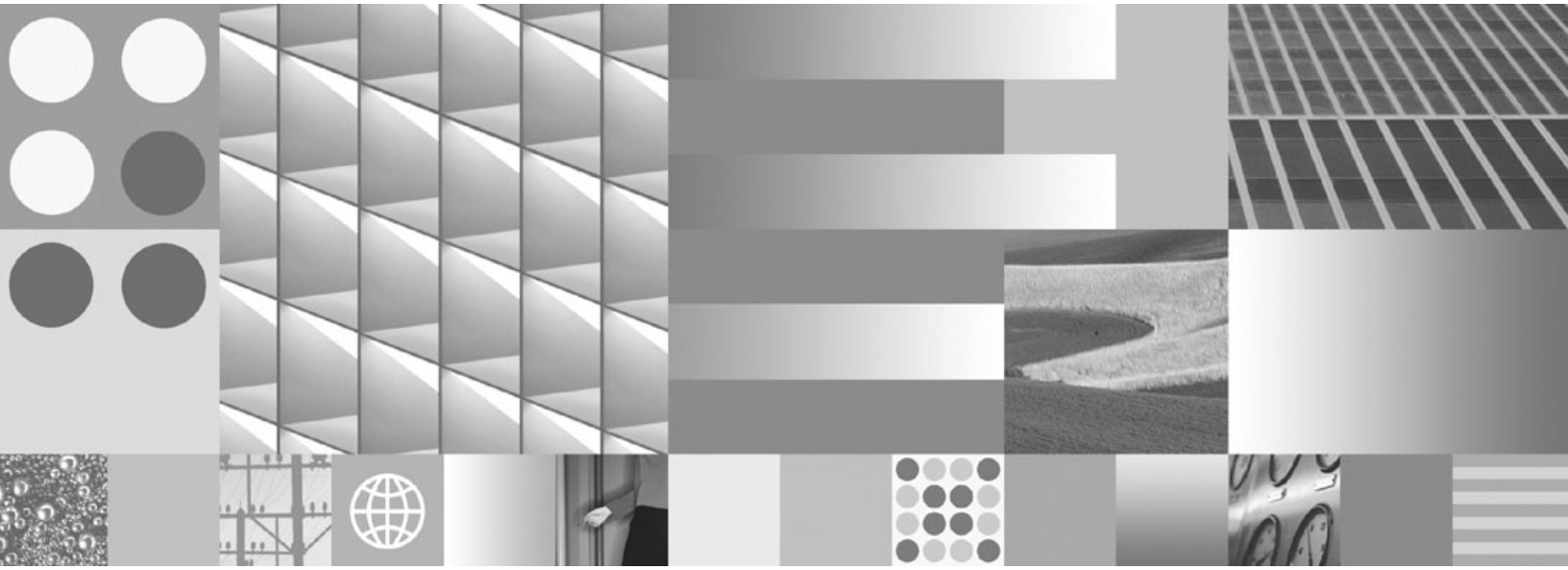
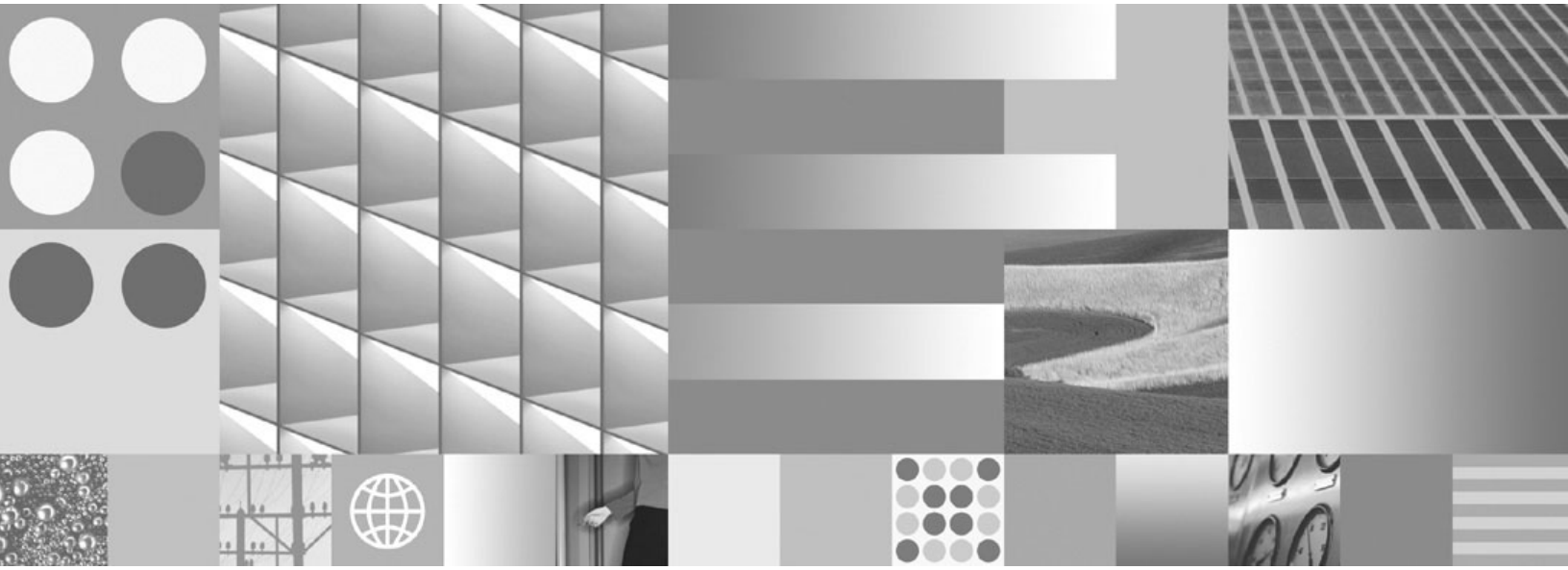
DB2 Version 9.5
for Linux, UNIX, and Windows



データベース・セキュリティー・ガイド

SC88-4418-00
(英文原典：SC23-5850-00)

DB2 Version 9.5
for Linux, UNIX, and Windows



データベース・セキュリティー・ガイド

SC88-4418-00
(英文原典：SC23-5850-00)

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、 291 ページの『付録 B. 特記事項』に記載されている情報をお読みください。

原　典： SC23-5850-00
DB2 Version 9.5 for Linux, UNIX, and Windows
Database Security Guide

発　行： 日本アイ・ビー・エム株式会社

担　当： ナショナル・ランゲージ・サポート

第1刷 2007.10

目次

本書について

「データベース・セキュリティー・ガイド」では、DB2® セキュリティー・フィーチャーを使用して、データベースのインストールで必要なレベルのセキュリティーをインプリメントおよび管理する方法について説明します。

「データベース・セキュリティー・ガイド」では、以下について詳しく説明します。

- DB2 データベースにアクセスできるユーザーの認証の管理
- ユーザーによるデータベース・オブジェクトおよびデータへのアクセスを制御するための許可の設定

第 1 章 DB2 のセキュリティー・モデル

DB2 データベース・システムのデータと関数に対するアクセスを制御するセキュリティー・モードが 2 つあります。 DB2データベース・システムへのアクセスは、DB2 データベース・システムの外部にある機能 (認証) によって管理するのに対し、DB2 データベース・システムの内部のアクセスは、データベース・マネージャー (許可) によって管理します。

認証

認証とは、システムがユーザーの身元を検証するプロセスのことです。ユーザー認証は、DB2 データベース・システムの外部にあるセキュリティー機能によって、認証セキュリティー・プラグイン・モジュールを経由して実行されます。オペレーティング・システム・ベースの認証に依存するデフォルトの認証セキュリティー・プラグイン・モジュールは、DB2 データベース・システムのインストール時に組み込まれます。より柔軟に具体的な認証要件に対応するには、独自の認証セキュリティー・プラグイン・モジュールを作成します。

認証プロセスでは、DB2 許可 ID が生成されます。認証時には、ユーザーのグループ・メンバーシップ情報も取得されます。デフォルトのグループ情報取得機能は、オペレーティング・システム・ベースのグループ・メンバーシップ・プラグイン・モジュールに依存しています。そのモジュールは、DB2 データベース・システムのインストール時に組み込まれます。必要に応じて、Lightweight Directory Access Protocol (LDAP) などのグループ・メンバーシップ・プラグイン・モジュールを使用して、グループ・メンバーシップ情報を取得することも可能です。

許可

ユーザーが認証されると、データベース・マネージャーは、そのユーザーが DB2 のデータやリソースにアクセスする許可を持っているかどうかを確認します。 DB2 データベース・マネージャーは、許可プロセスの中で、認証済みのユーザーがどのデータベース操作を実行でき、どのデータ・オブジェクトにアクセスできるのかを示す情報を取得します。

許可 ID に与えられる許可のさまざまなソースを以下にまとめます。

1. 1 次許可: 許可 ID に直接付与されている許可。
2. 2 次許可: 許可 ID がメンバーとして属しているグループやロールに付与されている許可。
3. パブリック許可: PUBLIC に付与されている許可。
4. コンテキスト依存許可: トラステッド・コンテキストロールに付与されている許可。

ユーザーに付与できる許可のカテゴリーを以下にまとめます。

- システム・レベルの許可

  システム管理者 (SYSADM)、システム制御 (SYSCTRL)、システム保守 (SYSMAINT)、システム・モニター (SYSMON) の各権限によって、インスタン

ス・レベルの機能をさまざまな程度で制御します。権限は、特権をグループ化するため、およびインスタンス、データベース、データベース・オブジェクトに対する保守およびユーティリティー操作を制御するための手段として使用します。

- データベース・レベルの許可

  セキュリティー管理者 (SECADM)、データベース管理者 (DBADM) の各権限によって、データベース内のアクセスを制御します。その他のデータベース権限としては、LOAD (表にデータをロードする権限)、CONNECT (データベースに接続する権限) があります。

- オブジェクト・レベルの許可

  オブジェクト・レベルの許可では、オブジェクトに対する操作の実行時に特権をチェックします。例えば、表のデータを選択 (抽出) するユーザーには、最低でも表に対する SELECT 特権が必要です。

- 内容に基づく許可

  特定のユーザーが読み取れる表の列や行を制御するためにビューを使用します。個々の行や個々の列に対する読み取りアクセスと書き込みアクセスを持っているユーザーを判別するために、ラベル・ベースのアクセス制御 (LBAC) を使用します。

これらの機能と、アクセスをモニターするための DB2 監査機能を併用して、それぞれのデータベース・インストール・システムで必要なレベルのセキュリティーを定義し、管理します。

認証

ユーザーの認証は、DB2 データベース・システムの外部のセキュリティー機能を使用して完了します。セキュリティー機能は、オペレーティング・システムの一部であるか、別個の製品にすることができます。

セキュリティーは、ユーザーを認証するのに 2 つの項目を必要とします。それは、ユーザー ID とパスワードです。ユーザー ID は、セキュリティー機能にユーザーを知らせます。正しいパスワード、ユーザーおよびセキュリティー機能にのみ認識されている情報を提供すれば、ユーザーの身元 (ユーザー ID に対応) が検証されます。

注: 非ルート・インストールでは、オペレーティング・システム・ベースの認証は、db2rfe コマンドを実行して有効にしなければなりません。

認証された後、次のようにします。

- ユーザーは SQL 許可名または *authid* を使用して、DB2 に識別されなければなりません。この名前は、ユーザー ID と同じものか、またはマップ値にすることができます。例えば、UNIX® オペレーティング・システムでは、デフォルトのセキュリティー・プラグイン・モジュールを使用している場合、DB2 *authid* は、DB2 命名規則に従ったUNIX ユーザー ID を大文字に変換することによって得られます。
- そのユーザーが属しているグループのリストが取得されます。グループ・メンバーシップは、ユーザーを許可するときに使用されます。グループは、DB2 許可名

にもマップする必要がある、セキュリティー機能のエンティティーです。このマッピングは、ユーザー ID のために使用される方法と類似した方法で行われます。

DB2 データベース・マネージャーは、セキュリティー機能を使用して、以下の 2 つの方法のうちの 1 つでユーザーを認証します。

- 成功したセキュリティー・システム・ログインを識別の証拠として使用し、次のことを可能にします。
  - ローカル・データをアクセスするためのローカル・コマンドの使用
  - サーバーがクライアント認証を委託している場合のリモート接続の使用
- ユーザー ID とパスワードの妥当性検査がセキュリティー機能によって成功すると、それをユーザーのアイデンティティーの証拠として使用して、以下のことを許可します。
  - サーバーが認証の検査を必要とするリモート接続の使用
  - ログインに使用されたアイデンティティー以外のアイデンティティーの下でユーザーがコマンドを実行する場合の操作の使用

注: 一部の UNIX システムでは、DB2 データベース・マネージャーはオペレーティング・システムで失敗したパスワード入力をロギングし、LOGINRETRIES パラメーターで指定されたログイン試行の許可回数をクライアントが超過したときを検出します。

許可

許可は、DB2 の機能を使用して実行されます。それぞれの許可名に関連する許可事項を記録するために、 DB2 表と構成ファイルが使用されます。

認証済みユーザーがデータにアクセスしようとすると、そのユーザーの許可名、そのユーザーが属しているグループの許可名、およびユーザーに直接付与されたロールや、またはグループあるいはロールを通して間接的に付与されたロールの許可名が、記録されている許可事項と比較されます。この比較に基づいて、DB2 サーバーは、要求されたアクセスを許すかどうかを判断します。

記録される許可事項のタイプは、「特権」、「権限レベル」、および「LBAC 信用証明情報」の 3 つです。

特権 は、 1 ユーザーがデータベース・リソースを作成またはアクセスできるようにするために、 1 つの許可名に対して単一の許可事項を定義します。特権は、データベース・カタログに保管されます。

権限レベル は、特権をグループ化する方法を提供し、より高いレベルで、データベース・マネージャーの保守とユーティリティー操作を制御します。データベース固有の権限は、データベース・カタログに保管されます。また、システム権限は、グループ・メンバーシップと関連付けられ、権限レベルに関連するグループ名は、特定のインスタンスについて、データベース・マネージャー構成ファイルの中に保管されます。

*LBAC 信用証明情報*は、ラベル・ベースのアクセス制御 (LBAC) によって保護されているデータへのアクセスを許可する LBAC セキュリティー・ラベルおよび LBAC 規則の免除です。 LBAC 信用証明情報は、データベース・カタログに保管されます。

グループは、それぞれのユーザーに個別に特権の付与または取り消しを行うことを必要とせずに、ユーザーの集合に対して許可を実行するための便利な手段を提供します。特に異なる指定がなければ、グループ許可名は、許可名が許可の目的で使用されるところであれば、どこでも使用することができます。一般に、グループ・メンバーシップは、動的 SQL およびデータベース以外のオブジェクト (インスタンス・レベルのコマンドおよびユーティリティーなど) の許可のためのものと考えられ、静的 SQL のためのものとは考えられません。この一般的なケースの例外は、特権が PUBLIC に与えられるときに生じ、この場合は静的 SQL が処理されるときに考慮されます。グループ・メンバーシップが適用されない特定のケースについては、DB2 の資料全体を通して、該当する場合にその旨の注が付いています。

ロールは、1 つ以上の特権をまとめたデータベース・オブジェクトですが、これを、GRANT ステートメントを使用してユーザー、グループ、PUBLIC、またはその他のロールに割り当てるか、あるいは、CREATE TRUSTED CONTEXT または ALTER TRUSTED CONTEXT ステートメントを使用して、トラステッド・コンテキストに割り当てることができます。ワークロード定義内で、SESSION_USER ROLE 接続属性用のロールを指定することができます。ロールの使用時には、データベース・オブジェクトに対するアクセス許可をそのロールに関連付けます。すると、そのロールのメンバーであるユーザーは、データベース・オブジェクトへのアクセス時に使用するロールに合わせて定義された特権を持つことになります。

ロールは、グループに似た機能を備えています。つまりロールは、各ユーザーごとに個別の特権の付与または取り消しを行うという手間をかけずに、ユーザーの集合に対して許可を実行します。ロールの利点の 1 つとして、これは、DB2 データベース・システムによって管理されます。ビュー、トリガー、マテリアライズ照会表 (MQT)、パッケージ、および SQL ルーチンの許可プロセスでは、グループに付与された許可とは違って、ロールに認可された許可は、検討の対象になります。ビュー、トリガー、MQT、パッケージ、および SQL ルーチンの許可プロセスでは、グループに付与された許可が検討の対象にならない理由は、DB2 データベース・システムは、グループ内でメンバーシップがいつ変更になったかを検出できないため、これらのオブジェクトを必要に応じて無効化できないからです。

注: ビュー、トリガー、MQT、パッケージ、および SQL ルーチンの許可プロセスでは、グループに付与されたロールに付与された許可は、検討の対象になりません。

SQL ステートメントの処理中に DB2 許可モデルが検討の対象とするのは、以下の一連の許可です。

1. その SQL ステートメントに関連付けられている 1 次許可 ID に付与された許可。
2. その SQL ステートメントに関連付けられている 1 次許可 ID に付与されたロールに付与された許可。
3. その SQL ステートメントに関連付けられている 2 次許可 ID (グループまたはロール) に付与された許可。

4. その SQL ステートメントに関連付けられている 2 次許可 ID (グループまたはロール) に付与されたロールに付与された許可。
5. PUBLIC に直接付与されたロールや、または他のロールを介して間接的に付与されたロールも含め、PUBLIC に付与された許可。
6. トラステッド・コンテキスト・ロールに付与された許可 (該当する場合)。

DB2 データベース・マネージャーのインストールおよび使用時のセキュリティー問題

セキュリティーに関する考慮事項は、製品がインストールされたときから、DB2 管理者にとって重要なことです。

DB2 データベース・マネージャーのインストールを完了するためには、ユーザー ID、グループ名、およびパスワードが必要です。 GUI ベースの DB2 データベース・マネージャーのインストール・プログラムは、さまざまなユーザー ID とグループのデフォルト値を作成します。 Linux および UNIX プラットフォームにインストールする場合と Windows® プラットフォームにインストールする場合とでは、作成されるデフォルト値が異なります。

- UNIX およびLinux® プラットフォームの場合、インスタンスのセットアップ・ウィンドウで DB2 インスタンスを作成することを選択すると、DB2 データベース・インストール・プログラムはデフォルトで、それぞれ異なるユーザーを DAS DAS (dasusr)、インスタンス所有者 (db2inst)、および fenced ユーザー (db2fenc) のために作成します。オプションとして、異なるユーザー名を指定することもできます。

  DB2 データベース・インストール・プログラムは、まだ存在していないユーザー ID を作成するため、デフォルト・ユーザー名に 1 から 99 までの数値を順番に付加します。たとえば、ユーザー db2inst1 および db2inst2 がすでに存在する場合、 DB2 データベース・インストール・プログラムはユーザー db2inst3 を作成します。 10 よりも大きな数値が使用される場合、デフォルト・ユーザー ID の名前の文字部分が切り捨てられます。たとえば、ユーザー ID db2fenc9 がすでに存在する場合、 DB2 データベース・インストール・プログラムはユーザー ID の c を切り捨てて、10 を付加します (db2fen10)。デフォルトの DAS ユーザーに数値が付加されるときに切り捨ては生じません (dasusr24 など)。
- Windows プラットフォームの場合、DB2 データベース・インストール・プログラムはデフォルトで、 DAS ユーザー、インスタンス所有者、および fenced ユーザーのためにユーザー db2admin を作成します (必要に応じて、これとは別のユーザー名をセットアップ時に指定することができます)。 Linux および UNIX プラットフォームの場合とは異なり、ユーザー ID に数値は付加されません。

管理者以外のユーザーがデフォルトを知って、データベースおよびインスタンス内で不適切な方式で使用するリスクを最小限にするには、インストールの際にデフォルトを新規または既存の任意のユーザー ID に変更します。

注: 応答ファイルのインストールでは、ユーザー ID またはグループ名にデフォルト値を使用しません。これらの値を、応答ファイルに指定しておく必要があります。

ユーザーを認証する際に、パスワードは非常に重要です。認証要件がオペレーティング・システム・レベルで設定されていないときに、データベースがオペレーティング・システムを使用してユーザーを認証する場合、ユーザーは接続を許可されます。例えば、Linux および UNIX オペレーティング・システムでは、未定義のパスワードは NULL として扱われます。この場合、定義されたパスワードを持っていないユーザーは、NULL パスワードを持っているものと見なされます。オペレーティング・システムの観点では、これが一致であり、ユーザーの妥当性検査が行われ、データベースに接続することができます。オペレーティング・システムがデータベースに対するユーザーの認証を行う場合は、オペレーティング・システム・レベルのパスワードを使用します。

Linux および UNIX オペレーティング・システム環境で、DB2 データ・パーティション化機能 (DPF) を操作する場合、リモート・ノードでコマンドを実行するために、DB2 データベース・マネージャーはデフォルトで rsh ユーティリティー (HP-UX では remsh) を使用します。 rsh ユーティリティーはネットワーク上で平文のパスワードを伝送するため、DB2 サーバーがセキュア・ネットワーク上にない場合には機密漏れが生じる可能性があります。DB2RSHCMD レジストリー変数を使用することにより、リモート・シェル・プログラムを、この機密漏れを防ぐ、より安全な別のものに設定できます。より安全な別のプログラムの一例として、ssh があります。リモート・シェル構成の制限については、DB2RSHCMD レジストリー変数の資料を参照してください。

DB2 データベース・マネージャーをインストールした後に、ユーザーに付与されたデフォルト特権の検討および (必要であれば) 変更も行ってください。デフォルトでは、インストール処理によってシステム管理 (SYSADM) の特権が、各オペレーティング・システム上で下記のユーザーに与えられます。

Windows 環境
管理者グループに属する有効な DB2 データベースのユーザー名。

Linux および UNIX プラットフォーム
インスタンス所有者の 1 次グループに属する有効な DB2 データベースのユーザー名。

SYSADM 特権は、DB2 データベース・マネージャーの中の使用可能な特権の最も強力なセットです。その結果、これらのユーザーのすべてがデフォルトによって、SYSADM 特権を持つことを望むことはできません。 DB2 データベース・マネージャーは、グループおよび個々のユーザー ID に特権を与えたり、取り消したりする能力を管理者に付与しています。

データベース・マネージャーの構成パラメーター *sysadm_group* を更新することにより、管理者はどのユーザーのグループが SYSADM 特権を持つようにするかを制御できます。 DB2 データベース・インストールとその後のインスタンスだけでなく、さらにデータベース作成のセキュリティー要件を完成するために、下記のガイドラインに従わなければなりません。

システム管理グループと定義される (*sysadm_group* を更新することにより) グループが、少なくとも 1 つは存在しなければなりません。このグループの名前を使用すれば、インスタンス所有者のために作成されたグループのように簡易識別することができます。このグループに属するユーザー ID およびグループは、それぞれのインスタンスに対してシステム管理者権限を持っています。

