSAM データ・セットへのアクセスにおける相違を見分けることができます。

注: リモート・ファイルは、PL/I STREAM ファイル属性を使ってオープンすることもできます。ただし、STREAM ファイル属性を使用して VSAM KSDS をオープンすると、サーバー・メッセージが出ます。これは、VSAM KSDS 内のレコードを到着順に処理できないためです。

CICS システムで CICS/DDM の排他的ファイル・ロック・プログラムを差し替えているのでない限り、リモート VSAM データ・セットをオープンするときに、ENVIRONMENT パラメーターに EXCL や EXCLRD のファイル・ロック・オプションを使用することはできません。

表 20. PL/I ファイル属性

PL/I ファイル属性	VSAM データ・セット編成			
	ESDS	RRDS	KSDS	VSAM パス
SEQUENTIAL	R	O	O	O
DIRECT	E	O	O	O
SEQL KEYED	E	O	O	O
INPUT	O	O	O	O
OUTPUT	O	O	O	E
UPDATE	O	O	E	E
CONSECUTIVE	R	R	E	E
INDEXED	-	-	R	R

図中の記号の意味は以下のとおりです。

R VSAM データ・セットにアクセスするには、この属性が必要なことを示します。

O VSAM データ・セットにアクセスするとき、この属性はオプションであることを示します。

E PL/I でこの属性を使用することはできますが、VSAM データ・セットにアクセスすると、オープンに失敗することを示します。

- このオプションは、キー付きファイルの場合のみ有効であることを示します。

PL/I 入出力要求:

このトピックでは、PL/I 入出力要求のタイプとその制約事項について説明します。

読み取り要求

- VSAM パスにアクセスする場合、BEFORE や EQLBFR の KEYSEARCH パラメーター値は CICS/DDM ではサポートされません。ただしこれらのパラメーター値は、VSAM KSDS へアクセスするときにはサポートされます。
- PREVIOUS と LAST の POSITION パラメーター値は、VSAM パスへアクセスするときにはサポートされません。ただしこれらのパラメーター値は、VSAM KSDS へアクセスするときにはサポートされます。
- DIRECT 属性または SEQUENTIAL KEYED 属性を使用して VSAM ESDS をオープンすることはできないため、相対レコード番号でレコードへアクセスすることも、KEY パラメーターや KEYTO パラメーターを使用して相対レコード番号を戻すこともできません。
- VSAM KSDS 代替索引と VSAM 代替索引は常に単一のキー・フィールドとして定義されるため、NBRKEYFLDS パラメーターを使用してはなりません。

書き込み要求

- VSAM RRDS へレコードを書き込むときには、KEYFROM パラメーターを使用することはできません。
- 同一キー値を持つレコードがすでに入っている VSAM KSDS へ書き込むときには、WRITE 要求を使用することはできません。
- OUTPUT ファイル属性も UPDATE ファイル属性も VSAM パスをオープンするために使用することはできないため、VSAM パスへレコードを書き込むことはできません。その代わりに、アプリケーション・プログラムは、VSAM パスの基本データ・セットを使用してレコードを書き込まなければなりません。

再書き込み要求

- VSAM KSDS のレコードを再書き込みしようとしたとき、そのレコードのキー値が変更されていれば、REWRITE を使用することはできません。
- UPDATE ファイル属性を使用して VSAM パスをオープンすることはできないため、VSAM パス内のレコードを再書き込みすることはできません。代わりに、アプリケーション・プログラムは、VSAM パスの基本データ・セットを使用してレコードを再書き込みする必要があります。

削除要求

- VSAM ESDS のレコードを削除するときには、DELETE を使用することはできません。
- UPDATE ファイル属性を使用して VSAM パスをオープンすることはできないため、VSAM パス内のレコードを削除することはできません。その代わりに、アプリケーション・プログラムは、VSAM パスの基本データ・セットを使用してレコードを削除する必要があります。ただし、VSAM パスの基本データ・セットが VSAM ESDS であれば、DELETE は使用できません。

ILE COBOL に関する考慮事項:

以下のトピックでは、ILE COBOL プログラミング言語を使用して IBM i からリモート VSAM データ・セットにアクセスするときに守らなければならない制約事項を要約します。

この制約は、特に注記がない限り、システム/36 にも IBM i にも適用されます。これらの制約事項は、本書で前述した制約事項を補足するものです。

ILE COBOL SELECT 文節:

IBM i ユーザーは、ILE COBOL プログラミング言語を使用してリモート CICS ファイルにアクセスできます。

ただし、ILE COBOL SELECT 文節は、以下の表に指定したファイル編成とアクセス方法を使用する必要があります。

表 21. ILE COBOL ファイル編成とアクセス方法

ILE COBOL プログラミング言語	VSAM データ・セット編成				
	ESDS	RRDS	KSDS	VSAM パス	プログラム指定の編成
プログラム指定のアクセス方法	ESDS	RRDS	KSDS	VSAM パス	プログラム指定の編成
順次	順次	X	X	E	E
相対	順次	E	X	E	E
	ランダム	E	X	E	E
	動的	E	X	E	E
索引付き	順次	-	-	X	X
	ランダム	-	-	X	X
	動的	-	-	X	X

表 21. ILE COBOL ファイル編成とアクセス方法 (続き)

ILE COBOL プログラミング言語	VSAM データ・セット編成				プログラム指定の編成
	ESDS	RRDS	KSDS	VSAM パス	
プログラム指定のアクセス方法					
<p>図中の記号の意味は以下のとおりです。</p> <p>X このアクセス方法は使用できることを示します。</p> <p>E ILE COBOL プログラミング言語ではこのアクセス方法が認められても、VSAM データ・セットへアクセスするとオープンが失敗することを示します。IBM i メッセージが作成されます。</p> <p>– このオプションは、キーなしファイルには有効でないことを示します。どのようなキーなしファイルの場合も、索引付きファイル編成を選択すると、常に IBM i メッセージが出されます。これは、ファイルがローカル・ファイルの場合にも当てはまります。</p> <p>注:</p> <p>1. VSAM パスにアクセスするときには、WITH DUPLICATE 句を使用する必要があります。</p> <p>2. VSAM KSDS へアクセスするときには、WITH DUPLICATE 句を使用してはなりません。</p>					

ILE COBOL ステートメント:

以下のトピックでは、ILE COBOL ステートメントを使ってリモート VSAM データ・セットにアクセスするときに注意しなければならない考慮事項について説明します。

ILE COBOL OPEN ステートメント

リモート CICS ファイルにアクセスする場合、ILE COBOL OPEN ステートメントは、以下の表に示したオープン・モードを使用する必要があります。

表 22. ILE COBOL プログラミング言語を使用した CICS ファイルのオープン

ILE COBOL オープン・モード	VSAM データ・セット編成			
	ESDS	RRDS	KSDS	VSAM パス
入力	X	X	X	X
出力	E	E	E	E
入出力	X	X	X	E
拡張	X	–	–	–
<p>図中の記号の意味は以下のとおりです。</p> <p>X オープン・モードを指定できることを示します。</p> <p>E オープン・モードが認められても、ILE COBOL OPEN ステートメントが VSAM データ・セットへアクセスするとオープン操作が失敗することを示します。IBM i オペレーティング・システムからメッセージが出ます。</p> <p>– オープン・モードは適用できないことを示します。</p>				

ILE COBOL READ ステートメント

- VSAM パスへのアクセスでは、PRIOR 句と LAST 句は CICS/DDM によってサポートされません。VSAM KSDS へアクセスするときには、サポートされます。

- RELATIVE ファイル編成は VSAM RRDS をオープンする場合しか使用できないため、リモート・ファイルが VSAM RRDS であるときを除き、相対レコード番号でレコードへアクセスすることも、リモート・ファイルから相対レコード番号を返すこともできません。

ILE COBOL WRITE ステートメント

- ILE COBOL プログラムを IBM i オペレーティング・システム上で実行している場合、またファイルが RELATIVE ファイル編成を指定してオープンされたものである場合、WRITE ステートメントは機能しません。
- レコードを VSAM KSDS へ書き込もうとしたとき、データ・セットに同一キー値を持ったレコードがすでに入っている場合、WRITE ステートメントを使用することはできません。
- 入出力オープン・モードや出力オープン・モードを使用して VSAM パスをオープンすることはできないため、VSAM パスへレコードを書き込むことはできません。その代わりに、アプリケーション・プログラムは、VSAM パスの基本データ・セットを使用してレコードを書き込まなければなりません。

ILE COBOL REWRITE ステートメント

- VSAM KSDS のレコードを再書き込みしようとしたとき、そのレコードのキー値が変更されていれば、REWRITE ステートメントを使用することはできません。
- 入出力オープン・モードを使用して VSAM パスをオープンすることはできないため、VSAM パスにレコードを再書き込みすることはできません。代わりに、アプリケーション・プログラムは、VSAM パスの基本データ・セットを使用してレコードを再書き込みする必要があります。

ILE COBOL START ステートメント

- オープン・モードが INPUT であれば、START ステートメントを使用できます。

ILE COBOL DELETE ステートメント

- VSAM ESDS をオープンするには順次ファイル編成を使用しなければならないため、VSAM ESDS 内のレコードを削除することはできません。
- 入出力オープン・モードを使って VSAM パスをオープンすることはできないため、VSAM パス内のレコードを削除することはできません。その代わりに、アプリケーション・プログラムは、VSAM パスの基本データ・セットを使用してレコードを削除する必要があります。ただし、VSAM パスの基本データ・セットが VSAM ESDS であれば、DELETE は使用できません。

ILE C に関する考慮事項:

以下のトピックでは、ILE C プログラミング言語を使用して IBM i オペレーティング・システムからリモート VSAM データ・セットにアクセスする場合の考慮事項を要約します。

ILE C オープンに関する考慮事項

ILE C プログラミング言語では順次入出力しかサポートされていないため、KSDS や VSAM パスがオープンされていても、オープン操作は失敗します。

オープン・モードに関する考慮事項

このトピックでは、ILE C プログラミング言語を使用する場合のオープン・モードに関する考慮事項を示します。

表 23. ILE C プログラミング言語を使用した CICS ファイルのオープン

ILE C オープン・モード	VSAM データ・セット編成			
	ESDS	RRDS	KSDS	VSAM パス
r, rb	X	X	X	X
w, wb	E	E	E	E
w+, wb+, w+b, a+, ab+, a+b, r+, rb+, r+b, a, ab	X	X	X	E
a, ab	X	—	—	—

図中の記号の意味は以下のとおりです。

X オープン・モードを使用できます。

E ILE C プログラミング言語でオープン・モードが認められていても、VSAM データ・セットへアクセスするとオープンは失敗します。

— オープン・モードは適用できません。

ILE RPG に関する考慮事項:

以下のトピックでは、IBM i システム/36 互換 RPG II または ILE RPG プログラミング言語を使って、IBM i オペレーティング・システムからリモート VSAM データ・セットにアクセスするときに守らなければならない制約事項を要約します。

これらの制約事項は、本書で前述した制約事項を補足するものです。

ファイル記述仕様書:

IBM i ユーザーは、IBM i システム/36 互換 RPG II 言語または ILE RPG プログラミング言語を使って、リモート VSAM データ・セットにアクセスできます。

ただし、ファイル記述仕様書を使って選択した ILE RPG 処理方式のすべてが、リモート VSAM データ・セットへアクセスできるとは限りません。使用するべきファイル記述仕様書を決めるには、次の表を参照してください。上記の表に示したものの以外のファイル記述仕様書を使用しても、CICS/DDM は、ファイルのオープン要求を拒否します。

表 24. リモート VSAM ESDS の ILE RPG 処理方式

処理のタイプ	桁番号						
	15	16	19	28	31	32	66
連続処理	I	P	F				
	I	S	F				
	I	T	F				
	I	D	F				
	U	P	F				
	U	S	F				
	U	D	F				
レコード追加のみ	O						A

表 25. リモート VSAM RRDS の ILE RPG 処理方式

処理のタイプ	桁番号						
	15	16	19	28	31	32	66
連続処理	I I I U U U	P S D P S D	F F F F F F				
連鎖によるランダム処理 注を参照 注を参照	I I U U	C C C C	F F F F	R R R R			A A
アドレスアウトによるランダム 処理	I I I U U U	P S F P S F	F F F F F F	R R R R	I I I I		
連続処理、ランダム処理、 または両方 注を参照 注を参照	I I U U	F F F F	F F F F				A A
注: 相対レコード番号処理を示すには、53 桁目に K を、そして 54 桁から 59 桁までには RECNO を指定する必要があります。							

表 26. VSAM KSDS の ILE RPG 処理方式

処理のタイプ	桁番号						
	15	16	19	28	31	32	66
順次	I	P	F	L		I	A
キー順、追加なし	I	S	F	L		I	A
キー順、追加なし	I	D	F	L		I	A
キー順、追加なし	I	P	F	L		I	A
キー順、追加あり	I	S	F	L		I	A
キー順、追加あり	I	D	F	L		I	A
キー順、追加あり	U	P	F	L		I	A
キー順、追加なし	U	S	F	L		I	A
キー順、追加なし	U	D	F	L		I	
キー順、追加なし	U	P	F			I	
キー順、追加あり	U	S	F			I	
キー順、追加あり	U	D	F			I	
キー順、追加あり	I	P	F			I	
限界による	I	S	F			I	
限界による	I	F	F			I	
限界による	U	D	F			I	
限界による	U	P	F			I	
限界による	U	D	F			I	
限界による	U	F	F			I	
限界による	I	F	F			I	
限界による、追加	I	F	F			I	
限界による、追加							
連鎖によるランダム処理	I	C	F	R		I	A
追加なし	I	C	F	R		I	A
追加あり	U	C	F	R		I	
追加なし	U	C	F	R		I	
追加あり							
アドレスアウトによるランダム処理	I	P	F	R	I	I	
	I	S	F	R	I	I	
	U	P	F	R	I	I	
	U	S	F	R	I	I	
順次処理、ランダム処理、 または両方	I	F	F			I	A
キー順、追加なし	I	F	F			I	A
キー順、追加あり	U	F	F			I	
キー順、追加あり	U	F	F			I	
キー順、追加なし							
キー順、追加なし							
レコード追加のみ	O					I	A

表 27. リモート VSAM 経路の場合の処理方式

処理のタイプ	桁番号						
	15	16	19	28	31	32	66
順次	I	P	F	L		I	
キー順、追加なし	I	S	F	L		I	
キー順、追加なし	I	D	F	L		I	
キー順、追加なし	I	P	F	L		I	
限界による	I	S	F			I	
限界による	I	F	F			I	
限界による	I	D	F			I	
連鎖によるランダム処理 追加なし	I	C	F	R		I	
アドレスアウトによるランダム 処理	I I	P S	F F	R R	I I	I I	
順次処理、ランダム処理、 または両方 キー順、追加なし	I	F	F			I	

ILE RPG 入出力操作:

リモート VSAM データ・セットへアクセスするときには、以下の制約事項に注意してください。

- 相対レコード番号別にレコードを読み取ったり追加できるのは、リモート VSAM データ・セットが RRDS の場合だけです。相対レコード番号別にランダム処理を行えるのは、VSAM RRDS を処理する場合だけです。
- ESDS は削除不可能であるため、ESDS 内のレコードを削除する要求を出すことはできません。
- リモート VSAM データ・セットが VSAM パスであるときには、この処理方式で 15 桁目に更新仕様または出力仕様を使用することはできません。その代わりに、VSAM パスの基本データ・セットを使用して、全レコードの追加、更新、または削除の要求を出さなければなりません。ただし、VSAM パスの基本データ・セットが VSAM ESDS であれば、DELETE は使用できません。
- READP 命令コードを使用しても、VSAM パス内の直前のレコードを読み取ることはできません。
- VSAM KSDS では重複キーは認められないため、KSDS 内の既存レコードのキーと重複するレコードの追加を要求しても追加されません。
- KSDS へアクセスするときには、レコードのキー値が変わるような更新要求を出すことはできません。
- ILE RPG プログラミング言語の場合、*HIVAL を使用すれば、リモート KSDS の最終レコードを取り出すことができます。ただし、VSAM パスへアクセスするときには、*HIVAL を使用することはできません。

IBM i と他の IBM システムでの DDM 使用の比較

このトピックでは、IBM i とシステム/36 の間、および IBM i とシステム/38 の間での DDM の相違点について説明します。

IBM iとシステム/36 の DDM の相違点:

このトピックでは、IBM iとシステム/36 の間の相違点をリストします。

- ネットワーク資源ディレクトリー (NRD) プロシージャは、IBM iではサポートされません。
 - システム/36 NRD は IBM i上で DDM ファイルが入っている 1 つ以上のライブラリーを使用しました。1 つのシステム/36 NRD 項目は、IBM iにある 1 つの DDM ファイルと等価になっています。
 - EDITNRD および DELNRD プロシージャの代わりに、**DDM ファイル処理 (WRKDDMF)** コマンドを使用します。
 - LISTNRD プロシージャの代わりに、**DDM ファイル表示 (DSPDDMF)** コマンドを使用します。
 - RESTNRD プロシージャの代わりに、**オブジェクト復元 (RSTOBJ)** コマンドを使用します。
 - SAVNRD プロシージャの代わりに、**オブジェクト保管 (SAVOBJ)** コマンドを使用します。
 - システム/36 上の NRD 項目がリモート・ファイルを参照しており、SAVE または RESTORE プロシージャが要求された場合、システム/36 はリモート・ファイルのデータを保管または復元します。**IBM i のオブジェクト保管 (SAVOBJ)** または**オブジェクト復元 (RSTOBJ)** コマンドは、リモート・データではなく、DDM ファイルの定義だけを保管または復元します。
 - システム/36 上で日付の異なった複数のファイルを使用すると、最新日付のファイルが選択されます。システム/36 から IBM iに対する実行の場合は、データベース・ファイル内の最新日付のファイルである最終メンバーにアクセスするには、NRD 項目に *LAST のメンバー名を指定する必要があります。メンバー名を指定しないと、*FIRST が使用されます。
- システム/36 のソース・システムにある NRD 内のリモート・ラベルは、サーバー・システムで必要とされている構文になっていなければなりません。
- システム/36 と IBM iでは、1 つのファイルに割り振られるレコード数が異なります。システム/36 でのファイル・スペース割り振りはブロック単位 (2560 バイト) で実行されますが、IBM iでのファイル・スペース割り振りはレコード数で実行されます。たとえば、10 個の 100 バイト・レコードを保管できる順次ファイルをユーザーが要求すると、システム/36 は、1 ブロックのファイル・スペース (2560 バイト) を割り振り、可能な限りのファイル・スペース (2500 バイト) を使用するため、ユーザーは 25 個の 100 バイト・レコードを使用できることとなります。

IBM iは、正確に 10 個のレコードを割り振ります。ユーザーがシステム/36 上でこの割り振り方式を活用していた場合、IBM i上で DDM を使用して作成したファイル (拡張不可) では小さすぎる場合があります。

- システム/36 で使用される **DDM ファイルの終わり変更 (CHGEOF)** コマンドは、IBM i ではサポートされません。
- IBM iでは、**ファイル削除 (DLTF)** コマンドでデータの上書きはできません。システム/36 にアクセスしている IBM iユーザーがそのデータを上書きしたい場合には、IBM iでアプリケーション・プログラムを作成するか、またはユーザーがターゲットのシステム/36 にアクセスして、消去命令を使用して削除操作を実行する必要があります。
- システム/36 クライアント・システムは、削除できないファイルが入っている IBM i ターゲット・サーバーで直接ファイルを作成することはできません。IBM iは、どのようなファイル編成においても削除不可能なファイルをサポートすることはありません。
- システム/36 では、先頭バイトが 16 進 FF になっているレコードを削除可能なファイルに挿入することはできません。IBM iでならば、挿入可能です。
- システム/36 アプリケーション・プログラムをもう 1 つ別のシステム/36 に対して実行中の場合、資源が使用可能になるまでアプリケーション・プログラムは無限に待機します。IBM iとの間でシステム/36

アプリケーション・プログラムを実行している場合、資源が使用可能になるのを待っている間に、これらのアプリケーション・プログラムがタイムアウトになる可能性があります。

- 直接ファイルは、システム/36 では拡張できますが、IBM iでは拡張できません。直接ファイルを拡張可能ファイルとして IBM iで作成した場合、そのファイルには 3 つのエクス Tent が割り振られますが、そのファイルが拡張された場合でもその初期サイズに 3 つのエクス Tent を加えた以上のサイズにはなりません。
- クライアント・システムからサーバー・システムへのパス上にあるネットワークにシステム/36 サーバーがない場合、システム/36 中継機能はサポートされません。IBM i では、中継機能はサポートしません。拡張対等間通信ネットワーク機能 (APPN) を使用する必要があります。
- システム/36 のキー・フィールドは、必ず文字フィールドと見なされます。システム/36 は、キー・フィールドをパック形式またはゾーン形式として認識しません。ソース IBM i アプリケーション・プログラムがシステム/36 ファイル内のパック 10 進数フィールドを参照すると、予期しない結果になる可能性があります。パック数値の保管の仕方が原因となって、また、システム/36 は相対キー順操作においてキー・フィールドをパック形式として認識しないため、所定の順序でレコードが返ってこなかったり、返ってくるはずのレコードが見つからないことがあります。

例えば、ILE RPG SETLL ステートメントは、システム/36 ファイルに対してパック 10 進数フィールドを使用している場合、レコードが見つからないために予期せずに失敗したり、予想されたレコードとは異なるレコードを選択したりする可能性があります。文字フィールドと符号なし数字フィールドを、キー・フィールドとして使用する必要があります。

IBM iとシステム/38 の DDM の相違点:

このトピックでは、IBM i とシステム/38 の間の DDM の相違点をリストします。

- DDM ファイルは、システム/38 環境、または IBM i オペレーティング・システムのいずれかで作成することができます。
- システム/38 環境でも、リモート・コマンド投入 (SBMRMTCMD) コマンドは、サーバー・システムの構文になっている必要があります。例えば、システム/38 製品が IBM i オペレーティング・システムにコマンドを投入する場合、IBM i の構文を使用する必要があります。
- リモート・ファイル名は、サーバー・システムの構文になっていなければなりません。
- システム/38 でのネットワーク属性変更 (CHGNETA) コマンドの DDMACC パラメーターのデフォルト値は、*REJECT です。IBM i オペレーティング・システムにおける DDMACC パラメーターのデフォルト値は、*OBJAUT です。
- システム/38 プラットフォームの場合、ファイルは FIFO (先入れ先出し) または LIFO (後入れ先出し) として作成されます。システム/38 でファイルを作成するときのデフォルトは、FIFO です。

IBM i オペレーティング・システムに対して、重複キーが FIFO または LIFO のいずれかに依存するシステム/38 アプリケーション・プログラムを実行する場合、IBM i ファイルを作成するのであれば、IBM i ファイルにはデフォルトがないため、FIFO または LIFO のいずれかを指定する必要があります。なぜなら、IBM i オペレーティング・システムでは、FIFO または LIFO のいずれかであるはずの共用索引パスが探し出されて使用されるからです。

- リモートのシステム/38 で DDM を使用して作成された文字以外の (ゾーン 10 進数 またはパック 10 進数) フィールドの入ったキー付きファイルの場合、文字フィールドとして定義されたフィールドになることがあります。このような場合、相対キー順操作を使用してこのようなファイルを処理すると、予期しない結果を生じることがあります。このファイルはパック 10 進数形式ではないフィールドを使用して作成されるため、所定の順序でレコードが返ってこなかったり、返ってくるはずのレコードが見つからないことがあります。

例えば、ILE RPG SETLL ステートメントは、システム/38 ファイルに対してパック 10 進数フィールドを使用している場合、レコードが見つからないために予期せずに失敗したり、予想されたレコードとは異なるレコードを選択したりする可能性があります。リモートのシステム/38 で DDM を使用してファイルを作成する場合、そのファイルのキー・フィールドとして使用できるのは、文字フィールドと符号なし数字フィールドだけです。

- 相対レコード番号操作によってレコードの追加をサポートするには、ILE RPG プログラムで DDM を使用して READ CHAIN(RRN) 命令および WRITE 命令を実行する必要があります。ファイルは読み取りと更新の権限の下にオープンしなければならないが、ユーザーは、そのファイルのデータの読み取りと更新を行えるユーザーでなければなりません。

相対レコード番号操作を使用してレコードを追加するための様式選択プログラムは、IBM i オペレーティング・システムでのみサポートされます。論理ファイル用の様式選択プログラムを備えたユーザーが、直接ファイル処理を行うと、互換性の問題が生じることがあります。

DRDA と DDM の管理

分散リレーショナル・データベースの管理担当者は、複数のシステム上で行われる作業に責任を負うこととなります。

ローカル・システムを起点とするクライアントとしての作業の監視は、他の作業の監視を IBM i オペレーティング・システム上で行うのと同じ要領で行うことができます。

ローカル・システム上で行われるサーバーとしての作業単位を追跡する際には、同じツールを使用しますが、異なる種類の情報を探すこととなります。

このトピックでは、ネットワークの中で行われる分散リレーショナル・データベース作業を管理できる方法について説明しています。ここで説明するほとんどのコマンド、プロセス、およびその他のリソースは、分散リレーショナル・データベースを使用するためにだけ存在しているわけではありません。これらのリソースは、すべての IBM i 操作のために提供されているツールです。ここで説明している管理コマンド、プロセス、およびリソースは、いずれもすべての DB2 for i 機能とともに IBM i ライセンス・プログラムに含まれています。IBM i の実行管理機能を使えば、複数のシステム上の作業を追跡する効果的な手段が得られます。

関連タスク:

74 ページの『DDM ファイルのセットアップ』

分散リレーショナル・データベース体系 (DRDA) サポートの IBM i 実装では、通信のために分散データ管理機能 (DDM) 会話を使用します。したがって、分散リレーショナル・データベース処理に関連して DDM を使用することができます。

リレーショナル・データベースの活動のモニター

次に挙げる制御言語 (CL) コマンド (すべてのコマンドがさまざまな方法で同じ情報を提供する) を使用すれば、IBM i オペレーティング・システムでの作業の概要を把握することができます。

分散リレーショナル・データベースにおけるジョブ処理

ジョブ処理 (WRKJOB) コマンドによって、「ジョブ処理」メニューが表示されます。このメニューを使用すれば、オプションを選択して、指定されたジョブに関連する情報を処理または変更することができます。現在使用しているジョブについての情報を入手する場合には、パラメーターを指定しないでコマンドを入力します。

あるジョブに関連する同じ情報を得たい場合には、次のようにコマンドにジョブの名前を入力することによって、ジョブを指定します。

WRKJOB JOB(ジョブ番号/ユーザー ID/ ジョブ名)

ジョブがジョブ待ち行列に入っているのか、出力待ち行列に入っているのか、または活動状態にあるのかについて、メニュー画面のオプションによって提供される情報を入手することができます。ただし、ジョブがシステム上にあると見なされるのは、その入力データがすべて完全に読み込まれてからになります。その時になって初めてジョブ待ち行列に項目が入ります。ジョブ情報のオプションには、次のものがあります。

- ジョブ状況属性
- ジョブ定義属性
- スプール・ファイル情報

以下に挙げるオプションに関する情報が表示されるのは、ジョブが活動状態にある時だけです。

- ジョブ実行属性
- ジョブ・ログ情報
- 呼び出しスタック情報
- ジョブ・ロック情報
- ライブラリー・リスト情報
- オープン・ファイル情報
- ファイル指定変更情報
- コミットメント制御状況
- 通信状況
- 活動化グループ
- 相互除外 (Mutex)

オプション 10 (ジョブ・ログの表示) では、活動ジョブまたはジョブ待ち行列に入っているジョブについての情報が得られます。終了したジョブについては、通常、オプション 4 (スプール・ファイルの処理) を使用することによって、同じ情報を見つけることができます。このオプションを選択すると、「**ジョブ・スプール・ファイルの処理**」画面が表示され、そこでオプション 5 を使用して、QPJOBLOG という名前のファイル (リストに載っている場合) を表示できます。 **ジョブ処理 (WRKJOB)** コマンドによって、「ジョブ処理」メニューが表示されます。

関連資料:

ジョブ処理 (WRKJOB) コマンド

400 ページの『ループ、待機、またはパフォーマンスの問題の解決』

要求が完了するのに想定された時間より長くかかっている場合、まずアプリケーション・リクエスター (AR) を検査します。

分散リレーショナル・データベースにおけるユーザー・ジョブの処理

ジョブにより使用されているユーザー・プロファイル (ユーザー名) を知る場合、**ユーザー・ジョブの処理 (WRKUSRJOB)** コマンドを使用して、ジョブ情報を表示または変更できます。自分のユーザー・プロファイルがあるシステムのジョブのリストを入手する場合には、パラメーターを指定しないでコマンドを入力してください。

次のようにコマンドに名前を入力することによって、ユーザーおよびジョブ状況を指定して、ジョブのリストを短縮することができます。

「ユーザー・ジョブの処理」画面が表示されて、システムで実行されている (*ACTIVE) か、ジョブ待ち行列に入っている (*JOBQ) か、あるいは出力待ち行列に入っている (*OUTQ) ユーザー・ジョブの名前および状況情報が表示されます。下の画面には、KCDBA という名前のユーザーの活動ジョブおよび終了ジョブが表示されています。

```

ユーザー・ジョブの処理                KC105
03/29/92 16:15:33
オプションを入力して、実行キーを押してください。
2= 変更  3= 保留  4= 終了  5= 処理  6= 解放  7= メッセージの表示
8= スプール・ファイルの処理  13= 切断

OPT  ジョブ      ユーザー   タイプ   ----- 状況 ----- 機能
---  KC000        KCDBA    CMNEVK  OUTQ
---  KC000        KCDBA    CMNEVK  OUTQ
---  KC000        KCDBA    CMNEVK  OUTQ
---  KC000        KCDBA    CMNEVK  OUTQ
---  KC000        KCDBA    CMNEVK  ACTIVE
---  KC0001       KCDBA    CMNEVK  ACTIVE      * -PASSTHRU
---  KC0001       KCDBA    INTER   ACTIVE      CMD-WRKUSRJOB

```

終わり

パラメーターまたはコマンド

====>

F3= 終了 F4= プロンプト F5= 最新表示 F9= コマンドの複写 F11= スケジュール・データの表示
F12= 取り消し F21= 援助レベルの選択

この画面には、システム上にあるこのユーザーのジョブがすべてリストされ、指定された状況 (この場合は *ALL) が示され、ジョブのタイプが表示されています。また、選択したジョブについてコマンドを入力するための 8 つのオプション (2 から 8 および 13) も提供されています。オプション 5 を選択すると、前述した「ジョブの処理」画面が表示されます。

ユーザー・ジョブの処理 (WRKUSRJOB) コマンドは、システムが TCP/IP を使用している場合に DDM TCP/IP サーバー・ジョブの状況を確認する場合に役立ちます。次のコマンドを実行してください。

```
WRKUSRJOB QUSER *ACTIVE
```

文字 QRWT で始まるジョブが表示されるまでページ送りしてください。システムが活動状態にある場合、(事前開始 DRDA ジョブがシステム上で実行されていないかぎり) QRWTLSTN という名前のジョブが 1 つと、QRWTSRVR という名前の 1 つ以上のジョブが表示されるはずですが、QRWTSRVR ジョブは、事前開始ジョブです。QRWTLSTN ジョブが表示されていない場合には、次のコマンドを実行して、ジョブを開始してください。

```
STRTCPSVR *DDM
```

QRWTLSTN ジョブが表示されていて、QRWTSRVR ジョブが表示されておらず、DRDA 事前開始ジョブが使用不可になっていない場合には、次のコマンドを実行し、事前開始ジョブを開始してください。

```
STRPJ subsystem QRWTSRVR
```

V5R2 より前のバージョンでは、通常 QRWTSRVR は QSYSWRK サブシステムで実行されていました。V5R1 より後のバージョンでは、QRWTSRVR は QUSRWRK で実行されます。

関連資料:

ユーザー・ジョブ処理 (WRKUSRJOB) コマンド

400 ページの『ループ、待機、またはパフォーマンスの問題の解決』
要求が完了するのに想定された時間より長くかかっている場合、まずアプリケーション・リクエスター (AR) を検査します。

分散リレーショナル・データベースにおける活動ジョブの処理

数人のユーザーに対して実行されているジョブをモニターする場合や、ジョブを探しているものの、ジョブの名前やユーザー ID がわからない場合は、**WRKACTJOB (活動ジョブの処理)** コマンドを使用してください。

このコマンドを入力すると、「活動ジョブの処理」画面が表示されます。この画面には、現在システム上で活動状態にあるジョブのパフォーマンスおよび状況の情報が表示されます。情報は、すべてある 1 つのジョブを基準に収集され、サブシステム別にグループ化されています。

次の画面には、KC105 システムでの標準的な 1 日の「活動ジョブの処理」画面が示されています。

活動ジョブの処理 KC105

10/16/09 10:55:40

CPU %: 41.7 経過時間 : 04:37:55 活動ジョブ数 : 42

オプションを入力して、実行キーを押してください。

2= 変更 3= 保留 4= 終了 5= 処理 6= 解放 7= メッセージの表示
8= スプール・ファイルの処理 13= 切断 ...

OPT	サブシステム/ ジョブ	ユーザー	タイプ	CPU %	機能	状況
—	QBATCH	QSYS	SBS	.0		DEQW
—	QCMN	QSYS	SBS	.0		DEQW
—	QINTER	QSYS	SBS	.0		DEQW
—	DSP01	CLERK1	INT	.0	CMD-STRSQL	DSPW
—	DSP02	CLERK2	INT	.0	* -CMDENT	DSPW

続く...

パラメーターまたはコマンド
===>

F3= 終了 F5= 最新表示 F7= 検索 F10= 統計の再始動
F11= 経過データの表示 F12= 取り消し F23= オプション続き F24= キーの続き

F11 (経過データの表示) を押すと、次の画面が表示されて、詳細な状況情報が得られます。

活動ジョブの処理 KC105
10/16/09 10:55:40
CPU %: 41.7 経過時間 : 04:37:55 活動ジョブ数 : 42

オプションを入力して、実行キーを押してください。

2= 変更 3= 保留 4= 終了 5= 処理 6= 解放 7= メッセージの表示
8= スプール・ファイルの処理 13= 切断 ...

OPT	サブシステム/ ジョブ	タイプ	プール	PTY	CPU	経過			
						INT	RSP	AUXIO	CPU %
—	QBATCH	SBS	2	0	4.4		108		.0
—	QCMN	SBS	2	0	20.7		668		.0
—	KC000	EVK	2	50	.1		9		.0
—	KC0001	EVK	2	50	.1		9		.0
—	MP000	EVK	2	50	.1		14		.0
—	QINTER	SBS	2	0	7.3		4		.0
—	DSP01	INT	2	20	.1		0		.0
—	DSP02	INT	2	20	.1		0		.0

続く...

パラメーターまたはコマンド

====>

F3= 終了 F5= 最新表示 F7= 検索 F10= 統計の再始動
F11= 経過データの表示 F12= 取り消し F23= オプション続き F24= キーの続き

「活動ジョブの処理」画面では、ジョブ優先順位およびシステム使用状況、「ユーザー・ジョブの処理」画面から得られるユーザーおよびタイプの情報が得られます。また、ジョブに対して 11 のオプション (2 から 11 および 13) が使用でき、この中にはオプション 5 が含まれていて、これを選択した場合には、選択したジョブに関する「ジョブの処理」画面が表示されます。数人のユーザーに対して実行されているジョブをモニターする場合や、ジョブを探しているものの、ジョブの名前やユーザー ID がわからない場合は、**WRKACTJOB (活動ジョブの処理)** コマンドを使用してください。

System i ナビゲーター を使用しても、ジョブ優先順位とシステム使用状況に関する情報を表示できます。これを行うためには、以下のステップに従ってください。

1. System i ナビゲーター インターフェース中の「データベース」を選択します。
2. 情報を表示したいリモート・データベースを選択します。
3. 「プロパティ」を右クリックして選択します。プロパティ・ウィンドウが開いて情報が表示されます。

関連資料:

活動ジョブの処理 (WRKACTJOB) コマンド

335 ページの『システムによる分散リレーショナル・データベースのパフォーマンスの向上』
効率的なシステム・パフォーマンスを達成するには、システム・リソース間の適正なバランスが必要です。過剰に使用しているリソースがあれば、パフォーマンスに悪影響があります。このトピックでは、システム・パフォーマンスの監視に役立つコマンドについて説明します。

400 ページの『ループ、待機、またはパフォーマンスの問題の解決』
要求が完了するのに想定された時間より長くかかっている場合、まずアプリケーション・リクエスター (AR) を検査します。

分散リレーショナル・データベースにおけるコミットメント定義の処理

システム上でコミットメント定義を処理する場合は、**WRKCMTDFN (コミットメント定義の処理)** コマンドを使用してください。

コミットメント定義を使用すると、**コミットメント制御開始 (STRCMTCTL)** コマンドでコミットメント制御が開始された場合に、そのコミットメント制御の情報を記憶できます。これらのコミットメント定義は、活動ジョブに関連している場合もしていない場合もあります。活動ジョブに関連していないコミットメント定義は終了していますが、その論理作業単位の 1 つまたはいくつかはまだ完了していません。

WRKCMTDFN (コミットメント定義の処理) コマンドを使用すると、コミットメント定義のジョブ名、状況または論理作業単位識別コードに基づいてコミットメント定義を処理することができます。

STATUS パラメーターには、すべてのジョブを指定するか、あるいは、*RESYNC または *UNDECIDED の状況値をもつジョブだけを指定することができます。*RESYNC は、同期点を再設定する過程においてそのリソースの再同期化に関係するジョブだけを示します。同期点 は、すべてのリソースが整合状態にある地点です。

*UNDECIDED は、リソースのコミットまたはロールバックについての決定が未知であるジョブだけを示します。

LUID パラメーターでは、他のシステムのコミットメント定義を処理しているコミットメント定義を表示することができます。このようなコミットメント定義を含んでいるジョブは、APPC 保護会話を使用して通信します。あるシステムのコミットメント定義を表示し、それを **WRKCMTDFN (コミットメント定義の処理)** コマンドへの入力として使用すると、LUID を見つけて、対応するコミットメント定義を発見することができます。

コミットメント定義の処理 (WRKCMTDFN) コマンドを使用して未決定のジョブ内のローカル・リソースを解放することができます。ただし、これができるのは、コミットメント定義が「Prepared (準備済み) (PRP)」状態または「最終エージェント保留中 (LAP)」状態である場合のみです。コミットメント定義は、強制的にコミットまたはロールバックすることができ、したがって、保持リソースを解放できます。開始プログラムがコミットメント定義でとられた処置を知るまでは、制御はもとのコミットメントを出したプログラムには戻りません。

また、再同期化が他のシステムで終わることができないことが分かった場合に、同期化を終わりにするために **WRKCMTDFN (コミットメント定義の処理)** コマンドを使用することができます。

関連概念:

トランザクションおよびコミットメント制御のトラブルシューティング

関連資料:

コミットメント制御開始 (STRCMTCTL) コマンド

コミットメント定義の処理 (WRKCMTDFN) コマンド

分散リレーショナル・データベースのジョブ・ログによる要求情報の追跡

すべての IBM i ジョブは、それぞれジョブについて入力された要求に関連する情報が入るジョブ・ログを持っています。

ジョブ・ログの中の情報には、次のものがあります。

- ジョブで使用されたコマンド。
- 送られたが、プログラム・メッセージ待ち行列から除去されなかったメッセージ。
- CL プログラムの中のコマンド (そのプログラムが LOGCLPGM(*JOB) を用いて作成され、ジョブが LOGCLPGM(*YES) を指定しているか、または CL プログラムが LOGCLPGM(*YES) を用いて作成された場合)。

ジョブの終了時に、ジョブ・ログは、QPJOBLOG という名前のスプール・ファイルを書き込むことができ、元のジョブ・ログは削除されます。ジョブ記述の LOG パラメーターを指定することによって、ジョブ・ログに書き込まれる情報を制御することができます。

ジョブ・ログを表示する方法は、ジョブの状況によって異なります。ジョブが終了していて、しかもジョブ・ログがまだ印刷されていない場合には、**ユーザー・ジョブの処理 (WRKUSRJOB)** コマンドを使用してジョブを見つけてから、オプション 8 (スプール・ファイルの表示) を選択してください。QPJOBLOG という名前のスプール・ファイルを見つけ、オプション 5 (ジョブ・ログの表示) を選択してください。**ジョブの処理 (WRKJOB)** コマンド、および「ジョブの処理」画面上の他のオプションを使用することによって、ジョブ・ログを表示することもできます。

バッチ・ジョブまたは対話式ジョブがまだ活動状態にあるか、あるいはジョブ待ち行列に入っていて、まだ開始されていない場合には、**WRKUSRJOB** コマンドを使用してジョブを見つけてください。**活動ジョブの処理 (WRKACTJOB)** コマンドを使用すると、活動ジョブのジョブ・ログは表示されますが、ジョブ待ち行列に入っているジョブは表示されません。オプション 5 (ジョブの処理) を選択してから、オプション 10 (ジョブ・ログの表示) を選択してください。

自分自身の対話式ジョブのジョブ・ログを表示する場合には、次のようにしてください。

- **ジョブ・ログの表示 (DSPJOBLOG)** コマンドを入力します。
- **ジョブの処理 (WRKJOB)** コマンドを入力し、「ジョブの処理」画面でオプション 10 (ジョブ・ログの表示) を選択してください。
- 「コマンド入力」画面で F10 (詳細メッセージの表示) を押して、ジョブ・ログに示されているメッセージを表示してください。

ジョブ・ログの表示 (DSPJOBLOG) コマンドを使用すると、「ジョブ・ログ」画面が表示されます。この画面には、プログラム名に次のような特殊記号がついて表示されます。

- >> 実行中のコマンドまたは次に実行されるコマンド。たとえば、CL プログラムまたは高水準言語プログラムが呼び出された場合には、プログラムに対する呼び出しが表示されます。
- > コマンドは処理を完了しました。
- .. コマンドはまだ処理されていません。
- ? 応答メッセージ。この記号によるマークが付くのは、応答を必要としているメッセージと応答がなされたメッセージの両方です。

関連タスク:

411 ページの『ジョブ・ログの印刷』

すべての IBM i ジョブは、それぞれジョブについて入力された要求に関連する情報が入るジョブ・ログを持っています。クライアントで問題が起きたら、その問題を診断するのにジョブ・ログ内の情報が役立つことがあります。

関連資料:

ジョブ・ログの表示 (DSPJOBLOG) コマンド

活動ジョブの処理 (WRKACTJOB) コマンド

ジョブ処理 (WRKJOB) コマンド

ユーザー・ジョブ処理 (WRKUSRJOB) コマンド

431 ページの『分散リレーショナル・データベースのメッセージ』

サーバーまたはクライアントでエラー・メッセージが出された場合、障害の理由を示すシステム・メッセージがジョブ・ログにログ記録されます。

分散リレーショナル・データベースのジョブの探索

クライアント上で分散リレーショナル・データベース・ジョブについての情報を探索する時に、使用されているユーザー・プロファイルが分かっている場合には、ユーザー・ジョブの処理 (WRKUSRJOB) コマンドを使用することによって、そのジョブを見つけることができます。

このコマンドは、サーバー・システム上でも使用することができますが、サーバー上のユーザー・プロファイルはクライアントで使用されているものとは異なっている場合があることを心得ていてください。

TCP/IP サーバーの場合、ジョブ名を修飾するユーザー・プロファイルは常に QUSER になり、ジョブ名は常に QRWTSRVR になります。ログの表示 (DSPLOG) コマンドは、完全なサーバー・ジョブ名を見つけやすくするために使用できます。メッセージは、次の形式になります。

```
DDM job 031233/QUSER/QRWTSRVR servicing user XY on 10/02/97 at 22:06
```

指定したユーザー・プロファイルに関して複数のジョブがリストされていて、そのリレーショナル・データベースが DRDA を使用してアクセスされている場合は、オプション 5 (ジョブの処理) を入力して、「ジョブの処理」画面を表示してください。この画面でオプション 10 (ジョブ・ログ) を入力して、ジョブ・ログを表示してください。このジョブ・ログでは、ジョブが分散リレーショナル・データベース・ジョブであるかどうかを示され、そうである場合には、そのジョブが接続されるリモート・システムが示されます。(接続が APPC または TCP/IP を使用しているかどうかによって) ジョブ・ログのページを送って次のメッセージを探してください。

CPI9150

| SNA 経由でデータベース会話が開始された。

CPI9160

| TCP/IP 経由でデータベース接続が開始された。

メッセージ CPI9150 および CPI9160 の第 2 レベル・テキストには、サーバー・ジョブのジョブ名が入っています。

サーバーで作業していて、ジョブ名は分かっていないが、ユーザー名が分かっている場合には、ユーザー・ジョブの処理 (WRKUSRJOB) コマンドを使用してください。ユーザーを指定しないでこのコマンドを使用した場合には、自分が使用しているユーザー・プロファイルのもとにあるジョブのリストが戻されます。

TCP/IP の場合、ジョブ名内のユーザー・プロファイルは常に QUSER です。「ユーザー・ジョブの処理」画面では、次の列を使用して、APPC 接続を行っているサーバー・ジョブの識別に役立ててください。

- 1 ジョブのタイプ列には、APPC 通信ジョブについて、タイプを CMNEVK としてリストしたジョブが表示されます。
- 2 状況の列には、ジョブが活動状態であるか、あるいは完了しているかが表示されます。ジョブのログを記録するためのシステムのセットアップ方法によっては、活動ジョブしか表示されない場合があります。
- 3 ジョブの列には、ジョブ名が示されます。サーバー上のジョブ名は使用されている装置と同じになります。

```

ユーザー・ジョブの処理                                KC105
03/29/92 16:15:33
オプションを入力して、実行キーを押してください。
2= 変更  3= 保留  4= 終了  5= 処理  6= 解放  7= メッセージの表示
8= スプール・ファイルの処理  13= 切断

— KC000      KCDBA      CMNEVK  OUTQ
— MP000      KCDBA      CMNEVK  OUTQ
— MP000      KCDBA      CMNEVK  OUTQ
— KC000      KCDBA      CMNEVK  OUTQ
— KC000      KCDBA      CMNEVK  ACTIVE
— KC0001     KCDBA      INTER   ACTIVE          CMD-WRKUSRJOB
  3                1      2

```

活動サーバー・ジョブを探索する時に、ユーザー名が分かっていない場合には、**WRKACTJOB** (活動ジョブの処理) コマンドを入力すると、システム上で活動状態にあるサブシステムのジョブのリストが表示されます。以下の例は、探索する項目を示しています。

```

活動ジョブの処理                                KC105
03/29/92 16:17:45
CPU %:  41.7   経過時間 :  04:37   活動ジョブ数 :  102

オプションを入力して、実行キーを押してください。
2= 変更  3= 保留  4= 終了  5= 処理  6= 解放  7= メッセージの表示
8= スプール・ファイルの処理  13= 切断

OPT  サブシステム/ ユーザー   タイプ   CPU % 機能          状況
      ジョブ
—  QBATCH      QSYS      SBS      .0
4  QCMN        QSYS      SBS      .0          WDEQ
—  KC0001     KCCLERK   EVK      .0 *        EVTW
  5                6

```

- 4 サーバー・ジョブを処理するためにセットアップされているサブシステムを探してください。この例では、サーバー・ジョブのサブシステムは **QCMN** です。TCP/IP サーバー・ジョブ用のサブシステムは、**V5R2** より前のバージョンでは **QSYSWRK**、**V5R1** より後のバージョンでは **QUSRWRK** です。
- 5 **APPC** サーバー・ジョブでは、ジョブ名はサーバー・ジョブ用に作成された装置の装置名です。
- 6 リストされているジョブ・タイプは、通常、プログラム開始要求によって開始された **EVK** です。TCP/IP サーバー・ジョブの場合、(ジョブ・タイプが **BCI** の場合に **DRDA** 事前開始ジョブがシステム上で活動状態でないかぎり) ジョブ・タイプは **PJ** です。

探索しているジョブと思われるものが見つかったら、オプション **5** を入力して、そのジョブを処理してください。次に、「ジョブ・メニューの処理」画面でオプション **10** を選択して、ジョブ・ログを表示してください。DB2 for i クライアントからサーバーにアクセスしているジョブの分散データベース・ジョブ・ログには、最上部の近くに次のようなステートメントが入っています。

CPI3E01

ローカル・リレーショナル・データベースが (システム名) によってアクセスされた。

サーバーで実行中のジョブが見つかったら、クライアントが **IBM i** 製品である場合には、そのジョブをトレースしてクライアントまでさかのぼることもできます。ジョブ・ログ内に以下のいずれかのメッセージが現れます。受け取ったメッセージ上にカーソルが置かれます。

CPI9152

ターゲット **DDM** ジョブがクライアント・システムによって開始された。

CPI9162

DDM 接続を処理するために割り当てられたターゲット・ジョブは、TCP/IP 経由でクライアントによって開始されました。

Help キーを押すと、このステートメントの詳細なメッセージが表示されます。クライアントのジョブ名は、このジョブの原因となったクライアント上のジョブです。

関連タスク:

411 ページの『ジョブ・ログの印刷』

すべての IBM i ジョブは、それぞれジョブについて入力された要求に関連する情報が入るジョブ・ログを持っています。クライアントで問題が起きたら、その問題を診断するのにジョブ・ログ内の情報が役立つことがあります。

関連資料:

ログ表示 (DSPLOG) コマンド

活動ジョブの処理 (WRKACTJOB) コマンド

ユーザー・ジョブ処理 (WRKUSRJOB) コマンド

400 ページの『ループ、待機、またはパフォーマンスの問題の解決』

要求が完了するのに想定された時間より長くかかっている場合、まずアプリケーション・リクエスター (AR) を検査します。

リモート・システムの操作

分散リレーショナル・データベースでは、管理担当者は、リモート IBM i 製品を操作しなければならない場合があります。

たとえば、リモート・サーバーを開始または停止しなければならない場合があります。IBM i 製品には、リモート・システムを必要に応じて操作できるようにするオプションが用意されています。確かに、リモート・システムが確実に動作するようにする最も単純な方法は、分散リレーショナル・データベース要件に応じて、リモート・ロケーションでサーバーの電源を入れられるようにすることです。しかし、このことは必ずしも常に可能であるとは限りません。自動電源オンおよび電源オフのスケジュールを用意するか、またはリモート・サーバーに対するリモート電源オンを可能にすることができます。

システムでは、リアルタイムに、または前もって予定した時刻に、このことを行う複数の方法が用意されています。さらに頻度が高いのは、リモート・システムの動作中に、そのリモート・システムで特定のタスクを実行する必要がある場合です。そのためには 3 つの主要な方法が使用できます。表示装置パススルーを使用する方法、SBMRMTCMD (リモート・コマンド投入) コマンドを用いる方法、およびストアード・プロシージャを使用する方法です。

SBMRMTCMD (リモート・コマンド投入) コマンドは、サーバー側で実行するために、分散データ管理機能 (DDM) サポートを使用する CL コマンドを投入するためのものです。まず DDM ファイルを作成する必要があります。DDM ファイルのリモート・ロケーション情報を使用して、使用する通信回線を決めます。こうして、投入されたコマンドを受信するサーバーを識別します。DDM ファイルを使用して、サーバー側で実行するためのコマンドを投入する際には、DDM ファイルに関連するリモート・ファイルは関与しません。

SBMRMTCMD (リモート・コマンド投入) コマンドは、バッチ環境でも、QCAEXEC システム・プログラムを介しても実行できる CL コマンドであれば、すべて投入することができます。つまり、コマンドは、ALLOW 属性に *BPGM および *EXEC の値が指定されています。表示コマンド (DSPCMD) コマンドを使用することによって、ALLOW 属性を表示することができます。

SBMRMTCMD (リモート・コマンド投入) コマンドの主要な目的は、クライアント側のユーザーやプログラムが、サーバー側にあるオブジェクトに対して、ファイル管理操作およびファイル許可活動を実行できるようにすることにあります。このコマンドの第 2 の目的は、サーバーで、非ファイル操作 (メッセージ待ち行列の作成など) を行ったり、実行しようとするユーザー作成のコマンドを投入したりできるようにすることにあります。CMD パラメーターを使えば、サーバーで実行したいコマンドを表すのに、最大 2000 文字の文字ストリングを指定できます。

投入される CL コマンド、およびそのコマンドの実行対象となるオブジェクトに関して、サーバー上で適正な権限が付与されていなければなりません。クライアント・ユーザーがそのための正しい権限 (サーバーのユーザー・プロファイルに決められている) を持っている場合には、以下に挙げる活動が、SBMRMTCMD (リモート・コマンド投入) コマンドを使用してリモート・ファイルに対して実行できることの例になります。

- リモート・テーブルに対するオブジェクト権限の認可または取り消し。
- テーブルまたはその他のオブジェクトの検査。
- テーブルまたはその他のオブジェクトの保管または復元。

このコマンドを使用すれば、リモート・システム上のテーブルまたはその他のオブジェクトに対して多くのことができますが、このコマンドの使用が、タスクによっては、システム上の他の方法に比べて効果的でない場合があります。たとえば、このコマンドを使用して、リモート・ファイルのファイル記述やフィールド属性を表示したり、または、ファイルやその他のオブジェクトをダンプしたりすることができますが、出力はサーバーに残ります。リモート・ファイル記述およびフィールド属性をクライアントに表示するには、SYSTEM(*RMT) を指定したファイル記述表示 (DSPFD) コマンドまたはファイル・フィールド記述表示 (DSPFFD) コマンドを使用し、リモート・ファイルに関連する DDM ファイルの名前を指定する方が、方法として優れています。

関連概念:

システムのシャットダウンおよび再始動をスケジュールする

IPL を制御するシステム値

関連タスク:

74 ページの『DDM ファイルのセットアップ』

分散リレーショナル・データベース体系 (DRDA) サポートの IBM i 実装では、通信のために分散データ管理機能 (DDM) 会話を使用します。したがって、分散リレーショナル・データベース処理に関連して DDM を使用することができます。

関連資料:

305 ページの『DDM 会話の制御』

通常、クライアント・システム・ジョブに関連した DDM 会話は、このトピックに記述された条件のいずれかを満たすまでは、活動状態に保持されます。

コマンド表示 (DSPCMD) コマンド

ファイル記述表示 (DSPFD) コマンド

ファイル・フィールド記述表示 (DSPFFD) コマンド

リモート・コマンド投入 (SBMRMTCMD) コマンド

プログラムで使用するオブジェクトの表示

プログラム参照表示 (DSPPGMREF) コマンドを使用して、プログラムまたは SQL パッケージで使用されるテーブル、データ域、および他のプログラムを判別することができます。この情報は SQL パッケージとコンパイル済みプログラムでのみ使用でき、表示、印刷、またはデータベース出力ファイルへの書き出しができます。

プログラムまたはパッケージの作成時に、そのプログラムまたはパッケージの中で使用される特定のオブジェクトについての情報が保管されます。この情報は、後で**プログラム参照表示 (DSPPGMREF)** コマンドで使うことができます。検索される情報には、次のものがあります。

- プログラムまたはパッケージの名前およびそのテキスト記述
- プログラムまたはパッケージが入っているライブラリーまたはコレクションの名前
- プログラム・パッケージによって参照されるオブジェクトの数
- システム・オブジェクトの修飾名
- 情報検索日付
- 参照されたオブジェクトのオブジェクト・タイプ

ファイルおよびテーブルの場合は、レコードに次のような追加のフィールドが入ります。

- プログラムまたはパッケージの中のファイルまたはテーブルの名前 (プログラムまたはパッケージの作成時に指定変更操作が有効であった場合には、システム・オブジェクト名とは異なることがあります)。

注: 指定変更はいずれもアプリケーション・リクエスター (AR) 側でのみ適用されます。

- ファイルまたはテーブル (入力、出力、更新、未指定、またはこれら 4 つの組み合わせ) のプログラムまたはパッケージによる使用。
- 参照されたレコード様式 (もしあれば) の数。
- ファイルまたはテーブルで使用されるレコード様式の名前、およびそのレコード様式レベル識別コード。
- 各様式ごとに参照されたフィールドの数。

プログラムの中のオブジェクトが表示されるためには、ユーザーがあらかじめそのプログラムに対して *USE 権限を持っていないければなりません。また、ライブラリー修飾子で指定されたライブラリーの中で、ユーザーが読み取り権を持っているライブラリーだけが、プログラム探索の対象となります。

次の表に、高水準言語およびユーティリティーが情報を保管するオブジェクトを示します。

表 28. 高水準言語でのオブジェクトに関する情報の保管方法

言語	ファイル	プログラム	データ域	注を参照
CL	する	する	する	1
COBOL/400 言語	する	する	しない	2
PL/I	する	する	該当せず	2
RPG/400 言語	する	しない	する	3
DB2 for i SQL	する	該当せず	該当せず	4

表 28. 高水準言語でのオブジェクトに関する情報の保管方法 (続き)

言語	ファイル	プログラム	データ域	注を参照
----	------	-------	------	------

注:

1. ファイル、プログラム、またはデータ域を参照するすべてのコマンドでは、コマンドを CL プログラム内でコンパイルする時に情報を保管することをコマンド定義に指定する必要があります。変数を使用する場合、変数の名前はオブジェクト名として使用されます (例: &FILE)。式を使用する場合、オブジェクトの名前は *EXPR として保管されます。ユーザー定義コマンドで、コマンドに指定されたファイル、プログラム、またはデータ域に関する情報を保管することもできます。PARM または ELEM コマンド・ステートメントの FILE、PGM、および DTAARA の各パラメーターの説明を参照してください。
2. プログラム名が保管されるのは、COBOL/400 識別コード (CALL PGM1 などの動的呼び出し) ではなく、リテラルがプログラム名に使用されている場合 (CALL 'PGM1' などの静的呼び出し) だけです。
3. 内部データ域の使用は保管されません。
4. SQL パッケージについての情報。

保管されたファイル情報には、使用のタイプを表す項目 (番号) が含まれています。 **プログラム参照表示 (DSPPGMREF)** コマンドのデータベース・ファイル出力 (OUTFILE パラメーターの使用時に作成される) では、この項目は以下のリストの 1 つ以上のコードを表示します。オブジェクト 1 つにつき 1 項目しか可能ではありませんが、組み合わせで使用することができます。たとえば、7 というコードのファイルであれば、入力、出力、および更新に使用されます。

コード 意味

- | | |
|---|---------|
| 1 | 入力 |
| 2 | 出力 |
| 3 | 入力および出力 |
| 4 | 更新 |
| 8 | 未指定 |

関連資料:

プログラム参照表示 (DSPPGMREF) コマンド

エレメント定義 (ELEM)

例: プログラム参照の表示

アプリケーション・リクエスト (AR) プログラムで使用されるオブジェクトを表示するには、次のようなコマンドを入力することができます。

```
DSPPGMREF PGM(SPIFFY/PARTS1) OBJTYPE(*PGM)
```

リクエスト側では、プログラムで使用されるすべてのコレクションおよびテーブルのリストを取り出すことができますが、それらが入っているリレーショナル・データベースがどれであるかを表示することはできません。複数のリレーショナル・データベースに入っている場合もあります。コマンドの出力は、データベース・ファイルまたは表示スプール・ファイルに渡すことができます。その出力は次のようになります。

ファイル :	QPDSPPGM	ページ/行	1/1
制御 :		桁	1 - 75
検索 :			

```
3/29/92          プログラム参照の表示
DSPPGMREF コマンド入力
プログラム . . . . . : PARTS1
ライブラリー . . . . . : SPIFFY
```



```

出力 . . . . . : *
SQL パッケージの組み込み . . . . . : *YES
プログラム . . . . . : PARTS1
ライブラリー . . . . . : SPIFFY
テキストの'記述' . . . . . : パーツの在庫を確認
参照されるオブジェクトの数 . . . . . : 3
オブジェクト . . . . . : PARTS1
ライブラリー . . . . . : SPIFFY
オブジェクト・タイプ . . . . . : *PGM
オブジェクト . . . . . : QSQRROUTE
ライブラリー . . . . . : *LIBL
オブジェクト・タイプ . . . . . : *PGM
オブジェクト . . . . . : INVENT
ライブラリー . . . . . : SPIFFY
オブジェクト・タイプ . . . . . : *FILE
プログラム中のファイル名 . . . . . :
ファイルの使用法 . . . . . : 入力

```

サーバー SQL パッケージで使用されるオブジェクトを表示する場合には、次のようなコマンドを入力することができます。

```
DSPPGMREF PGM(SPIFFY/PARTS1) OBJTYPE(*SQLPKG)
```

コマンドの出力は、データベース・ファイルまたは表示スプール・ファイルに渡すことができます。その出力は次のようになります。

```

ファイル . . . . . : QPDSPPGM                ページ/行  1/1
制御 . . . . .                桁          1 - 75
検索 . . . . .

```

3/29/92 プログラム参照の表示

```

DSPPGMREF コマンド入力
プログラム . . . . . : PARTS1
ライブラリー . . . . . : SPIFFY
出力 . . . . . : *
SQL パッケージの組み込み . . . . . : *YES
SQL パッケージ . . . . . : PARTS1
ライブラリー . . . . . : SPIFFY
テキストの'記述' . . . . . : パーツの在庫を確認
参照されるオブジェクトの数 . . . . . : 3
オブジェクト . . . . . : INVENT
ライブラリー . . . . . : SPIFFY
オブジェクト・タイプ . . . . . : *FILE
プログラム中のファイル名 . . . . . :
ファイルの使用法 . . . . . : 入力

```

分散リレーショナル・データベースからのコレクションの除去

ジャーナル・レシーバーが入っているコレクションを削除しようとする時、サーバー・ジョブの QSYSOPR メッセージ待ち行列に照会メッセージが送られます。この照会に対する応答があるまで、サーバーおよびクライアント・ジョブは待機します。

メッセージ待ち行列に送られるメッセージを次に示します。

CPA7025

(I C) ライブラリー 内のレシーバー (名前) は完全に保管されていない。

クライアント・ジョブが待機している時に、アプリケーションが停止したような状態になることがあります。クライアント・ジョブが予想以上に長時間待機したままである場合は、下記の項目を検討してください。

- 照会メッセージが QSYSOPR メッセージ待ち行列に送られていて、応答があるまで先へ進めない状態になっていないかどうかを確認する。
- サーバー応答リストを用いて、メッセージに対してサーバーに応答させる。

注: アプリケーションがこの無限ループのような状態になると、アプリケーションを要求するジョブは、システム上の照会メッセージに応答があるまで待機します。したがって、オブジェクトの移動 (MOV OBJ) コマンドを使用して、ジャーナル・レシーバーを別のライブラリーに移動できません。また、ジャーナル・レシーバーを別のライブラリーに保管したり復元したりすることもできません。ジャーナル・レシーバー作成 (CRTJRNRCV) コマンドを使用して新しいジャーナル・レシーバーを別のライブラリー中に作成することと、ジャーナル変更 (CHGJRN) コマンドを使用してジャーナル・レシーバーをジャーナルに付加することだけ行えます。ジャーナル変更 (CHGJRN) コマンドに JRNRCV(*GEN) パラメーターを指定して使用した際に、システムによって新しく作成されるジャーナル・レシーバーは、新しいライブラリー中に作成されます。ジャーナルを保管する際に、別のライブラリー中のレシーバーが付加されている場合は、保管したバージョンのジャーナルを復元する際には、新しいジャーナル・レシーバーが他のライブラリー中に作成されません。

サーバーのサーバー応答リストを使用してメッセージに対してサーバー応答を行うには、現在停止したような状態になっているジョブを変更するか、あるいは、システムで実行しているすべてのサーバー・ジョブのジョブ記述を変更します。しかし、まず、応答リスト項目追加 (ADDRPYLE) コマンドを使用して、サーバー応答リストにメッセージ CPA7025 についての項目を追加しなければなりません。

```
ADDRPYLE SEQNBR(...) MSGID(CPA7025) RPY(I)
```

サーバーで現在実行中のジョブのジョブ記述を変更するには、SBMRMTCMD (リモート・コマンド投入) コマンドを使用してください。下記の例では、カンザス・シティ地区にあるシステムのデータベース管理担当者が、KC105 システム (TEST/KC105TST DDM ファイルによってアドレス指定されているシステム) にあるジョブ記述を変更しています。

```
SBMRMTCMD CMD('CHGJOB JOB(KC105ASJOB) INQMSGRPY(*SYSRPLY)')
DDMFILE(TEST/KC105TST)
```

ジョブ記述変更 (CHGJOB) コマンドを使用して、該当のジョブ記述を使用するすべてのジョブがサーバー応答リストを使うようにしておけば、サーバーで上記のような状態が発生するのを永続的に防ぐことができます。下記の例は、同じサーバーで CHGJOB コマンドを入力する方法を示しています。

```
CHGJOB JOB(KC105ASJOB) INQMSGRPY(*SYSRPLY)
```

この方法を用いる場合は、注意が必要です。サーバー応答リストに CPA7025 を追加すると、そのサーバー応答リストを使用するすべてのジョブに影響が及びます。ジョブ記述の変更も、そのジョブ記述を使用するすべてのジョブに影響を及ぼします。したがって、サーバー・ジョブのそれぞれに別々のジョブ記述を作成することも考えてみてください。

関連概念:

ジャーナル管理

作業の管理

関連資料:

返信リスト項目追加 (ADDRPYLE) コマンド

ジョブ記述変更 (CHGJOB) コマンド

ジャーナル変更 (CHGJRN) コマンド

ジャーナル・レシーバー作成 (CRTJRNRCV) コマンド

オブジェクト移動 (MOV OBJ) コマンド

リモート・コマンド投入 (SBMRMTCMD) コマンド

分散リレーショナル・データベースのジョブ会計

IBM i ジョブ会計機能は、データを収集して、誰がシステムを使用し、どんなシステム・リソースを使用しているかを判別できるようにします。

標準的なジョブ会計では、システムで実行されるジョブや、使用されているリソース (処理装置、印刷装置、表示装置の使用状況など)、ならびにデータベースと通信機能の詳細が示されます。ジョブ会計は任意選択で、システムにセットアップしなければなりません。

システムにリソース会計をセットアップするには、次のステップを実行します。

1. **ジャーナル・レシーバー作成 (CRTJRNRCV)** コマンドを使用することによって、ジャーナル・レシーバーを作成してください。
2. **ジャーナル作成 (CRTJRN)** コマンドを使用することによって、**QSYS/QACGJRN** という名前のジャーナルを作成してください。**QSYS/QACGJRN** という名前を使用しなければならず、しかもこのジャーナルを作成するには、**QSYS** に項目を追加する権限を持っていなければなりません。このコマンドには、前のステップで作成したジャーナル・レシーバーの名前を指定してください。
3. **システム値処理 (WRKSYSVAL)** コマンドまたは**システム値変更 (CHGSYSVAL)** コマンドを使用して、会計レベルのシステム値 **QACGLVL** を変更してください。

システム値変更 (CHGSYSVAL) コマンドの **VALUE** パラメーターによって、いつジョブ会計ジャーナル項目が作成されるかが決まります。***NONE** の値は、システムがジョブ会計ジャーナルの中に項目を作成しないことを意味します。***JOB** の値は、システムがジョブ (**JB**) ジャーナル項目を作成することを意味します。値が ***PRINT** であれば、印刷されるファイルごとに、直接印刷 (**DP**) またはスプール印刷 (**SP**) ジャーナル項目が作成されます。

ジョブが開始されると、ジョブ記述がジョブに割り当てられます。ジョブ記述オブジェクトには、会計コード (**ACGCDE**) パラメーターの値が入りますが、これは会計コードまたはデフォルト値 ***USRPRF** です。***USRPRF** が指定された場合には、ジョブのユーザー・プロファイルの中の会計コードが使用されます。

ユーザー・プロファイル作成 (CRTUSRPRF) コマンドまたは**ユーザー・プロファイル変更 (CHGUSRPRF)** コマンドで会計コード・パラメーター **ACGCDE** を使用して、会計コードをユーザー・プロファイルに追加することができます。**ジョブ記述作成 (CRTJOBBD)** コマンドまたは**ジョブ記述変更 (CHGJOBBD)** コマンドの **ACGCDE** パラメーターに所要の会計コードを指定することによって、特定のジョブ記述の会計コードを変更することができます。

ジョブ会計ジャーナルがセットアップされると、**システム値変更 (CHGSYSVAL)** コマンドが有効になった後で次にシステムに入るジョブから始めて、ジョブ会計項目がジャーナル・レシーバーに入れられます。

ジャーナル表示 (DSPJRN) コマンドで **OUTFILE** パラメーターを使用して、処理することのできるデータベース・ファイルに会計項目を書き出すことができます。

関連概念:

作業の管理

282 ページの『TCP/IP サーバーの管理』

DRDA および **DDM TCP/IP** サーバーでは、通常は、既存のシステム構成に変更を加える必要はありません。ただし、場合によっては、自分の要求によりよく合わせるため、問題を解決するため、システム・パフォーマンスを向上させるため、あるいはシステムでジョブを監視するために、システムによるサーバー・ジョブの管理方法を変更したいことがあります。

関連資料:

49 ページの『分散リレーショナル・データベースのアカウント』
分散データの使用に関する会計および請求が可能でなければなりません。

ジョブ記述変更 (CHGJOB) コマンド

システム値変更 (CHGSYSVAL) コマンド

ユーザー・プロファイル変更 (CHGUSRPRF) コマンド

ジョブ記述作成 (CRTJOB) コマンド

ジャーナル・レシーバー作成 (CRTJRNRCV) コマンド

ジャーナル作成 (CRTJRN) コマンド

ユーザー・プロファイル作成 (CRTUSRPRF) コマンド

ジャーナル表示 (DSPJRN) コマンド

システム値処理 (WRKSYSVAL) コマンド

TCP/IP サーバーの管理

DRDA および DDM TCP/IP サーバーでは、通常は、既存のシステム構成に変更を加える必要はありません。ただし、場合によっては、自分の要求によりよく合わせるため、問題を解決するため、システム・パフォーマンスを向上させるため、あるいはシステムでジョブを監視するために、システムによるサーバー・ジョブの管理方法を変更したいことがあります。

そのような変更を行って、処理要件を満たすには、オブジェクトがシステムのどの部分に影響を及ぼし、それらのオブジェクトをどのように変更するかを知っている必要があります。このトピック・コレクションでは、TCP 上でソケットを使用して通信する DRDA および DDM サーバー・ジョブを管理する方法について説明します。ここでは、システムを実行するサブシステム、システムに影響を与えるオブジェクト、およびそれらの資源を管理する方法について説明します。

IBM i 製品を管理する方法を完全に理解するには、このトピックを読み進める前に実行管理機能のトピックを注意深く検討することをお勧めします。このトピックでは、サーバー・ジョブでの作業を行なうために理解しておく必要のあるいくつかの作業管理の概念と、それらの概念やオブジェクトがどのようにシステムに関連するかについて説明します。このトピック・コレクションでは、次に TCP/IP サーバーを管理する方法や、システムの残りの部分と適合させる方法について示します。

関連概念:

56 ページの『IBM i 実行管理機能』

IBM i オペレーティング・システム上の作業はすべて、実行管理機能を通して実行要求されます。システム上では、異なるタイプの作業を処理してシステム要件を充足するために、特別な操作環境を設計することができます。

作業の管理

関連タスク:

73 ページの『TCP/IP サーバーのセットアップ』

TCP/IP プロトコルを使用する Distributed Relational Database Architecture (DRDA) サーバーを所有する場合、DDM TCP/IP サーバーをセットアップする必要があります。

関連資料:

281 ページの『分散リレーショナル・データベースのジョブ会計』

IBM i ジョブ会計機能は、データを収集して、誰がシステムを使用し、どんなシステム・リソースを使用しているかを判別できるようにします。

ジョブ記述変更 (CHGJOB) コマンド
システム値変更 (CHGSYSVAL) コマンド
ユーザー・プロファイル変更 (CHGUSRPRF) コマンド
ジョブ記述作成 (CRTJOB) コマンド
ジャーナル・レシーバー作成 (CRTJRNRCV) コマンド
ジャーナル作成 (CRTJRN) コマンド
ユーザー・プロファイル作成 (CRTUSRPRF) コマンド
ジャーナル表示 (DSPJRN) コマンド
システム値処理 (WRKSYSVAL) コマンド

25 ページの『TCP/IP を使用した DDM ファイルの作成』

リモート・システムとの接続用の通信プロトコルとして TCP/IP を使用する DDM ファイルを作成することができます。

TCP/IP サーバーの用語

DB2 for i への DDM および DRDA TCP/IP の両方のアクセスのために同じソフトウェアが使用されます。

以下の説明の中では、*DRDA* および *DDM* サーバー の代わりに、略して *DDM* サーバー という用語を使用します。ただし、ある部分では、*TCP/IP* サーバー、*DRDA* サーバー、また文脈によりこれらの区別が必要ないときには、サーバーという呼び名が使われる場合もあります。

DDM サーバーは、2 つ以上のジョブから成り立っています。1 つは DDM リスナー と呼ばれ、接続要求を聴取して、その他のジョブへの作業をディスパッチします。その他のジョブは、最初の構成時に、初期接続が行われた後に DRDA または DDM クライアントから要求を出す事前開始ジョブです。関連したすべてのジョブのセット (リスナーおよびサーバー・ジョブ) は、ひとまとめにして *DDM* サーバー と呼ばれます。

クライアントという用語は、DRDA アプリケーション環境では *DRDA* アプリケーション・リクエスター (または AR) と相互に交換可能な語として使用されます。クライアントという用語は、DDM (分散ファイル管理機能) アプリケーション環境では、*DDM* ソース・システム と相互に交換可能な語として使用されます。

サーバー という用語は、DRDA アプリケーション環境では、*DRDA* アプリケーション・サーバー (または AS) と相互に交換可能な語として使用されます。サーバーという用語は、DDM アプリケーション環境では *DDM* 受動システム と相互に交換可能な語として使用されます。(文脈によっては、IBM i 製品 (ハードウェア) のことをサーバーともいうことに注意してください。)

TCP/IP 上での 接続の確立

- 1 TCP/IP 通信サポートを使用する IBM i DRDA アプリケーション・サーバーまたは DDM サーバーのジョブを開始するには、DRDA アプリケーション・リクエスターまたは DDM クライアント・システムを、構成されたセキュリティーに応じて、ウェルノウン・ポート番号 446、447、または 448 のいずれかに接続する必要があります。

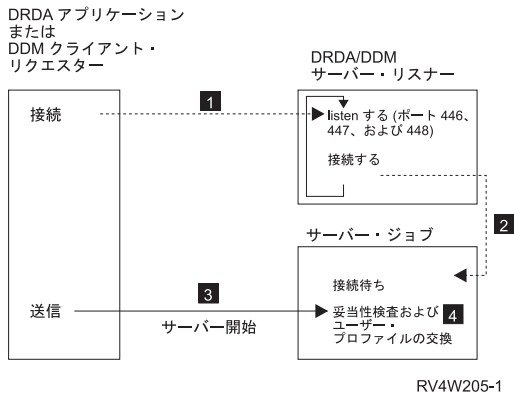


図 28. DRDA および DDM TCP/IP サーバー

1. DDM リスナー・プログラムは、クライアントの接続要求を聴取し、受け入れるために、(TCP/IP サーバーの開始 (STRTCPSVR SERVER(*DDM)) を使用することによって) 開始されていなければなりません。

DDM リスナーはこの接続要求を受け入れたとき、クライアントの接続を DDM サーバーのジョブにアタッチする内部要求を発行します 2。このサーバー・ジョブは、事前開始ジョブか、またはユーザーが QUSRSYS サブシステムあるいはユーザー定義サブシステムから QRWTSRVR 事前開始ジョブ項目を除去した場合 (事前開始ジョブが使用されていない場合)、クライアント接続要求が処理されるときに実行依頼されるバッチ・ジョブです。サーバー・ジョブは、クライアントとのそれ以降の通信を処理します。

発生する初期データ交換には、サーバー・ジョブが実行することになっているユーザー・プロフィールを識別する要求が含まれています 3。

ユーザー・プロフィールおよびパスワード (ユーザー・プロフィール ID とともに送信される場合) が妥当性検査されると、サーバー・ジョブは、CCSID など、ユーザー・プロフィール用に定義された属性を使用するようジョブを変更するだけでなく、このユーザー・プロフィールにスワップします 4。

リスナー・プログラムに接続し、サーバー・ジョブにクライアント接続を接続してデータを交換しユーザー・プロフィールとパスワードを妥当性検査する機能は、APPC プログラム開始要求が処理されるときに実行される機能に相当します。

関連資料:

TCP/IP サーバー開始 (STRTCPSVR) コマンド

リスナー・プログラム

DDM リスナー・プログラムはバッチ・ジョブで実行されます。DDM リスナーと実際のサーバー・ジョブとの間には、1 対多の関係があります。1 つのリスナーに多数の DDM サーバー・ジョブがある可能性があります。サーバー・ジョブは、通常、事前開始ジョブです。リスナー・ジョブは、QSYSWRK サブシステム内で実行されます。

DDM リスナーにより、クライアント・アプリケーションが、インバウンド接続要求を処理し経路指定することによって、関連したサーバー・ジョブとの TCP/IP 接続を確立することができるようになります。クライアントがサーバー・ジョブとの通信を確立すると、その接続の間中はクライアントとリスナーとの間の関連はなくなります。

DDM リスナーは、DRDA アプリケーション・リクエスターおよび DDM クライアント・システムが DDM TCP/IP サーバーとの接続を確立するために、活動状態でなければなりません。DDM TCP/IP 属性の変更 (CHGDDMTCPA) コマンドを使用するか、または System i ナビゲーターを使用することによって、

DRDA リスナーが自動的に開始されることを要求することができます。 System i ナビゲーターナビゲーターから DDM 設定値まで: ネットワーク > サーバー > TCP/IP。これにより、TCP/IP が開始されるときにリスナーも開始されることとなります。 DRDA リスナーを開始するとき、QSYSWRK サブシステムと TCP/IP の両方が活動状態でなければなりません。

関連資料:

DDM TCP/IP 属性変更 (CHGDDMTCPA) コマンド

TCP/IP サーバーの開始 (STRTCPSVR) CL コマンド

TCP/IP サーバーの開始 (STRTCPSVR) コマンドは、*DDM または *ALL という SERVER パラメーター値を使って、リスナーを開始するために使用されます。

関連資料:

TCP/IP サーバー開始 (STRTCPSVR) コマンド

リスナーの制約事項:

一度に活動状態にできる DDM リスナーは 1 つだけです。すでに活動状態である場合にリスナーの開始要求を行うと、コマンド発行者に通知メッセージが出されます。

注: DDM サーバーは、QUSER パスワードの有効期限が切れている場合には開始しません。 QUSER プロファイルでは、パスワードの有効期限の間隔を *NOMAX に設定することをお勧めします。この値を指定すると、パスワードの有効期限はなくなります。

例: TCP/IP サーバーの開始:

このトピックでは、TCP/IP サーバーの開始 (STRTCPSVR) CL コマンドの例を含んでいます。

例: すべての TCP/IP サーバーの開始

```
STRTCPSVR SERVER(*ALL)
```

このコマンドは、DDM サーバーを含むすべての TCP/IP サーバーを開始します。

例: DDM TCP/IP サーバーのみの開始

```
STRTCPSVR *DDM
```

このコマンドは、DDM TCP/IP サーバーのみを開始します。

TCP/IP サーバーの終了 (ENDTCPSVR) CL コマンド

TCP/IP サーバーの終了 (ENDTCPSVR) コマンドは、DDM サーバーを終了します。

DDM リスナーが終了され、クライアント・アプリケーションへの活動状態の接続を持つ関連したサーバーがある場合、サーバー・ジョブは、クライアント・アプリケーションとの通信が終了するまで活動状態のままです。ただし、リスナーが再び開始されるまで、クライアント・アプリケーションからの連続接続要求は失敗します。

関連資料:

TCP/IP サーバーの終了 (ENDTCPSVR) コマンド

TCP/IP サーバーの制限の終了:

DDM リスナーが活動状態でないときに、DDM リスナーを終了するために **TCP/IP サーバーの終了 (ENDTCPSVR)** コマンドを使用すると、診断メッセージが発行されます。(ENDTCPSVR) SERVER(*ALL) コマンドが発行されるときにリスナーが活動状態でない場合には、これと同じ診断メッセージは送信されません。

関連資料:

TCP/IP サーバーの終了 (ENDTCPSVR) コマンド

例: TCP/IP サーバーの終了:

このトピックでは、TCP/IP サーバーの終了の例を示します。

例: すべての TCP/IP サーバーの終了

```
ENDTCPSVR *ALL
```

このコマンドは、すべての活動状態の TCP/IP サーバーを終了します。

例: DDM サーバーのみの終了

```
ENDTCPSVR SERVER(*DDM)
```

このコマンドは、DDM サーバーを終了します。

System i ナビゲーター のリスナーの開始

IBM i Access Family ライセンス・プログラムの一部である System i ナビゲーターを使用して DDM リスナーを管理することもできます。

そのためには、次のパスに従います。「ネットワーク」 > 「サーバー」 > 「TCP/IP」ディレクトリー。

サーバー・ジョブ

これらのトピックでは、DRDA および DDM サーバー・ジョブを説明します。

サブシステム記述と DDM による事前開始ジョブ項目:

サブシステム記述は、どのように、どこに、どのくらいの作業をサブシステムに入れるか、そしてサブシステムが作業を実行するためにどのリソースを使用するかを定義します。以下では、QUSRWRK (OS/400 V5R2 より前のバージョンでは QSYSWRK) サブシステム内の事前開始ジョブ項目が DDM サーバーに及ぼす影響について説明します。

事前開始ジョブは、アプリケーション・リクエスター (AR) がサーバーとの通信を開始する前に実行を開始するバッチ・ジョブです。事前開始ジョブは、サブシステム記述内の事前開始ジョブ項目を使用して、ジョブを開始するときにどのプログラム、クラス、および記憶域プールを使用するかを判別します。事前開始ジョブ項目内に、サブシステムが事前開始ジョブのプールを作成および管理するために使用する属性を指定しなければなりません。

事前開始ジョブは、サーバーへの接続を開始するときのパフォーマンスを向上させます。事前開始ジョブ項目は、サブシステム内で定義されます。事前開始ジョブは、サブシステムが開始されるときに活動状態になるか、または事前開始ジョブの開始 (STRPJ) および事前開始ジョブの終了 (ENDPJ) コマンドで制御できます。

関連資料:

事前開始ジョブ終了 (ENDPJ) コマンド

事前開始ジョブ開始 (STRPJ) コマンド

事前開始ジョブ:

事前開始ジョブ (活動事前開始ジョブの表示 (DSPACTPJ) コマンドなど) に関する情報は、その情報が TCP/IP 接続要求の結果として開始された事前開始ジョブに関係がある場合でさえ、事前開始ジョブを開始するために行われる要求を示すためだけに、「プログラム開始要求」という用語を使用します。

以下のリストには、DDM TCP/IP サーバー用の最初に構成された値を持つ事前開始ジョブ項目属性が含まれています。これらは、事前開始ジョブ項目変更 (CHGPJE) コマンドを使用して変更できます。

- サブシステム記述。事前開始ジョブ項目を含むサブシステムは、V5R2 では QUSRWRK です。それ以前のリリースでは、QSYSWRK です。
- プログラム・ライブラリーおよび名前。事前開始ジョブが開始される時に呼び出されるプログラムは、QSYS/QRWTSRVR です。
- ユーザー・プロファイル。ジョブが実行するユーザー・プロファイルは、QUSER です。これは、ジョブがユーザー・プロファイルとして示すものです。サーバーへの接続要求がクライアントから受信されると、事前開始ジョブ機能は、その要求で受信されるユーザー・プロファイルにスワップします。
- ジョブ名。事前開始ジョブが開始される時のジョブ名は、QRWTSRVR です。
- ジョブ記述。事前開始ジョブに使用されるジョブ記述は、*USRPRF です。ユーザー・プロファイルは QUSER なので、QUSER のジョブ記述が何であっても *USRPRF になることに注意してください。ただし、ジョブの属性は、ユーザー ID およびパスワード (存在すれば) が検証された後に、要求しているユーザーのジョブ記述に対応するように変更されます。
- ジョブ開始。これは、サブシステムの開始時に事前開始ジョブが自動的に開始するかどうかを示します。これらの事前開始ジョブ項目には、*YES という開始ジョブ値が出荷時に設定されています。システム IPL の実行時に不必要なジョブが開始しないように、これらの値を *NO に変更してもかまいません。

注: DDM サーバー・ジョブが実行中でなく、DDM リスナー・ジョブがバッチの場合は、即時 DDM サーバー・ジョブは依然として QSYSWRK サブシステムの下で実行されます。

- 初期ジョブ数。最初に構成される時、サブシステムの開始時に開始されるジョブの数は 1 です。この値は、特定の環境および必要性に合わせて調整できます。
- 限界値。事前開始ジョブ項目の使用可能な事前開始ジョブの最小数は 1 に設定されます。この限界値に達すると、追加の事前開始ジョブが自動的に開始されます。これは、プール内に、ある特定のジョブ数を保持するために使用されます。
- 追加のジョブ数。限界値に達するときに開始される追加の事前開始ジョブ数は、最初 2 に構成されません。
- ジョブの最大数。この項目のために活動状態になることのできる事前開始ジョブの最大数は *NOMAX です。
- 最大使用数。ジョブの最大使用数は 200 に設定されます。この値は、サーバーを開始するための要求が 200 処理された後に事前開始ジョブが終了することを示しています。ある特定の状況では、TCP/IP サーバーが正しく機能するために MAXUSE パラメーターを 1 に設定する必要があります。サーバーがある特定の ILE ストアード・プロシージャを実行するとき、破棄されたオブジェクトへのポインターが事前開始ジョブ環境内に残っていることがあり、事前開始ジョブを続けて使用すると、MCH3402 例外が発生することがあります。IBM i ではこの可能性は最小になります。

- ジョブの待機。 *YES を設定すると、ジョブの最大数に達した場合に、クライアント接続要求は使用可能なサーバー・ジョブを待機します。
- プール識別コード。この事前開始ジョブが実行するサブシステム・プール識別コードは 1 に設定されます。
- クラス。事前開始ジョブが実行するクラスの名前およびライブラリーは QSYS/QSYSCLS20 に設定されます。

事前開始ジョブ項目の開始ジョブ値が *YES に設定されており、残りの値が初期設定値で提供されるような場合、それぞれの事前開始ジョブ項目ごとに以下のことが起こります。

- サブシステムが開始される時、1 つの事前開始ジョブが開始されます。
- 最初のクライアント接続要求が TCP/IP サーバーに対して処理される時、最初のジョブが使用され、限界値を超えます。
- 事前開始ジョブ項目内に定義されている数に基づいて、サーバーに対して追加のジョブが開始されます。
- 使用可能なジョブの数は 1 未満にはなりません。
- サブシステムは、プール内の未使用の事前開始ジョブの数を定期的に調べ、余分なジョブを終了します。サブシステムは、常に、少なくとも初期ジョブ・パラメーター内に指定されている事前開始ジョブの数を残します。

関連タスク:

290 ページの『サーバー・ジョブ・サブシステムの構成』

デフォルトでは DDM TCP/IP サーバー・ジョブは QUSRWRK サブシステム内で実行されます。System i ナビゲーターを使用して、クライアントの IP アドレスを基にした代替サブシステム中ですべてまたは特定のサーバー・ジョブを実行するように、DDM サーバー・ジョブを構成できます。

関連資料:

事前開始ジョブ項目変更 (CHGPJE) コマンド

アクティブな事前開始ジョブ表示 (DSPACTPJ) コマンド

事前開始ジョブのモニター:

事前開始ジョブは、**活動事前開始ジョブの表示 (DSPACTPJ)** コマンドを使用してモニターすることができます。

DSPACTPJ コマンドは次の情報を提供します。

- 事前開始ジョブの現行数
- 事前開始ジョブの平均数
- 事前開始ジョブの最大数
- 使用している事前開始ジョブの現行数
- 使用している事前開始ジョブの平均数
- 使用している事前開始ジョブの最大数
- 接続要求を待っている事前開始ジョブの現行数
- 接続要求を待っている事前開始ジョブの平均数
- 接続要求を待っている事前開始ジョブの最高数
- 平均待ち時間
- 受け入れられた接続要求の数

- 拒否された接続要求の数

関連資料:

アクティブな事前開始ジョブ表示 (DSPACTPJ) コマンド

事前開始ジョブの管理:

活動状態の事前開始ジョブについて表示されている情報は、「活動中の事前開始ジョブの表示」画面に在る間に F5 キーを押して最新表示することができます。

特に重要なのは、プログラム開始要求に関する情報です。この情報は、使用可能な事前開始ジョブの数を変更する必要があるかどうかを示します。プログラム開始要求が、使用可能な事前開始ジョブを待っていることを示す情報がある場合、**事前開始ジョブ項目変更 (CHGPJE)** コマンドを使用して事前開始ジョブを変更することができます。

プログラム開始要求が十分な速さで実行されていなかった場合、以下のいずれかの組み合わせを行うことができます。

- 限界値を増やします。
- 最初のジョブ数 (INLJOBS) パラメーター値を増やします。
- 追加のジョブ数 (ADLJOBS) パラメーター値を増やします。

肝要な点は、サーバー・ジョブを開始する要求が送信されるごとに、使用可能な事前開始ジョブが確実にあるようにすることです。

関連資料:

事前開始ジョブ項目変更 (CHGPJE) コマンド

事前開始ジョブ項目の除去:

事前開始ジョブ機能を使用しない場合は、サブシステム記述から事前開始ジョブ項目を除去してください。

1. **事前開始ジョブ終了 (ENDPJ)** コマンドを使用して、事前開始ジョブを終了します。

事前開始ジョブ項目内に開始ジョブ *YES が指定されている場合、**ENDPJ** コマンドを使用して終了された事前開始ジョブは、次回サブシステムが開始されるときに開始されます。事前開始ジョブを終了するだけで、次のステップを実行しない場合、特定のサーバーの開始要求はいずれも失敗します。

2. **事前開始ジョブ項目除去 (RMVPJE)** コマンドを使用して、サブシステム記述内の事前ジョブ項目を除去します。

RMVPJE コマンドで削除した事前開始ジョブ項目は、サブシステム記述から永続的に除去されます。項目が除去されると、システムに対する新しい要求は成功しますが、ジョブ開始のパフォーマンス・オーバーヘッドが発生します。

関連資料:

事前開始ジョブ終了 (ENDPJ) コマンド

事前開始ジョブ項目除去 (RMVPJE) コマンド

ルーティング項目:

IBM i ジョブは、サブシステム記述内のルーティング項目を使用してサブシステムにルーティングされます。QSYSWRK サブシステム内のリスナー・ジョブ用の経路指定項目は、IBM i のインストール後に表示されます。このジョブは、QUSER ユーザー・プロファイルの下で開始され、QSYSNOMAX ジョブ待ち行列が使用されます。

V5R2 より前のバージョンでは、サーバー・ジョブは QSYSWRK サブシステム中で実行されていました。V5R2 では、サーバー・ジョブはデフォルトでは QUSRWRK 内で実行されます。サーバー・ジョブの特性は、IBM i で自動的に構成されることにもなる事前開始ジョブ項目からとられます。事前開始ジョブがサーバーに対して使用されないようにするためにこの項目が除去される場合、サーバー・ジョブは、対応するリスナー・ジョブの特性を使用して開始されます。

以下に、リスナー・ジョブ用の QSYSWRK サブシステム内の初期構成をリストします。

サブシステム

QSYSWRK

ジョブ待ち行列

QSYSNOMAX

ユーザー

QUSER

経路指定データ

QRWTLSTN

ジョブ名

QRWTLSTN

クラス QSYSCLS20

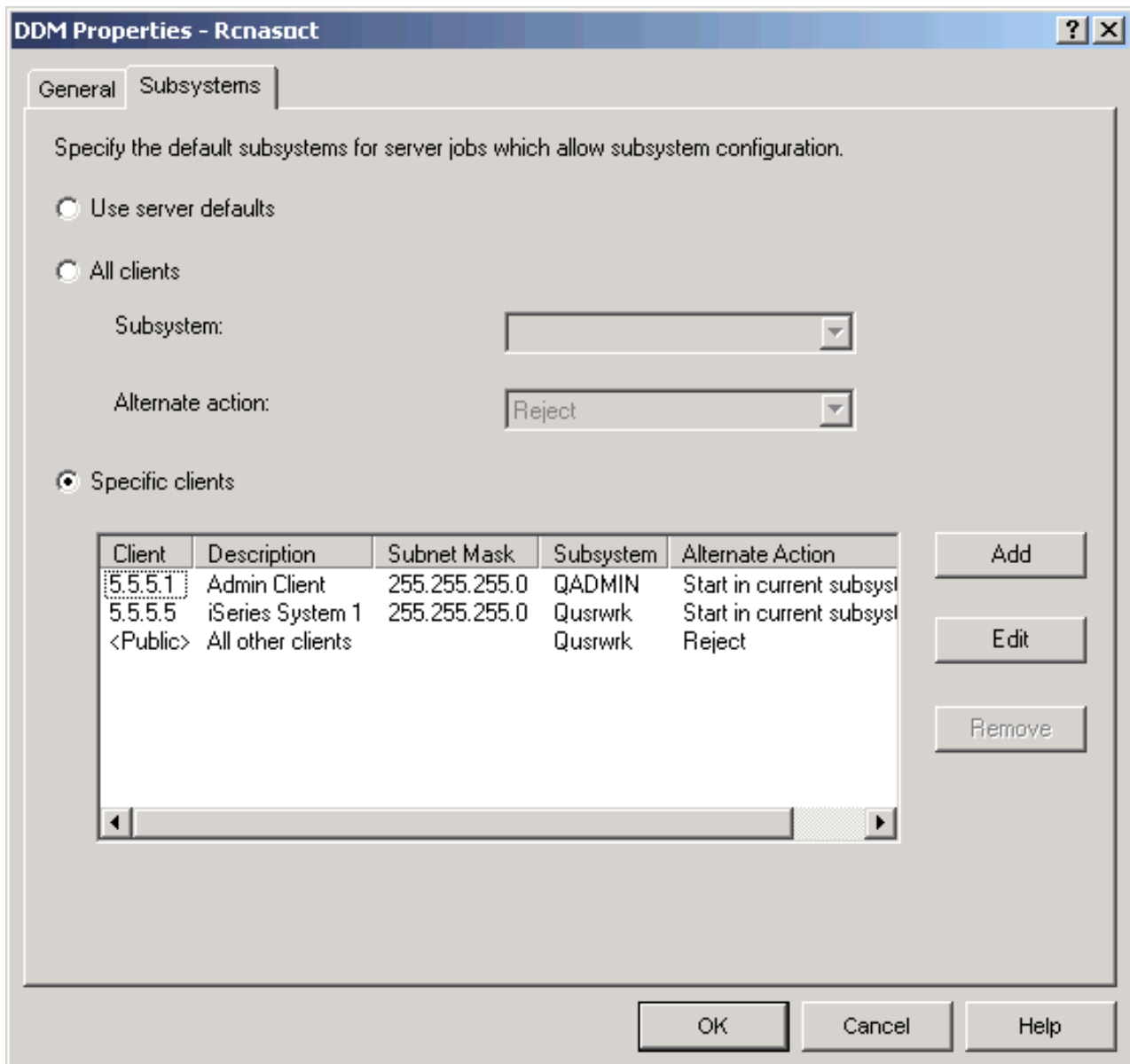
サーバー・ジョブ・サブシステムの構成

デフォルトでは DDM TCP/IP サーバー・ジョブは QUSRWRK サブシステム内で実行されます。System i ナビゲーターを使用して、クライアントの IP アドレスを基にした代替サブシステム中ですべてまたは特定のサーバー・ジョブを実行するように、DDM サーバー・ジョブを構成できます。

この構成をセットアップするには、次のようにします。

1. **事前開始ジョブ項目の追加 (ADDPJE)** コマンドを使用して、ご希望のサブシステムの事前開始ジョブ項目を作成します。
2. **事前開始ジョブの開始 (STRPJ)** コマンドを使用して、作成した事前開始ジョブ項目を開始します。
3. System i ナビゲーター から「ネットワーク」を展開します。
4. 「サーバー」を展開します。
5. 「TCP/IP」をクリックします。
6. 右側のパネルに表示されているサーバーのリスト中で、「DDM」を右クリックし、「プロパティ」を選択します。
7. 「サブシステム」タブで、特定のクライアントとサブシステムの名前を追加します。

以下の例で、管理担当者は QADMIN に接続して実行でき、ネットワーク中の別のサーバーは QUSRWRK に接続して実行できます。他のクライアントはすべて拒否されます。



関連資料:

事前開始ジョブ項目追加 (ADDPJE) コマンド

事前開始ジョブ開始 (STRPJ) コマンド

287 ページの『事前開始ジョブ』

事前開始ジョブ (活動事前開始ジョブの表示 (DSPACTPJ) コマンドなど) に関する情報は、その情報が TCP/IP 接続要求の結果として開始された事前開始ジョブに関係がある場合でさえ、事前開始ジョブを開始するために行われる要求を示すためだけに、「プログラム開始要求」という用語を使用します。

サーバー・ジョブの識別

ある特定のジョブを識別できることは、問題を調査し、パフォーマンス・データを収集するための前提条件です。

システムで開始したサーバーを見ても、サーバー・ジョブと特定のアプリケーション・リクエスターまたは特定の PC クライアントとを関連付けるのは、難しいことがあります。System i ナビゲーターは、ジョブ

をより容易にするこれらのタスク用のサポートを提供します。これらのトピックでは、System i ナビゲーター を使用していないときにデバッグまたはパフォーマンス調査を開始する前に、サーバー・ジョブを識別する方法に関する情報を提供します。

IBM i ジョブ名:

IBM i オペレーティング・システム上で使用されているジョブ名は、以下の部分から成り立っています。

- 単純ジョブ名
- ユーザー ID
- ジョブ番号 (昇順)

DDM サーバー・ジョブは、以下の規則に従います。

- ジョブ名は、QRWTSRVR です。
- ユーザー ID
 - 事前開始ジョブが使用されていなくても常に QUSER になります。
 - ジョブ・ログは、ユーザーが現在ジョブを使用していることを示します。
- ジョブ番号は、実行管理機能によって作成されます。

サーバー・ジョブの表示:

これらのメソッドを使用すると、サーバー・ジョブの識別に役立ちます。

1 つ目の方法は、**WRKACTJOB (活動ジョブの処理)** コマンドを使用することです。2 番目の方法は、**ユーザー・ジョブの処理 (WRKUSRJOB)** コマンドを使用することです。3 番目の方法は、どのジョブがどのクライアント・ユーザーによって使用されているかを判別するために履歴・ログを表示することです。

関連資料:

活動ジョブの処理 (WRKACTJOB) コマンド

ユーザー・ジョブ処理 (WRKUSRJOB) コマンド

WRKACTJOB コマンドを使用した活動ジョブの表示:

活動ジョブの処理 (WRKACTJOB) コマンドによって、すべての活動ジョブが表示されます。リスナー・ジョブと同様に、すべてのサーバー・ジョブが表示されます。

以下の図は、**(WRKACTJOB)** コマンドを使用した状況例を示しています。この図には、サーバーに関連したジョブのみが示されています。使用可能な事前開始ジョブを表示するには、F14 を押さなければなりません。

この図には、以下のタイプのジョブが示されています。

- 1 - リスナー・ジョブ
- 2 - 事前開始されたサーバー・ジョブ

```

                                活動ジョブの処理
                                AS400597
                                10/16/09 10:55:40
CPU %: 41.7      経過時間 : 04:37:55      活動ジョブ数 : 42

オプションを入力して、実行キーを押してください。

  2= 変更  3= 保留  4= 終了  5= 処理  6= 解放  7= メッセージの表示
  8= スプール・ファイルの処理  13= 切断 ...
                                現行
OPT  サブシステム/ ユーザー  タイプ  CPU % 機能          状況
    _____
    1  QSYSWRK      QSYS    SBS    .0          DEQW
      .
    1  QRWTLSTN    QUSER   BCH    .0          SELW
      .
    2  QUSRWRK      QSYS    SBS    .0          DEQW
    2  QRWTSRVR    QUSER   PJ     .0          TIMW
    2  QRWTSRVR    QUSER   PJ     .0          TIMW
    2  QRWTSRVR    QUSER   PJ     .0          TIMW
    2  QRWTSRVR    QUSER   PJ     .0          TIMW
    2  QRWTSRVR    QUSER   PJ     .0          TIMW
      .
                                続く...

```

以下のタイプのジョブが示されています。

- PJ** 事前開始サーバー・ジョブ。
- SBS** サブシステム・モニター・ジョブ。
- BCH** リスナー・ジョブ。

活動ジョブの処理 (WRKACTJOB) コマンドによって、すべての活動ジョブが表示されます。リスナー・ジョブと同様に、すべてのサーバー・ジョブが表示されます。

関連資料:

活動ジョブの処理 (WRKACTJOB) コマンド

WRKUSRJOB コマンドを使用した活動ユーザー・ジョブの表示:

ユーザー・ジョブの処理 (WRKUSRJOB) コマンド USER(QUSER) STATUS(*ACTIVE) は、QUSER の下で実行しているすべての活動サーバー・ジョブを表示します。その中には、DDM リスナー・ジョブやすべての DDM サーバー・ジョブが含まれます。このコマンドを使用すると、リストされるジョブの数が少ないため、DDM 関連のジョブを探すのに便利な場合があります。

関連資料:

ユーザー・ジョブ処理 (WRKUSRJOB) コマンド

ヒストリー・ログの表示:

クライアント・ユーザーがサーバーとの正常な接続を確立するたびに、そのジョブは、そのクライアント・ユーザーのプロファイルの下で実行するためにスワップされます。

特定のクライアント・ユーザーと関連したジョブを判別するには、**ログの表示 (DSPLOG)** コマンドを使用してヒストリー・ログを表示することができます。提供されている情報の例が、以下の図に示されています。

活動記録ログの内容の表示

```
DDM ジョブ 036995/QUSER/QRWTSRVR は 01/01/09 の 15:26:43 にユーザー MEL にサービス中。  
DDM ジョブ 036995/QUSER/QRWTSRVR は 01/01/09 の 15:45:08 にユーザー REBECCA にサービス中。  
DDM ジョブ 036995/QUSER/QRWTSRVR は 01/01/09 の 15:56:21 にユーザー NANCY にサービス中。  
DDM ジョブ 036995/QUSER/QRWTSRVR は 01/01/09 の 16:02:59 にユーザー ROD にサービス中。  
DDM ジョブ 036995/QUSER/QRWTSRVR は 01/01/09 の 16:48:13 にユーザー SMITH にサービス中。  
DDM ジョブ 036995/QUSER/QRWTSRVR は 01/01/09 の 17:10:27 にユーザー DAVID にサービス中。  
.  
.  
.  
.  
.
```

続行するためには、実行キーを押してください。

F3=終了 F10=すべての表示 F12=取り消し

ログの表示 (DSPLOG) コマンドに MSGID パラメーターを使用し、不要な項目をフィルターに掛ける方法の例を以下に示します。

```
DSPL0G MSGID(CPI3E34)
```

QRWOPTIONS データ域に該当するオプションを設定して、これらのレコードがヒストリー・ログに書き込まれないようにすることもできます。

関連概念:

424 ページの『QRWOPTIONS データ域』

DDM または DRDA TCP/IP サーバー・ジョブは、開始されると、ユーザーが診断や他のオプションを指定できるデータ域を探します。このデータ域の名前は QRWOPTIONS で、このデータ域は QGPL ライブラリーに常駐している場合にのみ有効になります。このデータ域は、48 文字のストリングから成っています。

関連資料:

ログ表示 (DSPLOG) コマンド

リレーショナル・データベース・ディレクトリーの監査

このトピックでは、リレーショナル・データベース・ディレクトリーへのプログラム・アクセスを監査する方法を説明します。

リレーショナル・データベース・ディレクトリーへのアクセスは、次のいずれかの場合にセキュリティー監査ジャーナルに記録されます。

- システムの QAUDLVL の値が *SYSMGT である。
- ユーザーの AUDLVL の値が *SYSMGT である。

システムは、*SYSMGT 値を使用して、次のコマンドによって行われたすべてのアクセスを監査します。

- リレーショナル・データベース・ディレクトリー項目の追加 (**ADDRDBDIRE**) コマンド
- リレーショナル・データベース・ディレクトリー項目の変更 (**CHGRDBDIRE**) コマンド
- リレーショナル・データベース・ディレクトリー項目の表示 (**DSRDBDIRE**) コマンド

- リレーショナル・データベース・ディレクトリー項目の除去 (RMVRDBDIRE) コマンド
- リレーショナル・データベース・ディレクトリー項目の処理 (WRKRDBDIRE) コマンド

リレーショナル・データベース・ディレクトリーは、ディレクトリー項目コマンドを使用せずに直接読み取ることのできるデータベース・ファイル (QSYS/QADBXRDBD) です。

V5R2 より前のバージョンでは、*PUBLIC に付与された操作権限を使用して、ライブラリー QSYS 内にリレーショナル・データベース (RDB) ディレクトリー・ファイル QADBXRDBD が作成されました。V5R2 以降では、この点が変わっています。したがって、このファイルを使用して RDB ディレクトリーにアクセスする既存のプログラムは、正しく実行されなくなる可能性があります。*ALLOBJ 特殊権限がない場合に限り、QADBXRDBD に上書きして作成された QADBXRMTNM という名前の論理ファイルにアクセスする必要があります。このファイルへの直接アクセスを監査するには、**オブジェクト監査の変更 (CHGOBJAUD)** コマンドによって監査をオンに設定します。

関連資料:

- リレーショナル・データベース・ディレクトリー項目の追加 (ADDRDBDIRE) コマンド
- リレーショナル・データベース・ディレクトリー項目の表示 (DSPRDBDIRE) コマンド
- リレーショナル・データベース・ディレクトリー項目の変更 (CHGRDBDIRE) コマンド
- リレーショナル・データベース・ディレクトリー項目の除去 (RMVRDBDIRE) コマンド
- リレーショナル・データベース・ディレクトリー項目の処理 (WRKRDBDIRE) コマンド
- オブジェクト監査変更 (CHGOBJAUD) コマンド

DDM の操作に関する考慮事項

このトピックでは、IBM i オペレーティング・システムが、クライアントまたはサーバー・システムとして、どのように別の IBM i と通信してリモート・ファイル処理を行うかについて述べています。また、IBM iが、IBM i以外のシステムと通信する場合の相違点についても述べます。

注: このトピックには、IBM i以外のシステムに関する記載もありますが、DDM を使用するその他のタイプのサーバーが、IBM iと通信するのに必要なすべての情報が述べられているわけではありません。特定のリモート・システムの場合に DDM をどのように使用するかの詳細については、該当するシステム資料を参照してください。

関連概念:

50 ページの『DDM の計画と設計』

分散データ管理機能 (DDM) を正しく使用するには、いくつかの要件を満たす必要があります。

DDM でのファイル・アクセス

このトピックでは、IBM i オペレーティング・システムによってサポートされるファイルのタイプ、DDM ファイルとリモート・ファイルが存在している必要がある状況、およびリモート・ファイル名の指定方法について説明します。また、IBM i と IBM i との間、および IBM i とシステム/36 との間で行うファイル・アクセスについての例と考慮事項も含まれています。

IBM i DDM でサポートされるファイルのタイプ:

IBM i の DDM は、サーバー・システムも IBM i であれば、すべての IBM i ファイル・タイプをサポートします。

サーバー・システムが IBM i でない場合、対応するファイル・タイプの名前は、そのサーバーでは別の名前と呼ばれていることがあります。次の表に、IBM i 以外のファイルと DDM 体系ファイルに相当するそれぞれの IBM i ファイル・タイプを示します。

IBM i タイプ	IBM i 以外および DDM 体系のタイプ
キーなし物理ファイル	順次 (または直接) アクセス・ファイル
キー付き物理ファイル	キー順アクセス・ファイル
論理ファイル	論理ファイル

次に、IBM i がサポートするファイル・タイプに適用される考慮事項を述べます。

- ソース・システムやターゲット・システムが IBM i でもシステム/38 でもない場合、IBM i の多様式論理ファイルは DDM ではサポートされません。
- ターゲット・システムの物理 (順次または直接) ファイルの場合、ファイルの終わりを超えるレコード番号を指定すると、そのファイルは拡張されず、エラーとなります。
- ターゲット・システムの非直接順次ファイルの場合、**物理ファイル・メンバー消去 (CLRPFM)** コマンドで、削除済みレコードを使ってファイル・メンバーを作成することはできません。
- 高水準言語 (HLL) プログラムにおいて DDM ファイルは、データ・ファイルとしても、またはソース・ファイルとしても使用することができます。ただし、DDM ファイルをソース・ファイルとして使用するには、ターゲット・システムが IBM i またはシステム/38 でなければならず、その DDM ファイルに関連するリモート・ファイルは、ターゲット・システム上でソース・ファイルとして定義される必要があります。つまり、そのリモート・ファイルは、ターゲット IBM i またはターゲット・システム/38 上で、**物理ファイル作成 (CRTPF)** コマンドを使って FILETYPE (*SRC) として作成されたか、または、**ファイル・コピー (CPYF)** コマンドに FMTOPT (*CVTSRC) を指定して作成されたファイルでなければなりません。

DDM ファイルとリモート・ファイルの存在:

リモート・ファイルに関連した DDM ファイルがクライアント・システムにすでに存在していない限り、どのようなタイプの操作 (オープン、読み取り、書き込み、または表示など) の場合もサーバー・システムにあるそのファイルへアクセスすることはできません。

ただし、**DDM ファイル作成 (CRTDDMF)** コマンドや **DDM ファイル変更 (CHGDDMF)** コマンドを使って DDM ファイルを作成、または変更する時点で、そのリモート・ファイルは存在していなくてもかまいません。というのは DDM ファイルがアクセス用に実際にオープンされるまでは、そのリモート・ファイルが参照されることはないからです。

DDM のサーバー・システム・ファイル名の指定の規則:

DDM ファイル (ローカル IBM i 内の) の名前指定に関して定められた規則は、IBM i の他のすべてのファイル・タイプの場合と同じです。ただし、リモート・ファイル名の指定に関する規則は、サーバー・システムのタイプによって異なります。

リモート・ファイル名は、**DDM ファイル作成 (CRTDDMF)** および **DDM ファイル変更 (CHGDDMF)** コマンドの RMTFILE パラメーターでのみ指定できます。以下のリストは、リモート・ファイル名を指定するために RMTFILE パラメーターで使用できる最大文字数を示します。

- IBM i (データベース管理) の場合: 33 文字。ライブラリー修飾子とメンバー名のついた完全名を指定すると、この最大文字数になります。以下にその例を示します。

```
LIBRARY123/FILE123456(MEMBER1234)
```

値 DM をこの名前に付け加えれば、これがデータ管理ファイルであることを指定することができます。その名前と DM との間に、1 つ以上の空白を入れることができます。これはデフォルトです。

- IBM i (フォルダー管理サービス) の場合: 76 文字。完全修飾パス名 (76 文字から成る) を指定すると、この最大文字数になります。以下にその例を示します。

```
/Path123/Path223/Path323/Path423/  
Path523/Path623/Path723/Path823/Path923/DOC1 FMS
```

値 FMS は、これがフォルダー管理オブジェクトであることを指定します。この名前と FMS との間に、1 つ以上の空白を入れることができます。

- システム/38 の場合: 33 文字。ライブラリー修飾子とメンバー名の入った完全名を指定すると、この最大文字数になります。以下にその例を示します。

```
FILE123456.LIBRARY123(MEMBER1234)
```

- システム/36 および CICS の場合: 8 文字。以下にその例を示します。

```
FILE1234
```

- 他のシステムの場合: DDM 体系で許容される最大長は、255 文字です。実際の最大長と構文は、サーバー・システムにより異なります。

DDM の IBM i ターゲット・システムのファイル名:

ローカル・ファイルの場合と同様に、IBM i リモート・ファイル、ライブラリー名、またはメンバーは、英字 (A から Z まで、\$、#、または @) で始める必要があり、その後最大 9 文字の英数字 (A から Z、0 から 9、\$、#、@、_、またはピリオド (.)) を続けることができます。この名前は、10 文字を超えてはなりません。空白を使用することはできません。

拡張名を使用すると、引用符 (") で囲んで図形文字を追加して入れることができます。拡張名も 10 文字を超えてはならず、また、引用符もその名前の一部となるため、図形文字数は 8 文字に限定されます。小文字は小文字のままになります。拡張名の例を次に示します。

```
"Test.Job"  
"()/+="
```

IBM i がサーバー・システムであれば、次の例に示すとおり、さまざまな形でファイル名を指定することができます。

library-name

リモート・ファイルの入ったライブラリーの名前を指定します。*LIBL を指定すれば、サーバー・システムにあるジョブのライブラリー・リストの中で、指定ファイル名が探されることとなります。*CURLIB は、リモート・サーバーの現行ライブラリーを指定します。

remote-file-name

IBM i ターゲットにあるデータベース・ファイル (物理ファイル、論理ファイル、またはソース・ファイル) の名前を指定します。

***NONSTD**

IBM i ターゲット・システムの場合に、メンバー名が、リモート・ファイル名とともに組み入れられるように指定します。完全名の前に値 *NONSTD を付ける必要があり、完全名は単一引用符で囲み、すべて英大文字でなければなりません。

注: DDM ファイル作成画面や DDM ファイル変更画面で F4 (プロンプト) を押し、リモート・ファイル名 abcde を指定して *NONSTD の値を指定すると、サーバーは、abcde を 'ABCDE' (すべ

て英大文字) に変換してから、要求を処理します。ただしリモート・ファイル名内に斜線や括弧があると、システムはその名前の前後に単一引用符を付けますが、その名前を英大文字に変換することはありません。

したがって、リモート・ファイル名に *NONSTD 値を使用していて、ターゲット・サーバーが英大文字のファイル名を要求してきた場合は、F4 (プロンプト) を使用してリモート・ファイルを英大文字で入力します。

member-name

リモート・ファイル内のメンバーの名前を指定します。メンバー名は括弧で囲み、ファイル名の直後に (スペースを入れずに) 続けて指定する必要があります。メンバー名を指定しないと、*FIRST がとられ、ファイル内の最初の (または唯一の) メンバーがアクセスされます。このメンバーは、ファイル内で最も古い (または唯一の) メンバーです。

*LAST をサポートするのは、データベース・ファイル一時変更 (OVRDBF)、物理ファイル・メンバー削除 (CLRPFM)、物理ファイル・メンバー初期設定 (INZPFM)、物理ファイル・メンバー再編成 (RGZPFM)、データベース・ファイル・オープン (OPNDBF)、および QUERY ファイル・オープン (OPNQRYP) コマンドのみです。*LAST は、ファイル内で最も新しい (または唯一の) メンバーです。

以下に、有効な IBM i リモート・ファイル名の例を示します。

```
CUSTMAST
PRODLIB/CUSTMAST
*NONSTD 'CUSTMAST(MBR1) '
*NONSTD '*LIBL/CUSTMAST(MBR2) '
*NONSTD 'PRODLIB/CUSTMAST(MBR3) DM'
*NONSTD 'PRODLIB/CUSTMAST(*FIRST) '
```

DDM の IBM i 以外のターゲット・システムのファイル名:

IBM i 以外のシステムのリモート・ファイル名の場合、その名前は、サーバー・システムで要求される形式でなければなりません。

リモート・ファイル名に特殊文字を用いるときには、IBM i メンバー名の指定方法に示したように、その名前の指定には *NONSTD と単一引用符を使用する必要があります。名前ストリングが 10 文字以下で、特殊文字も入っていないならば、入力するときに *NONSTD 値と単一引用符を付けなくてもかまいません。

DDM の共通名をもつファイルに対するロケーション固有ファイル名の使用:

ネットワークに複数のシステムがあるときには、DDM ファイルにそれぞれロケーション固有のファイル名を付けるようにすれば、共通する名前を持つファイルを処理する際に、どのサーバー・システムへアクセスすればよいかについて混乱が生じるのを防ぐことができます。

例えば、複数のシステムで在庫ファイルに INVEN という名前を使用している場合、ニューヨーク、セントルイス、およびダラスのファイルにアクセスするのに、ローカル・システムの DDM ファイル用にそれぞれ NYCINVEN、STLINVEN、および DALINVEN などのロケーション固有の名前を使うことで、正しいファイルにアクセスしやすくなります。

DDM ファイル名の一部から宛先サーバー・システムを識別できるように省略語やコードを設けておけば、該当リモート・ファイルがどこにあるかを保管するのが容易になります。

レコード様式を持った IBM i 以外のリモート・ファイルの場合、そのレコード様式の場合と同じ名前を DDM ファイルにも用いれば便利です。

例: IBM i DDM リモート・ファイルへのアクセス (IBM i から IBM i):

次の例は、DDM ファイルへのアクセスが、どのようにして他のいずれかのシステムにある実際のファイルに対する間接参照 (DDM を使用した) になるかを示しています。いずれも IBM i から IBM i へのアクセスの例です。

注: この例ではすべて、ローカル IBM i オペレーティング・システムにある DDM ファイルは DDMLIB/RMTCAR という名前であり、そのファイルは、シカゴのシステムにある SALES/CAR という名前のリモート・ファイルと関連しているものとします。

リモート・ファイルへアクセスするための DDM ファイルの作成

```
CRTDDMF FILE(DDMLIB/RMTCAR) RMTFILE(SALES/CAR)
      RMTLOCNAME(CHICAGO) TEXT('Chicago file SALES/CAR')
```

このコマンドは、RMTCAR という名前の DDM ファイルを作成し、それをローカル・システムにある DDMLIB ライブラリーに保管します。アクセス先のリモート・ファイルは、シカゴのシステムにある SALES ライブラリー内の CAR データベース・ファイルになります。(DDM ファイル作成 (CRTDDMF) コマンドを使用して DDM ファイルを作成する時点では、リモート・ファイルはアクセスされません。DDM ファイルの作成時には、ファイル SALES/CAR が存在しているかどうかの検査は、行われません。) 後で DDM ファイルへローカル・プログラムからアクセスしたときに、DDM がリモート・ロケーション CHICAGO を使って、シカゴのシステムにある SALES/CAR ファイルへアクセスします。

リモート・ファイルへのローカル・ファイルのコピー

```
CPYF FROMFILE(QGPL/AUTO) TOFILE(DDMLIB/RMTCAR)
```

このコマンドは、DDM ファイル DDMLIB/RMTCAR を使用して、ローカル・システムにある QGPL ライブラリー内の AUTO ファイルのデータを、シカゴのシステムにある SALES/CAR という名前のリモート・ファイルへコピーするためのものです。

使用するリモート・ファイルとメンバーの割り振り

```
ALCOBJ OBJ((DDMLIB/RMTCAR *FILE *EXCL))
```

オブジェクト割り振り (ALCOBJ) コマンドを用いれば、クライアント・システム上の DDM ファイル (RMTCAR) と、サーバー・システム上のリモート・ファイルの最初のメンバー (ファイルそのものも一緒に) を両方とも割り振る (ロックする) ことができます。実際には、次のコマンドがターゲット・システムで実行されます。

```
ALCOBJ OBJ((SALES/CAR *FILE *EXCL *FIRST))
```

サーバー・システムで実行されます。

DDM ファイルへのローカル・ファイルの一時変更

```
OVRDBF FILE(FILEA) TOFILE(DDMLIB/RMTCAR)
      POSITION(*RRN 3000)
```

このコマンドは、データベース・ファイル FILEA を、DDMLIB ライブラリー内に保管されている DDM ファイル RMTCAR に一時変更します。このファイルは両方とも、クライアント・システムにあります。DDM ファイル内で識別されたリモート・ファイル (この例では、シカゴのシステムの SALES/CAR) は、クライアント・システム・プログラムが実際に使用するファイルになります。このリモート・ファイルをオープンすると、最初にアクセスされるレコードは、レコード 3000 になります。

リモート・ファイル内のレコードの表示

```
DSPPFM FILE(DDMLIB/RMTCAR)
```

このコマンドは、DDM ファイル DDMLIB/RMTCAR と関連したリモート・ファイル SALES/CAR の最初のメンバー内のレコードを表示します。

DDM ファイルのオブジェクト記述の表示

```
DSPOBJD OBJ(DDMLIB/RMTCAR) OBJTYPE(*FILE)
```

このコマンドは、ローカル・サーバーで、RMTCAR DDM ファイルのオブジェクト記述を表示します。このコマンドからは、シカゴのシステムにある関連リモート・ファイルへの参照は行いません。

DDM ファイルのファイル記述の表示

```
DSPFD FILE(DDMLIB/RMTCAR) TYPE(*ATR) FILEATR(*DDM)  
SYSTEM(*LCL)
```

このコマンドは、クライアント・システムで、DDMLIB ライブラリー内の RMTCAR という名前の DDM ファイルのファイル記述を表示します。TYPE パラメーターでの指示どおりに、DDM ファイルの属性が表示されます。FILEATR(*DDM) を指定しているため、DDM ファイルの属性のみが表示されます。

SYSTEM(*LCL) を指定しているため、DDM ファイルの属性が表示され、そしてリモート・システムにはアクセスされません。SYSTEM(*RMT) を指定すれば、関連リモート・ファイルの属性が表示されます。*RMT または *ALL を指定すれば、リモート・ファイルの属性を取り出すために、リモート・システムへのアクセスが行われます。

DDM ファイルの削除

```
DLTF FILE(DDMLIB/RMTCAR) SYSTEM(*LCL)
```

このコマンドは、ローカル・システムにある DDM ファイルを削除します。ここでも、シカゴのシステムにある関連 SALES/CAR ファイルに対する参照は行われません。SYSTEM(*RMT) または SYSTEM(*FILETYPE) を指定すると、シカゴのシステムにある SALES/CAR が削除されます。

例: システム/36 DDM リモート・ファイルへのアクセス (IBM i から IBM i):

前項に示したコマンド例 (IBM i - IBM i 間の例) のうち、最初の例以外はすべて、システム/36 のファイルにアクセスする場合と同じようにコーディングを行うことができます。

つまり、システム/36 での命名規則に合わせて、リモート・ファイル名の SALES/CAR を CAR に変更すれば、IBM i ファイルの代わりにリモートのシステム/36 ファイルへアクセスするように変更しなくても、すべてのコマンド (最初のものを除く) をそのまま使うことができます。

ここでは、トピック『例: IBM i DDM リモート・ファイルへのアクセス (IBM i から IBM i)』の最初の例をコーディングし直して、リモートのシステム/36 ファイルにアクセスできるようにしています。リモート・ファイル名を変更する以外に、パラメーター LVLCHK(*NO) もコーディングする必要があります。

```
CRTDDMF FILE(DDMLIB/RMTCAR) RMTFILE(*NONSTD 'CAR')  
RMTLOCNAME(CHICAGO) TEXT('Chicago file CAR on S/36')  
LVLCHK(*NO)
```

このコマンドは、RMTCAR という名前の DDM 参照ファイルを作成し、それをローカル IBM i オペレーティング・システムの DDMLIB ライブラリー内に保管します。アクセスされるリモート・ファイルは、CHICAGO という名前のシステム/36 にある CAR ファイルです。レベル検査が行われないようにするた

め、LVLCHK(*NO) を指定しています。その理由は、システム/36 ファイル用に作成されたレベル識別コードと、システム/36 ファイルへのアクセスのために備わったプログラム内の識別コードが一致していないからです。

DDM でのメンバーへのアクセス

データベース入出力操作でメンバーがサポートされるのは、サーバー・システムが IBM i 製品またはシステム/38 製品の場合だけです。

サーバー・システムが IBM i 製品またはシステム/38 製品の場合、メンバーは、オブジェクト割り振り (ALCOBJ) コマンドでロックしてから使用することができます。

DDM ファイル自体は、データベース・ファイルのようにメンバーを持つことはありません。ただし、クライアント・システム上でメンバーが識別され (例えば、データベース・ファイル一時変更 (OVRDBF) コマンドを使用して)、サーバー・システムが IBM i 製品またはシステム/38 製品の場合、そのメンバー名は、サーバー・システムのファイル内のメンバーを識別するのに使用されます。サーバー・システムが IBM i 製品でもシステム/38 製品でもなく、メンバー名を *FIRST (場合によっては *LAST) と指定した場合、またはそのファイル名がメンバー名と同じである場合には、DDM ファイル内の RMTFILE パラメーター値は変更なしで送られます。このため、メンバーをサポートしないシステムのファイルにもアクセスすることができます。

以下の状態では、要求元プログラムにはエラー・メッセージが送られ、その機能は実行されません。

- メンバー名が *FIRST でなかったり、場合によっては *LAST でないとき
- ファイル名とメンバー名が異なっていて (例えば、そのファイルのオープン時に)、しかもサーバー・システムがメンバーをサポートしていないとき

例: DDM リモート・メンバーへのアクセス (IBM iのみ):

次の例は、DDM ファイルへのアクセスが、どのようにしてリモート IBM i オペレーティング・システムにあるファイルのメンバーへの間接参照 (DDM を使用した) になるかを示しています。いずれも IBM i から IBM i へのアクセスの例です。

```
CRTDDMF FILE(DDMLIB/RMTCAR) RMTFILE(SALES/CAR)
      RMTLOCNAME(CHICAGO)
OVRDBF FILE(FILE1) TOFILE(DDMLIB/RMTCAR) MBR(TEST1)
OVRDBF FILE(FILE2) TOFILE(DDMLIB/RMTCAR)
```

この例は、前に示したと同様、DDM ファイルの作成を示しています。この後 **OVRDBF** コマンドを使って、FILE1 および FILE2 という名前の 2 つのローカル・ファイルを、ローカル DDM ファイル RMTCAR に一時変更します。アプリケーション・プログラムからこれらのファイルをオープンしようとしても、そのファイルではなく、DDM ファイル DDMLIB/RMTCAR が 2 回オープンされます。(FILE1 も FILE2 もオープンされません。)

指定どおりのサーバー・システムとの通信が確立された後、サーバー・システムの TDDM は、リモート・ファイル SALES/CAR を 2 回オープン (2 回反復) し、そのファイル内で 2 つの別々のメンバーをオープンします (この例の場合)。2 つのメンバーは、TEST1 および *FIRST (最初のメンバー) です。この例では、DDM 会話とターゲット・ジョブはそれぞれ 1 つしか必要ありません。それは、いずれのオープン操作でも、同一の DDM ファイルを使用するため、同じロケーションが使用されるからです。

```
CLRPFM FILE(DDMLIB/RMTCAR) MBR(FRED)
```

このコマンドは、DDMLIB/RMTCAR という名前の DDM ファイルを使用して、サーバー・システムにあるファイル SALES/CAR のメンバー FRED を消去します。

例: 特定メンバーをオープンする DDM ファイル:

DDM ファイル作成 (CRTDDMF) および **DDM ファイル変更 (CHGDDMF)** コマンドでのみ使用できる RMTFILE パラメーターで、*NONSTD 値の後にファイル、ライブラリー、およびメンバー名を使用することにより、特定のファイル・メンバーを指定できます。

このようにファイル・メンバーを指定すれば、アプリケーション・プログラムは、ファイル一時変更を使用しなくても、最初のメンバー (*FIRST) 以外のメンバーを処理することもできます。ただし、複数のメンバーへのリダイレクトを必要とするプログラムの場合は、一時変更を使用する必要があります。また、ローカル・ファイルのメンバーを指定するのに一時変更をすでにプログラムで使用していれば、リモート・ファイルに対しても一時変更を同時に使用する場合でも、それを引き続き使用する必要があります。そうしないと、ローカルで稼働していたプログラムが稼働しなくなります。RMTFILE パラメーターにメンバー名を入れたうえで、別のメンバー名を指定した一時変更を行うと、ファイル・オープン要求は失敗します。

注: 「DDM ファイル作成」または「DDM ファイル変更」画面で F4 (プロンプト) を押し、リモート・ファイル名 abcde を指定して *NONSTD の値を指定すると、サーバーは、abcde を 'ABCDE' (すべて英大文字) に変換してから、要求を処理します。ただしリモート・ファイル名内に斜線や括弧があると、システムはその名前の前後に単一引用符を付けますが、その名前を英大文字に変換することはありません。したがって、リモート・ファイル名に *NONSTD 値を使用していて、サーバー・システムが英大文字のファイル名を要求してきた場合は、F4 (プロンプト) を使用していてもリモート・ファイルを英大文字で入力します。

```
CRTDDMF FILE(DDMLIB/RMTCAR) RMTFILE(*NONSTD  
'SALES/CAR(JULY)') RMTLOCNAME(CHICAGO)
```

プログラムが、クライアント・システムの DDMLIB ライブラリーにある RMTCAR という名前の DDM ファイルをオープンすると、ターゲット IBM i オペレーティング・システムは、ファイル SALES/CAR 内のメンバー JULY をオープンします。

DDM のアクセス方法の処理

アクセス方法は、特定のリモート・ファイルがオープンされた後、どの機能サブセットを実行できるかを制御します。

IBM i プログラムや、IBM i 以外のファイルを共用するプログラム・グループは、ローカル IBM i オペレーティング・システムにあるファイルを使用して実行できる操作を、必ずしもすべて実行できるとは限りません。たとえば、IBM i アプリケーション・プログラムが、SHARE (*YES) を指定されたキー付きファイルをオープンして、キー順入出力操作を行うと仮定します。このプログラムは次に、同じオープン・データ・パス (ODP) を使用して (SHARE が指定されているため) 相対レコード番号操作を行う別のプログラムを呼び出します。相対レコード番号は、レコード・ロケーションと、データベース・ファイル、メンバー、またはサブファイルの開始点との関係を指定します。**データベース・ファイル一時変更 (OVRDBF)** コマンドで、システム/36 プラットフォームにあるリモート・キー付きファイルを使用するように最初のプログラムをリダイレクトすると、このスキーマは機能しなくなります。キー順 アクセス方法を選択すると、レコード番号操作は失敗します。レコード番号 アクセス方法を選択すると、キー順操作は失敗します。