管理者は、特定インスタンスに関連付けられていると容易に認識される、インスタンス所有者ユーザー ID を作成することを考慮する必要があります。このユーザー ID は、そのグループの名前の 1 つとして上記で作成された SYSADM グループの名前を持つ必要があります。もう一つの方法としては、インスタンス所有者のユーザー ID をインスタンス所有者グループの一員としてだけに使用し、他のグループでは使用しないようにすることです。このようにすることによって、インスタンスまたはインスタンス内の任意のオブジェクトを変更できるユーザー ID およびグループの急増を制御できます。

作成されたユーザー ID は、インスタンス内のデータおよびデータベースへの入力を許可される前に認証を行うため、 1 つのパスワードと関連付ける必要があります。パスワード作成時には、組織のパスワード命名ガイドラインに従うことをお勧めします。

注: インスタンス構成ファイルまたはその他のファイルを誤って削除したり上書きしたりしてしまうことを防ぐために、管理者は、サーバー上で直接実行される日常の管理タスク用に、インスタンス所有者と同じ 1 次グループに属さない別のユーザー・アカウントを使用することを検討する必要があります。

インスタンス・ディレクトリーとデータベース・ディレクトリーのファイル許可要件

DB2 データベース・システムでは、インスタンス・ディレクトリーとデータベース・ディレクトリーに少なくとも以下の許可を設定する必要があります。

注: インスタンス・ディレクトリーとデータベース・ディレクトリーが DB2 データベース・マネージャーによって作成された場合は、許可が正確に設定されているので、変更するべきではありません。

UNIX マシンと Linux マシンのインスタンス・ディレクトリーと NODE000*x*/sqldbdir ディレクトリーの最小の許可は、`u=rwx` と `go=rx` でなければなりません。これらの文字の意味を以下の表で説明します。

文字	意味
u	ユーザー (所有者)
g	グループ
o	他のユーザー
r	読み取り
w	書き込み
x	実行

例えば、/home にあるインスタンス db2inst1 の許可は、以下のとおりです。

```
drwxr-xr-x  36 db2inst1  db2grp1         4096 Jun 15 11:13 db2inst1
```

データベースが組み込まれているディレクトリーについては、NODE000*x* までのあらゆるディレクトリー・レベル (このレベルも含む) で以下の許可が必要です。

```
drwxrwxr-x  11 db2inst1  db2grp1         4096 Jun 14 15:53 NODE0000/
```

例えば、データベースが /db2/data/db2inst1/db2inst1/NODE0000 にあれば、ディレクトリー /db2、/db2/data、/db2/data/db2inst1、/db2/data/db2inst1/db2inst1、/db2/data/db2inst1/db2inst1/NODE0000 で drwxrwxr-x が必要になります。

NODE000x ディレクトリーの中では、sqldbdir ディレクトリーで許可 drwxrwxr-x が必要です。例えば、次のようになります。

```
drwx------   5 db2inst1  db2grp1         256 Jun 14 14:17 SAMPLE/
drwxr-x---   7 db2inst1  db2grp1        4096 Jun 14 13:26 SQL00001/
drwxrwxr-x   2 db2inst1  db2grp1         256 Jun 14 13:02 sqldbdir/
```

注意:
ファイルのセキュリティーを維持するために、*DBNAME* ディレクトリー (SAMPLE など) と SQL*xxxx* ディレクトリーの許可は、DB2 データベース・マネージャーによってそれらのディレクトリーが作成されたときに割り当てられた許可から変更しないでください。

認証の詳細

サーバーでの認証方式

インスタンスまたはデータベースにアクセスするためには、まず、そのユーザーが *認証* されていることが必要です。各インスタンスの *認証タイプ* によって、ユーザーを検査する方法と場所が決まります。認証タイプは、サーバーの構成ファイルに保管されます。認証タイプは、インスタンスの作成時に初期設定されます。インスタンスごとに 1 つの認証タイプがあり、それが、そのデータベース・サーバーおよびその制御下のすべてのデータベースのアクセスをカバーしています。

フェデレーテッド・データベースからデータ・ソースにアクセスする場合、データ・ソース認証処理およびフェデレーテッド認証タイプの定義を考慮する必要があります。

注: SERVER_ENCRYPT 認証の使用時にユーザー ID とパスワードを暗号化するため、および DATA_ENCRYPT 認証の使用時にユーザー ID、パスワード、およびユーザー・データを暗号化するために、DB2 データベース管理システムによって使用される暗号ルーチンに関する証明情報について、次の Web サイトをチェックすることができます。http://www.ibm.com/security/standards/st_evaluations.shtml。

明示的トラステッド接続でのユーザーの切り替え

CLI/ODBC および XA CLI/ODBC アプリケーションでは、認証が必要となるユーザー切り替え要求を処理する時に使用される認証メカニズムは、最初にトラステッド接続自体を確立するために使用されるメカニズムと同じです。そのため、明示的トラステッド接続の確立中に使用された他のすべての折衝されたセキュリティー属性 (暗号化アルゴリズム、暗号鍵、プラグイン名など) は、そのトラステッド接続でのユーザー切り替え要求に必要なすべての認証の場合と同じと見なされます。 JAVA アプリケーションでは、ユーザー切り替え要求における認証方式を (データ・ソース・プロパティーを使用することにより) 変更することができます。

トラステッド・コンテキスト・オブジェクトは、トラステッド接続でのユーザーの切り替えに認証を *必要としない* ように定義できるため、明示的トラステッド接続で

のユーザー切り替えフィーチャーを十分に活用するためには、ユーザー作成のセキュリティー・プラグインで以下の操作ができなければなりません。

- ユーザー ID のみのトークンを受け入れる
- そのユーザー ID の有効な DB2 許可 ID を戻す

注: CLIENT タイプの認証が有効である場合には、明示的トラステッド接続は確立できません。

指定された認証タイプ

以下の認証タイプがあります。

SERVER

その構成に有効なセキュリティー・メカニズム (例えば、セキュリティー・プラグイン・モジュール) を使用して、サーバー上で認証が行われることを指定します。デフォルトのセキュリティー・メカニズムは、接続の試行中にユーザー ID およびパスワードが指定されると、それらがサーバーにある有効なユーザー ID とパスワードの組み合わせと比較され、そのユーザーがそのインスタンスへのアクセスを許されているかどうかが判別されるというものです。

注: サーバー・コードは、接続がローカルなのかリモートなのかを検出します。ローカル接続の場合、認証が SERVER であると、ユーザー ID とパスワードは、認証の成功のためには必要とされません。

SERVER_ENCRYPT

サーバーが、暗号化された SERVER 認証スキーマを受け入れるように指定します。クライアント認証が指定されない場合、クライアントはサーバーで選択された方式を使用して認証されます。

CLIENT

オペレーティング・システムのセキュリティーを使用して、アプリケーションが呼び出されたデータベース・パーティション上で認証が行われることを指定します。接続の試行中にユーザー ID およびパスワードが指定されると、それらがクライアント・ノードにある有効なユーザー ID とパスワードの組み合わせと比較され、そのユーザー ID がそのインスタンスへのアクセスを許されているかどうかが判別されます。データベース・サーバーでは、それ以上の認証は行われません。これはしばしば、シングル・サインオンと呼ばれます。

ユーザーがローカルまたはクライアントのログインを行った場合、そのユーザーは、そのローカルのクライアント・ワークステーションでのみ認識されます。

リモート・インスタンスが CLIENT 認証である場合、 *trust_allclnts* と *trust_clntauth* という他の 2 つのパラメーターが最終的な認証タイプを決定します。

TRUSTED クライアントのみに対する CLIENT レベルのセキュリティー

トラステッド・クライアントとは、信頼できるローカル・セキュリティー・システムをもつクライアントのことです。

CLIENT の認証タイプが選択されている場合、固有のセキュリティーをオペレーティング環境が持っていないクライアントに対する保護のために、追加のオプションを選択することができます。

非セキュアのクライアントに対する保護のために、管理者は、*trust_allclnts* パラメーターを NO に設定することによって、「トラステッド・クライアント認証」を選択することができます。これは、すべてのトラステッド・プラットフォームが、サーバーに代わってユーザーの認証ができることを意味します。非トラステッド・クライアントは、サーバー上で認証され、ユーザー ID とパスワードを提供しなければなりません。ユーザーは、クライアントを信頼するかどうかを示すために、 *trust_allclnts* 構成パラメーターを使用します。このパラメーターのデフォルトは YES です。

注: 一部のクライアントが認証のための安全な固有のセキュリティー・システムを持っていない場合であっても、すべてのクライアントをトラステッド・クライアント (*trust_allclnts* が YES) とすることは可能です。

トラステッド・クライアントの場合であっても、サーバー側で認証を完了させたい場合があります。トラステッド・クライアントをどこで妥当性検査するかを指示するために、 *trust_clntauth* 構成パラメーターを使用します。このパラメーターのデフォルトは CLIENT です。

注: トラステッド・クライアントの場合のみ、 CONNECT または ATTACH を試みているときにユーザー ID またはパスワードが明示的に提供されないと、ユーザーの妥当性検査は、そのクライアントで行われます。
trust_clntauth パラメーターは、 USER または USING 節で提供された情報をどこで妥当性検査するかを判別するためだけに使用されます。

OS/390® および z/OS® 上の DB2、VM および VSE 上の DB2、System i™ 上の DB2 からの DRDA® クライアントを除く、すべてのクライアントから保護するためには、*trust_allclnts* パラメーターを DRDAONLY に設定します。上記のクライアントだけを、クライアント側の認証を行うよう承認することができます。他のすべてのクライアントには、サーバーによって認証されているユーザー ID とパスワードが必要です。

trust_clntauth パラメーターは、上記のクライアントが認証される位置を判別するのに使用されます。 *trust_clntauth* が ″client″ である場合、認証はクライアントで行われます。 *trust_clntauth* を ″server″ に設定すると、認証は、クライアント (ユーザー ID およびパスワードが指定されなかった場合) およびサーバー (ユーザー ID およびパスワードが指定された場合) で行われます。

表 1. TRUST_ALLCLNTS および TRUST_CLNTAUTH パラメーターの組み合わせを使用した認証モード

TRUST_ ALLCLNTS	**TRUST_ CLNTAUTH**	**非トラステッドである非 DRDA クライアント認証、ユーザー ID およびパスワードなし**	**非トラステッドである非 DRDA クライアント認証、ユーザー ID およびパスワードあり**	**トラステッドである非 DRDA クライアント認証、ユーザー ID およびパスワードなし**	**トラステッドである非 DRDA クライアント認証、ユーザー ID およびパスワードあり**	**DRDA クライアント認証、ユーザー ID およびパスワードなし**	**DRDA クライアント認証、ユーザー ID およびパスワードあり**
YES	CLIENT	CLIENT	CLIENT	CLIENT	CLIENT	CLIENT	CLIENT

表 1. TRUST_ALLCLNTS および TRUST_CLNTAUTH パラメーターの組み合わせを使用した認証モード (続き)

TRUST_ ALLCLNTS	TRUST_ CLNTAUTH	非トラステッドである非 DRDA クライアント認証、ユーザー ID およびパスワードなし	非トラステッドである非 DRDA クライアント認証、ユーザー ID およびパスワードあり	トラステッドである非 DRDA クライアント認証、ユーザー ID およびパスワードなし	トラステッドである非 DRDA クライアント認証、ユーザー ID およびパスワードあり	DRDA クライアント認証、ユーザー ID およびパスワードなし	DRDA クライアント認証、ユーザー ID およびパスワードあり
YES	SERVER	CLIENT	SERVER	CLIENT	SERVER	CLIENT	SERVER
NO	CLIENT	SERVER	SERVER	CLIENT	CLIENT	CLIENT	CLIENT
NO	SERVER	SERVER	SERVER	CLIENT	SERVER	CLIENT	SERVER
DRDAONLY	CLIENT	SERVER	SERVER	SERVER	SERVER	CLIENT	CLIENT
DRDAONLY	SERVER	SERVER	SERVER	SERVER	SERVER	CLIENT	SERVER

KERBEROS

DB2 クライアントとサーバーが両方とも、 Kerberos セキュリティー・プロトコルをサポートしているオペレーティング・システム上で実行されている場合に使用します。 Kerberos セキュリティー・プロトコルは、従来の暗号を使用して共有秘密鍵を作成することにより、サード・パーティーの認証サービスとして認証を実行します。この鍵がユーザーの証明書になり、ローカルまたはネットワーク・サービスが要求されるたびに、ユーザーの身元の確認に使用されます。この鍵を使用することにより、ネットワークを介して平文でユーザー名およびパスワードを渡す必要がなくなります。 Kerberos セキュリティー・プロトコルにより、リモート DB2 データベース・サーバーへのシングル・サインオンを行えるようになります。 KERBEROS 認証タイプはさまざまなオペレーティング・システムでサポートされています。詳しい情報については、関連情報セクションを参照してください。

Kerberos 認証は、以下のように行われます。

1. ドメイン・アカウントを使用してクライアント・マシンにログオンしているユーザーは、ドメイン・コントローラーの Kerberos 鍵配布センター (KDC) へ認証を行います。鍵配布センターはチケット許可チケット (TGT) をクライアントに発行します。
2. 接続の最初の段階で、サーバーはクライアントにターゲット・プリンシパル名を送信します。これは、DB2 データベース・サーバー・サービスのサービス・アカウント名です。サーバーのターゲット・プリンシパル名、およびチケット許可チケットを使用して、クライアントはチケット許可サービス (TGS) (これもドメイン・コントローラーにあります) へサービス・チケットを要求します。クライアントのチケット許可チケット、およびサーバーのターゲット・プリンシパル名の両方が有効であれば、 TGS はクライアントへサービス・チケットを発行します。データベース・ディレクトリーに記録されるプリンシパル名は、name/instance@REALM の形式で指定できます。 (これは、Windows で指定可能な DOMAIN¥userID および userID@xxx.xxx.xxx.com フォーマットに追加されるものです。)

3. クライアントはこのサービス・チケットを通信チャネル (例えば、TCP/IP を使用してサーバーへ送信します。
4. サーバーはクライアントのサーバー・チケットの妥当性検査を行います。クライアントのサービス・チケットが有効であれば、これで認証は完了します。

クライアント・マシンでデータベースをカタログし、サーバーのターゲット・プリンシパル名と共に Kerberos 認証タイプを明示的に指定することも可能です。そうすれば、接続の最初の段階はバイパスすることができます。

ユーザー ID およびパスワードが指定されている場合、クライアントはそのユーザー・アカウントに対するチケット許可チケットを要求し、それを認証に使用します。

KRB_SERVER_ENCRYPT

サーバーが、KERBEROS 認証または暗号化された SERVER 認証スキーマを受け入れるように指定します。クライアント認証が KERBEROS である場合、クライアントは Kerberos セキュリティー・システムを使用して認証されます。クライアント認証が SERVER_ENCRYPT である場合、クライアントはユーザー ID および暗号化パスワードを使用して認証されます。クライアント認証が指定されない場合、クライアントは Kerberos が使用可能であればそれを使用し、それが使用可能でなければパスワード暗号化を使用して認証されます。その他のクライアント認証タイプでは、認証エラーが戻されます。クライアントの認証タイプを KRB_SERVER_ENCRYPT として指定することはできません。

注: Kerberos 認証タイプは特定のオペレーティング・システムで実行されているクライアントおよびサーバーでサポートされています。詳しい情報については、関連情報セクションを参照してください。Windows オペレーティング・システムでは、クライアントおよびサーバー・マシンは両方とも同じ Windows ドメインに属しているか、またはトラステッド・ドメインに属していなければなりません。この認証タイプは、サーバーが Kerberos をサポートしており、クライアント・マシンのいくつか (すべてである必要はありません) が Kerberos 認証をサポートしている場合に使用してください。

DATA_ENCRYPT

サーバーは、暗号化された SERVER 認証スキーマおよびユーザー・データの暗号化を受け入れます。認証が機能する方法は、SERVER_ENCRYPT に関して示した方法と同じです。詳しくは、その認証タイプを参照してください。

この認証タイプを使用するとき、以下のユーザー・データが暗号化されます。

- SQL および XQuery ステートメント
- SQL プログラム変数データ
- SQL または XQuery ステートメントのサーバー処理の出力データで、データについての説明を含むもの
- 照会から生じる応答セット・データの一部またはすべて
- ラージ・オブジェクト (LOB) データ・ストリーム
- SQLDA 記述子

DATA_ENCRYPT_CMP

サーバーは、暗号化された SERVER 認証スキーマおよびユーザー・データの暗号化を受け入れます。さらに、この認証タイプは DATA_ENCRYPT 認証タイプをサポートしていない下位レベルの製品との互換性を可能にします。これらの製品は、SERVER_ENCRYPT 認証タイプを使って、暗号化ユーザー・データがない状態での接続を許可されます。新しい認証タイプをサポートしている製品は、これを使用する必要があります。この認証タイプは、サーバーのデータベース・マネージャー構成ファイル内のみで有効であり、CATALOG DATABASE コマンドで使用するときには無効です。

GSSPLUGIN

サーバーが認証を行うために GSS-API プラグインを使用するように指定します。クライアント認証が指定されていない場合、サーバーは *srvcon_gssplugin_list* データベース・マネージャー構成パラメーターにリストされている Kerberos プラグインを含む、サーバーによってサポートされているプラグインのリストをクライアントに戻します。クライアントは、クライアント・プラグイン・ディレクトリーにある最初のプラグインをリストから選択します。クライアントがリスト内のどのプラグインもサポートしない場合、そのクライアントは Kerberos 認証方式 (それが戻される場合) を使用して認証されます。クライアント認証が GSSPLUGIN 認証方式の場合、クライアントはリスト内にあるサポートされる最初のプラグインを使用して認証されます。

GSS_SERVER_ENCRYPT

サーバーが、プラグイン認証または暗号化されたサーバー認証スキーマを受け入れるように指定します。クライアント認証がプラグインを介して行われる場合、クライアントはサーバーがサポートするプラグインのリストにある、クライアントがサポートする最初のプラグインを使用して認証されます。

クライアント認証が指定されないで暗黙的接続が行われる場合 (つまり、接続が確立されるときにクライアントがユーザー ID とパスワードを供給しない場合)、サーバーはサーバーがサポートするプラグインのリスト、Kerberos 認証方式 (リスト内のプラグインの 1 つが Kerberos に基づくものである場合)、および暗号化サーバー認証方式を戻します。クライアントは、クライアント・プラグイン・ディレクトリーにある、最初にサポートされているプラグインを使用して認証されます。クライアントがリスト内のどのプラグインもサポートしない場合、そのクライアントは Kerberos 認証方式を使用して認証されます。クライアントが Kerberos 認証をサポートしない場合、そのクライアントは暗号化サーバー認証方式を使用して認証され、パスワードがないために接続が失敗します。クライアントは、 DB2 が提供する Kerberos プラグインがオペレーティング・システムに対して存在するか、または Kerberos ベースのプラグインが *srvcon_gssplugin_list* データベース・マネージャー構成パラメーターに指定されている場合に、Kerberos 認証方式をサポートします。

クライアント認証を指定しないで明示接続が実行されている場合 (つまり、ユーザー ID とパスワードの両方が供給されている場合)、認証タイプは SERVER_ENCRYPT と等しくなります。

注:

1. 構成ファイル自体へのアクセスは構成ファイル内の情報によって保護されているため、認証情報を変更しているときに、誤って自分自身を自分のインスタンスからロックアウトしてしまわないようにしてください。以下のデータベース・マネージャー構成ファイル・パラメーターは、インスタンスへのアクセスを制御します。
   - AUTHENTICATION *
   - SYSADM_GROUP *
   - TRUST_ALLCLNTS
   - TRUST_CLNTAUTH
   - SYSCTRL_GROUP
   - SYSMAINT_GROUP

   * は、2 つの最も重要なパラメーターを示し、これらが最も問題を引き起こす可能性があります。

   このようなことが起こらないようにするために、行えることがいくつかあります。誤って自分自身を DB2 データベース・システムからロックアウトしてしまった場合、すべてのプラットフォームで使用可能なフェイルセーフのオプションがあります。これは、高い特権をもったローカルのオペレーティング・システムのセキュリティー・ユーザーを使用して、通常の DB2 データベース・セキュリティー検査をオーバーライドしてデータベース・マネージャー構成ファイルを更新することです。このユーザーは、常にデータベース・マネージャー構成ファイルを更新するための特権を持っており、それによって問題を訂正します。ただし、このセキュリティー上の迂回は、データベース・マネージャー構成ファイルのローカル更新にのみ制限されています。フェイルセーフのためのユーザーは、リモートで、または他の DB2 データベース・コマンドに対して使用することはできません。この特別のユーザーは、以下のように識別されます。
   - UNIX プラットフォームの場合: インスタンス所有者
   - Windows プラットフォームの場合: ローカル「管理者」グループに属する人
   - その他のプラットフォームの場合: その他のプラットフォーム上ではローカル・セキュリティーがないため、すべてのユーザーがローカル・セキュリティー検査に合格します。

リモート・クライアントの認証に関する考慮事項

リモート・アクセスのためにデータベースをカタログする場合、認証タイプをデータベース・ディレクトリー項目の中に指定することができます。

認証タイプは必須ではありません。指定されない場合、クライアントはデフォルトの SERVER_ENCRYPT になります。ただし、サーバーが SERVER_ENCRYPT をサポートしていない場合は、クライアントはサーバーのサポートしている値を使用して再試行を行います。サーバーが複数の認証タイプをサポートしている場合は、クライアントはそれらの中から選択せずにエラーを戻します。エラーを戻すのは、正しい認証タイプが確実に使用されるようにするためです。この場合、クライアントはサポートされている認証タイプを使用してデータベースをカタログしなければなりません。認証タイプが指定されると、指定された値がサーバー側の値と一致した場合に、認証は即時に開始できます。不一致が検出された場合、DB2 データベースはリカバリーを試行します。リカバリーにより、相違を調整するためにさらに多く

のフローが実行されるか、または DB2 データベースがリカバリーできなければエラーになります。不一致がある場合は、サーバーにある値の方が正しいと見なされます。

認証タイプ DATA_ENCRYPT_CMP により、データ暗号化をサポートしない、前のリリースのクライアントが、DATA_ENCRYPT でなく SERVER_ENCRYPT 認証を使用するサーバーに接続できるようになります。この認証は、以下の条件が当てはまる場合は機能しません。

- クライアント・レベルがバージョン 7.2。
- ゲートウェイ・レベルがバージョン 8 フィックスパック 7 以降。
- サーバーがバージョン 8 フィックスパック 7 以降。

上記がすべて当てはまる場合、クライアントはサーバーに接続できません。接続できるようにするには、クライアントをバージョン 8 にアップグレードするか、またはバージョン 8 フィックスパック 6 以前のゲートウェイ・レベルを使用する必要があります。

接続時に使用される認証タイプは、ゲートウェイのデータベース・カタログ項目として適切な認証タイプを指定することによって決定されます。これは、DB2 Connect™ のシナリオと、パーティション・データベース環境 (クライアントの DB2NODE レジストリー変数が設定されている) のクライアント/サーバーのどちらにも当てはまります。適切なパーティションに「ホップ」する目的で、認証タイプをカタログ・パーティションでカタログします。このシナリオの場合、ネゴシエーションはもっぱらクライアント/サーバー間で行われるので、ゲートウェイでカタログされる認証タイプは使用されません。

認証タイプの異なるクライアントを必要とする場合には、異なる認証タイプを使って複数のデータベース別名をゲートウェイでカタログする必要が生じるかもしれません。ゲートウェイでカタログする認証タイプを決める際には、その認証タイプを、クライアントおよびサーバーで使用されるものと同じにすることができます。あるいは、NOTSPEC がデフォルトの SERVER になることを理解した上で、NOTSPEC 認証タイプを使用することもできます。

パーティション・データベースの認証に関する考慮事項

パーティション・データベースでは、データベースの各区画に、同じ組のユーザーとグループが定義されていなければなりません。定義が同じでないと、ユーザーは、異なる区画で異なることを実行できるように許可されてしまうことがあります。すべての区画にわたって一貫していることが推奨されます。

Kerberos 認証についての詳細

DB2 データベース・システムは、AIX®、Solaris、Linux IA32 および AMD64、および Windows オペレーティング・システム上での Kerberos 認証プロトコルのサポートを提供します。

Kerberos サポートは、サーバーおよびクライアント認証プラグインとして使用される「IBMkrb5」という名前の GSS-API セキュリティー・プラグインとして提供されます。このライブラリーが置かれている場所は、 UNIX および Linux では

sqllib/security{32|64}/plugin/IBM/{client|server} ディレクトリー、 Windows では sqllib/security/plugin/IBM{client|server} ディレクトリーです。

注: 64 ビット Windows の場合、プラグイン・ライブラリーの名前は IBMkrb564.dll です。さらに、UNIX および Linux プラグインの実際のプラグイン・ソース・コード IBMkrb5.C が sqllib/samples/security/plugins ディレクトリーにあります。

Kerberos 認証を DB2 データベースでご使用になる前に、 Kerberos の使用方法と構成方法を十分に理解しておくことをお勧めします。

Kerberos の概要

Kerberos はサード・パーティーのネットワーク認証プロトコルで、無保護のネットワーク環境の中でユーザーを安全に認証するために、共有秘密鍵認証を使用します。 3 層システムが使用され、 Kerberos 鍵配布センター (KDC) という別個のサーバーが提供する暗号化されたチケットが、アプリケーション・サーバーとクライアントの間で交換されます (テキスト形式のユーザー ID とパスワードは交換されません)。このような暗号化されたサービス・チケット (*証明書* という) は存続期間が有限で、クライアントとサーバーしかこれを理解できません。これによって、たとえチケットがネットワークから傍受された場合でも、セキュリティー上のリスクが軽減されます。各ユーザー (Kerberos では*プリンシパル* という) は、 KDC によって共有される暗号化された秘密鍵を所有します。 1 つの KDC に登録されたプリンシパルとコンピューターは、集合的に*レルム* と呼ばれます。

Kerberos の主な特徴であるシングル・サインオン環境では、各ユーザーが Kerberos レルム内のリソースに対して自分の身元を 1 度だけ検証します。これは、DB2 データベースにおいて、ユーザーがユーザー ID およびパスワードを提供しなくても DB2 データベース・サーバーに接続またはアタッチできるということです。もう 1 つの利点として、プリンシパル用の中心的なリポジトリーを 1 つだけ使用するため、ユーザー ID 管理が単純化されます。さらに、Kerberos は相互認証をサポートするため、クライアントはサーバーの身元を検証することができます。

Kerberos セットアップ

DB2 データベース・システムで Kerberos をサポートするには、 DB2 データベースを設定する前に、参加するすべてのマシンで Kerberos をインストールして正しく構成しなければなりません。これには、以下の要件が含まれます (この他にも要件が存在する可能性があります)。

1. クライアント・マシン、サーバー・マシン、およびプリンシパルがすべて同じレルムに所属するか、複数のトラステッド・レルム (Windows 用語では信頼されるドメインという) に属する必要がある
2. 適切なプリンシパルを作成する
3. 該当する場合、サーバー keytab ファイルを作成する
4. 参加するすべてのマシンのシステム・クロックを同期する必要があります。 (通常、Kerberos では 5 分間のスキューが許容されます。そうしないと、証明書を取得するときに事前認証エラーが発生する可能性があります。)

Kerberos のインストールと構成についての詳細は、インストールする Kerberos 製品の資料を参照してください。

DB2 データベース・システムで必要な設定は、接続するアプリケーションが提供する証明書に基づいて、 Kerberos セキュリティー・コンテキストを正常に作成することだけです (つまり認証)。その他の Kerberos 機能 (署名、メッセージ暗号化など) は、使用されません。さらに、可能な場合には、相互認証がサポートされます。

Kerberos の前提条件は以下のとおりです。

- AIX、Solaris オペレーティング環境、および Linux プラットフォームでは、IBM® Network Authentication Service (NAS) Toolkit v1.4 以上が必要です。NAS Toolkits はこちらからダウンロードできます。https://www6.software.ibm.com/dl/dm/dm-nas-p
- Windows プラットフォームでは、前提条件はありません。

Kerberos とクライアント・プリンシパル

プリンシパルは、2 つまたは 3 つの部分から成るフォーマットに従います (つまり、 *name@REALM* または *name/instance@REALM*)。「name」の部分は許可 ID (AUTHID) マッピングに使用されるため、この名前は DB2 データベースの命名規則に従う必要があります。つまり、名前の長さは最大 30 文字で、使用可能な文字に関する現在の制限に従う必要があります。 (AUTHID マッピングについては、後で説明します。)

注: Windows では、 Kerberos プリンシパルがドメイン・ユーザーに直接関連付けられます。このため、ドメインまたはレルムに関連付けられていない Windows マシンでは、 Kerberos 認証を使用できません。さらに、Windows では 2 部からなる名前 (つまり *name@domain*) だけがサポートされます。

プリンシパル自体は、ターゲット・データベースへのサービス・チケットの要求および受信に使用されるアウトバウンド証明書を取得できなければなりません。取得するには、通常、 UNIX または Linux では kinit コマンドを使用します。Windows ではログオン時に暗黙的に取得されます。

Kerberos と許可 ID マッピング

オペレーティング・システムのユーザー ID の有効範囲が 1 つのマシンに限定されるのとは異なり、Kerberos プリンシパルは自身のレルム以外のレルムでも認証可能です。プリンシパルにレルム名を付けて完全修飾することにより、プリンシパル名の重複の問題を避けることができます。 Kerberos では、完全修飾されたプリンシパルの形式は name/instance@REALM です。実際には、区切り文字 ″/″ を使って instance フィールドに複数のインスタンスを含めることができます (たとえば、name/instance1/instance2@REALM)。または、instance フィールドを省略することもできます。明確な制限として、レルム名は、ネットワーク内に定義されたすべてのレルムの中で固有でなければなりません。 DB2 データベースにとって問題となるのは、プリンシパルから AUTHID への単純なマッピングを実現するために、プリンシパル名 (完全修飾されたプリンシパルの中の「name」) と AUTHID の間で 1 対 1 のマッピングが望ましいことです。DB2 データベースでは AUTHID がデフォルト・スキーマとして使用され、簡単かつ論理的に派生する必要があるので、単純なマッピングが必要とされます。このため、データベース管理者は、以下の問題が発生する可能性があることに注意しなければなりません。

- 異なるレルムに属する同じ名前の複数のプリンシパルは、同じ AUTHID にマップされます。
- 同じ名前でインスタンスが異なる複数のプリンシパルは、同じ AUTHID にマップされます。

この点を考慮して、以下のようにすることをお勧めします。

- DB2 データベース・サーバーにアクセスするすべてのトラステッド・レルム内の名前用として、1 つの固有のネーム・スペースを維持する
- 同じ名前のすべてのプリンシパルを、インスタンスにかかわらず、同じユーザーに所属させる

Kerberos とサーバー・プリンシパル

UNIX および Linux では、DB2 データベース・インスタンスのサーバー・プリンシパル名は <インスタンス名>/<完全修飾ホスト名>@REALM と想定されます。このプリンシパルは Kerberos セキュリティー・コンテキストを受け入れる必要があり、DB2 データベース・インスタンスの開始前にすでに存在している必要があります。これは、初期化の際にサーバー名がプラグインによって DB2 データベースに報告されるためです。

Windows では、サーバー・プリンシパルは、 DB2 データベース・サービスが開始されたドメイン・アカウントであると想定されます。ただし、ローカル SYSTEM アカウントによってインスタンスが開始される場合は例外です。この場合、サーバー・プリンシパル名は host/<hostname> と報告されます。これは、クライアントとサーバーの両方が Windows ドメインに属している場合にのみ有効です。

Windows では、2 つより多い部分からなる名前はサポートされません。このため、Windows クライアントが UNIX サーバーに接続しようとするときに問題が発生する可能性があります。したがって、UNIX Kerberos との相互運用が必要な場合、Kerberos プリンシパルから Windows アカウントへのマッピングを Windows ドメイン内で設定する必要があるかもしれません。 (関連情報については、該当する Microsoft® 資料を参照してください。)

UNIX および Linux オペレーティング・システム上の DB2 サーバーで使用される Kerberos サーバーのプリンシパル名はオーバーライド可能です。DB2_KRB5_PRINCIPAL 環境変数を、該当の完全修飾サーバー・プリンシパル名に設定します。サーバー・プリンシパル名は **db2start** を実行してからでなければ DB2 データベース・システムによって認識されないため、インスタンスを再始動する必要があります。

Kerberos keytab ファイル

セキュリティー・コンテキスト要求を受け入れる UNIX または Linux 上のすべての Kerberos サービスは、証明書を *keytab* (鍵テーブル) ファイル内に格納する必要があります。これは、DB2 データベースによってサーバー・プリンシパルとして使用されるプリンシパルに当てはまります。デフォルト keytab ファイルでのみ、サーバーの鍵が検索されます。 keytab ファイルに鍵を追加する方法については、Kerberos 製品に付属の資料を参照してください。

Windows には keytab ファイルの概念がなく、システムがプリンシパルの証明書の保管および獲得を自動的に処理します。

Kerberos とグループ

Kerberos 認証プロトコルには、グループ化の概念がありません。このため、DB2 データベースは Kerberos プリンシパルのグループ・リストを取得するためにローカル・オペレーティング・システムに依存します。 UNIX または Linux の場合、各プリンシパルごとに同等のシステム・アカウントが存在しなければなりません。たとえば、プリンシパルが name@REALM である場合、DB2 データベースは、オペレーティング・システム・ユーザー *name* が属するすべてのグループ名をローカル・オペレーティング・システムに対して照会することにより、グループ情報を集めます。 オペレーティング・システム・ユーザーが存在しない場合、 AUTHID は PUBLIC グループにのみ所属します。一方、Windows では、ドメイン・アカウントが Kerberos プリンシパルに自動的に関連付けられるため、別のオペレーティング・システム・アカウントを作成するための追加のステップは必要ありません。

クライアントでの Kerberos 認証の使用可能化

データベース・マネージャー構成パラメーター *clnt_krb_plugin* を、使用する Kerberos プラグインの名前に更新する必要があります。サポートされるプラットフォームでは、これを IBMkrb5 に設定してください。このパラメーターは、DB2 データベースに対して、 AUTHENTICATION パラメーターが KERBEROS または KRB_SERVER_ENCRYPT に設定されていれば、DB2 データベースで接続およびローカル・インスタンス・レベルのアクションに Kerberos を使用できることを通知します。そうでない場合、クライアント・サイドの Kerberos サポートは想定されません。

注: Kerberos サポートが利用可能かどうかの検証は実行されません。

オプションで、クライアント上でデータベースのカタログを作成する場合には、認証タイプを以下のように指定できます。

```
db2 catalog db testdb at node testnode authentication kerberos target
   principal service/host@REALM
```

ただし、認証情報が提供されない場合、サーバーはサーバー・プリンシパルの名前をクライアントに送ります。

サーバーでの Kerberos 認証の使用可能化

データベース・マネージャー構成パラメーター *srvcon_gssplugin_list* を、サーバー Kerberos プラグイン名に更新する必要があります。複数のサポートされるプラグインからなるリストをこのパラメーターに含めることもできますが、 Kerberos プラグインは 1 つだけ指定してください。ただし、このフィールドがブランクで、AUTHENTICATION が KERBEROS または KRB_SERVER_ENCRYPT に設定されている場合には、デフォルト Kerberos プラグイン (IBMkrb5) が想定され、使用されます。 Kerberos 認証をすべての操作で使用するか、それとも着信接続だけで使用するかに応じて、パラメーター AUTHENTICATION または SVRCON_AUTH を KERBEROS または KRB_SERVER_ENCRYPT に設定してください。

Kerberos プラグインの作成

Kerberos プラグインを作成するとき、以下の点を考慮する必要があります。

- GSS-API プラグインとして Kerberos プラグインを作成します。例外として、初期化関数で DB2 データベースに戻される関数ポインター配列内の *plugintype* は、 DB2SEC_PLUGIN_TYPE_KERBEROS に設定されなければなりません。
- 特定の条件のもとでは、サーバーがサーバー・プリンシパル名をクライアントに報告する場合があります。このため、プリンシパル名を GSS_C_NT_HOSTBASED_SERVICE 形式 (service@host) で指定しないでください。 DRDA では、プリンシパル名が GSS_C_NT_USER_NAME 形式 (server/host@REALM) でなければならないためです。
- 通常は、KRB5_KTNAME 環境変数によってデフォルト keytab ファイルを指定することが可能です。ただし、サーバー・プラグインは DB2 データベース・エンジン・プロセス内で実行されるため、この環境変数にアクセスできない可能性があります。

zSeries® とSystem i の互換性

zSeries およびSystem i に接続するには、AUTHENTICATION KERBEROS パラメーターを使ってデータベースをカタログする必要があり、TARGET PRINCIPAL パラメーター名を明示的に指定しなければなりません。

zSeries および System i はどちらも相互認証をサポートしません

Windows での問題

Windows プラットフォーム上で Kerberos を使用する場合には、次の問題点を認識しておく必要があります。

- Windows によるエラーの検出方式および報告方式のために、次の条件が存在すると予期しないクライアント・セキュリティー・プラグイン・エラー (SQL30082N、rc=36) が発生します。
  - アカウントの有効期限が切れている
  - パスワードが無効である
  - パスワードの有効期限が切れている
  - パスワードが管理者によって強制的に変更された
  - アカウントが使用不可である

  さらに、DB2 管理ログまたは db2diag.log で常に「ログオンに失敗 (Logon failed)」または「ログオン拒否 (Logon denied)」が指摘されます。
- ドメイン・アカウント名がローカルで定義される場合、接続においてドメイン名とパスワードが明示的に指定されていると、次のエラーにより失敗します。 `The Local Security Authority cannot be contacted`。

  このエラーは、Windows がローカル・ユーザーを最初に位置決めするために生じるものです。これは接続ストリングで完全修飾ユーザー名を指定することにより解決されます。たとえば、`name@DOMAIN.IBM.COM` のように指定します。
- Windows アカウントの名前には @ 文字を含めることができません。これは DB2 Kerberos プラグインによってドメイン区切り文字と見なされます。

- Windows 以外のプラットフォームと相互運用する場合には、Windows ドメイン・サーバー・アカウントと Windows クライアント・アカウントのすべてが DES 暗号化を使用するように構成します。 DB2 サービスの開始で使用されるアカウントが DES 暗号化を使用するように構成されないと、DB2 サーバーは Kerberos コンテキストを受け取れません。特に、DB2 は予期しないサーバー・プラグイン・エラーによって失敗し、AcceptSecurityContext API が SEC_I_CONTINUE_NEEDED (0x00090312L) を戻したことをログに記録します。

  Windows アカウントが DES 暗号化を使用するように構成されているかどうかを判別するには、「**アカウント・プロパティー (Account properties)**」の「**Active Directory**」を確認してください。アカウント・プロパティーが変更される場合には再始動が必要になる場合があります。
- クライアントとサーバーの両方が Windows 上にある場合、ローカル・システム・アカウントで DB2 サービスを開始できます。ただし、クライアントとサーバーが異なるドメインにある場合には、無効なターゲット・プリンシパル名のエラーによって接続が失敗する可能性があります。これは、クライアント上で host/*server hostname@server domain name* という形式の完全修飾サーバー・ホスト名と完全修飾ドメイン名を使ってターゲットのプリンシパル名を明示的にカタログすることによって回避できます。

  たとえば、`host/myhost.domain.ibm.com@DOMAIN.IBM.COM` のように指定します。

  これを行わない場合は、有効なドメイン・アカウントで DB2 サービスを開始する必要があります。

サーバー上でのパスワードの保守

パスワードのメンテナンス作業を実行する必要があるかもしれません。 そのような作業は通常、サーバーにおいて必要ですが、その際、サーバー環境で作業できないユーザーや快適に作業できないユーザーが多数生じるため、これらの作業の実行は簡単ではありません。

DB2 データベース・システム提供する機能を使用すれば、サーバーにいなくても、パスワードを更新および確認することができます。DB2 データベース製品を使用して、AIX、Linux、およびWindows オペレーティング・システム上でパスワードを変更することができます。

例えば、エラー・メッセージ SQL1404N 「パスワードの有効期限が切れています。」または SQL30082N「セキュリティー処理は、理由 1 (PASSWORD EXPIRED) により失敗しました。」を受け取った場合は、次のように CONNECT ステートメントを使用してパスワードを変更します。

```
CONNECT TO database USER userid USING
    password NEW new_password CONFIRM new_password
```

また、ATTACH コマンドおよび DB2 構成アシスタント (CA) の「**パスワードの変更 (Password change)**」ダイアログを使用してパスワードを変更することもできます。

許可、特権、およびオブジェクト所有権

ユーザー (許可 ID で識別される) は、指定された関数を実行する権限を持っている場合にのみ、SQL または XQuery ステートメントを正常に実行することができます。表を作成するには、ユーザーに表作成の許可が必要であり、表を変更するには、表変更の許可が必要となります。その他も同様です。

許可には、この後に解説されているとおり、*管理権限*、*特権* および *LBAC 信用証明情報* の 3 つの形式があります。

データベース・マネージャーでは、特定のタスクを実行するのに必要なデータベース機能を使用するために、各ユーザーが特定の許可を暗黙または明示的に与えられていなければなりません。*明示的* な権限あるいは特権は、ユーザーに対して付与されます (データベース・カタログでは GRANTEETYPE が U)。*暗黙* の権限あるいは特権は、各ユーザーが所属するグループに対して付与され (データベース・カタログでは GRANTEETYPE は G)、または、ユーザー、グループ、または別のロールをメンバーとして持つロールに対して付与されます (データベース・カタログでは GRANTEETYPE は R)。

管理権限

管理権限のある担当者はいずれも、データベース・マネージャーを制御するタスクに携わり、データの安全と整合性に対する責任を持ちます。 SYSADM および DBADM レベルの管理権限のある担当者は、データベース・セキュリティーに関連するオブジェクト以外のすべてのオブジェクトについてすべての特権が暗黙的にあり、だれがデータベース・マネージャーにアクセスするか、およびこのアクセスの程度を制御します。

権限レベル によって、特権のグループ分けの方法、およびより高いレベルのデータベース・マネージャーの保守とユーティリティー操作が得られます。データベース *権限* は、ユーザーがデータベース・レベルのアクティビティーを実行できるようにします。ユーザー、グループ、またはロールは、以下のような 1 つ以上の権限を持つことができます。

- インスタンス・レベルで機能する管理権限レベル、SYSADM (システム管理者)

  SYSADM 権限レベルは、データベース・マネージャーによって作成および保守されるすべてのリソースに対する制御を可能にします。システム管理者は DBADM、SYSCTRL、SYSMAINT、および SYSMON 権限をすべて所有し、DBADM 権限および SECADM 権限を付与または取り消す権限を持っています。

  SYSADM 権限を持つユーザーは、データベース・マネージャーの制御、およびデータの保護と整合性を担当します。 SYSADM 権限はデータベース内で暗黙の DBADM 権限を与えますが、データベース内で暗黙の SECADM 権限は与えません。
- データベース・レベルで機能する管理権限レベル:
  - DBADM (データベース管理者)

    DBADM 権限レベルはデータベース・レベルで適用され、1 つのデータベースに対する管理権限を与えます。このデータベース管理者は、オブジェクトの作

成、データベース・コマンドの発行、および表データへのアクセスに必要な権限を所有します。また、データベース管理者は、 CONTROL や個々の特権を付与または取り消すことができます。

– SECADM (セキュリティー管理者)

SECADM 権限レベルはデータベース・レベルで適用されます。これは表を保護するために使用されるロール、トラステッド・コンテキスト、監査ポリシー、セキュリティー・ラベル・コンポーネント、セキュリティー・ポリシー、およびセキュリティー・ラベルの作成、変更 (該当する場合)、およびドロップを行うのに必要な権限です。また、ロール、セキュリティー・ラベル、および免除の認可および取り消しのためと、SETSESSIONUSER 特権の認可および取り消しのために必要な権限でもあります。 SECADM 権限を持つユーザーは、所有していないオブジェクトの所有権を移行することができます。また、このようなユーザーは、AUDIT ステートメントを使用して、サーバー側の特定のデータベースまたはデータベース・オブジェクトに監査ポリシーを関連付けることもできます。

SECADM 権限には表に格納されたデータにアクセスする固有の特権はなく、他の追加の固有の特権もありません。この権限の付与が行えるのは、SYSADM 権限をもつユーザーだけとなります。SECADM 権限をユーザーに付与することはできますが、グループ、ロール、または PUBLIC には付与できません。

- インスタンス・レベルで機能するシステム制御権限レベル:
  – SYSCTRL (システム制御)

SYSCTRL 権限レベルは、システム・リソースに影響を与える操作に対する制御を可能にします。例えば、SYSCTRL 権限を持つユーザーは、データベースの作成、更新、開始、停止、またはドロップを行うことができます。さらに、このユーザーはインスタンスの開始または停止を行うことができますが、表データへのアクセスはできません。 SYSCTRL 権限を持つユーザーには、SYSMON もまた与えられます。

  – SYSMAINT (システム保守)

SYSMAINT 権限レベルは、インスタンスに関連したすべてのデータベースに対する保守操作を実行するのに必要な権限を与えます。 SYSMAINT 権限を持つユーザーは、データベースの更新と構成、データベースまたは表スペースのバックアップ、既存のデータベースのリストア、およびデータベースのモニターを行うことができます。 SYSCTRL と同様に、SYSMAINT は表データへのアクセス権限を与えません。 SYSMAINT 権限を持つユーザーには、SYSMON 権限もまた与えられます。