クライアント・システムとサーバー・システムが両方とも IBM i 製品のときには、アクセス方法を使用しません。サーバー・システムが IBM i 製品でもシステム/38 製品でもない場合は、潜在的な問題が存在します。混合アクセス・アクセス方法 (*COMBINED) は システム/36 ではサポートされず、IBM i またはシステム/38 プラットフォーム以外のどのターゲット・システムでもサポートされません。

アクセス・インテント:

プログラムでのファイルをオープンするときには、そのファイル内のレコードをどのように処理したいかを指定する必要があります。処理とは、読み取り、追加、更新、削除、またはこれらを組み合わせたものをいいます。

このような操作を正常に実行するには、そのプログラムを実行するジョブまたはユーザー、あるいはその両方が、操作に応じたデータ権限を持っていないと仮定してなりません。IBM i オペレーティング・システムでは、ファイルのオープン時にはデータ権限がすべて検査されるとは限りませんが、そのファイルを使用して対応する入出力操作が行われるときには、個々の必須データ権限があるかどうかの検査が行われます。システム/36 では、オープン時にこれらのデータ権限の検査を行うので、リモート・ファイルに対する要求側のデータ権限がローカル・ファイルの場合と同じ（この場合は機能する）であっても、システム/36 でリモート・ファイルを使用するプログラムは機能しなくなります。

たとえば、IBM i オペレーティング・システムの 2 つのユーザー・グループが、ある 1 つのプログラムを使って同一のローカル IBM i ファイルにアクセスすると仮定します。このとき、グループ A には *READ 権限しかないのに対して、グループ B には *READ、*ADD、および *UPDATE があると、プログラムは常に、*READ、*ADD、および *UPDATE 用にファイルをオープンするとします。しかし、プログラムには、グループ A のメンバーがプログラムを呼び出すときに使用する読み取り専用論理パスしかありません。この前提の場合、グループ A のメンバーがレコードを追加または更新しようとするれば、例外は作成されますが、権限例外が生じることはありません。ここで、両方のユーザー・グループのメンバーが、ローカル IBM i ファイルに対するのと同じデータ権限を持っていても、リモートのシステム/36 ファイルにプログラムをリダイレクトすると、グループ A のメンバーに関してはプログラムが作動しなくなることがあります。これは、要求側のデータ権限がオープン要求に伴うアクセス・インテント・リストに指定されているデータ権限と一致しない場合、システム/36 がそのようなオープン要求を拒否することができるからです。

キー・フィールドの更新:

IBM i プログラムは、キー・フィールドを含め、データ・レコードの任意の部分を変更することができます。

この例外は ILE COBOL プログラムで、ILE COBOL 言語はキー・フィールドの変更を許可しません。システム/36 プログラムでは、ファイル・オープン時に指定されるアクセス方法にかかわらず、レコードの基本キー・フィールドを変更することはできません。場合によっては論理ファイル・キー・フィールドを変更することはできますが、基本キー・フィールドは決して変更できません。

たとえば、通常はローカル・キー付きファイル内のキー・フィールドを変更するための ILE RPG プログラムを、システム/36 (または類似の制約事項のあるその他のシステム) にあるリモート・キー付きファイルへリダイレクトすると失敗する可能性があるという意味です。キー・フィールドを変更しようとしたときに使用していたアクセス方法やアクセス・パスに応じて、それぞれ異なったいくつかのエラーが DDM ターゲット・システムから戻されることがあります。

削除済みレコード:

IBM i オペレーティング・システムでは、レコードの削除済みのマークは、システムが付けます。

このマーク付けが行われるのは、活動レコードをアプリケーションが削除したときか、または、削除済みレコードを使ってファイルを作成した (たとえば、物理ファイル・メンバー初期設定 (INZPFM) コマンドを使用して) ときです。レコードをファイルに追加しても、またはファイル内で変更しても、その後から削除操作を行わない限り、そのレコードには削除済みのマークはつきません。システム/36 など、その他のシス

テムの場合は、削除状況を表すため、そのレコードで特殊なデータ値が使用されることがあります。たとえば、レコードの桁がすべて 16 進数 FF であると、そのレコードは削除済みと見なされます。

これは、通常はローカル・ファイルでのレコードの追加や変更を使用する IBM i アプリケーションの場合、IBM i プラットフォームでもシステム/38 プラットフォームでもないサーバーにあるリモート・ファイルを使用してこれらの操作を行おうとすると、エラーが生じることがあるという意味です。ターゲット DDM サーバーにより削除されたと見なされているレコードがアプリケーション・プログラムによって指定されると、ターゲット・システムはその追加/変更要求を拒否する場合があります。

ブロック化レコードの処理:

リモート・システムに送るレコードを、SEQONLY を使用してブロック化すると、ブロックがいっぱいになるまでレコードは送られません。ブロックの送信前にソース・ジョブを取り消すと、そのブロック内のレコードは失われます。ブロック化を使用しているときに、ソース・ジョブを取り消す場合は、その前にデータ操作の強制終了またはファイルのクローズを必ず行わなければなりません。

可変長レコード:

IBM i DDM は、DDM 体系の定義どおりに、可変長レコード・ファイルをサポートします。

IBM i DDM を使用すると、IBM i でもシステム/38 プラットフォームでもないサーバー・システムで可変長レコード・ファイルをオープンすることができます。(更新目的でファイルをオープンするのでないかぎり、最初から可変長レコード・ファイルをオープンしておくことができます。) その後から読み取り操作を行うと、そのファイルの最大レコード長になるまで、可変長レコードにはブランクが埋め込まれます。末尾ブランクは、書き込み操作で除去されます。

また、実際のレコード長 (ファイルの最大レコード長から、埋め込まれたブランク数を引いた長さ) が、各レコードの終わりに付加されます。書き込み操作の場合、サーバー・システムに送る可変長レコードの長さは、実際のレコード長を使用して決められます。レコード・データの実際の長さには末尾ブランクの数を含めません。

可変長レコード・ファイルは、ファイル作成要求を出せば、IBM i サーバー・システムで作成することもできます。

リモート・ファイルを対象とするその他の DDM 関連機能

リモート・ファイルヘデータ・レコードの入出力操作のためにアクセスする以外に、リモート・ファイルに関連したその他の操作を行うこともできます。これについて、以下のトピックで簡単に説明します。

リモート・システムでのファイル管理機能の実行:

IBM i DDM は、リモート・システムでのファイルの作成、削除、または名前変更をサポートします。

リモート・コマンド投入 (SBMRMTCMD) コマンドを用いれば、このタイプのファイル管理コマンドやその他の CL コマンドの実行をサーバー・システムへ依頼して、そのサーバー上で実行させることができます。**ネットワーク・ジョブ投入 (SBMNETJOB)** コマンドや表示装置パススルーを用いることもでき、その場合 DDM は不要です。

注: 209 ページの『IBM i ターゲット・システムに必要なファイル管理コマンド』、209 ページの『DDM でのメンバー関連コマンド』、および 211 ページの『ソース・ファイル・コマンド』にある CL コマンドは、**SBMRMTCMD** コマンドとともに使用する必要はありません。これらのコマンドの場合、CL コマンドそのものに DDM ファイル名を指定すれば、サーバー・システムで直接実行することができます。

DDM のファイルとメンバーのロック:

DDM ファイルとそれに関連したリモート・ファイルでオブジェクト・ロックをする際は、特別な配慮が必要です。

オブジェクト割り振り (ALCOBJ) およびオブジェクト割り振り解除 (DLCOBJ) コマンド:

ALCOBJ コマンドは、クライアント・システムにある DDM ファイルと、サーバー・システムにあるその関連リモート・ファイルをロックするコマンドです。

ターゲット・システムが IBM i または システム/38 であれば、リモート・ファイルのロックの結果、ローカル・ファイルの場合と同じになります。サーバー・システムが IBM i や システム/38 でなくても同等にロックされますが、**ALCOBJ** コマンドに指定されたものより強力なロック状態が、ターゲット・サーバーによって課せられることがあります。

注: IBM i でも システム/38 でもないサーバー・システムの場合、リモート・ファイルは指定されたロック状態でロックされます。IBM i および システム/38 サーバー・システムの場合に限り、リモート・メンバーは指定値のうち最小のロック状態でロックされます。(IBM i または システム/38 のリモート・ファイルは、共用読み込みロックでロックされます。)

ジョブ処理 (WRKJOB) およびオブジェクト・ロック処理 (WRKOBJLCK) コマンド:

WRKOBJLCK コマンドと、**WRKJOB** コマンドのメニュー・オプション 12 (活動中の場合は、ロック処理) のいずれの場合も、リモート・ファイル (またはそのメンバー) 用に保留されているロックではなく、ローカル DDM ファイル用に保留されているロックのみが示されます。

ロックされる場合の DDM ファイルは、関連リモート・ファイルやメンバー用に設定されているロック状態がどのようなものであっても、常に共用読み取り (*SHRRD) としてロックされます。

DDM 会話の制御:

通常、クライアント・システム・ジョブに関連した DDM 会話は、このトピックに記述された条件のいずれかを満たすまでは、活動状態に保持されます。

1. 会話で使用する DDM ファイルとリモート・ファイルがすべてクローズされ、アンロックされた (割り振り解除された) 場合。
2. リモート・コマンド投入 (SBMRMTCMD) コマンドやファイル記述表示 (DSPFD) コマンド (リモート・ファイル情報を表示するためのコマンド) を使用するなどの DDM 関連機能が他に なにも実行されていない場合。
3. DDM 関連機能が実行中に中断 (たとえば、中断プログラムで) されていない場合。
4. ENDCMTCTL コマンドが出された (DDM ファイルとともに、コミットメント制御を使用した場合) 場合。
5. 分散リレーショナル・データベース体系関連の機能がなにも実行されていない場合。
6. DDM 会話を開始した活動化グループが終了した場合。次のような場合に活動化グループが終了したときには、DDM 会話は停止されません。
 - DDM 会話の有効範囲が、ジョブ・レベルまでである場合。
 - 活動化グループのコミットメント制御の有効範囲がジョブ・レベルまであって、作業単位が未解決の場合。会話は、次のジョブ・レベルのコミットやロールバックまでか、またはジョブの終わりまで残ります。
7. ジョブまたは経路指定ステップが終了した場合。

上記の 1、2、および 3 の条件が満たされていて、しかもソース・ジョブまたは活動化グループが終了していない場合には、その会話は未使用であると見なされます。つまり、その会話は活動状態のままになりますが、要求の処理は行われないうことです。

DDMCNV ジョブ属性のデフォルト値は *KEEP であるため、DDM 会話は未使用のまま活動状態を維持します。クライアント・システム・プログラムが、複数の入出力操作のためにリモート・ファイルへアクセスするクライアント・システム・プログラムの通常の状態下にあつては、これは望ましい状況です。32 ページの図 14 に示されているとおり、またその後の本文の説明にあるとおり、このような操作は、1 回に 1 つずつ行われます。

あるジョブ内で複数の DDM 要求を出すとき、そのジョブ内で DDM ファイルを続けてオープンしてクローズするには、*KEEP を使用して、使用されていない DDM 会話を活動状態に保たなければなりません。(ただし、DDM ファイルが 1 つでもオープンまたはロックされたままであれば、*KEEP は何の効果もありません。)

リモート・ファイルにアクセスはしても、そのファイル内のデータ・レコードへはアクセスしないソース・ジョブでは、ファイル・アクセスの頻度によっては、各ファイルに関連したソース・ジョブ要求が完了するごとに、その DDM 会話を自動的に停止する方が望ましいことがあります。会話が使用されていない間、ソース・ジョブを活動状態に保つか、または自動的に停止するかは、DDMCNV ジョブ属性値 (*KEEP または *DROP) で決めます。

DDMCNV ジョブ属性の値がどのようなものであっても、次のいずれかが生じた場合は、会話は停止されません。

- ジョブが終了したとき。
- 活動化グループが終了したとき。次のような場合に活動化グループが終了したときには、DDM 会話は停止されません。
 - DDM 会話の有効範囲が、ジョブ・レベルまでである場合。
 - 活動化グループのコミットメント制御の有効範囲がジョブ・レベルまでであつて、作業単位が未解決の場合。会話は、次のジョブ・レベルのコミットやロールバックまでか、またはジョブの終わりまで残ります。
- ジョブが、ジョブ経路再指定 (RRTJOB) コマンドを開始したとき。

活動ジョブ内の未使用会話は、DDM 会話再利用 (RCLDDMCNV) またはリソース再利用 (RCLRSC) コマンドによって停止することもできます。また、通信回線障害などのエラーも、会話が停止する原因となります。

DDMCNV 値表示 (WRKJOB コマンド):

特定のソース・ジョブでの DDMCNV ジョブ属性の現行値 (*KEEP または *DROP) を表示するのに、**ジョブ処理 (WRKJOB)** コマンド画面のメニュー・オプション 2 (定義属性表示) を使用することができます。また、**ジョブ属性検索 (RTVJOBA)** コマンドを使えば、CL プログラム内の値を明らかにすることができます。

DDMCNV 値変更 (CHGJOB) コマンド:

ソース・ジョブ内の DDM 会話が使用されなくなるたびにシステムが自動的にその会話を再使用する (または停止する) かどうかを制御するには、**ジョブ変更 (CHGJOB)** コマンドを使用して、システム・デフォルトの *KEEP を *DROP に変更します。その値を *KEEP のままにすれば、任意の時点で **DDM 会話再利用 (RCLDDMCNV)** または**資源再利用 (RCLRSC)** コマンドを使用して、活動ユーザーのいない DDM 会話をすべて (そのジョブ内のみ) 停止することができます。

DDM 資源再使用 (RCLRSC および RCLDDMCNV コマンド):

IBM i ユーザーが、もう活動状態にはないすべての APPC 会話 (DDM 会話を含む) 用の資源を必ずシステムへ返すようにしたい場合には、資源再使用 (RCLRSC) コマンドを使用できます。

分散データ管理機能会話再利用 (RCLDDMCNV) コマンドは、そのコマンドが入力されるクライアント側のジョブの DDM 会話に適用されます。クライアント・ジョブで使用される DDM 会話には、対応するサーバー・ジョブがあります。そのサーバー・ジョブは、対応する DDM 会話を終了すると、自動的に終了します。終了する TCP/IP 会話の場合、サーバー・ジョブは、通常、事前開始ジョブであり、通常は終了されるよりも再生されます。このコマンドを使用すれば、すべてのオープン・ファイルをクローズしたり、リソース再利用 (RCLRSC) コマンドで実行されるその他の機能のいずれかを実行せずに、未使用 DDM 会話を再利用することができます。

RCLDDMCNV コマンドは 1 つのジョブで使用されるすべての DDM 会話に適用されますが、このコマンドを使用しても、そのような DDM 会話のすべてが再利用されることにはなりません。会話が再利用されるのは、活動的に使用されていない場合だけです。コミットメント制御が使用されている場合には、COMMIT または ROLLBACK 操作を行ってからでないと、DDM 会話は再利用できません。

DDM リモート・ファイル情報の表示:

IBM i クライアント・システムのユーザーは、CL コマンドの**ファイル記述表示 (DSPFD)** および**ファイル・フィールド記述表示 (DSPFFD)** を使用して、クライアント・システム上の 1 つ以上の DDM ファイルの属性を表示したり、サーバー・システム上の 1 つ以上のリモート・ファイルの属性を表示したりすることができます。

DDM リモート・ファイル・レコードの表示:

物理ファイル・メンバー表示 (DSPPFM) コマンドを使用すれば、サーバー・システムにあるリモート・ファイルを表示することができます。

ただし、パフォーマンス上の理由の場合、可能な限り表示装置パススルーを使用して、リモート・システムにサインオンし、ファイルを直接表示する必要があります。表示装置パススルーを使用すると、表示イメージのみが通信回線を経由して送られます。DDM を使用してリモート・ファイルにアクセスするときには、同回線を経由して各レコードが別々に送られますが、この方が伝送回数が多くなります。

パススルーを使用できない (たとえば、リモート・ファイルが IBM i、システム/38、またはシステム/36 がない場合、またはパススルーが構成されていないシステムの場合) ときには、できるだけ相対的位置付けではなく直接レコード位置付けを使用するようにしてください。たとえば、レコード番号 100 を表示していて、次にレコード番号 200 を見たい場合には、制御フィールド内には +100 ではなく 200 を入力した方が、そのレコードへより早くアクセスすることができます。ファイルに削除済みレコードが入っていない限り、結果は同じです。

DDM でのコード化文字セット ID:

各国語のサポートを使用する場合は、文字セットを正しく取り扱わなければなりません。

文字情報を管理するためのシステム間サポートには、Character Data Representation Architecture (CDRA) が備わっています。CDRA では、文字を表すために使用されるコード・ポイントを識別し、それらの意味を保持するための必要に応じて、それらのコード (文字データ) を変換するためのコード化文字セット識別コード (CCSID) を定義します。

DDM で CCSID を使用する場合は、以下の考慮事項に注意してください。

- クライアント・システムとサーバー・システムが両方とも CCSID をサポートしていれば、データは、ソース・ジョブのプロセス CCSID に変換されます。
- 片方のシステムが CCSID をサポートする IBM i オペレーティング・システムで、もう片方が CCSID をサポートしない他の何らかのシステムであれば、データは変換されません。
- CCSID をサポートしない任意のクライアント・システムが IBM i サーバー・システム上でファイルを作成するときには、そのファイルは常に CCSID 65535 を指定して作成されます。
- SBMRMTCMD (リモート・コマンド投入) コマンドを使用すれば、CHGPF (物理ファイル変更) コマンドと CCSID パラメーターを指定して、IBM i サーバー・システムにあるファイル CCSID を変更することができます。

オブジェクト配布の使用:

DDM ファイル名をネットワーク・ファイル送信 (SNDNETF) コマンドや ネットワーク・ファイル受信 (RCVNETF) コマンドに指定することはできますが、これらのコマンドはできる限り実際にデータがあるシステム上で実行してください。したがって、両方のシステムが IBM i オペレーティング・システムであって、しかも両方とも SNADS ネットワークにつながっていれば、DDM の代わりにオブジェクト配布を使用して相互にデータをやりとりすることができます。

- **SNDNETF** コマンドは、送ろうとするデータの入ったシステムで直接実行する必要があります。必要であれば、リモート・コマンド投入 (SBMRMTCMD) コマンドや ネットワーク・ジョブ投入 (SBMNETJOB) コマンドを使用して、データがあるシステムへ **SNDNETF** コマンドの実行を依頼することができます。

注: DDM を使用しないで **SNDNETF** コマンドを使用する別の方法として、表示装置パススルーを使用してそのコマンドをサーバー・システムで実行することもできます。

- **RCVNETF** コマンドは、データが送られた先のシステム上で実行する必要があります。必要であれば、**RCVNETF** コマンド上で DDM ファイルを参照し、データを別のシステムへ入れてもかまいません。ただし、DDM ファイルを使用しないようにするには、可能であれば、データが使用されるシステムへデータが送られるよう手配する必要があります。

送信操作と受信操作の両方で、データ・ファイルのファイル・タイプは一致していなければならず、それは、保管ファイルまたは物理データベース・ファイルでなければなりません。ただし DDM を使用するのであれば、転送するファイルは保管ファイルであってはなりません。

DDM でのオブジェクト配布の使用:

SNADS (IBM i オペレーティング・システム上) と DDM (IBM i および非 IBM i 上) の両方を使用して、IBM i オペレーティング・システムと SNADS ネットワークには属していないが DDM を導入しているシステムとの間でファイルを転送することもできます。

システム/36 で SNADS を使用することは可能ですが、IBM i オブジェクト配布には使用できません。

例えば、IBM i DDM ファイルがシステム/36 上のファイルを参照している場合、IBM i オペレーティング・システムで **SNDNETF** コマンドを使用し、オブジェクト配布を使用して、ファイルを別の IBM i に送ることができます。同様に、ファイルが IBM i オペレーティング・システムに送信された場合、**RCVNETF** コマンドを使用することにより、DDM を使用してファイルをシステム/36 で受け取るようにすることができます。

分散データ管理作業の取り消し

アプリケーションをテストしたり、ユーザーの問題を処理したり、特定の装置をモニターしたりしているときに、システム上で実行している作業を終了したい場合があります。

対話式ジョブを使用しているときには、通常はシステムからサインオフしてジョブを終了します。システム上のジョブを取り消したり、中断したりする方法が他にもあります。それらの方法は、ジョブの種類およびジョブが置かれているシステムの種類によって決まります。

ジョブの終了 (ENDJOB) コマンド:

ジョブの終了 (ENDJOB) コマンドは任意のジョブを終了します。

ジョブは、ジョブ待ち行列で活動状態にできますが、活動状態にしなければ終了します。ジョブは、即時に終了したり、時間間隔を指定して終了してジョブ終了処理を行うことができます。

ソース・ジョブを終了すると、ソースとターゲットの両方のジョブが終了します。アプリケーションがコミットメント制御下にある場合、コミットされていない変更はすべてロールバックされます。

要求の終了 (ENDRQS) コマンド:

要求の終了 (ENDRQS) コマンドは、現在ブレークポイントで停止しているローカルまたはソース操作 (要求) を取り消します。

これは、このコマンドがアプリケーション・リクエスター操作または要求を取り消すという意味です。要求の取り消しは、コマンド行で **ENDRQS** を入力するか、「システム要求」メニューからオプション 2 を選択して行うことができます。

現在、中断できないサーバー機能を実行中で、即時に処理できない場合には、このコマンドは割り込みが許可されるまで待機します。

要求が終了すると、現在、取り消し中の要求レベルで呼び出されている要求処理プログラムにエスケープ・メッセージが送信されます。要求処理プログラムは、要求が取り消されたときに終結処理を行うことができるように、エスケープ・メッセージをモニターします。静的記憶域およびオープン・ファイルは、要求処理プログラムによって呼び出されたプログラムに再使用されます。要求処理プログラムによって呼び出されたプログラムには取り消しは通知されません。そのため、それらのプログラムが処理を停止することはできません。

重要: **ENDRQS** コマンドをソース・ジョブで使用すると、予期しない結果が起きて、ターゲットへの接続が失われることになります。

DDM のシステム/36 クライアントとサーバーに関する考慮事項

IBM i オペレーティング・システムが、システム/36 プラットフォームにあるファイルにアクセスするためには、そのシステム/36 にレベル 1.0 の DDM 体系を導入していなければなりません。このトピックには、IBM iがシステム/36 と通信するクライアントまたはサーバー・システムである場合に適用される情報が含まれています。

IBM i とシステム/36 ファイル間の DDM 関連の相違点:

IBM iとシステム/36 がサポートするファイル・タイプはそれぞれ異なるため、この 2 つのシステム間で DDM を使用するときには、いくつかの事項について考慮する必要があります。

一般的に、システム/36 ファイルがローカルで作成される (たとえば、BLDFILE ユーティリティを使用し) と、システム/36 のユーザーは、ファイル・タイプ (S = 順次、D = 直接、または I = 索引付き) や、レコードまたはブロックを割り振るかどうか、そのうちいくつかを割り振るか、また、ファイルを拡張するためにその量をさらに何回そのファイルへ追加できるかを指定します。

また、ファイルを削除可能 (DFILE) または削除不可能 (NDFILE) のいずれにするかを指定することもできます。ファイルを削除不可能 と指定すると、そのファイルにレコードの追加または変更は行えますが、削除することはできません。

上記の属性の指定が終われば、システム/36 はファイルを作成し、それに適切な 16 進文字を埋めます。システム/36 のユーザーが以下のファイルを指定した場合に生じる事例を以下に示します。

- 順次 ファイルの場合、ファイル・スペース全体に 16 進文字 00 が埋められ、ファイルの終わり (EOF) ポインターは、初期エクステントの先頭に設定されます。空の順次ファイルを読み取ろうとすると、EOF 条件が届きます。
- 削除可能 な直接 ファイルの場合、ファイル・スペース全体に 16 進文字 FF (削除済みレコード) が埋められ、EOF ポインターは、初期エクステントの終わりに設定されます。空の削除可能直接ファイルを読み取ろうとすると、該当レコードなしの条件が届きます。
- 削除不可能 な直接 ファイルの場合、ファイル・スペース全体に 16 進文字 40 (ブランクまたはヌル・レコード) が埋められ、EOF ポインターは、初期エクステントの終わりに設定されます。削除不可能な、空の直接ファイルを読み取ろうとすると、ファイルの終わりまでの間の各レコードごとに 1 つずつブランク・レコードが返されます。
- 索引付き ファイルの場合、順次ファイルと同じ方法で作成されます。

通常は、削除可能ファイルがいったん使用されると、削除済みレコードのほとんどない比較的連続した活動レコード・セット、次に、たいていは、データの終わりマーカー、またその後に、連続した削除済みレコード・セットが入って、ファイル終わり (EOF) のスペースになります。つまり、ファイルを再編成しない限り、削除済みレコードの削除取り消し (回復) をすることができるということです。

3 つのタイプのシステム/36 ファイルのうち、システム/36 索引付きファイルは、IBM i サポート論理ファイルとほとんど変わりません。IBM i ソース・プログラムが DDM を使用してシステム/36 上の他のタイプのファイルにアクセスする場合、アプリケーション・プログラマーは、最初に、このトピックの後方にあるシステム/36 に関連した項目を考慮する必要があります。

DDM の システム/36 クライアントからIBM i サーバーへの考慮事項:

システム/36 がクライアント・システムとして通信するために DDM を使用して、IBM i サーバー・システム上のファイルにアクセスするときは、このトピックの情報を適用し、考慮する必要があります。

- システム/36 が IBM i オペレーティング・システム上に直接ファイルを作成すると、IBM i は、最大数のレコードを持つキーなし物理ファイルを作成し、それを削除済みレコードとして準備します。DDM 体系コマンドのファイル消去 (CLRFIL) を非 IBM i クライアント・システムから出すと、ファイルを消去してから作成します。CL コマンドの物理ファイル・メンバー消去 (CLRPFM) をローカルまたはリモート IBM i システムから出しても、ファイルは作成されません。
- システム/36 は、論理ファイルの場合は最大 3 つのキー定義を、キー付き物理ファイルの場合は 1 つのキー定義をサポートします。
- 削除不可能な直接ファイルは、IBM i 上で DDM を使用しても、作成することはできません。さらに、IBM i は、どのようなファイル編成の場合も、削除不可能なファイルをサポートすることはありません。

DDM の IBM i クライアントからシステム/36 サーバーへの考慮事項:

IBM i オペレーティング・システムがクライアント・システムとして通信するために DDM を使用して、システム/36 サーバー・システム上のファイルにアクセスするときは、このトピックの情報を適用し、考慮する必要があります。

- ファイル操作によっては、IBM i サーバー・システムでは拒否されなくても、ターゲット・システム/36 では拒否されるものがあります。例を示します。
 - システム/36 ファイルが削除可能ファイルでない場合には、レコード削除操作は拒否されます。IBM i ソース・システム・ユーザーの側からすれば、理由不明のまま拒否されたことになります。
 - 変更操作で、システム/36 ファイルの 1 次索引内のキーを変更しようとする、常に拒否されます。
- システム/36 環境においては、システム/36 のユーザーが、NOREORG パラメーターを指定された削除不可能なファイルへ削除可能ファイルをコピーしようとする、削除済みレコードがコピーされることがある旨を知らせる警告メッセージが出ます。この場合、オプション 0 (継続) を選択すれば、プロセスを継続することができます。このオプションを選択すると、そのファイルはコピーされ、入力ファイル内に削除済みレコードがあれば、それは出力ファイル内の活動レコードになります。ユーザーが COMPRESS(*NO) を指定すると、IBM i オペレーティング・システムは、コピー要求を拒否します。
- 直接ファイルであって、しかも削除不可能なターゲットのシステム/36 ファイルにデータをコピーするときは、FROMMBR、TOMBR、および MBROPT 以外のすべてのファイル・コピー (CPYF) コマンド・パラメーターに対してデフォルト値を指定する必要があります。
- IBM i オペレーティング・システムでは、ファイル削除 (DLTF) コマンド上のデータに上書きすることはできません。システム/36 にアクセスしている IBM i ユーザーがそのデータを上書きしたい場合には、IBM i でアプリケーション・プログラムを作成するか、または、ユーザーがターゲット・システム/36 にアクセスして、上書き操作を実行する必要があります。
- システム/36 ファイルが当初どのように作成されているかによりませんが、そのファイルに収容できる最大レコード数はおよそ 80 万個です。さらにレコードを追加できるよう、ファイルが拡張可能ではなかったり、ファイルを拡張するのに十分な記憶域スペースが使用できない場合には、最大数はこれよりかなり少なくなります。
- システム/36 は、論理ファイルの場合は最大 3 つのキー定義を、キー付き物理ファイルの場合は 1 つのキー定義をサポートします。
- システム/36 のファイル・サポートでは、活動論理ファイルを持ったファイルを消去することはできません。特定の IBM i プログラム (ILE COBOL プログラムなど) が出力専用のファイルをオープンすると、そのファイルを消去するためのコマンドが出されます。この消去しようとするファイルの上に論理ファイルが存在すれば、ターゲット・システム/36 は、このようなファイルを消去するためのコマンドをすべて拒否します。
- システム/36 ファイル・サポートは、削除済みレコードを自動的にスキップします。IBM i ソース・システム・ユーザーが変更したいレコードが、システム/36 基本ファイル内にあって、少なくとも 1 つの論理ファイルが作成される基礎となったレコードであれば、直接編成とレコード番号別のランダム・アクセスを指定したうえで、そのファイルを入出力モードでオープンする必要があります。この後で、レコード番号別に各レコードを読み取って変更することができます。削除済みレコードが見つかった場合、該当レコードなしの通知が返され、その位置にあったレコードを、書き直すのではなく、書き込むことができます (変更操作ではなく書き込み操作)。
- システム/36 のファイル・サポートでは、使用するファイル・タイプと言語に応じて、ファイル拡張子の扱い方が異なります。ただし、ファイルへアクセスするのに使用するアクセス方法が、そのファイルが作成されたときに使用された方式と同様のものでない限り、IBM i システム・ユーザーは、どのようなタイプのシステム/36 ファイルも拡張することはできません。

IBM i システム・ユーザーが、ファイルが作成されたときに使用された方式と同様のアクセス方法を使用してシステム/36 ファイルへアクセスすれば、次のような方法を使ってそのファイルの使用時にファイルを拡張することができます。

- ファイルが順次ファイルとして作成されていれば、IBM i ユーザーは以下のアクションを行わなければなりません。

- ILE COBOL プログラミング言語の場合: EXTEND オプションを使用してファイルをオープンします。
- PL/I PRPQ の場合: UPDATE オプションを使用してファイルをオープンします。LAST を指定した POSITION オプションを使用して読み取り操作を行ってから、書き込み操作を行います。

(ILE RPG プログラミング言語では、すべての必要なファイル拡張は自動的に処理されます。)

- ファイルが直接ファイルとして作成されていれば、使用する IBM i 言語に応じて、IBM i ユーザーは以下のことを行わなければなりません。
 - ILE COBOL プログラミング言語の場合: I-O オプションを使用してファイルをオープンし、ファイルの終わりポインターをそのファイルの終わりに置き (たとえば、READ LAST を使用して)、書き込み操作を行います。
 - PL/I PRPQ の場合: UPDATE オプションを使ってファイルをオープンし、ファイルの終わり (EOF) ポインターをそのファイルの終わりに置き (たとえば、READ LAST を使って)、書き込み操作を行います。

(ILE RPG プログラミング言語では、すべての必要なファイル拡張は自動的に処理されます。)

- ファイルが索引付きファイルとして作成されていれば、そのファイル内にまだ存在していないキーを持ったレコード用の書き込み操作を行うたびに、そのファイルは拡張されていきます。
- IBM i ユーザーは、順次方式と直接方式 (相対レコード番号による) のどちらを使用してもシステム/36 の順次ファイルへアクセスすることができますが、EOF やデータの終わりに達した時点で大きな相違が生じます。システム/36 順次ファイルを、相対レコード番号アクセス方式によって処理しようとして、入出力用または出力専用オープンすると、活動レコードの終わり (EOF) まできたら、データの終わりより先の使用可能なフリー・スペースに新たにレコードを追加することはできません。そのファイルを一度クローズしてから、再度オープンしてファイルを拡張する必要があります。ファイルを拡張するには、そのファイルを順次ファイルとして再度オープンするか、または、そのファイルを基本ファイルとして使用している論理ファイルをオープンします。
- システム/36 ファイルの場合に通常使用されるアクセス方法は、IBM i パラメーターで *RMTFILE 以外の値に変えられてしまうことがあるため、DDM がシステム/36 ではサポートされない方式でシステム/36 ファイルへアクセスしてしまふことがあります。通常は、ACCMTH パラメーターにデフォルト値 (*RMTFILE) を指定すれば、所定のアクセス方式を使用することができます。一般的ではないアクセス方法 (索引付きファイルへの直接アクセスや順次アクセス、または直接ファイルへの順次アクセス) を用いるには、そのアクセスで ACCMTH パラメーターを明示的に指定する必要があります。

以下の方法で、システム/36 ファイルに通常使用されるアクセス方式を、IBM i オペレーティング・システム上で変更できます。

- DDM ファイル・コマンドの DDM ファイル作成 (CRTDDMF) および DDM ファイル変更 (CHGDDMF) の ACCMTH パラメーターにより
- データベース・ファイル一時変更 (OVRDBF) コマンドの SEQONLY パラメーターにより
- OVRDBF コマンドを使用して、ある DDM ファイルを、別の有効な ACCMTH 値のある別の DDM ファイルに一時変更することにより
- メンバー名が *FIRST または、場合によっては *LAST であるか、あるいは、ファイル名がメンバー名と同じであれば、IBM i ユーザーは、そのメンバー名を使ってシステム/36 へアクセスすることができます。
- ターゲット・システム/36 DDM は、重複 (非固有) キーを持った論理ファイルの作成をサポートすることはできませんが、それは、システム/36 ローカル・データ管理キー分類プログラムが、重複キーを検出するとオプション 1 または 3 のメッセージをサーバー・システムのコンソールへ送るからです。この

場合、サーバー・システムのオペレーターは、ファイル属性を変更して重複キーを使用できるようにするか、または受動側のデータ管理プログラムを取り消さなければなりません。

注: 絶対に、SYSLOG HALT を使用して受動側のデータ管理プログラムを取り消さないでください。

DDM のシステム/36 に対する一時変更に関する考慮事項:

システム/36 上の論理ファイル内のレコードを入手するために、ファイル一時変更が IBM i 上で出された場合、システムごとにキー付きファイル処理する方法が異なるため、結果は予期したものと異なるかもしれません。

IBM i オペレーティング・システムは、アクセス・パスと論理ファイルを使用しますが、これは、ファイルの単一ビューを生成するものです。システム/36 の論理ファイルは、キーと相対レコード番号のリストと見なすことができます。

IBM i オペレーティング・システムからシステム/36 論理ファイルにアクセスする場合

- 相対レコード番号を指定すると、基礎となっているシステム/36 基本ファイルから、そのレコード番号に対応するレコードが届きます。この後、次のレコードを読み取るよう要求すると、その基本ファイルから次の順次レコードを受け取ります。
- キーを指定すると、索引ファイル内で最初に見つかった該当キーに対応するレコードを受け取ります。次のレコードを読み取るよう要求すると、索引ファイル内の次の項目に一致するレコードを受け取ります。

以下の例は、IBM i プログラムがシステム/36 論理ファイルにあるレコードを検索した場合のさまざまな結果を表しています。この例では、以下を前提としています。

- ファイル S36FILEA は基本ファイルであり、S36FILEB は、その基本ファイルをもとに作成された論理ファイルです。
- いずれのファイルも、ターゲット・システム/36 にある対応リモート・ファイルを示す S36FILEA および S36FILEB という名前の DDM ファイルを持っています。
- キー・フィールドは数字フィールドであって、その中には常にレコード番号が入ります。
- 基本ファイル (S36FILEA) 内のレコードは、キーの昇順になっていて、論理ファイル (S36FILEB) 内のレコードは、それと同じキーの降順になっています。
- 次の表に示されたそれぞれの結果を生じるよう、POSITION パラメーター値は可変値として示され、また、いずれのコマンドにも NBRRCDs パラメーターは指定されていません (つまり、読み取られる合計レコード数は、POSITION パラメーター値によってのみ異なるということです)。

```
OVRDBF FILE(S36FILEA) TOFILE(S36FILEB)
        POSITION(*RRN ... or *KEY ...)
CPYF FROMFILE(S36FILEA) TOFILE(ISERIESFILEB)
CRTFILE(*YES) FMTOPT(*NOCHK)
```

データベース・ファイル一時変更 (OVRDBF) コマンド上で POSITION パラメーターに指定する値に応じて、ファイル ISERIESFILEB が IBM i クライアント・システム上で作成された場合に、そのファイルにコピーされることになるレコードを次に示します。

POSITION パラメーター (注を参照)	検索されるレコード
*RRN 1	299 個のレコード (1 から 299 まで)
*KEY 1	1 個のレコード (最初のレコードのみ)
*RRN 299	1 個のレコード (最終のレコードのみ)
*KEY 299	299 個のレコード (299 から 1 まで)

POSITION パラメーター (注を参照)	検索されるレコード
*RRN 150	150 個のレコード (150 から 299 まで)
*KEY 150	150 個のレコード (150 から 1 まで)
注: この欄では、*KEY 値には 1 つのキー・フィールドしかないものと想定し、リモート・ファイル名を、レコード様式名のデフォルト値として使用しています。	

DDM のパーソナル・コンピューター・クライアントから IBM i サーバーへの考慮事項

IBM i Access Family では DDM を使用すれば、パーソナル・コンピューターがクライアント・システムとして、IBM i ターゲットにあるオブジェクトにアクセスすることができます。

IBM i Access Family は、レベル 3.0 の DDM 体系ストリーム・ファイル・アクセス・サポートを使って、フォルダー管理サービス (FMS) のフォルダーや文書にアクセスします。次の考慮事項は、IBM i Access Family で、レベル 3.0 の DDM 体系用の IBM i DDM ターゲット・システム・サポートを使用する場合に関するものです。その他のクライアント・システムで、ストリーム・ファイルやディレクトリーを求めるために、レベル 2.0 またはレベル 3.0 の DDM 体系要求を出す場合にも、この情報は、DDM を使用して IBM i オペレーティング・システムに接続する際に役立つことがあります。

- フォルダー管理サービス (FMS) のフォルダーや文書にアクセスするには、ファイル名またはディレクトリー名の後に FMS を付けなければなりません。名前の終わりとは FMS の間に、1 つ以上の空白を入れてもかまいません。
- 名前が斜線 (/) で始まる場合は、その名前が完全修飾であることを意味します。斜線が先頭でない場合、指定された名前の前に、使用している現行ディレクトリーが追加されます。
- 完全修飾文書名の全長は、76 文字です。これには、使用中の現行ディレクトリーがあればそれも含まれますが、タイプする場合に備えて使用される後書き FMS は含まれません。
- / FMS は、ディレクトリー名のルート・フォルダーを意味します。
- ジョブ・ログに記録されるメッセージ数を減らすため、文書のオープン、GET、PUT、およびクローズの命令の実行中に IBM i ターゲット・システムで生じた特定のエラーは、ジョブ・ログには記録されません。これらの戻りコードの説明は、表 29 を参照してください。

表 29. IBM i 戻りコード

説明	DDM の応答	関数
データ・ストリーム (DS) は使用中	STRIUSRM	GET
データ・ストリーム (DS) は使用中	STRIUSRM	PUT
文書削除 SHDONL (TRUE) が指定されたが、シャドーが存在していない	EXSCNDRM	DELFIL
文書は使用中	FILIUSRM	OPEN
文書は読み取り専用	ACCINTRM	OPEN
文書が見つからない	EXSCNDRM	DELFIL
文書が見つからない	FILNFNRM	OPEN
データの終わり	SUBSTRM	GET
その宣言名のファイルはすでにオープンされている	OPNCNFRM	OPEN
ファイルがオープンされていない	FILNOPRM	GET、PUT、LOCK、UNLOCK
フォルダーは使用中	DRCIUSRM	OPEN
フォルダーが見つからない	DRCNFNRM	OPEN

表 29. IBM i 戻りコード (続き)

説明	DDM の応答	関数
サブストリングが無効である	SUBSTRRM	UNLOCK
ロックされていない領域のアンロック	EXSCNDRM	UNLOCK

- パフォーマンスを向上させるため、IBM i ターゲット・システムは、文書をクローズする際に、文書が実際にクローズされる前にコマンド完了応答メッセージ (CMDCMPRM) をクライアント・システムに戻すという方法で、文書のクローズを処理します。文書がクローズ時に損傷を受けても、ジョブ・ログを見ない限り、そのことを知らせる応答メッセージをユーザーが受け取ることはありません。次回ユーザーがそのファイルを再オープンしたとき、更新済みデータがないこともあります。
- IBM i オペレーティング・システムは、データ・ストリームのロック機能での WAIT をサポートしません。クライアント・システムのユーザーが、WAIT 機能を処理する必要があります。

データの可用性および保護

分散リレーショナル・データベース環境では、データ可用性の観点から、ネットワーク内の個々のシステムでのデータの保護だけでなく、ユーザーがネットワーク内のデータに確実にアクセスできるようにすることが必要です。

IBM i オペレーティング・システムでは、分散リレーショナル・データベース・ネットワーク・システム上にあるデータを確実に使用できる状態にしておくために、以下の一連の機能が用意されています。

- 保管/復元
- ジャーナル管理およびアクセス・パス・ジャーナル処理
- コミットメント制御
- 補助記憶域プール
- チェックサム保護
- ミラー保護および無停電電源装置

通常、各システムのシステム・オペレーターは、そのシステムにあるデータのバックアップと回復に責任を持ちますが、データ冗長性の局面に加えて、ネットワーク冗長性の局面についても考慮する必要があります。方針を計画する場合には、ネットワーク内のデータを最適に利用できることを確認してください。特定のデータの重要性が企業にとって大きければ大きいほど、そのデータにアクセスする方法を多く備えていなければなりません。

分散リレーショナル・データベースの回復のサポート

コンピューター・システムに起こり得る障害には、システム障害 (システム全体が作動しない時)、火災や洪水などの災害によるサイトの消失、またはオブジェクトの損傷ないしは消失があります。分散リレーショナル・データベースの場合は、ネットワークの中の 1 つのシステムに障害が生じると、ネットワーク全体のユーザーが、そのシステム上のリレーショナル・データベースにアクセスできなくなります。

そのリレーショナル・データベースが他のロケーションでの日常の業務活動にとって不可欠である場合には、1 つのシステムが回復するまでの間、ネットワーク全体にわたって企業運営が混乱することになりかねません。データの保護および障害後の回復に関する計画が、分散リレーショナル・データベースでは特に重要であることは明白です。

分散リレーショナル・データベースの各システムは、それぞれ自分のところにあるデータについて、バックアップおよび回復の責任を負います。また、ネットワークの中の各システムは、それぞれがシステム異常終

了後の回復手順も処理します。ただし、オペレーターが経験不足であったり、オペレーターがまったくいないシステムの場合には、分散リレーショナル・データベース管理担当者がディスプレイ装置パススルーを使用して、バックアップおよび回復の手順を行うことができます。

最も一般的に発生する消失のタイプは、オブジェクトまたはオブジェクトのグループの消失です。オブジェクトの消失または損傷は、電源障害、ハードウェア障害、システム・プログラム・エラー、アプリケーション・プログラム・エラー、またはオペレーター・エラーなど、複数の要因が原因となって発生します。IBM i オペレーティング・システムには、システム・プログラム、アプリケーション・プログラム、およびデータの永久消失を防ぐための方法がいくつも用意されています。障害のタイプおよび選択する保護レベルによって異なりますが、プログラムおよびデータのほとんどは保護することができ、回復時間は大幅に縮小することができます。

以下の方法を使用してデータやプログラムを保護できます。

補助記憶域へのデータの書き出し

ファイルの作成コマンドで強制書き出し率 (FRCRATIO) パラメーターを使用すると、データを補助記憶域に強制的に書き出すことができます。強制書き出し率を 1 にすると、該当のテーブルについて、すべての追加、更新、および削除要求が補助記憶域に即時に書き出されることとなります。しかし、このオプションを選択すると、システム・パフォーマンスが低下することがあります。したがって、データベースを保護するための第 1 の方法としては、テーブルの保管およびテーブルのジャーナル処理を考慮すべきです。

物理的な保護

電力供給が突然停止した場合にも、システムが確実に保護されるようにしておくことが、サーバーをクライアントにとって確実に使用可能にしておくための重要な部分となります。無停電電源装置は、別途発注することができ、電源障害、停電、または電圧降下によって電力供給が停止した場合に、電力供給が回復するまでの間、システム装置に電源を提供することによって、システムを電力供給停止から保護するためのものです。通常、無停電電源装置は、すべてのワークステーションを対象に電力を供給するものではありません。IBM i 製品の場合には、無停電電源装置を使用すると、システムでは次のことを行うことができます。

- 短時間の停電または一時的な電圧降下が生じている間、操作を続行する。
- ファイルをクローズし、オブジェクトの保全性を維持することによって、正常に操作を終了する。

分散リレーショナル・データベースのディスク障害後のデータの回復

ディスク障害が起こった場合には、オブジェクトが障害の直前にすべてテープまたはディスクに保管されていない限り、最新の入力データについては回復は不可能です。前に保管されたオブジェクトが復元された後、システムは操作可能ですが、データベースは現行状態ではありません。

補助記憶域プール (ASP)、チェックサム保護、およびミラー保護は、IBM i ディスク回復機能であり、ディスク関連の障害の後に最新入力データを回復するための方法を提供するものです。これらの機能は、追加のシステム・リソースを使用しますが、分散リレーショナル・データベースのシステムに高水準の保護を提供します。システムによっては、サーバーとして他のサーバーよりも重要な機能を果たしているものがあるので、分散リレーショナル・データベース管理担当者は、どのようにすれば、上記のディスク・データ保護方式がネットワーク内の個々のシステムで最適使用できるかについて検討しなければなりません。

関連概念:

バックアップおよび回復

補助記憶域プール:

補助記憶域プール (ASP) は、同一の記憶域に割り当てられた 1 つまたは複数の物理磁気ディスク装置です。ASP を使用すると、指定した物理磁気ディスク上に、特定のタイプのオブジェクトを分離することができます。

システム ASP は、システム・プログラム、およびシステム・プログラムによる処理の結果として作成される一時オブジェクトを分離します。ユーザー ASP は、ライブラリー、SQL オブジェクト、ジャーナル、ジャーナル・レシーバー、アプリケーション、およびデータなどのオブジェクトの分離に使用することができます。IBM i 製品は、最大 32 の基本ユーザー ASP と、223 の独立ユーザー ASP をサポートしています。ライブラリーやオブジェクトは、ユーザー ASP の中に分離すると、他の ASP 中のディスク障害から保護されるので、回復時間が短縮されます。

オブジェクトを ASP に入れると、回復時間の短縮およびオブジェクトの分離に加えて、パフォーマンスの向上を図ることができます。ジャーナル・レシーバーはユーザー ASP の中に分離した場合には、その ASP に関連するディスクはそのジャーナル・レシーバー専用になります。データベース・ファイルを対象とする読み取りおよび書き出し操作を多く必要とする環境では、これによってその ASP 中のディスクに対するアームの競合が減り、ジャーナル処理のパフォーマンスを向上させることができます。

分散リレーショナル・データベースのチェックサム保護:

チェックサム保護は、補助記憶域プール (ASP) のすべてのディスクでのデータ損失を保護します。

チェックサム・ソフトウェアが、ASP データのエンコードされたコピーを、その ASP 内の特別のチェックサム・データ域に維持します。チェックサム保護 ASP 中の永続オブジェクトに加えられた変更は、すべて自動的にチェックサム・セットのチェックサム・データの中に維持されます。チェックサム・セットの中の 1 つの磁気ディスク装置のデータが消失した場合には、システムは、チェックサムおよびセットの中に残っている機能装置上のデータを使用して、消失した装置の内容を再構成します。このようにして、装置のいずれかに障害が生じた場合にも、その内容を回復することができます。こうして再構成されたデータは、障害の時点でディスク上にあった最新情報を反映しています。チェックサム保護は、システム・パフォーマンスに著しい影響を与える可能性があります。分散リレーショナル・データベースでは、これが問題になる場合があります。

分散リレーショナル・データベースのミラー保護:

ミラー保護では、ディスク制御装置、ディスク入出力装置、またはバスなど、異なるディスク関連ハードウェア構成要素を 2 重にすることによって、システムの可用性を増大させます。システムは障害後もそのまま使用可能であり、障害のあったハードウェア構成要素の保守は適当な時期に予定することができます。

ミラー保護のレベルが異なれば、それに応じてシステム可用性のレベルも異なります。たとえば、システム上の磁気ディスク装置だけがミラー保護された場合には、すべての磁気ディスク装置が磁気ディスク装置レベルの保護となり、システムは 1 台の磁気ディスク装置の障害から保護されます。しかしこのような状態では、コントローラー、入出力装置、またはバスの障害が起こった場合には、障害部品の修理または交換が行われるまで、システムは運転できません。システム上のミラー保護された装置は、すべて同一の磁気装置レベルの保護でなければならず、また同一の ASP に存在していなければなりません。1 つの ASP 中の装置は、ミラー保護が開始されると、システムによって自動的に対にされます。

分散リレーショナル・データベースのジャーナル管理

ジャーナル管理を、リレーショナル・データベースおよび索引のバックアップおよび回復の戦略の一つとして使用することができます。

IBM i ジャーナル・サポートでは、監査証跡と正方向および逆方向回復が用意されています。正方向回復は、テーブルの古いバージョンを取り出し、ジャーナルにログ記録された変更をテーブルに適用するのに使用することができます。逆方向回復は、ジャーナルにログ記録された変更をテーブルから除去するのに使用することができます。

コレクションが作成されると、ジャーナルおよびジャーナル・レシーバーと呼ばれるオブジェクトがそのコレクションの中に作成されます。ジャーナル・レシーバーは、テーブルとは別の ASP にあった方が、パフォーマンスが向上します。しかし、コレクションをユーザー ASP 上に置いてしまうと、テーブルとジャーナル、そしてジャーナル・レシーバーは、すべて同じユーザー ASP に置かれることとなります。これでは、高いパフォーマンスは得られません。そこで、別の ASP (このジャーナルのジャーナル・レシーバーのためだけに使用される) に新しいジャーナル・レシーバーを作成し、**ジャーナル変更 (CHGJRN)** コマンドを使用してそれを付加すると、次からは、システムによって生成されるジャーナル・レシーバーが、すべて別のユーザー ASP に置かれるようになり、それによってパフォーマンスの向上を得ることができます。

テーブルは、作成されると、コレクションの中に作成されているジャーナル SQL に自動的にジャーナル処理されます。次に責任を持って、ジャーナル機能を使用して、ジャーナル、ジャーナル・レシーバー、およびジャーナルへのテーブルのジャーナル処理を管理する必要があります。たとえば、テーブルをコレクションの中へ移動する場合には、ジャーナル処理状況に対する自動変更は行われません。テーブルが復元される場合には、通常のジャーナル規則が適用されます。つまり、テーブルが保管される時にジャーナル処理されている場合には、テーブルはそのシステム上に復元した時に同一のジャーナルにジャーナル処理されます。テーブルが保管される時にジャーナル処理されていない場合には、テーブルは復元した時にジャーナル処理されません。ジャーナル機能を使用して、任意のテーブルに対するジャーナル処理を停止することができますが、それを行った場合には、SQL 操作はコミットメント制御下で実行できなくなります。

COMMIT(*NONE) を指定してある場合には、SQL 操作はまだ実行できますが、この場合には、ジャーナル処理およびコミットメント制御の場合と同じレベルの保安全性は提供されません。

ジャーナル処理を活動状態にして、データベースに変更を加えると、その変更は、データベースに加えられの前に、ジャーナル・レシーバーの中でジャーナル処理されます。ジャーナル・レシーバーには、常に最新のデータベース情報が入っています。すべての活動は、変更がどのように行われたかに関係なく、データベース・テーブルについてジャーナル処理されます。

ジャーナル・レシーバー項目は、特定の行 (追加、変更、または削除された行)、あるいはテーブル (オープンされたテーブルまたは保管されたメンバーなど) についての活動を記録します。各項目には、活動の源、ユーザー、ジョブ、プログラム、時刻、および日付を識別する追加の制御情報が含まれています。

システムは、テーブルの移動およびテーブルの名前変更をも含めて、ファイル・レベルの変更のジャーナル処理を行います。システムはまた、物理ファイル・メンバーの初期設定など、メンバー・レベルの変更、および初期プログラム・ロード (IPL) など、システム・レベルの変更のジャーナル処理も行います。ジャーナル・レシーバーに項目を追加して、重要な事象 (ジョブ・ステップが後で再開できるように、ジョブおよびシステムに関する情報をジャーナル処理できるチェックポイントなど) の識別、またはアプリケーションの回復の援助を行うことができます。

単一行に影響する変更の場合は、制御情報に続いて、行イメージが常に含まれます。変更が行われた後の行のイメージが常に含まれます。任意で、変更が加えられる前の行イメージも含めることができます。

物理ファイル・ジャーナル開始 (STRJRNP) コマンドで IMAGES パラメーターを指定し、ジャーナル処理を変更前と変更後の両方の行イメージにするか、変更後の行イメージだけにするかを制御します。

ジャーナル処理データベース・ファイルは、すべて、システムの開始時 (IPL 時) か、独立 ASP がオンに変更されるときに自動的にジャーナルと同期化されます。システムが異常終了した場合や、独立 ASP が異常な状態でオフにされた場合は、データベースに対する変更の一部がジャーナルに入っている場合でも、まだデー

データベースそのものに変更が反映されていない場合があります。このような場合、システムは自動的にジャーナルからデータベースを更新して、テーブルを最新の状態にします。

ジャーナル処理を使用すると、データベース・テーブルの保管が一層容易で迅速なものとなります。たとえば、毎日テーブル全体を保管する代わりに、そのテーブルに対する変更が入っているジャーナル・レシーバーを保管できます。さらにテーブル全体の保管を定期的に行うこともできます。この方法を用いると、日常の保管操作の実行に要する時間を削減できます。

ジャーナル表示 (DSPJRN) コマンドを使用すると、ジャーナル・レシーバー項目をデータベース・ファイルに変換することができます。このようなファイルは、活動報告書、監査証拠、セキュリティ、およびプログラム・デバッグに使用できます。

関連概念:

ジャーナル管理

関連資料:

ジャーナル変更 (CHGJRN) コマンド

ジャーナル表示 (DSPJRN) コマンド

物理ファイル・ジャーナルの開始 (STRJRNP) コマンド

索引の回復:

索引とは、テーブルから行を読み取る順序を記述するものです。索引がジャーナルに記録されていれば、システムは、索引を回復できるため、システムが異常終了した後の IPL 時や、異常な状態でオフにされた独立 ASP をオンに戻すときなどに、多大の時間をかけて索引を再作成しなくて済みます。

テーブルをジャーナル処理すると、テーブルの中の行に対する変更のイメージがジャーナルに書き込まれます。システムまたは独立 ASP が異常終了したときは、これらの行イメージを使用してテーブルが回復されます。ただし、異常終了の後で、システムは、テーブルについて作成された索引がテーブルの中のデータと同期していないことを発見する場合があります。アクセス・パスとそのデータが同期していない場合には、システムでは索引を再作成して、この 2 つが必ず同期して使用可能であるようにしなければなりません。

索引がジャーナル処理されると、システムでは索引のイメージをジャーナルの中に記録して、索引とデータの間認識された同期点を用意します。その情報がジャーナルの中に入っていることによって、システムでは、データと索引の両方を回復し、両方の同期を確保することができます。このような場合には、索引の再作成に要する長い時間を回避することができます。

IBM i オペレーティング・システムには、索引の回復を援助する複数の機能が用意されています。システム上のすべての索引には、その索引のメンテナンスが行われる時期を指定するメンテナンス・オプションがあります。SQL 索引は、*IMMED メインテナンスという属性を指定して作成されます。

電源障害またはサーバーの異常障害が生じた場合には、変更中の索引は、データとの一致を確保するために、再作成が必要になることがあります。サーバー上のすべての索引には、必要な場合に、その索引を再作成する時期を指定する回復オプションがあります。UNIQUE という属性が指定されている SQL 索引は、すべて *IPL という回復属性を指定して作成され、これによって、これらの索引は IBM i ライセンス・プログラムが開始される間に再作成されることを意味します。一方、こうしたオプションが指定されていない SQL 索引は、すべて *AFTIPL 回復属性で作成されることとなります。この属性を持つ索引は、オペレーティング・システムが開始された後、あるいは独立 ASP がオンにされた後に再作成されます。IPL 時や、独立 ASP をオンにするときには、画面で、再作成に必要な索引とそれらの回復オプションを確認できます。これらの回復オプションは、指定変更することが可能です。

SQL 索引は自動的にジャーナル処理されません。アクセス・パス・ジャーナル開始 (STRJRNAP) コマンドを使用して、SQL 操作で作成された索引をジャーナル処理することができます。システム保管および復元機能を使用すれば、オブジェクト保管 (SAVOBJ) またはライブラリー保管 (SAVLIB) コマンドで ACCPTH (*YES) を使用することによって、テーブルを保管する時に、索引を保管することができます。テーブルを復元しなければならない場合でも、索引を再作成する必要はありません。前に保管および復元されていない索引は、いずれもデータベースによって自動的にかつ非同期的に再作成されます。

索引のジャーナル処理の前に、その索引に対応するテーブルについて、ジャーナル処理を開始しなければなりません。さらに、索引およびそれに対応するテーブルには、同一のジャーナルを使用しなければなりません。

索引ジャーナル処理は、追加の出力操作を最小限に抑制するように設計されています。たとえば、システムは、変更された行および変更された索引のジャーナル・データを、同一の出力操作で書き出します。ただし、索引のジャーナル処理を開始するにあたっては、ユーザー ASP 中のジャーナル・レシーバーを分離することを特に考慮しなければなりません。ジャーナル・レシーバーをそれ自体の ASP に入れば、最高のジャーナル管理のパフォーマンスが得られると同時に、ジャーナル・レシーバーをディスク障害から保護する上でも役立ちます。

関連資料:

アクセス・パス・ジャーナルの開始 (STRJRNAP) コマンド

オブジェクト保管 (SAVOBJ) コマンド

ライブラリー保管 (SAVLIB) コマンド

索引再ビルド時間を短縮するためのテーブルの設計:

テーブルの設計しだいでは、索引回復時間の短縮に役立つ場合があります。

大規模のマスター・テーブルをヒストリー・テーブルとトランザクション・テーブルに分割することができます。そうすれば、トランザクション・テーブルは新しいデータを追加するのに使用し、ヒストリー・テーブルは照会専用として使用することができます。毎日、トランザクション・データをヒストリー・テーブルに組み合わせてから、翌日のデータに備えてトランザクション・ファイルを消去することができます。このような設計を採用すれば、その日にシステムが異常終了した場合でも、比較的小規模のトランザクション・テーブルの索引を再作成するだけで済むために、索引を再作成する時間を短縮することができます。ただし、大規模なヒストリー・テーブルの索引については、1 日の大半が読み取り専用となるため、データとの同期がずれることはほとんどなく、再作成する必要はないといえます。

テーブル設計によって索引の再作成時間の短縮を図る場合と、アクセス・パス・ジャーナル処理など、システム提供機能を使用する場合でどちらが有利であるかを考慮してください。前段のテーブル設計では、より複雑なアプリケーションの設計が必要となる場合があります。実際の状況を考慮した後、より複雑なアプリケーションを設計するよりも、アクセス・パスのジャーナル処理のようなシステムが提供する機能を使うこともできます。

システム管理アクセス・パス保護:

システム管理アクセス・パス保護 (SMAPP) は、アクセス・パスの自動保護を行います。

SMAPP サポートを使うと、アクセス・パス・ジャーナルの開始 (STRJRNAP) コマンド などのジャーナル処理コマンドを使用しなくても、アクセス・パスのジャーナル処理の利点が得られます。SMAPP サポートでは、システムの再始動時や、独立 ASP をオンに変更するときにアクセス・パスを再作成するのではなく、システムが異常終了した後にアクセス・パスを回復します。

SMAPP サポートは、出荷時にオンになっています。

システムは、ユーザーが定めるターゲットのアクセス・パス回復回数に基づいて、またはシステムで指定されたデフォルトの時間により、どのアクセス・パスを保護するかを判別します。ターゲットのアクセス・パス回復回数は、システム全体の値として指定するか、または ASP を基礎として指定できます。ユーザー定義のジャーナルにジャーナルがとられるアクセス・パスは、SMAPP 保護の対象とはなりません。それらのアクセス・パスはすでに保護されているからです。

関連概念:

システム管理アクセス・パス保護

関連資料:

アクセス・パス・ジャーナルの開始 (STRJRNAP) コマンド

コミットメント制御によるトランザクションの回復

コミットメント制御は、IBM i ジャーナル管理機能を拡張したものです。システムは、リレーショナル・データベース変更のグループを単一の作業単位 (トランザクション) として識別し処理することができます。

SQL COMMIT ステートメントは、操作のグループが完了することを保証します。SQL ROLLBACK ステートメントは、操作のグループが取り消されることを保証します。コミットまたはロールバックすることができない SQL ステートメントは、次に挙げるものだけです。

- DROP COLLECTION
- GRANT または REVOKE (ただし、指定されたオブジェクトに権限保有者が存在している場合)

コミットメント制御の下では、トランザクション中に使用されているテーブルおよび行は、他のジョブからロックされます。したがって、そのトランザクションが完了するまで、他のジョブではそのデータを使用することができません。トランザクションが終了すると、プログラムは SQL COMMIT または ROLLBACK ステートメントを出して、行を解放します。コミット操作が実行される前に、システムまたはジョブが異常終了した場合には、最後にコミットまたはロールバック操作が行われた時以後にそのジョブに加えられたすべての変更がロールバックされます (元に戻される)。影響を受けた行でまだロックされているものがある場合は、そこでアンロックされます。ロック・レベルは、次のようになっています。

*NONE

コミットメント制御は使用されない。他のジョブのコミットされていない変更は、見ることができない。

***CHG** SQL ALTER、COMMENT ON、CREATE、DROP、GRANT、LABEL ON、および REVOKE ステートメントの中で参照されているオブジェクト、および更新、削除、挿入が行われた行は、その作業単位 (トランザクション) が完了するまで、ロックされる。他のジョブのコミットされていない変更は、見ることができる。

***CS** SQL ALTER、COMMENT ON、CREATE、DROP、GRANT、LABEL ON、および REVOKE ステートメントの中で参照されているオブジェクト、および更新、削除、挿入が行われた行は、その作業単位 (トランザクション) が完了するまで、ロックされる。選択されても更新されなかった行は、次の行が選択されるまで、ロックされる。他のジョブのコミットされていない変更は、見ることができない。

***ALL** SQL ALTER、COMMENT ON、CREATE、DROP、GRANT、LABEL ON、および REVOKE ステートメントの中で参照されているオブジェクト、および読み取り、更新、削除、挿入が行われた行は、その作業単位 (トランザクション) が完了するまで、ロックされる。他のジョブのコミットされていない変更は、見ることができない。

323 ページの表 30 には、上記のロック・レベルのそれぞれについて、レコード・ロック期間が示してあります。

プログラムのプリコンパイル時、または対話式 SQL の開始時に、COMMIT (*CHG)、COMMIT (*CS)、または COMMIT (*ALL) を要求した場合には、SQL は、**コミットメント制御開始 (STRCMTCTL)** コマンドを暗黙的に呼び出すことによって、コミットメント制御環境をセットアップします。SQL がコミットメント制御を開始する時に指定される LCKLVL パラメーターは、CRTSQLxxx コマンドの COMMIT パラメーターで指定されるロック・レベルです。SQL がコミットメント制御を開始する時には、NFYOBJ(*NONE) が指定されます。別の NFYOBJ パラメーターを指定する場合には、SQL を開始する前に STRCMTCTL コマンドを出してください。

注: コミットメント制御を用いて実行する時には、アプリケーション・プログラムの中でデータ処理言語ステートメントによって参照されているテーブルは、ジャーナル処理しなければなりません。そのようなテーブルは、プリコンパイル時にジャーナル処理する必要はありませんが、アプリケーションの実行時にはジャーナル処理しなければなりません。

リモート・リレーショナル・データベースがシステム上のデータにアクセスし、コミットメント・レベル回復可能読み取り (*RR) を要求している場合には、照会がクローズされるまで、テーブルはロックされません。カーソルが読み取り専用である場合には、テーブルは (*SHRNUP) ロックされます。カーソルが更新モードである場合には、テーブルは (*EXCLRD) ロックされます。

SQL コレクションの中に作成されたジャーナルは、通常、SQL テーブルに対するすべての変更のログを記録するのに使用されるジャーナルです。ただし、システム・ジャーナル機能を使用すれば、SQL テーブルを別のジャーナルにジャーナル処理することができます。