- SYSMON (システム・モニター) 権限レベル

SYSMON は、データベース・システム・モニターの使用に必要な権限を与えます。インスタンス・レベルで機能します。

- データベース権限

表やルーチンの作成、表へのデータのロードなどのアクティビティーを実行するには、特定のデータベース権限が必要です。例えば、ロード・ユーティリティー

を使ってデータを表にロードするには、LOAD データベース権限が必要です (その表に対する INSERT 特権も必要です)。

図 1 は、権限とその制御の範囲 (データベース、データベース・マネージャー) の間の関係を示します。

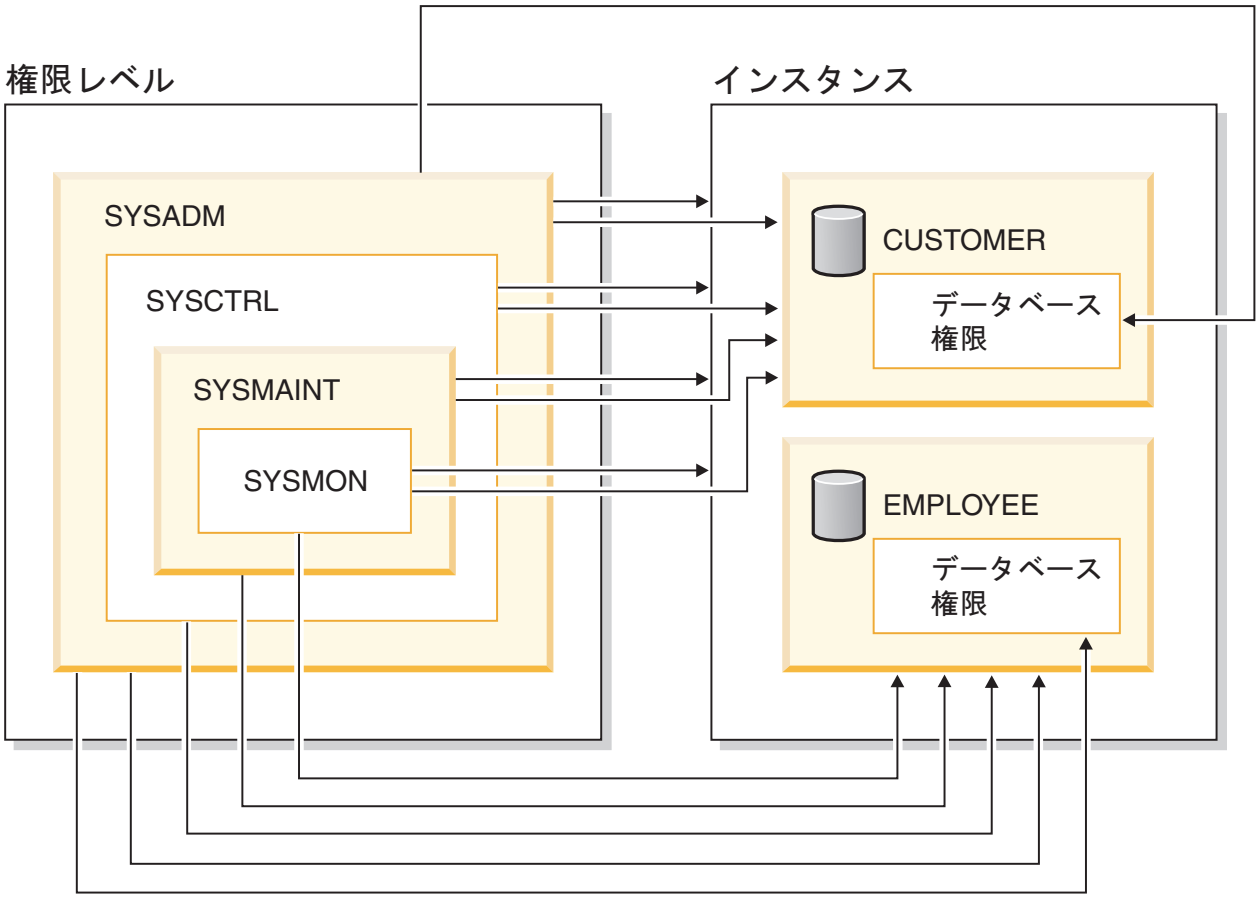


図 1. 権限の階層

特権

特権 とは、ユーザーが実行を許可されているアクティビティーです。許可ユーザーは、オブジェクトを作成することができ、所有しているオブジェクトにアクセス権を持ち、GRANT ステートメントを使用することによって、所有オブジェクトに対する特権を他のユーザーに渡すことができます。

特権は、個々のユーザー、グループ、または PUBLIC に付与できます。 PUBLIC は、将来のユーザーを含むすべてのユーザーで構成される特殊グループです。グループのメンバーであるユーザーは、グループがサポートされている場合は、グループに付与された特権を間接的に利用できます。

CONTROL 特権: オブジェクトに対する CONTROL 特権を持っているユーザーは、そのデータベース・オブジェクトにアクセスでき、そのオブジェクトに対する他のユーザーの特権を付与または取り消すことができます。

注: CONTROL 特権は、表、ビュー、ニックネーム、索引、およびパッケージにのみ適用されます。

他のユーザーがそのオブジェクトに対する CONTROL 特権を要求した場合、SYSADM または DBADM 権限を持つユーザーが、そのオブジェクトに対する CONTROL 特権を付与することができます。 CONTROL 特権は、オブジェクト所有者から取り消されることがありませんが、オブジェクト所有者は、TRANSFER OWNERSHIP ステートメントを使用して変更される場合があります。

場合によっては、オブジェクトの作成者がそのオブジェクトに対する CONTROL 特権を自動的に取得します。

個別特権: ユーザーが特定オブジェクトに対して特定のタスクを実行できるようにするために、個別特権を与えることができます。管理権限 (SYSADM または DBADM) を持つユーザー、または CONTROL 特権を持つユーザーは、他のユーザーの特権を付与または取り消すことができます。

個別特権およびデータベース権限は特定の機能の実行を許可しますが、同じ特権または権限を他のユーザーに与えることはできません。 GRANT ステートメントで WITH GRANT OPTION を使用すれば、表、ビュー、スキーマ、パッケージ、ルーチン、シーケンスに関する特権を他のユーザーに対して GRANT できる権利を、他のユーザーに拡張して与えることができます。ただし、WITH GRANT OPTION を使用する場合、特権を GRANT する人が、いったん GRANT された特権を取り消すことはできません。特権を取り消すためには、SYSADM 権限、DBADM 権限、または CONTROL 特権を持っていなければなりません。

パッケージまたはルーチン内のオブジェクトに対する特権: ユーザーにパッケージまたはルーチンを実行する特権があると、パッケージまたはルーチン内で使用されるオブジェクトに対する特定の特権が必ずしも必要とされません。パッケージまたはルーチンに静的 SQL または XQuery ステートメントが含まれる場合、パッケージの所有者の特権がそれらのステートメントに使用されます。パッケージまたはルーチンに動的 SQL または XQuery ステートメントが含まれる場合、特権の検査に使用される許可 ID は、動的照会ステートメントを発行するパッケージの DYNAMICRULES バインド・オプションの設定と、パッケージがルーチンのコンテキストで使用される際にそれらのステートメントが発行されるかどうかによって異なります。

1 つのユーザーまたはグループに対して、個々の特権または権限をいくつか組み合わせて許可することもできます。特権をオブジェクトに関連付ける場合、そのオブジェクトはすでに存在していなければなりません。例えば、表がそれ以前に作成されているのでなければ、その表についての SELECT 特権をユーザーに与えることはできません。

注: ユーザーまたはグループを表す許可名が権限と特権を付与され、しかもその許可名で作成されたユーザーまたはグループがいない場合には、注意が必要です。後で、その許可名を使用してユーザーまたはグループが作成され、その許可名に関連するすべての権限と特権を自動的に受け取る可能性があります。

すでに付与された特権を取り消すには、REVOKE ステートメントを使用します。 1 つの許可名から特権を取り消すと、すべての許可名によって付与された特権が取り消されます。

ある許可名から特権を取り消しても、その許可名によって特権を付与された他の許可名からその同じ特権が取り消されることはありません。例えば、ユーザー CLAIRE が SELECT WITH GRANT OPTION をユーザー RICK に与えた後、RICK が SELECT を BOBBY および CHRIS に与えたとします。もし CLAIRE が SELECT 特権を RICK から取り消しても、 BOBBY と CHRIS は引き続き SELECT 特権を保持します。

LBAC 信用証明情報

セキュリティー管理者は、ラベル・ベースのアクセス制御 (LBAC) を使用して、個々の行および個々の列ごとに、どのユーザーに書き込みアクセスがあり、どのユーザーに読み取りアクセスがあるのかを厳密に決定することができます。セキュリティー管理者は、セキュリティー・ポリシーを作成して LBAC システムを構成します。セキュリティー・ポリシー では、どのデータに誰がアクセスできるかの決定で使用される基準が記述されます。任意の 1 つの表を保護するためには、1 つのセキュリティー・ポリシーしか使用できませんが、異なる表には、異なるセキュリティー・ポリシーを用いて保護することができます。

セキュリティー・ポリシーを作成した後、セキュリティー管理者は、そのポリシーの一部分となる、セキュリティー・ラベルおよび免除と呼ばれるデータベース・オブジェクトを作成します。セキュリティー・ラベルは一連のセキュリティー基準を表現したものとなります。免除は、作成したセキュリティー・ポリシーで保護されたデータにアクセスする場合に、これを保有するユーザーがセキュリティー・ラベルの比較について、定められた規則を免れることができるものとなります。

作成が完了すると、セキュリティー・ラベルを表の個々の列と行に関連付けてそこに保持されているデータを保護することができます。セキュリティー・ラベルにより保護されるデータは、保護データと呼ばれます。セキュリティー管理者は、ユーザーにセキュリティー・ラベルを付与することにより、保護データへのアクセスを許可します。ユーザーが保護データへのアクセスを試行すると、そのユーザーのセキュリティー・ラベルが、データを保護しているセキュリティー・ラベルと比較されます。セキュリティー・ラベルには、保護ラベルによってブロックされるものと、されないものがあります。

オブジェクトの所有権

オブジェクトが作成されるとき、1 つの許可 ID に対して、そのオブジェクトの*所有権* が割り当てられます。所有権を与えられているユーザーは、任意の適用できる SQL または XQuery ステートメントを使ってそのオブジェクトを参照することを許可されます。

スキーマ内でオブジェクトを作成するとき、ステートメントの許可 ID は、暗黙的または明示的に指定されるスキーマ内でオブジェクトを作成するのに必要な特権を持っていなければなりません。つまり、許可名がスキーマの所有者であるか、スキーマに対する CREATEIN 特権を持っている必要があります。

注: 表スペース、バッファー・プール、またはデータベース・パーティション・グループを作成するときには、この要件は適用されません。これらのオブジェクトはスキーマ内には作成されません。

オブジェクトが作成されるとき、ステートメントの許可 ID がそのオブジェクトの所有者になります。

注: ただし、1 つの例外があります。 CREATE SCHEMA ステートメントで AUTHORIZATION オプションを指定した場合、 CREATE SCHEMA 操作の一部として作成されるすべてのオブジェクトは、 AUTHORIZATION オプションが指定する許可 ID によって所有されます。ただし、最初の CREATE SCHEMA 操作の後でスキーマ内で作成されるすべてのオブジェクトは、特定の CREATE ステートメントに関連した許可 ID によって所有されます。

例えば、ステートメント `CREATE SCHEMA SCOTTSTUFF AUTHORIZATION SCOTT CREATE TABLE T1 (C1 INT)` によって、スキーマ `SCOTTSTUFF` および表 `SCOTTSTUFF.T1` が作成され、このどちらもユーザー SCOTT によって所有されます。ここで、ユーザー BOBBY に対して `SCOTTSTUFF` スキーマに対する CREATEIN 特権が与えられ、BOBBY が表 `SCOTTSTUFF.T1` への索引を作成するとします。索引はスキーマの後で作成されるため、 `SCOTTSTUFF.T1` への索引を所有するのは BOBBY です。

特権は、作成されるオブジェクトのタイプに応じて、以下のようにオブジェクト所有者に割り当てられます。

- CONTROL 特権は、新しく作成される表、索引、およびパッケージに対して暗黙的に付与されます。この特権を持つオブジェクト作成者は、そのデータベース・オブジェクトにアクセスでき、そのオブジェクトに対する他のユーザーの特権を付与または取り消すことができます。他のユーザーがそのオブジェクトに対する CONTROL 特権を要求した場合、 SYSADM または DBADM 権限を持つユーザーが、そのオブジェクトに対する CONTROL 特権を付与する必要があります。オブジェクト所有者は、CONTROL 特権を取り消すことができません。
- ビュー定義によって参照されるすべての表、ビュー、およびニックネームに対する CONTROL 特権をオブジェクト所有者が持っている場合、新しく作成されるビューに対して CONTROL 特権が暗黙的に付与されます。
- 他のオブジェクト (トリガー、ルーチン、シーケンス、表スペース、バッファー・プールなど) には、 CONTROL 特権が関連付けられません。オブジェクト所有者は、オブジェクトに関連付けられるすべての特権を自動的に受け取ります (さらに所有者は、サポートされている場合、GRANT ステートメントで WITH GRANT オプションを使用することで、これらの特権を他のユーザーに与えることができます)。また、オブジェクト所有者は、オブジェクトの変更、コメントの追加、およびオブジェクトのドロップを行うことができます。これらの許可はオブジェクト所有者に暗黙的に与えられ、取り消すことはできません。

表の変更など、オブジェクトに対する特定の特権は、所有者によって付与できます。また SYSADM または DBADM 権限を持つユーザーによって所有者から取り消せます。 表にコメントするなど、オブジェクトに対する特定の特権は、所有者によって付与できません。また所有者から取り消せません。TRANSFER OWNERSHIP ステートメントを使用してこれらの特権を別のユーザーに移動します。オブジェクトが作成されるとき、ステートメントの許可 ID がそのオブジェクトの所有者になります。ただし、パッケージが作成され、OWNER バインド・オプションが指定さ

れている場合、パッケージ内の静的 SQL ステートメントによって作成されたオブジェクトの所有者は OWNER バインド・オプションの値となります。 さらに、AUTHORIZATION 節が CREATE SCHEMA ステートメントに指定される場合、AUTHORIZATION キーワードの後に指定される許可名はスキーマの所有者です。

セキュリティー管理者またはオブジェクト所有者は、TRANSFER OWNERSHIP ステートメントを使用してデータベース・オブジェクトの所有権を変更することができます。そこで、許可 ID を修飾子として使用してオブジェクトを作成してから TRANSFER OWNERSHIP ステートメントを使用して管理者オブジェクトに持つ所有権を許可 ID に移動することで、管理者は許可 ID のためにオブジェクトを作成できます。

さまざまなコンテキスト内の許可 ID

許可 ID は、識別および許可検査の 2 つの目的で使用されます。例えば、セッション許可 ID は、初期許可検査で使用されます。

特定のコンテキストで許可 ID の使用を参照する場合、以下に示されているとおり、そのコンテキストを識別するように許可の参照が修飾されます。

許可 ID のコンテキスト上の参照
定義

システム許可 ID
CONNECT 処理時の CONNECT 特権の検査のように、どの初期許可検査の実行でも使用される許可 ID。 CONNECT の処理時の認証プロセスの一環として、DB2 の命名要件との互換性のある許可 ID が作成されます。これは、DB2 データベース・システム内の外部ユーザー ID を表します。システム許可 ID は、接続を作成したユーザーを表します。システム許可 ID の現行値を確認するには、SYSTEM_USER 特殊レジスターを使用します。接続でのシステム許可 ID は変更できません。

セッション許可 ID
CONNECT の処理時に実行される初期検査の後に続くどのセッション許可検査でも使用される許可 ID 。セッション許可 ID のデフォルト値は、システム許可 ID の値になります。セッション許可 ID の現行値を確認するには、SESSION_USER 特殊レジスターを使用します。 USER 特殊レジスターは、SESSION_USER 特殊レジスターと同義です。 SET SESSION AUTHORIZATION ステートメントを使用すれば、セッション許可 ID を変更することができます。

パッケージ許可 ID
パッケージをデータベースにバインドするのに使用される許可 ID。この許可 ID は、OWNER バインド・オプションの値からとられます。パッケージ許可 ID は、場合によってはパッケージ・バインダーまたはパッケージ所有者と呼ばれます。

ルーチン所有者許可 ID
起動された SQL ルーチンの所有者としてシステム・カタログにリストされている許可 ID。

ルーチン呼び出し側許可 ID

SQL ルーチンを呼び出したステートメントのステートメント許可 ID である許可 ID。

ステートメント許可 ID

任意の許可要件で使用される以外に、オブジェクトの所有権の判別 (該当する場合) でも使用される特定の SQL ステートメントに関連付けられた許可 ID。これは、以下の SQL ステートメント・タイプに該当するソース許可 ID からその値をとります。

- 静的 SQL

  パッケージ許可 ID が使用されます。

- 動的 SQL (非ルーチン・コンテキストから)

  ケースごとにどの許可 ID が使用されるかを以下の表に示してあります。

パッケージの発行用の DYNAMICRULES オプションの値	使用される許可 ID
RUN	セッション許可 ID
BIND	パッケージ許可 ID
DEFINERUN、INVOKERUN	セッション許可 ID
DEFINEBIND、INVOKEBIND	パッケージ許可 ID

- 動的 SQL (ルーチン・コンテキストから)

  ケースごとにどの許可 ID が使用されるかを以下の表に示してあります。

パッケージの発行用の DYNAMICRULES オプションの値	使用される許可 ID
DEFINERUN、DEFINEBIND	ルーチン所有者許可 ID
INVOKERUN、INVOKEBIND	ルーチン呼び出し側許可 ID

ステートメント許可 ID の現行値を確認するには、CURRENT_USER 特殊レジスターを使用します。ステートメント許可 ID を直接変更することはできません。この ID は、各 SQL ステートメントの特性を反映するように、DB2 データベース・システムで自動的に変更されます。

インスタンス・レベルの権限

システム管理権限 (SYSADM)

SYSADM 権限レベルは、最も高いレベルの管理権限です。 SYSADM 権限を与えられたユーザーは、ユーティリティーを実行したり、データベースおよびデータベース・マネージャーのコマンドを発行したり、データベース・マネージャー・インスタンス内のデータベースの表の LBAC によって保護されていないデータにアクセスしたりできます。この権限は、インスタンス内のすべてのデータベース・オブジェクトを制御します。制御されるデータベース・オブジェクトには、データベース、表、ビュー、索引、パッケージ、スキーマ、サーバー、別名、データ・タイプ、関数、プロシージャー、トリガー、表スペース、データベース・パーティション・グループ、バッファー・プール、およびイベント・モニターがあります。

SYSADM 権限は、 *sysadm_group* 構成パラメーターによって指定されたグループに割り当てられます。このグループのメンバーシップは、データベース・マネージャーの外で、プラットフォームで使われているセキュリティー機能によって制御されます。

SYSADM 権限を持つユーザーだけが実行できる機能は、次のとおりです。

- データベースの移行
- データベース・マネージャー構成ファイルの変更 (SYSCTRL、SYSMAINT、または SYSMON 権限のあるグループを指定することを含む)
- DBADM 権限の付与および取り消し
- SECADM 権限の付与および取り消し

SYSADM 権限は、他のほとんどの権限によって提供されているすべての機能を提供しますが、SECADM 権限のどの機能も提供しません。 SECADM 権限によって提供されている機能は、他のどの権限によっても提供されません。 SYSADM 権限は、LBAC によって保護されているデータへのアクセスも提供しません。

注: SYSADM 権限を持つユーザーがデータベースを作成した場合、そのユーザーには、データベースに対する明示的な DBADM 権限が自動的に付与されます。データベース作成者を SYSADM グループから除去する場合、このユーザーがそのデータベースに DBADM としてアクセスできないようにするには、ユーザーの DBADM 権限を明示的に取り消す必要があります。

システム制御権限 (SYSCTRL)

SYSCTRL 権限は、最も高いレベルのシステム制御権限です。この権限があると、データベース・マネージャーのインスタンスとそのデータベースに対して、保守およびユーティリティー操作を実行することができます。これらの操作はシステム・リソースに影響を及ぼす場合がありますが、データベース内のデータに対するアクセスは認められていません。システム制御権限は、重要データの入ったデータベース・マネージャーのインスタンスを管理するユーザーを対象としたものです。

SYSCTRL 権限は、 *sysctrl_group* 構成パラメーターによって指定されたグループに割り当てられます。グループが指定されると、そのグループのメンバーシップは、プラットフォーム上で使用されるセキュリティー機能によって、データベース・マネージャーの外で制御されます。

SYSCTRL 以上の権限を持つユーザーだけが実行できることは、次のとおりです。

- データベース、ノード、または分散接続サービス (DCS) ディレクトリーの更新
- システムからのユーザーの切断
- データベースの作成またはドロップ
- 表スペースのドロップ、作成、または変更
- 新しいデータベースへのリストア

さらに、SYSCTRL 権限を持つユーザーは、システム保守権限 (SYSMAINT) およびシステム・モニター権限 (SYSMON) を持つユーザーの機能を実行できます。

SYSCTRL 権限を持つユーザーは、データベースへの接続に関する暗黙の特権も持っています。

注: SYSCTRL 権限を持つユーザーがデータベースを作成すると、そのユーザーには、そのデータベースに対する明示的な DBADM 権限が自動的に付与されます。データベースの作成者が SYSCTRL グループから除去され、そのデータベースに DBADM としてアクセスすることも防止する場合は、この DBADM 権限を明示的に取り消さなければなりません。

システム保守権限 (SYSMAINT)

SYSMAINT 権限は、2 番目のレベルのシステム制御権限です。この権限があると、データベース・マネージャー・インスタンスとそのデータベースに対して、保守およびユーティリティー操作を実行することができます。これらの操作はシステム・リソースに影響を及ぼす場合がありますが、データベース内のデータに対するアクセスは認められていません。システム保守権限は、重要データの入ったデータベース・マネージャー・インスタンス内のデータベースを保守するユーザーを対象としています。

SYSMAINT 権限は、 *sysmaint_group* 構成パラメーターによって指定されたグループに割り当てられます。グループが指定されると、そのグループのメンバーシップは、プラットフォーム上で使用されるセキュリティー機能によって、データベース・マネージャーの外で制御されます。

SYSMAINT 以上の権限を持つユーザーだけが実行できることは、次のとおりです。

- データベースの構成ファイルの更新
- データベースまたは表スペースのバックアップ
- 既存のデータベースへのリストア
- ロールフォワード・リカバリーの実行
- インスタンスの開始または停止
- 表スペースのリストア
- トレースの実行
- データベース・マネージャー・インスタンスまたはそのデータベースのデータベース・システム・モニター・スナップショットの取得

SYSMAINT、DBADM、またはそれ以上の権限を持つユーザーは、次のことを実行できます。

- 表スペースの状態の照会
- ログ履歴ファイルの更新
- 表スペースの静止
- 表の再編成
- RUNSTATS ユーティリティーを使用してのカタログ統計の収集

さらに、SYSMAINT 権限を持つユーザーは、データベースに接続する特権を暗黙的に与えられ、システム・モニター権限 (SYSMON) を持つユーザーに許可された機能を実行することもできます。

システム・モニター権限 (SYSMON)

SYSMON 権限は、データベース・マネージャー・インスタンスまたはそのデータベースを対象とするデータベース・システム・モニター・スナップショットの取得を

許可します。 SYSMON 権限は、構成パラメーター *sysmon_group* によって指定されたグループに割り当てられます。グループが指定されると、そのグループのメンバーシップは、プラットフォーム上で使用されるセキュリティー機能によって、データベース・マネージャーの外で制御されます。

SYSMON 権限を持つユーザーは、以下のコマンドを実行できます。
- GET DATABASE MANAGER MONITOR SWITCHES
- GET MONITOR SWITCHES
- GET SNAPSHOT
- LIST ACTIVE DATABASES
- LIST APPLICATIONS
- LIST DCS APPLICATIONS
- RESET MONITOR
- UPDATE MONITOR SWITCHES

SYSMON 権限を持つユーザーは、以下の API を使用できます。
- db2GetSnapshot - スナップショットの取得
- db2GetSnapshotSize - db2GetSnapshot() 出力バッファーに必要なサイズの見積もり
- db2MonitorSwitches - モニター・スイッチの入手/更新
- db2ResetMonitor - モニターのリセット

SYSMON 権限を持つユーザーは、以下の SQL 表関数を使用できます。
- すべての表スナップショット関数 (SYSPROC.SNAP_WRITE_FILE を実行する必要はありません)

  SYSPROC.SNAP_WRITE_FILE はスナップショットを取得して、その内容をファイルに保管します。ヌルの入力パラメーターを使って表スナップショット関数を呼び出した場合、リアルタイムのシステム・スナップショットの代わりに、ファイルの内容が戻されます。

SYSADM、SYSCTRL、または SYSMAINT 権限レベルを持つユーザーには、SYSMON 権限もまた与えられます。

データベース・レベルの権限

セキュリティー管理者権限 (SECADM)

SECADM 権限は、ロール、トラステッド・コンテキスト、監査ポリシー、セキュリティー・ラベル・コンポーネント、セキュリティー・ポリシー、およびセキュリティー・ラベルの作成、変更 (該当する場合)、およびドロップを行うのに必要な権限です。また、ロール、セキュリティー・ラベル、および免除と、SETSESSIONUSER 特権の認可および取り消しのために必要な権限でもあります。 SECADM 権限には表に格納されたデータにアクセスする固有の特権はありません。

SECADM (セキュリティー管理者) 権限を付与できるのは、システム管理者 (SYSADM 権限の保有者) のみであり、付与される対象となりうるのは、グループやロールではなくユーザーです。この権限は、以下の機能のみを付与できます。
- セキュリティー・ラベル・コンポーネントの作成、変更、コメント作成、およびドロップ
- セキュリティー・ポリシーの作成、変更、コメント作成、およびドロップ

- セキュリティー・ラベルの作成、コメント作成、およびドロップ
- ロールの作成、コメント作成、およびドロップ
- トラステッド・コンテキストの作成、変更、コメント作成、およびドロップ
- 監査ポリシーの作成、変更、コメント作成、およびドロップ
- 監査システム・ストアード・プロシージャーおよび表関数の使用: SYSPROC.AUDIT_ARCHIVE、SYSPROC.AUDIT_LIST_LOGS、および SYSPROC.AUDIT_DELIM_EXTRACT。これらを起動できるのは、セキュリティー管理者のみです。
- サーバー側の特定のデータベースまたはデータベース・オブジェクトに監査ポリシーを関連付けるための AUDIT ステートメントの使用
- セキュリティー・ラベルの付与および取り消し
- 免除の付与および取り消し
- SETSESSIONUSER 特権の付与および取り消し
- ロールの付与および取り消し
- SQL ステートメントの許可 ID によって所有されていないオブジェクトに対する SQL ステートメント TRANSFER OWNERSHIP の実行

他の権限 (SYSADM を含む) は、これらの機能を付与できません。

データベース管理権限 (DBADM)

DBADM 権限は、特定のデータベースに対する管理権限であり、それによってユーザーは特定のアクションを実行し、そのデータベースに対してデータベース・コマンドを発行することができます。データが LBAC によって保護されていない限り、DBADM 権限はデータベースの任意の表内のデータへのアクセスを許可します。LBAC によって保護されているデータにアクセスするには、適切な LBAC 信用証明情報が必要です。

DBADM 権限が付与されると、以下のデータベース権限も同じデータベースに対して明示的に付与されます (これは、DBADM 権限が後に取り消された場合に、自動的に取り消されることはありません)。

- BINDADD
- CONNECT
- CREATETAB
- CREATE_EXTERNAL_ ROUTINE
- CREATE_NOT_FENCED_ROUTINE
- IMPLICIT_SCHEMA
- QUIESCE_CONNECT
- LOAD

SYSADM 権限を持つユーザーだけが DBADM 権限の付与または取り消しを実行できます。 DBADM 権限を持つユーザーは、データベースに対する特権を他のユーザーに付与できます。また、だれが特権を付与したかにかかわりなく、ユーザーの特権を取り消すこともできます。

データベースに対して DBADM、またはそれ以上の権限を保持すると、ユーザーはそのデータベースで以下のアクションを実行することができます。

- ログ・ファイルの読み取り
- イベント・モニターの作成、活動化、およびドロップ

データベースに対して DBADM 権限を持つユーザー、または SYSMAINT 権限かそれ以上の権限を持つユーザーは、データベースで以下のアクションを実行することができます。

- 表スペースの状態の照会
- ログ履歴ファイルの更新
- 表スペースの静止
- 表の再編成
- RUNSTATS ユーティリティーを使用してのカタログ統計の収集

DBADM 権限は、他の権限と同じ機能のいくつかを提供しますが、SECADM 権限のどの機能も提供しません。 SECADM 権限によって提供されている機能は、他のどの権限によっても提供されません。

LOAD 権限

データベース・レベルの LOAD 権限、および表に対する INSERT 特権を持っているユーザーは、 LOAD コマンドを使用してデータを表にロードすることができます。

データベース・レベルの LOAD 権限、および表に対する INSERT 特権を持っているユーザーは、直前のロード操作でデータを挿入するロードを行った場合に、LOAD RESTART または LOAD TERMINATE を行うことができます。

データベース・レベルの LOAD 権限、および表に対する INSERT 特権と DELETE 特権を持っているユーザーは、 LOAD REPLACE コマンドを使用できます。

直前のロード操作でロード置換を行った場合、ユーザーは、DELETE 特権が付与されていないと、 LOAD RESTART または LOAD TERMINATE を行うことができません。

ロード操作の一部として例外表が使用される場合、ユーザーには、その例外表に対する INSERT 特権が必要です。

この権限を持っているユーザーは、 QUIESCE TABLESPACES FOR TABLE、RUNSTATS、および LIST TABLESPACES コマンドを実行することができます。

データベース権限

個々のデータベース権限により、その権限を保持する許可 ID が、特定のタイプの処置をデータベース全体に対して実行できるようになります。データベース権限は特権とは違います。特権の場合は、表や索引などの特定のデータベース・オブジェクトに対して特定の処置を取ることができます。

以下にデータベース権限を示します。

SECADM
: これの所有者が実行できる内容として、セキュリティー管理者としての振る舞い、セキュリティー・オブジェクトの作成およびドロップ、セキュリティー・オブジェクトの許可または特権の付与および取り消し、およびオブジェ

クトの所有権の移譲があります。セキュリティー管理者は、トラステッド・コンテキスト、監査ポリシー、データベースの役割、およびデータの LBAC 保護を管理します。

DBADM
保有者にデータベース管理者として振る舞う権限を付与します。特に、SECADM 以外の他のデータベース権限をすべて保有者に付与します。

CONNECT
保有者は、データベースに接続できます。

BINDADD
保有者は、データベース内に新しいパッケージを作成できます。

CREATETAB
保有者は、データベース内に新しい表を作成できます。

CREATE_EXTERNAL_ ROUTINE
保有者は、アプリケーションによって、またデータベースの他のユーザーによって使用されるプロシージャーを作成できます。

CREATE_NOT_FENCED_ROUTINE
保有者は、「not fenced」のユーザー定義関数 (UDF) またはプロシージャーを作成できます。 CREATE_EXTERNAL_ROUTINE は、CREATE_NOT_FENCED_ROUTINE を持つすべてのユーザーに対して自動的に付与されます。

重要: データベース・マネージャーはそのストレージや制御ブロックを、「not fenced」の UDF またはプロシージャーから保護しません。したがって、この権限を持つユーザーは、UDF を「not fenced」として登録する前に、十分にテストするよう特に注意しなければなりません。

IMPLICIT_SCHEMA
どのユーザーも、まだ存在していないスキーマ名を指定した CREATE ステートメントを使用してオブジェクトを作成することによって、暗黙にスキーマを作成することができます。 SYSIBM が暗黙に作成されたスキーマの所有者になり、 PUBLIC にこのスキーマ内にオブジェクトを作成するための特権が与えられます。

LOAD 保有者は表にデータをロードできます。

QUIESCE_CONNECT
保有者は静止中のデータベースにアクセスできます。

SYSADM 権限を持つ許可 ID のみ、SECADM および DBADM 権限を付与できます。他のすべての権限は、SYSADM または DBADM 権限を保持する許可 ID によって付与できます。

データベース作成時に、新しいデータベースに関して以下のデータベース権限が自動的に PUBLIC に付与されます。

- CREATETAB
- BINDADD
- CONNECT
- IMPLICIT_SCHEMA

さらに、以下の特権が付与されます。

- USERSPACE1 表スペースに対する USE 特権
- システム・カタログ・ビュー上での SELECT 特権

PUBLIC からデータベース権限を除去するには、DBADM または SYSADM 権限を持つ許可 ID によって明示的に取り消さなければなりません。

暗黙スキーマ権限 (IMPLICIT_SCHEMA) に関する考慮事項

新しいデータベースが作成されるとき、 PUBLIC に IMPLICIT_SCHEMA データベース権限が与えられます。この権限を使用して、どのユーザーも、オブジェクトを作成し、すでに存在していないスキーマ名を指定することによって、スキーマを作成することができます。 SYSIBM が暗黙に作成されたスキーマの所有者になり、PUBLIC にこのスキーマ内にオブジェクトを作成するための特権が与えられます。

だれが暗黙にスキーマ・オブジェクトを作成できるかを制御することがデータベースで必要な場合は、 IMPLICIT_SCHEMA データベース権限を PUBLIC から取り消す必要があります。いったんこれを行うと、スキーマ・オブジェクトが作成される方法は、以下の 3 つしかありません。

- どのユーザーも、CREATE SCHEMA ステートメントで自分自身の許可名を使用してスキーマを作成することができます。
- DBADM 権限を持つどのユーザーも、すでに存在していなければどのスキーマでも明示的に作成することができ、オプションで、別のユーザーをそのスキーマの所有者として指定することができます。
- DBADM 権限をもつどのユーザーも (PUBLIC と独立して) IMPLICIT_SCHEMA データベース権限を持っているため、他のデータベース・オブジェクトを作成しているときに、任意の名前を持ったスキーマを暗黙に作成することができます。SYSIBM が暗黙に作成されたスキーマの所有者になり、 PUBLIC がスキーマ内にオブジェクトを作成する特権を持ちます。

特権

許可 ID 特権

許可 ID 特権は、許可 ID に対するアクションに関する特権です。現在、このような特権として唯一あるのが、SETSESSIONUSER 特権です。

SETSESSIONUSER 特権はユーザーまたはグループに付与でき、この特権の所有者は、特権が付与されているどの許可 ID にでも、ID を切り替えることができます。ID の切り替えは、SQL ステートメント SET SESSION AUTHORIZATION を使用して行われます。SETSESSIONUSER 特権の付与は、SECADM 権限を持つユーザーのみが行えます。

注: バージョン 8 のデータベースをバージョン 9.1 以降にマイグレーションすると、そのデータベースに対して明示的 DBADM 権限を持つ許可 ID には、PUBLIC に対する SETSESSIONUSER 特権が自動的に付与されます。こうすることで、セッション許可 ID を任意の許可 ID に設定できる DBADM 権限を持つ許可 ID に依存するアプリケーションが中断することがないようにしています。これは、許可 ID が SYSADM 権限を持っていても DBADM が明示的に付与されていないときは、行われません。

スキーマ特権

スキーマ特権は、オブジェクト特権区分に入ります。オブジェクト特権は、図 2 に示されています。

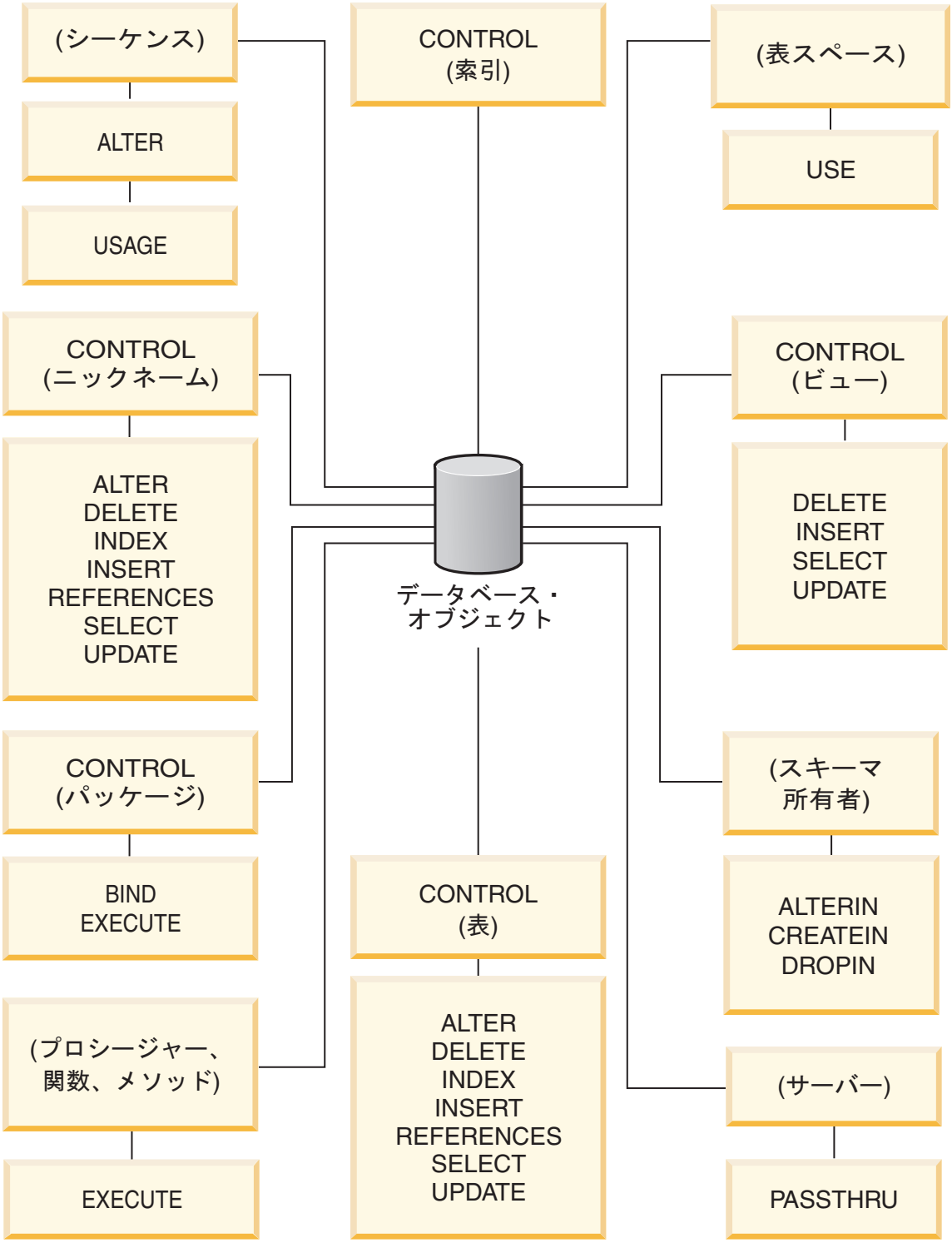


図 2. オブジェクト特権

スキーマ特権には、データベース内のスキーマ上でのアクションが含まれます。ユーザーには、以下の特権のどれでも付与することができます。

- CREATEIN により、ユーザーはスキーマ内にオブジェクトを作成できます。
- ALTERIN により、ユーザーはスキーマ内のオブジェクトを変更できます。
- DROPIN により、ユーザーはスキーマ内からオブジェクトをドロップできます。

スキーマの所有者は、これらの特権をすべて持ち、その特権を他のユーザーに付与する能力を持ちます。スキーマ・オブジェクト内で操作されるオブジェクトには、表、ビュー、索引、パッケージ、データ・タイプ、関数、トリガー、プロシージャー、および別名があります。

表スペース特権

表スペース特権は、データベースの表スペースに対してアクションを実行することを可能にします。表スペースの USE 特権を付与されたユーザーは、その表スペース内で表を作成できます。

表スペースの所有者 (多くの場合、 SYSADM または SYSCTRL 権限を持っている作成者) は USE 特権を持っており、他のユーザーにこの特権を付与することができます。デフォルトでは、データベースの作成時に、表スペース USERSPACE1 に関する USE 特権が PUBLIC に付与されますが、この特権は取り消すこともできます。

USE 特権は、 SYSCATSPACE または SYSTEM TEMPORARY 表スペースでは使用できません。

表およびビューの特権

表およびビューの特権には、データベース内の表やビューに対するアクションが含まれます。

次に挙げる特権のいずれかを使用するユーザーには、データベースについての CONNECT 権限が必要です。

- CONTROL 特権は、表やビューのドロップ、表についての個別の特権の GRANT や REVOKE を含む、表やビューに対する全ての特権を許可します。 CONTROL 特権を付与するには、SYSADM または DBADM 権限が必要です。表の作成者は、自動的にその表に対する CONTROL 特権を受け取ります。ビューの作成者は、ビュー定義の中で参照されているすべての表、ビュー、およびニックネームに対する CONTROL 特権を持っているか、 SYSADM または DBADM 権限を持っている場合にのみ、自動的に CONTROL 特権を受け取ります。
- ALTER は、ユーザーが表を変更することを許可します (たとえば、表への列またはユニーク制約の追加)。 ALTER 特権を持つユーザーは、表または表の列に対する COMMENT ON も可能です。表に対する実行可能な変更操作については、ALTER TABLE および COMMENT ステートメントの説明を参照してください。
- DELETE 特権は、表やビューからの行の削除をユーザーに許可します。
- INDEX 特権は、表についての索引の作成をユーザーに許可します。索引の作成者には、索引についての CONTROL 特権が自動的に与えられます。
- INSERT 特権は、表やビューに対する行の挿入や、 IMPORT ユーティリティーの実行をユーザーに許可します。

- REFERENCES 特権は、表に対するリレーションシップの親としての指定や、外部キーの作成および削除をユーザーに許可します。ユーザーは、特定の列にのみこの特権を持つことができます。
- SELECT 特権は、表やビューからの行の取り出しや、表に基づくビューの作成、EXPORT ユーティリティーの実行をユーザーに許可します。
- UPDATE 特権は、表またはビューの項目の変更、あるいは表またはビューの 1 つ以上の特定の列の中の項目の変更をユーザーに許可します。ユーザーは、特定の列にのみこの特権を持つことができます。

GRANT ステートメントの WITH GRANT OPTION を使用して、これらの特権を他のユーザーに GRANT する特権を GRANT することもできます。

注: ユーザーまたはグループが、ある表に対する CONTROL 特権を GRANT された場合、その表に対する他のすべての特権は、自動的に WITH GRANT OPTION によって GRANT されます。その後、表に対する CONTROL 特権をユーザーから取り消しても、そのユーザーは自動的に付与された他の特権を依然として持っています。 CONTROL 特権と一緒に付与された特権をすべて取り消す場合は、特権を個別に明示的に取り消すか、または REVOKE ステートメントに ALL キーワードを指定しなければなりません。以下にその例を示します。

```
REVOKE ALL
  ON EMPLOYEE FROM USER HERON
```

型付き表に関しては、表およびビューの特権に関連したインプリメンテーションがあります。

注: 特権は、表階層の各レベルで別々に付与されます。その結果、型付き表の階層内のスーパー表で特権を付与されたユーザーは、副表にも間接的に影響を与えることがあります。しかし、その副表で必要な特権が保持されている場合は、ユーザーは副表に対する操作を直接的にしか行えません。

表階層の表の間のスーパー表/副表のリレーションシップは、SELECT、UPDATE、および DELETE などの操作が、操作のターゲット表とそのすべての副表 (あれば) の行に影響を与えることを意味します。この性質を*代替性*と呼ぶことができます。例えば、タイプ Manager_t の副表マネージャーを使用して、タイプ Employee_t の Employee 表を作成したとします。構造化タイプ Employee_t と Manager_t 間のタイプ/サブタイプのリレーションシップ、また表 Employee と Manager 間の対応する表/副表のリレーションシップによって示されているとおり、マネージャーはある種の (特殊な) 従業員です。このリレーションシップの結果、次の SQL 照会は、

```
SELECT * FROM Employee
```

従業員とマネージャー両方のオブジェクト ID と Employee_t 属性を戻します。同様に、次の更新操作は、

```
UPDATE Employee SET Salary = Salary + 1000
```

マネージャーと従業員の給与を 1000 ドル引き上げます。

Employee 表で SELECT 特権を持つユーザーは、 Manager 表で明示的な SELECT 特権を持っていなくても、この SELECT 操作を実行できます。しかし、そのような

ユーザーは、 Manager 副表に対して SELECT 操作を直接実行することは許可されませんので、 Manager 表の継承されたのではない列にアクセスすることはできません。

同様に、Employee 表で UPDATE 特権を持つユーザーは、Manager 表で明示的な UPDATE 特権がなくても、通常の従業員とマネージャー両方に影響を与えるような、Manager に対する UPDATE 操作を実行できます。しかし、そのようなユーザーは、 Manager 副表に対して UPDATE 操作を直接実行することは許可されませんので、 Manager 表の継承されたのではない列を更新することはできません。

パッケージ特権

パッケージとは、データベース・オブジェクトの 1 つで、データベース・マネージャーが、特定のアプリケーション・プログラムにとって最も効率的な方法でデータにアクセスするのに必要な情報が入っています。パッケージの特権を与えられたユーザーは、パッケージの作成と操作を行うことができます。