コミットメント制御では、最高 500,000,000 個の行変更を 1 つの作業単位で処理することができます。COMMIT(*ALL) が指定されている場合には、読み取られた行もすべて 500,000,000 の限度内に含まれます。(1 つの作業単位内である 1 行が複数回、変更または読み取られた場合、500,000,000 の限度に対しては 1 回としてカウントされます。) ロックが多数維持されていると、システム・パフォーマンスに悪影響を及ぼします。また、作業単位が完了するまで、他のユーザーはその作業単位でロックされている行にアクセスすることができません。したがって、1 つの作業単位で処理される行の数を少ない数に抑える方が、効率的にはよいと言えます。

COMMIT および ROLLBACK ステートメントの HOLD 値を使用すると、カーソルをオープンにしておき、OPEN ステートメントを再度出さなくても別の作業単位を開始することができます。プログラムに解放されていない IBM i 以外の接続があって、呼び出しスタックにまだ SQL が入っている場合は、HOLD 値を使用できません。プログラムのプリコンパイル時に、ALWBK(*ALLREAD) および COMMIT(*CHG) と COMMIT(*CS) のいずれかが指定された場合には、すべての読み取り専用カーソルで列のブロック化が可能になり、ROLLBACK HOLD ステートメントではカーソル位置をロールバックしません。

ロックされた行 (レコード) が SQL プリコンパイル・プログラム、または対話式 SQL セッションの実行で保留になっている場合には、サーバーの「コマンド入力」画面で COMMIT または ROLLBACK ステートメントを出すことができます。それ以外の場合には、暗黙の ROLLBACK 操作がジョブの終了時に行われます。

WRKMTDFN (コミットメント定義の処理) コマンドを使用すれば、コミットメント定義の状況を監視し、システム間のコミットメント制御活動にかかわっているロックと保持リソースの解除を行うことができます。

表 30. レコード・ロック期間

SQL ステートメント	COMMIT パラメータ	レコード・ロックの期間	ロック・タイプ
SELECT INTO	*NONE *CHG *CS *ALL (注 2 参照)	ロックなし ロックなし 読み取りおよび解放時に行のロック 読み取りから ROLLBACK または COMMIT まで	READ READ
FETCH (読み取り専用カーソル)	*NONE *CHG *CS *ALL (注 2 参照)	ロックなし ロックなし 読み取りから次の FETCH まで 読み取りから ROLLBACK または COMMIT まで	READ READ
FETCH (更新または削除可能カーソル) 注 1 参照	*NONE *CHG *CS *ALL	レコードが更新または削除されなかった場合、読み取りから次の FETCH まで レコードが更新または削除された場合、読み取りから UPDATE または DELETE まで レコードが更新または削除されなかった場合、読み取りから次の FETCH まで レコードが更新または削除された場合、読み取りから UPDATE または DELETE まで レコードが更新または削除されなかった場合、読み取りから次の FETCH まで レコードが更新または削除された場合、読み取りから UPDATE または DELETE まで 読み取りから ROLLBACK または COMMIT まで	UPDATE UPDATE UPDATE UPDATE ³
INSERT (ターゲット・テーブル)	*NONE *CHG *CS *ALL	ロックなし 挿入から ROLLBACK または COMMIT まで 挿入から ROLLBACK または COMMIT まで 挿入から ROLLBACK または COMMIT まで	UPDATE UPDATE UPDATE ⁴
INSERT (副選択の中のテーブル)	*NONE *CHG *CS *ALL	ロックなし ロックなし 読み取りの間、各レコードをロック 読み取りから ROLLBACK または COMMIT まで	READ READ
UPDATE (非カーソル)	*NONE *CHG *CS *ALL	更新の間、各レコードをロック 読み取りから ROLLBACK または COMMIT まで 読み取りから ROLLBACK または COMMIT まで 読み取りから ROLLBACK または COMMIT まで	UPDATE UPDATE UPDATE UPDATE
DELETE (非カーソル)	*NONE *CHG *CS *ALL	削除の間、各レコードをロック 読み取りから ROLLBACK または COMMIT まで 読み取りから ROLLBACK または COMMIT まで 読み取りから ROLLBACK または COMMIT まで	UPDATE UPDATE UPDATE UPDATE

表 30. レコード・ロック期間 (続き)

SQL ステートメント	COMMIT パラメータ	レコード・ロックの期間	ロック・タイプ
UPDATE (カーソル付き)	*NONE *CHG *CS *ALL	レコード更新時にロック解放 読み取りから ROLLBACK または COMMIT まで 読み取りから ROLLBACK または COMMIT まで 読み取りから ROLLBACK または COMMIT まで	UPDATE UPDATE UPDATE UPDATE
DELETE (カーソル付き)	*NONE *CHG *CS *ALL	レコード削除時にロック解放 読み取りから ROLLBACK または COMMIT まで 読み取りから ROLLBACK または COMMIT まで 読み取りから ROLLBACK または COMMIT まで	UPDATE UPDATE UPDATE UPDATE
副照会 (更新または削除可能カーソルまたは UPDATE または DELETE 非カーソル)	*NONE *CHG *CS *ALL (注 2 参照)	読み取りから次の FETCH まで 読み取りから次の FETCH まで 読み取りから次の FETCH まで 読み取りから ROLLBACK または COMMIT まで	READ READ READ READ
副照会 (読み取り専用カーソルまたは SELECT INTO)	*NONE *CHG *CS *ALL	ロックなし ロックなし 読み取りの間、各レコードをロック 読み取りから ROLLBACK または COMMIT まで	READ READ

注:

- 結果テーブルが読み取り専用でない (DECLARE CURSOR の説明を参照) 場合で、しかも次のいずれか 1 つの項目に該当する場合には、カーソルは UPDATE または DELETE 機能についてオープンされます。
 - カーソルが FOR UPDATE 文節で定義されている。
 - カーソルが FOR UPDATE、FOR FETCH ONLY、または ORDER BY 文節なしで定義され、プログラムに少なくとも次の 1 つの項目が含まれている。
 - 同じカーソル名を参照するカーソル UPDATE
 - 同じカーソル名を参照するカーソル DELETE
 - CRTSQLxxx コマンドで ALWBLK(*READ) または ALWBLK(*NONE) を指定した EXECUTE または EXECUTE IMMEDIATE ステートメント
- テーブルまたはビューは、COMMIT(*ALL) を充足するために、排他ロックすることができます。グループ化または合併が組み込まれている副選択が処理される場合、または照会の処理が一時的な結果の使用を必要とする場合には、コミットされていない変更の表示から保護するために、排他ロックが獲得されます。
- 行が更新または削除されない場合には、ロックは *READ になります。
- ターゲット・テーブルの行に対しては UPDATE ロックで、副選択テーブルの行に対しては READ ロックです。
- テーブルまたはビューは、反復可能読み取りを充足するために、排他ロックすることができます。行のロックは、反復可能読み取りのもとでも行われます。獲得されるロックおよびその期間は、*ALL と同じです。

関連概念:

トランザクションおよびコミットメント制御のトラブルシューティング

関連タスク:

270 ページの『分散リレーショナル・データベースにおけるコミットメント定義の処理』システム上でコミットメント定義を処理する場合は、**WRKCMDFN (コミットメント定義の処理)** コマンドを使用してください。

関連資料:

DECLARE CURSOR

コミットメント制御開始 (STRCMTCTL) コマンド

コミットメント定義の処理 (WRKCMTDFN) コマンド

『分散リレーショナル・データベースの処理の保管と復元』

データおよびプログラムの保管および復元を使用すれば、プログラム障害またはシステム障害からの回復、システム間での情報の交換、またはオフラインでのオブジェクトまたはデータの記憶が可能です。包括的なバックアップ方針が分散リレーショナル・データベース・ネットワークの各システムで確立されていれば、問題が発生した場合にも、確実にシステムの回復を図って、ネットワーク・ユーザーのために迅速に使用可能にすることができます。

オブジェクト保管 (SAVOBJ) コマンド

ライブラリー保管 (SAVLIB) コマンド

変更されたオブジェクトの保管 (SAVCHGOBJ) コマンド

保管ファイル・データの保管 (SAVSAVFDTA) コマンド

システム保管 (SAVSYS) コマンド

ライブラリー復元 (RSTLIB) コマンド

オブジェクト復元 (RSTOBJ) コマンド

ユーザー・プロファイル復元 (RSTUSRPRF) コマンド

権限復元 (RSTAUT) コマンド

構成復元 (RSTCFG) コマンド

分散リレーショナル・データベースの処理の保管と復元

データおよびプログラムの保管および復元を使用すれば、プログラム障害またはシステム障害からの回復、システム間での情報の交換、またはオフラインでのオブジェクトまたはデータの記憶が可能です。包括的なバックアップ方針が分散リレーショナル・データベース・ネットワークの各システムで確立されていれば、問題が発生した場合にも、確実にシステムの回復を図って、ネットワーク・ユーザーのために迅速に使用可能にすることができます。

システムをテープなど、外部メディアに保管すれば、システム・プログラムおよびデータを火災や洪水などの災害から保護することができます。しかし、情報は、保管ファイルと呼ばれるディスク・ファイルに保管することもできます。保管ファイルは、ディスクに常駐しているファイルで、データが入出力操作で使用されたり、通信回線によって別の IBM i オペレーティング・システムへ転送される時まで、データを保管するために使用されるものです。保管ファイルを使用すると、オペレーターがテープをロードする必要がないので、オペレーターがいなくても保管操作が可能になります。分散リレーショナル・データベースでは、保管ファイルは、保護方法と 1 つとして別のシステムに送ることができます。

情報の復元時には、情報はテープ、または保管ファイルから補助記憶域へ書き込まれるので、システム・ユーザーはそこに書き込まれた情報にアクセスすることができます。

IBM i オペレーティング・システムには、データベース・テーブルと SQL オブジェクトを保管および復元するためのコマンドの完全なセットが備わっています。

- **ライブラリーの保管 (SAVLIB)** コマンドは、1 つ以上のコレクションを保管します。
- **オブジェクト保管 (SAVOBJ)** コマンドでは、SQL テーブル、表示、および索引など、1 つまたは複数のオブジェクトを保管します。

- **変更されたオブジェクトの保管 (SAVCHGOBJ)** コマンドは、コレクションが最後に保管された時点以降、または指定された日付以後に変更されたオブジェクトを保管します。
- **保管ファイル・データの保管 (SAVSAVFDTA)** コマンドでは、保管ファイルの内容を保管します。
- **システム保管 (SAVSYS)** コマンドでは、オペレーティング・システム、セキュリティー情報、入出力装置構成、およびシステム値を保管します。
- **ライブラリー復元 (RSTLIB)** コマンドは、コレクションを復元します。
- **オブジェクト復元 (RSTOBJ)** コマンドでは、SQL テーブル、ビュー、および索引など、1 つまたは複数のオブジェクトを復元します。
- **ユーザー・プロファイル復元 (RSTUSRPRF)、権限復元 (RSTAUT)、および構成復元 (RSTCFG)** コマンドでは、**システム保管 (SAVSYS)** コマンドによって保管されたユーザー・プロファイル、権限、および構成を復元します。

関連概念:

トランザクションおよびコミットメント制御のトラブルシューティング

関連タスク:

270 ページの『分散リレーショナル・データベースにおけるコミットメント定義の処理』システム上でコミットメント定義を処理する場合は、**WRKCMTDFN (コミットメント定義の処理)** コマンドを使用してください。

関連資料:

321 ページの『コミットメント制御によるトランザクションの回復』コミットメント制御は、IBM i ジャーナル管理機能を拡張したものです。システムは、リレーショナル・データベース変更のグループを単一の作業単位 (トランザクション) として識別し処理することができます。

DECLARE CURSOR

コミットメント制御開始 (STRCMTCTL) コマンド
 コミットメント定義の処理 (WRKCMTDFN) コマンド
 オブジェクト保管 (SAVOBJ) コマンド
 ライブラリー保管 (SAVLIB) コマンド
 変更されたオブジェクトの保管 (SAVCHGOBJ) コマンド
 保管ファイル・データの保管 (SAVSAVFDTA) コマンド
 システム保管 (SAVSYS) コマンド
 ライブラリー復元 (RSTLIB) コマンド
 オブジェクト復元 (RSTOBJ) コマンド
 ユーザー・プロファイル復元 (RSTUSRPRF) コマンド
 権限復元 (RSTAUT) コマンド
 構成復元 (RSTCFG) コマンド

分散リレーショナル・データベース環境での索引の保管と復元:

SQL 索引については、復元の方が再作成より高速で行うことができます。所要時間は多くの要因に左右されますが、データベース索引の再作成には、10 000 行当たり約 1 分を要します。

索引を復元した後では、最新のジャーナル変更を適用することによって、テーブルを更新する必要がある場合があります (ジャーナル処理が活動状態にあるかどうかによります)。この余分な回復時間を考慮に入れても、索引については、再作成よりも復元の方が速いかもしれません。

システムは、索引の保全性を確保します。システムが索引について使用不能であると判別した場合には、システムはその回復を試みます。索引をいつ回復するかは、制御することができます。システムが異常終了した場合には、次の IPL 時に、システムは、索引またはビューの回復を必要とするテーブルを自動的にリストします。索引を再作成するか、または次のいずれかの時点で索引の回復を試みるかを定めることができます。

- IPL 中
- IPL の後
- テーブルを初めて使用する時

関連概念:

バックアップおよび回復

分散リレーショナル・データベース環境でのセキュリティ情報の保管と復元:

各種の CL コマンドを使用して、セキュリティ情報を保管または復元できます。

ユーザー・プロファイルを更新したり、分散リレーショナル・データベース・ネットワーク内のユーザーの権限を更新するなど、システム・セキュリティ環境に頻繁に変更を加える場合は、完全な**システム保管 (SAVSYS)** コマンドを実行したり、専用システムを使用して実行に時間のかかる処理を行わなくても、セキュリティ情報をメディアや保管ファイルに保管できます。**セキュリティ・データの保管 (SAVSECDTA)** コマンドを使用すれば、専用システムを使用しなくても、比較的短時間でセキュリティ・データを保管できます。**SAVSECDTA** コマンドを使用して保管したデータは、**ユーザー・プロファイル復元 (RSTUSRPRF)** コマンドや**権限復元 (RSTAUT)** コマンドで復元できます。

SAVSECDTA コマンドや **RSTUSRPRF** コマンドで保管および復元できるセキュリティ情報に含まれるのは、DRDA TCP/IP サポートがリモート・システム・ユーザー ID およびパスワード情報を保管および検索するために使用するサーバー認証項目です。

関連資料:

システム保管 (SAVSYS) コマンド

セキュリティ・データの保管 (SAVSECDTA) コマンド

ユーザー・プロファイル復元 (RSTUSRPRF) コマンド

権限復元 (RSTAUT) コマンド

分散リレーショナル・データベース環境での SQL パッケージの保管と復元:

リモート・システム上のリレーショナル・データベースを参照するアプリケーション・プログラムがプリコンパイルまたはバインドされると、アプリケーションの中の SQL ステートメントを処理するのに必要な制御構造を入れるために、SQL パッケージがサーバー上に作成されます。

SQL パッケージは、IBM i オブジェクトなので、オブジェクト保管 (SAVOBJ) コマンドを使用してメディアまたは保管ファイルに保管し、オブジェクト復元 (RSTOBJ) コマンドを使用して復元することができます。

SQL パッケージは、保管された元のコレクションと同じ名前を持つコレクションに復元される必要があります。SQL パッケージは名前を変更できません。

関連資料:

オブジェクト復元 (RSTOBJ) コマンド

オブジェクト保管 (SAVOBJ) コマンド

リレーショナル・データベースのディレクトリーの保管と復元:

リレーショナル・データベース・ディレクトリーは IBM i オブジェクトではありません。その代わりに、初期プログラム・ロード (IPL) 時にシステムによって開かれたファイルから構成されます。

このため、**オブジェクト保管 (SAVOBJ)** コマンドは、これらのファイルを直接保管するためには使用できません。リレーショナル・データベース・ディレクトリー・データから出力ファイルを作成することによって、リレーショナル・データベース・ディレクトリーを保管することができます。そうすれば、ディレクトリーが損傷した場合には、この出力ファイルを使用して、ディレクトリーに項目を追加することができます。

項目が追加され、リレーショナル・データベース・ディレクトリーを保管するときは、**リレーショナル・データベース・ディレクトリー項目の表示 (DSPRBDIRE)** コマンドで **OUTFILE** パラメーターを指定して、コマンドの結果を出力ファイルに送ります。この出力ファイルは、テープまたは保管ファイルに保管し、システムに復元することができます。リレーショナル・データベース・ディレクトリーが損傷し、システムが回復を必要とする場合には、制御言語 (CL) プログラムを使用して、リレーショナル・データベース項目データが入っている出力ファイルを復元することができます。CL プログラムが復元された出力ファイルからデータを読み取り、新しいリレーショナル・データベース・ディレクトリーに項目を追加する CL コマンドを作成します。

たとえば、Spiffy 社の MP000 システムのリレーショナル・データベース・ディレクトリーは、次のようにして **RDBDIRM** という名前の出力ファイルに送られます。

```
DSPRBDIRE OUTPUT(*OUTFILE) OUTFILE(RDBDIRM)
```

次に挙げる CL プログラム例では、出力ファイル **RDBDIRM** の内容を読み取り、**RDB ディレクトリー項目の追加 (ADDRBDIRE)** コマンドを使用して、**RDB ディレクトリー項目**を再作成しています。なお、古いディレクトリー項目は、新しいディレクトリー項目が作成される前に除去されます。

注: コードの例を使用することは、443 ページの『コードに関するライセンス情報および特記事項』の条件に同意したものとされます。

```
/* - Restore RDB Entries from output file created with: - */
/* - DSPRBDIRE OUTPUT(*OUTFILE) OUTFILE(RDBDIRM) - */
/* - FROM A V4R2 OR LATER LEVEL OF OS/400 or IBM i - */
/*****/
PGM PARM(&ACTIVATE)
DCL VAR(&ACTIVATE) TYPE(*CHAR) LEN(7)

/* Declare Entry Types Variables to Compare with &RWTYPE */
DCL &LOCAL *CHAR 1
DCL &SNA *CHAR 1
DCL &IP *CHAR 1
DCL &ARD *CHAR 1
DCL &ARDSNA *CHAR 1
DCL &ARDIP *CHAR 1
DCL &RWTYPE *CHAR 1
DCL &RWRDB *CHAR 18
DCL &RWRLOC *CHAR 8
DCL &RWTEXT *CHAR 50
DCL &RWDEV *CHAR 10
DCL &RWLLOC *CHAR 8
DCL &RWNTID *CHAR 8
DCL &RWMODE *CHAR 8
DCL &RWTPN *CHAR 8
DCL &RWSLOC *CHAR 254
DCL &RWPORT *CHAR 14
DCL &RWDPGM *CHAR 10
```

```

DCL &RWDLIB  *CHAR 10

DCLF FILE(RDBSAV/RDBDIRM) /* SEE PROLOG CONCERNING THIS */
IF COND(&ACTIVATE = SAVE) THEN(GOTO CMBLBL(SAVE))
IF COND(&ACTIVATE = RESTORE) THEN(GOTO CMDLBL(RESTORE))
SAVE:
CRTLIB RDBSAV
DSPRDBDIRE OUTPUT(*OUTFILE) OUTFILE(RDBSAV/RDBDIRM)
GOTO CMDLBL(END)

RESTORE:
/* Initialize Entry Type Variables to Assigned Values */
CHGVAR &LOCAL  '0' /* Local RDB (one per system) */
CHGVAR &SNA    '1' /* APPC entry (no ARD pgm) */
CHGVAR &IP     '2' /* TCP/IP entry (no ARD pgm) */
CHGVAR &ARD    '3' /* ARD pgm w/o comm parms */
CHGVAR &ARDSNA '4' /* ARD pgm with APPC parms */
CHGVAR &ARDIP  '5' /* ARD pgm with TCP/IP parms */

RMVRDBDIRE RDB(*ALL) /* Clear out directory */

NEXTENT: /* Start of processing loop */
RCVF /* Get a directory entry */
MONMSG MSGID(CPF0864) EXEC(DO) /* End of file processing */
  QSYS/RCVMSG PGMQ(*SAME (*)) MSGTYPE(*EXCP) RMV(*YES) MSGQ(*PGMQ)
  GOTO CMDLBL(LASTENT)
ENDDO

/* Process entry based on type code */
IF (&RWTYPE = &LOCAL) THEN( +
  QSYS/ADDRDBDIRE RDB(&RWRDB) RMTLOCNAME(&RWRLC) TEXT(&RWTEXT) )

ELSE IF (&RWTYPE = &SNA) THEN( +
  QSYS/ADDRDBDIRE RDB(&RWRDB) RMTLOCNAME(&RWRLC) TEXT(&RWTEXT) +
  DEV(&RWDEV) LCLLOCNAME(&RWLLC) +
  RMTNETID(&RWNTID) MODE(&RWMODE) TNSPGM(&RWTPN) )

ELSE IF (&RWTYPE = &IP) THEN( +
  QSYS/ADDRDBDIRE RDB(&RWRDB) RMTLOCNAME(&RWSLOC *IP) +
  TEXT(&RWTEXT) PORT(&RWPORT) )

ELSE IF (&RWTYPE = &ARD) THEN( +
  QSYS/ADDRDBDIRE RDB(&RWRDB) RMTLOCNAME(&RWRLC) TEXT(&RWTEXT) +
  ARDPGM(&RWDLIB/&RWDPGM) )

ELSE IF (&RWTYPE = &ARDSNA) THEN( +
  QSYS/ADDRDBDIRE RDB(&RWRDB) RMTLOCNAME(&RWRLC) TEXT(&RWTEXT) +
  DEV(&RWDEV) LCLLOCNAME(&RWLLC) +
  RMTNETID(&RWNTID) MODE(&RWMODE) TNSPGM(&RWTPN) +
  ARDPGM(&RWDLIB/&RWDPGM) )

ELSE IF (&RWTYPE = &ARDIP) THEN( +
  QSYS/ADDRDBDIRE RDB(&RWRDB) RMTLOCNAME(&RWSLOC *IP) +
  TEXT(&RWTEXT) PORT(&RWPORT) +
  ARDPGM(&RWDLIB/&RWDPGM) )

GOTO CMDLBL(NEXTENT)

LASTENT:
RETURN
DLTLIB RDBSAV
END

ENDPGM

```

前述のようなタイプの出力ファイルが利用できない時には、ディレクトリー復元の次のような代替手段があります。これは、保管されているシステムからオブジェクトを抽出して、それを他の何らかのライブラリーに復元し、**RDB ディレクトリー項目の追加 (ADDRDBDIRE)** コマンドを使用して手動で項目を入力する、という方法です。

システム保管 (SAVSYS) コマンドでは、リレーショナル・データベース・ディレクトリーを構成しているファイルが保管されます。リレーショナル・データベース・ディレクトリーが入っている物理ファイルは、下記の**オブジェクト復元 (RSTOBJ)** コマンドによって保管メディアからライブラリーに復元することができます。

```
RSTOBJ      OBJ(QADBXRDBD)  SAVLIB(QSYS)
            DEV(TAP01)  OBJTYPE(*FILE)
            LABEL(Qppppppvrmxx0003)
            RSTLIB(your lib)
```

この例では、リレーショナル・データベース・ディレクトリーはテープから復元されます。LABEL パラメーターの中の文字 *ppppppp* は、IBM i の製品番号 (例えば、バージョン 5 リリース 3 の場合には 5722SS1) です。LABEL パラメーターの中の *vrm* は、IBM i のバージョン、リリース、およびモディフィケーション・レベルです。また、LABEL パラメーターの中の *xx* は、現行のシステム言語の値の最後の 2 桁を参照します。たとえば、英語の場合は 2924 です。したがって、*xx* の値は 24 です。

このファイルをライブラリーに復元したら、そのファイルの情報を使用して、手動でリレーショナル・データベース・ディレクトリーを再作成できます。

関連資料:

69 ページの『例: リレーショナル・データベースのディレクトリーのセットアップ』

Spiffy 社のネットワークが示す例を見れば、分散リレーショナル・データベース・ネットワークの中のシステムでリレーショナル・データベース・ディレクトリーをセットアップして使用方法が分かります。

リレーショナル・データベース・ディレクトリー項目の追加 (ADDRDBDIRE) コマンド

リレーショナル・データベース・ディレクトリー項目の表示 (DSPRDBDIRE) コマンド

オブジェクト復元 (RSTOBJ) コマンド

オブジェクト保管 (SAVOBJ) コマンド

システム保管 (SAVSYS) コマンド

分散リレーショナル・データベースのネットワーク冗長性の考慮事項

ネットワーク冗長性によって、分散リレーショナル・データベース・ネットワーク上のユーザーは、そのネットワーク内のリレーショナル・データベースにアクセスする手段を他にも得られることになります。

クライアントからサーバーに至る通信経路が 1 つしかない場合は、その通信回線が故障すると、クライアント側のユーザーはサーバー・リレーショナル・データベースにアクセスできなくなります。したがって、Spiffy 社の分散リレーショナル・データベース管理者にとっても、ネットワーク冗長性は重要な問題となります。たとえば、ある販売店の保守サービス受注または得意先部品購入の問題を考えてみましょう。ある得意先が保守サービス待機中または部品購入待ちである場合、サービス担当員には、作業の予定を立てたり部品を販売したりするために、企業情報に関して許可されているすべてのテーブルへのアクセス権が必要です。

ローカル・システムが故障している場合には、作業は行えません。ローカル・システムが作動していても、作業を処理するためにはリモート・システムに対する要求が必要であり、そのリモート・システムが故障している場合には、その要求は処理できません。Spiffy 社の例で言えば、これはある販売店が地域在庫センターに部品情報を要求することができないような場合に該当します。また、多くのクライアント・ジョブを

処理しているサーバーが故障した場合には、それらのクライアントのいずれも要求を完了することができません。Spiffy 社のネットワークの例では、ある地域センターが故障している場合には、その支援下にあるサーバーはどれも部品を発注することができません。

地域の販売店に対して地域在庫データへのアクセスを提供することは、Spiffy 社の分散リレーショナル・データベース管理担当者にとって重要なことです。ネットワーク内でデータに至る経路の提供は、複数の方法が可能です。Spiffy 社の元々のネットワーク構成では、エンド・ノード販売店をそれぞれのネットワーク・ノード地域センターにリンクしていました。

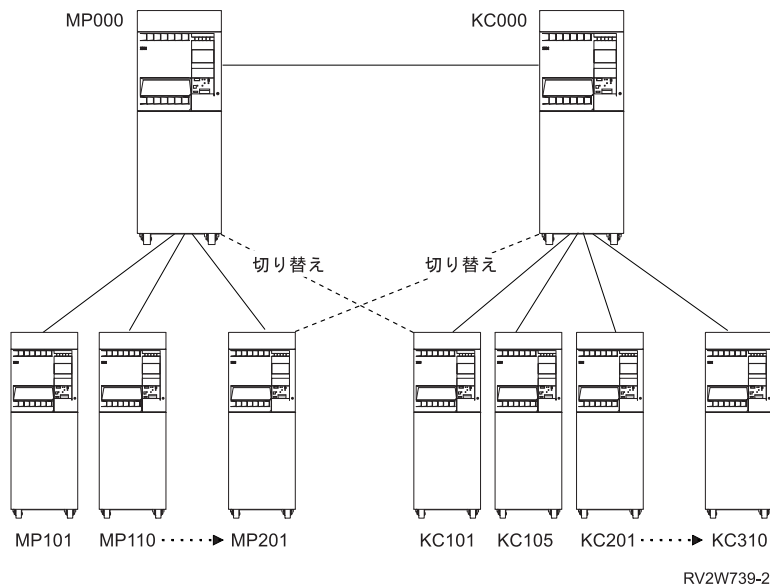


図 29. 代替ネットワーク経路

販売店によっては、別の地域センターとの交換回線接続という代替方法を取ることができます。ローカル地域センターがネットワークで使用不能になった場合には、別のサーバーにアクセスすることによって、要求側の販売店では、作業に必要な情報を入手することができます。最初の図では、MP000 システムからサービスを提供されている一部の販売店 (クライアント) が KC000 システムへのリンクを確立し、MP000 システムが使用不能になった場合には利用できるようにしています。システム・オペレーターや分散リレーショナル・データベース管理者は、構成変更 (VRYCFG) コマンドや構成状況処理 (WRKCFGSTS) コマンドを使用して、必要なときに回線をオンにし、1 次サーバーが使用可能になったら回線をオフにすることができます。

区域内で比較的大規模な販売店の 1 つが他の販売店に対するサーバーとしても機能するならば、これもやはり別の代替方法になり得ます。2 番目の図に示すように、1 つのエンド・ノードは、他のエンド・ノードに対しては、ネットワーク・ノードを介したサーバーに過ぎなくなります。最初の図では、ミネアポリスとのリンクが故障した場合には、販売店 (エンド・ノード) に在庫を照会できません。上に図示した構成を変更して、販売店の 1 つを APPN ネットワーク・ノードとして構成し、他地区の販売店からその販売店に至る回線を設けることができます。

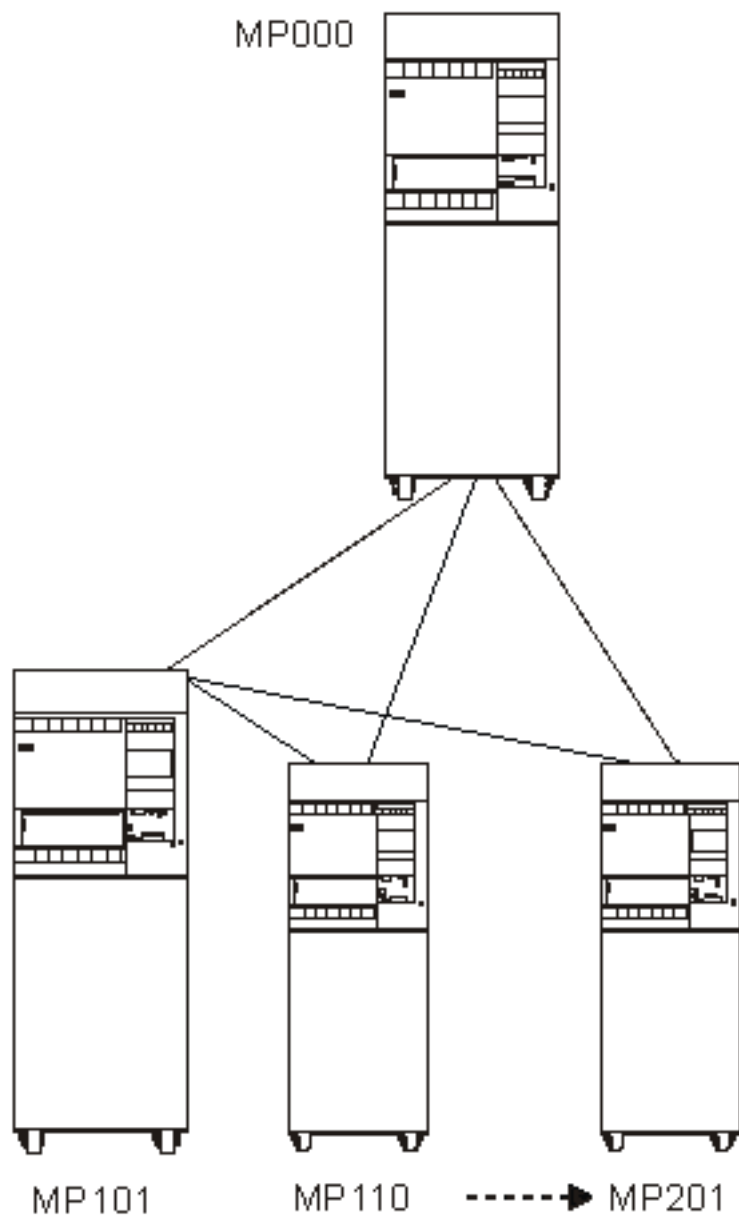


図 30. 代替サーバー

関連資料:

構成変更 (VRYCFG) コマンド

構成状況処理 (WRKCFGSTS) コマンド

分散リレーショナル・データベース・ネットワーク内のデータの冗長性

分散リレーショナル・データベースのデータの冗長性によっても、分散リレーショナル・データベース・ネットワーク上のユーザーは、そのネットワーク内のリレーショナル・データベースにアクセスする手段を他に得られることとなります。

データの冗長性についての方針を作成するために、分散リレーショナル・データベース管理者が検討する考慮事項は、データに至る通信経路の可用性を確保する場合に比べて複雑です。

ネットワーク内のシステム相互間でテーブルをコピーすることもでき、データのスナップショットを使用してデータの可用性に備えることもできます。DB2 DataPropagator でこの機能を提供できます。

下の図は、MP000 システムの分散リレーショナル・データベースのコピーは、KC000 システム上に保管することができ、KC000 システムの分散リレーショナル・データベースのコピーは、MP000 システム上に保管することができることを示しています。一方の地域のクライアントは、他方のサーバーにリンクして、自らのリレーショナル・データベースの複製コピーを照会または更新することができます。

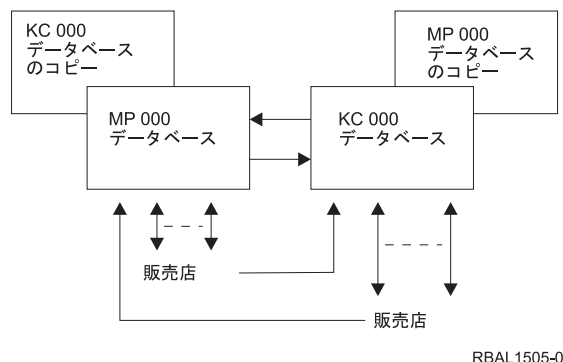


図 31. データ冗長性の例

管理担当者は、分散リレーショナル・データベース処理を可能にする方針として、最も効率的で効果的なものを決定しなければなりません。方針としては次のようなものがあります。

方針の 1 つは、MP000 が使用不能になった時には、その地域のクライアントは KC000 システムに接続して、MP000 分散リレーショナル・データベースの読み取り専用スナップショットを照会し、保守サービス作業のスケジュールを決めることができるようにするというものです。

DB2 DataPropagator は、テーブルの読み取り専用コピーを定期的リモート・システムに提供できます。Spiffy 社の場合であれば、これは毎営業日の終了時または開始時に行うようにすることができます。この例では、MP000 データベースのスナップショットは、販売店がスケジュール専用として使用するよう、24 時間前の最終時点状況を提供します。MP000 システムがオンラインに復帰したら、そのクライアントは MP000 分散リレーショナル・データベースに照会して、読み取り専用コピーに照会していた在庫要求またはその他の作業を完全に処理します。

ある地域のサーバーが使用不能になった時には、その域内販売店ユーザーは、別の地域のサーバーにある複製テーブルを更新できるようにしたいと Spiffy 社が考える場合には、別の方針を立てることになります。

たとえば、通常は MP000 データベースに接続されているクライアントが、KC000 システム上にある複製 MP000 データベースに接続して、作業を処理できるようにします。MP000 システムが再び使用可能になったら、MP000 リレーショナル・データベースは、KC000 ロケーションにあるその複製テーブルに生じている活動によるジャーナル項目を適用することによって、更新することができます。このようなジャーナル項目が元の MP000 テーブルに適用されてしまえば、分散リレーショナル・データベース・ユーザーは、サーバーとしての MP000 に再びアクセスすることができます。

各地域システムでのジャーナル管理処理によって、すべてのリレーショナル・データベースを更新します。この状況におけるジャーナル管理コピー活動は、これらのシステムでパフォーマンス上の悪影響を及ぼす可能性があるため、その量を調べなくてはなりません。

DRDA と DDM のパフォーマンス

どのような種類のアプリケーション・プログラムをシステムで実行する場合でも、パフォーマンスは常に重要な問題となります。分散リレーショナル・データベースの場合、ネットワーク、システムおよびアプリケーションのパフォーマンスすべてが重要になります。

システム・パフォーマンスは、主記憶域および補助記憶域のサイズおよび編成によって影響を受けることがあります。SQL プログラムの長所と短所を心得ていれば、パフォーマンスを向上させることができます。

関連概念:

396 ページの『DRDA と DDM のトラブルシューティング』

分散リレーショナル・データベースへのアクセスで問題が生じた場合、問題の性質を判別し、問題がアプリケーションによるものか、ローカル・システムまたはリモート・システムによるものかを判別してください。

ネットワークによる分散リレーショナル・データベースのパフォーマンスの向上

ネットワークのパフォーマンスは、さまざまな方法で改善できます。

次の方法があります。


- 回線速度
- ペーシング
- フレーム・サイズ
- RU のサイズ
- 接続タイプ (非交換と交換)

注: 無保護の会話が、以下の環境のいずれかのもとで DRDA 接続に使用されます。

- 接続が、リモート作業単位 (RUW) 接続管理を使用してプログラムから実行される場合
 - 接続を行うプログラムがコミットメント制御のもとで実行されない場合
 - 接続先のデータベースが、使用するプロトコルの 2 フェーズ・コミットをサポートしていない場合
- データに、トランザクションが 1 つのデータベース管理システムだけに影響するなどの特性がある場合、RUW 接続管理使用のプログラムから、またはコミットメント制御を使わずに実行するプログラムから接続を確立すると、2 フェーズ・コミット・フローに関連したオーバーヘッドを避けられます。

さらに、会話が DDMCNV(*KEEP) で活動状態になっていて、なおかつその会話が保護会話であると、その作業単位で DRDA または DDM 処理のどちらのために会話が使用されたかに関係なく、2 フェーズ・コミット・フローが送信されます。したがって、DDMCNV(*KEEP) を使って実行する場合、可能な限り非保護会話を使用する方が望ましいといえます。保護会話で実行しているときは、DDMCNV(*DROP) で実行するようにします。そして、それ以降の作業単位でその会話を使わない場合は、次のコミットで、RELEASE ステートメントを使用して接続と会話を終わらせます。

関連概念:

 [通信管理 \(PDF\)](#)

APPC、APPN、および HPR

TCP/IP のセットアップ

システムによる分散リレーショナル・データベースのパフォーマンスの向上

効率的なシステム・パフォーマンスを達成するには、システム・リソース間の適正なバランスが必要です。過剰に使用しているリソースがあれば、パフォーマンスに悪影響があります。このトピックでは、システム・パフォーマンスの監視に役立つコマンドについて説明します。

パフォーマンス・ツール・ライセンス・プログラムを使用して、システム・パフォーマンスの分析に役立てることができます。加えて、システム・パフォーマンスの観察に役立ついくつかのシステム・コマンドも使用できます。

- システム状況の処理 (WRKSYSSTS) コマンド
- ディスク状況の処理 (WRKDSKSTS) コマンド
- 活動ジョブの処理 (WRKACTJOB) コマンド

それらのコマンドを使う場合、一般的なレベルの活動時にシステム・パフォーマンスを観察するようにします。たとえば、システム上でジョブがまったく実行されていないときに統計を収集しても、システム・パフォーマンスの評価を行う上でほとんど価値がありません。システム・パフォーマンスを観察するときには、以下のステップを完了してください。

1. WRKSYSSTS、WRKDSKSTS、または WRKACTJOB コマンドを入力する。
2. システムが最低 5 分間データを収集できるようにする。
3. F5 (再表示) を押して、画面を再表示し、パフォーマンス・データを表示する。
4. 新しいデータに基づいてシステムを調整する。

経過時間カウンター機構を再始動するには、F10 (再始動) を押してください。

システムのパフォーマンスを観察する際は、システム状況の処理 (WRKSYSSTS) コマンドとWRKACTJOB (活動ジョブの処理) コマンドを両方使用します。各観察期間ごとに、サーバー・パフォーマンスの測定値を、あらかじめ設定してある目標値と対比して、検討および評価してください。

一般的な測定値には、次のものが含まれます。

- WRKACTJOB 画面から得られる対話式スループットおよび応答時間。
- バッチ・スループット。活動化しているバッチ・ジョブの AuxIO および CPU% の値を監視します。
- スプール・スループット。活動化している書き込み機能の AuxIO および CPU%の値を監視します。

調整を加えるたびに、主要なパフォーマンス測定値のすべてを、測定し比較する必要があります。調整の実行および評価は、一度に 1 つずつ行ってください。

関連概念:

実行管理機能

関連タスク:

269 ページの『分散リレーショナル・データベースにおける活動ジョブの処理』

数人のユーザーに対して実行されているジョブをモニターする場合や、ジョブを探しているものの、ジョブの名前やユーザー ID がわからない場合は、WRKACTJOB (活動ジョブの処理) コマンドを使用してください。

関連資料:

システム状況の処理 (WRKSYSSTS) コマンド

ディスク状況の処理 (WRKDSKSTS) コマンド

活動ジョブの処理 (WRKACTJOB) コマンド

DRDA のパフォーマンスに関する考慮事項

分散リレーショナル・データベースのパフォーマンスは、データベースの総合的な設計の影響を受けます。分散データの位置、使用するコミットメント制御のレベル、および SQL 索引の設計など、すべてがパフォーマンスに影響します。

関連概念:

40 ページの『計画と設計』

分散リレーショナル・データベースを準備するにあたっては、業務上の要件とリレーショナル・データベース技術の両方を理解していなければなりません。DDM の使用を準備するにあたっては、通信要件、セキュリティ要件、および変更要件を含む複数の要件を満たしている必要があります。

DRDA のブロック化に影響を与える要因

パフォーマンスに影響する最も重要な要因は、アプリケーション・リクエスター (AR) とアプリケーション・サーバー (AS) の間でデータが転送されるときに、ブロック化が行われるかどうかということです。データのブロックとして伝送される行のグループの場合、同じデータを一度に 1 行ずつ伝送する場合と比べて、通信のオーバーヘッドははるかに少なく済みます。

別の IBM i プラットフォームに接続されている場合に、ブロック化を制御する方法の 1 つとして、SQL の複数行 INSERT ステートメントおよび複数行 FETCH ステートメントを使う方法があります。複数行 FETCH を使うと、ハード・エラーの発生やデータ終わりになった場合を除き、FOR n ROWS 文節に指定された行数のブロック化が強制的に行われます。以下の説明では、単一行 FETCH の場合に、ブロック化が行われるかどうかを判別するための規則を紹介します。

DB2 for IBM i から DB2 for IBM i へのブロック化は、これらの条件下で発生します。これらの条件は、複数行 FETCH ステートメントを使用する場合には、適用されません。以下のそれぞれのケースでリストされている条件が 1 つでも当てはまれば、ブロック化は行われません。

- カーソルがスクロール・センシティブである。
- ロケーターとして指定変更できる LOB がカーソルに含まれている。
- 結果セットが配列結果セットではない。
- DRDA AR が非ブロッキングを強制しない。

詳しくは、ご使用のデータベース・インターフェースのカーソル属性を参照してください。

関連資料:

400 ページの『ループ、待機、またはパフォーマンスの問題の解決』

要求が完了するのに想定された時間より長くかかっている場合、まずアプリケーション・リクエスター (AR) を検査します。

DRDA 照会ブロックのサイズに影響を与える要因

DRDA 照会ブロックは、以下のいずれかの方法で制御されます。

- アプリケーション・リクエスター (AR) がサイズ x の照会行セットを指定する場合、アプリケーション・サーバー (AS) は x 個の行を含む照会ブロックを戻します。結果セットに x 個未満の行が残っている場合、この残りの行が戻されます。
- AR が照会行セットを指定しない場合、AS は、DRDA AR で指定されている照会ブロック・サイズに適合する数だけ、照会ブロック中の全行を戻します。クライアントのインターフェースかクライアントのデータベース管理システムで、照会行セットのサイズと照会ブロックのサイズを構成できます。

DDM のパフォーマンスに関する考慮事項

このトピックには、DDM を使用するときパフォーマンスを向上させるための参考情報と、ある種の機能を実行するのに DDM 以外のものを使用する場合に関する情報が記述されています。

- DDM ファイルを CPYF コマンドに指定するとき、次の項目がすべて満たされれば、最良のパフォーマンスを得ることができます。
 - 取り出し元ファイルは論理ファイルまたは物理ファイルであり、受け入れ先ファイルが物理ファイルである場合。
 - FMTOPT は、*NONE または *NOCHK であるか、または何も指定されていない場合。
 - INCHAR、INCREL、ERRLVL、RCDDMT(*ALL)、PRINT(*COPIED)、PRINT(*EXCLD)、SRCSEQ、TOKEY、SRCOPT、または FROMKEY パラメーターは指定されていない場合。
 - 取り出し元ファイルが、*NONE または *START 以外の POS キーワードで一時変更されていない場合。
 - 受け入れ先ファイルが、INHWRT(*YES) で一時変更されていない場合。
- QUERY ファイル・オープン (OPNQRYF) コマンドでは、DDM 体系へのシステム/38 拡張機能を使用しません。システム/38 DDM 体系拡張機能では、DDM システム処理時間が極力短縮されています。次の場合には、この拡張機能を使用することはできません。
 - クライアント・システムがシステム/38 プラットフォームでも IBM i オペレーティング・システムでもない場合。
 - サーバー・システムがシステム/38 でも IBM i オペレーティング・システムでもない場合。
- IBM i コマンド OPNQRYF OPTIMIZE(*YES) などの QUERY 機能を使用すれば、各システム間で転送されるデータ量を大幅に削減することができます。ただし、ユーザー作成アプリケーションの場合、システム間で交換されるデータ量は、DDM を使って、IBM i を実行していないシステムとの通信を行うのに使用するものより多くなります。この追加データで、IBM i DDM 拡張機能が提供され、クライアント・システムの DDM 処理オーバーヘッドも軽減されます。たとえば、通常の読み取り、書き込み、更新、追加、および削除の各操作を行う場合には、次の事項を考慮してください。
 - 標準 DDM 体系の DDM オーバーヘッド・データには、ファイル識別コード、命令コード、および簡単な操作結果情報などの情報が含まれます。ユーザー・プログラムのキー別読み取り操作は、キー・データの長さ以外に、約 40 文字の DDM 情報を使用します。リモート・システムから返されるデータは、約 32 文字の DDM 情報と、データ・ファイル・レコードの長さを使用します。
 - システム/38 DDM 体系拡張機能を追加すると、レコード様式の識別、および入出力フィールドバック域情報の主要な部分など、データ・オーバーヘッドが増加します。ユーザー・プログラムのキー別読み取り操作は、キー・データの長さ以外に、約 60 文字の DDM 情報を使用します。リモート・システムから返されるデータは、約 80 文字の DDM 情報と、データ・ファイル・レコードの長さを使用します。通常、データ・ストリーム内で追加される長さは、顕著な長さではありません。ただし、回線活動が多くなるにしたがって、このような拡張データ・ストリームを使用した方が、標準 DDM データ・ストリームを使用するよりも、回線使用率のピークには、早めに達してしまいます。
- ターゲット DDM ジョブ優先順位は、関連サブシステム記述経路指定項目で指定するジョブ・クラスで制御します。次の経路指定項目は通常、すべてのターゲット (プログラム開始要求) ジョブで使用されるものです。

```
ADDRTGE ... PGM(*RTGDTA) ... CMPVAL(PGMEVOKE 29)
```

IBM i オペレーティング・システムとともに出荷されるサブシステム QBASE と QCMN には、この経路指定項目が備わっています。

ターゲット DDM ジョブをサブシステム内で実行するときに、同じサブシステム内で他の APPC ターゲット・ジョブを実行する場合とは異なる優先順位で実行する場合は、適切な順序番号を指定して次の経路指定項目を挿入します。

```
ADDRTGE SBSD(xxx) SEQNBR(nnn) CMPVAL(QCNTEDDM 37)
        PGM(*RTGDTA) CLS(uuu)
```

クラス *uuu* は、ターゲット・ジョブ優先順位を決めるのに使用します。

- リモート・ファイル内のレコードを表示するときは、できる限り表示装置パススルーを使用してください。そうでない場合は、物理ファイル・メンバー表示 (**DSPPFM**) コマンドで直接レコード位置決めを使用する必要があります。
- DDM ユーザー出口機密保護プログラムが CL プログラムで、このプログラムにより IBM i 例外と照会メッセージが作成されたために、サーバー・システムのオペレーターによる応答が必要になった場合は、ユーザー出口プログラムとクライアント・システム・ジョブは両方ともこの応答が来るまで待機する必要があります。APPC リモート・ロケーション情報用の通信項目追加 (**ADDCMNE**) コマンドに指定した TDDM ジョブ記述に INQMSGRPY(*SYSRPLY) を指定して、デフォルト・システム応答リストを使用できるかどうか考慮してください。
- WAIT パラメーターと WAITFILE パラメーターは、オブジェクト割り振り (**ALCOBJ**) コマンドや メッセージ受け取り (**RCVMSG**) コマンドのようなコマンドで使用するパラメーターですが、これらのパラメーターは、クライアント・システム・プログラムに対しては何も影響を与えません。これらのパラメーターは、ローカル・ファイルへアクセスする場合と同じように機能します。クライアント・システムで実行するコマンド上に指定する待ち時間の値は、クライアント・システムに対しては影響を与えず、サーバー・システムにのみ、しかもそれが IBM i オペレーティング・システムまたはシステム/38 プラットフォームの場合にのみ影響を与えます。

注:

1. IBM i システム間通信機能 (ICF) ファイルの作成コマンドや変更コマンドの WAITFILE パラメーターで、獲得操作や開始機能を実行しているときに、セッション資源が使用可能になるまでにどのくらいの時間 APPC サポートが待つかが決まります。隣接システムとの接続が交換回線接続を経由している場合には、WAITFILE 値をセッション用には使用しません。その例には、SDLC 交換回線、X.25 SVC 回線、イーサネット回線、またはトークンリング接続などがあります。交換回線接続を用いると、獲得操作や開始機能がタイムアウトになることはありません。
 2. APPN セッションは、リモート・システムまで達するのに複数のシステムと回線を経由することがあるため、このような場合には、WAITFILE タイマーを調整して、時間に余裕を持たせてください。アプリケーション・プログラムが、APPN 機能を使用するよう構成されたネットワーク内で稼働するときには、WAITFILE パラメーターに *IMMED を指定してはなりません。このパラメーターに指定する値は、ネットワークのサイズとタイプによって異なります。
- すべての LU セッション・タイプ 6.2 データ交換の場合と同様、特定の SNA パラメーターは、パフォーマンスに影響を与えます。このようなパラメーターには、パス情報単位サイズ (MAXFRAME)、要求/応答単位サイズ (MAXLENRU)、SNA ペーシング (INPACING、OUTPACING)、および X.25 の場合の packetsize とウィンドウ・サイズがあります。一般的に、使用する値が大きければ大きいほど、より良いパフォーマンスが実現されます。
 - **SNA パス情報単位サイズ**

パス情報単位 (PIU) とは、2 つのシステム間でやりとりされる実際のデータ伝送ブロックのサイズのことです。APPC 制御装置記述作成 (CRTCTLAPPC) コマンドや SNA ホスト制御装置記述作成 (CRTCTLHOST) コマンド上の MAXFRAME パラメーターで、ローカル・システムが使用するパス情報

単位サイズを指定します。セッション確立時に、両方のシステムがどのサイズを使用するかを決めますが、サイズは常に小さい方の値になります。リモート・システムごとに、PIU サイズに関する考慮事項が異なることもあります。

- **SNA 応答/要求単位サイズ**

応答/要求単位 (RU) サイズ (CRTMODD MAXLENRU) は、実際に伝送するバス情報単位にデータが収まるよう、システム・バッファーに入れる量を制御します。APPC では、送信 RU および受信 RU の長さは、セッション確立時に折衝されます。この場合も、折衝の結果は、使用されている最小値になります。リモート・システムによっては、RU サイズに関する考慮事項が異なることもあります。

- **SNA ペーシング値**

ペーシング値は、バッファー記憶域が次の伝送を処理できるようになったことを示す応答が必要になるまでに、何回、要求/応答単位 (RU) を受信または送信できるかを決めます。セッション確立時に、両方のシステムがどのサイズを使用するかを決めますが、サイズは常に小さい方の値になります。

バッチおよび対話式処理が同じ通信回線で同時発生し、バッチ処理より対話式処理を優先したい場合は、IBM i ジョブ優先順位を使用できます。さらに、バッチ・アプリケーションのペーシングを小さくする一方、対話式アプリケーションの方の値を大きくして、対話式アプリケーションに回線活動の優先レベルを与えなければならないことがあります。

IBM i オペレーティング・システムでは、アプリケーションごとに異なったモードを作成して (モード記述作成 [CRTMODD] コマンド)、それぞれ異なるペーシング値を指定することができます。リモート・システムによっては、SNA ペーシング値に関する考慮事項が異なることもあります。

- **X.25 パケット**

MAXFRAME 値より小さい X.25 パケットの場合、非 X.25 データ・リンクを通してデータ伝送時間が長くなります。一般的に X.25 では、MAXFRAME と実際に伝送されるデータ量が大きくなればなるほど、この時間差が大きくなります。DDM では、通常ファイル・レコード・データに DDM 制御情報が追加されるため、ローカル・ファイル処理とリモート・ファイル処理との差や、非 X.25 データ・リンクと X.25 データ・リンクとの差に対するパケット・サイズの影響がさらに大きくなります。

多くの非ブロック化 DDM 操作の場合、データを伝送するのに必要なパケット数が非常に多くなるため、X.25 アダプター内のパケット処理オーバーヘッドでパフォーマンスに多大な影響が出ます。ネットワークと通信プロダクトでサポートされている最大 X.25 パケット・ウィンドウ・サイズを使用して、パフォーマンスを最大限に保ちます。

リモート・ファイルにアクセスするのに X.25 を使用しなければならないとき、80 文字より小さいレコードなどの小さい非ブロック化レコードを多数連続して伝送すると、ユーザー・データの伝送の場合と比較して、X.25 パケット文字を X.25 アダプターが処理するのに要する時間が釣り合わなくなることがあります。

一般的に、X.25 パケットの処理でのオーバーヘッドのため、同じ伝送速度を使用し、データ転送が単一方向に行われたとしても、従来の回線を用いたときよりもスループットが低下することになります。データを双方向に同時に転送すれば、X.25 全二重サポートの利点を生かせます。システム/38 での処理に対する全体的な影響は最小限にとどまりますが、これは、X.25 統合アダプターでパケット処理におけるオーバーヘッドが発生するためです。

一般的に、DDM を使用したリモート・ファイルの処理は、ファイル・コピー (CPYF) コマンドで利用できるようなアプリケーション・プログラムやユーティリティー機能から見ると、透過処理です。ただし、通信回線を使用してリモート・ファイルへアクセスするほうが、時間が長くなります。ローカル・ファイル処

理とリモート・ファイル処理におけるパフォーマンス上の差は、パフォーマンスの測定単位あたりのリモート・ファイルへのアクセス回数、データ・レコード長、および伝送速度に比例します。

ローカル・ファイル処理とリモート・ファイル処理での別の相違点として、ローカル・ファイルに対して入力操作や出力操作を行っても、システムは、ディスクからデータ・ブロックを転送してからそのディスクにデータ・ブロックを書き込むので、ただちに物理ディスク操作にはつながらないという点があります。このため、場合によっては、ユーザー・プログラムが主記憶域内のデータにアクセスした時点とは別の時点で、物理入出力操作が行われることがあります。したがって、ローカル・ファイルとリモート・ファイルでのパフォーマンスの差を最小限にとどめるには、アプリケーション設計をよく了解し、また、DDM を使用してどのリモート・ファイルにアクセスするかを決める際に、ファイルへのアクセス量とアクセス・タイプに対して配慮することが肝要です。

1 回のリモート・アクセスにおいて、余分な時間がかかる原因は以下のとおりです。

- ローカル・システム・ファイル・インターフェースを、DDM 体系インターフェースへ変換するための余分なシステム処理。
- 通信回線を通して伝送されるデータ量。
- リモート・システムでのファイル操作の処理量。
- 通信回線の速度。

通信回線時間がこの余分な時間の大半を占めますが、実際にかかる時間は、回線速度と、DDM 機能の実行時における回線使用量によって異なります。

DDM 以外の場合と同様、ローカル・システムやリモート・システムのジョブ優先順位が、パフォーマンスに対して最も大きい影響を与えます。IBM i オペレーティング・システムでは、使用するクラスの PRIORITY 値と TIME SLICE 値で、ジョブ優先順位を制御します。SDDM はソース・ジョブの下で稼働する一方、TDDM は、サーバー・システムのサブシステムの APPC 経路指定項目に割り当てられたクラスの下で稼働します。複数のファイルにアクセスするアプリケーションでは、最もアクセスされる回数の多いファイルを、実行しているプログラムと同じシステムに入れ、それよりアクセス回数の少ないファイルはリモート・システムに入れておくのが最善です。ファイルとアプリケーション・プログラムの配置に関する主要な考慮事項を次に示します。

- ファイル保守の第一責任担当システムを明確にしておかなければなりません。どの複数システム・アプリケーションでも、1 つのシステムにのみファイル保守を担当させれば、最高のパフォーマンスが得られます。アプリケーション・プログラムが排他的 (非共用) 処理を介してファイルを保守する場合、そのアプリケーション・プログラムがそのファイルとともにシステムにあるときに、最高のパフォーマンスを実現できます。

場合によっては、ファイルをローカル・システムへ送り返すには、次のものが必要になることがあります。

- APPC プログラム。
- リモート DDM ファイルを使用するプログラム。
- DDM を使用したファイル・コピー (CPYF) コマンド。
- オブジェクト配布機能の SNDNETF 命令と RCVNETF 命令。対話式アプリケーションにおいて、転送する表示データ量が DDM を使用して送られるデータベース・ファイル量よりはるかに少ない場合には、表示装置パススルーを使用できるかどうかを検討してください。
- ファイルを配置するとき、最高のパフォーマンスを得るためにアプリケーション処理をリモート・システムに移動しなければならない場合は、リモート・コマンド投入 (SBMRMTCMD) コマンドの使用を検討する必要があります。このコマンドの使用は、バッチ処理入力ストリームにおいて最も効果がありま

す。バッチ処理入力ストリームでは、いずれのプログラムも前のプログラムが完了するまで待機させられます。SBMRMTCMD コマンドを使用できるのは、ソース・システムとサーバー・システムが IBM i またはシステム/38 の場合のみです。たとえば、プログラム A がローカル・ファイルへアクセスすると仮定します。プログラム A は、ローカル・システムで稼働します。プログラム B は、リモート・ファイルへアクセスします。SBMRMTCMD コマンドを用いれば、リモート・システムでプログラム B を実行することができます。

- ファイル保守をシステム間で共用している場合、ファイルの更新、追加、および削除の操作の比率の最も大きいシステムにそのファイルを置けば最高のパフォーマンスを得ることができます。

場合によっては、ソース・システムとターゲット・システムの APPC プログラムをペア化すれば、DDM を使用した場合のパフォーマンスが向上します。たとえば、リモート・システムにある 10 個のレコードを取り出すと仮定します。DDM を使用している場合で、レコード・ブロック化を使用できないとき (例：ユーザー・プログラム・ランダム入力命令、変更用の順次入力、または OVRDBF SEQONLY[*NO] コマンドの使用) に、10 回のデータ伝送が行われ、10 回受信されて、合計 20 回の伝送が行われたと仮定します。ユーザー作成 APPC プログラムは、そのデータ・ストリーム内に情報を追加すれば、データ要求とそのデータの受け取りを、20 回ではなく 2 回のデータ伝送で行えるようになります。その 2 回の内訳は、1 回が顧客 00010 のすべてのレコードを求める要求、もう 1 回は顧客 00010 の 10 個のレコードの入った応答になります。

バッチ・ファイル処理と対話式ファイル処理の 2 つのサンプル・アプリケーション処理方法について検討します。

関連概念:

50 ページの『DDM の計画と設計』

分散データ管理機能 (DDM) を正しく使用するには、いくつかの要件を満たす必要があります。

DDM でのバッチ・ファイル処理

DDM でバッチ・ファイル処理を使用するときは、以下の項目を考慮してください。

- アプリケーションが、**順次入力専用** のため、または**出力追加** のためにローカル・ファイルをオープンするときには、システムは、ブロック化技法を使って最大スループットを達成します。DDM を使用してアクセスするリモート・ファイルに必ずブロック化が使用されるようにするには、プログラム内でランダム・レコード処理操作を使用しないで、プログラムがオープンする DDM ファイルに OVRDBF SEQONLY(*YES) を指定します。
- ファイルへアクセスするのに、高水準言語 (HLL) プログラム内で読み取り操作と次の読み取り操作を使用すれば、指定した SEQONLY(*YES) の効果が最大に発揮されます。
- ILE RPG の CHAIN 命令や ILE COBOL プログラミング言語の START 命令のようなランダム処理命令を使用すると、アプリケーションがファイル・データを順次処理する場合でも、DDM は通信回線を経由して非ブロック化命令を送ることになります。この場合、ローカル・ファイルとリモート・ファイルの処理には大きな時間差が生じることになります。
- 単純な物理ファイルの転送 (全レコードを転送して、アプリケーションはデータを処理しない) が要求される場合は、**ファイル・コピー (CPYF)** コマンドを使用して DDM を使用するか、または**データベース・ファイル一時変更 (OVRDBF)** コマンドの SEQONLY(*YES) を指定した DDM を使用するユーザー作成プログラムを使用した方が、ユーザー作成 APPC プログラムを使用するよりも早くデータを転送することができます。**ファイル・コピー・コマンド**と DDM SEQONLY(*YES) サポートを使えば、標準 ILE RPG または ILE COBOL APPC プログラムの場合よりも、プログラムと APPC データ管理モジュールとの間でやりとりする呼び出しと応答の回数が少なくて済むからです。

- ILE RPG または ILE COBOL の順次入力専用アプリケーションの場合、最良のスループットを得るには、レコード数を付けずに SEQONLY(*YES) を指定する必要があります。ILE RPG または ILE COBOL のキー付きファイルへの順次出力専用アプリケーションの場合は、レコード数を大きな値に指定する必要があります。
- **ネットワーク・ファイル送信 (SNDNETF)** コマンドは、ファイル内のすべてのレコードをリモート IBM i オペレーティング・システムへ転送する場合に、DDM またはユーザー作成 APPC プログラムに代わるものと見なすことができます。**SNDNETF** コマンドを使用するには、ソースとターゲットの IBM i オペレーティング・システム上で SNADS を構成する必要があります。IBM i オペレーティング・システムのソース・システムとターゲット・システムとの間に 1 つ以上の中間システムがあるときには、SNADS は、正しく構成されたデータの中間ノード経路指定を行います。
- SNADS を使用して **SNDNETF** コマンドを使用すると、複数のノード・ネットワークを経由して、1 つ以上のサーバー・システムの複数のユーザーへ同じデータ・コピーを送信することができ、また、SNADS 配布待ち行列パラメーターの使用により、そのデータを伝送する際の時間スケジュールをたてることができるという利点があります。

ただし、サーバー・システムのユーザーが **SNDNETF** コマンドを使用するには、SNADS がなければならぬだけでなく、**ネットワーク・ファイル受け取り (RCVNETF)** コマンドを実行して、サーバー・システムでそのファイルを使用できるようにする必要があります。DDM を使用すれば、サーバー・システムでこのように追加の処理を行う必要はありません。

一般的に、SNADS を使用したファイル伝送時間 (ユーザー・プログラム DDM 順次ファイル処理、DDM の **ファイル・コピー** コマンド、およびユーザー作成 APPC プログラムを 2 つの IBM i オペレーティング・システム間で使用した場合) の個別差は、いずれも 10% 以内です。ただし、サーバー・システム上でそのデータ・コピーを使用できるようにするのに **SNDNETF** および **RCVNETF** コマンドを使用すると、他のファイル転送方式の場合より、合計処理時間は長くなります。

- **SNDNETF** コマンドは、保管ファイルにあるオブジェクトを送信することができるため、この技法を使用して実際に送られるデータ量は、他の技法を使用して送るデータ量より少なくなることがあります。送るデータベース・ファイル・データに、相当な数の重複文字ストリングが入っている場合、**オブジェクト保管 (SAVOBJ)** コマンドのパラメーター DTACPR(*YES) (データ圧縮) を使用すれば、SNADS 配布機能を使用して実際に送るデータ量を大幅に削減することができます。ただし、重複文字ストリングが少ししかない場合には、送られるデータ量にほとんど変わりはありません。
- IBM i ファイル転送サブルーチンを使用して、IBM i オペレーティング・システム相互間、および IBM i とシステム/36 プラットフォームの間で、ファイル全体を転送することもできます。これらのサブルーチンは、高水準言語プログラムから呼び出すことが可能で、場合によっては、DDM を使用した場合と同様のスループットを得ることもできます。

DDM での対話式ファイル処理

DDM を使って対話式ファイル処理を行うときは、次の事項を考慮してください。

- パフォーマンス測定単位あたりのランダム・ファイル命令数が多ければ多いほど、ローカル・システムとリモート・システムのファイル処理の時間差は大きくなります。その理由は、各命令が、通信回線を通して別々に送られるからです。DDM には、次の命令がいつ来るか予測できません。

ワークステーション・サブファイル・サポートを介して (例にあるとおりに)、画面出力を生成する単純な照会アプリケーションを使用して、実行キーを押すごとに 2 回のランダム・レコード検索を行うアプリケーションと、15 回のランダム・レコード検索を行うアプリケーションとを対比しつつ検討を行います。2 回レコードが検索されるときには、オペレーターは応答時間の遅延にほとんど気付くことはありません。しかし、リモート・システムのレコードが 15 回ランダムに検索されると、ローカル・システムとリモート・システムの応答時間には、明らかな差が出ます。