以下のいずれかの特権を使用するユーザーには、データベースに対する CONNECT 権限が必要です。

- CONTROL 特権は、パッケージの再バインド、ドロップ、または実行、およびこれらの特権を他のユーザーに与えることをユーザーに許可します。パッケージの作成者には自動的にこの特権が与えられます。 CONTROL 特権を持つユーザーには、BIND 特権と EXECUTE 特権が付与されます。さらに、そのユーザーは GRANT ステートメントを使って他のユーザーにこれらの特権を付与することもできます。 (WITH GRANT OPTION を使用して特権を GRANT すれば、 BIND または EXECUTE 特権を受け取るユーザーは、その特権をさらに他のユーザーに与えることができます。) CONTROL 特権を付与するためには、SYSADM または DBADM 権限が必要です。
- パッケージの BIND 特権は、そのパッケージの再バインドやバインド、また、同じパッケージ名と作成者の新規のバージョンの追加をユーザーに許可します。
- EXECUTE 特権は、パッケージの実行をユーザーに許可します。

注: すべてのパッケージ特権は、パッケージ名と作成者が同じすべてのバージョンに適用されます。

これらのパッケージ特権に加えて、BINDADD データベース特権によって、ユーザーは、データベース内に新しいパッケージを作成するか、または既存のパッケージを再バインドすることができます。

ニックネームによって参照されるオブジェクトは、オブジェクトを格納するデータ・ソースでの認証検査をパスする必要があります。さらに、パッケージ・ユーザーには、データ・ソースにあるデータ・ソース・オブジェクトに対する適切な特権、または権限レベルが必要です。

DB2 データベースは DB2 ファミリーのデータ・ソースと通信するときに動的照会を使用するため、ニックネームを含むパッケージが付加的な許可ステップを必要とする可能性があります。データ・ソースでパッケージを実行する許可 ID には、そのデータ・ソースでパッケージを動的に実行するための適切な権限が必要です。

索引特権

索引または索引の仕様の作成者には、索引に対する CONTROL 特権が自動的に与えられます。索引の CONTROL 特権は、実際には、索引を削除することを可能にします。ある索引の CONTROL 特権を付与するためには、そのユーザーには SYSADM または DBADM 権限が必要です。

表レベルの INDEX 特権は、表に関する索引の作成をユーザーに許可します。

ニックネーム・レベルの INDEX 特権は、そのニックネームに対する索引の仕様の作成をユーザーに許可します。

シーケンス特権

シーケンスの作成者には、そのシーケンスに対する USAGE および ALTER 特権が自動的に与えられます。 USAGE 特権は、シーケンスに対して NEXT VALUE および PREVIOUS VALUE 式を使用するために必要とされます。他のユーザーに NEXT VALUE および PREVIOUS VALUE 式の使用を許可するには、シーケンスの USAGE 特権を PUBLIC に付与しなければなりません。これで、すべてのユーザーが指定されたシーケンスでこれらの式を使用できるようになります。

シーケンスに対する ALTER 特権を持つユーザーは、シーケンスの再始動、今後のシーケンス値の増分の変更といったタスクを実行できます。シーケンスの作成者は ALTER 特権を他のユーザーに GRANT でき、 WITH GRANT OPTION を使用すれば、それらのユーザーもまた、これらの特権をさらに他のユーザーに GRANT できるようになります。

ルーチン特権

EXECUTE 特権には、データベース内の関数、プロシージャー、およびメソッドといったすべてのタイプのルーチンのアクションが含まれます。 EXECUTE 特権を与えられたユーザーはルーチンを呼び出すことができ、そのルーチンからのソース関数を作成できます (関数にのみ適用されます)。また、CREATE VIEW、CREATE TRIGGER といった任意の DDL ステートメント内でルーチンを参照できます。

外部のストアード・プロシージャー、関数、またはメソッドを定義するユーザーは、 EXECUTE WITH GRANT 特権を付与されます。 WITH GRANT OPTION を使用して EXECUTE 特権を他のユーザーに GRANT した場合、そのユーザーは、さらに他のユーザーに EXECUTE 特権を GRANT できます。

ワークロードの使用特権

ワークロードの使用を有効にするには、データベース管理者が GRANT USAGE ON WORKLOAD ステートメントを使用して、そのワークロードの USAGE 特権をユーザー、グループ、ロールのいずれかに付与します。

DB2 データベース・システムは、一致するワークロードを検出すると、セッション・ユーザーがそのワークロードの USAGE 特権を持っているかどうかを確認します。セッション・ユーザーがそのワークロードの USAGE 特権を持っていなければ、DB2 データベース・システムは、番号付きリストの中で一致する次のワークロードを検索します。したがって、セッション・ユーザーがワークロードの USAGE 特権を持っていなければ、そのワークロードは存在しないかのように扱われる、ということです。

USAGE 特権の情報はカタログに格納されており、SYSCAT.WORKLOADAUTH ビューで表示できます。

USAGE 特権を取り消すには、REVOKE USAGE ON WORKLOAD ステートメントを使用します。

ワークロードが接続属性に合致している限り、SYSADM 権限または DBADM 権限を保持しているユーザーは、カタログ内に存在するあらゆるワークロードを使用できます。

SYSDEFAULTUSERWORKLOAD ワークロードと USAGE 特権

RESTRICT オプションを指定しないでデータベースを作成する場合は、SYSDEFAULTUSERWORKLOAD の USAGE 特権がデータベース作成時に PUBLIC に付与されます。そうでない場合は、SYSADM 権限または DBADM 権限を持っているユーザーが USAGE 特権を明示的に付与する必要があります。

セッション・ユーザーが、SYSDEFAULTUSERWORKLOAD をはじめ、どのワークロードについても USAGE 特権を持っていない場合は、SQL エラーが戻されます。

SYSDEFAULTADMWORKLOAD ワークロードと USAGE 特権

SYSDEFAULTADMWORKLOAD の USAGE 特権をいずれかのユーザーに明示的に付与することはできません。このワークロードを使用できるのは、SYSADM 権限または DBADM 権限のある SESSION 許可 ID を持っているユーザーが SET WORKLOAD TO SYSDEFAULTADMWORKLOAD コマンドを実行した場合に限られます。

GRANT USAGE ON WORKLOAD ステートメントも REVOKE USAGE ON WORKLOAD ステートメントも、SYSDEFAULTADMWORKLOAD には無効です。

タスクおよび実行に必要な許可

仕事の責任の分担方法は、それぞれの組織によって違います。 表 2 に、他の一般的な仕事の種類、それらの仕事に通常伴うタスク、およびそれらのタスクを行うために必要な権限または特権を挙げます。

表 2. 一般的な仕事の種類、タスク、および必要な権限許可

仕事の種類	タスク	必要な許可
部署管理者	部署のシステムを監督する。データベースを作成する。	SYSCTRL 権限。部署に独自のインスタンスのある場合は SYSADM 権限。
セキュリティー管理者	権限許可と特権の一部または全部を他のユーザーに許可する。	SYSADM または DBADM 権限
データベース管理者	データベースの設計、開発、操作、保全、保守を行う。	データベースについての DBADM および SYSMAINT 権限。場合によっては、SYSCTRL 権限。
システム・オペレーター	データベースをモニターし、バックアップ機能を実行する。	SYSMAINT 権限。

表 2. 一般的な仕事の種類、タスク、および必要な権限許可 (続き)

仕事の種類	タスク	必要な許可
アプリケーション・プログラマー	データベース・マネージャーのアプリケーション・プログラムの開発およびテストを行う。テスト・データの表を作成することもある。	既存のパッケージに対する BINDADD、BIND 特権、 1 つ以上のデータベースの CONNECT および CREATETAB 特権、一部の特定のスキーマの特権、および一部の表についての一連の特権。 CREATE_EXTERNAL_ROUTINE が必要な場合もあります。
ユーザー・アナリスト	システム・カタログ・ビューを調べて、アプリケーション・プログラムのデータ要件を定義する。	カタログ・ビューの SELECT 特権。 1 つ以上のデータベースの CONNECT 特権。
プログラム・エンド・ユーザー	アプリケーション・プログラムを実行する。	パッケージの EXECUTE 特権。 1 つ以上のデータベースの CONNECT 特権。この表の下にある注記を参照してください。
インフォメーション・センター・コンサルタント	照会ユーザーについてのデータ要件を定義する。表とビューを作成し、データベース・オブジェクトへのアクセス権を付与することによって、データを提供する。	データベースについての DBADM 権限。
照会ユーザー	データの検索、追加、削除、または変更のために SQL ステートメントを出す。場合によっては、結果を表に保管する。	1 つ以上のデータベースの CONNECT 特権。作成する表およびビューのスキーマの CREATEIN 特権。一部の表およびビューの SELECT、 INSERT、 UPDATE、 DELETE 特権。

注: アプリケーション・プログラムに動的 SQL ステートメントが含まれている場合、プログラムのエンド・ユーザーには、 EXECUTE と CONNECT に加えてさらにほかの特権 (SELECT、INSERT、DELETE、および UPDATE など) が必要になる場合があります。

アクセスの付与、取り消し、モニター

特権の付与

ほとんどのデータベース・オブジェクトに対する特権を GRANT するには、ユーザーがそのオブジェクトに対する SYSADM 権限、DBADM 権限、または CONTROL 特権を持つか、またはそのユーザーが WITH GRANT OPTION 特権を保持していなければなりません。特権を付与できるのは既存のオブジェクトについてだけです。

他ユーザーに CONTROL 特権を付与するためには、SYSADM または DBADM 権限が必要です。 DBADM 権限を付与するには、SYSADM 権限が必要です。

GRANT ステートメントは、許可ユーザーが特権を付与することができるようにするものです。特権は、1 つのステートメントで、1 つ以上の許可名に付与するか、

あるいは、特権をすべてのユーザーが使用可能なようにする PUBLIC に付与することができます。許可名は、個別のユーザーかまたはグループのいずれかにすることができることに注意してください。

オペレーティング・システムに同じ名前のユーザーとグループがある場合、ユーザーとグループのどちらに特権を付与するのかを指定する必要があります。 GRANT および REVOKE ステートメントのどちらにおいても、 USER および GROUP というキーワードがサポートされています。これらのオプションのキーワードが使用されない場合、データベース・マネージャーはオペレーティング・システムのセキュリティー機能をチェックして、その許可名がユーザーであるかグループであるかを判別します。許可名がユーザーとグループの両方である可能性がある場合、エラーが戻されます。次の例は、HERON というユーザーに対して、 EMPLOYEE 表についての SELECT 特権を付与するものです。

```
GRANT SELECT
 ON EMPLOYEE TO USER HERON
```

次の例は、HERON というグループに対して、 EMPLOYEE 表についての SELECT 特権を付与するものです。

```
GRANT SELECT
   ON EMPLOYEE TO GROUP HERON
```

コントロール・センターで、「スキーマ特権」ノートブック、「表スペース特権」ノートブック、および「ビュー特権」ノートブックを使用して、これらのデータベース・オブジェクトに対する特権を付与または取り消すことができます。これらのノートブックのいずれかをオープンするには、以下のステップに従ってください。

1. コントロール・センターで、処理するオブジェクトを含むフォルダー (「ビュー」フォルダーなど) が表示されるまで、オブジェクト・ツリーを展開します。
2. フォルダーをクリックします。

   このフォルダー内の既存のデータベース・オブジェクトがコンテンツ・ペインに表示されます。
3. コンテンツ・ペインで興味のあるオブジェクトを右クリックし、ポップアップ・メニューで「**特権**」を選択します。

   該当する「特権」ノートブックがオープンします。

特権の取り消し

REVOKE ステートメントを使用すれば、許可ユーザーは、他のユーザーに付与されている特権を取り消すことができます。

データベース・オブジェクトに対する特権を取り消すには、DBADM 権限、SYSADM 権限、またはそのオブジェクトに対する CONTROL 特権が必要です。 WITH GRANT OPTION 特権を保持しているだけでは、その特権を取り消すには十分でないことに注意してください。他のユーザーの CONTROL 特権を取り消すには、SYSADM または DBADM 権限が必要です。 DBADM 権限を取り消すには、SYSADM 権限が必要です。特権を取り消すことができるのは、既存のオブジェクトについてだけです。

注: DBADM 権限または CONTROL 特権を持っていないユーザーは、 WITH GRANT OPTION を使用して GRANT した特権を取り消すことはできません。ま

た、取り消された人から付与された特権を受け取っているユーザーに対する取り消しには、連鎖はありません。
明示的に付与された表 (またはビュー) に対する特権が DBADM 権限によってユーザーから取り消される場合、その表に定義されている他のビューに対する特権は取り消されることは**ありません**。ビューに対する特権は DBADM 権限によって使用可能になるものであり、基本表の明示特権とは関係ないからです。

同じ名前のユーザーとグループの両方に特権が付与されている場合、特権を取り消す時に、GROUP と USER キーワードのどちらかを指定する必要があります。次の例は、HERON というユーザーの EMPLOYEE 表に対する SELECT 特権を取り消すものです。

```
REVOKE SELECT
   ON EMPLOYEE FROM USER HERON
```

次の例は、HERON というグループの EMPLOYEE 表に対する SELECT 特権を取り消すものです。

```
REVOKE SELECT
   ON EMPLOYEE FROM GROUP HERON
```

1 つのグループから特権を取り消しても、そのグループに属するすべてのメンバーからその特権が取り消されるとは限らないことに注意してください。個別の名前が特権を直接付与されている場合は、その特権が直接取り消されるまで保持されます。

表特権がユーザーから取り消される場合、取り消された表特権に依存するそのユーザーによって作成されたすべてのビューに対する特権も取り消されます。ただし、システムによって暗黙に付与された特権のみが取り消されます。ビューに対する特権が別のユーザーによって直接付与された場合、その特権は引き続き保持されます。

表特権がユーザーから取り消される場合、取り消された表特権に依存するそのユーザーによって作成されたすべてのビューに対する特権も取り消されます。ただし、システムによって暗黙に付与された特権のみが取り消されます。ビューに対する特権が別のユーザーによって直接付与された場合、その特権は引き続き保持されます。

特権をグループに付与してから、そのグループの 1 人のメンバーだけから特権を取り消すという状況があります。エラー・メッセージ SQL0556N を受け取らないでこれを行うには、次の 2 つの方法だけを行ってください。

- グループからそのメンバーを除去します。あるいは、メンバーを減らして新規グループを作成し、その新規グループに特権を付与します。
- グループから特権を取り消してから、個々のユーザー (許可 ID) に特権を付与します。

注: 表またはビューに対する CONTROL 特権がユーザーから取り消された場合でも、そのユーザーは、その特権を他のユーザーに付与する能力は持ち続けます。CONTROL 特権が与えられると、そのユーザーは他の WITH GRANT OPTION 特権もすべて受け取ります。 CONTROL が取り消されても、他の特権のすべては、それらが明示して取り消されるまで、 WITH GRANT OPTION のまま残されます。

取り消された特権に依存しているすべてのパッケージは無効と見なされますが、十分な権限を持つユーザーによって再バインドされると再び有効になります。特権が後で再びアプリケーションをバインドしたユーザーに付与される場合、パッケージも再作成することができます。そのアプリケーションを実行すると、暗黙の再バインドが正常に実行されるトリガーとなります。特権が PUBLIC から取り消された場合、 PUBLIC 特権に基づいたバインドしかできないユーザーによってバインドされていたすべてのパッケージが無効にされます。ユーザーの持つ DBADM 権限が取り消されると、そのユーザーによってバインドされたパッケージはすべて無効になります。データベース・ユーティリティーに関連するパッケージも例外ではありません。無効のマークが付けられているパッケージを使用しようとすると、システムは、そのパッケージの再バインドを試みます。この再バインドの試みが失敗すると、エラー (SQLCODE -727) が発生します。この場合、それらのパッケージを明示的に再バインドするには、以下の権限が必要です。

- それらのパッケージを再バインドするための権限
- それらのパッケージ内で使われているオブジェクトに対する該当する権限

そうしたパッケージの再バインドは、特権を取り消す時に行うべきです。

1 つまたは複数の特権に基づいてトリガーまたは SQL 関数を定義した場合、これらの特権のいくつかを失うと、そのトリガーまたは SQL 関数を使用できなくなります。

オブジェクトの作成とドロップによる暗黙許可の管理

データベース・マネージャーは、表、パッケージなどのデータベース・オブジェクトを作成するユーザーに対して、いくつかの特権を暗黙的に付与します。特権は、SYSADM または DBADM 権限を持つユーザーによってオブジェクトが作成されるときにも付与されます。同様に、オブジェクトをドロップすると特権はドロップされます。

作成されるオブジェクトが、表、ニックネーム、索引、またはパッケージであれば、ユーザーにはそのオブジェクトに対する CONTROL 特権が与えられます。オブジェクトがビューの場合、そのビューに対する CONTROL 特権が暗黙のうちに付与されるのは、ユーザーがそのビュー定義の中で参照されるすべての表、ビュー、およびニックネームに対する CONTROL 特権を持っている場合に限られます。

明示的に作成されたオブジェクトがスキーマである場合、そのスキーマの所有者には、 WITH GRANT OPTION によって ALTERIN、CREATEIN、および DROPIN 特権が与えられます。暗黙に作成されたスキーマは、PUBLIC に付与された CREATEIN 特権を持ちます。

パッケージの所有権の確立

BIND および PRECOMPILE コマンドは、アプリケーション・パッケージを作成または変更します。どちらのコマンドでも、 OWNER オプションを使って結果パッケージの所有者の名前を付けてください。

パッケージの所有者の命名には、単純なルールがあります。

- どのユーザーも、自分を所有者として命名できます。 OWNER オプションが指定されていない場合、これがデフォルトです。

- SYSADM または DBADM 権限を持つ ID は、OWNER オプションを使って、任意の許可 ID を所有者として命名することができます。

DB2 データベース製品を使用してパッケージをバインドできるすべてのオペレーティング・システムが、 OWNER オプションをサポートしているわけではありません。

パッケージを通した暗黙特権

アプリケーション・プログラムによって、および対話式ワークステーション・セッションを操作しているユーザーによって、データベース内のデータに対するアクセスが要求されることがあります。パッケージに含まれているステートメントによって、ユーザーは多数のデータベース・オブジェクトに対してさまざまなアクションを実行できます。そのような各アクションには、1 つまたは複数の特権が必要です。

パッケージをバインドしている個別ユーザーおよび PUBLIC に付与される特権、および個人および PUBLIC に付与されたロールに付与される特権は、静的 SQL および XQuery ステートメントのバインド時の許可検査で使用されます。グループを通して付与された特権と、グループに対して付与されたロールは、静的 SQL および XQuery ステートメントがバインドされるときの許可検査には使用されません。有効な *authID* を持ち、パッケージをバインドするユーザーは、パッケージのバインド時に VALIDATE RUN が指定されている場合を除いて、そのパッケージ内の静的 SQL または XQuery ステートメントの実行に必要なすべての特権を明示的に付与されているか、あるいは PUBLIC、PUBLIC に付与されたロール、またはユーザーに付与されたロールを通して、必要な特権を暗黙で付与されていなければなりません。 BIND 時に VALIDATE RUN を指定すると、そのパッケージ内のどの静的 SQL または XQuery ステートメントに許可の障害が発生したとしても BIND は失敗せず、その SQL または XQuery ステートメントは実行時に再び有効になります。ユーザーがパッケージをバインドするのに適した許可 (BIND または BINDADD 特権) を持っているかどうか検査を行う場合には、PUBLIC、グループ、ロール、およびユーザーの特権がすべて使用されます。

パッケージには、静的と動的の両方の SQL および XQuery ステートメントが入っていることがあります。静的照会を含むパッケージを処理する場合、ユーザーに必要なのはパッケージについての EXECUTE 特権だけです。したがってそのユーザーは、パッケージ内の静的照会では、パッケージ・バインド・プログラムの特権を暗黙で取得することができます。ただし、これはパッケージに対して定められた制限の範囲内に限られます。

動的 SQL または XQuery ステートメントがパッケージに含まれる場合、必要な特権は、パッケージのプリコンパイル時またはバインド時に DYNAMICRULES に指定された値に応じて異なります。詳しくは、動的照会に対する DYNAMICRULES の影響について説明したトピックを参照してください。

ニックネームが定義されているパッケージ経由の間接特権

パッケージにニックネームへの参照が含まれる場合、パッケージ作成者およびパッケージ・ユーザーの許可処理はもう少し複雑です。パッケージ作成者がニックネームを含むパッケージを正常にバインドする場合、ニックネームがデータ・ソースで

参照する表およびビューに関して、認証検査または特権検査をパスする必要はありません。しかし、パッケージの実行者は、データ・ソースで認証および許可検査をパスすることが必要です。

たとえば、パッケージ作成者の .SQC ファイルに、複数の SQL または XQuery ステートメントが入っているとします。 1 つの静的ステートメントはローカル表を参照します。別の動的ステートメントはニックネームを参照します。パッケージがバインドされると、ローカル表およびニックネームの特権を検証するためにパッケージ作成者の許可 ID が使用されます。しかし、ニックネームが識別するデータ・ソース・オブジェクトに関しては何も検査されません。別のユーザーが、そのパッケージ用の EXECUTE 特権があることを前提としてパッケージを実行する場合、そのユーザーは表を参照するステートメントへの付加的な特権検査をパスする必要はありません。しかし、ニックネームを参照するステートメントの場合は、パッケージを実行するユーザーはデータ・ソースで認証検査と特権検査をパスすることが必要です。

.SQC ファイルに、動的 SQL および XQuery ステートメントと、表とニックネームの参照の混合のみが含まれる場合、ローカル・オブジェクトとニックネームへの DB2 データベース許可検査は類似しています。パッケージ・ユーザーは、ステートメント内のすべてのローカル・オブジェクト (表、ビュー) に関する特権検査をパスしなければならず、さらにニックネーム・オブジェクトの特権検査もパスする必要があります (パッケージ・ユーザーは、ニックネームが識別するオブジェクトを含むデータ・ソースで認証および特権検査をパスしなければなりません)。どちらの場合も、パッケージのユーザーには EXECUTE 特権が必要です。

パッケージの実行者の ID およびパスワードは、すべてのデータ・ソース認証、および特権処理に使用されます。この情報は、ユーザー・マッピングの作成によって変更できます。

注: 静的 SQL および XQuery ステートメントにニックネームは指定できません。DYNAMICRULES オプション (BIND に設定) を、ニックネームを含むパッケージで使用しないでください。

DB2 データベースは DB2 ファミリーのデータ・ソースと通信するときに動的 SQL を使用するため、ニックネームを含むパッケージが付加的な許可ステップを必要とする可能性があります。データ・ソースでパッケージを実行する許可 ID には、そのデータ・ソースでパッケージを動的に実行するための適切な権限が必要です。

ビューを使用したデータ・アクセスの制御

ビューを使用すれば、表に対するアクセス制御や特権付与を行うことができます。

ビューを使用すると、表に対する以下のようなアクセス制御が可能になります。

- 表内の指定した列だけにアクセスを制限する

  表内の特定の列だけに対するアクセスが必要なユーザーやアプリケーション・プログラムのために、許可されたユーザーは、必要な列だけを指定したビューを作成できます。
- 表内の行のサブセットだけにアクセスを制限する

許可されたユーザーは、ビュー定義の副照会の中に WHERE 節を指定することにより、ビューによってアクセスする行を限定できます。

- データ・ソース表またはビュー内の行または列のサブセットだけにアクセスを制限します。ニックネームによってデータ・ソースにアクセスしている場合、ニックネームを参照するローカル DB2 データベース・ビューを作成できます。これらのビューは、1 つまたは複数のデータ・ソースからニックネームを参照することができます。

  注: 複数のデータ・ソースを参照するニックネームを含むビューを作成できるので、ユーザーは 1 つのビューから複数のデータ・ソースのデータにアクセスできます。これらのビューは、*マルチ・ロケーション・ビュー* と呼ばれます。このようなビューは、分散環境全体で重要な表の列の情報を結合する場合や、個々のユーザーに、特定のオブジェクトに対してデータ・ソースで必要な特権がない場合に役立ちます。

ビューを作成するには、SYSADM 権限、DBADM 権限が必要です。または、そのビュー定義の中で参照されるそれぞれの表、ビュー、またはニックネームに対する CONTROL 特権あるいは SELECT 特権が必要です。さらに、ユーザーは、そのビュー用に指定されたスキーマ内にオブジェクトを作成できなければなりません。つまり、スキーマがまだ存在していなければ、既存のスキーマに対する CREATEIN 特権、または、データベースに対する IMPLICIT_SCHEMA 権限が必要です。

ニックネームを参照するビューを作成するとき、ビューでニックネームが参照するデータ・ソース・オブジェクト (表とビュー) に対する付加的な権限は必要ありません。ただし、ビューのユーザーがビューにアクセスするとき、基礎となるデータ・ソース・オブジェクトに対する SELECT 権限または同等の権限レベルが必要です。

ユーザーに、基礎となるオブジェクト (表およびビュー) に対してデータ・ソースでの適切な権限がない場合、次のことを実行できます。

1. データ・ソース表の中のユーザーのアクセスを許可する列に対してデータ・ソース・ビューを作成する
2. このビューの SELECT 特権をユーザーに付与する
3. ビューを参照するためのニックネームを作成する

その後、新しいニックネームを参照する SELECT ステートメントを発行することによって、列にアクセスすることができます。

以下のシナリオは、情報へのアクセスを制限するために、ビューを使用する方法をより詳細に示した例です。

次のようなさまざまな理由から、 STAFF 表の情報にアクセスする必要のある人が多数いるとします。例:

- 人事部では、表全体についての更新と参照ができなければなりません。

  この要件を満たすのに必要なことは、次のようにして、 PERSONNL グループに STAFF 表に対する SELECT 特権と UPDATE 特権を付与するだけです。

  ```
  GRANT SELECT,UPDATE ON TABLE STAFF TO GROUP PERSONNL
  ```

- 各部署のマネージャーは、部下の給与についての情報を参照する必要があります。

これは、各部署のマネージャーごとに、専用のビューを作成することによって解決できます。たとえば、51 番の部署のマネージャーに対しては、次のようにビューを作成できます。

```
CREATE VIEW EMP051 AS
   SELECT NAME,SALARY,JOB FROM STAFF
   WHERE DEPT=51
GRANT SELECT ON TABLE EMP051 TO JANE
```

JANE という許可名を持つマネージャーは、STAFF 表と同じように EMP051 ビューを照会します。このマネージャーが STAFF 表の EMP051 というビューにアクセスするとき、表示される情報は次のようになります。

名前	SALARY	JOB
Fraye	45150.0	Mgr
Williams	37156.5	Sales
Smith	35654.5	Sales
Lundquist	26369.8	Clerk
Wheeler	22460.0	Clerk

- すべてのユーザーは他の従業員の場所を知る必要があります。この要件を満たすには、次のようにして、 STAFF 表の NAME 列と ORG 表の LOCATION 列についてのビューを作成し、それぞれ DEPT 列と DEPTNUMB 列に基づいて 2 つの表を結合します。

  ```
  CREATE VIEW EMPLOCS AS
     SELECT NAME, LOCATION FROM STAFF, ORG
     WHERE STAFF.DEPT=ORG.DEPTNUMB
  GRANT SELECT ON TABLE EMPLOCS TO PUBLIC
  ```

  従業員の場所に関するビューにアクセスするユーザーは、次の情報を見ることになります。

名前	LOCATION
Molinare	New York
Lu	New York
Daniels	New York
Jones	New York
Hanes	Boston
Rothman	Boston
Ngan	Boston
Kermisch	Boston
Sanders	Washington
Pernal	Washington
James	Washington
Sneider	Washington
Marenghi	Atlanta
O'Brien	Atlanta
Quigley	Atlanta
Naughton	Atlanta

名前	LOCATION
Abrahams	Atlanta
Koonitz	Chicago
Plotz	Chicago
Yamaguchi	Chicago
Scoutten	Chicago
Fraye	Dallas
Williams	Dallas
Smith	Dallas
Lundquist	Dallas
Wheeler	Dallas
Lea	San Francisco
Wilson	San Francisco
Graham	San Francisco
Gonzales	San Francisco
Burke	San Francisco
Quill	Denver
Davis	Denver
Edwards	Denver
Gafney	Denver

SYSADM 権限と DBADM 権限を保持するユーザーによるアクセスの制御

SYSADM 権限と DBADM 権限を保持するユーザーによるデータ・アクセスをモニターしたり制御したりすることが必要な場合もあります。

システム管理者とデータベース管理者によるアクセスをモニターして制御するには、以下の手順を実行します。

1. SYSADM 権限と DBADM 権限を保持するユーザーに関して、どのイベントをキャプチャーするのかを決めて、そのイベントをモニターするための監査ポリシーを作成します。
2. その監査ポリシーを SYSADM 権限と DBADM 権限に関連付けます。
3. ロールを作成し、そのロールに DBADM 権限を付与します。
4. トラステッド・コンテキストを定義し、そのロールをこのトラステッド・コンテキストのデフォルトのロールとして設定します。

   そのロールのメンバーシップをいずれかの許可 ID に明示的に付与する設定は、しないでください。これを行なうと、そのロールはこのトラステッド・コンテキストでのみ使用できるようになり、ユーザーが DBADM 権限を取得できるのも、そのトラステッド・コンテキストの中にいる場合に限られることになります。

注: このオプションは、SYSADM 権限を保持するユーザーに対する保護にはなりません。そのようなユーザーは、暗黙的な DBADM 権限を保持しているからです。

5. そのトラステッド・コンテキストに対するユーザー・アクセスを制御するには、2 つの方法があります。
   - 暗黙アクセス: 各ユーザーごとに固有のトラステッド・コンテキストを作成します。そのトラステッド・コンテキストの属性に合致した通常の接続をユーザーが確立すると、そのユーザーは暗黙的に信頼され、そのロールに対するアクセスを取得します。
   - 明示アクセス: WITH USE FOR 節を使用してトラステッド・コンテキストを作成し、そのトラステッド・コンテキストにアクセスできるすべてのユーザーを定義します。それらのユーザーがデータベース要求を実行するためのアプリケーションを作成します。そのアプリケーションが明示的なトラステッド接続を確立して、ユーザーが要求を送信すると、そのアプリケーションはそのユーザー ID に切り替えて、そのユーザーとしてデータベース要求を実行します。
6. トラステッド・コンテキストのユーザーに関して、どのイベントをキャプチャーするのかを決めて、そのイベントをモニターするための監査ポリシーを作成し、そのポリシーをトラステッド・コンテキストに関連付けます。
7. 機密データがある場合は、EXECUTE カテゴリーをモニターするための監査ポリシーを作成し、そのポリシーを、モニター対象の機密データが含まれている表に関連付けます。EXECUTE カテゴリーをモニターすれば、それらの表にアクセスするすべての照会が (だれがその照会を実行したかにかかわりなく) キャプチャーされます。

注: SYSADM 権限と DBADM 権限を保持するユーザーが表のデータにアクセスすることを明示的に禁止する場合は、機密データが含まれている表で LBAC (ラベル・ベースのアクセス制御) セキュリティー・メカニズムを使用することを検討してください。

データ暗号化

(ここで説明する) ストレージ内のデータを暗号化するには、暗号化および暗号化解除組み込み関数である ENCRYPT、DECRYPT_BIN、DECRYPT_CHAR、および GETHINT を使用することができます。クライアントと DB2 データベースの間で転送中のデータを暗号化するには、DATA_ENCRYPT および SERVER_ENCRYPT 認証タイプを使用するか、または Secure Socket Layer (SSL) 用の DB2 データベース・システム・サポートを使用できます。

ENCRYPT 組み込み関数は、パスワード・ベースの暗号化方式によってデータを暗号化します。これらの関数によって、パスワード・ヒントをカプセル化することもできます。パスワード・ヒントが、暗号化データに組み込まれます。暗号化したデータの暗号化を解除するには、正しいパスワードを使用する必要があります。これらの関数を使用する開発者は、パスワードを忘れた場合の管理や使用できないデータの管理について考慮しなければなりません。

ENCRYPT 関数の結果は、VARCHAR FOR BIT DATA です (32631 という制限があります)。

暗号化できるデータは、CHAR、VARCHAR、および FOR BIT DATA だけです。

DECRYPT_BIN および DECRYPT_CHAR 関数は、パスワード・ベースの暗号化解除を使用して、データの暗号化を解除します。

DECRYPT_BIN は常に VARCHAR FOR BIT DATA を戻し、 DECRYPT_CHAR は常に VARCHAR を戻します。最初の引数が CHAR FOR BIT DATA または VARCHAR FOR BIT DATA の可能性があるため、結果が最初の引数と異なる場合もあります。

結果の長さは、次の 8 バイト境界までのバイト数に依存します。オプションのヒント・パラメーターが指定されている場合、結果の長さは、データ引数の長さプラス 40 に、次の 8 バイト境界までのバイト数を加えた長さになる可能性があります。オプションのヒント・パラメーターが指定されない場合、結果の長さは、データ引数の長さプラス 8 に、次の 8 バイト境界までのバイト数を加えた長さになる可能性があります。

GETHINT 関数は、カプセル化されたパスワード・ヒントを返します。パスワード・ヒントとは、データ所有者がパスワードを思い浮かべるために役立つフレーズです。例えば、パスワード ″Pacific″ を思い浮かべるために、ワード 『Ocean』 をヒントとして使用することができます。

データの暗号化に使用するパスワードは、以下のいずれかの方法で決定されます。

- パスワード引数。パスワードは、ENCRYPT 関数が呼び出されるときに明示的に渡されるストリングです。データは、与えられたパスワードで暗号化および暗号化解除されます。
- 暗号化パスワード特殊レジスター。 SET ENCRYPTION PASSWORD ステートメントはパスワード値を暗号化し、その暗号化されたパスワードをデータベース・マネージャーに送信して、特殊レジスターに保管します。パスワード・パラメーターなしで呼び出された ENCRYPT、DECRYPT_BIN、および DECRYPT_CHAR 関数は、 ENCRYPTION PASSWORD 特殊レジスターの値を使用します。ENCRYPTION PASSWORD 特殊レジスターは、暗号化形式でのみ保管されます。

  特殊レジスターの初期値 (デフォルト値) は空ストリングです。

パスワードに有効な長さは 6 から 127 文字です。ヒントに有効な長さは 0 から 32 文字です。

DB2 インスタンスの Secure Socket Layer (SSL) サポートの構成

DB2 データベース・システムは、SSL をサポートしています。したがって、IBM DB2 Driver for JDBC and SQLJ を使用するクライアント・アプリケーションは、SSL ソケットを使用して DB2 データベースに接続できます。DB2 インスタンスで SSL サポートを有効にするには、**DB2COMM** レジストリー変数を SSL に設定し、SSL 構成ファイルを作成し、インスタンスを再始動します。

SSL サポートを構成する前に、以下のようにします。

- Windows では、**PATH** 環境変数に、Linux と UNIX では、**LIBPATH**、**SHLIB_PATH**、**LD_LIBRARY_PATH** のいずれかの環境変数に、GSKit ライブラリーのパスが設定されていることを確認します。
- 接続コンセントレーターがアクティブになっていないことを確認します。接続コンセントレーターの実行中は、DB2 インスタンスで SSL サポートが有効になりません。

  接続コンセントレーターがアクティブになっているかどうかを確認するには、GET DATABASE MANAGER CONFIGURATION コマンドを実行します。構成パラメーター **MAX_CONNECTIONS** が **MAX_COORDAGENTS** の値より大きい値に設定されている場合は、接続コンセントレーターがアクティブになっています。

SSL は、IBM DB2 Driver for JDBC and SQLJ (タイプ 4 の接続) と DB2 データベース製品の間の通信でサポートされています。

DB2 データ・サーバーに対応した SSL サポートが組み込まれているサポート・プラットフォームは、以下のとおりです。

- AIX
- Itanium ベースの HP Integrity Series システム (IA-64) 上の HP-UX
- x86、x64、IA64、64 ビット POWER™ サーバー、64ビット zSeries または System z9™ 上の Linux
- x64 上の Solaris
- 32 ビット、x64、Itanium ベース・システム上の Windows

SSL 通信は、常に FIPS モードになります。

中間サーバー・マシンで DB2 Connect または DB2 Enterprise Edition を使用して、DB2 クライアントをホストまたは iSeries™ のデータベースに接続する場合は、ゲートウェイの DB2 データベース製品とホストまたは iSeries のデータベースの間で SSL サポートを使用できません。ただし、その同じ状況でも、DB2 クライアントの IBM DB2 Driver for JDBC and SQLJ (タイプ 4 の接続) とゲートウェイの DB2 データベース製品の間では、SSL サポートを使用できます。

DB2 インスタンスで SSL サポートを構成するには、以下のようにします。

1. DB2 インスタンス所有者としてログインします。
2. SSL 構成ファイルを作成します。
   - Linux および UNIX:*INSTHOME*/cfg/SSLconfig.ini
   - Windows: *INSTHOME*/SSLconfig.ini

   INSTHOME は、インスタンスのホーム・ディレクトリーです。

   SSLconfig.ini ファイルには、機密データが含まれている場合があるので、そのファイルへのアクセスを制限するためのファイル許可を設定することをお勧めします。例えば、そのファイルにパスワードや鍵ストアが含まれている場合は、そのファイルの読み取り権限と書き込み権限を SYSADM グループのメンバーだけに制限してください。

3. SSL 構成ファイルに SSL パラメーターを追加します。 SSLconfig.ini ファイルには、SSL をロードして開始するための SSL パラメーターが含まれています。SSL パラメーターのリストは、以下のとおりです。

表 3. SSL 構成ファイルの SSL パラメーター

SSL パラメーターの名前	説明
DB2_SSL_KEYSTORE_FILE	サーバー証明書が格納されている鍵ストアの完全修飾ファイル名。
DB2_SSL_KEYSTORE_PW	サーバー証明書が格納されている鍵ストアのパスワード。
DB2_SSL_KEYSTORE_LABEL	サーバー証明書のラベル。
DB2_SSL_LISTENER	SSL リスナーのサービス名またはポート番号。

注:

- **DB2_SSL_KEYSTORE_PW** は、NULL 可能です。鍵ストア・ファイルのパスワードが必要ない場合は、省略してもかまいません。
- **DB2_SSL_KEYSTORE_LABEL** パラメーターを省略すると、デフォルトのサーバー証明書が使用されます。デフォルトのサーバー証明書が存在しなければ、SSL のセットアップは失敗します。
- **DB2_SSL_LISTENER** パラメーターでは、**SVCENAME** データベース・マネージャー構成パラメーターで使用されている値とは異なる値を使用する必要があります。DB2 インスタンスを開始しようとしたときに、SSL と TCP/IP の両方が同じポート番号を聴取していると、SQL5043N エラーが発生します。

SSLconfig.ini ファイルの例を以下に示します。

```
DB2_SSL_KEYSTORE_FILE=/home/test1/GSKit/Keystore/key.kdb
DB2_SSL_LISTENER=20397
DB2_SSL_KEYSTORE_PW=aaa111
```

4. **DB2COMM** レジストリー変数に値 SSL を追加します。 例:

```
db2set -i db2inst1 DB2COMM=SSL
```

db2inst1 は、DB2 インスタンス名です。 データベース・マネージャーは、同時に複数のプロトコルをサポートできます。例えば、通信プロトコルとして TCP/IP と SSL の両方を有効にするには、以下のようにします。

```
db2set -i db2inst1 DB2COMM=SSL,TCPIP
```

5. DB2 インスタンスを再始動します。 例:

```
db2stop
db2start
```

DB2 アクティビティーの監査

DB2 監査機能の紹介

機密データへのアクセスを管理するためには、さまざまな認証およびアクセス制御メカニズムを使用して、規則を確立し、認識済みの受け入れ可能なデータ・アクセス動作を制御することができます。しかし、認識されていない動作や受け入れることのできない動作からデータを保護したり、そのような動作を発見したりするため

には、データ・アクセスをモニターすることも必要です。このタスクを支援するため、DB2 データベース・システムには、監査機能が用意されています。

不適切なデータ・アクセスを正常にモニターして分析することにより、データ・アクセスの制御を改善し、データへの悪意のあるまたは不注意な無許可アクセスを最終的に防止することができます。システム管理処置を含む、アプリケーションおよび個々のユーザー・アクセスのモニターは、データベース・システムのアクティビティーの履歴レコードを提供します。

DB2 監査機能は、事前に定義したデータベースのイベントに対して監査証跡を生成し、かつその監査証跡を保存できるようにします。この機能により生成されたレコードは、監査ログ・ファイルに保持されます。これらのレコードを分析することで、使用パターンが明らかになり、システム誤用を識別することができます。そのようなシステム誤用を識別した場合、それを制限または除去できます。

監査機能は、インスタンス・レベルと個々のデータベース・レベルの両方で監査を行い、すべてのインスタンス・レベルおよびデータベース・レベルのアクティビティーを、それぞれ別々のログに分けて記録する能力を備えています。システム管理者 (インスタンス・レベルで SYSADM 権限を保持するユーザー) は、db2audit ツールを使用して、インスタンス・レベルで監査を構成したり、いつそのような監査情報を収集するかを制御したりすることができます。 db2audit ツールでは、システム管理者がインスタンス監査ログとデータベース監査ログの両方をアーカイブできるほか、アーカイブされているいずれかのタイプのログから監査データを抽出することもできます。

セキュリティー管理者 (データベース・レベルで SECADM 権限を持つユーザー) は、監査ポリシーと SQL ステートメント AUDIT を組み合わせて使用することによって、個々のデータベースごとに監査要件を構成および制御できます。セキュリティー管理者は、SYSPROC.AUDIT_ARCHIVE ストアード・プロシージャー、SYSPROC.AUDIT_LIST_LOGS 表関数、およびSYSPROC.AUDIT_DELIM_EXTRACT ストアード・プロシージャーを使用して、監査ログをアーカイブし、必要な情報が含まれているログを探索し、分析のために区切りファイルにデータを抽出することができます。

パーティション・データベース環境において作動しているとき、ユーザーが接続しているデータベース・パーティション (コーディネーター・パーティション)、またはカタログ・パーティション (同じデータベース・パーティションではない場合) で、多数の監査可能なイベントが発生します。つまり、監査レコードが複数のデータベース・パーティションによって生成される可能性があります。それぞれの監査レコードの中には、コーディネーター・パーティションと起点パーティション (監査レコードの起点となっているパーティション) を識別する情報が含まれています。

インスタンス・レベルでは、監査機能は、db2audit start および db2audit stop コマンドを使用することによって明示的に停止および開始する必要があります。インスタンス・レベルの監査を開始する際、監査機能は既存の監査構成情報を使用します。監査機能は DB2 データベース・サーバーから独立しているので、インスタンスが停止されてもアクティブのままです。実際、インスタンスが停止されるとき、

監査レコードが監査ログに生成される可能性があります。データベース・レベルで監査を開始する場合は、監査の対象とするオブジェクトに監査ポリシーを関連付けます。

監査レコードの区分

生成される監査レコードにはさまざまな区分があります。監査するために使用可能なイベントの区分に関する以下の記述では、各区分の名前に続いて、区分タイプの識別に使用される 1 つの単語のキーワードがあることに注意してください。監査のために使用可能なイベントの区分は次のようになります。

- 監査 (AUDIT)。監査設定が変更されるとき、または監査ログがアクセスされるときにレコードを生成する。
- 許可検査 (CHECKING)。 DB2 データベース・オブジェクトや関数に対するアクセス試行または操作試行の許可検査のときに、レコードを生成する。
- オブジェクト保守 (OBJMAINT)。データ・オブジェクトを作成またはドロップするとき、および特定のオブジェクトに変更を加えるときにレコードを生成する。
- セキュリティー保守 (SECMAINT)。以下の場合にレコードを生成する。
  - データベース特権やデータベース権限を付与または取り消すとき
  - セキュリティー・ラベルや免除を付与または取り消すとき
  - LBAC セキュリティー・ポリシーのグループ許可、ロール許可、あるいはオーバーライドまたは制限属性を変更するとき
  - SETSESSIONUSER 特権を付与または取り消すとき
  - DBADM 権限や SECADM 権限を付与または取り消すとき
  - SYSADM_GROUP、SYSCTRL_GROUP、SYSMAINT_GROUP、または SYSMON_GROUP 構成パラメーターのいずれかに変更を加えるとき
- システム管理 (SYSADMIN)。 SYSADM、SYSMAINT、または SYSCTRL 権限を必要とする操作が実行されるとき、レコードを生成する。
- ユーザー検証 (VALIDATE)。ユーザーを認証しているとき、またはシステムのセキュリティー情報を検索しているときにレコードを生成する。
- 操作コンテキスト (CONTEXT)。データベースの操作が実行されるとき、操作コンテキストを表示するレコードを生成する。この区分を使用すると、監査ログ・ファイルのより良い変換処理を可能にします。ログのイベント相関関係子フィールドを同時に使用することで、イベントのグループを 1 つのデータベース操作に戻って関連付けることができます。例えば、動的照会の照会ステートメント、静的照会のパッケージ ID、つまり CONNECT のような実行されている操作タイプのインディケーターは、監査結果を分析しているときに必要なコンテキストを提供できます。

  注: 操作コンテキストを提供する SQL または XQuery ステートメントは、かなり長くなる可能性があり、CONTEXT レコード内にすべてが示されます。このため、CONTEXT レコードが非常に大きくなる可能性があります。
- 実行 (EXECUTE)。SQL ステートメントの実行中にレコードを生成する。

上記の区分のいずれにおいても、失敗、成功、またはその両方を監査できます。

データベース・サーバーに対して何らかの操作が行われる場合、いくつかのレコードが生成されることがあります。監査ログに生成されるレコードの実際の数は、監査機能構成により指定された、記録されるイベントの区分の数によって決定されます。さらに、成功、失敗、またはその両方を監査するかどうかによっても異なります。このため、監査対象のイベントの選択は重要です。

監査ポリシー

セキュリティー管理者は、監査ポリシーを使用することによって、必要なデータやオブジェクトに関する情報だけを収集するように監査システムを構成できます。

セキュリティー管理者は、監査ポリシーを作成して、個々のデータベース内で何を監査の対象にするかを制御することができます。監査ポリシーを関連付けることができるオブジェクトには、以下のものがあります。