- ローカル (クライアント) システムへ送り返されるデータの、パフォーマンス測定単位あたりの量が、表示装置に表示されるデータ量をはるかに超えるときには、表示装置パススルーの使用を検討する必要があります。テスト結果で明らかになった点は、1 つの非ブロック化 DDM レコード GET 操作と、それと同等のユーザー作成 APPC 操作とでは、総経過時間にはほとんど違いはなく、APPC を使用した方が若干、時間が短縮されるということです。DDM 操作では、直接 APPC インターフェースの場合より長い処理時間を要します。

また、いずれの DDM 操作でも常に、データの保全性を確保するためにリモート・システムからの操作結果応答を必要とするのに対して、これに対応するユーザー設計のパートナー APPC プログラムでは、後からしか操作結果の検査を行わないため、更新、追加、および削除の各操作に関しては有利になります。

- 他のシステムにあるファイルへアクセスするときは余分に時間がかかること、特に、回線を経由する通信のための時間がかかることに注意してください。ファイルをローカル・ファイルにまたはリモート・ファイルのどちらにするかを決める場合に (何回も使用するファイルの場合は特に)、この件についての配慮が必要です。

DDM 会話の長さに関する考慮事項

DDM を使用する場合、会話の長さに関して次の事項に考慮してください。

- ソース・ジョブにおいては、DDM 会話を複数のプログラムや DDM ファイルが使用する予定であれば、DDMCNV ジョブ属性には *KEEP の値を指定してください。この値を指定すれば、ソース・ジョブ内で同じ組み合わせのロケーションとモード用に DDM ファイルにアクセスするごとに、ターゲット・ジョブ (TDDM) を始動するのに必要な時間と資源が節減されます。
- ターゲット DDM 管理プログラムを始動すると、システムと通信回線に相当なオーバーヘッドが生じます。この処理には、APPC プログラム開始要求、システム・タイプ識別、およびファイル・オープン処理が含まれます。ただし、会話を活動状態に保つ必要がない場合、DDMCNV には *DROP を指定してください。ローカル DDM ファイルがクローズされると使用中のセッションは解放されるため、DDM、SNADS や表示装置パススルーなどのその他の APPC 機能を使う他のジョブが、その同じリモート・システムに対してそのセッションを使用できるようになります。
- クライアント・システムとサーバー・システムが IBM i オペレーティング・システムまたはシステム/38 プラットフォームであれば、アプリケーション・プログラムから出すファイル入出力要求で、それぞれの要求をエンコードしたりデコードするのに必要な時間を最短化するような方式の DDM サポートを使用することができます。これは、DDM 体系へのシステム/38 拡張機能を使用して行います。

クライアント・システムとサーバー・システムが IBM i オペレーティング・システムでもシステム/38 でもない場合には、DDM 体系へのシステム/38 拡張機能は使用されません。

例: アプリケーション・プログラミング

これらの例では、分散リレーショナル・データベースを機能指定作業に使用方法と、DDM 関連タスクをコード化する方法を示しています。

DRDA の例

分散リレーショナル・データベースの使用に関するこのアプリケーション例は、RPG/400、COBOL/400、Java、および ILE C プログラミング言語で書かれています。

例: 分散リレーショナル・データベースに対するビジネス要件

この例の分散リレーショナル・データベース用のアプリケーションは、自動車販売店または代理店ネットワーク内での部品在庫管理です。

このプログラムでは、地区部品在庫テーブルの各部品ごとに在庫のレベルを検査します。レベルが追加発注点以下である場合には、プログラムは中央のテーブルを検査して未処理のままになっている既存発注の有無、および各発注に対して出荷されている数量を調べます。

正味数量 (地区在庫 + 発注量 - 出荷量) が追加発注点以下である場合には、中央システムの該当するテーブルに行を挿入することによって、該当の部品の発注を行います。地区システム側で報告書が印刷されます。

技術上の注記

コミットメント制御

このプログラムでは、ローカルおよびリモートの論理作業単位 (LUW) の概念を使用しています。このプログラムでは、リモート作業単位を使用しているため、別のシステムで新しい作業単位を開始する前に、このシステムの現行 LUW をクローズ (COMMIT) する必要があります。

カーソル位置変更

LUW がコミットされて、アプリケーションが別のデータベースに接続すると、すべてのカーソルはクローズされます。このアプリケーションでは、部品在庫ファイルを読み取るカーソルは、次の部品番号で再オープンされる必要があります。このことを実現するために、カーソルは、部品番号が部品番号の現行値より大きいところで始まり、部品番号によって順序付けられるように定義されています。

注: 同じ部品番号に重複行がある場合には、この技法は機能しません。

例: プログラム定義

以下に、自動車販売店または代理店ネットワーク内での部品在庫管理に関するプログラム定義の例を示します。

注: コードの例を使用することは、443 ページの『コードに関するライセンス情報および特記事項』の条件に同意したものとされます。

```

/*****
/* Local Database */
/*****
CREATE COLLECTION DRDA

CREATE TABLE DRDA/PART_STOCK (PART_NUM CHAR(5) NOT NULL,
                                PART_UM CHAR(2) NOT NULL,
                                PART_QUANT INTEGER NOT NULL WITH DEFAULT,
                                PART_ROP INTEGER NOT NULL,
                                PART_EOQ INTEGER NOT NULL,
                                PART_BIN CHAR(6) NOT NULL WITH DEFAULT
                                )

CREATE UNIQUE INDEX DRDA/PART_STOCI ON DRDA/PART_STOCK (PART_NUM ASC)

/*****
/* Remote Database */
/*****
CREATE COLLECTION DRDA

CREATE TABLE DRDA/PART_ORDER (ORDER_NUM SMALLINT NOT NULL,
                                ORIGIN_LOC CHAR(4) NOT NULL,
                                ORDER_TYPE CHAR(1) NOT NULL,
                                ORDER_STAT CHAR(1) NOT NULL,
                                NUM_ALLOC SMALLINT NOT NULL WITH DEFAULT,
                                URG_REASON CHAR(1) NOT NULL WITH DEFAULT,
                                CREAT_TIME TIMESTAMP NOT NULL,
                                ALLOC_TIME TIMESTAMP,
                                CLOSE_TIME TIMESTAMP,
                                REV_REASON CHAR(1)
                                )

CREATE UNIQUE INDEX DRDA/PART_ORDEI ON DRDA/PART_ORDER (ORDER_NUM ASC)

CREATE TABLE DRDA/PART_ORDLN (ORDER_NUM SMALLINT NOT NULL,
                                ORDER_LINE SMALLINT NOT NULL,
                                PART_NUM CHAR(5) NOT NULL,
                                QUANT_REQ INTEGER NOT NULL,
                                LINE_STAT CHAR(1) NOT NULL
                                )

CREATE UNIQUE INDEX PART_ORDLI ON DRDA/PART_ORDLN (ORDER_NUM ASC,
                                                    ORDER_LINE ASC)

CREATE TABLE DRDA/SHIPMENTLN (SHIP_NUM SMALLINT NOT NULL,
                                SHIP_LINE SMALLINT NOT NULL,
                                ORDER_LOC CHAR(4) NOT NULL,
                                ORDER_NUM SMALLINT NOT NULL,
                                ORDER_LINE SMALLINT NOT NULL,
                                PART_NUM CHAR(5) NOT NULL,
                                QUANT_SHIP INTEGER NOT NULL,
                                QUANT_RECV INTEGER NOT NULL WITH DEFAULT
                                )

CREATE UNIQUE INDEX SHIPMENTLI ON DRDA/SHIPMENTLN (SHIP_NUM ASC,
                                                    SHIP_LINE ASC)

```

図 32. コレクションとテーブルの作成

```

/*****
/* Local Database */
/*****
/*****
/* PART_STOCK */
/*****
INSERT INTO DRDA/PART_STOCK VALUES('14020','EA',038,050,100,' ')
INSERT INTO DRDA/PART_STOCK VALUES('14030','EA',043,050,050,' ')
INSERT INTO DRDA/PART_STOCK VALUES('14040','EA',030,020,030,' ')
INSERT INTO DRDA/PART_STOCK VALUES('14050','EA',010,005,015,' ')
INSERT INTO DRDA/PART_STOCK VALUES('14060','EA',110,045,090,' ')
INSERT INTO DRDA/PART_STOCK VALUES('14070','EA',130,080,160,' ')
INSERT INTO DRDA/PART_STOCK VALUES('18020','EA',013,025,050,' ')
INSERT INTO DRDA/PART_STOCK VALUES('18030','EA',015,005,010,' ')
INSERT INTO DRDA/PART_STOCK VALUES('21010','EA',029,030,050,' ')
INSERT INTO DRDA/PART_STOCK VALUES('24010','EA',025,020,040,' ')
INSERT INTO DRDA/PART_STOCK VALUES('24080','EA',054,050,050,' ')
INSERT INTO DRDA/PART_STOCK VALUES('24090','EA',030,025,050,' ')
INSERT INTO DRDA/PART_STOCK VALUES('24100','EA',020,015,030,' ')
INSERT INTO DRDA/PART_STOCK VALUES('24110','EA',052,050,080,' ')
INSERT INTO DRDA/PART_STOCK VALUES('25010','EA',511,300,600,' ')
INSERT INTO DRDA/PART_STOCK VALUES('36010','EA',013,005,010,' ')
INSERT INTO DRDA/PART_STOCK VALUES('36020','EA',110,030,060,' ')
INSERT INTO DRDA/PART_STOCK VALUES('37010','EA',415,100,200,' ')
INSERT INTO DRDA/PART_STOCK VALUES('37020','EA',010,020,040,' ')
INSERT INTO DRDA/PART_STOCK VALUES('37030','EA',154,055,060,' ')
INSERT INTO DRDA/PART_STOCK VALUES('37040','EA',223,120,120,' ')
INSERT INTO DRDA/PART_STOCK VALUES('43010','EA',110,020,040,' ')
INSERT INTO DRDA/PART_STOCK VALUES('43020','EA',067,050,050,' ')
INSERT INTO DRDA/PART_STOCK VALUES('48010','EA',032,030,060,' ')

```

図 33. テーブルへのデータの挿入

```

/*****
/* Remote Database
/*****
/*****
/* PART_ORDER TABLE
/*****
INSERT INTO DRDA/PART_ORDER VALUES(1,'DB2B','U','0',0,' ','1991-03-12-17.00.00',
NULL,NULL,NULL)

INSERT INTO DRDA/PART_ORDER VALUES(2,'SQLA','U','0',0,' ','1991-03-12-17.01.00',
NULL,NULL,NULL)

INSERT INTO DRDA/PART_ORDER VALUES(3,'SQLA','U','0',0,' ','1991-03-12-17.02.00',
NULL,NULL,NULL)

INSERT INTO DRDA/PART_ORDER VALUES(4,'SQLA','U','0',0,' ','1991-03-12-17.03.00',
NULL,NULL,NULL)

INSERT INTO DRDA/PART_ORDER VALUES(5,'DB2B','U','0',0,' ','1991-03-12-17.04.00',
NULL,NULL,NULL)

/*****
/* PART_ORDLN TABLE
/*****
INSERT INTO DRDA/PART_ORDLN VALUES(1,1,'24110',005,'0')

INSERT INTO DRDA/PART_ORDLN VALUES(1,2,'24100',021,'0')

INSERT INTO DRDA/PART_ORDLN VALUES(1,3,'24090',018,'0')

INSERT INTO DRDA/PART_ORDLN VALUES(2,1,'14070',004,'0')

INSERT INTO DRDA/PART_ORDLN VALUES(2,2,'37040',043,'0')

INSERT INTO DRDA/PART_ORDLN VALUES(2,3,'14030',015,'0')

INSERT INTO DRDA/PART_ORDLN VALUES(3,2,'14030',025,'0')

INSERT INTO DRDA/PART_ORDLN VALUES(3,1,'43010',003,'0')

INSERT INTO DRDA/PART_ORDLN VALUES(4,1,'36010',013,'0')

INSERT INTO DRDA/PART_ORDLN VALUES(5,1,'18030',005,'0')

/*****
/* SHIPMENTLN TABLE
/*****
INSERT INTO DRDA/SHIPMENTLN VALUES(1,1,'DB2B',1,1,'24110',5,5)

INSERT INTO DRDA/SHIPMENTLN VALUES(1,2,'DB2B',1,2,'24100',10,1)

INSERT INTO DRDA/SHIPMENTLN VALUES(2,1,'SQLA',2,1,'14070',4,4)

INSERT INTO DRDA/SHIPMENTLN VALUES(2,2,'SQLA',2,2,'37040',45,25)

INSERT INTO DRDA/SHIPMENTLN VALUES(2,3,'SQLA',2,3,'14030',5,5)

INSERT INTO DRDA/SHIPMENTLN VALUES(3,1,'SQLA',2,3,'14030',5,5)

```

図 34. テーブルへのデータの挿入 (続き)

DRDA の例: RPG プログラム

この例のプログラムは、RPG プログラミング言語で作成されています。

注: コードの例を使用することは、443 ページの『コードに関するライセンス情報および特記事項』の条件に同意したものとされます。

```

100 *****
200 *
300 *   DESCRIPTIVE NAME = D-DB SAMPLE APPLICATION
400 *   REORDER POINT PROCESSING
500 *   IBM i
600 *
700 *   FUNCTION = THIS MODULE PROCESSES THE PART_STOCK TABLE AND
800 *   FOR EACH PART BELOW THE ROP (REORDER POINT)
900 *   CREATES A SUPPLY ORDER AND PRINTS A REPORT.
1000 *
1100 *
1200 *   INPUT = PARAMETERS EXPLICITLY PASSED TO THIS FUNCTION:
1300 *
1400 *   LOCADB      LOCAL DB NAME
1500 *   REMODB      REMOTE DB NAME
1600 *
1700 *   TABLES = PART-STOCK      - LOCAL
1800 *   PART_ORDER      - REMOTE
1900 *   PART_ORDLN      - REMOTE
2000 *   SHIPMENTLN      - REMOTE
2100 *
2200 *   INDICATORS = *IN89      - '0' ORDER HEADER NOT DONE
2300 *   '1' ORDER HEADER IS DONE
2400 *   *IN99      - '1' ABNORMAL END (SQLCOD<0)
2500 *
2600 *   TO BE COMPILED WITH COMMIT(*CHG) RDB(remotedbname)
2700 *
2800 *   INVOKE BY : CALL DDBPT6RG PARM(localdbname remotedbname)
2900 *
3000 *   CURSORS WILL BE CLOSED IMPLICITLY (BY CONNECT) BECAUSE
3100 *   THERE IS NO REASON TO DO IT EXPLICITLY
3200 *
3300 *****
3400 *
3500 FQPRINT 0 F 33 OF PRINTER
3600 F*
3700 I*
3800 IMISC DS
3900 I B 1 20SHORTB
4000 I B 3 60LONGB
4100 I B 7 80INDNUL
4200 I 9 13 PRTTBL
4300 I 14 29 LOCTBL
4400 I I 'SQLA' 30 33 LOC
4500 I*
4600 I*
4700 C*
4800 C *LIKE DEFN SHORTB NXTORD NEW ORDER NR
4900 C *LIKE DEFN SHORTB NXTORL ORDER LINE NR
5000 C *LIKE DEFN SHORTB RTCOD1 RTCOD NEXT_PART
5100 C *LIKE DEFN SHORTB RTCOD2 RTCOD NEXT_ORD
5200 C *LIKE DEFN SHORTB CURORD ORDER NUMBER
5300 C *LIKE DEFN SHORTB CURORL ORDER LINE
5400 C *LIKE DEFN LONGB QUANTI FOR COUNTING
5500 C *LIKE DEFN LONGB QTYSTC QTY ON STOCK
5600 C *LIKE DEFN LONGB QTYORD REORDER QTY

```

図 35. RPG プログラム例

```

5700      C          *LIKE      DEFN LONGB      QTYROP          REORDER POINT      00/00/00
5800      C          *LIKE      DEFN LONGB      QTYREQ          QTY ORDERED        00/00/00
5900      C          *LIKE      DEFN LONGB      QTYREC          QTY RECEIVED       00/00/00
6000      C*
6100      C*
6200      C*****
6300      C*      PARAMETERS                                *      00/00/00
6400      C*****
6500      C*
6600      C          *ENTRY      PLIST
6700      C                      PARM              LOCADB 18          LOCAL DATABASE      00/00/00
6800      C                      PARM              REMODB 18          REMOTE DATABASE     00/00/00
6900      C*
7000      C*
7100      C*****
7200      C*      SQL CURSOR DECLARATIONS                    *      00/00/00
7300      C*****
7400      C*
7500      C* NEXT PART WHICH STOCK QUANTITY IS UNDER REORDER POINTS QTY 00/00/00
7600      C/EXEC SQL
7700      C+      DECLARE NEXT_PART CURSOR FOR
7800      C+          SELECT PART_NUM,
7900      C+                      PART_QUANT,
8000      C+                      PART_ROP,
8100      C+                      PART_EOQ
8200      C+          FROM PART_STOCK
8300      C+          WHERE PART_ROP > PART_QUANT
8400      C+          AND PART_NUM > :PRTTBL
8500      C+          ORDER BY PART_NUM ASC
8600      C/END-EXEC
8700      C*
8800      C* ORDERS WHICH ARE ALREADY MADE FOR CURRENT PART 00/00/00
8900      C/EXEC SQL
9000      C+      DECLARE NEXT_ORDER_LINE CURSOR FOR
9100      C+          SELECT A.ORDER_NUM,
9200      C+                      ORDER_LINE,
9300      C+                      QUANT_REQ
9400      C+          FROM PART_ORDLN A,
9500      C+                      PART_ORDER B
9600      C+          WHERE PART_NUM = :PRTTBL
9700      C+          AND LINE_STAT <> 'C'
9800      C+          AND A.ORDER_NUM = B.ORDER_NUM
9900      C+          AND ORDER_TYPE = 'R'
10000     C/END-EXEC
10100     C*
10200     C*****
10300     C*      SQL RETURN CODE HANDLING                        *      00/00/00
10400     C*****
10500     C/EXEC SQL
10600     C+      WHENEVER SQLERROR GO TO DBERRO
10700     C/END-EXEC
10800     C/EXEC SQL
10900     C+      WHENEVER SQLWARNING CONTINUE
11000     C/END-EXEC
11100     C*
11200     C*

```



```

11300 C*****
11400 C*   PROCESS - MAIN PROGRAM LOGIC *
11500 C*   MAIN PROCEDURE WORKS WITH LOCAL DATABASE *
11600 C*****
11700 C*
11800 C*CLEAN UP TO PERMIT RE-RUNNING OF TEST DATA
11900 C           EXSR CLEANU
12000 C*
12100 C*
12200 C           RTCOD1   DOUEQ100
12300 C*
12400 C/EXEC SQL
12500 C+   CONNECT TO :LOCADB
12600 C/END-EXEC
12700 C/EXEC SQL
12800 C+   OPEN NEXT_PART
12900 C/END-EXEC
13000 C/EXEC SQL
13100 C+   FETCH NEXT_PART
13200 C+           INTO :PRTTBL,
13300 C+           :QTYSTC,
13400 C+           :QTYROP,
13500 C+           :QTYORD
13600 C/END-EXEC
13700 C           MOVE SQLCOD RTCOD1
13800 C/EXEC SQL
13900 C+   COMMIT
14000 C/END-EXEC
14100 C           RTCOD1   IFNE 100
14200 C           EXSR CHECKO
14300 C           ENDIF
14400 C*
14500 C           ENDDO
14600 C*
14700 C           GOTO SETLR
14800 C*
14900 C*
15000 C*****
15100 C*   SQL RETURN CODE HANDLING ON ERROR SITUATIONS *
15200 C*****
15300 C*
15400 C           DBERRO TAG
15500 C*   *-----*
15600 C           EXCPTERRLIN
15700 C           MOVE *ON *IN99
15800 C/EXEC SQL
15900 C+   WHENEVER SQLERROR CONTINUE
16000 C/END-EXEC
16100 C/EXEC SQL
16200 C+   ROLLBACK
16300 C/END-EXEC
16400 C/EXEC SQL

```

```

16500 C+          WHENEVER  SQLERROR  GO TO  DBERR0          00/00/00
16600 C/END-EXEC                                         00/00/00
16700 C*                                                  00/00/00
16800 C*                                                  00/00/00
16900 C          SETLR      TAG                               00/00/00
17000 C*          *-----*                               00/00/00
17100 C/EXEC SQL                                         00/00/00
17200 C+          CONNECT  RESET                           00/00/00
17300 C/END-EXEC                                         00/00/00
17400 C          MOVE *ON          *INLR                   00/00/00
17500 C*                                                  00/00/00
17600 C*****                                             00/00/00
17700 C*          THE END OF THE PROGRAM                    *          00/00/00
17800 C*****                                             00/00/00
17900 C*                                                  00/00/00
18000 C*                                                  00/00/00
18100 C*****                                             00/00/00
18200 C*          SUBROUTINES TO WORK WITH REMOTE DATABASES *          00/00/00
18300 C*****                                             00/00/00
18400 C*                                                  00/00/00
18500 C*                                                  00/00/00
18600 C          CHECKO    BEGSR                             00/00/00
18700 C*          *-----*                               00/00/00
18800 C*****                                             00/00/00
18900 C*          CHECKS WHAT IS CURRENT ORDER AND SHIPMENT STATUS FOR THE PART.* 00/00/00
19000 C*          IF ORDERED AND SHIPPED IS LESS THAN REORDER POINT OF PART, *    00/00/00
19100 C*          PERFORMS A SUBROUTINE WHICH MAKES AN ORDER. *          00/00/00
19200 C*****                                             00/00/00
19300 C*                                                  00/00/00
19400 C          MOVE 0          RTCOD2                       00/00/00
19500 C          MOVE 0          QTYREQ                      00/00/00
19600 C          MOVE 0          QTYREC                      00/00/00
19700 C*                                                  00/00/00
19800 C/EXEC SQL                                         00/00/00
19900 C+          CONNECT  TO      :REMOB                     00/00/00
20000 C/END-EXEC                                         00/00/00
20100 C/EXEC SQL                                         00/00/00
20200 C+          OPEN      NEXT_ORDER_LINE                   00/00/00
20300 C/END-EXEC                                         00/00/00
20400 C*                                                  00/00/00
20500 C          RTCOD2    DOWNE100                           00/00/00
20600 C*                                                  00/00/00
20700 C/EXEC SQL                                         00/00/00
20800 C+          FETCH     NEXT_ORDER_LINE                   00/00/00
20900 C+          INTO      :CURORD,                          00/00/00
21000 C+          :CURORL,                                    00/00/00
21100 C+          :QUANTI                                       00/00/00
21200 C/END-EXEC                                         00/00/00
21300 C*                                                  00/00/00

```

```

21400 C          SQLCOD  IFEQ 100          00/00/00
21500 C          MOVE 100      RTCOD2      00/00/00
21600 C          ELSE          00/00/00
21700 C          ADD QUANTI    QTYREQ      00/00/00
21800 C*          00/00/00
21900 C/EXEC SQL          00/00/00
22000 C+          SELECT      SUM(QUANT_RECV) 00/00/00
22100 C+          INTO       :QUANTI:INDNUL 00/00/00
22200 C+          FROM       SHIPMENTLN     00/00/00
22300 C+          WHERE      ORDER_LOC = :LOC 00/00/00
22400 C+          AND        ORDER_NUM = :CURORD 00/00/00
22500 C+          AND        ORDER_LINE = :CURORL 00/00/00
22600 C/END-EXEC          00/00/00
22700 C*          00/00/00
22800 C          INDNUL      IFGE 0          00/00/00
22900 C          ADD QUANTI    QTYREC      00/00/00
23000 C          ENDIF          00/00/00
23100 C*          00/00/00
23200 C          ENDIF          00/00/00
23300 C          ENDDO          00/00/00
23400 C*          00/00/00
23500 C/EXEC SQL          00/00/00
23600 C+          CLOSE NEXT_ORDER_LINE     00/00/00
23700 C/END-EXEC          00/00/00
23800 C*          00/00/00
23900 C          QTYSTC      ADD QTYREQ    QUANTI 00/00/00
24000 C          SUB QUANTI    QTYREC      00/00/00
24100 C*          00/00/00
24200 C          QTYROP      IFGT QUANTI    00/00/00
24300 C          EXSR ORDERP  00/00/00
24400 C          ENDIF          00/00/00
24500 C*          00/00/00
24600 C/EXEC SQL          00/00/00
24700 C+          COMMIT          00/00/00
24800 C/END-EXEC          00/00/00
24900 C*          00/00/00
25000 C          ENDSR          CHECKO      00/00/00
25100 C*          00/00/00
25200 C*          00/00/00
25300 C          ORDERP      BEGSR          00/00/00
25400 C*          *-----*          00/00/00
25500 C*****          00/00/00
25600 C* MAKES AN ORDER. IF FIRST TIME, PERFORMS THE SUBROUTINE, WHICH * 00/00/00
25700 C* SEARCHES FOR NEW ORDER NUMBER AND MAKES THE ORDER HEADER. * 00/00/00
25800 C* AFTER THAT, MAKES ORDER LINES USING REORDER QUANTITY FOR THE * 00/00/00
25900 C* PART. FOR EVERY ORDERED PART, WRITES A LINE ON REPORT. * 00/00/00
26000 C*****          00/00/00

```

```

26100 C* 00/00/00
26200 C *IN89 IFEQ *OFF FIRST ORDER ? 00/00/00
26300 C EXSR STORD 00/00/00
26400 C MOVE *ON *IN89 ORD.HEAD.DONE 00/00/00
26500 C EXCPTHEADER WRITE HEADERS 00/00/00
26600 C ENDIF 00/00/00
26700 C* 00/00/00
26800 C ADD 1 NXTORL NEXT ORD.LIN 00/00/00
26900 C/EXEC SQL 00/00/00
27000 C+ INSERT 00/00/00
27100 C+ INTO PART_ORDLN 00/00/00
27200 C+ (ORDER_NUM, 00/00/00
27300 C+ ORDER_LINE, 00/00/00
27400 C+ PART_NUM, 00/00/00
27500 C+ QUANT_REQ, 00/00/00
27600 C+ LINE_STAT) 00/00/00
27700 C+ VALUES (:NXTORD, 00/00/00
27800 C+ :NXTORL, 00/00/00
27900 C+ :PRTTBL, 00/00/00
28000 C+ :QTYORD, 00/00/00
28100 C+ '0') 00/00/00
28200 C/END-EXEC 00/00/00
28300 C* 00/00/00
28400 C *INOF IFEQ *ON 00/00/00
28500 C EXCPTHEADER 00/00/00
28600 C END 00/00/00
28700 C EXCPTDETAIL 00/00/00
28800 C* 00/00/00
28900 C ENDSR ORDERP 00/00/00
29000 C* 00/00/00
29100 C* 00/00/00
29200 C STORD BEGSR 00/00/00
29300 C* *-----* 00/00/00
29400 C***** 00/00/00
29500 C* SEARCHES FOR NEXT ORDER NUMBER AND MAKES AN ORDER HEADER * 00/00/00
29600 C* USING THAT NUMBER. WRITES ALSO HEADERS ON REPORT. * 00/00/00
29700 C***** 00/00/00
29800 C* 00/00/00
29900 C/EXEC SQL 00/00/00
30000 C+ SELECT (MAX(ORDER_NUM) + 1) 00/00/00
30100 C+ INTO :NXTORD 00/00/00
30200 C+ FROM PART_ORDER 00/00/00
30300 C/END-EXEC 00/00/00
30400 C/EXEC SQL 00/00/00
30500 C+ INSERT 00/00/00
30600 C+ INTO PART_ORDER 00/00/00
30700 C+ (ORDER_NUM, 00/00/00
30800 C+ ORIGIN_LOC, 00/00/00

```

```

30900 C+ ORDER_TYPE, 00/00/00
31000 C+ ORDER_STAT, 00/00/00
31100 C+ CREAT_TIME) 00/00/00
31200 C+ VALUES (:NXTORD, 00/00/00
31300 C+ :LOC, 00/00/00
31400 C+ 'R', 00/00/00
31500 C+ 'O', 00/00/00
31600 C+ CURRENT_TIMESTAMP) 00/00/00
31700 C/END-EXEC 00/00/00
31800 C ENDSR STRORD 00/00/00
31900 C* 00/00/00
32000 C* 00/00/00
32100 C CLEANU BEGSR 00/00/00
32200 C* *-----* 00/00/00
32300 C***** 00/00/00
32400 C* THIS SUBROUTINE IS ONLY REQUIRED IN A TEST ENVIRONMENT 00/00/00
32500 C* TO RESET THE DATA TO PERMIT RE-RUNNING OF THE TEST 00/00/00
32600 C***** 00/00/00
32700 C* 00/00/00
32800 C/EXEC SQL 00/00/00
32900 C+ CONNECT TO :REMO DB 00/00/00
33000 C/END-EXEC 00/00/00
33100 C/EXEC SQL 00/00/00
33200 C+ DELETE 00/00/00
33300 C+ FROM PART_ORDLN 00/00/00
33400 C+ WHERE ORDER_NUM IN 00/00/00
33500 C+ (SELECT ORDER_NUM 00/00/00
33600 C+ FROM PART_ORDER 00/00/00
33700 C+ WHERE ORDER_TYPE = 'R') 00/00/00
33800 C/END-EXEC 00/00/00
33900 C/EXEC SQL 00/00/00
34000 C+ DELETE 00/00/00
34100 C+ FROM PART_ORDER 00/00/00
34200 C+ WHERE ORDER_TYPE = 'R' 00/00/00
34300 C/END-EXEC 00/00/00
34400 C/EXEC SQL 00/00/00
34500 C+ COMMIT 00/00/00
34600 C/END-EXEC 00/00/00
34700 C* 00/00/00
34800 C ENDSR CLEANU 00/00/00
34900 C* 00/00/00
35000 C* 00/00/00
35100 C***** 00/00/00
35200 O* OUTPUT LINES FOR THE REPORT * 00/00/00
35300 O***** 00/00/00
35400 O* 00/00/00
35500 OQPRINT E 2 HEADER 00/00/00
35600 O + 0 '***** ROP PROCESSING' 00/00/00

```

```

35700      0                      + 1 'REPORT *****'          00/00/00
35800      0*                      00/00/00
35900      QQPRINT  E 2            HEADER                          00/00/00
36000      0                      + 0 '  ORDER NUMBER = '      00/00/00
36100      0                      NXTORDZ + 0                    00/00/00
36200      0*                      00/00/00
36300      QQPRINT  E 1            HEADER                          00/00/00
36400      0                      + 0 '-----'                00/00/00
36500      0                      + 0 '-----'                00/00/00
36600      0*                      00/00/00
36700      QQPRINT  E 1            HEADER                          00/00/00
36800      0                      + 0 '  LINE      '          00/00/00
36900      0                      + 0 'PART        '          00/00/00
37000      0                      + 0 'QTY      '          00/00/00
37100      0*                      00/00/00
37200      QQPRINT  E 1            HEADER                          00/00/00
37300      0                      + 0 '  NUMBER  '          00/00/00
37400      0                      + 0 'NUMBER    '          00/00/00
37500      0                      + 0 'REQUESTED '          00/00/00
37600      0*                      00/00/00
37700      QQPRINT  E 11           HEADER                          00/00/00
37800      0                      + 0 '-----'                00/00/00
37900      0                      + 0 '-----'                00/00/00
38000      0*                      00/00/00
38100      QQPRINT  EF1            DETAIL                          00/00/00
38200      0                      NXTORLZ + 4                    00/00/00
38300      0                      PRTTBL  + 4                    00/00/00
38400      0                      QTYORD1 + 4                    00/00/00
38500      0*                      00/00/00
38600      QQPRINT  T 2            LRN99                          00/00/00
38700      0                      + 0 '-----'                00/00/00
38800      0                      + 0 '-----'                00/00/00
38900      QQPRINT  T 1            LRN99                          00/00/00
39000      0                      + 0 'NUMBER OF LINES '      00/00/00
39100      0                      + 0 'CREATED = '          00/00/00
39200      0                      NXTORLZ + 0                    00/00/00
39300      0*                      00/00/00
39400      QQPRINT  T 1            LRN99                          00/00/00
39500      0                      + 0 '-----'                00/00/00
39600      0                      + 0 '-----'                00/00/00
39700      0*                      00/00/00
39800      QQPRINT  T 2            LRN99                          00/00/00
39900      0                      + 0 '*****'              00/00/00
40000      0                      + 0 ' END OF PROGRAM '      00/00/00
40100      0                      + 0 '*****'              00/00/00
40200      0*                      00/00/00
40300      QQPRINT  E 2            ERRRLIN                        00/00/00
40400      0                      + 0 '** ERROR **'          00/00/00
40500      0                      + 0 '** ERROR **'          00/00/00
40600      0                      + 0 '** ERROR **'          00/00/00
40700      QQPRINT  E 1            ERRRLIN                        00/00/00
40800      0                      + 0 '* SQLCOD: '          00/00/00
40900      0                      SQLCODM + 0                    00/00/00
41000      0                      33 '* '                        00/00/00
41100      QQPRINT  E 1            ERRRLIN                        00/00/00
41200      0                      + 0 '* SQLSTATE: '        00/00/00
41300      0                      SQLSTT  + 2                    00/00/00
41400      0                      33 '* '                        00/00/00
41500      QQPRINT  E 1            ERRRLIN                        00/00/00
41600      0                      + 0 '** ERROR **'          00/00/00
41700      0                      + 0 '** ERROR **'          00/00/00
41800      0                      + 0 '** ERROR **'          00/00/00

```

DRDA の例: COBOL プログラム

この例のプログラムは、COBOL プログラミング言語で作成されています。

注: コードの例を使用することは、443 ページの『コードに関するライセンス情報および特記事項』の条件に同意したものとされます。

```

100      IDENTIFICATION DIVISION.
200      *-----*
300      PROGRAM-ID. DDBPT6CB.                                00/00/00
400      *****
500      *   MODULE NAME = DDBPT6CB                          00/00/00
600      *
700      *   DESCRIPTIVE NAME = D-DB SAMPLE APPLICATION
800      *                                   REORDER POINT PROCESSING
900      *                                   IBM i                00/00/00
1000     *                                   COBOL
1100     *
1200     *   FUNCTION = THIS MODULE PROCESSES THE PART_STOCK TABLE AND
1300     *                   FOR EACH PART BELOW THE ROP (REORDER POINT)
1400     *                   CHECKS THE EXISTING ORDERS AND SHIPMENTS,    00/00/00
1500     *                   CREATES A SUPPLY ORDER AND PRINTS A REPORT.  00/00/00
1600     *
1700     *   DEPENDENCIES = NONE                                         00/00/00
1800     *
1900     *   INPUT = PARAMETERS EXPLICITLY PASSED TO THIS FUNCTION:
2000     *
2100     *           LOCAL-DB      LOCAL DB NAME                00/00/00
2200     *           REMOTE-DB     REMOTE DB NAME               00/00/00
2300     *
2400     *   TABLES = PART-STOCK      - LOCAL                00/00/00
2500     *                   PART_ORDER - REMOTE              00/00/00
2600     *                   PART_ORDLN - REMOTE              00/00/00
2700     *                   SHIPMENTLN - REMOTE              00/00/00
2800     *
2900     *   CRTSQLCBL SPECIAL PARAMETERS                            00/00/00
3000     *   PGM(DDBPT6CB) RDB(remotedbname) OPTION(*APOST *APOSTSQL)  00/00/00
3100     *
3200     *   INVOKE BY : CALL DDBPT6CB PARM(localdbname remotedbname)  00/00/00
3300     *
3400     *****
3500     ENVIRONMENT DIVISION.
3600     *-----*
3700     INPUT-OUTPUT SECTION.
3800     FILE-CONTROL.
3900     SELECT RELAT ASSIGN TO PRINTER-QPRINT.                    00/00/00
4000     DATA DIVISION.
4100     *-----*
4200     FILE SECTION.
4300     *-----*                                00/00/00
4400     FD RELAT
4500     RECORD CONTAINS 33 CHARACTERS
4600     LABEL RECORDS ARE OMITTED
4700     DATA RECORD IS REPREC.
4800     01 REPREC          PIC X(33).
4900     WORKING-STORAGE SECTION.
5000     *-----*                                00/00/00
5100     *   PRINT LINE DEFINITIONS                                    00/00/00
5200     01 LINE0           PIC X(33) VALUE SPACES.
5300     01 LINE1           PIC X(33) VALUE

```

図 36. COBOL プログラム例

```

5400      '***** ROP PROCESSING REPORT *****'.
5500      01 LINE2.
5600          05 FILLER          PIC X(18) VALUE ' ORDER NUMBER = '.
5700          05 MASK0          PIC ZZZ9.
5800          05 FILLER          PIC X(11) VALUE SPACES.
5900      01 LINE3          PIC X(33) VALUE
6000          '-----'.
6100      01 LINE4          PIC X(33) VALUE
6200          ' LINE PART QTY '.
6300      01 LINE5          PIC X(33) VALUE
6400          ' NUMBER NUMBER REQUESTED '.
6500      01 LINE6.
6600          05 FILLER          PIC XXXX VALUE SPACES.
6700          05 MASK1          PIC ZZZ9.
6800          05 FILLER          PIC XXXX VALUE SPACES.
6900          05 PART-TABLE     PIC XXXXX.
7000          05 FILLER          PIC XXXX VALUE SPACES.
7100          05 MASK2          PIC Z,ZZZ,ZZ.ZZ.
7200      01 LINE7.
7300          05 FILLER          PIC X(26) VALUE
7400          'NUMBER OF LINES CREATED = '.
7500          05 MASK3          PIC ZZZ9.
7600          05 FILLER          PIC XXX VALUE SPACES.
7700      01 LINE8          PIC X(33) VALUE
7800          '***** END OF PROGRAM *****'.
7900      * MISCELLANEOUS DEFINITIONS                                00/00/00
8000      01 WHAT-TIME     PIC X VALUE '1'.
8100          88 FIRST-TIME VALUE '1'.
8200      01 CONTL          PIC S9999 COMP-4 VALUE ZEROS.          00/00/00
8300      01 CONTD          PIC S9999 COMP-4 VALUE ZEROS.          00/00/00
8400      01 RTCODE1       PIC S9999 COMP-4 VALUE ZEROS.          00/00/00
8500      01 RTCODE2       PIC S9999 COMP-4.                      00/00/00
8600      01 NEXT-NUM      PIC S9999 COMP-4.                      00/00/00
8700      01 IND-NULL      PIC S9999 COMP-4.                      00/00/00
8800      01 LOC-TABLE     PIC X(16).
8900      01 ORD-TABLE     PIC S9999 COMP-4.                      00/00/00
9000      01 ORL-TABLE     PIC S9999 COMP-4.                      00/00/00
9100      01 QUANT-TABLE   PIC S9(9) COMP-4.                      00/00/00
9200      01 QTY-TABLE     PIC S9(9) COMP-4.                      00/00/00
9300      01 ROP-TABLE     PIC S9(9) COMP-4.                      00/00/00
9400      01 EOQ-TABLE     PIC S9(9) COMP-4.                      00/00/00
9500      01 QTY-REQ       PIC S9(9) COMP-4.                      00/00/00
9600      01 QTY-REC       PIC S9(9) COMP-4.                      00/00/00
9700      * CONSTANT FOR LOCATION NUMBER                          00/00/00
9800      01 XPARAM.                                             00/00/00
9900          05 LOC          PIC X(4) VALUE 'SQLA'.              00/00/00
10000     * DEFINITIONS FOR ERROR MESSAGE HANDLING              00/00/00
10100     01 ERROR-MESSAGE.                                     00/00/00
10200         05 MSG-ID.                                         00/00/00
10300         10 MSG-ID-1     PIC X(2)                            00/00/00
10400             VALUE 'SQ'.                                     00/00/00
10500         10 MSG-ID-2     PIC 99999.                          00/00/00

```


15500	*-----	00/00/00
15600	****	00/00/00
15700	EXEC SQL CONNECT RESET END-EXEC.	00/00/00
15800	****	
15900	CLOSE RELAT.	
16000	GOBACK.	
16100	MAIN-PROGRAM-EXIT. EXIT.	00/00/00
16200	*-----	00/00/00
16300		00/00/00
16400	START-UP.	00/00/00
16500	*-----	00/00/00
16600	OPEN OUTPUT RELAT.	00/00/00
16700	****	00/00/00
16800	EXEC SQL COMMIT END-EXEC.	00/00/00
16900	****	00/00/00
17000	PERFORM CLEAN-UP THRU CLEAN-UP-EXIT.	00/00/00
17100	*****	00/00/00
17200	* CONNECT TO LOCAL DATABASE *	00/00/00
17300	*****	00/00/00
17400	****	00/00/00
17500	EXEC SQL CONNECT TO :LOCAL-DB END-EXEC.	00/00/00
17600	****	00/00/00
17700	START-UP-EXIT. EXIT.	00/00/00
17800	*-----	00/00/00
17900	EJECT	
18000	MAIN-PROC.	
18100	*-----	
18200	EXEC SQL OPEN NEXT_PART END-EXEC.	00/00/00
18300	EXEC SQL	
18400	FETCH NEXT_PART	
18500	INTO :PART-TABLE,	
18600	:QUANT-TABLE,	
18700	:ROP-TABLE,	
18800	:EQQ-TABLE	
18900	END-EXEC.	
19000	IF SQLCODE = 100	
19100	MOVE 100 TO RTCODE1	00/00/00
19200	PERFORM TRAILER-PROC THRU TRAILER-EXIT	00/00/00
19300	ELSE	
19400	MOVE 0 TO RTCODE2	
19500	MOVE 0 TO QTY-REQ	
19600	MOVE 0 TO QTY-REC	
19700	* --- IMPLICIT "CLOSE" CAUSED BY COMMIT ---	00/00/00
19800	****	00/00/00
19900	EXEC SQL COMMIT END-EXEC	00/00/00
20000	****	00/00/00
20100	*****	00/00/00
20200	* CONNECT TO REMOTE DATABASE *	00/00/00
20300	*****	00/00/00

```

20400      ****                                00/00/00
20500      EXEC SQL CONNECT TO :REMOTE-DB END-EXEC 00/00/00
20600      ****                                00/00/00
20700      EXEC SQL OPEN NEXT_ORDER_LINE END-EXEC 00/00/00
20800      PERFORM UNTIL RTCODE2 = 100
20900          EXEC SQL                                00/00/00
21000              FETCH NEXT_ORDER_LINE
21100              INTO :ORD-TABLE,
21200                  :ORL-TABLE,
21300                  :QTY-TABLE
21400          END-EXEC
21500          IF SQLCODE = 100
21600              MOVE 100 TO RTCODE2
21700          EXEC SQL CLOSE NEXT_ORDER_LINE END-EXEC
21800          ELSE
21900              ADD QTY-TABLE TO QTY-REQ
22000              EXEC SQL
22100                  SELECT SUM(QUANT_RECV)          00/00/00
22200                  INTO :QTY-TABLE:IND-NULL
22300                  FROM SHIPMENTLN                00/00/00
22400                  WHERE ORDER_LOC = :LOC
22500                  AND ORDER_NUM = :ORD-TABLE
22600                  AND ORDER_LINE = :ORL-TABLE
22700              END-EXEC
22800              IF IND-NULL NOT < 0
22900                  ADD QTY-TABLE TO QTY-REC
23000              END-IF
23100          END-IF
23200          END-PERFORM
23300          IF ROP-TABLE > QUANT-TABLE + QTY-REQ - QTY-REC
23400              PERFORM ORDER-PROC THRU ORDER-EXIT
23500          END-IF
23600      END-IF.
23700      ****                                00/00/00
23800      EXEC SQL COMMIT END-EXEC.                00/00/00
23900      ****                                00/00/00
24000      *****
24100      * RECONNECT TO LOCAL DATABASE *          00/00/00
24200      *****
24300      ****                                00/00/00
24400      EXEC SQL CONNECT TO :LOCAL-DB END-EXEC. 00/00/00
24500      ****                                00/00/00
24600      MAIN-EXIT. EXIT.
24700      *-----
24800      ORDER-PROC.
24900      *-----
25000          IF FIRST-TIME
25100              MOVE '2' TO WHAT-TIME
25200              PERFORM CREATE-ORDER-PROC THRU CREATE-ORDER-EXIT. 00/00/00
25300          ADD 1 TO CNTL.

```

```

25400      EXEC SQL
25500          INSERT
25600              INTO      PART_ORDLN
25700                  (ORDER_NUM,
25800                   ORDER_LINE,
25900                   PART_NUM,
26000                   QUANT_REQ,
26100                   LINE_STAT)
26200          VALUES (:NEXT-NUM,
26300                  :CONTL,
26400                  :PART-TABLE,
26500                  :EQQ-TABLE,
26600                  '0')
26700      END-EXEC.
26800      PERFORM DETAIL-PROC THRU DETAIL-EXIT.
26900      ORDER-EXIT. EXIT.
27000      *-----
27100
27200      CREATE-ORDER-PROC.
27300      *-----
27400      *GET NEXT ORDER NUMBER
27500      EXEC SQL
27600          SELECT (MAX(ORDER_NUM) + 1)
27700              INTO :NEXT-NUM:IND-NULL
27800              FROM PART_ORDER
27900      END-EXEC.
28000      IF IND-NULL < 0
28100          MOVE 1 TO NEXT-NUM.
28200      EXEC SQL
28300          INSERT
28400              INTO      PART_ORDER
28500                  (ORDER_NUM,
28600                   ORIGIN_LOC,
28700                   ORDER_TYPE,
28800                   ORDER_STAT,
28900                   CREAT_TIME)
29000          VALUES (:NEXT-NUM,
29100                  :LOC, 'R', '0',
29200                  CURRENT_TIMESTAMP)
29300      END-EXEC.
29400      MOVE NEXT-NUM TO MASK0.
29500      PERFORM HEADER-PROC THRU HEADER-EXIT.
29600      CREATE-ORDER-EXIT. EXIT.
29700      *-----
29800
29900      DB-ERROR.
30000      *-----
30100          PERFORM ERROR-MSG-PROC THRU ERROR-MSG-EXIT.
30200      *****
30300      *   ROLLBACK THE LUW *

```

00/00/00

00/00/00

00/00/00

00/00/00

00/00/00

00/00/00

00/00/00

00/00/00

00/00/00

00/00/00

00/00/00

00/00/00

00/00/00

00/00/00

00/00/00

00/00/00

00/00/00

00/00/00

00/00/00

00/00/00

00/00/00

00/00/00

00/00/00

00/00/00

00/00/00

00/00/00

00/00/00

00/00/00

00/00/00

00/00/00

00/00/00

00/00/00

00/00/00

00/00/00

00/00/00


```

35600 *****
35700 * THIS PARAGRAPH IS ONLY REQUIRED IN A TEST ENVIRONMENT*
35800 * TO RESET THE DATA TO PERMIT RE-RUNNING OF THE TEST *
35900 *****
36000 CLEAN-UP.
36100 *-----
36200 *****
36300 * CONNECT TO REMOTE DATABASE *
36400 *****
36500 ****
36600 EXEC SQL CONNECT TO :REMOTE-DB END-EXEC.
36700 ****
36800 *-----DELETE ORDER ROWS FOR RERUNABILITY
36900 EXEC SQL
37000 DELETE
37100 FROM PART_ORDLN
37200 WHERE ORDER_NUM IN
37300 (SELECT ORDER_NUM
37400 FROM PART_ORDER
37500 WHERE ORDER_TYPE = 'R')
37600 END-EXEC.
37700 EXEC SQL
37800 DELETE
37900 FROM PART_ORDER
38000 WHERE ORDER_TYPE = 'R'
38100 END-EXEC.
38200 ****
38300 EXEC SQL COMMIT END-EXEC.
38400 ****
38500 CLEAN-UP-EXIT. EXIT.
38600 *-----

```

* * * * ソース仕様の終わり * * * *

DRDA の例: 組み込み SQL を使用する C プログラム

この例のプログラムは、C プログラミング言語で作成されています。

注: コードの例を使用することは、443 ページの『コードに関するライセンス情報および特記事項』の条件に同意したものとされます。

```

/*****
/* PROGRAM NAME: SAMPEMBC */
/* */
/* DESCRIPTIVE NAME: Sample embedded C application using DRDA */
/* */
/* FUNCTION: This module processes the PART_STOCK table and */
/* for each part below the ROP (REORDER POINT) */
/* creates a supply order. */
/* */
/* LOCAL TABLES: PART_STOCK */
/* */
/* REMOTE TABLES: PART_ORDER, PART_ORDLN, SHIPMENTLN */
/* */
/* COMPILE OPTIONS: */
/* CRTSQLCI OBJ(SAMPEMBC) COMMIT(*CHG) RDB(rdbname) OBJTYPE(*PGM) */
/* RDBCNNMTH(*RUW) */
/* */
/* INVOKED BY: */
/* CALL PGM(SAMPEMBC) PARM('lcldbname' 'rmtdbname') */
*****/
#include <stdlib.h>
#include <string.h>
#include <stdio.h>

EXEC SQL INCLUDE SQLCA;

/*****
/* SQL Variables */
*****/
EXEC SQL BEGIN DECLARE SECTION;
char loc[4] = "SQLA"; /* dealer's database name */
char remote_db[18] = " "; /* sample remote database */
char local_db[18] = " "; /* sample local database */
char part_table[5] = " "; /* part number in table part_stock */
long quant_table; /* quantity in stock, tbl part_stock */
long rop_table; /* reorder point , tbl part_stock */
long eqq_table; /* reorder quantity , tbl part_stock */
short next_num; /* next order nbr, table part_order */
short ord_table; /* order nbr. , tbl order_line */
short orl_table; /* order line , tbl order_line */
long qty_table; /* ordered quantity , tbl order_line */
long line_count = 0; /* total number of order lines */
short ind_null; /* null indicator for qty_table */
short contl = 0; /* continuation line, tbl order_line */
EXEC SQL END DECLARE SECTION;

/*****
/* Other Variables */
*****/
char first_time, what_time;
long qty_rec = 0, qty_req = 0;

EXEC SQL WHENEVER SQLERROR GOTO error_tag;
EXEC SQL WHENEVER SQLWARNING CONTINUE;

```

図 37. 組み込み SQL を使用する C プログラムの例

```

/*****
/* Function Declarations */
/*****
/*****
/* Function For Declaring Cursors */
/*****
declare_cursors() {

    /* SQL Cursor declaration and reposition for local UW */
    EXEC SQL DECLARE NEXT_PART CURSOR FOR
        SELECT PART_NUM, PART_QUANT, PART_ROP, PART_EOQ
        FROM DRDA/PART_STOCK
        WHERE PART_ROP > PART_QUANT AND
            PART_NUM > :part_table
        ORDER BY PART_NUM;

    /* SQL Cursor declaration and connect for RUW */
    EXEC SQL DECLARE NEXT_ONLINE CURSOR FOR
        SELECT A.ORDER_NUM, ORDER_LINE, QUANT_REQ
        FROM DRDA/PART_ORDLN A,
            DRDA/PART_ORDER B
        WHERE PART_NUM = :part_table AND
            LINE_STAT <> 'C' AND
            A.ORDER_NUM = B.ORDER_NUM AND
            ORDER_TYPE = 'R';

    /* upline exit function in connectable state */
    goto function_exit;

error_tag:
    error_function();

function_exit:
    ;
} /* function declare_cursor */

/*****
/* Function For Resetting Tables */
/*****
reset_tables() {

    /* Clean up for rerunability in test environment */
    EXEC SQL CONNECT TO :remote_db;
    EXEC SQL DELETE FROM DRDA/PART_ORDLN
        WHERE ORDER_NUM IN
            (SELECT ORDER_NUM
            FROM DRDA/PART_ORDER
            WHERE ORDER_TYPE = 'R');
    EXEC SQL DELETE FROM DRDA/PART_ORDER
        WHERE ORDER_TYPE = 'R';
    /* Exit function in connectable state */
    EXEC SQL COMMIT;
    goto function_exit;

error_tag:
    error_function();

function_exit:
    ;
} /* function delete_for_rerun */

```



```

/*****
/* Function For Declaring Cursors */
/*****
calculate_order_quantity() {

    /* Set current connection to local database */
    EXEC SQL CONNECT TO :local_db;
    /* available qty = Stock qty + qty in order - qty received */
    EXEC SQL OPEN NEXT_PART;
    EXEC SQL FETCH NEXT_PART
        INTO :part_table, :quant_table, :rop_table, :eqq_table;

    if (sqlca.sqlcode == 100) {
        printf("-----\n");
        printf("NUMBER OF LINES CREATED = %d\n",line_count);
        printf("-----\n");
        printf("***** END OF PROGRAM *****\n");
        rop_table = 0; /* no (more) orders to process */
    }
    else {
        qty_rec = 0;
        qty_req = 0;
        EXEC SQL COMMIT;
        EXEC SQL CONNECT TO :remote_db;
        EXEC SQL OPEN NEXT_OLINE;
        do {
            EXEC SQL FETCH NEXT_OLINE
                INTO :ord_table, :orl_table, :qty_table;
            qty_rec = qty_rec + qty_table;
        } while(sqlca.sqlcode != 100);
        EXEC SQL CLOSE NEXT_OLINE;
        EXEC SQL SELECT SUM(QUANT_RECV)
            INTO :qty_table:ind_null
            FROM DRDA/SHIPMENTLN
            WHERE ORDER_LOC = :loc AND
                ORDER_NUM = :ord_table AND
                ORDER_LINE = :orl_table;
        if (ind_null != 0)
            qty_rec = qty_rec + qty_table;
    } /* end of else branch */

    goto function_exit;

error_tag:
    error_function();

function_exit:
    ;
} /* end of calculate_order_quantity */

```

```

/*****
/* Function For Declaring Cursors */
/*****
process_order() {

/* insert order and order_line in remote database */
if (contl == 0) {
EXEC SQL SELECT (MAX(ORDER_NUM) + 1)
      INTO :next_num
      FROM DRDA/PART_ORDER;
EXEC SQL INSERT INTO DRDA/PART_ORDER
      (ORDER_NUM, ORIGIN_LOC, ORDER_TYPE, ORDER_STAT, CREAT_TIME)
      VALUES (:next_num, :loc, 'R', '0', CURRENT_TIMESTAMP);
printf("***** ROP PROCESSING *****\n");
printf("ORDER NUMBER = %d \n\n",next_num);
printf("-----\n");
printf("  LINE      PART      QTY      \n");
printf("  NBR       NBR       REQUESTED\n");
printf("-----\n");
contl = contl + 1;
} /* if contl == 0 */

EXEC SQL INSERT INTO DRDA/PART_ORDLN
      (ORDER_NUM, ORDER_LINE, PART_NUM, QUANT_REQ, LINE_STAT)
      VALUES (:next_num, :contl, :part_table, :eq_table, '0');
line_count = line_count + 1;
printf("  %d      %.5s      %d\n",line_count,part_table,eq_table);
contl = contl + 1;
/* Exit function in connectable state */
EXEC SQL COMMIT;

goto function_exit;

error_tag:
error_function();

function_exit:
;
} /* end of function process_order */

```

```

/*****
/* Function For Declaring Cursors
/*****
error_function() {

    printf("*****\n");
    printf("*      SQL ERROR      *\n");
    printf("*****\n");
    printf("SQLCODE   = %d\n",sqlca.sqlcode);
    printf("SQLSTATE   = %5s",sqlca.sqlstate);
    printf("\n*****\n");
    EXEC SQL WHENEVER SQLERROR CONTINUE;
    EXEC SQL ROLLBACK;
    /* Reset Current Connection To Local Database
    EXEC SQL CONNECT RESET;

    exit(999);
} /* end of function error_function

/*****
/* Mainline
/*****
main(int argc, char *argv[]) {

    memcpy(local_db,argv[1],strlen(argv[1]));
    memcpy(remote_db,argv[2],strlen(argv[2]));

    /* Initialization
    declare_cursors();
    reset_tables();

    /* Main Work
    do {
        calculate_order_quantity();
        if (rop_table > quant_table + qty_req - qty_rec) {
            process_order();
            quant_table = qty_req = qty_rec = 0;
        }
    } while (sqlca.sqlcode == 0);
    EXEC SQL COMMIT;
    /* Reset Current Connection To Local Database
    EXEC SQL DISCONNECT :local_db;

    exit(0);
} /* end of main

```

DRDA の例: Java プログラム

この例のプログラムは、Java プログラミング言語で作成されています。

注: コードの例を使用することは、443 ページの『コードに関するライセンス情報および特記事項』の条件に同意したものとされます。

```

/*****
/* PROGRAM NAME: SampJava */
/* */
/* DESCRIPTIVE NAME: Sample Java application using DRDA */
/* */
/* FUNCTION: This module processes the PART_STOCK table and */
/* for each part below the ROP (REORDER POINT) */
/* creates a supply order. */
/* */
/* LOCAL TABLES: PART_STOCK */
/* */
/* REMOTE TABLES: PART_ORDER, PART_ORDLN, SHIPMENTLN */
/* */
/* COMPILE OPTIONS: */
/* javac SampJava.java */
/* */
/* INVOKED BY: */
/* java SampJava lcldbname rmtdbname */
/*****
import java.sql.*;

public class SampJava {
    private static String JDBCdriver = "com.ibm.db2.jcc.DB2Driver";
    private static String part_table = " "; /* part number in table part_stock */
    private static long line_count = 0; /* total number of order lines */
    private static long eqq_table = 0; /* reorder quantity , tbl part_stock */
    private static long quant_table = 0; /* quantity in stock, tbl part_stock */
    private static long rop_table = 0; /* reorder point , tbl part_stock */
    private static int contl = 0; /* continuation line, tbl order_line */
    private static short next_num = 0; /* next order nbr, table part_order */

    /*****
    /* Method For Reseting Environment */
    /*****
    private static void resetTables(Connection rmtConn) throws SQLException {

        Statement stmt1 = rmtConn.createStatement();

        /* Clean up for rerunability in test environment */
        stmt1.executeUpdate("DELETE FROM DRDA.PART_ORDLN WHERE ORDER_NUM IN " +
            " (SELECT ORDER_NUM FROM DRDA.PART_ORDER " +
            " WHERE ORDER_TYPE = 'R')");
        stmt1.executeUpdate("DELETE FROM DRDA.PART_ORDER WHERE ORDER_TYPE = 'R'");
        stmt1.close();
        rmtConn.commit();

    } /* function delete_for_rerun */

```

```

/*****
/* Method For Calculating Order Quantity */
/*****
private static void calculateOrderQuantity(Connection lclConn, Connection rmtConn, String loc)
throws SQLException {
    PreparedStatement prpStmt1;
    PreparedStatement prpStmt2;
    ResultSet rs1Set1;
    ResultSet rs1Set2;
    short ord_table = 0;          /* order nbr.          , tbl order_line */
    short orl_table = 0;          /* order line        , tbl order_line */

    prpStmt1 = lclConn.prepareStatement("SELECT PART_NUM, PART_QUANT, PART_ROP, PART_EQQ " +
                                        "          FROM DRDA.PART_STOCK WHERE PART_ROP > PART_QUANT AND " +
                                        "          PART_NUM > ? ORDER BY PART_NUM");

    prpStmt1.setString(1,part_table);
    rs1Set1 = prpStmt1.executeQuery();
    if (rs1Set1.next() == false) {
        System.out.println("-----");
        System.out.println("NUMBER OF LINES CREATED = " + line_count);
        System.out.println("-----");
        System.out.println("***** END OF PROGRAM *****");
        rop_table = 0;          /* no (more) orders to process */
    }
    else {
        /* available qty = Stock qty + qty in order - qty received */
        part_table = rs1Set1.getString(1);
        quant_table = rs1Set1.getLong(2);
        rop_table = rs1Set1.getLong(3);
        eqq_table = rs1Set1.getLong(4);
        long qty_rec = 0;

        prpStmt2 = rmtConn.prepareStatement("SELECT A.ORDER_NUM, ORDER_LINE, QUANT_REQ " +
                                            "          FROM DRDA.PART_ORDLN A, DRDA.PART_ORDER B " +
                                            "          WHERE PART_NUM = ? AND LINE_STAT <> 'C' AND " +
                                            "          A.ORDER_NUM = B.ORDER_NUM AND ORDER_TYPE = 'R'");

        prpStmt2.setString(1,part_table);
        rs1Set2 = prpStmt2.executeQuery();
        while (rs1Set2.next()) {
            ord_table = rs1Set2.getShort(1);
            orl_table = rs1Set2.getShort(2);
            long qty_table = rs1Set2.getLong(3);
            qty_rec = qty_rec + qty_table;
        }
        rs1Set2.close();
    }
}

```

```

prpStmt2 = rmtConn.prepareStatement("SELECT SUM(QUANT_RECV) FROM DRDA.SHIPMENTLN " +
    " WHERE ORDER_LOC = ? AND ORDER_NUM = ? AND " +
    " ORDER_LINE = ?");

prpStmt2.setString(1,loc);
prpStmt2.setShort(2,ord_table);
prpStmt2.setShort(3,orl_table);
rsltSet2 = prpStmt2.executeQuery();
rsltSet2.next();
long qty_table = rsltSet2.getLong(1);
qty_rec = qty_rec + qty_table;
rsltSet2.close();
prpStmt2.close();
}
rsltSet1.close();
prpStmt1.close();

} /* end of calculate_order_quantity */

/*****
/* Method For Processing Orders */
/*****
private static void processOrder(Connection rmtConn, String loc) throws SQLException {
    PreparedStatement prpStmt1;
    ResultSet rsltSet1;

    /* insert order and order_line in remote database */
    if (cont1 == 0) {
        prpStmt1 = rmtConn.prepareStatement("SELECT (MAX(ORDER_NUM) + 1) FROM DRDA.PART_ORDER");
        rsltSet1 = prpStmt1.executeQuery();
        rsltSet1.next();
        next_num = rsltSet1.getShort(1);
        rsltSet1.close();
        prpStmt1 = rmtConn.prepareStatement("INSERT INTO DRDA.PART_ORDER (ORDER_NUM, ORIGIN_LOC,
ORDER_TYPE, ORDER_STAT, CREAT_TIME) " +
            " VALUES (?, ?, 'R', 'O', CURRENT_TIMESTAMP)");

        prpStmt1.setShort(1,next_num);
        prpStmt1.setString(2,loc);
        prpStmt1.executeUpdate();
        System.out.println("***** ROP PROCESSING *****");
        System.out.println("ORDER NUMBER = " + next_num);
        System.out.println("-----");
        System.out.println("  LINE    PART    QTY    ");
        System.out.println("  NBR     NBR     REQUESTED");
        System.out.println("-----");
        cont1 = cont1 + 1;
    } /* if cont1 == 0 */
}

```

```

prpStmt1 = rmtConn.prepareStatement("INSERT INTO DRDA.PART_ORDLN (ORDER_NUM, ORDER_LINE,
PART_NUM, QUANT_REQ, LINE_STAT) " +
                                "          VALUES (?, ?, ?, ?, '0')");
prpStmt1.setShort(1,next_num);
prpStmt1.setInt(2,cont1);
prpStmt1.setString(3,part_table);
prpStmt1.setLong(4,eqq_table);
prpStmt1.executeUpdate();
line_count = line_count + 1;
System.out.println("    " + line_count + "          " + part_table + "          " + eqq_table + "");
cont1 = cont1 + 1;
prpStmt1.close();

} /* end of function processOrder                                     */

/*****
/* Method For Displaying Errors                                     */
/*****
private static void errorFunction(SQLException e, Connection lclConn, Connection rmtConn) {

    System.out.println("*****");
    System.out.println("*          SQL ERROR          *");
    System.out.println("*****");
    System.out.println("SQLCODE      = " + e.getErrorCode());
    System.out.println("SQLSTATE    = " + e.getSQLState());
    System.out.println("*****");
    try {
        lclConn.rollback();
        rmtConn.rollback();
    }
    catch (SQLException uowErr) {
    }

} /* end of function errorFunction                                     */

```

```

/*****
/* Mainline */
/*****
public static void main(String[] args) {
    String User = "myuser";
    String Password = "mypwd";
    String lclUrl = null;
    String rmtUrl = null;
    String loc = "SQLA";          /* dealer's database name */
    Connection lclConn = null;
    Connection rmtConn = null;

    try {
        Class.forName(JDBCdriver).newInstance();
    }
    catch (Exception e) {
        System.out.println("Error: Failed to load DB2 driver.");
        System.exit(1);
    }

    try {
        lclUrl = "jdbc:db2:" + args[0];
        lclConn = DriverManager.getConnection(lclUrl, User, Password);
        rmtUrl = "jdbc:db2:" + args[1];
        rmtConn = DriverManager.getConnection(rmtUrl, User, Password);
    }
    catch (Exception e) {
        System.out.println("Error: Failed to get database connections.");
        System.exit(1);
    }

    try {
        /* Initialization */
        resetTables(rmtConn);

        /* Main Work */
        do {
            calculateOrderQuantity(lclConn, rmtConn, loc);
            if (rop_table > quant_table) {
                processOrder(rmtConn, loc);
                quant_table = 0;
            }
        } while (rop_table != 0);

        /* End Work */
        lclConn.commit();
        rmtConn.commit();
    }
    catch (SQLException e) {
        e.printStackTrace();
        errorFunction(e, lclConn, rmtConn);
        System.exit(1);
    }
}
}

```

例: プログラム出力

以下に、部品在庫管理の例のプログラム出力を示します。

注: コードの例を使用することは、443 ページの『コードに関するライセンス情報および特記事項』の条件に同意したものとされます。

```
***** ROP PROCESSING *****
ORDER NUMBER = 6
-----
LINE      PART      QTY
NBR      NBR      REQUESTED
-----
1         14020    100
2         14030    50
3         18020    50
4         21010    50
5         37020    40
-----
NUMBER OF LINES CREATED = 5
-----
***** END OF PROGRAM *****
```

図 38. 例: プログラム出力

DDM の例

このトピックにある例は、ローカル IBM i プラットフォームと、1 つ以上のリモート・システムの両方でデータを処理するのに使用できる代表的なアプリケーション・プログラムに基づいています。

最初の例は、単純な照会アプリケーションであり、2 番目の例は、受注アプリケーションです。3 番目の例は、複数の IBM i プラットフォームにある複数のファイルへのアクセス例です。4 番目の例は、複数の IBM i プラットフォームと 1 つのシステム/36 へのアクセス例です。

ここにある例とタスクの個々のコーディングは、次の 1 つまたは 2 つの部分に分かれます。

- 疑似コード形式で示され、DDM とは関連していないが、プログラミング環境を構築するのに使用されるコーディング。例では、この方法で必要なタスク・ステップを、アプリケーションに使用する言語から独立させて示します。例示したのと同じまたはそれと類似のタスクを実行できるようにコーディングを備えたプログラムを、使用したい言語で作成することができます。
- 多くの場合 CL で書かれ、DDM を使用したネットワーク内の他のシステムとの通信のためのコーディング。

DDM の通信のセットアップの例とタスク

このトピックでは、以下のタスク例で DDM が使用されているネットワークについて説明します。

このネットワークでは、セントラル・システム (IBM i 製品) がフィラデルフィアにあり、リモート IBM i プラットフォームがトロントとニューヨーク市に 1 台ずつ、1 つのシステム/38 がシカゴに、そして 1 つのシステム/36 がダラスにあります。これらのシステム用の拡張プログラム間通信機能 (APPC) ネットワークは、次の図に示した値を使用して構成されています。

この一連のタスク例では、システム/36 にはリリース 5 の DDM および、互換性のある PTF を備えた DDM が導入されています。システム/38 にはリリース 8 の CPF が、DDM ライセンス・プログラムおよびこのシステムに適用される互換プログラム一時修正 (PTF) 変更と一緒に導入されています。

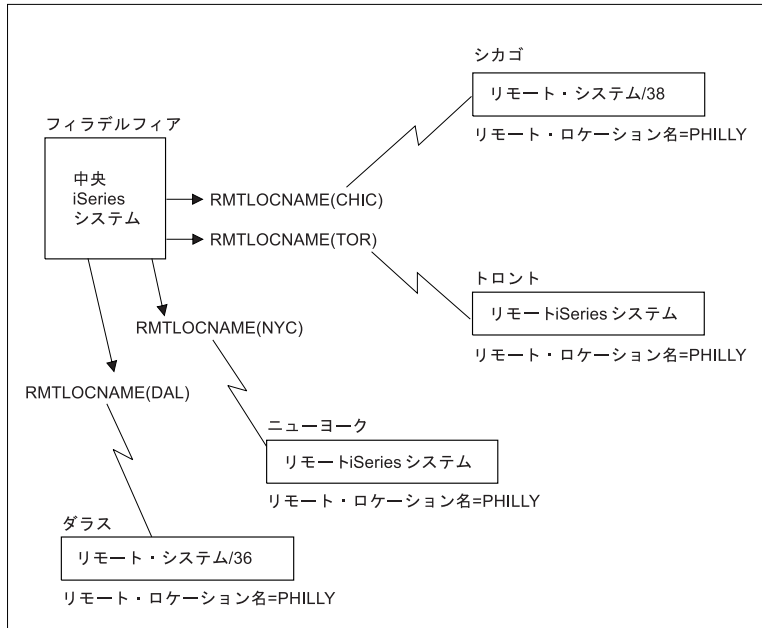


図 39. ORDERENT アプリケーション・タスクで使用される DDM ネットワーク

DDM の例 1: 簡単な照会アプリケーション

最初の例は、顧客の複数の営業拠点で、それぞれの 1 次ファイルを使って、それらのシステムにおいてどのように同一の照会アプリケーションを処理できるかを示します。

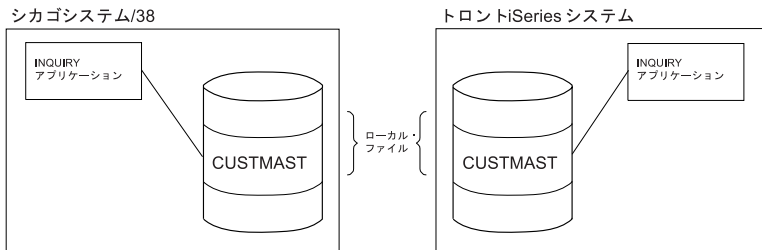


図 40. ローカル照会を行う 2 つの非 DDM サーバー

DDM を使わないと、ここに示した 2 つのロケーション (シカゴとトロント) はそれぞれ、相違のあるレベルと重複したレベルの情報の入った専用の 1 次ファイル (CUSTMAST) を持つことになります。

次のプログラム (疑似コード形式) は、CUSTMAST という名前の各システム独自の 1 次ファイルにアクセスするため、各ロケーションで実行されるプログラムです。

```

Open CUSTMAST
LOOP: Prompt for CUSTNO
      If function 1, go to END
      Get customer record
      Display
      Go to LOOP
END:  Close CUSTMAST
      RETURN

```

DDM を使用すると、CUSTMAST ファイルは中央ロケーション (この例ではフィラデルフィア) で 1 つのファイルが統合されるので、シカゴとトロントのローカル・ファイルを削除することができます。各リモート・ロケーションと中央ロケーションで、そのファイルにアクセスするために使用される照会プログラムは、前に使用したプログラムと同じものです。

プログラムに変更を加えずにリモート照会を実行しようとする場合、各リモート・ロケーションでは DDM ファイルを作成し、次のような一時変更コマンドを使用するだけで十分です。

```
CRTDDMF FILE(INQ) RMTFILE(CUSTMAST) RMTLOCNAME(PHILLY)
```

```
OVRDBF FILE(CUSTMAST) TOFILE(INQ)
```

DDM ファイルは、サーバー・システムとしてフィラデルフィアのシステムを指し示し、リモート・ファイルとして CUSTMAST ファイルを指し示します。PHILLY という名前のリモート・ロケーションであれば、このコマンドの値をそのままそれぞれのリモート・ロケーションで使用することができます。

CUSTMAST はプログラム内で使用するファイル名であるため、存在しない CUSTMAST ファイルを DDM ファイル INQ に一時変更するには、データベース・ファイル一時変更 (OVRDBF) コマンドを使用する必要があります。(CUSTMAST ファイルがローカル・システムにまだ存在するときには、セントラル・システムの 1 次ファイルにアクセスするには一時変更が必要です。一時変更を行わないと、ローカル・ファイルにアクセスしてしまいます。)

次の図は、INQ という名前のそれぞれの DDM ファイルを使用して、中央システムの CUSTMAST ファイルにアクセスする同じ 2 つのシステムを示しています。

別の方法としては、CUSTMAST ファイルをシカゴとトロントのシステムに残しておいて、名前や住所などの特に重要ではない照会の際にはそのファイルを使用し、変更を行う場合にフィラデルフィアの中央 CUSTMAST ファイルを使用する方法があります。シカゴとトロントのシステムにある CUSTMAST ファイルは定期的に変更すれば、フィラデルフィアのシステムの 1 次ファイルの現行レベルに合わせるすることができます。

ORDERENT アプリケーション実例では、この別の方法が使用されています。

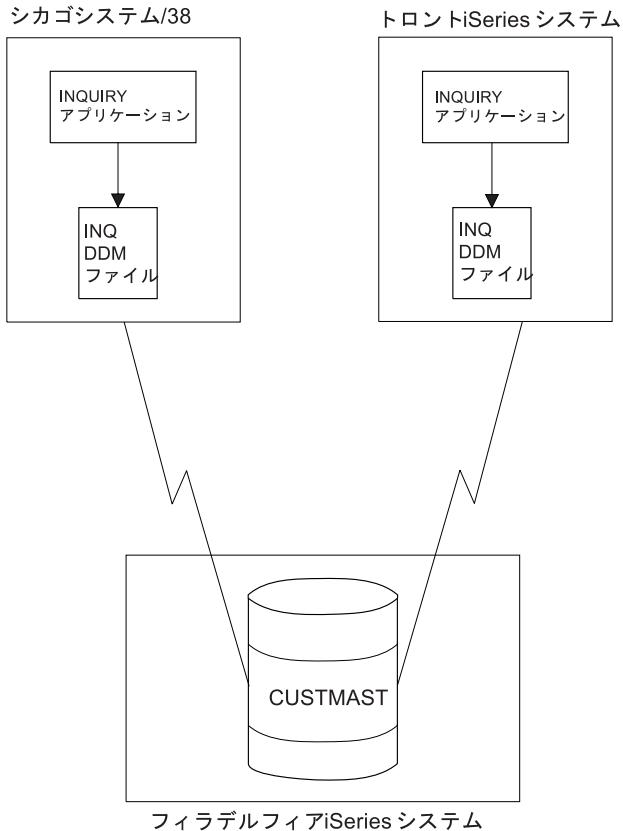


図 41. リモート照会を行う 2 つの DDM サーバー

DDM の例 2: ORDERENT アプリケーション

2 番目の例は、顧客の複数の営業拠点で、どのように DDM を使って 1 つの受注アプリケーションを処理できるかを示します。

この例の最初のタスクは、どのように DDM を使用して、同じアプリケーション・プログラムのコピーを各リモート・システムに配置し、1 つの 1 次ファイルを中央ロケーションに置くかを示します。この例の 2 番目のタスクは、どのように DDM を使用して、ファイルをリモート・システムへコピーするかを示しています。

DDM の例 2: セントラル・システムの ORDERENT ファイル:

フィラデルフィアの中央サイトでは、図に示した 4 つのファイルを、ORDERENT アプリケーション・プログラムが使用します。

セントラル・システムでは、CUSTMAST ファイルは、すべてのロケーション用の顧客データの 1 次ファイルである物理ファイルです。CUSTMST2 ファイルは、CUSTMAST 物理ファイルに基づく論理ファイルです。セントラル・システムで論理ファイルを使用すれば、少なくとも次の 2 つの利点があります。

- 同じ ORDERENT プログラムを変更しないまま、セントラル・システムと各リモート・システムで使用することができます。
- データには、別々のファイルを介してアクセスすることができ、受注処理の期間、顧客の 1 次レコードをロックしたままにしておくことはできません。

中央サイトの 4 つのファイルは、次のように使用します。

- CUSTMAST ファイルには、その顧客すべてに関する全データを入力します。顧客受注処理が完了すれば、CUSTMAST ファイルは、その受注で得たすべての新情報で変更されます。
- CUSTMST2 ファイルはセントラル・システムの論理ファイルですが、このファイルは、顧客受注処理の開始時点で使用します。オペレーターが顧客番号を入力すると、プログラムは CUSTMST2 論理ファイルにある顧客データを読み込みますが、実際にはそのデータは 1 次ファイルの CUSTMAST にあるものです。
- INVEN ファイルには、顧客へは販売できるすべての品目の現在庫量を入力します。オペレーターが受注品目番号と数量を入力すると、INVEN ファイル内の対応する 1 次品目が変更されます。
- DETAIL ファイルは、個々の受注品目すべてのリストです。このファイルには、顧客から受注された各品目と数量のレコードが入ります。

中央Series システムORDERENT アプリケーション

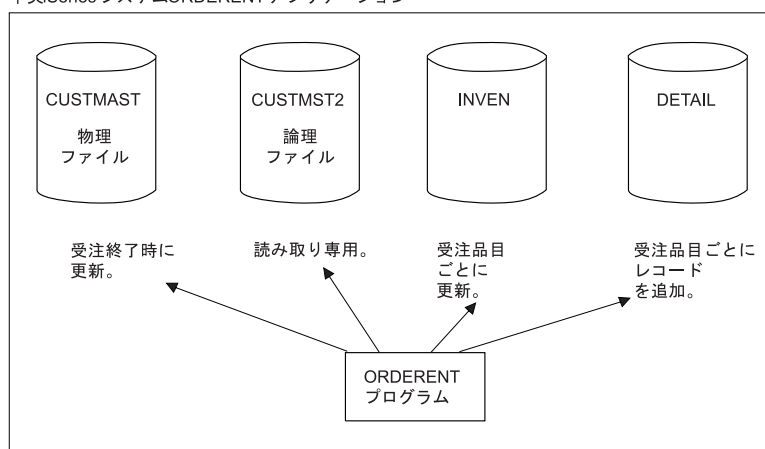


図 42. セントラル・システムの ORDERENT プログラムが使用するファイル

DDM の例 2: ORDERENT プログラムの説明:

当初 ORDERENT プログラムは、セントラル・システム (フィラデルフィア) にあるライブラリー PGMLIB 内にしかありません。

このプログラムは、以下のことを行います。

- 受注担当オペレーターが顧客番号を入力すると、ORDERENT はその顧客番号を読み込んでから、PGMLIB ライブラリー内のファイル CUSTMST2 の最初のメンバーを読み取って、顧客名、住所、およびその他の情報を探し出します。このようにして取り出された情報は、オペレーターに表示され、プログラムから希望する品目番号と数量を入力するよう求められます。
- オペレーターが希望する品目番号と数量を入力して実行キーを押すと、プログラムは、INVEN ファイルの最初のメンバー内の対応する 1 次項目を変更し、入力された各品目と数量ごとに DETAIL ファイルへレコードを 1 つずつ追加します。オペレーターがプログラムを終了するまでそのプログラムは、次の品目と数量を入力するよう連続して求めます。
- オペレーターがプログラムを終了すると、ファイル CUSTMAST は、受注全体に関する情報に基づいて変更されます。(詳細については、ORDERENT の疑似コードを参照してください。)

次の例の場合、フィラデルフィアの CUSTMAST へアクセスする必要のあるリモート・システム・ユーザーは全員、アクセス権をすでに持っている、アクセスの必要のないユーザーはアクセス権を持っていないことを前提としています。これらの例では、シカゴの IBM i プラットフォームにはコンパイラーはありません。

このプログラムを、物理的在庫も持っているすべてのリモート・ロケーションで使用できるようにしたい場合には、これらの各リモート・システムへこのプログラムを送らなければなりません。各リモート・システムには、それぞれ専用の在庫ファイルと 1 次ファイルの INVEN、DETAIL、および CUSTMST2 (これは、CUSTMAST のコピー) があるものと仮定します。どのようにプログラムをリモート・システムへ送れるかについては、381 ページの『DDM の例 2: サーバー・システムへのプログラム転送』に説明があります。

注: コードの例を使用することは、443 ページの『コードに関するライセンス情報および特記事項』の条件に同意したものとされます。

ORDERENT プログラムの疑似コード

```
•
•
•
DECLARE CUSTMAST CHANGE
    * ファイル CUSTMAST を宣言して、変更できるようにする。
DECLARE CUSTMST2 READ
    * ファイル CUSTMST2 を読み取り専用と宣言する。
DECLARE INVEN CHANGE
    * 在庫ファイル INVEN を宣言し、変更できるようにする。
DECLARE DETAIL OUTPUT
    * ファイル DETAIL を出力専用と宣言する。
•
•
•
CUSTMAST、CUSTMST2、INVEN、および DETAIL ファイルをオープンする。
    * プログラムを開始する。
    CUSTNO を尋ねる受注入力画面を表示する。
        * 受注担当オペレーターが CUSTNO を入力する。
    機能キーの場合、End に行く。
    画面から CUSTNO を読み取る。
        CUSTNO の場合、NAME、ADDR、および
        他の情報を CUSTMST2 ファイルから戻す。
    NAME、ADDR、および他の情報を画面に表示する。
    LOOP: '品目番号 _____ 要求数量 _____' を表示する。
        * 受注担当オペレーターが、品目番号と数量を入力する。
    画面から ITEMNO および要求数量を読み取る。
    ITEMNO = 0 の場合、LOOPEND に行く。
        INVEN の ITEMNO および要求数量を変更する。
        項目レコードを DETAIL ファイルに書き込む。
    LOOP に行く。
    LOOPEND: CUSTNO の場合、ファイル INVEN を使用して
        CUSTMAST を変更する。

End
    * プログラムを終了する。
CUSTMAST、CUSTMST2、INVEN、および DETAIL ファイルをクローズする。
RETURN
```

DDM の例 2: リモート・システムの ORDERENT ファイル:

ORDERENT プログラムは、すべてのロケーションで変更はありませんが、CUSTMST2 ファイルはこれで、セントラル・システムの顧客 1 次ファイル CUSTMAST のコピー になりました。

何回も変更しないデータの場合、できる限り CUSTMST2 を使えば、各受注入力要求を処理するのに要する通信時間を最小限にとどめることができます。リモート ORDERENT プログラムは、各受注処理の開始

時点でローカル CUSTMST2 ファイルを読み込んでから、1 つの受注処理が完了したときに初めて、DDM を使用してセントラル・システムにある CUSTMAST ファイルを更新します。

他の 2 つのファイル INVEN と DETAIL には、セントラル・システムと同じ機能が各リモート・システムにもあります。

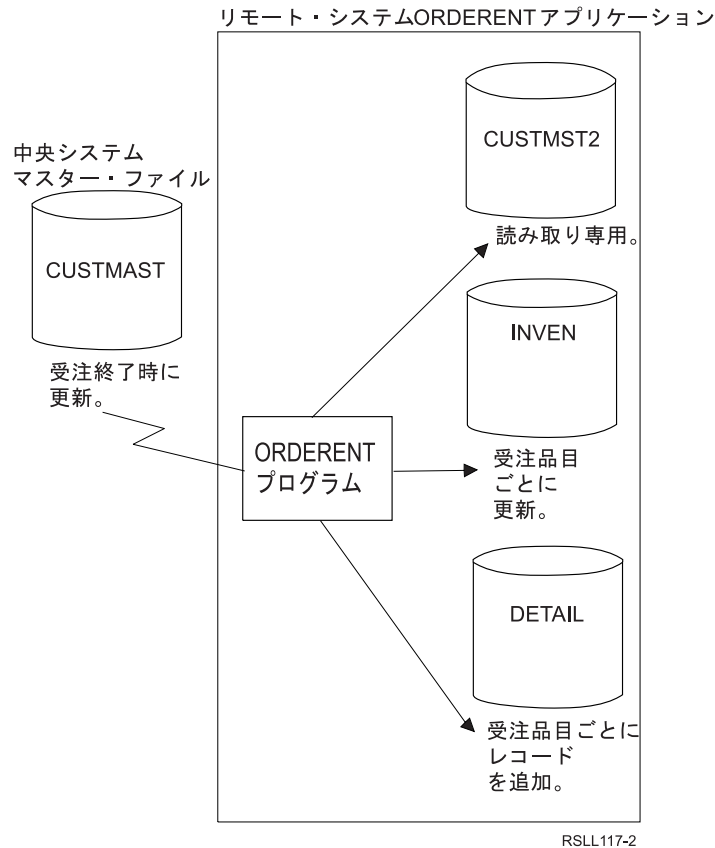


図 43. リモート ORDERENT プログラムが使用するファイル

CUSTMAST ファイルはすべてのロケーションによって変更されるファイルであり、このファイルには、各顧客別の最新情報（顧客の勘定残高などの随時変更されるデータに関して）が入ります。CUSTMST2 ファイルは、必要なときにしか変更されないデータ（名前や住所など）を読み取るのに使用されるファイルであり、このファイルには、CUSTMAST ファイルを再コピーして、定期的に変更（例：毎週 1 回）する必要があります。この例のタスク 2 は、これを行うための方法を説明します。

DDM の例 2: サーバー・システムへのプログラム転送:

このタスクでは、フィラデルフィアにある DDM ネットワーク内のセントラル・システムが、ORDERENT という名前のプログラムをシカゴにあるリモート・システム/38 に送ります。

プログラム ORDERENT は、フィラデルフィアのシステムから、ANDERSON CHICAGO というユーザー ID を持ったシカゴのユーザーに送られてから、シカゴの ORDERENT で、フィラデルフィアのセントラル・システムにあるライブラリー PGMLIB 内の CUSTMAST ファイルを変更できるようにセットアップされます。読み取り専用機能をローカル・ファイル（シカゴにあるファイル）に対して実行すると、リモート・ファイル（フィラデルフィアにあるファイル）内で変更が行われます。

このタスクでは、フィラデルフィアの ORDERENT プログラムをシカゴのリモート・システムへ転送するのに使用できる 2 とおりの方法を示しています。基本的に、いずれの方式でも同じ一連のコマンドを使用していますが、パススルー方式で使用している 2 番目のコマンド・グループは、SBMRMTCMD コマンド方式では、使用するリモート・コマンド投入 (SBMRMTCMD) コマンド内に組み込まれているという相違点があります。

- 最初の方式は、パススルーとオブジェクト配布を使用するため、クライアント・システムのオペレーターは、サーバー・システムのオペレーターが関与しなくても、また、SBMRMTCMD コマンドを使用しなくても、両方のシステムをセットアップすることができます。この方式は、IBM i または システム/38 のみ使用できます。
- このタスクでのサーバー・システムがシステム/38 であるため、2 番目の方式では SBMRMTCMD コマンドを使用します。(SBMRMTCMD コマンドは、サーバー・システムが IBM i または システム/38 のときに使用できます。)

DDM の例 2: パススルー方式:

一連のコマンドをクライアント・システムで入力すると、サーバー・システムでパススルー・セッションが始動し、次に 2 番目の一連のコマンドをクライアント・システムで入力すると、サーバー・システムでそれらのコマンドが実行されます。

次のコマンドは、フィラデルフィアのクライアント・システムから出されるものです。

```
CRSAVF FILE(TRANSFER)
SAVOBJ OBJ(ORDERENT) LIB(PGMLIB) SAVF(TRANSFER)
        UPDHIST(*NO) DTACPR(*YES)
SNDNETF FILE(TRANSFER) TOUSRID(ANDERSON CHICAGO)
```

次に、パススルー開始 (BGNPASTHR) コマンドを使って、フィラデルフィアとシカゴのシステムとの間でパススルー・セッションを始動します。このセッションを使用して、クライアント・システムで次のコマンドを入力すると、それらのコマンドは、サーバー・システムで実行されます。

```
CRSAVF FILE(RECEIVE)
RCVNETF FROMFILE(TRANSFER) TOFILE(RECEIVE)
CRTLIB LIB(PGMLIB)
RSTOBJ OBJ(ORDERENT) SAVLIB(PGMLIB) SAVF(RECEIVE)
CRTDDMF FILE(CUSTMAST.PGMLIB) RMTFILE(*NONSTD 'PGMLIB/CUSTMAST')
        DEVD(PHILLY)
```

これらのコマンドは、RECEIVE という名前の保管ファイルを作成し、その中へフィラデルフィアのクライアント・システムからネットワーク・ファイルとして受け取った TRANSFER ファイルがコピーされます。シカゴのシステムではライブラリーが作成されます。PGMLIB という名前のこの新たに作成されたライブラリー内に、RECEIVE ファイルが ORDERENT プログラムとして復元されます。最後に、フィラデルフィアのシステム (PHILLY という名前のリモート・ロケーション) にある CUSTMAST ファイルにシカゴのシステムからアクセスできるように、シカゴのシステムで DDM ファイルが作成されます。

DDM の例 2: SBMRMTCMD コマンド方式:

タスクを実行するのに必要なコマンドは、クライアント・システムで入力します。クライアント・システムは、ターゲット・システムに対してリモート・コマンド投入 (SBMRMTCMD) コマンドを使用して、ターゲット IBM i オペレーティング・システムに必要なコマンドを送ります。

次のコマンドは、シカゴのサーバー・システムへ ORDERENT プログラムを送るのに、フィラデルフィアのクライアント・システムから出されるコマンドです。


```

CRTSAVF FILE(TRANSFER)
SAVOBJ OBJ(ORDERENT) LIB(PGMLIB) SAVF(TRANSFER)
  UPDHIST(*NO)
SNDNETF FILE(TRANSFER) TOUSRID(ANDERSON CHICAGO)
CRTDDMF FILE(CHICAGO) RMTFILE(XXXXX) RMTLOCNAME(CHIC)
SBMRMTCMD CMD('CRTSAVF FILE(RECEIVE)') DDMFILE(CHICAGO)
SBMRMTCMD CMD('RCVNETF FROMFILE(TRANSFER)
  TOFILE(RECEIVE)') DDMFILE(CHICAGO)
SBMRMTCMD CMD('CRTLIB LIB(PGMLIB)') DDMFILE(CHICAGO)
SBMRMTCMD CMD('RSTOBJ OBJ(ORDERENT) SAVLIB(PGMLIB)
  SAVF(RECEIVE)') DDMFILE(CHICAGO)
SBMRMTCMD CMD('CRTDDMF FILE(CUSTMAST.PGMLIB)
  RMTFILE(*NONSTD "PGMLIB/CUSTMAST") DEVD(PHILLY)')
  DDMFILE(CHICAGO)

```

これらのコマンドは、TRANSFER という名前の保管ファイルを作成し、このファイルに ORDERENT プログラムを保管してから、シカゴのサーバー・システムへネットワーク・ファイルとしてこのプログラムを送ります。シカゴでは、SBMRMTCMD コマンドに組み込まれていたコマンドが使用され、サーバー・システム上で保管ファイル（名前は RECEIVE）を作成し、TRANSFER ファイルを受け取り、そして、新たに作成された PGMLIB ライブラリー内でそのファイルを ORDERENT に復元します。最後に、フィラデルフィアのシステムにある CUSTMAST ファイルにシカゴのシステムからアクセスできるよう、シカゴのシステムで DDM ファイルが作成されます。DDM ファイル作成 (CRTDDMF) コマンドは、システム/38 構文になっています。

このいずれかの方式を使ってシカゴのシステムへ ORDERENT プログラムを送り、そこで DDM ファイルを作成できたら、シカゴのシステムにある ORDERENT プログラムを使用して、フィラデルフィアのシステムにある CUSTMAST ファイルへアクセスすることができます。

DDM の例 2: ファイルのコピー:

例 2 (サーバー・システムへのプログラム転送) の最初のタスクを実行し終われば、現行レベルの CUSTMAST ファイル (フィラデルフィアの) をシカゴのシステムへコピーして、CUSTMST2 ファイルを最新の内容にすることができます。

この例では、シカゴには CUSTMST2 ファイルがすでに存在することを前提としています。

次のコマンドを使用すれば、フィラデルフィアのシステムにある CUSTMAST ファイルを、シカゴのシステムにある CUSTMST2 ファイルへコピーすることができます。(これらのコマンドは、フィラデルフィアのシステムから実行します。)

```

CRTDDMF FILE(PHILLY/COPYMAST) RMTFILE(*NONSTD 'CUSTMST2.CHICAGO')
  RMTLOCNAME(CHIC)
CPYF FROMFILE(PGMLIB/CUSTMAST) TOFILE(PHILLY/COPYMAST)
  MBROPT(*REPLACE)

```

注: 別の方式として、クライアント・システムで DDM ファイルを作成し、SBMRMTCMD コマンドを使ってサーバー・システムへ分散データ管理 (DDM) ファイル作成 (CRTDDMF) コマンドを投入してから、別の SBMRMTCMD コマンドを使用して、新たに作成されたこの DDM ファイルを使用しさえすれば、元のシステムへ戻ってコピー機能を実行できると想定されるかもしれませんが。しかしこの方式は成功しません。その理由は、IBM i オペレーティング・システムは、同一ジョブ内で同時にクライアント・システムとサーバー・システムにはなりえないからです。

DDM の例 3: 複数の IBM i ファイルへのアクセス

例 1 および 2 と同じ通信環境を使って、2 つのリモート IBM i プラットフォームとリモートのシステム/38 にある同じ名前のファイルについて、在庫に関する質問を出すことができます。

このためには、シカゴ、トロント、およびニューヨークのシステムにある LIB/MASTER という名前のファイルへアクセスできるプログラム (ここでは疑似コードで示してあります) をセントラル・システムで作成する必要があります。(この例では、MASTER ファイルはキー付きファイルであり、この各ファイルの最初のメンバーを使用します。また、MASTER ファイル用のデータ記述仕様 (DDS) は、フィラデルフィアのセントラル・システムにあります。)

プログラムは、ローカル受注担当オペレーターに品目番号 (ITEMNO) を尋ねて、シカゴ、トロント、およびニューヨークのファイルにある在庫数量 (QOH) を戻します。

次のコマンドは、フィラデルフィアのシステムから出されるものです。

```
CRTDDMF  PGMLIB/CHIFILE  RMTFILE(*NONSTD 'MASTER.LIB')
          RMTLOCNAME(CHIC)
CRTDDMF  PGMLIB/TORFILE  RMTFILE(LIB/MASTER)    RMTLOCNAME(TOR)
CRTDDMF  PGMLIB/NYCFILE  RMTFILE(LIB/MASTER)    RMTLOCNAME(NYC)
```

以下に、このタスクを実行するための疑似コードの例を示します。

```
DECLARE CHIFILE, TORFILE, NYCFILE INPUT
CHIFILE, TORFILE, NYCFILE をオープンする
LOOP:  ITEMNO を尋ねる画面を表示する
      画面から ITEMNO を読み取る
      キー ITEMNO を使用して CHIFILE からレコードを読み取る
      キー ITEMNO を使用して TORFILE からレコードを読み取る
      キー ITEMNO を使用して NYCFILE からレコードを読み取る
      すべての QOH 値を画面に書き込む
      機能キーでない場合、LOOP に行く
CHIFILE, TORFILE, NYCFILE をクローズする
END
```

プログラムをコンパイルするのに、データベース・ファイル一時変更 (OVRDBF) コマンドを使用して、プログラム内で使用する 3 つのファイルをローカル・ファイルに一時変更することができます。なお、このローカル・ファイルには外部記述形式が入っていて、アクセスしようとするリモート・ファイルと同じものです。このようにすれば、リモート・システムにあるファイルへはその時点でアクセスしなくてもよいため、コンパイルに要する時間が大幅に短縮されます。

プログラムを正しくコンパイルし終わったら、プログラムがリモート・ファイルへアクセスできるよう、一時変更を除去します。

一時変更の使用に代わる別の方法に、ファイル定義を別のライブラリーに保管しておく方法があります。プログラムを、そのライブラリー内のファイル定義を使ってコンパイルしてから、実ライブラリーを使用して実行することができます。

DDM の例 4: システム/36 上のファイルへのアクセス

このトピックでは、例 3 の疑似コード・プログラムをどのように変更すれば、例 3 にある IBM i オペレーティング・システムとシステム/38 の MASTER ファイルへの場合と同じ方法で、ダラスのシステム/36 にある MASTER ファイルにアクセスできるかを示します。

システム/36 に対するパススルーが備わっている、または、必要に応じて、システム/36 上でシステム/36 オペレーターが代わりに変更できるものと仮定します。

次のコマンドは、フィラデルフィアのサーバーから出されるコマンドです。

```
CRTDDMF  FILE(PGMLIB/DALFILE)  RMTFILE(MASTER)
          RMTLOCNAME(DAL)  ACCMTH(*KEYED)
```

DALFILE という名前の DDM ファイルが参照するリモート・ファイルは、システム/36 にあるため、次のうちのいずれかを行う必要があります。

- リモート・ファイルのレコード様式を、プログラム内で記述する必要があります。つまりこのファイルは、プログラム記述ファイルでなければなりません。
- プログラムをシステム/36 ファイルではなく、ローカル IBM i ファイルが参照するプログラムでコンパイルする必要があります。このローカル・ファイルには、DDM ファイル名と同じレコード様式名が付いていなければなりません。ローカル・ファイルには、データ・レコードが入っている必要はありません。

以下に、このタスクを実行するための疑似コードの例を示します。

```
DECLARE CHIFILE, TORFILE, NYCFILE, DALFILE INPUT
CHIFILE, TORFILE, NYCFILE, および DALFILE をオープンする
LOOP: ITEMNO を尋ねる画面を表示する
画面から ITEMNO を読み取る
  キー ITEMNO を使用して CHIFILE からレコードを読み取る
  キー ITEMNO を使用して TORFILE からレコードを読み取る
  キー ITEMNO を使用して NYCFILE からレコードを読み取る
  キー ITEMNO を使用して DALFILE からレコードを読み取る
  すべての QOH 値を画面に書き込む
  機能キーでない場合、LOOP に行く
CHIFILE, TORFILE, NYCFILE, DALFILE をクローズする
END
```

ユーザー FAQ

他の特定の IBM 製品を処理するときには、以下の状況を考慮する必要があります。

このトピックでは、IBM i 製品のネットワーク (同種 環境) における分散リレーショナル・データベースのための IBM i サポートについて集中的に説明しています。しかし、DRDA をサポートするさまざまなプラットフォームから成るネットワークの中には、分散リレーショナル・データベースを実装したものが数多く存在しています。

このトピックでは、IBM i 製品を異種 DRDA 環境で使用する際に必要とされるヒントおよび技法のリストを示します。ただし、この付録の目的は包括的なリストを提示することではありません。この付録で説明されているような問題または条件の多くは、アプリケーションに左右されることが少なくありません。各 IBM プラットフォームの相違の詳細については、『*IBM SQL Reference Volume 2*』または『*DRDA Application Programming Guide*』(SC26-4773) を参照してください。

関連資料:

138 ページの『IBM i サポート』

ジョブ変更 (CHGJOB) コマンドを使用することによって、IBM i ジョブの CCSID を変更することができます。

分散リレーショナル・データベースへの接続

IBM i 以外のアプリケーション・リクエスト (AR) から DB2 for IBM i アプリケーション・サーバー (AS) に接続する場合、CCSID 65535 でタグ付けされている列は変換できません。これらの列が入っているファイルに明示的に CCSID を指示した列が含まれていない場合は、すべての文字列の CCSID を別の CCSID の値に変更することができます。CCSID を変更するには、物理ファイルの変更 (CHGPF) コマンドを使用します。物理ファイルを基礎にする論理ファイルが作成されている場合は、受け取ったエラー・メッセージ (CPD322D) の回復トピックに示されている指示に従ってください。

IBM i システム値 QCCSID

QCCSID (コード化文字セット ID) のデフォルトのシステム値は 65535 です。

この CCSID でタグ付けされたデータは、受信側システムで変換することができません。IBM i アプリケーション・リクエスター (AR) がこの CCSID を使用していると、異種システムに接続できない場合があります。また、この CCSID でタグ付けされているソース・ファイルは、異種システム上でアプリケーションを作成するために使用できない場合もあります。

接続時点で使用される CCSID は、ジョブの CCSID によって決まります。ジョブが開始されると、ジョブの CCSID は、そのジョブが実行されるユーザー・プロファイルによって決まります。ユーザー・プロファイルは、デフォルトと同様に、システム値 QCCSID を使用することができます。

システムのデフォルト CCSID をサポートしないシステムに接続する場合は、ジョブ CCSID を変更する必要があります。ジョブ変更 (CHGJOB) コマンドを使用することによって、ジョブの CCSID を変更することができます。ただし、この解決法が有効なのは、現在処理中のジョブの場合だけです。次にはまたジョブの CCSID を変更しなければなりません。

分散リレーショナル・データベースの中で使用されるユーザー・プロファイルで指定されている CCSID を変更すれば、上記の方法に比べてより永続的な解決策になります。ユーザー・プロファイルを変更すると、データを変換する必要のあるユーザーだけに影響が生じます。DB2 for i アプリケーション・サーバー (AS) を使用する場合は、AS で使用するユーザー・プロファイルを変更する必要があります。

異なるアプリケーション・リクエスターが、ジョブ CCSID 65535 を使用して DB2 for i アプリケーション・サーバーに接続する場合、ジョブは、ジョブ・デフォルト CCSID を使用するように切り替えられます。ジョブ・デフォルト CCSID は、ジョブの言語 ID (LANGID) によって決定されます。この場合は、ジョブの CCSID を 65535 以外の値に切り替えたほうが高いパフォーマンスが得られます。たとえば、サーバー・ジョブが実行されるユーザー・プロファイルの値に CCSID 値を変更できます。

ユーザー・プロファイルの中のデフォルトの CCSID は *SYSVAL です。これは QCCSID システム値を参照します。すべてのユーザー・プロファイルが システム値変更 (CHGSYSVAL) コマンドで使用する QCCSID システム値は変更できます。このようにする場合、システム上のほとんどの (すべてではないが) ユーザーが使用できる CCSID を選択するとよいでしょう。

システムまたはジョブによって使用される CCSID をサポートしていないシステムで作業を行っていると思われる場合には、ジョブ・ログ、SQLCA、または SQL 診断域の中に次の標識があるかどうか探索してください。

メッセージ

SQ30073

SQLCODE または DB2_RETURNED_SQLCODE

-30073

SQLSTATE

58017

Text 分散データ管理 (DDM) パラメーター値 X'0035' はサポートされていない。

メッセージ

SQL0332

SQLCODE または DB2_RETURNED_SQLCODE

-332

SQLSTATE

57017

Text CCSID &1 と CCSID &2 の間の全変換は正しくない。

関連概念:

IBM i グローバリゼーション

関連資料:

ジョブ変更 (CHGJOB) コマンド

システム値変更 (CHGSYSVAL) コマンド

DB2 for z/OS および DB2 Server for VM データベース・マネージャーの CCSID 変換の考慮事項

DB2 for i と他の DB2 データベースの間の違いの 1 つに、IBM i オペレーティング・システムの方がより大きな CCSID のセットをサポートすることが挙げられます。そのため、他のデータベース・マネージャーがデータに対して文字変換を行おうとすると、エラーとなる可能性があります (SQLCODE -332 および SQLSTATE 57017)。

DB2 SQL カタログ・テーブルの中の特定のフィールドには、DBCS 混用データ・タイプを持つよう定義することができます。これは、2 バイト文字セット (DBCS) 文字と 1 バイト文字セット (SBCS) 文字を両方とも使用できるデータ・タイプです。これらのフィールド・タイプの CCSID は、デフォルトの CCSID に基づいています。

これらのフィールドが DB2 for z/OS または DB2 Server for VM アプリケーション・リクエスター (AR) から選択されると、SELECT ステートメントが失敗する場合があります。これは、DB2 for z/OS および DB2 Server for VM データベースがこの CCSID への変換をサポートしない場合があることによります。

このエラーを避けるためには、DB2 for z/OS データベースまたは DB2 Server for VM AR を変更して、以下のいずれかで実行されるようにしなければなりません。

- IBM i SQL カタログ・テーブル内の DBCS-OPEN フィールドと同じ混合バイト CCSID。
- IBM i SQL カタログ・テーブル内の DBCS-OPEN フィールドの混合バイト CCSID からのデータであるときに、サーバーでデータの変換ができる CCSID。この CCSID は、IBM i SQL カタログ・テーブルの DBCS-OPEN フィールド内のデータがすべて 1 バイト・データである場合には、1 バイト CCSID になります。

システムに正しい変更を加えられるようにするためには、DB2 for z/OS または DB2 Server for VM 上でサポートされている CCSID 変換を分析する必要があります。このエラーの処理方法に関する固有の情報については、*DB2 UDB for z/OS Administration Guide* を参照してください。

DB2 for Linux、UNIX、および Windows から接続しようとするとき SQL5048N メッセージが表示されるのはなぜですか？

メッセージ SQL5048N の定義は、データベース・サーバーのリリース・レベルがデータベース・クライアントのリリース・レベルをサポートしていないことを示すものです。しかし、このメッセージは、時おり誤って戻されることがあります。

この問題は、一般に次のようなことが原因で発生します。

1. Client Application Enabler だけがインストールされている状態のとき、このエラー・メッセージが表示されます。この場合は、ゲートウェイ・サーバーを通してクライアント・システムを IBM i プラットフォームに接続させる必要があります。直接接続はサポートされていません。
2. 接続の手動構成中にどこかでエラーが発生していた場合にも、このエラーが戻される可能性があります。

クライアント構成アシスタント (CCA) を使用することによって、SQL5048N が戻されないようにすることができます。

この他にこの問題の原因として考えられるのは、コレクション NULLID に関する問題です。DB2 for Linux、UNIX、および Windows、IBM DB2 Universal Driver for SQLJ および JDBC、および他のアプリケーション・リクエスターは、それぞれに必要な SQL パッケージを作成するためにコレクション NULLID を使用します。コレクションとパッケージは、最初の接続で作成されます。ユーザー・プロファイルにコレクションを作成する権限がない場合は、最初に接続するプロファイルを、それよりも高い権限を持つ別のプロファイルにして、これらのオブジェクトが作成できるようにしてください。

このエラーに別の原因があると思われる場合は、Authorized Problem Analysis Report  Web サイトを参照してください。「Search」フィールドに「APAR II12722」と入力してください。

IBM i ファイルはジャーナルする必要があるか？

クライアント・アプリケーションが非コミット (NC) または非コミット読み取り (UR) の分離レベルを使用していて、しかも DB2 for i SQL 機能が、照会データはブロック化できると判別した場合は、ジャーナル処理は必要ありません。その場合、コミットメント制御は使用可能ではなく、それによってジャーナル処理は不要になります。

この質問への回答は、『パフォーマンスを向上させるために、照会データはどのような場合にブロック化されるのか?』の質問と密接に関係しています。

分離レベルを変更する方法の例は、以下のとおりです。

- DB2 for Linux、UNIX、および Windows のプリコンパイラーが、ISOLATION UR パラメーターを使用して、非コミット読み取りを指定します。
- DB2 for Linux、UNIX、および Windows のコマンド行プロセッサ (CLP) が、コマンド DBM CHANGE SQLISL TO UR を使用して、非コミット読み取りを指定します。
- DB2 for Linux、UNIX、および Windows のコマンド行プロセッサ (CLP) が、コマンド DBM CHANGE SQLISL TO NC を使用して、非コミットを指定します。
- JDBC クライアントが、接続のプロパティ分離レベルを TRANSACTION_READ_UNCOMMITTED に設定して、非コミット読み取りを指定します。

パフォーマンスを向上させるために、照会データはどのような場合にブロック化されるのか？

照会データは、以下の条件のいずれも該当しない場合にブロック化されます。

- カーソルが更新可能である (注 1 参照)。
- カーソルが潜在的に更新可能である (注 2 参照)。
- SQLPREP または SQLBIND で BLOCKING NO プリコンパイラーまたはバインド・オプションが使用されている。

BLOCKING NO のプリコンパイル/バインド・オプションを付けた単一行プロトコルを強制しないかぎり、以下の条件のどちらの場合も、ブロック化が行われます。

- カーソルが読み取り専用である (注 3 参照)。
- 以下のすべてが該当する。
 - SELECT の中に FOR UPDATE OF 文節がなく、しかも、
 - プログラム中にカーソルに対する UPDATE または DELETE WHERE CURRENT OF ステートメントがなく、しかも、
 - プログラムに動的 SQL ステートメントが入っていないか、または BLOCKING ALL が使用されている。

注:

1. カーソルは、それが読み取り専用ではなく (注 3 参照)、しかも以下のいずれかの項目に該当する場合には更新可能です。
 - 選択ステートメントに、FOR UPDATE OF 文節が含まれている。または
 - プログラム中に、カーソルに対する UPDATE または DELETE WHERE CURRENT OF がある。
2. カーソルは、それが読み取り専用ではなく (注 3 参照)、しかもプログラムに動的ステートメントが含まれており、SQLPREP または SQLBIND で BLOCKING UNAMBIG プリコンパイルまたはバインド・オプションが使用されている場合は、潜在的に更新可能です。
3. カーソルは、次の条件の 1 つまたはいくつかに該当する場合は読み取り専用です。
 - DECLARE CURSOR ステートメントに ORDER BY 文節の指定があり、FOR UPDATE OF 文節の指定はない。
 - DECLARE CURSOR ステートメントに FOR FETCH ONLY 文節の指定がある。
 - 以下の条件のうち 1 つ以上が、カーソルについて、あるいはそのカーソルの参照先の外側の副選択内で参照されているビューまたは論理ファイルについて当てはまる。
 - 外側の副選択に、DISTINCT キーワード、GROUP BY 文節、HAVING 文節、または列関数が含まれている。
 - その選択に結合関数が含まれている。
 - その選択に UNION 演算子が含まれている。
 - その選択に最も外側の副選択のテーブルと同じテーブルを参照する副照会が含まれている。
 - その選択に一時ファイルにコピーしなければならない複合論理ファイルが含まれている。
 - 選択された列のすべてが、式、スカラー関数、または定数である。
 - 参照された論理ファイルの列のすべてが入力専用である。

SQL0969N エラー・メッセージで報告される SQLCODE と関連のトークンの解釈方法は?

DB2 for Linux、UNIX、および Windows で使用されるクライアント・サポートは、同等のコードを持たないホスト SQLCODE およびトークンを報告する場合にメッセージ SQL0969N を戻します。

以下はメッセージ SQL0969N の例です。

```
SQL0969N There is no message text corresponding to SQL error
"-7008" in the Database Manager message file on this workstation.
The error was returned from module "QSQOPEN" with original
tokens "TABLE1  PRODLIB1  3".
```

コードおよびトークンを解釈するには、**メッセージ記述表示 (DSPMSGD)** コマンドを使用します。

オプション 1 (メッセージ・テキストの表示) を選択すると、システムは「定様式メッセージ・テキストの表示」画面を表示します。メッセージの中の 3 つのトークンは、画面では、&1、&2、および &3 によって表されます。この例のメッセージの理由コードは 3 で、これは、画面に示されているコード 3 を指し示します。

```

                                定様式メッセージ・テキストの表示
システム :   KC000
メッセージ ID   . . . . . :   SQL7008
メッセージ・ファイル . . . . . :   QSQLMSG
ライブラリー . . . . . :   QSYS

メッセージ . . . . . :   操作には &2 の &1 が正しくない。
原因 . . . . . :   理由コードは &3 です。理由コードとその意味は次の通りです。
1 -- &1 にメンバーがない。
2 -- &1 は記憶域を解放して保管された。
3 -- &1 はテーブルがジャーナル処理されていないことを示す。このテーブルは
   コミットメント制御の下に処理されるか、そのジャーナルに
   権限のない他のテーブルとは異なるジャーナルにジャーナル処理される。
4 および 5 -- &1 は実動ライブラリーに入っていてそこで作成中であるが、ユー
   ザーはデバッグ・モード UPDPROD(*NO) を指定している。
6 -- コレクションを作成中であるが、ユーザーは UPDPROD(*NO) でデバッグ・モ
   ドになっている。
7 -- ビューの作成で使用された基になったテーブルが無効である。テーブルがプ
   ログラム記述テーブルであるか、または一時ライブラリーに入っています。
続く ...
続行するには、実行キーを押してください。

F3= 終了   F11= 不定様式メッセージ・テキストの表示   F12= 取り消し

```

関連資料:

メッセージ記述表示 (DSPMSGD) コマンド

WHERE 文節のホスト変数のタイプはパフォーマンスにどのような影響を与えるのか?

IBM i オペレーティング・システムでの性能低下の原因として考えられるものの 1 つは、クライアントが C プログラムで、SELECT ステートメントの WHERE 文節の中の比較に浮動小数点変数を使用していることです。

オペレーティング・システムがその列のデータを変換する必要がある場合は、その列の索引は使用できません。比較の中で用いる列、リテラル、およびホスト変数には常に同じタイプを使用するようになります必要があります。データベース内の列がパックまたはゾーン 10 進数として定義されている場合に、ホスト変数が他のタイプになっていると、C で問題が発生する可能性があります。

関連概念:

データベース・パフォーマンスに関するプログラミング手法

ライブラリー・リストを使用して、修飾されていないテーブルおよびビューの名前を解決できるか?

IBM i オペレーティング・システムでは、DB2 for Linux, UNIX, and Windows 製品を使用するクライアント・プログラムなどの、IBM i 以外の DRDA クライアント・プログラムから DB2 for i データにアクセスするときに、オペレーティング・システムのシステム命名オプションを使用するための限定機能をサポートします。

それまでは、異種 DRDA クライアントから接続するときには SQL 命名オプションしか使用できませんでした。システムの命名により、DB2 for i の複数の特性が変更されます。以下にその例を示します。

1. 修飾されていないテーブルおよびビューの名前の解決で、ライブラリー・リストが検索される。
2. CREATE SQL ステートメントを実行するときに、修飾されていないオブジェクトが現行ライブラリーに作成される。
3. ピリオド (.) でなくスラッシュ (/) を使用して、修飾されているオブジェクト名を、それらが入っているライブラリーまたはコレクションから分ける。
4. ある権限特性が変更されている。

詳細は、SQL 解説書のシステム命名に関する項を参照してください。権限に関する説明については、『計画および設計』を参照してください。

DB2 for Linux、UNIX、および Windows は、プリコンパイル (PREP) コマンドと (BIND) コマンドという 2 つのプログラム準備コマンドでの、総称バインド・オプションの指定をサポートします。IBM i 命名は、これらのコマンドのいずれかで、Windows バッチ・ファイルからの抜粋である以下の例のようにして指定できます。

DB2 for Linux、UNIX、および Windows、バージョン 8 以上の場合:

```
DB2 PREP %1.SQC BINDFILE OS400NAMING SYSTEM ...
DB2 BIND %1.BND OS400NAMING SYSTEM ...
```

DB2 for Linux、UNIX、および Windows、バージョン 8 より前のバージョンの場合:

```
DB2 PREP %1.SQC BINDFILE GENERIC 'OS400NAMING SYSTEM' ...
DB2 BIND %1.BND GENERIC 'OS400NAMING SYSTEM' ...
```

Windows 開発プラットフォームでは総称オプションの名/値の組みの前後に単一引用符を使用し、AIX または UNIX プラットフォームでは引用符を使用する必要があります。

注: OS/400 V4R5 および V5R1 では、オプションの名前は AS400NAMING であり、OS400NAMING ではありません。

SYSTEM のほかに有効な OS400NAMING オプションの値は SQL だけです。これはデフォルト値で、このフィーチャーが導入される前は、これが非 IBM i クライアントから使用可能な唯一のオプションでした。

OS400NAMING オプションを (BIND) コマンドで使用し、(PREP) コマンドでは使用しない場合は、システムで SQL エラーが検出されてもバインド・ファイルを作成するよう指示するパラメーターを (PREP) コマンド上にコーディングする必要があります。DB2 for Linux、UNIX、および Windows の場合、このために SQLERROR CONTINUE パラメーターを使用します。この機能は限定と呼ばれていますが、これは、特定の状況では、クライアント・サイドのソフトウェアがリモート・システムで実行することを意図した SQL ステートメントを構文解析することがあるためです。スキーマ ID をテーブル ID と区切るために、システムの命名で求められているように、ピリオドではなくスラッシュを使用すると、そのステートメントは、正しくない構文が含まれているとしてリジェクトされる可能性があります。

関連概念:

40 ページの『計画と設計』

分散リレーショナル・データベースを準備するにあたっては、業務上の要件とリレーショナル・データベース技術の両方を理解していなければなりません。DDM の使用を準備するにあたっては、通信要件、セキュリティ要件、および変更要件を含む複数の要件を満たしている必要があります。

関連資料:

異種クライアントが NLSS ソート・シーケンス、システム命名、別の日時形式などのパッケージ・オプションをオーバーライドする方法

IBM i オペレーティング・システムは、DRDA 総称バインド・オプションをサポートします。

多くの環境オプションはサーバーにとって固有なので、他の製品から提供されたプリコンパイルおよびバインド・コマンドでは使用できません。DRDA は、バインド要求上で総称オプションを渡せるようにして、この欠点に対処しています。DB2 によってサポートされる総称バインド・オプションのリストを以下に示します。

表 31. サポートされている総称バインド・オプション

バインド・オプション名	バインド・オプション値
PATH	<i>schema-names</i>
OS400NAMING	SQL (デフォルト)、SYSTEM
SORTSEQ	HEX (デフォルト)、JOB RUN
MINDIVSCALE	<i>number-char</i>
MAXSCALE	<i>number-char</i>
ROUNDING	HALFEVEN (デフォルト)、CEILING、DOWN、FLOOR、HALFDOWN、HALFUP、UP
EXTENDEDINDICATOR	NO (デフォルト)、YES
DATEOVERRIDE	ISO (デフォルト)、USA、EUR、JIS
CONCURRENTACCESSRESOLUTION	NO (デフォルト)、YES

DB2 for Linux、UNIX および Windows は、プリコンパイル (PREP) コマンドと (BIND) コマンドという 2 つのプログラム準備コマンドでの、総称バインド・オプションの指定をサポートします。以下の例は、時刻形式を日本工業規格にオーバーライドし、日付形式を欧州規格にオーバーライドします。この処理が行われる理由は、DATETIME が同じ形式であることが前提とされ、DB2 ではそれが必須ではないからです。この例は Windows バッチ・ファイルからの抜粋です。

```
DB2 PREP %1.SQC BINDFILE DATETIME JIS GENERIC "DATEOVERRIDE EUR"...
DB2 BIND %1.BND DATETIME JIS GENERIC "DATEOVERRIDE EUR"...
```

注: Windows 開発プラットフォームでは、総称オプションの名/値の組みの前後に単一引用符が使用されますが、AIX または UNIX プラットフォームでは引用符が使用されません。


照会を実行するときに、戻される行がない理由は？

この問題の原因の可能性の 1 つは、DB2 for Linux, UNIX, and Windows データベース通信サービス・ディレクトリーでの IBM i プラットフォームに関する項目の追加が失敗したということです。

DB2 for IBM i との対話には、どのレベルの DB2 for Linux, UNIX, and Windows が必要ですか？

対話には以下のフィックスパックが必要です。

- DB2 for Linux, UNIX and Windows バージョン 7 フィックスパック 10
- DB2 for Linux, UNIX and Windows バージョン 8 フィックスパック 4

これらのフィックスパックは、DB2 9 for Linux, UNIX, and Windows  Web サイトから入手できます。

どのようにすれば IBM i プラットフォームでの DB2 for Linux, UNIX, and Windows のスクロール可能なカーソルのサポートを使用可能にできるか?

DB2 for Linux, UNIX, and Windows V8 クライアントでは、フィックスパック 4 以降を使用していなければなりません。スクロール可能なカーソルのサポートを IBM i プラットフォームに提供する新バージョンの DB2 for LUW があり、この新バージョンの DB2 for LUW を実行するクライアントも使用できます。

DB2 for Linux, UNIX, and Windows V8 を実行するクライアントでフィックスパック 4 を使用している場合は、以下のアクションの**いずれか**を行ってください。

注: スクロール可能なカーソルのサポートを IBM i プラットフォームに提供する新バージョンの DB2 for LUW クライアントを使用している場合は、これらのアクションを実行する必要はありません。

- 次のコマンドを実行します。

```
UPDATE CLI CFG FOR SECTION dbname USING CURSORTYPES 1
```

dbname は IBM i データベースの名前です。

- 次の構文を使用して、db2cli.ini ファイルを編集します。

```
CURTYPES = 1
```

異種環境での相互運用に関するその他のヒント

このトピック・コレクションでは、DB2 for IBM i を DB2 for Linux, UNIX, および Windows とともに使用する場合は、他の情報を提供します。

DB2 for Linux, UNIX, および Windows と対比した DB2 Connect™

アプリケーション・リクエスト (クライアント) 機能に対して、DRDA アプリケーション・サーバー機能を実行するにはどのプロダクトが必要なのか、ユーザーが混乱する場合があります。AR DB2 Connect という場合があります、AR と AS を両方とも DB2 for Linux, UNIX, および Windows という場合があります。DB2 の中には、以下が含まれます。

- DB2 for AIX
- DB2 for HP-UX
- DB2 for Linux
- DB2 for Sun Solaris
- DB2 for Windows

適切な構成および保守レベル

プロダクト・マニュアルにあるインストールおよび構成の指示に必ず注意深く従ってください。プロダクトのレベルが最新のものであることを確認してください。まだ行っていない場合は、適切な fix パックを適用してください。

テーブル名およびコレクション名

DRDA アプリケーションがアクセスする SQL テーブルには 3 部からなる名前があります。最初の部分はデータベース名、2 番目はコレクション ID、そして 3 番目はベース・テーブル名です。最初の 2 つの部分はオプションです。DB2 for i は、テーブル名を第 2 レベルで、コレクション名 (またはライブラリー名) によって修飾します。テーブルは、DB2 for i データベースに常駐します。

DB2 では、テーブルは、ユーザー ID (そのテーブルの作成者のユーザー ID) によって与えられ、プラットフォーム上の、複数のデータベースのいずれかに常駐する可能性があります。DB2 for Linux、UNIX、および Windows にも、コレクション ID に対するユーザー ID の使用に関して同じことが言えます。

DB2 for Linux、UNIX、and Windows から DB2 for i に対して行われる動的照会の際、指定された照会先のテーブル名にコレクション名がない場合、照会ではデフォルトのコレクション名にターゲット側ジョブ (IBM i オペレーティング・システム上) のユーザー ID が使用されます。これは、ユーザーにとって予想外である場合があります、このために、テーブルが見つからない可能性もあります。

DB2 for i から DB2 に対して行われる動的照会の際、*qualifier.table-name* という形式の照会で指定されない場合、暗黙のテーブル修飾子を持つ場合があります。第 2 レベルの DB2 テーブル修飾子は、デフォルトにより、照会を作成するユーザーのユーザー ID になります。

DB2 データベースおよびテーブルを、共通ユーザー ID で作成したい場合があります。しかし、DB2 では DB2 for i のような物理コレクションが存在しないことにご注意ください。テーブル修飾子しかなく、それは作成者のユーザー ID です。

APPC 通信のセットアップ

AR として DB2 for Linux、UNIX、および Windows のいずれか、または AS として DB2 を APPC とともに使用する際には、ワークステーション用に作成されたコントローラーおよび装置とともに、IBM i 通信を適切に構成する必要があります。

RDB ディレクトリーのセットアップ

IBM i 製品と接続する各 DB2 データベースの RDB ディレクトリーに項目を追加するには、リレーショナル・データベース項目の追加 (ADDRDBDIRE) コマンドを使用します。RDB 名は DB2 データベース名になります。

APPC 通信を使用する場合、リモート・ロケーション名はワークステーション名になります。

TCP/IP を使用する場合は、リモート・ロケーション名はワークステーションのドメイン名、またはその IP アドレスになります。DRDA サーバーが使用するポートは、通常、IBM i オペレーティング・システムが使用する DRDA ポート、446 ではありません。

DB2 プロダクトの資料を調べて、ポート番号を判別してください。共通値として 50000 が使用されています。以下は、RDB 項目が DB2 サーバーに対して適切に構成されたことを示す DSPRDBDIRE 表示画面の例です。

```
リレーショナル・データベース項目の詳細の表示
リレーショナル・データベース . . . . . : SAMPLE
リモート・ロケーション:
  リモート・ロケーション . . . . . : 9.5.36.17
  タイプ . . . . . : *IP
ポート番号またはサービス名 . . . . . : 50000
テキスト . . . . . : My DB2 server
```

DB2 for Linux、UNIX、および Windows および IBM DB2 Universal Driver for SQLJ and JDBC で使用される NULLID パッケージはどのように作成しますか？

DB2 for Linux, UNIX, and Windows を使用して、DB2 for i 上のデータにアクセスする前に、IBM i SQL パッケージをアプリケーション・プログラムおよび DB2 for Linux, UNIX, and Windows ユーティリティ一用に作成する必要があります。

DB2 (PREP) コマンドを使用すれば、組み込み SQL を持つアプリケーション・プログラムのソース・ファイルを処理することができます。この処理により、SQL ステートメントのホスト言語呼び出しを含む変更ソース・ファイルが作成されます。そしてデフォルトで、現在の接続先データベースに SQL パッケージを作成します。

DB2 for Linux, UNIX, and Windows を DB2 for i サーバーに結合するには、次のステップを実行します。

1. CONNECT TO rdbname
2. Bind path@ddcs400.lst BLOCKING ALL SQLERROR CONTINUE MESSAGES DDACS400.MGS GRANT PUBLIC

上記の path@ddcs400.lst パラメーターの 'path' を、デフォルトのパス C:¥SQLLIB¥BND¥ (非 INTEL のプラットフォームの場合は c:/sqllib/bin/) に置き換えてください。また、インストール先がデフォルトのディレクトリでなかった場合は、インストール時に指定した値に置き換えてください。

3. CONNECT RESET

対話式 SQL パッケージはどのようにセットアップしますか？

対話式 SQL を使用するためには、IBM DB2 Query Manager and SQL Development Kit for i プロダクトを IBM i にインストールしておく必要があります。DB2 上のデータにアクセスするには、次のことを行います。

1. STRSQL でセッションを開始する際、NAMING(*SQL)、DATFMT(*ISO)、および TIMFMT(*ISO) のセッション属性を使用します。*ISO 以外の形式 (すべてではない)、および日付形式 (DATFMT) で使用される形式を、時刻形式 (TIMFMT) でも使用しなければなりません。
2. IBM i オペレーティング・システムでのスキーマと、DB2 でのテーブル修飾子 (作成者のユーザー ID) との対応関係に留意してください。
3. 最初の対話式セッションでは、DB2 で作成されるパッケージを入手するために、以下のような順序で SQL ステートメントを実行しなければなりません。(1) RELEASE ALL、(2) COMMIT、(3) CONNECT TO rdbname ('rdbname' は、特定のデータベースに置き換えられます)。

対話式 SQL の使用のセットアップの一環として、GRANT EXECUTE ON PACKAGE QSQL400.QSQLabcd TO PUBLIC (または特定のユーザー) ステートメントを実行し、他のユーザーも対話式 SQL 用の PC で作成された SQL PKG を使用できるようにしてもよいでしょう。以下の GRANT ステートメントの 'abcd' に入る実際の値は、以下のテーブルから判別できます。そこには、パッケージが作成されたときの、様々なオプション・セットの実際のパッケージ名が示されています。たとえば、パッケージが作成されたときに以下のオプションが使用されている場合は、GRANT EXECUTE ON PACKAGE QSQL400.QSQL0200 TO some-user ステートメントを、ある特定のユーザーに対して実行することもできます。*ISO 日付、*ISO 時刻、*CS コミットメント制御、単一引用符ストリング区切り文字、1 バイト文字サブタイプ。

位置	オプション	値
a	日付形式	0 = ISO、JIS 日付形式。 1 = USA 日付形式。 2 = EUR 日付形式。
b	時刻形式	0 = JIS 時刻形式。 1 = USA 時刻形式。 2 = EUR、ISO 時刻形式。
c	コミットメント制御小数点文字	0 = *CS コミットメント制御ピリオド小数点文字。 1 = *CS コミットメント制御コンマ小数点文字。 2 = *RR コミットメント制御ピリオド小数点文字。 3 = *RR コミットメント制御コンマ小数点文字。
d	ストリング区切り文字のデフォルト文字サブタイプ	0 = 単一引用符ストリング区切り文字、1 バイト文字サブタイプ。 1 = 単一引用符ストリング区切り文字、2 バイト文字サブタイプ。 2 = 引用符ストリング区切り文字、1 バイト文字サブタイプ。 3 = 引用符ストリング区切り文字、2 バイト文字サブタイプ。

QUERY のクローズ

DB2 for Linux, UNIX, and Windows は、暗黙的、明示的を問わず Query が終了されたときに読み取りロックの解放を要求するオプションが追加されました。なお、システムがこの要求を尊重しない場合、それはエラーとはみなされません (IBM i 製品の場合)。DB2 for Linux、UNIX、および Windows は、読み取る行がなくなったとき、システムで非スクロール可能カーソルに対して暗黙的に Query を終了させるかどうかを指定する、別のオプションがあります。これは、システムによって前もって決定されます。IBM i AS では、この機能がサポートされています。

異種環境でのユーザー ID とパスワードの最大長はどうなりますか？

アプリケーション・リクエスト (AR) として稼働する DB2 for i では、異種のアプリケーション・サーバー (AS) に対して実行される時、10 文字以上のユーザー ID とパスワードを使用することができます。正確な文字数の制限は、使用されている特定のインターフェースの記述で指定されます。例として、SQL CONNECT ステートメントの場合の制限については、DB2 for IBM i SQL 解説書のトピックを参照してください。

DB2 Server for VM 対話式 SQL パッケージの作成

DB2 Server for VM では、コレクション名はユーザー ID と同義です。DB2 Server for VM アプリケーション・サーバー上で対話式 SQL または DB2 for i Query Manager によって使用されるパッケージを作成するには、IBM i オペレーティング・システム上で QSQL400 というユーザー ID を作成してください。このユーザー ID を使用すれば、DB2 Server for VM アプリケーション・サーバー上で必要なパッケージをすべて作成することができます。そうすれば、ユーザーは自分自身のユーザー ID を使用して、IBM i 上で、対話式 SQL または DB2 Query Manager によって Server for VM にアクセスすることができます。

DRDA と DDM のトラブルシューティング

分散リレーショナル・データベースへのアクセスで問題が生じた場合、問題の性質を判別し、問題がアプリケーションによるものか、ローカル・システムまたはリモート・システムによるものかを判別してください。

問題を解決するか、お客様サポートの援助を受けて問題を解決するには、以下のスキルが必要です。

- IBM i ライセンス・プログラムについて理解する。
- 問題がクライアント上またはサーバー上のどちらにあるかを判別する。
- IBM i の問題管理機能の使用方に習熟している。

分散リレーショナル・データベースでの問題の診断方法の詳細は、『分散関係データベース 問題判別の手引き』を参照してください。

関連概念:

334 ページの『DRDA と DDM のパフォーマンス』

どのような種類のアプリケーション・プログラムをシステムで実行する場合でも、パフォーマンスは常に重要な問題となります。分散リレーショナル・データベースの場合、ネットワーク、システムおよびアプリケーションのパフォーマンスすべてが重要になります。

IBM i 問題処理の概要

ローカルおよびリモート IBM i プラットフォームで、ユーザー検出またはシステム検出のどちらの問題が発生した場合でも、その問題を管理するのに IBM i ライセンス・プログラムが役立ちます。

IBM i オペレーティング・システムは、問題ログと問題管理機能を使って、ユーザー検出とシステム検出の両方の問題を追跡します。問題が検出されたとき (OPENED) から、それが解決される (CLOSED) まで、問題プログラム状態は保持されるので、追跡に役立てることができます。

問題処理のサポートには次のものが含まれます。

- 初期問題処理に関する情報が含まれるメッセージ
- システム検出の問題の自動警報
- 統合問題ロギングおよび追跡
- 第 1 障害データ検知 (FFDC)
- エレクトロニック支援のサービス要求
- エレクトロニック支援、プログラム一時修正 (PTF) 要求

システム検出問題

IBM i オペレーティング・システムとその接続装置は、特定のタイプの問題を検出することができます。これをシステム検出問題と呼びます。問題が検出されると、次のようないくつかの操作が行われます。

- プロダクト活動記録ログ内に項目が作成される
- 問題記録が作成される
- メッセージが QSYSOPR メッセージ待ち行列に送られる

情報は、エラー・ログと問題記録に記録されます。深刻な問題が検出された場合は、FFDC 情報のスプール・ファイルも作成されます。エラー・ログと問題記録には、次のような情報が入ります。

- 重要プロダクト・データ
- 構成情報
- 参照コード
- 対応する装置の名前
- 補足的な障害情報

ユーザー検出問題

通常、ユーザー検出問題は、次のような問題を引き起こす原因となるプログラム・エラーに関連したものです。

- ジョブ問題
- 正しくない出力
- プログラム障害を示すメッセージ
- システムでは検出されない装置障害
- 劣悪なパフォーマンス

ユーザーが問題を検出しても、問題分析を実行するか、または、操作援助機能 (Operational Assistant) の「USERHELP」メニューで、問題の解決に役立つ情報を保管するためのオプションを選択しない限り、サーバーでは何も情報は集められません。

関連概念:

『システムおよび通信の問題』

システムまたはその通信に関する問題が起こるとき、メッセージが生成されます。システムが検出した問題は、問題ログに自動的に入れられ、それらの問題を表示したり分析したりできます。

システムおよび通信の問題

システムまたはその通信に関する問題が起こるとき、メッセージが生成されます。システムが検出した問題は、問題ログに自動的に入れられ、それらの問題を表示したり分析したりできます。

任意のコマンド行から**問題分析 (ANZPRB)** コマンドを入力すれば、記録されている問題の分析をいつでも実行することができます。このコマンドは、分析手順を順にたどって、問題関連の追加情報を問題ログに保管します。

問題ログを見るには、**問題処理 (WRKPRB)** コマンドを使います。

問題ログを使用すると、ローカル・システム上で記録されたすべての問題を表示できます。また、次の項目に示すような特定の問題に関する詳細情報を表示することもできます。

- 問題が生じている装置の型式および製造番号
- 問題の日付および時刻
- 障害を起こした部品およびその位置
- 問題状況

問題ログを使って、問題を分析したり、問題を報告したり、行われたすべてのサービス活動を判別したりすることができます。

関連概念:

397 ページの『IBM i問題処理の概要』

ローカルおよびリモート IBM i プラットフォームで、ユーザー検出またはシステム検出のどちらの問題が発生した場合でも、その問題を管理するのに IBM i ライセンス・プログラムが役立ちます。

関連資料:

問題分析 (ANZPRB) コマンド

問題処理 (WRKPRB) コマンド

DRDA アプリケーション問題

アプリケーションでの問題を処理するには、実動段階に入る前が最も適した時期です。しかし、アプリケーションが一般の使用に供されるとき、そのアプリケーションで起こり得る状態をすべて予測することは不可能です。

分散リレーショナル・データベース・アプリケーションの実行時に検出される問題は、不正な出力またはアプリケーションが予期した時間内に終了しないという、2 つの一般的な徴候を示す可能性があります。これらのトピックの図は、アプリケーション・プログラム問題、パフォーマンス関連問題、およびシステム関連問題のいずれかに問題を一般的に分類する方法を示しています。このように分類すれば、IBM i の標準の問題分析方法を使って、問題を解決できます。

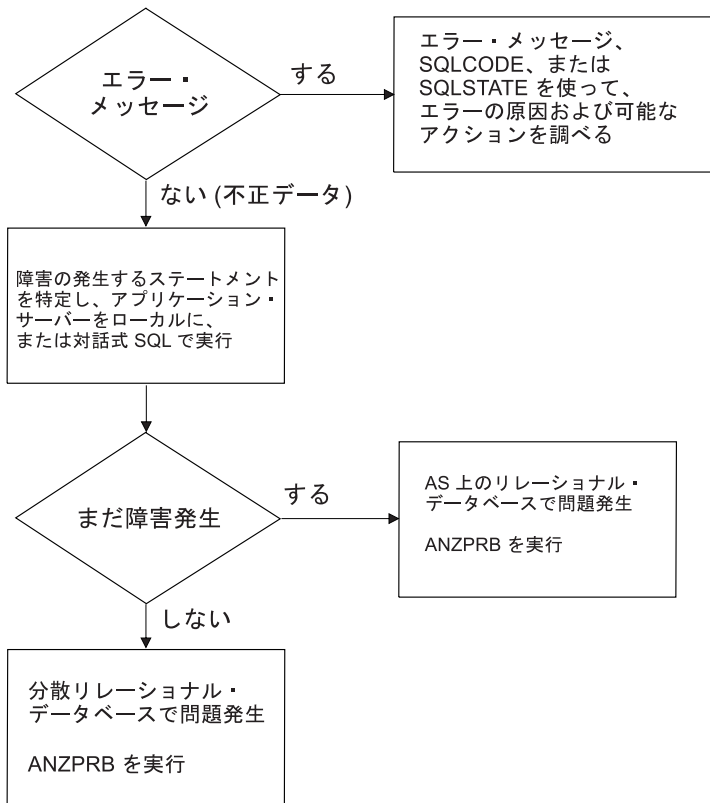
アプリケーション・リクエスター (AR) またはアプリケーション・サーバー (AS) のジョブ・ログに、パッケージに障害が起きたことが示されていたり、プログラムまたはパッケージのリストに、その障害の理由が示されていたりすることがあります。SQL コンパイラには診断テストが用意されています。そこでは、プリコンパイル・プロセスで生成された SQLCODE および SQLSTATE が診断リスト内に示されます。

統合言語環境 (ILE) のプリコンパイルでは、任意で OPTION(*XREF) および OUTPUT(*PRINT) を指定して、プリコンパイル・ソースと相互参照表を印刷できます。ILE 以外のプリコンパイルでは、SQL プログラムの作成 (CRTSQLxxx) コマンドの OPTIONS パラメーターに任意で *SOURCE および *XREF を指定して、プリコンパイルおよび相互参照を印刷することができます。

不正出力の問題の解決

エラー・メッセージ、SQLCODE、または SQLSTATE を使用して、不正出力がある場所の問題の原因を判別してください。

不正出力の問題を分離するプロセスについては、以下の図を参照してください。不正出力の問題に関するエラー・メッセージを受け取った場合、メッセージの説明に、どのような問題であるかが示され、訂正アクションが指示されています。エラー・メッセージが出なかった場合、分散リレーショナル・データベースが障害の原因かどうかを自分で判断しなければなりません。原因を判別するには、障害が起きたステートメントをローカル・アプリケーション・サーバー (AS) 上で実行するか、または対話式構造化照会言語 (ISQL) を使ってステートメントを AS 上で実行します。問題をローカルで生成できる場合、問題の原因は分散リレーショナル・データベース・サポートにはありません。この操作の結果に応じて、IBM i の問題分析方法を使って、サポート担当者に該当情報を提出してください。



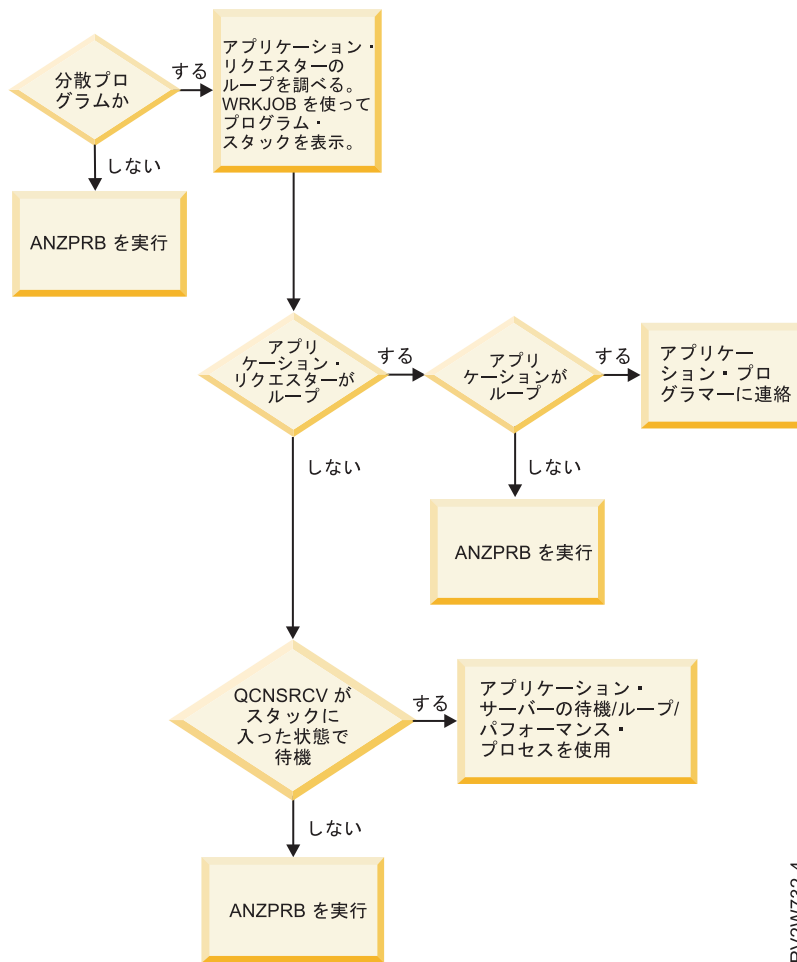
RV2W731-1

図 44. 不正出力の問題の解決

ループ、待機、またはパフォーマンスの問題の解決

要求が完了するのに想定された時間より長かかっている場合、まずアプリケーション・リクエスター (AR) を検査します。

1. ジョブ・ログを調べて、リレーショナル・データベースへの接続が完了したことを知らせるメッセージ SQL7969 を確かめます。これは、アプリケーションが分散リレーショナル・データベース・アプリケーションであることを知らせるメッセージです。
2. システムまたはアプリケーションが待ちまたはループになっているかどうかを判別します。
 - a. **ジョブ処理 (WRKJOB)** コマンドを使って、プログラム・スタックを表示し、AR を検査してループを探します。
 - b. そのプログラム・スタックを調べて、システムがループになっているかどうかを判別します。アプリケーションそのものがループになっている場合、解決のためアプリケーション・プログラマーに連絡してください。
3. アプリケーション・リクエスターが待ちになっているかどうかを判別します。
 - スタック上に QAPDEQUE および QCNSRCV が見つかった場合、AR はアプリケーション・サーバー (AS) 待ちになっています。
 - システムが通信待ち状態でない場合、問題分析手順を使って、パフォーマンス上の問題または待ちの問題があるかどうかを判別します。



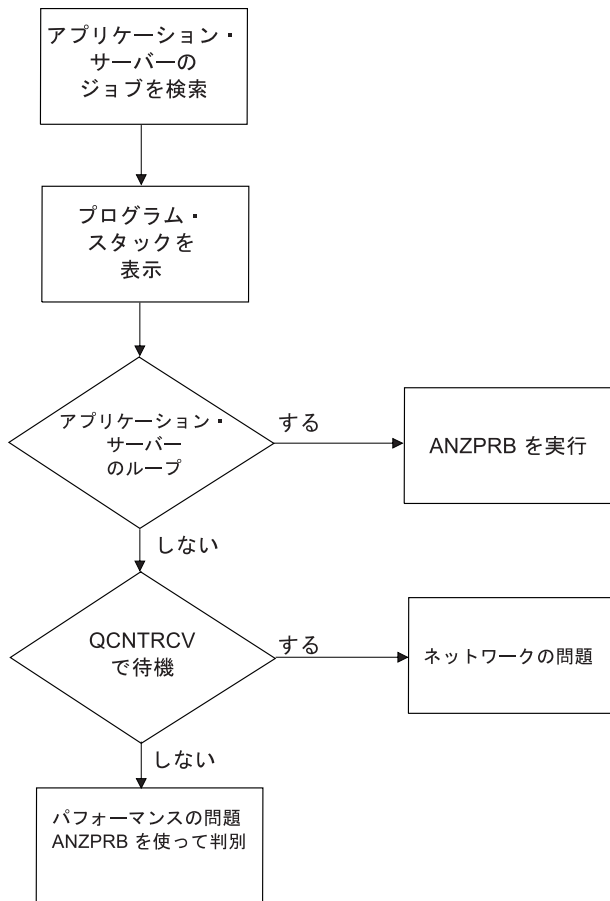
RV2W732-4

図 45. アプリケーション・リクエスター側のループ、待機、またはパフォーマンスの問題の解決

AS 上のジョブ・ログを探せば、AR ジョブ名を見つけることができます。AS ジョブを検査する必要が生じたときは、**ジョブ処理 (WRKJOB)**、**WRKACTJOB (活動ジョブの処理)**、または**ユーザー・ジョブ処理 (WRKUSRJOB)** コマンドを使って、AS 上のジョブを見つけ出します。

これらのジョブのうちのいずれかの表示でプログラム・スタックを調べて、AS がループになっているかどうかを確認します。

- ループになっている場合、問題分析を使って問題を処理します。
- ループになっていない場合、プログラム・スタックが、QCNSRCV での WAIT になっているかどうかを調べます。そうなっていると、AS は AR 待ちであることを意味します。
- どちらのシステムもこのような通信待ち状態であれば、ネットワークに問題があるということになります。
- AS が待ち状態でなければ、パフォーマンス上の問題があるということです。



RV2W733-1

図 46. アプリケーション・サーバー側の待機、ループ、またはパフォーマンスの問題の解決

照会のパフォーマンスの低下に共通する 2 つの原因として以下のことが挙げられます。

- アクセスされているテーブルに、索引がない。そのような場合、適切なフィールドをキーとして使って索引を作成します。
- 照会要求で戻された行がブロック化されていない。行がブロック化されているかどうかによって、照会のパフォーマンスに大きな差が生じることがあります。アプリケーションを調整して、ブロック化を利用するようにアプリケーションを調整します。

DB2 JDBC Universal Driver または DB2 for Linux, UNIX, and Windows などの製品を使用して、ワークステーションから DB2 for IBM i に初めて接続する際、その製品用の SQL パッケージが DB2 for IBM i 内にあらかじめ作成されていない場合は、そのパッケージが自動的に作成されます。NULLID コレクションも同じように自動的に作成される必要がある場合があります。このため、初期接続後に発行される最初の SQL ステートメントの 1 つに関して、システムからの応答が遅れが生じます。

TCP/IP を介して接続しようとしているシステムが使用可能になっていないと、長い遅延が生じることがあります。数分間の遅延の後、メッセージ A remote host did not respond within the timeout period が出されます。RDB ディレクトリー内の IP アドレスが正しくない場合にも、遅延が生じることがあります。

関連概念:

336 ページの『DRDA のブロック化に影響を与える要因』

パフォーマンスに影響する最も重要な要因は、アプリケーション・リクエスター (AR) とアプリケーション・サーバー (AS) の間でデータが転送されるときに、ブロック化が行われるかどうかということです。データのブロックとして伝送される行のグループの場合、同じデータを一度に 1 行ずつ伝送する場合と比べて、通信のオーバーヘッドははるかに少なく済みます。

関連タスク:

273 ページの『分散リレーショナル・データベースのジョブの探索』

クライアント上で分散リレーショナル・データベース・ジョブについての情報を探索する時に、使用されているユーザー・プロファイルが分かっている場合には、ユーザー・ジョブの処理 (WRKUSRJOB) コマンドを使用することによって、そのジョブを見つけることができます。

266 ページの『分散リレーショナル・データベースにおけるジョブ処理』

ジョブ処理 (WRKJOB) コマンドによって、「ジョブ処理」メニューが表示されます。このメニューを使用すれば、オプションを選択して、指定されたジョブに関連する情報を処理または変更することができます。現在使用しているジョブについての情報を入手する場合には、パラメーターを指定しないでコマンドを入力します。

267 ページの『分散リレーショナル・データベースにおけるユーザー・ジョブの処理』

ジョブにより使用されているユーザー・プロファイル (ユーザー名) を知る場合、**ユーザー・ジョブの処理 (WRKUSRJOB)** コマンドを使用して、ジョブ情報を表示または変更できます。自分のユーザー・プロファイルがあるシステムのジョブのリストを入手する場合には、パラメーターを指定しないでコマンドを入力してください。

269 ページの『分散リレーショナル・データベースにおける活動ジョブの処理』

数人のユーザーに対して実行されているジョブをモニターする場合や、ジョブを探しているものの、ジョブの名前やユーザー ID がわからない場合は、**WRKACTJOB (活動ジョブの処理)** コマンドを使用してください。

リスト作成

SQL プログラムの作成 (CRTSQLxxx) コマンドによって作成されるリストには、次のような情報が含まれています。

- プリコンパイル・コマンドのパラメーターに提供される値
- ソース・プログラム
- 識別コード相互参照
- プリコンパイルの結果のメッセージ

プリコンパイラー・リストの作成:

以下はプリコンパイラー・リストの例です。

```

57xxST1 VxRxMx yymmdd          SQL ILE C オブジェクトの作成 UPDATEPGM 02/10/06 14:30:10   ページ   1
ソース仕様タイプ.....C
オブジェクト名.....TST/UPDATEPGM
ソース・ファイル.....*LIBL/QCSRC
メンバー.....*OBJ
オプション.....*XREF
プログラムの印刷.....*PRINT
ターゲット・リリース.....*CURRENT
INCLUDE ファイル.....*LIBL/*SRCFILE
コミット.....*CHG
データのコピー可能.....*YES
SQL カーソルのクローズ.....*ENDACTGRP
ブロック化可能.....*READ
PREPARE 遅延.....*NO
生成レベル.....10
マージン.....*SRCFILE
印刷装置ファイル.....*LIBL/QSYSPRT
日付の形式.....*JOB
日付区切り記号.....*JOB
時刻の形式.....*HMS
時刻区切り記号.....*JOB
置き換え.....*YES
リレーショナル・データベース...RCHASLKM
ユーザー .....*CURRENT
RDB 接続方式.....*DUW
省略時のコレクション.....*NONE
パッケージ名.....*OBJLIB/*OBJ
作成オブジェクト・タイプ.....*PGM
デバッグ・ビュウ.....*NONE
動的ユーザー・プロファイル....*USER
ソート順序.....*JOB
言語 ID.....*JOB
IBM SQL フラグづけ.....*NOFLAG
ANS フラグ付け.....*NONE
テキスト.....*SRCMBRTXT
ソース・ファイルの CCSID.....37
ジョブの CCSID.....65535
02/10/06 14:25:33 にソース・メンバーが変更された。

```

```

57xxST1 VxRxMx yymmdd          SQL ILE C オブジェクトの作成 UPDATEPGM 02/10/06 14:30:10   ページ   2
レコード*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 SEQNBR 最終変更
1  /*****/ 100
2  /* This program is called to update the DEPTCODE of file RWDS/DPT1 */ 200
3  /* to NULL. This is run once a month to clear out the old */ 300
4  /* data. */ 400
5  /* */ 500
6  /* NOTE: Because this program was compiled with an RDB name, it is */ 600
7  /* not necessary to do a connect, as an implicit connect will take */ 700
8  /* place when the program is called. */ 800
9  /*****/ 900
10 #include <stdio.h> 1000
11 #include <stdlib.h> 1100
12 exec sql include sqlca; 1200
13 1300
14 main() 1400
15 { 1500
16     /* Just update RWDS/DPT1, setting deptcode = NULL */ 1600
17     exec sql update RWDS/DPT1 1700
18         set deptcode = NULL; 1800
19 } 1900
* * * * * ソースの終わり * * * * *-->

```

図 47. プリコンパイラーによるリスト

図 48. プリコンパイラーによるリスト (続き)

```

57xxST1 VxRxMx yymdd      SQL ILE C オブジェクトの作成 UPDATEPGM 06/04/01 14:30:10 ページ 3
相互参照
データ名                  定義      参照
DEPTCODE                  ****      COLUMN
18
DPT1                      ****      TABLE IN RWDS
17
RWDS                      ****      COLLECTION
17
57xxST1 VxRxMx yymdd      SQL ILE C オブジェクトの作成 UPDATEPGM 06/04/01 14:30:10 ページ 4
診断メッセージ
MSG ID SEV レコード テキスト
SQL0088 0 17 桁 15 UPDATE はテーブル全体に適用される。
SQL1103 10 17 RWDS 中のファイル DPT1 のフィールド定義が見つかりません。
メッセージの要約
合計 情報 警告 エラー 重大 打ち切り
2 1 1 0 0 0
ソースに 10 レベルの重大度エラーが見つかった。
19 ソース・レコードが処理された。
***** リストの終わり *****

```

CRTSQLPKG リスト:

構造化照会言語パッケージの作成 (CRTSQLPKG) コマンドからのリストの例には、以下のタイプの情報が含まれています。

- コマンドのパラメーターに使用される値
- エラーのステートメント (ある場合)
- 構造化照会言語パッケージの作成 (CRTSQLPKG) コマンドを実行した結果のメッセージ

次の図に、この情報を示します。

```

57xxST1 VxRxMx yymdd      Create SQL package 02/10/06 14:30:31 Page 1
レコード*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 SEQNBR 最終変更
プログラム名.....TST/UPDATEPGM
リレーショナル・データベース..*PGM
ユーザー .....*CURRENT
置き換え.....*YES
省略時のコレクション.....*PGM
生成レベル.....10
印刷装置ファイル.....*LIBL/QSYSPRT
オブジェクト・タイプ.....*PGM
モジュール・リスト.....*ALL
テキスト.....*PGMTXT
ソース・ファイル.....TST/QCSRC
メンバー.....UPDATEPGM
57xxST1 VxRxMx yymdd      Create SQL package 02/10/06 14:30:31 Page 2
レコード*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 SEQNBR 最終変更
17 UPDATE RWDS / DPT1 SET deptcode = NULL
診断メッセージ
MSG ID SEV レコード テキスト
SQL0204 10 17 桁 17 RWDS のタイプ *FILE の DPT1 が見つからない。
SQL5057 TST の SQL パッケージ UPDATEPGM がモジュール UPDATEPGM から KC000 に作成された。
メッセージの要約
合計 通知 警告 エラー 重大度 打ち切り
1 0 1 0 0 0
ソースに 10 レベルの重大度エラーが見つかった。
***** リストの終わり *****

```

図 49. CRTSQLPKG によるリスト

関連資料:

構造化照会言語パッケージの作成 (CRTSQLPKG) コマンド

SQLCODE および SQLSTATE

SQL を使用するプログラム・インターフェースは、エラーが発生したとき、エラー情報をアプリケーション・プログラムに戻します。SQLSTATE とそれに対応する SQLCODE は、SQL 通信域 (SQLCA) か SQL 診断域にあるアプリケーション・プログラムに戻されます。

SQLCA は、アプリケーションによって提供されたスペースの制御ブロックにある変数の集まりであり、これは、データベース管理システムによって、最後に実行された SQL ステートメントに関する情報で更新されます。SQL 診断域は、データベース・マネージャーによって提供されたスペースにある、もっと複雑なストレージ域で、最後に更新された SQL ステートメントに関するより広範な情報を通信するように設計されています。

SQL エラーが検出されると、そのエラーの性質は、SQLSTATE と呼ばれる 5 文字のグローバル変数によって識別されます。また、SQLSTATE に加えて、整数 SQLCODE も使用できます。ただし、SQLCODE は、現行の 4 種類の IBM リレーショナル・データベース製品において、同じエラー条件に対して同じ戻りコードを戻すわけではありません。SQLSTATE は、アプリケーションが接続する DB2 製品が何であるかに関係なく、アプリケーション・プログラムが特定のエラー条件やエラーのクラスをテストできるように、設計されています。

ステートメントの処理時に SQL がハード・エラーを検出すると、SQLCODE は負の数 (たとえば、SQLCODE -204) になります。ステートメントの処理時に SQL が、例外ではあるが有効な条件 (警告) を検出すると、SQLCODE は正の数 (たとえば、SQLCODE +100) になります。ステートメントの処理時に SQL が何もエラーや例外条件を検出しないと、SQLCODE は 0 になります。どの DB2 for i SQLCODE にも、対応するメッセージが、ライブラリー QSYS のメッセージ・ファイル QSQLMSG 内にあります。たとえば、SQLCODE -204 は、メッセージ ID SQL0204 として記録されます。

戻されるエラー情報は貴重な問題診断ツールであるので、戻された SQLCA や SQL 診断域に入っている特定の情報を表示するのに必要な指示を、アプリケーション・プログラムに組み込んでおくといでしょう。また、ここで扱うメッセージ・トークンも、問題診断に非常に有用です。

- SQLSTATE

戻りコード。

- SQLCODE (SQLCA) または DB2_RETURNED_SQLCODE (SQL 診断域)

戻りコード。

- SQLERRD(3) (SQLCA) または ROW_COUNT (SQL 診断域)

SQL によって更新、挿入、または削除された行数。

メッセージ記述表示 (DSPMSGD) コマンドを使って、すべてのメッセージをオンラインで見ることができます。

関連概念:

SQL メッセージおよびコード

関連資料:

SQL リファレンス

メッセージ記述表示 (DSPMSGD) コマンド

分散リレーショナル・データベース SQLCODE および SQLSTATES:

リストは、分散リレーショナル・データベース処理に関連した一部の共通の SQLCODE と SQLSTATE を示しています。

ここでの簡単な SQLCODE (およびそれに関連した SQLSTATE) の説明では、メッセージ・データ・フィールドはアンパーサンド (&) と数字で示されています (たとえば &1)。このフィールドの置換テキストは、アプリケーション・プログラムが SQLCA を使用していれば SQLCA 内の SQLERRM に、アプリケーション・プログラムが SQL 診断域を使用していれば DB2_ORDINAL_TOKEN_n (n はトークン番号) に保管されます。**メッセージ記述表示 (DSPMSGD)** コマンドを使って、すべての SQLCODE のさらに詳細な原因と回復情報を検索することができます。

表 32. SQLCODE および SQLSTATE

SQLCODE	SQLSTATE	説明
+100	02000	この SQLSTATE は、空テーブル上での SQL 操作のため、「No Data」という例外警告を報告する。SQL UPDATE または SQL DELETE ステートメント内で識別されるゼロ行、あるいは SQL FETCH ステートメント内のカーソルが結果テーブルの最終行の後にあった。
+114	0A001	リレーショナル・データベース名 &1 が現行システム &2 と同じではない。
+304	01515	この SQLSTATE は、ホスト変数が検索値を保持するために十分な大きさでなかったために起こったホスト変数リストまたは構造に、FETCH または SELECT での警告を報告する。FETCH または SELECT が、指定された SELECT 項目にデータを戻さない。標識変数は、NULL 値の戻りを示すために -2 に設定される。処理が続行される。
+331	01520	文字変換が実行できない。
+335	01517	文字変換の結果が置換文字になった。
+551	01548	タイプ &3 の &2 の中のオブジェクト &1 に対して許可されていない。
+552	01542	&1 に対して許可されていない。
+595	01526	コミット・レベル &1 が &2 ロックにエスカレートした。
+802	01519	この SQLSTATE は、SQL 選択ステートメントの SELECT リスト、SELECT、UPDATE、または DELETE ステートメントの検索条件、あるいは UPDATE ステートメントの SET 文節内にあった SQL 算術関数または算術式の処理中に起こった算術例外警告を報告する。エラーのあるそれぞれの式ごとに、標識変数は、NULL 値の戻りを示すために -2 に設定される。関連したデータ変数は未変更のままである。処理が続行される。
+863	01539	リレーショナル・データベース &1 には SBCS 文字だけが許可されている。
+990	01587	この SQLSTATE は、2 フェーズ処理中に少なくとも 1 人の参加者からの保留応答または混合結果を報告する。
+30104	01615	バインド・オプションは無視される。
-114	42961	リレーショナル・データベース &1 が現行システム &2 と同じではない。

表 32. SQLCODE および SQLSTATE (続き)

SQLCODE	SQLSTATE	説明
-144	58003	セクション番号 &1 は無効である。現在の上位セクション番号は &3。理由は &2。
-145	55005	異機種のアプリケーション・サーバー用の反復はサポートされていない。
-175	58028	コミット操作が失敗した。
-189	22522	コード化文字セット識別コード &1 は無効。
-191	22504	混合データ値は無効である。
-250	42718	ローカル・リレーショナル・データベースがディレクトリーに定義されていない。
-251	2E000 42602	リレーショナル・データベース名 &1 の中の文字が有効ではない。
-300	22024	NUL 終了入力ホスト変数またはパラメーターには NUL が含まれていなかった。
-302	22001	入力ホスト変数 &2 での変換エラー。
-330	22021	文字変換が実行できない。
-331	22021	文字変換が実行できない。
-332	57017	CCSID &1 と CCSID &2 の間の文字変換は正しくない。
-334	22524	文字変換の結果が切り捨てに終わった。
-351 -352	56084	選択リストまたは入力リスト内にサポートされない SQLTYPE が見つかった。
-426	2D528	アプリケーション実行環境に無効な操作。この SQLSTATE は、EXCSQLIMM または EXCSQLSTT を使用して、動的 COMMIT 制限環境で COMMIT を実行する試みを報告する。
-427	2D529	アプリケーション実行環境に無効な操作。
-501 -502 -507	24501	無効なカーソル状態のため、実行が失敗した。指定されたカーソルがオープンされない。
-510	42828	この SQLSTATE は、ブロック化プロトコルを使用して行を取り出しているカーソル上での DELETE WHERE CURRENT OF CURSOR または UPDATE WHERE CURRENT OF CURSOR への試みを報告する。
-525	51015	ステートメントがエラーである。
-551	42501	タイプ *&3 の &2 の中のオブジェクト &1 に対して許可されていない。
-552	42502	&1 に対して許可されていない。
-683	42842	指定されたタイプの場合は、FOR DATA 文節または CCSID 文節が有効ではない。
-752	0A001	アプリケーション・プロセスは接続可能状態にない。理由コード &1。
-802	22003 22012	数値が範囲外であり、ゼロによる除算は無効である。
-805	51002	&2 内に SQL パッケージ &1 が見つからない。
-818	51003	整合性トークンが一致しない。
-842	08002	接続はすでに存在する。

表 32. SQLCODE および SQLSTATE (続き)

SQLCODE	SQLSTATE	説明
-862	55029	ローカル・プログラムが、リモート・リレーショナル・データベースへの接続を試みた。
-871	54019	指定された CCSID 値が多過ぎる。
-900	08003	接続が存在しない。
-918	51021	アプリケーション・プロセスがロールバック操作を実行するまで、SQL ステートメントを実行できない。
-922	42505	この SQLSTATE は、アプリケーション・サーバーへの接続処理中にユーザーを認証する際の障害を報告する。
-925 -926	2D521	現行環境では、SQL COMMIT または ROLLBACK ステートメントは無効である。
-950	42705	リレーショナル・データベース &1 はリレーショナル・ディレクトリー内にない。
-969	58033	アプリケーション・リクエスター・ドライバー・プログラムに要求を渡すときにエラーが起きた。
-7017	42971	コミットメント制御が受動側 DDM に対してすでに活動状態である。
-7018	42970	COMMIT HOLD または ROLLBACK HOLD が使用できない。
-7021	57043	アプリケーション・サーバー上でローカル・プログラムを実行しようとした。
-30000	58008	Distributed Relational Database Architecture (DRDA) プロトコルのエラー。
-30001	57042	分散 SQL プログラムの呼び出しは許可されていない。
-30020	58009	Distributed Relational Database Architecture (DRDA) プロトコルのエラー。
-30021	58010	リモート・システムでは分散リレーショナル・データベースはサポートされていない。
-30040	57012	リレーショナル・データベース &1 の DDM リソース &2 はアクセス不能になっている。
-30041	57013	リレーショナル・データベース &1 の DDM リソースはアクセス不能になっている。
-30050	58011	バインド・プロセスの進行中は DDM コマンド &1 は無効である。
-30051	58012	指定されたパッケージ名と一貫性トークンを使用するバインド・プロセスは活動中ではない。
-30052	42932	プログラム準備の前提事項が正しくない。
-30053	42506	所有者 &1 用のパッケージの作成を許可されていない。
-30060	08004	ユーザーはリレーショナル・データベース &1 に対する許可を受けていない。
-30061	08004	リレーショナル・データベース &1 が見つからない。
-30070	58014	分散データ管理機能 (DDM) コマンド &1 はサポートされていない。
-30071	58015	分散データ管理機能 (DDM) オブジェクト &1 はサポートされていない。

表 32. SQLCODE および SQLSTATE (続き)

SQLCODE	SQLSTATE	説明
-30072	58016	分散データ管理機能 (DDM) パラメーター &1 はサポートされていない。
-30073	58017	分散データ管理機能 (DDM) パラメーター値 &1 はサポートされていない。
-30074	58018	分散データ管理機能 (DDM) 応答メッセージ &1 はサポートされていない。
-30080	08001	分散データベース処理中に通信エラーが起きた。
-30082	08001	分散データベースの接続の試行の際の許可の障害。
-30090	25000 2D528 2D529	読み取り専用アプリケーション・サーバーの場合は、変更要求は有効ではない。
-30104	56095	バインド・オプションが無効である。この SQLSTATE は、1 つ以上のバインド・オプションがシステムで無効だったことを報告する。バインド操作は終了する。エラーのある最初のバインド・オプションは SQLERRMC 内に報告される。
-30105	56096	バインド・オプションが矛盾している。バインド・オプションは終了する。矛盾のあるバインド・オプションは SQLERRMC 内に報告される。
Unrecognized by AR	58020	エラーまたは警告用の SQLSTATE 値が定義されていない。

関連概念:

SQL メッセージおよびコード

関連資料:

メッセージ記述表示 (DSPMSGD) コマンド

第 1 障害データ検知データの検索

このトピックのヒントを使用して、IBM i オペレーティング・システムでの第 1 障害データ検知 (FFDC) データを位置指定できます。この情報が最も役に立つのは、サーバーで FFDC データの出力の原因になる障害が発生した場合です。通常、クライアントの FFDC データは、アプリケーション・プログラムを実行しているジョブに関連したスプール・ファイル内で見つけることができます。

注: FFDC データは、QSFWERRLOG システム値が *LOG に設定されていない限り生成されません。

1. メッセージの表示 (DSPMSG) QSYSOPR コマンドを実行してから、QSYSOPR メッセージ・ログ内で Software problem detected in Qccxyyyy というメッセージを探します。(プログラム名中の cc は通常は RW ですが、CN または SQ である場合もあります。)このメッセージがあると、FFDC データが作成されたことを示します。ヘルプ・キーを使って、メッセージに関する詳細を入手することができます。メッセージ・ヘルプに問題 ID が示されるので、それを使って、問題処理 (WRKPRB) コマンドで提示されたリストの中で問題を確認することができます。このステップはスキップすることができます。問題記録 (ある場合) は、リストの最上部またはその近辺にあるかもしれないからです。
2. 問題処理 (WRKPRB) コマンドを入力し、Software problem detected in Qccxyyyy メッセージで見つけたプログラム名 (Qccxyyyy) を指定します。そのプログラム名を使って、不要なリスト項目をフィルターで取り除きます。問題リストが提示されたら、問題 ID の入った行にオプション 5 を指定して、徴候ストリングやエラー・ログ ID などの問題の詳細を入手します。
3. エラー・ログ ID がある場合、システム保守ツール開始 (STRSST) コマンドを入力します。最初の画面で「サービス・ツールの開始」を選択します。次の画面で 1 を入力して、「エラー・ログ・ユーテ

イリティー」を選択します。その次の画面で 2 を入力し、「エラー・ログ ID による表示または印刷」を選択します。さらに次の画面で、次のいずれかを行うことができます。

- そのエラー・ログ ID を入力する。
- Y を入力して 16 進表示を行う。
- 印刷または表示のオプションを選択する。

表示オプションでは、行あたり 32 バイトではなく、16 バイトが用意されています。これは、80 文字のワークステーション・プリンターで画面をオンライン表示したり印刷したりする場合に便利です。表示オプションを選択した場合、実行キーを押した後、16 進データを見るには F6 キーを押します。

16 進データには、FFDC ダンプ・データの最初の 1 KB が入っていますが、その前に他の何らかのデータが入っています。FFDC データの先頭は、FFDC データ索引で識別されます。ターゲット・ジョブの名前（サーバーの場合）は、データ索引の前に置かれます。FFDC ダンプ・スプール・ファイルがまだ削除されていないときに、そのスプール・ファイルを検索するには、この完全修飾ジョブ名を使います。スプール・ファイルがなくなった場合、次のいずれかを行います。

- エラー・ログに保管されている最初の 1 KB のダンプを使用します。
- 1 KB の FFDC データでは足りない場合はその問題を再現します。

エラー・ログから FFDC データを解釈するとき、エラー・ログ内の FFDC データは、スプール・ファイル内のデータと同様、読み取り用にフォーマットされていません。エラー・ログ内の FFDC ダンプの各セクションの前には、4 バイトのヘッダーが付けられます。このヘッダーの最初の 2 バイトは、後続のセクションの長さ（接頭部は含まれません）です。その次の 2 バイトは、セクション番号を示しますが、索引内のセクション番号に対応します。

関連資料:

メッセージ表示 (DSPMSG) コマンド

問題処理 (WRKPRB) コマンド

システム保守ツール開始 (STRSST) コマンド

障害の報告のためのデータの入手

IBM i ライセンス・プログラムは、分散リレーショナル・データベースの問題を診断するときに印刷してヘルプできるようなデータを提供します。

また、システム・オペレーター・メッセージとアプリケーション・プログラム（データも一緒に）を使って、問題を診断することもできます。

ジョブ・ログの印刷

すべての IBM i ジョブは、それぞれジョブについて入力された要求に関連する情報が入るジョブ・ログを持っています。クライアントで問題が起きたら、その問題を診断するのにジョブ・ログ内の情報が役立つことがあります。

その情報を入手するには、次のようなコマンドを使ってそのユーザーをサインオフさせるのが 1 つの簡単な方法です。

```
SIGNOFF *LIST
```

このコマンドは、そのユーザーのジョブ・ログを印刷するか、または、印刷用の出力待ち行列にそのログを入れます。

ジョブ・ログを印刷する別の方法としては、アプリケーション・ジョブ記述に LOG(4 00 *SECLVL) を指定します。ジョブが終了すると、すべてのメッセージは、そのジョブ用のジョブ・ログに記録されます。出力待ち行列上でジョブ・ログを見つけ出してから、印刷手順を実行すれば、ジョブ・ログを印刷することができます。

サーバーのジョブ・ログも、問題の診断に役立つことがあります。

関連タスク:

271 ページの『分散リレーショナル・データベースのジョブ・ログによる要求情報の追跡』
すべての IBM i ジョブは、それぞれジョブについて入力された要求に関連する情報が入るジョブ・ログを持っています。

273 ページの『分散リレーショナル・データベースのジョブの探索』
クライアント上で分散リレーショナル・データベース・ジョブについての情報を探索する時に、使用されているユーザー・プロファイルが分かっている場合には、ユーザー・ジョブの処理 (WRKUSRJOB) コマンドを使用することによって、そのジョブを見つけることができます。

TCP/IP サーバー事前開始ジョブからのジョブ・ログの検出

分散データ管理 (DDM) TCP/IP サーバーに関連した QRWTSRVR 事前開始ジョブの 1 つを利用していた接続が終了すると、その事前開始ジョブはリサイクルされて、別の接続で使われます。その場合、終了した接続に関連したジョブ・ログは消去されます。

ただし、ある特定の場合、終了した接続に関連したジョブ・ログを消去する前に、ジョブ・ログはプリンター・ファイルにスプールされます。クライアント・ユーザー ID およびパスワードが正常に妥当性検査されなかった場合、ジョブ・ログはスプール・ファイルに出力されません。妥当性検査が正常に行われた場合、以下は、ジョブ・ログがスプール・ファイルに出力される条件です。

- オペレーティング・システムが V5R1 以上で、サーバー・ジョブのメッセージ・ロギング・テキスト・レベルが *SECLVL または *MSG の場合
- 要求の処理中に、接続を終了させるような重大なエラーが発生したことをサーバー要求ハンドラーの経路指定プログラムが検出した場合。
- 事前開始ジョブがサービスされていた場合 (サービス・ジョブの開始 (STRSRVJOB) コマンドの使用による)。
- クライアントまたはサーバー上の QRWOPTIONS データ域が、サーバー・ジョブによって満たされたジョブ・ログ出力条件を指定した場合。

いくつかの理由で、ジョブ・ログ情報を入手したいと思われることでしょう。ジョブ・ログ情報は、エラーを診断するために明らかに役立ちます。パフォーマンス問題を分析するためにも役立ちます。たとえば、デバッグ下で実行時に生成される SQL 最適化プログラム・データを入手したい場合、手動でサービス・ジョブを開始してデバッグの開始 (STRDBG) コマンドを実行するか、またはジョブ・ログを保存するように QRWOPTIONS データ域内に 1 つ以上の適切なオプションを設定することができます。

接続フェーズ中に障害が起こるジョブのログは、上記で説明されているプロセスによって保管されません。そのプロセスによって保管されるジョブは、事前開始ジョブ ID の下には保管されません。ジョブ・ログを見つけ出すには、次のようなコマンドを実行します。

```
WRKJOB userid/QPRTJOB
```

ここで、*userid* は、サーバーへの接続で使われるユーザー ID です。このユーザー ID が不明の場合、サーバー上でログの表示 (DSPLOG) コマンドを使って見つけることができます。

以下のようにパラメーターを使用することによって、不要なメッセージをフィルターで取り除くことができます。

```
DSPLOG PERIOD(('11:00')) MSGID(CPI3E34)
```

次のようなメッセージを探してください。ただし、このメッセージ (CPI3E34) を抑制するために QRWOPTIONS データ域が使用されている場合、このメッセージはヒストリー・ログ内には表示されないことに注意してください。

```
DDM job xxxx servicing user yyy on ddd at ttt.
```

関連概念:

424 ページの『QRWOPTIONS データ域』

DDM または DRDA TCP/IP サーバー・ジョブは、開始されると、ユーザーが診断や他のオプションを指定できるデータ域を探します。このデータ域の名前は QRWOPTIONS で、このデータ域は QGPL ライブラリーに常駐している場合にのみ有効になります。このデータ域は、48 文字のストリングから成っています。

関連資料:

サービス・ジョブの開始 (STRSRVJOB) コマンド

ログ表示 (DSPLOG) コマンド

デバッグの開始 (STRDBG) コマンド

製品活動記録の印刷

IBM i 製品活動記録ログは、マシン・チェック、装置エラー、およびテープの統計の記録です。またこれには、各第 1 障害データ検知 (FFDC) ダンプの最初の 1000 バイトを含めた FFDC 情報も入っています。これらのエラーを検討すれば、問題の特性を判別できる可能性があります。

サインオンしているシステムの製品活動記録を印刷するには、次のステップを実行します。

1. 任意のコマンド行に**エラー・ログの印刷 (PRERRLOG)** コマンドを入力してから、F4 キー (プロンプト) を押します。「エラー・ログの印刷」画面が表示されます。
2. 印刷したいログ情報の種類のパラメーター値を入力してから、実行キーを押します。そのログ情報は、ジョブで指定されている出力待ち行列に送られます。
3. **ジョブの処理 (WRKJOB)** コマンドを入力します。「ジョブの処理」画面が表示されます。
4. スプール・ファイルを処理するためのオプションを選択します。「ジョブ・スプール・ファイルの処理」画面が表示されます。
5. スプール・ファイル・リストの下端またはその近くで、作成したばかりのログ・ファイルを探します。
6. ログ・ファイルの次の「Opt」列に、印刷状況の処理オプションを入力します。「印刷状況の処理」画面が表示されます。
7. 「印刷状況の処理」画面で、状況変更オプションを使用して、ファイルの状況を変更し、印刷装置をファイルの印刷に指定してください。

関連資料:

エラー・ログ印刷 (PRERRLOG) コマンド

ジョブ処理 (WRKJOB) コマンド

ジョブ・トレース

場合によっては、問題を追跡していてもプログラムを特定できないことがあります。このようなケースでは、トレース開始 (STRTRC) およびジョブ・トレース (TRCJOB) コマンドを使用して、モジュール・フロー、IBM i のデータ収集、および実行された CL コマンドをトレースすることができます。

これらのツールは、問題分析手順で問題について十分な情報が得られない場合に使用してください。分散データベース・アプリケーションの場合、これらのコマンドは、分散データベースの要求と応答のデータ・ストリームを収集するのに便利です。

サーバー・ジョブのジョブ・トレースを取得する必要がある場合、サーバーでサービス・ジョブを始動する必要があります。

関連概念:

421 ページの『アプリケーション・サーバー問題の診断のためのサービス・ジョブの開始』
アプリケーションが分散リレーショナル・データベース・アーキテクチャ (DRDA) を使用すると、SQL ステートメントは、アプリケーション・サーバー・ジョブ内で実行されます。そのため、IBM i オペレーティング・システム上で実行されるアプリケーション・サーバー・ジョブのデバッグまたはジョブ・トレースを開始することが必要になる場合があります。これを行うための技法は、拡張プログラム間通信 (APPC) または TCP/IP のどちらを使用しているかによって異なります。

415 ページの『通信トレース』

Distributed Relational Database Architecture (DRDA) を使って分散リレーショナル・データベースにアクセスしようとして、CPF3Exx の範囲または CPF91xx の範囲のメッセージが出されたら、通信トレースを実行する必要があります。

ジョブ・トレース:

ジョブ・トレース (TRCJOB) コマンドは、2 つの追跡ツールのうちの古い方のツールです。各トレース・レコードは、生成されるたびに内部トレース記憶域に保管されます。トレースが終了したら、トレース・レコードをスプール印刷装置ファイル (QPSRVTRC) に書き込むことができますが、データベース出力ファイルに送ることもできます。

以下に、トレースのシナリオの例を示します。

```
TRCJOB SET(*ON) TRCTYPE(*ALL) MAXSTG(2000)
          TRCFULL(*WRAP) EXITPGM($SCFTRC)
CALL QCMD
TRCJOB SET(*OFF) OUTPUT(*PRINT)
WRKOUTQ output-queue-name
```

QPSRVTRC という名前のスプール・ファイルが示されます。このスプール・ファイルには、行ったトレースが入っています。

関連資料:

ジョブ・トレース (TRCJOB) コマンド

トレース開始:

トレースの実行には、トレース開始 (STRTRC) コマンドを使用することも可能です。STRTRC コマンドはトレース・ジョブ (TRCJOB) コマンドより柔軟性が高く、煩わしいものではありません。このツールでは、複数のジョブにまたがって追跡を行い、モジュール・フローに関してより綿密に詳細を表示することができます。

各トレース・レコードは、生成されるたび、セッション ID によって識別された内部トレース記憶域に保管されます。トレース終了 (ENDTRC) を使用してトレースが終了すると、トレース・レコードは、一連のデータベース・ファイルとしてユーザー指定のライブラリーに置かれます。こうしてライブラリーに置かれたファイルは、スプール印刷装置ファイル (QPSRVTRCJ) に書き込むこともできますし、PRTRC を発行してデータベース出力ファイルに送ることもできます。

以下に、トレースのシナリオの例を示します。


```
STRTRC SSNID(DRDATRACE) JOB((*ALL/QUSER/QRWTSRVR)) MAXSTG(160000)
TRCFULL(*STOPTRC)
```

障害のある DRDA のシナリオを実行します。

```
ENDTRC SSNID(DRDATRACE) DTALIB(TRACELIB)
PRTRTC DTAMBR(DRDATRACE) DTALIB(TRACELIB)
```

関連資料:

トレース終了 (ENDTRC) コマンド

トレース開始 (STRTRC) コマンド

通信トレース

Distributed Relational Database Architecture (DRDA) を使って分散リレーショナル・データベースにアクセスしようとして、CPF3E xx の範囲または CPF91 xx の範囲のメッセージが出されたら、通信トレースを実行する必要があります。

以下に、表示される可能性のあるこれらの範囲の共通メッセージを示します。

表 33. 通信トレース・メッセージ

MSG 識別コード	説明
CPF3E80	DDM データ・ストリームの構文エラー。
CPF91 xx	DDM プロトコル・エラー。
CPF3E83	FD0:CA 記述子が無効。
CPF3E84	データのミスマッチ・エラー。

実行できる通信トレースは 2 通りあります。1 つは、標準の通信トレースです。もう 1 つは TRCTCPAPP 機能です。IPSec またはセキュア・ソケット・プロトコルにより暗号化されたデータ・ストリームがある場合、TRCTCPAPP 関数は判別可能なトレースを提供します。これは、暗号化前と暗号化解除後のデータを取り込みます。ただし、暗号化されていないデータ・ストリームのトレース取得の場合にも十分に働きます。これは、LOOPBACK が使用されるシステム内の DRDA フローのトレースを行うために必要です。

関連概念:

103 ページの『接続セキュリティー・プロトコル』

TCP/IP を使用した分散データ管理 (DDM) または Distributed Relational Database Architecture (DRDA) の現在の DB2 for i 実装によって、以下のいくつかの接続セキュリティー・プロトコルがサポートされています。

413 ページの『ジョブ・トレース』

場合によっては、問題を追跡していてもプログラムを特定できないことがあります。このようなケースでは、トレース開始 (STRTRC) およびジョブ・トレース (TRCJOB) コマンドを使用して、モジュール・フロー、IBM i のデータ収集、および実行された CL コマンドをトレースすることができます。

関連タスク:

418 ページの『TCP/IP 通信トレース』

トレース・ツールの 1 つの使い方は、データが暗号化されている環境で伝送の平文を表示するというものです。

標準通信トレース:

通信トレース機能を使って、通信構成オブジェクトでのデータのトレースを開始したり停止したりすることができます。データのトレースの実行を完了した後、印刷または表示用にそのデータをフォーマットすることができます。印刷装置ファイルは、出力待ち行列の中でしか見ることはできません。

通信トレース・オプションは、システム保守ツール (SST) のもとで実行します。SST を使うと、通信トレースの活動中に構成オブジェクトを使用することができます。分散データベース・ネットワークで使えるどの通信タイプのデータでも、トレースしてフォーマットすることができます。

システムに接続されている任意の画面から、通信トレースを実行することができます。*SERVICE の特殊権限 (SPCAUT) をもつユーザーはだれでも、IBM i オペレーティング・システム上でトレースを実行することができます。通信トレースは、すべての回線速度をサポートします。

通信トレースは、次のような状況下で使用してください。

- 問題分析手順では、問題について十分な情報が得られない場合。
- プロトコル違反の問題があると疑われる場合。
- 回線ノイズが問題であると疑われる場合。
- システム・ネットワーク体系 (SNA) バインド問題があることをエラー・メッセージが示している場合。

通信トレースで生成されるデータを正しく解釈するには、使用する回線プロトコルの詳しい知識が必要です。

可能な限り、回線をオンに構成変更する前に通信トレースを開始してください。そうすれば、オンに構成変更されたときの回線の最も正確なサンプルが得られます。

APPC トレースを実行してその出力を処理するには、どの回線、制御装置、および装置で実行しているかが分かっているなければなりません。

TCP/IP トレースの出力で不要なデータをフォーマットし、回避するには、ソースおよびサーバーの IP アドレスを指定することができます。その代わりに、単にポート番号を指定するだけで十分な場合もあり、より容易です。

次のコマンドを使って、通信トレースを開始、終了、印刷、削除、およびダンプすることができます。

通信トレース開始 (STRCMNTRC) コマンド

指定された回線またはネットワーク・インターフェース記述の通信トレースを開始します。「トレースするバイト数 (Number of bytes to trace)」パラメーターの「先頭バイト (Beginning bytes)」の値に *MAX を指定します。通信トレースは、通信トレースの終了 (ENDCMNTRC) コマンドが実行されるまで続きます。

例えば、以下のコマンドを入力して、イーサネット回線に関する通信トレースを開始できます。

```
STRCMNTRC CFGOBJ(ETHLINE) CFGTYPE(*LIN) MAXSTG(20000) USRDTA(*MAX)
```

使用しているイーサネット回線を判別するには、以下のコマンドを入力します。

```
WRKCFGSTS *LIN
```

通信トレース終了 (ENDCMNTRC) コマンド

指定された回線またはネットワーク・インターフェース記述で実行している通信トレースを終了します。

イーサネット回線に関する通信トレースを終了するには、以下のコマンドを入力します。

```
ENDCMNTRC CFGOBJ(ETHLINE) CFGTYPE(*LIN)
```

通信トレース・ダンプ (DMPCMNTRC) コマンド

指定した回線、ネットワーク・インターフェース記述、またはネットワーク・サーバー記述の不定様式のトレース・データを、ユーザー指定のストリーム・ファイルにコピーします。

イーサネット回線に関する通信トレースを統合ファイル・システムのファイルにダンプできます。追加のヘッダー情報は統合ファイル・システムのファイルにダンプされます。

注: STRDST 通信トレースはダンプできないので、出力中にヘッダー情報は表示されません。

イーサネット回線に関する通信トレースをダンプするには、以下のコマンドを入力します。

```
DMPCMNTRC CFGOBJ(ETHLINE) CFGTYPE(*LIN) TOSTMF(COMMTRACE)
```

通信トレース印刷 (PRTCMNTRC) コマンド

指定した回線またはネットワーク・インターフェース記述の通信トレース・データを、スプール・ファイルまたは出力ファイルに移動します。SNA データのみのフォーマット・パラメーターには *YES を指定します。

サーバー上のポート 446 上の統合ファイル・システムのファイルからスプール・ファイルに通信トレースを印刷するには、以下のコマンドを入力します。

```
PRTCMNTRC FROMSTMF(COMMTRACE) SLTPORT(446) FMTBCD(*NO)
```

通信トレース削除 (DLTCMNTRC) コマンド

指定した回線またはネットワーク・インターフェース記述の通信トレースを削除します。

サーバー上のポート 446 上の統合ファイル・システムのファイルのフィルター処理から通信トレースを削除するには、以下のコマンドを入力します。

```
DLTCMNTRC CFGOBJ(ETHLINE) CFGTYPE(*LIN)
```

統合ファイル・システムのストリーム・ファイルを削除するには、以下のコマンドを入力します。

```
qsh  
rm commtrace
```

関連概念:



通信管理 (PDF)

関連資料:

通信トレース削除 (DLTCMNTRC) コマンド

通信トレース終了 (ENDCMNTRC) コマンド

通信トレース印刷 (PRTCMNTRC) コマンド

通信トレース開始 (STRCMNTRC) コマンド

回線、制御装置、および装置記述の検索:

サーバー・ジョブの始動に使われる制御装置と装置を見つけるには、構成状況処理 (WRKCFGSTS) コマンドを使います。

以下にその例を示します。

```
WRKCFGSTS CFGTYPE(*DEV)  
          CFGD(*LOC)  
          RMTLOCNAME(DB2ESYS)
```

RMTLOCNAME キーワードの値は、サーバー名です。

構成状況処理 (WRKCFGSTS) コマンドは、指定されたシステム名をリモート・ロケーション名としてもつすべての装置を表示します。一度に 1 つの装置しかオンに構成変更できないため、どの装置を使用中であるかが分かります。オプション 8 を使って装置記述を処理してから、オプション 5 でそれを表示します。接続された制御装置のフィールドに、制御装置の名前が示されます。(WRKCFGSTS) コマンドを使って制御装置と装置の記述を処理することができます。以下にその例を示します。

```
WRKCFGSTS CFGTYPE(*CTL)
          CFGD(PCXZZ1205) /* workstation */
WRKCFGSTS CFGTYPE(*CTL)
          CFGD(LANSLKM) /* IBM i on token ring */
```

CFGD 値は装置記述名ですが、これは、このトピックの最初の例で装置記述から得たものです。

構成状況処理 (WRKCFGSTS) コマンドの出力には、通信トレースを処理するときに必要な回線記述の名前も入っています。オプション 8 を選んでからオプション 5 を選んで制御装置記述を表示した場合、活動中の交換回線パラメーターが回線記述の名前を表示します。LAN リモート・アダプター・アドレスに、リモート・システムのトークンリング・アドレスが示されます。

回線名を見つけ出す別の方法では、回線記述の処理 (WRKLIND) コマンドを使って、システムのすべての回線記述を表示します。

関連資料:

構成状況処理 (WRKCFGSTS) コマンド

回線記述の処理 (WRKLIND) コマンド

TCP/IP 通信トレース

トレース・ツールの 1 つの使い方は、データが暗号化されている環境で伝送の平文を表示するというものです。

トレース・データは、暗号化前に送信側でキャプチャーし、暗号化後に受信側でキャプチャーします。ただし、このトレース・ツールは、他の環境でも有用です。この機能は、通信に TCP/IP を使用する場合だけ使用できます。

TCP/IP アプリケーションのトレース (TRCTCPAPP) コマンドを使うには、*SERVICE 特殊権限を受けたユーザー・プロファイルがなければなりません。このトレースを開始するには、次の行を入力します。

```
TRCTCPAPP *DDM
```

トレースを特定のポートにだけ限定したい場合は (例えば、SSL 用のポート 448)、次の例のようにします。

```
TRCTCPAPP *DDM *ON RMTNETADR(*INET *N '255.255.255.255' 448)
```

追跡している通信が終了したら、次のようなコマンドを実行してから、処理結果のスパール・ファイル調べます。

```
TRCTCPAPP *DDM *OFF
```

複数の接続をトレースする場合は、各 QRWTSRVR ジョブにスパール・ファイルを配置し、突き合わせる必要があります。スパール・ファイル名は QZBSTRC (OS/400 V5R2 またはそれ以前) または QCNTRC (IBM i V5R3 以降) で、ジョブは QRWxxxxxx です。この xxxxxx の部分には、スパール・ファイルのユーザー・データにあるジョブ番号が入ります。

*DDM アプリケーションでの使用時の制約事項

*DDM アプリケーションで **TCP/IP アプリケーションのトレース (TRCTCPAPP)** コマンドを使用する場合、単一の送受信メッセージごとにトレースできるデータの最大量は、6000 バイトに限定されます。

関連概念:

415 ページの『通信トレース』

Distributed Relational Database Architecture (DRDA) を使って分散リレーショナル・データベースにアクセスしようとして、CPF3Exx の範囲または CPF91xx の範囲のメッセージが出されたら、通信トレースを実行する必要があります。

関連資料:

TCP/IP アプリケーション・トレース (TRCTCPAPP) コマンド

TCP/IP 通信トレースのフォーマット設定:

TCP/IP アプリケーションのトレース (TRCTCPAPP) コマンドは、DRDA および DDM フローを分析して、読みやすい論理表記にするために使用できます。このコマンドは、情報を ASCII 形式で表示することもできるため、異種環境でも役立ちます。

この形式を要求する場合は、通信トレースの末尾に続けて以下を入力します。

```
TRCTCPAPP APP(*DDM) SET(*OFF) ARGLIST('lv1=2')
```

以下は、フォーマット設定されていないトレースの例です。ただし、この例は、このトピックの幅に合わせて編集してあります。

```
0080D0010001007A 200100162110D9C3 C8C1E2D5E3E24040
4040404040404040 *..}.....RCHASNTS *
0006210F2407000D 002FD8E3C4E2D8D3 F4F0F0000C112ED8
E2D8F0F5F0F3F000 *.....QTDSQL400....QSQ05030.*
0A00350006119C00 2500062121241E00 062120241E0010D1
2A01000000000000 *.....J.....*
0000000000001621 35C1D7D7D54BD3D7 F0F6F6C1C2B9191C
F706F90005213BF1 *.....APPN.LP066AB...7.9....1*
```

以下は、同じトレースを TRCTCPAPP でフォーマット設定したものです。

```
-Datastream-----
DATA:                (ASCII)                (EBCDIC)
0080D0010001007A    200100162110D9C3    .8'.....a...b.êã .0}.....RC
C8C1E2D5E3E24040    4040404040404040    ç ë+ëë..... HASNTS
0006210F2407000D    002FD8E3C4E2D8D3    .Lb.f"....êèàëé< .....QTDSQL
F4F0F0000C112ED8    E2D8F0F5F0F3F000    .....éëé..... 400....QSQ05030.
0A00350006119C00    2500062121241E00    C...L.ô...LbbfK. ....ä.....
062120241E0010D1    2A01000000000000    LbafK..çk..... .....J.....
0000000000001621    35C1D7D7D54BD3D7    .....b. &&+.<& .....APPN.LP
F0F6F6C1C2B9191C    F706F90005213BF1    ... â` .c.L...bB. 066AB¾..7.9....1
-Parsed-----
```

RECV(AS) RQSDSS - Request Data Stream Structure

LL: 128 CORR: 0001 CHAINED: n CONT ON ERR: n SAME CORR FOR NEXT DSS: n

NM: ACCRDB - Access RDB

LL: 122 CP: 2001

NM: RDBNAM - Relational Database Name

LL: 22 CP: 2110

ASCII: êãç ë+ëë.....

EBCDIC: RCHASNTS

NM: RDBACCCL - RDB Access Manager Class
 LL: 6 CP: 210F
 CODE POINT DATA: 2407
 NAME: SQLAM - SQL Application Manager
 NM: TYPDEFNAM - Data Type Definition Name
 LL: 13 CP: 002F
 ASCII: éèâëé<...
 EBCDIC: QTDSQL400
 NM: PRDID - Product-Specific Identifier
 LL: 12 CP: 112E
 DATA: (ASCII) (EBCDIC)
 D8E2D8F0F5F0F3F0 éëé..... QSQ05030
 NM: TYPDEFOVR - TYPDEF Overrides
 LL: 10 CP: 0035
 NM: CCSIDSBC - CCSID for Single-Byte Characters
 LL: 6 CP: 119C
 DATA: (ASCII) (EBCDIC)
 0025
 NM: STTDECDEL - Statement Decimal Delimiter
 LL: 6 CP: 2121
 CODE POINT DATA: 241E
 NAME: DFTPKG - Package Default
 NM: STTSTRDEL - Statement String Delimiter
 LL: 6 CP: 2120
 CODE POINT DATA: 241E
 NAME: DFTPKG - Package Default
 NM: SXXPRDDTA - Extended Product Data
 LL: 16 CP: D12A
 DATA: (EBCDIC)
 0100000000000000 00000000

C1D7D7D54BD3D7F0 F6F6C1C2B9191CF7 &&+.<&... ä".c. APPN.LP066AB%.7

06F9 L. .9

NM: TRGDFTRT - Target Default Value Return

LL: 5 CP: 213B

BOOLEAN: TRUE

関連資料:

TCP/IP アプリケーション・トレース (TRCTCPAPP) コマンド

アプリケーション・サーバー問題の診断のためのサービス・ジョブの開始

アプリケーションが分散リレーショナル・データベース・アーキテクチャー (DRDA) を使用すると、SQL ステートメントは、アプリケーション・サーバー・ジョブ内で実行されます。そのため、IBM i オペレーティング・システム上で実行されるアプリケーション・サーバー・ジョブのデバッグまたはジョブ・トレースを開始することが必要になる場合があります。これを行うための技法は、拡張プログラム間通信 (APPC) または TCP/IP のどちらを使用しているかによって異なります。

関連概念:

413 ページの『ジョブ・トレース』

場合によっては、問題を追跡していてもプログラムを特定できないことがあります。このようなケースでは、トレース開始 (STRTRC) およびジョブ・トレース (TRCJOB) コマンドを使用して、モジュール・フロー、IBM i のデータ収集、および実行された CL コマンドをトレースすることができます。

APPC サーバーのサービス・ジョブ

DB2 for i アプリケーション・サーバーは、特別トランザクション・プログラム名 (TPN) を認識すると、アプリケーション・サーバーからシステム・オペレーターにメッセージを送信させてから、応答待ちになります。

それによって、**サービス・ジョブの開始 (STRSRVJOB)** コマンドを出して、アプリケーション・サーバー・ジョブのジョブ・トレースまたはデバッグを開始することができます。

DB2 for i アプリケーション・サーバー・ジョブを停止してから、デバッグ・モードで再始動するには、以下のステップを実行します。

1. アプリケーション・リクエスターで QCNTRVC を TPN として指定します。これを行う方法は、プラットフォームごとに異なります。このトピック・コレクションでは、いくつかの方法を説明します。
IBM i アプリケーションは、QCNTRVC の TPN を受け取ると、QSYSOPR にメッセージ CPF9188 を送り、G (go) の応答を待ちます。
2. G 応答を入力する前に、**サービス・ジョブの開始 (STRSRVJOB)** コマンドを使用して、アプリケーション・サーバー・ジョブのサービス・ジョブを開始し、そのジョブをデバッグ・モードにします。(ジョブ名を表示するには、CPF9188 メッセージでヘルプを要求します。)
3. **デバッグの開始 (STRDBG)** コマンドを入力します。
4. アプリケーション・サーバー・ジョブのデバッグを開始したら、QSYSOPR メッセージに G を応答します。アプリケーション・サーバーは、G 応答を受信すると、通常の DRDA 処理を続けます。
5. アプリケーションの実行後、アプリケーション・サーバーのジョブ・ログを表示して、SQL デバッグ・メッセージを調べます。

関連概念:

『独自のトランザクション・プログラム名の作成と QCNTSRVC の設定』

独自のトランザクション・プログラム名 (TPN) を作成し QCNTSRVC を設定するには、以下のタスクを実行します。

関連資料:

サービス・ジョブの開始 (STRSRVJOB) コマンド

デバッグの開始 (STRDBG) コマンド

独自のトランザクション・プログラム名の作成と QCNTSRVC の設定

独自のトランザクション・プログラム名 (TPN) を作成し QCNTSRVC を設定するには、以下のタスクを実行します。

関連タスク:

421 ページの『APPC サーバーのサービス・ジョブ』

DB2 for i アプリケーション・サーバーは、特別トランザクション・プログラム名 (TPN) を認識すると、アプリケーション・サーバーからシステム・オペレーターにメッセージを送信させてから、応答待ちになります。

DB2 for i アプリケーション・リクエスター上でトランザクション・プログラム名として QCNTSRVC を設定する:

リレーショナル・データベース・ディレクトリー項目の追加 (ADDRDBDIRE) コマンドまたは リレーショナル・データベース・ディレクトリー項目の変更 (CHGRDBDIRE) コマンドの TNSPGM パラメーターに QCNTSRVC を指定します。

RDB ディレクトリー項目のテキスト内の特殊トランザクション・プログラム名 (TPN) を書き留めておけば、デバッグの終了後にその TPN を元に戻して変更するための覚え書きとして役に立ちます。

注: トランザクション・プログラム名は、常に SNA リレーショナル・データベースのディレクトリー項目と連動します。

関連資料:

リレーショナル・データベース・ディレクトリー項目の追加 (ADDRDBDIRE) コマンド

リレーショナル・データベース・ディレクトリー項目の変更 (CHGRDBDIRE) コマンド

DB2 for i アプリケーション・サーバー・ジョブのデバッグに使用する独自のトランザクション・プログラム名の作成:

末尾にデバッグ・ステートメントと TFRCTL QSYS/QCNTEDDM ステートメントの入った制御言語プログラムをコンパイルすれば、独自のトランザクション・プログラム名 (TPN) を作成することができます。これを作成した場合の利点は、接続を行うときに手動介入なくて済むことにあります。

プログラム例については、次のコードを参照してください。

注: コードの例を使用することは、443 ページの『コードに関するライセンス情報および特記事項』の条件に同意したものとされます。

PGM

MONMSG CPF0000

STRDBG UPDPROD(*YES) PGM(CALL/QRWTEXEC) MAXTRC(9999)

ADDBKP STMT(CKUPDATE) PGMVAR((*CHAR (SQLDA@))) OUTFMT(*HEX) +
LEN(1400)


```
ADDTRC PGMVAR((DSLENGTH ( )) (LNTH ( )) (FDODTA_LNTH ( )))
TRCJOB *ON TRCTYPE(*DATA) MAXSTG(2048) TRCFULL(*STOPTRC)
TFRCTL QSYS/QCNTEDDM
ENDPGM
```

アプリケーション・リクエスター (AR) の RDB ディレクトリー項目内の TPN 名は、プログラマーが指定する名前です。特殊 TPN が使用中であったり、デバッグを完了した後で TPN 名を必ず元に戻すよう指示したりする警告を指定するには、テキスト・フィールドを使います。

注: トランザクション・プログラム名は、常に SNA RDB 項目と連動します。

RDB の TPN をいったん変更すると、再び元に戻さない限り、その AR からのすべての接続はその新しい TPN を使い続けます。その場合、パフォーマンスの低下やオペレーター応答の長時間待ちが生じたり、記憶域がデバッグ・データでいっぱいになったりして、事情を知らないユーザーの不意を突くこととなります。

DB2 for VM クライアント上でトランザクション・プログラム名として QCNTRVC を設定する:

UCOMDIR NAMES ファイルを変更して、TPN タグに QCNTRVC を指定します。

以下にその例を示します。

```
:nick.RCHASLAI :tpn.QCNTRVC
:lname.VM4GATE RCHASLAI
:modename.MODE645
:security.NONE
```

次に、SET COMDIR RELOAD USER を発行します。

DB2 for z/OS クライアント上でトランザクション・プログラム名として QCNTRVC を設定する:

SYSIBM.LOCATIONS テーブルを更新して、DB2 for i サーバーの RDB-NAME の入った行について、TPN 列に QCNTRVC を指定します。

DB2 for Linux, UNIX, and Windows クライアント上でトランザクション・プログラム名として QCNTRVC を設定する:

DB2 for Linux, UNIX, および Windows を操作し、このプロダクトのファミリーで TPN 設定の説明が必要な場合、Web ページがお役に立ちます。

Web ページ DB2 for Linux, UNIX, and Windows  を参照してください。そこで、異なるバージョンに特定のいくつかの資料を見つけることができます。

TCP/IP サーバーのサービス・ジョブ

DDM TCP/IP サーバーは、拡張プログラム間通信機能 (APPC) サーバーのように、トランザクション・プログラム名 (TPN) を使用することはありません。しかし、TCP/IP サーバーで事前開始ジョブを使えば、その環境でサービス・ジョブを開始する手段になります。

ただし、QRWOPTIONS データ域の使用法と関連した機能の紹介では、多くの場合、サービス・ジョブを開始する必要がないことに注意してください。その機能を使用すると、トレースを開始したり、その他の診断機能を開始することができます。ジョブの接続フェーズのトレースが必要な場合には、サービス・ジョブを開始する必要が生じることがあります。

以下のステートメントが真である場合、**ログの表示 (DSPLOG)** コマンドを使用して、指定された接続に使用されるサーバー・ジョブの名前を報告している CPI3E34 メッセージを見つけることができます。

- 接続操作中にサーバーのアクションをトレースする必要はありません。
- QRWOPTIONS 機能を使用しないことを選択します。
- 対話式 SQL からなど、サーバー上にいくつかのセットアップを行うことができるまで、アプリケーション・リクエスト (AR) ジョブの実行を遅らせる機能を持っています。

これで **サービス・ジョブの開始 (STRSRVJOB)** コマンドを使用できます。

接続ステートメントをトレースする必要があるか、または、接続後にサーバー上で手動セットアップを行う時間がない場合、接続が確立される前に、その接続でどの事前開始ジョブが使われるかを知っている必要があります。そのためには、可能であれば、テスト中に他のユーザーが接続できないようにしてから、1 つ以外の事前開始ジョブをすべて終了するのが 1 つの方法です。

事前開始ジョブの数を強制的に 1 つにするには、QSYSWRK システム (OS/400 V5R2 またはそれ以前) または QUSRWRK (OS/400 V5R2、IBM i V5R3 以降) 内で実行される QRWTSRVR ジョブ用の事前開始ジョブ項目変更 (CHGPJE) コマンドで、以下のパラメーターを、それぞれ示してある値に設定します。

- 最初のジョブ数: **1**
- 限界値: **1**
- 追加のジョブ数: **0**
- 最大ジョブ数: **1**

この技法を使用した場合、必ずパラメーターを後で変更して、各自の環境に適した値に戻すよう気を付けてください。そうしないと、1 つの事前開始ジョブが使用中のときにユーザーが接続を試みると、'A connection with a remote socket was reset by that socket' というメッセージが出されます。

関連概念:

『QRWOPTIONS データ域』

DDM または DRDA TCP/IP サーバー・ジョブは、開始されると、ユーザーが診断や他のオプションを指定できるデータ域を探します。このデータ域の名前は QRWOPTIONS で、このデータ域は QGPL ライブラリーに常駐している場合にのみ有効になります。このデータ域は、48 文字のストリングから成っています。

関連資料:

サービス・ジョブの開始 (STRSRVJOB) コマンド

事前開始ジョブ項目変更 (CHGPJE) コマンド

ログ表示 (DSPLOG) コマンド

QRWOPTIONS データ域

DDM または DRDA TCP/IP サーバー・ジョブは、開始されると、ユーザーが診断や他のオプションを指定できるデータ域を探します。このデータ域の名前は QRWOPTIONS で、このデータ域は QGPL ライブラリーに常駐している場合にのみ有効になります。このデータ域は、48 文字のストリングから成っています。

注: データ域内の情報は、CCSID 37 または 500 に大文字で入力されなければなりません。

データ域の形式は、以下のとおりです。

表 34. データ域の形式

列	内容
1-15	スイッチの値として I が指定されるとき (その他の場合には無視される) に使用される小数点付き 10 進数のクライアント IP アドレス。
16	サーバーによって無視される予約済み域 (人間が使用できる文字を含めることができる)
17-26	スイッチの値として U が指定されるとき (その他の場合には無視される) に比較されるユーザー・プロファイル名
27	A、I、または U に設定される場合にジョブ・ログが保持されるようにするスイッチ (注 1 および 2 を参照)
28	A、I、または U に設定される場合に DSPJOB 出力が印刷されるようにするためのスイッチ (注 1 および 2 を参照)
29	A、I、または U に設定される場合にジョブがトレースされるようにするスイッチ (注 1 および 2 を参照)。
30	A、I、または U に設定される場合にジョブにデバッグが開始されるようにするスイッチ (注 1 を参照)。
31	<p>QRYOPTLIB 値を A、I または U に設定すると、Query 属性の変更 (CHGQRYA) コマンドの呼び出しに切り替えられます。QRYOPTLIB 値は、正しい QAQQINI ファイルを含むライブラリーの名前を含んでいなければならない列 39-48 から抽出されます (注 1 を参照)。</p> <p>注: この列に I または A が指定されている場合、QUSER は、それが実施される *JOBCTL 特殊権限を持っていなければなりません。</p>
32	A、I、または U に設定されている場合、クライアント・デバッグ・オプションをシャドウ作成するためのスイッチ (注 1 を参照)。
33	<p>T に設定される場合、ジョブ・トレースのために新しい STRTRC の代わりに古い TRCJOB を使用するためのスイッチ。列 29 はトレースを要求する。</p> <p>注: この列が T に設定される場合、ジョブ・トレースのために TRCJOB が使用される。STRTRC を使用するには、この列にブランクまたは S に設定する。</p>
34	ヒストリー・ログ内の CPI3E34 メッセージを抑制するには、この列を N に設定する (これは PTF SI02613 とともにのみ OS/400 V5R1 で使用可能である)
35	A、I、または U に設定される場合に特殊なサブルーチン・トレースを開始するスイッチ (注 1 および 2 を参照)。
36-38	予約済み
39-48	汎用データ域 (列 31 内の適当な値によって 照会変更属性 (CHGQRYA) コマンドがトリガーされる場合にはライブラリー名を含む)

注:

- 以下は、表示される列に対応する機能を活動化するスイッチの値です。
 - A は、サーバー・ジョブをすべて使用するための機能を活動化します。
 - I は、列 1 から 15 で指定されたクライアント IP アドレスが、接続の試みで使用されるクライアント IP アドレスと一致する場合に機能を活動化します。
 - U は、列 17 から 26 で指定されたユーザー ID が、接続の試みで使用されるユーザー ID と一致する場合に機能を活動化します。
- この機能を活動化した結果生じるスプール・ファイルを見つけるには、**ジョブ処理コマンド (WRKJOB user-profile/QPRTJOB)** を使用します。ここで、user-profile は、接続要求で使用されるユーザー ID です。オプション 4 を選択します。1 つ以上のスプール・ファイルが表示されます。

表 35. WRKJOB user-profile/QPRTJOB コマンドからのファイル・リスト

ファイル	装置および待ち行列	ユーザー・データ
QPJOBLOG	QEZJOBLOG	QRWTSRVR
QPDSJOB	PRT01	
QPSRVTRC	PRT01	

3. 特殊な DRDA サブルーチン・トレースを含むファイルはライブラリー QGPL に作成され、QRWDBmmddy (mm はトレースが記録された月、dd は日、y は年の最後の 1 桁を表します) のフォーマットで命名されます。ただし、すべてのシステム・プログラムがトレースされるわけではありません。

関連タスク:

293 ページの『ヒストリー・ログの表示』

クライアント・ユーザーがサーバーとの正常な接続を確立するたびに、そのジョブは、そのクライアント・ユーザーのプロファイルの下で実行するためにスワップされます。

関連資料:

431 ページの『分散リレーショナル・データベースのメッセージ』

サーバーまたはクライアントでエラー・メッセージが出された場合、障害の理由を示すシステム・メッセージがジョブ・ログにログ記録されます。

412 ページの『TCP/IP サーバー事前開始ジョブからのジョブ・ログの検出』

分散データ管理 (DDM) TCP/IP サーバーに関連した QRWTSRVR 事前開始ジョブの 1 つを利用していた接続が終了すると、その事前開始ジョブはリサイクルされて、別の接続で使われます。その場合、終了した接続に関連したジョブ・ログは消去されます。

423 ページの『TCP/IP サーバーのサービス・ジョブ』

DDM TCP/IP サーバーは、拡張プログラム間通信機能 (APPC) サーバーのように、トランザクション・プログラム名 (TPN) を使用することはありません。しかし、TCP/IP サーバーで事前開始ジョブを使えば、その環境でサービス・ジョブを開始する手段になります。

Query 属性変更 (CHGQRYA) コマンド

ジョブ処理 (WRKJOB) コマンド

例: データ域を作成するための CL コマンド:

この例では、表に示されている関数を要求します。

```
CRTDTAARA DTAARA(QGPL/QRWOPTIONS) TYPE(*CHAR) LEN(48)
          VALUE('9.5.114.107 :MYUSERID AAUIU TN INILIBRARY')
          TEXT('DRDA TCP SERVER DIAGNOSTIC OPTIONS')
```

注: この例にある正しいスペーシングが重要なので、VALUE パラメーターの内容は、テーブル形式で繰り返されます。

表 36. CRTDTAARA の VALUE パラメーター内のデータ・エレメントの説明

列	内容	説明
1-15	9.5.114.107	列 30 にあるスイッチで使用するための IP アドレス。
16	:	IP アドレス・フィールドの終わりにマークを付けるための文字
17-26	MYUSERID	列 29 および 31 にあるスイッチで使用するためのユーザー ID
27	A	すべてのユーザーのためのサーバー・ジョブ・ログ (QRWTSRVR 用) をスプールする。

表 36. CRTDTAARA の VALUE パラメーター内のデータ・エレメントの説明 (続き)

列	内容	説明
28	A	すべてのサーバーを使用するために DSPJOB 出力をスプールする。
29	U	接続要求上のユーザー ID がデータ域の列 17 から 26 (この例では 'MYUSERID') に指定されているユーザー ID と一致する場合、 (TRCJOB) コマンドを使用してジョブをトレースする。
30	I	クライアント ID アドレス (この例では、'9.5.114.107') がデータ域の列 1 から 15 に指定されているクライアント IP アドレスと一致する場合、 デバッグの開始 (STRDBG) コマンド (プログラムを指定しない) を使用してデバッグを開始する。
31	U	接続要求上のユーザー ID がデータ域の列 17 から 26 (この例では 'MYUSERID') に指定されているユーザー ID と一致する場合、 照会変更属性 (CHGQRYA) QRYOPTLIB(INILIBRARY) コマンドを呼び出す。 注: ライブラリー名は、データ域の列 39 から 48 から選択する。
32		サーバーに、クライアント・デバッグ・オプションをシャドー作成しない。
33	T	ジョブ・トレースのために古い TRCJOB 機能を使用する。
34	N	ヒストリー・ログ内に CPI3E34 メッセージを入れない。
35-38		予約済みのバイト。
39-48	INILIBRARY	スイッチ 31 で使用されるライブラリー。

関連資料:

Query 属性変更 (CHGQRYA) コマンド

デバッグの開始 (STRDBG) コマンド

分散リレーショナル・データベース・ユーザーの処理

通常、問題の調査は、ユーザーと一緒にに行います。ユーザーは、プログラムを実行したときに予定の結果を得られなかったり、問題を示すメッセージを受け取ったりすることがあります。問題を診断して解決するには、場合によっては、ユーザーと一緒に手順をたどっていくのが最良の方法です。

そのためには、「画面コピー」機能を使用して、ユーザーと一緒にリアルタイムで、または、ユーザーが見た表示のファイルを調べながら作業します。

また、表示の下端に表示されるテキスト行だけでなく、メッセージからもさらに情報を得ることができます。このトピック・コレクションでは、別のユーザーが表示していた画面をコピーする方法と、分散リレーショナル・データベースの作業中に出されるメッセージに関する追加情報の入手方法について説明します。

拡張プログラム間通信機能 (APPC) または TCP/IP を使用する場合、プログラミング問題に加えて、プログラムの開始時またはシステムへの接続時に問題を抱えることがあります。

画面コピー

画面コピー開始 (STRCPYSCN) コマンドを使って、自分のワークステーションから、別のワークステーションで誰かが見ているのと同じ画面を見ることができます。

ユーザーと同じ IBM i プラットフォームにサインオンする必要があります。そのユーザーがリモート・システムにいる場合、ディスプレイ・パススルーを使ってそのシステムにサインオンしてから、**STRCPYSCN** コマンドを入力してそうした別の画面を見ることができます。画面イメージをデータベース・ファイルにコピーできるのは、別のワークステーションにそのイメージがコピーされるときと同時、または別のワークステ

ーションが使えなくなっているときです。コピーしておけば、そのデータを後で処理して、問題の発生時に行われる操作用の監査証跡を準備することができます。

画面イメージを別のワークステーションにコピーするためには、次の要件が満たされなければなりません。

- 両方の画面がともにシステムに対して定義されている。
- 両方の画面がともにカラーであるか、またはともにモノクロームであり、一方がカラーで他方がモノクロームであるということはない。
- 両方の画面が縦横ともに同数の文字位置を持っている。

送信側装置として自分自身のステーション ID を入力した際には、画面イメージのコピーの開始時に、受信側のディスプレイは、サインオン画面を表示していなければなりません。グラフィックスはブランクとしてコピーされます。

同じシステムにまだサインオンしていない場合、次のようなプロセスを使って、他方のユーザーがリモート・システム上で見ている画面を表示します。

1. **パススルーの開始 (STRPASTHR)** コマンドを入力します。

```
STRPASTHR RMTLOCNAME(KC105)
```

2. アプリケーション・サーバー (AS) にログオンします。

3. SRCDEV、OUTDEV、および OUTFILE パラメーターを使用して、**STRCPYSCN** コマンドを入力します。このパラメーターには、以下の意味があります。

- SRCDEV には、ソース装置 (表示イメージを送信するディスプレイ) の名前を指定します。表示コマンドを別の装置に送るには、このパラメーターに ***REQUESTER** 値を入力します。
- OUTDEV には、表示イメージの送信先の出力装置の名前を指定します。この例では、表示イメージは、コマンド (***REQUESTER**) を入力した人のディスプレイに送られます。また、別のディスプレイや別の装置 (第三者が見ているもの) を指定することもできます。あるいは、どの装置にも送らないという指定も可能です (***NONE**)。 ***NONE** 値を使うときは、表示イメージ用の出力ファイルを指定します。
- OUTFILE は、コマンドが活動状態にある間に表示されるすべての画面のイメージが入る出力ファイルの名前を指定します。

```
STRCPYSCN SRCDEV(KC105)  
          OUTDEV(*REQUESTER)  
          OUTFILE(KCHELP/TEST)
```

ソース装置に照会メッセージが送られ、その装置のユーザーに対して、別の装置またはファイルに画面がコピーされることが知らされます。

4. 要求側の装置にイメージを送るには、**g (Go)** と入力します。

送信側のディスプレイの画面が、他方のディスプレイにコピーされます。受信側のディスプレイに表示されるイメージは、送信側のディスプレイのもの 1 画面後をたどります。送信側のディスプレイのユーザーが、活動状態にないキー (Home キーなど) を押すと、両方のディスプレイに同じ画面が表示されます。

画面のコピー中は、受信側表示装置のオペレーターは、画面のコピーが終了するまで、表示装置で他の作業を行うことはできません。

送信側のディスプレイでコピー画面機能を終了するには、任意のコマンド行から**画面コピー終了 (ENDCPYSCN)** コマンドを入力して、実行キーを押します。

```
ENDCPYSCN
```

コピー画面機能を開始したときに表示されていた画面が表示されます。

関連資料:

パススルーの開始 (STRPASTHR) コマンド

メッセージ

IBM i オペレーティング・システムは、さまざまなシステム・メッセージを送って、単純なタイプ入力エラーから、システム装置またはプログラムでのエラーまでの各種の状態を示します。

表示されるメッセージをリストします。

- 現在の画面に表示されるエラー・メッセージ。

このメッセージは、ジョブに割り込んだり、アラームを鳴らしたりすることがあります。このメッセージを表示するには、任意のコマンド行に DSPMSG と入力します。

- システム・オペレーター・メッセージ待ち行列に送られて別の「メッセージの処理」画面に表示される、システム問題に関するメッセージ。

これらのメッセージを表示するには、任意のシステム・コマンド行に DSPMSG QSYSOPR と入力します。

- 装置記述に指定されたメッセージ待ち行列に送られる、システム問題に関するメッセージ。

これらのメッセージを表示するには、任意のシステム・コマンド行に DSPMSG message-queue-name と入力します。

システム・イベントによって、システムは、通知メッセージまたは照会メッセージを送ります。通知メッセージは、システムに関する状況を知らせます。照会メッセージは、システムに関する情報を知らせるとともに、応答が必要です。

メッセージ画面によっては、次のような文字や数字コードがメッセージに添えられていることがあります。

CPF0083

最初の 2 つまたは 3 つの文字は、メッセージのカテゴリーを示します。その後の 4 桁 (接頭部が SQ の場合は 5 桁) の数字は、メッセージの順序番号を示します。前述の例で示されているメッセージ ID は、オペレーティング・システムからの番号 0083 のメッセージであることを示しています。以下に、分散リレーショナル・データベースのいくつかのメッセージ・カテゴリーを示します。

表 37. メッセージ・カテゴリー

カテゴリー	説明	ライブラリー
CPA から CPZ	オペレーティング・システムからのメッセージ	QSYS/QCPFMSG
MCH	ライセンス内部コード・メッセージ	QSYS/QCPFMSG
SQ および SQL	構造化照会言語 (SQL) メッセージ	QSYS/QSQLMSG
TCP	TCP/IP メッセージ	QTCP/QTCPMSGF

画面のメッセージ行に表示されているか、メッセージ待ち行列に入っているメッセージについて情報の続きを入手したい場合には、次のようにします。

1. カーソルをメッセージと同じ行に移動してください。
2. ヘルプ・キーを押します。「追加のメッセージ情報」画面が表示されます。

メッセージ識別コードとそれが置かれているライブラリーが分かっているならば、画面に示されていないメッセージの詳細を知ることができます。この情報を入手するには、次の **メッセージ記述表示 (DSPMSGD)** コマンドを入力します。

DSPMSGD RANGE(SQL0204) MSGF(QSYS/QSQLMSG)

DSPMSGD コマンドを使用すると、メッセージに関する以下の情報を表示できます。

- メッセージ・テキスト
- フィールド・データ
- メッセージ属性

このテキストは、「追加のメッセージ情報」画面に表示されるのと同じメッセージおよびメッセージ・ヘルプ・テキストです。フィールド・データは、そのメッセージに定義されているすべての置換変数と、それらの変数の属性のリストです。メッセージの属性は、メッセージの重大度、ログ記録、レベルの値 (定義されている場合) と、デフォルト・プログラム、デフォルト応答、およびダンプ・パラメーターです。この情報を使って、メッセージが出されたときにユーザーが何をしていたかを判断できます。

関連資料:

メッセージ記述表示 (DSPMSGD) コマンド

メッセージ・タイプ:

「追加のメッセージ情報」画面に、メッセージのメッセージ・タイプと重大度コードが表示されます。

次の表に、IBM i メッセージの異なったメッセージ・タイプとそれに関連する重大度コードを示します。

表 38. メッセージ重大度コード

メッセージ・タイプ	重大度コード
通知メッセージ。 通知だけを目的としており、応答は必要ありません。このメッセージは、機能が進行中であるか、または正常に完了したことを示すことがあります。	00
警告。 エラー条件の可能性が存在します。プログラムは、欠落データを補うなどのデフォルト処置をとった可能性があります。操作の結果は正常と想定されます。	10
エラー。 エラーが見つかりましたが、自動回復手順が行われたと考えられる種類のエラーです。処理は続行されました。誤ったデータの代わりにデフォルト値がとられた可能性があります。操作の結果は正しくない可能性があります。機能は完了しなかったかもしれません。たとえば、リスト内の項目には、正しく実行されたものと、そうでないものがあるかもしれません。	20
重大エラー。 見つかったエラーは、自動回復手順を行うには重大すぎ、デフォルト値はとられません。エラーがソース・データ内にあった場合、データ・レコード全体がスキップされました。プログラム中にエラーが起きた場合、プログラムの異常終了 (重大度 40) に至ります。操作の結果は正しくありません。	30
重大エラー: プログラムまたは機能の異常終了。 プログラムが、正しくないデータを処理することができなかったか、またはユーザーが取り消したために、操作は終了しました。	40

表 38. メッセージ重大度コード (続き)

メッセージ・タイプ	重大度コード
ジョブまたはプログラムの異常終了。ジョブが開始されなかったかまたは正常に開始しなかったか、ジョブ・レベルの機能が所定どおりに行われなかったか、またはジョブが取り消されたかのいずれかです。	50
システム状況。システム操作員メッセージ待ち行列に対してのみ出されます。これは、装置、サブシステム、またはシステムに関する状況または警告を示します。	60
装置安全性。システム操作員メッセージ待ち行列に対してのみ出され、装置が正しく作動していないか、または作動不能になっていることを示します。	70
システム警報およびユーザー・メッセージ。システムをただちに停止するほど重大ではないけれども、予防措置を講じない限り重大度が増す可能性のある事態が存在しています。	80
システム保全性。システム操作員メッセージ待ち行列に対してのみ出されます。サブシステムまたはシステムが作動できない事態があることを示しています。	90
処置。応答の入力またはプリンター用紙の交換などの、何らかの手作業による処置が必要です。	99

分散リレーショナル・データベースのメッセージ:

サーバーまたはクライアントでエラー・メッセージが出された場合、障害の理由を示すシステム・メッセージがジョブ・ログにログ記録されます。

DB2 for i プログラムがサポートする SQL ステートメントからの SQLCODE ごとに、システム・メッセージが存在します。このメッセージは、プリコンパイラ・リスト内、対話式 SQL 上、またはデバッグ・モードでの実行時にジョブ・ログに示されます。ただし、IBM i 製品以外の AS で作業するときは、次のような場合にはエラー条件ごとに固有のメッセージがあるとは限りません。

- エラーが IBM i 製品では使用されない機能に関連している場合。

例えば、DB2 for i では特殊レジスタの CURRENT SQLID をサポートしていないので、SQLCODE -411 (SQLSTATE 56040) 「CURRENT SQLID はリモート・オブジェクトを参照するステートメントでは使用できない」というメッセージはありません。

- エラーがプロダクトに固有のもので、DB2 for i では決して起こらない場合。

DB2 for i では、SQLCODE -925 (SQLSTATE 56021) 「IMS™ または CICS 環境では SQL コミットまたはロールバックは無効です」が出されることはありません。

対応するメッセージをもたない SQLCODE の場合、未認識の SQLCODE、SQLSTATE、およびトークンを示した一般メッセージと、そのメッセージを生成した AS のリレーショナル・データベースの名前が戻されます。個々の条件を判別したりトークンの解釈方法を確かめたい場合は、接続先の AS のリリースに対応する製品資料を調べてください。

CPx3E00 から CPx3EFF および CPI9100 から CPI91FF の範囲のメッセージは、分散リレーショナル・データベース・メッセージに使われます。以下のリストは、すべてが網羅されているわけではありませんが、IBM i オペレーティング・システムの分散データベース・ジョブ・ログに、比較的よく出されるメッセージを示しています。

表 39. 分散リレーショナル・データベースのメッセージ

MSG 識別コード	説明
CPA3E01	*LOCAL ディレクトリー項目を除去すると、構成データが失われることがある。
CPC3EC5	リレーショナル・データベースのディレクトリー項目の一部のパラメーターが無視された。
CPD3E30	指定されたリモート・ネットワーク ID の値が矛盾している。
CPD3E35	リモート・ロケーション・タイプ &2 にはこのリモート・ロケーションは正しくない。
CPD3E36	ポート識別コードが正しくない。
CPD3E38	リモート・ロケーションがそのタイプと矛盾している。
CPD3E39	パラメーター &2 に値 &3 を使用することはできない。
CPD3E3B	サーバー &2 のサーバー許可情報を検索中にエラーが起こった。
CPD3ECA	リレーショナル・データベース・ディレクトリーの操作が完了していない可能性がある。
CPD3E01	混合バイトまたは図形 CCSID はサポートされていない。
CPD3E03	ローカル・リレーショナル・データベース名がディレクトリーにない。
CPD3E05	DDM 会話パスが見つからない。
CPD3E31	DDM TCP/IP サーバーが活動状態になっていない。
CPD3E32	DDM TCP/IP サーバーの終了時にエラーが起こった。
CPD3E33	理由コード &1 で DDM TCP/IP サーバーにエラーが起こった。
CPD3E34	&1 で DDM TCP/IP 通信エラーが起こった。
CPD3E37	理由コード &1 で IP アドレスに対するホスト名のマップが失敗した。
CPF9190	DRDA/DDM TCP/IP 接続の試行時に許可に失敗した。
CPF3E30	DDM TCP/IP サーバーの開始中にエラーが起こった。
CPF3E31	DDM TCP/IP サーバーを開始することができない。
CPF3EC6	DDM TCP/IP 属性変更が正しく実行されなかった。
CPF3EC9	RDB 割り込みの範囲メッセージ
CPF3E0A	リソースの限界を超えた。
CPF3E0B	照会がオープンされない
CPF3E0C	データ・タイプ ID の限界に達した。
CPF3E0D	リレーショナル・データベース終了要求を使用することはできない。
CPF3E01	この DDM パラメーター値はサポートされていない。
CPF3E02	アプリケーション・リクエスターが要求された機能をサポートすることができない。
CPF3E04	1 バイト CCSID がサポートされていない。
CPF3E05	指定された SQL パッケージ・バインド・プロセスが活動状態でない。
CPF3E06	リレーショナル・データベース &1 が見つからない。
CPF3E07	パッケージ・バインド・プロセスが活動状態の時には、このコマンドは正しくない。
CPF3E08	照会オープン障害
CPF3E09	パッケージ・バインド・プロセスの開始の試みが正常に実行されなかった。
CPF3E10	混合バイトまたは図形データはサポートされていない。
CPF3E12	COMMIT または ROLLBACK CL コマンドを使用することはできない。

表 39. 分散リレーショナル・データベースのメッセージ (続き)

MSG 識別コード	説明
CPF3E13	コミットメント制御操作が正常に実行されなかった。
CPF3E14	リレーショナル・データベース終了要求が正常に実行されなかった。
CPF3E16	リレーショナル・データベース &1 のアクセスは認可されていない。
CPF3E17	リレーショナル・データベース終了要求が進行中である。
CPF3E18	リモート・データベースでの COMMIT または ROLLBACK が正常に実行されなかった。
CPF3E19	コミットメント制御操作が正常に実行されなかった。
CPF3E20	DDM 会話パスが見つからない。
CPF3E21	リレーショナル・データベース終了要求が正常に実行されなかった。
CPF3E22	コミットでサーバーのロールバックが起こった。
CPF3E23	DDM データ・ストリームが会話処理機能に違反している。
CPF3E30	DDM TCP/IP サーバーの開始中にエラーが起こった。
CPF3E32	クライアント要求の処理中に DDM TCP/IP サーバー・エラーが起こった。
CPF3E80	DDM データ・ストリームで構文エラーが検出された。
CPF3E81	受け取ったデータ記述子が正しくない。
CPF3E82	リレーショナル・データベースがすでにアクセスされた。
CPF3E83	データのミスマッチ・エラー。
CPF3E84	DDM 会話式プロトコル・エラーが検出された。
CPF3E85	リレーショナル・データベース &4 がアクセスされなかった。
CPF3E86	分散データベースの処理時にエラーが起こった。
CPF3E87	永続エラー条件が検出された。
CPF3E88	リモート・システムで SQL カーソルがすでにオープンされていた。
CPF3E89	照会がオープンされない
CPF3E99	リモート・ユーザーによってリレーショナル・データベース要求が終了された。
CPI9150	SNA 経由でデータベース会話が開始された。
CPI9151	SNA 経由のデータベース会話が終了した。
CPI9152	ターゲット DDM ジョブがソース・システムによって開始された。
CPI9160	TCP/IP 経由でデータベース接続が開始された。
CPI9161	TCP/IP 経由のデータベース接続が終了した。
CPI9162	DDM 接続を処理するために割り当てられたターゲット・ジョブは、TCP/IP 経由でソース・システムによって開始されました。
CPI3E01	ローカル・リレーショナル・データベースが &1 によってアクセスされた。
CPI3E02	ローカル・リレーショナル・データベース・アクセスが終了した。
CPI3E04	リレーショナル・データベース &1 に対する接続が終了した。
CPI3E30	DDM TCP/IP サーバーはすでに活動状態である。
CPI3E31	DDM TCP/IP サーバーは機密保護機構タイプをサポートしない。
CPI3E32	DDM サーバー・ジョブが正常に投入された。
CPI3E33	DDM サーバーは正常に終了した。
CPI3E34	DDM ジョブ xxxx は hh:mm:ss の mm/dd/yy にユーザー yyy をサービス中。(これは、QRWOPTIONS を使って抑制できます。)
CPI3E35	QSYS のプログラム QRWTSRVR 事前開始ジョブ項目が存在していない。

表 39. 分散リレーショナル・データベースのメッセージ (続き)

MSG 識別コード	説明
CPI3E36	リレーショナル・データベース &1 への接続が終了した。
SQ30082	分散データベースの接続の試みで認可が正常に実行されない。
SQL7992	TCP/IP 経由のリレーショナル・データベース &1 に対する CONNECT が完了した。
SQL7993	すでにリレーショナル・データベース &1 に接続されている。

関連概念:

424 ページの『QRWOPTIONS データ域』

DDM または DRDA TCP/IP サーバー・ジョブは、開始されると、ユーザーが診断や他のオプションを指定できるデータ域を探します。このデータ域の名前は QRWOPTIONS で、このデータ域は QGPL ライブラリーに常駐している場合にのみ有効になります。このデータ域は、48 文字のストリングから成っています。

SQL メッセージおよびコード

関連タスク:

271 ページの『分散リレーショナル・データベースのジョブ・ログによる要求情報の追跡』

すべての IBM i ジョブは、それぞれジョブについて入力された要求に関連する情報が入るジョブ・ログを持っています。

APPC のプログラム開始要求障害の処理

サーバー・システム上の IBM i サブシステムは、プログラム開始要求を受け取ると、その要求と一緒に送られてきた情報に基づいてジョブの開始を試みます。サーバーに対するクライアント・ユーザーの権限、要求されたデータベースの存在、およびその他多くの項目が検査されます。

サーバー・サブシステムは、ジョブを開始できないと判断した場合 (たとえば、サーバー上にユーザー・プロファイルが存在しない、ユーザー・プロファイルは存在するが使用不可になっている、またはサーバー上の要求対象のオブジェクトに対してユーザーが正しい許可を受けていないなど)、メッセージ CPF1269 を QSYSMSG メッセージ待ち行列に送ります (ただし、QSYSMSG が存在しなければ QSYSOPR に送ります)。CPF1269 メッセージには、理由コードが 2 つあります (理由コードの片方が 0 であることがあり、その場合は無視できます)。

ゼロ以外の理由コードは、プログラム開始要求が拒否された理由を示しています。リモート・ジョブはサーバーで開始されたことになっているので、メッセージと理由コードは、クライアントではなくサーバーで示されます。クライアントのユーザーには、プログラム開始要求が失敗したことだけが知らされ、失敗した理由は知らされません。クライアントのユーザーは、その要求が失敗した理由を知るには、サーバーのシステム・オペレーターに問い合わせるか、またはサーバーへのディスプレイ・パススルーを使用しなければなりません。

関連概念:



ICF プログラミング (PDF)

TCP/IP の接続要求障害の処理

TCP/IP を使うよう構成された 分散リレーショナル・データベース・アーキテクチャー (DRDA) サーバーでの接続要求が失敗した場合、DDM TCP/IP サーバーの未開始、許可エラーの発生、またはマシンの未稼働がその主な原因です。

サーバーが開始しないかポート ID が有効でない:

DDM TCP/IP サーバーが開始されていない場合に出されるエラー・メッセージは CPE3425 です。

メッセージ:

リモート・ホストが接続操作を拒絶した。

リレーショナル・データベース・ディレクトリー項目の追加 (ADDRDBDIRE) またはリレーショナル・データベース・ディレクトリー項目の変更 (CHGRDBDIRE) コマンドに誤ったポートを指定した場合も、このメッセージが出されます。DB2 for i サーバーの場合、ポートは通常は *DRDA (446 の DRDA ウェルノウン・ポート) でなければなりません。ただし、IPSec と一緒に使用するようポート 447 を構成している場合、機密データの送信にはそのポートを使うのがよいかもしれません。Secure Socket Layer (SSL) をサポートする DRDA クライアントを使用する場合、サーバー上のポート 448 に接続しなければなりません。

リモート・システムで DDM サーバーを開始するには、TCP/IP サーバーの開始 (STRTCPSVR) *DDM コマンドを実行します。DDM TCP/IP 属性の変更 (CHGDDMTCPA) AUTOSTART(*YES) コマンドを実行すれば、TCP/IP の開始のたびに DDM サーバーを開始するよう要求することができます。

関連概念:

104 ページの『Secure Sockets Layer』

DB2 for i 分散リレーショナル・データベース体系 (DRDA) クライアントおよびサーバーは、Secure Sockets Layer (SSL) をサポートします。インターネット・プロトコル・セキュリティ・アーキテクチャー (IPSec) を使用して同様の機能が提供されます。

関連資料:

リレーショナル・データベース・ディレクトリー項目の追加 (ADDRDBDIRE) コマンド

DDM TCP/IP 属性変更 (CHGDDMTCPA) コマンド

リレーショナル・データベース・ディレクトリー項目の変更 (CHGRDBDIRE) コマンド

TCP/IP サーバー開始 (STRTCPSVR) コマンド

DRDA 接続許可の障害:

許可障害の場合に出されるエラー・メッセージは SQ30082 です。

メッセージ・テキスト:

分散データベースの接続の試行の際の許可の障害。

このメッセージの原因のセクションに、理由コードと、考えられる理由コードの意味のリストが示されません。理由コード 17 は、サポートされていないセキュリティ・メカニズムがあることを意味します。

DB2 for i は、IBM i アプリケーション・リクエスター (AR) が使用できるいくつかの分散リレーショナル・データベース体系 (DRDA) セキュリティ・メカニズムを実装します。

- ユーザー ID のみ
- ユーザー ID とパスワード
- 暗号化パスワード・セキュリティ・メカニズム
- 暗号化ユーザー ID およびパスワード・セキュリティ・メカニズム
- Kerberos

暗号化パスワードが送られるのは、接続の開始時にパスワードが使用可能になっている場合のみです。

デフォルトの IBM i のセキュリティー・メカニズムは、ユーザー ID とパスワードのメカニズムです。アプリケーション・リクエスターが、デフォルトのセキュリティー構成を使って、パスワードなしでユーザー ID をシステムに送ると、理由コード 17 付きのエラー・メッセージ SQ30082 が表示されます。

サポートされていないセキュリティー・メカニズムの障害の解決法は次のとおりです。

- クライアントがサーバーによって信頼され、認証が不要の場合、DDM TCP/IP サーバーの認証設定をパスワード不要に変更します。
- クライアントがサーバーによって信頼されていないで、認証が必要な場合、アプリケーションを変更してパスワードまたは認証セキュリティー・トークン (たとえば、Kerberos トークン) を送信するようにします。

DDM TCP/IP サーバーの認証設定を変更するには、DDM TCP/IP 属性変更 (CHGDDMTCPA) コマンドか、System i ナビゲーター を使用します。System i ナビゲーター を使用する場合は、「ネットワーク」 > 「サーバー」 > 「TCP/IP」 > 「DDM」を展開し、「DDM」を右クリックして、「プロパティ」を選択して、設定を変更します。

SQL CONNECT ステートメントの USER/USING 形式を使用すると、パスワードを送信できます。サーバー認証項目追加 (ADDSVRAUTE) コマンドを使用してパスワードを送信することもできます。このコマンドは、接続の試行時に使用するユーザー・プロファイルのサーバー認証項目中にリモート・ユーザー ID とパスワードを追加します。暗号化されたパスワードの送信が自動的に試みられます。

リモート・パスワードをサーバーの認証項目に保管するには、システム値 QRETSVRSEC (サーバーのセキュリティー・データの保存) を 1 に設定する必要があることに注意してください。

注: DRDA で使用するサーバー認証項目追加 (ADDSVRAUTE) コマンド上の RDB 名は、大文字で入力しなければなりません。そうしないと、接続処理中にその名前は認識されず、認証項目内の情報は使われなくなります。

関連資料:

サーバー認証項目追加 (ADDSVRAUTE) コマンド

DDM TCP/IP 属性変更 (CHGDDMTCPA) コマンド

システムが使用不可:

リモート・システムが立ち上がって作動していないか、またはサーバー用の IP アドレスを RDB ディレクトリー項目に誤って入力した場合、CPE3447 メッセージを受け取ります。

メッセージ・テキスト:

リモート・ホストがタイムアウト期間内に応答しなかった。

通常、このメッセージが出されるまでに、数分の遅延があります。その時点で、何かがハングアップしたか、またはループが起きたように見えることがあります。

対話式 SQL 固有の接続の障害:

対話式 SQL から CONNECT ステートメントを実行しているときに、場合によっては、SQ30080 一般メッセージが出されます。

メッセージのテキスト:

分散データベース処理中に通信エラーが起きた

エラーの詳細を知るには、対話式 SQL を終了してジョブ・ログを調べる必要があります。

単一フェーズ・コミット機能しかないシステムに初めて接続した (どのレベルのコミットメント制御であっても) ときに、メッセージ SQL7020 (SQL パッケージの作成が正常に実行されなかった) が出された場合は、リモート・システムに読み取り専用システムとしてアクセスしたのに、それを更新して SQL パッケージを作成しようとしたことが原因と考えられます。

ジョブ・ログ内のメッセージを調べれば、それを確認できます。それを解決するには、RELEASE ALL と COMMIT を行って、まずすべての接続を取り除いてから接続すれば、接続は更新可能になります。

関連タスク:

74 ページの『対話式 SQL 用 SQL パッケージのセットアップ』

このトピックは、IBM i を実行していないサーバー・システムのみに適用されます。

サーバーの事前開始ジョブが不十分:

QSYSWRK (OS/400 V5R1 またはそれ以前) または (OS/400 V5R2、IBM i V5R3 以降) QSYSWRK サブシステムの QRWTSRVR 事前開始ジョブ項目によって、TCP/IP サーバーに関連付けられる事前開始ジョブの数が限定されている場合に、1 つの接続ですべての事前開始ジョブが使われていると、新たに接続を試みても失敗して、次のようなメッセージが表示されます。

CPE3426

リモート・ソケットとの接続はそのソケットによってリセットされた。

CPD3E34

recv() で DDM TCP/IP 通信エラーが起こった — MSG_PEEK。

サーバーでこのような問題が起きないようにするには、QRWTSRVR 項目用の事前開始ジョブ項目変更 (CHGPJE) コマンドの MAXJOBS パラメーターをもっと大きな数または *NOMAX に設定するか、ADLJOBS パラメーターを 0 以外の値に設定します。

関連資料:

事前開始ジョブ項目変更 (CHGPJE) コマンド








分散データベースのプログラミングに関連する情報



製品資料、IBM Redbooks 資料、Web サイト、およびその他の Information Center のトピックには、分散データベース・プログラミングのトピックに関連した情報が記載されています。すべての PDF ファイルは表示および印刷できます。

IBM i 情報

これらの資料および Information Center のトピックには、必要な情報が含まれています。

- APPC、APPN、および HPR (高性能経路指定) トピックは、アプリケーション・プログラマーを対象に、IBM i オペレーティング・システムによって提供される拡張対等間通信ネットワーク機能 (APPN) サポートに関する説明を行っています。このトピックは、APPN ネットワークの構成法に関して説明し、APPN を用いるときの考慮事項について述べています。この資料は、APPC 通信の通信環境のプログラミングや定義を行うための手引きにもなります。
- Application programming interfaces のトピックには、システム・パフォーマンスの管理、スプーリングの効率的な使用、ならびにデータベース・ファイルの効率的な維持管理に役立つ、オブジェクトの作成、使用、および削除の方法が説明されています。このトピックには、システム・オブジェクト用のプログラムの作成と維持およびオブジェクト、データベース・ファイル、ジョブ、およびスプーリングで作業するときの IBM i 情報の検索に関する情報もあります。

- 『バックアップおよび回復』。システム・データを保管および復元するのに使用することができるさまざまなメディアについて、またデータベース・ファイルに加えられた変更の記録方法と、その情報をシステム回復と活動報告書に利用する方法についての詳細を、システム・プログラマーを対象に説明しています。
- 『制御言語 (CL)』。オブジェクトおよびライブラリーの一般説明、制御言語 (CL) プログラミング、プログラム相互の流れの制御と通信、CL プログラムにおけるオブジェクトの処理、および CL プログラムの作成などの広範囲にわたるプログラミングの解説を述べています。他に、事前定義メッセージと即時メッセージ、およびメッセージ処理法について、さらに、デバッグ・モード、ブレークポイント、トレース、および表示機能も含め、ユーザー定義のコマンドとメニューの定義と作成、およびアプリケーションのテストの方法についても述べています。
- **Communications Management** 。これには、通信状況の処理に関する詳細、通信関連の作業の管理に関する項、通信エラー、パフォーマンス、回線速度とサブシステム記憶域についての解説があります。
- **Control language** トピックでは、アプリケーション・プログラマーを対象に、IBM iの制御言語 (CL) とそのコマンドについて説明しています。
- **DSNX Support** 。リモート管理サポート (分散ホスト・コマンド機能)、変更管理サポート (分散システム・ノード管理機能)、および問題管理サポート (警報) を使えるよう IBM i 製品を構成する方法の詳細が述べられています。
- **ICF Programming** 。これは、アプリケーション・プログラマーを対象に、IBM i 通信および ICF ファイルを使用するアプリケーション・プログラムの作成に必要な情報を提供しています。また、データ記述仕様 (DDS) キーワード、システム提供形式、戻りコード、ファイル転送サポート、およびプログラミング例も示されています。
- **ILE Concepts (ILE 概念、SD88-5033)**  は、アプリケーション・プログラマーを対象に、IBM i オペレーティング・システムの統合言語環境 (Integrated Language Environment) の概念と用語について説明しています。この資料には、ILE モデルの概説もあり、プログラムの作成、実行時、およびデバッグの概念について述べています。また、記憶域と状態の管理の解説と、呼び出しと API の説明もあります。
- 『LAN、フレームリレーおよび ATM サポート』 。これには、トークンリング・ネットワーク、イーサネット・ネットワーク、またはブリッジ・ネットワーク環境での IBM i 製品の使用に関する情報が含まれています。
- **Local Device Configuration** 。システム・オペレーターまたはシステム管理者を対象に、初期ローカル・ハードウェアの構成方法とその構成の変更方法について説明しています。また、装置構成の概念情報と、9406、9404、および 9402 システム装置での装置構成の計画情報も記載しています。
- **OptiConnect** は、OptiConnect を使用した通信のインストール、使用、および管理について説明しています。
- 『Query 管理プログラミング』 。この資料は、アプリケーション・プログラマーを対象に、報告書用に照会するデータベース・ファイルの判別方法、構造化照会言語 (SQL) の Query 定義の定義方法、および照会管理コマンドを使用するプロシージャの使用と作成の方法を説明しています。この資料では、照会グローバル変数サポートの使用法について、また、オペレーティング・システムの Query 管理機能と IBM Query for IBM i との関係についても解説されています。

- Remote Work Station Support 。ディスプレイ・パススルー、分散ホスト・コマンド機能、および 3270 リモート接続機構などのリモート・ワークステーション・サポートのセットアップと使用の方法について説明しています。また、アプリケーション・プログラマーまたはシステム・プログラマーを対象に、構成コマンド、および回線、コントローラー、および装置の定義に関する情報も提供します。
- 『セキュリティ』。システム・プログラマー（またはセキュリティの責任担当者）を対象に、システム・セキュリティの概念、セキュリティの計画、およびシステムでのセキュリティのセットアップについて詳述しています。
- SQL プログラミングのトピックでは、アプリケーション・プログラマー、プログラマー、またはデータベース管理者を対象に、非 SQL ステートメントを設計、作成、テストおよび実行する方法の概要が提供されます。さらに、対話式構造化照会言語 (SQL) についても説明しています。
- 『SQL リファレンス (SQL reference)』。アプリケーション・プログラマー、プログラマー、またはデータベース管理担当者を対象に、SQL ステートメントおよびそのパラメーターに関して詳述しています。
- X.25 Network Support  には、X.25 ネットワークでの IBM i 製品の使用に関する情報が含まれています。
- 以下の資料は、V6R1 IBM i Information Center に組み込まれていません。しかし、これらの資料は有用な参照資料になることがあります。各資料は、印刷されたハードコピーを IBM Publications Center にオーダーするか、オンライン形式で無料でダウンロードして入手できます（両方も可能）。
 - 通信構成
 - SNA 配布サービス

分散リレーショナル・データベース・ライブラリー

このトピックの資料は IBM Distributed Relational Database Architecture (DRDA) の実装について、その背景と一般的なサポート情報を提供します。

- *DRDA: Every Manager's Guide* (GC26-3195)。分散リレーショナル・データベースと配布ファイルについての簡潔でしかもハイレベルな研修にお使いいただけます。この資料では、分散データ・システムの開発に対する IBM のサポートについて述べ、分散データ用の現在の IBM 製品と発表済みサポートについて説明しています。この資料の内容は、管理職、管理者、および技術担当者が、分散データの概念を理解したいときに参考にさせていただくためのものです。
- *DRDA: 分散リレーショナル・データベースの計画*、(N: SC26-4650)。分散リレーショナル・データの計画に役立ちます。段階的なステップ、決定事項、および決定にあたって選択するオプションについて説明しています。また、現在入手可能であるか、またはすでに発表済みの、分散リレーショナル・データベース製品とその機能についても述べており、将来における分散リレーショナル・データのサポートに関する弊社の方針を説明しています。この資料の内容は、計画担当者を対象としています。
- *DRDA: 接続の手引き*、(SC88-7070)。Distributed Relational Database Architecture をサポートする IBM 製品の相互接続の方法を説明しています。分散リレーショナル・データベースとネットワーク・システムに関連した概念と用語について説明しています。また、分散環境内の異種システムの接続方法について述べています。「接続の手引き」の記載事項は、どの製品資料にも含まれていません。この資料の内容は、システム管理者、データベース管理者、通信管理者、およびシステム・プログラマーを対象としています。
- *DRDA: 適用業務プログラミングの手引き*、(N: SC26-4773)。IBM リレーショナル・データベース管理システムにアクセスするアプリケーション・プログラムの設計、作成、および変更の方法を説明しています。この資料では、異種環境用の分散リレーショナル・データベース・アプリケーションを作成する場合に、プログラマーがそれぞれに応じて変えなければならない点に焦点が置かれています。記載内容には、プログラムの設計、準備、および実行に加え、パフォーマンス上の考慮事項も含まれています。


IBM C で作成されたプログラミング例も入っています。この資料の情報は、IBM 高水準言語のうちの少なくとも 1 つと、構造化照会言語 (SQL) を使って作業するアプリケーション・プログラマーを対象とします。

- *DRDA: 問題判別の手引き*, (N: SC26-4782)。分散リレーショナル・データベース環境において問題の原因を究明するのに役立ちます。この資料は、各製品をよく知らない人を対象に、それぞれの製品の紹介を記載し、各製品ごとに問題の診断と報告の方法を詳述しています。この手引きは、各ホスト・システムごとに固有の手順およびツールと、各種のシステムに共通の手順およびツールについて説明しています。この資料の情報は、分散リレーショナル・データベースの問題を IBM サポート・センターに報告する担当者を対象とします。
- *IBM SQL Reference, Volume 2*, (SC26-8416)。DRDA DRDA を参照して、以下の機能を比較しています。
 - IBM SQL リレーショナル・データベース製品
 - IBM SQL
 - ISO-ANSI SQL (SQL92E)
 - X/Open SQL (XPG4-SQL)
 - ISO-ANSI SQL 呼び出しレベル・インターフェース (CLI)
 - X/Open CLI
 - Microsoft オープン・データベース・コネクティビティ (ODBC) バージョン 2.0

他の IBM 分散リレーショナル・データベース・プラットフォームのライブラリー

このトピックでは、他の IBM 分散リレーショナル・データベース・プラットフォームのライブラリーについて説明します。

DB2 Connect および DB2 Universal Database

DB2 Connect および Universal Database を使用して作業し、詳細が必要な場合は、Web ページ [DB2 for Linux, UNIX, および Windows](#)  を参照してください。このサイトには以下の資料があります。

- *DB2 Connect Enterprise Edition Quick Beginning*
- *DB2 Connect Personal Edition Quick Beginning*
- *DB2 Connect User's Guide*
- *DB2 UDB Administration Guide*
- *DB2 UDB Command Reference*
- *DB2 UDB for OS/2 Quick Beginnings*
- *DB2 UDB for UNIX Quick Beginnings*
- *DB2 UDB for Windows NT Quick Beginnings*
- *DB2 UDB Messages Reference*
- *DB2 UDB Personal Edition Quick Beginnings*
- *DB2 UDB SQL Getting Started*
- *DB2 UDB SQL Reference*
- *DB2 UDB Troubleshooting Guide*

DB2 for z/OS and OS/390®

DB2 for z/OS and OS/390 を使用し詳細が必要な場合は、Web ページ [DB2 for z/OS !\[\]\(1d3a1175dd4902218e694b9c098adb83_img.jpg\)](#) を参照してください。このサイトには以下の資料があります。

- *DB2 for z/OS and OS/390 Command Reference*
- *DB2 for z/OS and OS/390 Messages and Codes*
- *DB2 for z/OS and OS/390 Reference for Remote DRDA*
- *DB2 for z/OS and OS/390 SQL Reference*
- *DB2 for z/OS and OS/390 Utility Guide and Reference*

DB2 Server for VSE & VM

DB2 Server for VSE & VM を使用し詳細が必要な場合は、Web ページ [DB2 Server for VSE & VM !\[\]\(cbe80b694ebd74fcfe136a095b608235_img.jpg\)](#) を参照してください。このサイトには以下の資料があります。

- *DB2 and Data Tools for VSE and VM*
- *DB2 for VM Control Center Installation*
- *DB2 Server Data Spaces Support for VM/ESA*
- *DB2 Server for VM Application Programming*
- *DB2 Server for VM Database Administration*
- *DB2 Server for VM Database Services Utilities*
- *DB2 Server for VM Diagnosis Guide*
- *DB2 Server for VM Interactive SQL Guide*
- *DB2 Server for VM Master Index and Glossary*
- *DB2 Server for VM Messages and Codes*
- *DB2 Server for VM Operation*
- *DB2 Server for VM System Administration*
- *DB2 Server for VM/VSE Training Brochure*
- *DB2 Server for VSE & VM Quick Reference*
- *DB2 Server for VSE & VM SQL Reference*
- *DB2 Server for VSE & VM LPS*
- *DB2 Server for VSE & VM Data Restore*
- *SBOF for DB2 Server for VM*

アーキテクチャー・ブック

DDM および DRDA のアーキテクチャー・ブックを以下にリストします。

分散データ管理アーキテクチャー

- *Distributed Data Management Architecture: General Information, GC21-9527*
- *Distributed Data Management Architecture: Implementation Programmer's Guide, SC21-9529*
- *Distributed Data Management Architecture: Reference, SC21-9526*

DRDA アーキテクチャー


- *Character Data Representative Architecture: Details*, SC09-2190

この資料には CD-ROM が添付されています。それには、オンライン・ブック形式の CDRA 資料、バイナリー・フォームの変換テーブル、多数の変換バイナリー用のマッピング・ソース、コード・ページと文字セット・リソースの集合体、IBM で使用されている文字命名情報が入っています。また CD には、付属媒体で使用するための表示ユーティリティーも入っています。ビューアーは、OS/2、Windows 3.1、および Windows 95 でご利用いただけます。

- *Character Data Representative Architecture: Overview*, GC09-2207


1 • DRDA V5 Vol. 1: Distributed Relational Database Architecture

このテクニカル資料は、Distributed Relational Database Architecture の仕様を文書化した 3 つの資料のうちの一つです。この巻では、アプリケーション・プログラムが分散リレーショナル・データにアクセスするのに使用できるリレーショナル・データベースの管理機能の相互接続について説明しています。分散環境におけるアプリケーションとリレーショナル・データベース管理システムとの間に必要な接続、関係者の責務と、いつやりとりを行う必要があるか、また、分散リレーショナル・データベース管理システムの処理に必要なフォーマットとプロトコルについて述べています。分散データベース管理システムの処理用の API は説明されていません。このドキュメントは、Open Group Web サイト

(<https://collaboration.opengroup.org/dbiop/?gpid=453&gpid=453&gpid=453>) から入手できます。


1 • DRDA V5 Vol. 2: Formatted Data Object Content Architecture

このドキュメントは、Distributed Relational Database Architecture、バージョン 4 の仕様を文書化した 3 つの資料のうちの一つです。この巻では、定様式データ・オブジェクト内容体系 (FD:OCA) を構成する機能とサービスについて説明しています。この体系を使って、別々のデータ・タイプとデータ表示方式を利用するそれぞれの環境相互の接続のギャップを埋めることができます。FD:OCA は、DRDA に組み込まれています。このドキュメントは、Open Group Web サイト

(<https://collaboration.opengroup.org/dbiop/?gpid=453&gpid=453&gpid=453>) から入手できます。

1 • DRDA V5 Vol. 3: Distributed Data Management Architecture

このドキュメントは、Distributed Relational Database Architecture (DRDA) の仕様を文書化した 3 つの資料のうちの一つです。この巻では、DDM データ・ストリームの体系化コマンド、パラメーター、オブジェクト、およびメッセージについて説明しています。このデータ・ストリームは、さまざまな DDM モデルが互いにデータを交換するのに使われます。このドキュメントは、Open Group Web サイト

(<https://collaboration.opengroup.org/dbiop/?gpid=453&gpid=453&gpid=453>) から入手できます。

IBM Redbooks

このトピックでは、分散リレーショナル・データベースに使用できる IBM Redbooks について説明します。

- DRDA Client/Server for VM and VSE Setup for System and Performance Management 

- WOW! DRDA Supports TCP/IP: DB2 Server for OS/390 and DB2 Universal Database 

コードに関するライセンス情報および特記事項

IBM は、お客様に、すべてのプログラム・コードのサンプルを使用することができる非独占的な著作使用权を許諾します。お客様は、このサンプル・コードから、お客様独自の特別のニーズに合わせた類似のプログラムを作成することができます。

強行法規で除外を禁止されている場合を除き、IBM、そのプログラム開発者、および供給者は「プログラム」および「プログラム」に対する技術的サポートがある場合にはその技術的サポートについて、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。

いかなる場合においても、IBM および IBM のサプライヤーならびに IBM ビジネス・パートナーは、その予見の有無を問わず発生した以下のものについて賠償責任を負いません。

1. データの喪失、または損傷。
2. 直接損害、特別損害、付随的損害、間接損害、または経済上の結果的損害
3. 逸失した利益、ビジネス上の収益、あるいは節約すべかりし費用

国または地域によっては、法律の強行規定により、上記の責任の制限が適用されない場合があります。

特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒103-8510

東京都中央区日本橋箱崎町19番21号

日本アイ・ビー・エム株式会社

法務・知的財産

知的財産権ライセンス渉外

以下の保証は、国または地域の法律に沿わない場合は、適用されません。IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Corporation

Software Interoperability Coordinator, Department YBWA

3605 Highway 52 N

Rochester, MN 55901

U.S.A.

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性があります。その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

本書はプランニング目的としてのみ記述されています。記述内容は製品が使用可能になる前に変更になる場合があります。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。これらのサンプル・プログラムは特定物として現存するままの状態を提供されるものであり、いかなる保証も提供されません。IBM は、お客様の当該サンプル・プログラムの使用から生ずるいかなる損害に対しても一切の責任を負いません。

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生的創作物にも、次のように、著作権表示を入れていただく必要があります。

© (お客様の会社名) (西暦年). このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。

© Copyright IBM Corp. _年を入れる_.

プログラミング・インターフェース情報

本書「分散データベース・プログラミング」には、プログラムを作成するユーザーが IBM i のサービスを使用するためのプログラミング・インターフェースが記述されています。

商標

IBM、IBM ロゴおよび ibm.com は、世界の多くの国で登録された International Business Machines Corporation の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、『www.ibm.com/legal/copytrade.shtml』をご覧ください。

Adobe、Adobe ロゴ、PostScript、PostScript ロゴは、Adobe Systems Incorporated の米国およびその他の国における登録商標または商標です。

インテル、Intel、Intel ロゴ、Intel Inside、Intel Inside ロゴ、Intel Centrino、Intel Centrino ロゴ、Celeron、Intel Xeon、Intel SpeedStep、Itanium、および Pentium は、Intel Corporation または子会社の米国およびその他の国における商標または登録商標です。

Linux は、Linus Torvalds の米国およびその他の国における登録商標です。

Microsoft、Windows、Windows NT および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。

UNIX は The Open Group の米国およびその他の国における登録商標です。

Java およびすべての Java 関連の商標およびロゴは Oracle やその関連会社の米国およびその他の国における商標または登録商標です。

他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。

使用条件

これらの資料は、以下の条件に同意していただける場合に限りご使用いただけます。

個人使用: これらの資料は、すべての著作権表示その他の所有権表示をしていただくことを条件に、非商業的な個人による使用目的に限り複製することができます。ただし、IBM の明示的な承諾をえずに、これらの資料またはその一部について、二次的著作物を作成したり、配布（頒布、送信を含む）または表示（上映を含む）することはできません。

商業的使用: これらの資料は、すべての著作権表示その他の所有権表示をしていただくことを条件に、お客様の企業内に限り、複製、配布、および表示することができます。ただし、IBM の明示的な承諾をえずにこれらの資料の二次的著作物を作成したり、お客様の企業外で資料またはその一部を複製、配布、または表示することはできません。

ここで明示的に許可されているもの以外に、資料や資料内に含まれる情報、データ、ソフトウェア、またはその他の知的所有権に対するいかなる許可、ライセンス、または権利を明示的にも黙示的にも付与するものではありません。

資料の使用が IBM の利益を損なうと判断された場合や、上記の条件が適切に守られていないと判断された場合、IBM はいつでも自らの判断により、ここで与えた許可を撤回できるものとさせていただきます。

お客様がこの情報をダウンロード、輸出、または再輸出する際には、米国のすべての輸出入関連法規を含む、すべての関連法規を遵守するものとします。

IBM は、これらの資料の内容についていかなる保証もしません。これらの資料は、特定物として現存するままの状態を提供され、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任なしで提供されます。



プログラム番号: 5770-SS1

Printed in Japan

日本アイ・ビー・エム株式会社

〒103-8510 東京都中央区日本橋箱崎町19-21