- データベース全体

  データベース内で発生する監査可能イベントすべてが、監査ポリシーに従って監査の対象となります。

- 表

  表 (非型付き)、MQT (マテリアライズ照会表)、またはニックネームへのデータ操作言語 (DML) アクセスおよび XQUERY アクセスは、すべて監査の対象となります。表がアクセス中になっているときは、ポリシーが他にも監査の対象となる区分を指示している場合でも、EXECUTE 区分の監査イベント (データを伴う場合と伴わない場合があります) のみが生成されます。

- トラステッド・コンテキスト

  特定のトラステッド・コンテキストで定義されたトラステッド接続内で発生する監査可能イベントすべてが、監査ポリシーに従って監査の対象となります。

- ユーザー、グループ、またはロールを示す許可 ID

  指定されたユーザーによって開始される監査可能イベントすべてが、監査ポリシーに従って監査の対象となります。

  指定されたグループやロールに属しているユーザーが開始する監査可能イベントは、すべて監査ポリシーに従って監査の対象となります。他のロールやグループを介する場合など、間接的にロールに属している場合も、監査の対象になります。

  同様なデータのキャプチャーは、グループのワークロードを定義してアクティビティーの詳細をキャプチャーすることにより、Work Load Management イベント・モニターでも行うことができます。なお、ワークロードのマッピングには、許可 ID だけではなく、属性も関係する場合があるので注意が必要です。このことが原因となって、監査の際に期待した細分度が得られなかったり、それらの他の属性が変更された場合には接続が別の (おそらく、モニターされていない) ワークロードにマップされてしまったりすることがあります。監査用のソリューションを使用するなら、ユーザー、グループ、またはロールの監査を確実に行うことができます。

- 権限 (SYSADM、SECADM、DBADM、SYSCTRL、SYSMAINT、SYSMON)

指定された権限を持つユーザーによって開始される監査可能イベントすべてが、そのイベントにその権限が必要であるかどうかにかかわりなく、監査ポリシーに従って監査の対象となります。なお、監査ポリシーが DBADM 権限に関連付けられている場合には、SYSADM 権限を持つユーザーも、すべてこのポリシーに従って監査の対象になります。これは、SYSADM 権限を持つユーザーは DBADM 権限も持っているものと見なされるためです。

セキュリティー管理者は、複数の監査ポリシーを作成できます。例えば、会社は、機密データを監査するためのポリシーと、DBADM 権限を持つユーザーのアクティビティーを監査するためのポリシーを必要とするかもしれません。1 つのステートメントで複数の監査ポリシーが有効になれば、それぞれの監査ポリシーで監査が必要とされるイベントすべてを (1 回の監査だけで) 監査することができます。例えば、データベースの監査ポリシーで特定の表における成功した EXECUTE イベントの監査が必要とされ、ユーザーの監査ポリシーで同じ表における EXECUTE イベントの失敗の監査が必要とされる場合は、その表へのアクセスの試行が成功した場合と失敗した場合の両方の監査が行われます。

特定のオブジェクトごとに有効にできる監査ポリシーは、1 つだけです。例えば、同じ表に複数の監査ポリシーを同時に関連付けることはできません。

監査ポリシーは、ビューや型付き表には関連付けることができません。監査ポリシーが関連付けられている表にアクセスするビューは、基礎表のポリシーに従って、監査の対象となります。

表に適用される監査ポリシーが、その表に基づいている MQT に自動的に適用されることはありません。表に監査ポリシーを関連付けている場合は、その表に基づいているすべての MQT に同じポリシーを関連付けてください。

トランザクション中に実行される監査は、監査ポリシーと、トランザクション開始時の関連付けに基づいて実行されます。例えば、セキュリティー管理者が、あるユーザーに監査ポリシーを関連付けていて、監査の時点でそのユーザーがトランザクションに入っている場合、監査ポリシーは、そのトランザクション内で実行される残りのステートメントには一切影響を与えません。また、監査ポリシーに対する変更は、コミットされるまで有効になりません。セキュリティー管理者が ALTER AUDIT POLICY ステートメントを発行する際は、そのステートメントがコミットされるまで、変更は有効になりません。

セキュリティー管理者は、監査ポリシーの作成には CREATE AUDIT POLICY ステートメントを、監査ポリシーの変更には ALTER AUDIT POLICY ステートメントを使用します。これらのステートメントでは、以下を指定できます。

- 監査の対象とするイベントの状況値: None、Success、Failure、または Both。

  指定された状況値と一致する監査可能イベントだけが監査の対象になります。
- 監査中にエラーが発生した場合のサーバーの振る舞い。

セキュリティー管理者は、AUDIT ステートメントを使用することにより、現行サーバーにおいて、現行のデータベースと監査ポリシーまたはデータベース・オブジェクトと監査ポリシーを関連付けることができます。オブジェクトが使用中になっているときは常に、この監査ポリシーに従って監査が行われます。

監査ポリシーを削除する場合、セキュリティー管理者は DROP ステートメントを使用します。監査ポリシーは、何らかのオブジェクトに関連付けられているとドロップできません。AUDIT REMOVE ステートメントを使用して、残っているオブジェクトとの関連付けをすべて除去してください。監査ポリシーにメタデータを追加する場合、セキュリティー管理者は COMMENT ステートメントを使用します。

完全な接続が確立される前に生成されるイベント

接続とユーザー切り替え操作の過程で生成されるいくつかのイベントの場合、利用できる監査ポリシー情報は、データベースに関連付けられているポリシーだけになります。このようなイベントを、次の表に示します。

表 4. 接続イベント

イベント	監査区分	コメント
CONNECT	CONTEXT	
CONNECT_RESET	CONTEXT	
AUTHENTICATION	VALIDATE	これには、トラステッド接続内でのユーザーの接続および切り替えの両方における認証が含まれます。
CHECKING_FUNC	CHECKING	試行されるアクセスは SWITCH_USER です。

これらのイベントは、データベースに関連付けられている監査ポリシーに基づいてのみ監査され、ユーザー、ユーザーのグループ、または権限などのそれ以外のオブジェクトに関連付けられている監査ポリシーでは監査されません。接続の過程で発生する CONNECT および AUTHENTICATION イベントについては、データベースがアクティブになるまで、インスタンス・レベルの監査設定が使用されます。データベースは、最初の接続の過程で、または ACTIVATE DATABASE コマンドが発行されたときにアクティブになります。

ユーザー切り替えの影響

トラステッド接続内でユーザーの切り替えが行われる場合、元のユーザーの名残が後に残ることはありません。このような場合、元のユーザーに関連付けられていた監査ポリシーは考慮されなくなり、新しいユーザーに合わせて適合する監査ポリシーが再評価されます。なお、トラステッド接続に関連付けられている監査ポリシーは引き続き有効です。

SET SESSION USER ステートメントが使用される場合は、セッション許可 ID だけが切り替わります。元のユーザーの許可 ID (システム許可 ID) の監査ポリシーは有効なまま残り、それに加えて新しいユーザーの監査ポリシーも使用されます。セッション内で複数の SET SESSION USER ステートメントが発行される場合は、元のユーザー (システム許可 ID) と現行ユーザー (セッション許可 ID) に関連付けられている監査ポリシーだけが考慮されます。

データ定義言語に関する制約事項

以下のデータ定義言語 (DDL) ステートメントは、AUDIT 排他 SQL ステートメントと呼ばれます。

- AUDIT
- CREATE AUDIT POLICY、ALTER AUDIT POLICY、および DROP AUDIT POLICY
- DROP ROLE および DROP TRUSTED CONTEXT (ドロップされるロールまたはトラステッド・コンテキストが監査ポリシーに関連付けられている場合)

AUDIT 排他 SQL ステートメントには、使用に際していくつかの制約事項があります。

- 各ステートメントの後には COMMIT または ROLLBACK を発行する必要があります。
- これらのステートメントを XA トランザクションなどのグローバル・トランザクション内で発行することはできません。

コミットされていない AUDIT 排他 DDL ステートメントは、全パーティションを通じて同時に 2 つ以上存在してはなりません。コミットされていない 1 つの AUDIT 排他 DDL ステートメントが実行されている間は、後続の AUDIT 排他 DDL ステートメントは現行の AUDIT 排他 DDL ステートメントがコミットまたはロールバックされるまで待機します。

注: 変更はカタログに記録されますが、COMMIT が発行されるまでは、ステートメントを発行した接続においても、変更は有効になりません。

特定の表に対するアクセスすべてを監査する例

EMPLOYEE 表に極めて機密性の高い情報が含まれていて、その表のデータに対するありとあらゆる SQL アクセスを監査しようとしている企業について考えます。表に対する全アクセスのトラッキングには、EXECUTE 区分を使用できます。この区分では、SQL ステートメント、およびオプションとしてそのステートメントの実行時に提供される入力データの値を監査できます。

表でのアクティビティーのトラッキングには、2 つのステップがあります。まず最初に、セキュリティー管理者は EXECUTE 区分を指定する監査ポリシーを作成します。そして次に、そのポリシーを表に関連付けます。

```
CREATE AUDIT POLICY SENSITIVEDATAPOLICY
    CATEGORIES EXECUTE STATUS BOTH ERROR TYPE AUDIT
COMMIT

AUDIT TABLE EMPLOYEE USING POLICY SENSITIVEDATAPOLICY
COMMIT
```

SYSADM または DBADM によるアクションすべてを監査する例

セキュリティー準拠の証明を完成させるには、企業は、システム管理 (SYSADM) またはデータベース管理 (DBADM) 権限を持つユーザーによるデータベース内でのありとあらゆるアクティビティーがモニター可能であることを示す必要があります。

データベース内でのすべてのアクションをキャプチャーするためには、EXECUTE 区分と SYSADMIN 区分の両方を監査する必要があります。セキュリティー管理者は、これらの 2 つの区分を監査する監査ポリシーを作成します。セキュリティー管理者は、AUDIT ステートメントを使用して、この監査ポリシーを SYSADM 権限や

DBADM 権限に関連付けることができます。そして、SYSADM または DBADM 権限を保持しているすべてのユーザーについては、すべての監査可能イベントのログが記録されるようになります。次の例は、このような監査ポリシーを作成して、それを SYSADM および DBADM 権限に関連付ける方法を示しています。

```
CREATE AUDIT POLICY ADMINSPOLICY CATEGORIES EXECUTE STATUS BOTH,
   SYSADMIN STATUS BOTH ERROR TYPE AUDIT
COMMIT
AUDIT SYSADM, DBADM USING POLICY ADMINSPOLICY
COMMIT
```

特定のロールによるアクセスすべてを監査する例

自社の Web アプリケーションが自社の企業データベースにアクセスできるようにしている企業があります。その Web アプリケーションを使用している個人については、厳密なことがわかりません。使用されるロールだけがわかっており、そのロールを使用してデータベース許可を管理しています。企業は、そのロールに属している個人がデータベースにサブミットしている要求を調べて、それらのユーザーが Web アプリケーション以外からデータベースにアクセスしていないかどうかを確認するために、そのロールに属しているすべてのユーザーのアクションをモニターすることを望んでいます。

EXECUTE 区分には、この状況でユーザーのアクティビティーをトラッキングするために必要なレベルの監査が含まれています。最初のステップは、次のように、適切な監査ポリシーを作成して、Web アプリケーションで使用されるロール (この例では、TELLER と CLERK のロール) にその監査ポリシーを関連付けます。

```
CREATE AUDIT POLICY WEBAPPPOLICY CATEGORIES EXECUTE WITH DATA
   STATUS BOTH ERROR TYPE AUDIT
COMMIT
AUDIT ROLE TELLER, ROLE CLERK USING POLICY WEBAPPPOLICY
COMMIT
```

監査ログの保管と分析

システム管理者は、db2audit configure コマンドを使用して、アクティブ監査ログやアーカイブされた監査ログのパスを構成できます。監査ログをアーカイブすると、アクティブ監査ログはアーカイブ・ディレクトリーに移され、サーバーは新しいアクティブ監査ログの書き込みを開始します。これによって、オフラインで監査ログを保管することができ、必要になるときまでデータを抽出する必要はありません。セキュリティー管理者やシステム管理者がログをアーカイブした後は、ログから区切りファイルにデータを抽出できます。区切りファイルのデータは、分析用の DB2 データベース表にロードできます。

監査ログのロケーションを構成すると、監査ログを大容量の高速なディスクに置くことができ、データベース・パーティション・フィーチャー (DPF) のインストール済み環境では、オプションでノードごとにディスクを分けることも可能です。 DPF 環境では、アクティブ監査ログのパスを、ノードごとに固有のディレクトリーにすることができます。ノードごとに固有のディレクトリーを使用すると、各ノードが別々のディスクに書き込みを行うため、ファイルの競合を防ぐのに役立ちます。

Windowsオペレーティング・システムでの監査ログのデフォルト・パスは *instance*¥security¥auditdata で、Linux および UNIX オペレーティング・システムでのデフォルト・パスは *instance*/security/auditdata です。デフォルトのロケーションを

使用することを望まない場合は、別のディレクトリーを選択できます (代替のロケーションとして使用するディレクトリーがまだない場合は、ご使用のシステムに新規ディレクトリーを作成できます)。アクティブ監査ログのロケーションとアーカイブされた監査ログのロケーションのパスを設定するには、次の例のように、datapath および archivepath パラメーターを指定した db2audit configure コマンドを使用します。

```
db2audit configure datapath /auditlog archivepath /auditarchive
```

db2audit を使用して設定された監査ログの保管場所は、インスタンス内のすべてのデータベースに適用されます。

注: サーバー上に複数のインスタンスが存在する場合、各インスタンスはそれぞれ別個のデータおよびアーカイブ・パスを持つべきです。

DPF 環境におけるアクティブ監査ログのパス (datapath)

DPF 環境では、各パーティションで同じアクティブ監査ログのロケーション (**datapath** パラメーターで設定される) を使用する必要があります。これを実現する方法は 2 つあります。

1. **datapath** パラメーターを指定する際に、データベース・パーティション式を使用します。データベース・パーティション式を使用すると、監査ログ・ファイルのパスにパーティション番号を含めることができるため、データベース・パーティションごとに異なるパスになります。
2. すべてのノードで同一の共用ドライブを使用します。

データベース・パーティション式は、**datapath** パラメーターに指定する値の中のどの位置でも使用できます。例えば、3 つのノードを持つ、データベース・パーティション番号が 10 のシステムで、次のコマンドを実行します。

```
db2audit configure datapath '/pathForNode $N'
```

すると、以下のファイルが作成されます。

- /pathForNode10
- /pathForNode20
- /pathForNode30

注: アーカイブ・ログ・ファイルのパス (**archivepath** パラメーター) の指定にデータベース・パーティション式を使用することはできません。

アクティブ監査ログのアーカイブ

db2audit ツールでは、システム管理者がインスタンス監査ログとデータベース監査ログの両方をアーカイブできるほか、アーカイブされているいずれかのタイプのログから監査データを抽出することもできます。アクティブ監査ログをアーカイブするには、セキュリティー管理者は SYSPROC.AUDIT_ARCHIVE ストアード・プロシージャーを使用します。セキュリティー管理者は、ログからデータを抽出して区切りファイルにロードするには、SYSPROC.AUDIT_DELIM_EXTRACT ストアード・プロシージャーを使用できます。

監査ログをアーカイブおよび抽出するためには、セキュリティー管理者は以下のステップを実行する必要があります。

1. アプリケーションがストアード・プロシージャー SYSPROC.AUDIT_ARCHIVE を使用してアクティブ監査ログのアーカイブを定期的に実行するように、スケジュールを作成します。
2. どのアーカイブされたログ・ファイルについて調べるかを決めます。SYSPROC.AUDIT_LIST_LOGS 表関数を使用して、アーカイブされた監査ログをすべてリストします。
3. ログからデータを抽出して区切りファイルにロードするため、ファイル名を SYSPROC.AUDIT_DELIM_EXTRACT ストアード・プロシージャーにパラメーターとして渡します。
4. 監査データを分析用の DB2 データベース表にロードします。

アーカイブされたログ・ファイルは、すぐに分析用の表にロードする必要はなく、将来の分析のために保管しておくことができます。例えば、企業の監査が行われるときにしか閲覧する必要がないということもあるかもしれません。

アーカイブの途中で、アーカイブ・パスのディスク・スペースがいっぱいになってしまった、あるいはアーカイブ・パスが存在しない、などの問題が発生した場合は、アーカイブ・プロセスは失敗し、監査ログ・データのパスに .bk という拡張子を持つ中間ログ・ファイルが生成されます (例えば db2audit.instance.log.0.20070508172043640941.bk)。 (アーカイブ・パスに十分なディスク・スペースを割り振ることによって、またはアーカイブ・パスを作成することによって) 問題が解決されたなら、この中間ログをアーカイブ・パスに移動させてください。アーカイブ・パスに移動させた後は、この中間ログは正常にアーカイブされたログと同じ方法で扱うことができます。

DPF 環境におけるアクティブ監査ログのアーカイブ

DPF 環境では、インスタンスの実行中にアーカイブ・コマンドが発行されると、自動的にすべてのノードでアーカイブ・プロセスが実行されます。アーカイブされたログ・ファイルの名前には、すべてのノードで同じタイム・スタンプが使用されます。例えば、3 つのノードを持つ、データベース・パーティション番号が 10 のシステムで、次のコマンドを実行します。

```
db2audit archive to /auditarchive
```

すると、以下のファイルが作成されます。

- /auditarchive/db2audit.log.10.*timestamp*
- /auditarchive/db2audit.log.20.*timestamp*
- /auditarchive/db2audit.log.30.*timestamp*

アーカイブ・コマンドが発行されたときにインスタンスが実行中でない場合は、以下のいずれかの方法によって、どのノードでアーカイブを実行するかを制御できます。

- 現行ノードのみのアーカイブを実行する場合は、db2audit コマンドに node オプションを使用します。
- すべてのノードでアーカイブを実行する場合は、db2_all コマンドを使用します。

  例:

  ```
  db2_all db2audit archive node to /auditarchive
  ```

これにより、コマンドが呼び出されたノードを示すように DB2NODE 環境変数が設定されます。

あるいは別の方法として、各ノード別に個々のアーカイブ・コマンドを発行することもできます。例:

- ノード 10 で:

```
db2audit archive node 10 to /auditarchive
```

- ノード 20 で:

```
db2audit archive node 20 to /auditarchive
```

- ノード 30 で:

```
db2audit archive node 30 to /auditarchive
```

注: インスタンスが実行中でない場合、アーカイブされた監査ログ・ファイル名のタイム・スタンプは、ノードごとに異なります。

注: すべてのノードでアーカイブ・パスを共用することが勧められていますが、必須ではありません。

注: AUDIT_DELIM_EXTRACT ストアード・プロシージャーと AUDIT_LIST_LOGS 表関数でアクセスできるのは、現行の (コーディネーター) ノードから見えるアーカイブされたログ・ファイルだけです。

ログのアーカイブと表へのデータ抽出の例

監査データを確実にキャプチャーし、将来の使用に備えて保管しておくために、ある企業では、6 時間ごとに新しい監査ログを作成し、現行の監査ログを WORM ドライブにアーカイブする必要があります。この企業では、セキュリティー管理者によって 6 時間ごとに以下の SYSPROC.AUDIT_ARCHIVE ストアード・プロシージャーが発行されるよう、スケジュールを立てています。アーカイブされたログのパスはデフォルトのアーカイブ・パス /auditarchive で、アーカイブはすべてのノードで実行されます。

```
CALL SYSPROC.AUDIT_ARCHIVE( '/auditarchive', -2 )
```

セキュリティー手順の一環として、この企業は、監査データ内で注意を必要とする多くの疑わしい動作や許可されないアクティビティーを識別し、定義しています。この企業は、1 つ以上の監査ログからすべてのデータを抽出し、それをリレーショナル表に置き、SQL 照会を使用してこれらのアクティビティーを探すことを希望しています。監査する適切な区分を判別してあり、データベースや他のデータベース・オブジェクトには必要な監査ポリシーが関連付けられています。

例えば、SYSPROC.AUDIT_DELIM_EXTRACT ストアード・プロシージャーを呼び出し、デフォルトの区切り文字を使用して、2006 年 4 月のタイム・スタンプで作成されたすべてのノードのすべての区分について、アーカイブされた監査ログを抽出できます。

```
CALL SYSPROC.AUDIT_DELIM_EXTRACT(
     '', '', '/auditarchive',  'db2audit.%.200604%', '' )
```

別の例として、SYSPROC.AUDIT_DELIM_EXTRACT ストアード・プロシージャーを呼び出し、調べているタイム・スタンプが付いているファイルから、EXECUTE

区分の成功イベントについてのアーカイブされた監査レコードと、CHECKING 区分の失敗イベントについてのアーカイブされた監査レコードを抽出できます。

```
CALL SYSPROC.AUDIT_DELIM_EXTRACT( '', '', '/auditarchive',
    'db2audit.%.20060419034937', 'categories
     execute status success, checking status failure );
```

監査ログ・ファイル名:

監査ログ・ファイルの名前は、そのログがインスタンス・レベルのログかデータベース・レベルのログかを区別し、データベース・パーティション・フィーチャー (DPF) 環境では、そのログがどのパーティションで作成されたかを示します。アーカイブされた監査ログのファイル名には、いつアーカイブ・コマンドが実行されたのかを示すタイム・スタンプが付加されます。

アクティブ監査ログ・ファイル名

DPF 環境では、アクティブ監査ログのパスを各パーティションに固有のディレクトリーにして、各パーティションが個々のファイルに書き込みを行うようにすることができます。監査レコードの起源を正確に追跡するためには、監査ログ・ファイルの名前の一部にパーティション番号を含めます。例えば、パーティション 20 の場合、インスタンス・レベルの監査ログ・ファイル名は db2audit.instance.log.20 となります。このインスタンスの testdb というデータベースの監査ログ・ファイルは、db2audit.db.testdb.log.20 となります。

非 DPF 環境では、パーティション番号は 0 (ゼロ) と見なされます。この場合、インスタンス・レベルの監査ログ・ファイル名は db2audit.instance.log.0 になります。このインスタンスの testdb というデータベースの監査ログ・ファイルは、db2audit.db.testdb.log.0 になります。

アーカイブされた監査ログ・ファイルの名前

アクティブ監査ログのアーカイブ時には、ファイル名に *YYYYMMDDHHMMSS* (YYYY は年、MM は月、DD は日、HH は時間、MM は分、SS は秒) という形式で現行タイム・スタンプが付加されます。

アーカイブ監査ログのファイル名の形式は、監査ログのレベルによって異なります。

インスタンス・レベルのアーカイブされた監査ログ
インスタンス・レベルのアーカイブされた監査ログのファイル名は、db2audit.instance.log.*partition.YYYYMMDDHHMMSS* となります。

データベース・レベルのアーカイブされた監査ログ
データベース・レベルのアーカイブされた監査ログのファイル名は、db2audit.db*database*.log.*partition.YYYYMMDDHHMMSS* となります。

非 DPF 環境の場合は、*partition* の値は 0 (ゼロ) になります。

タイム・スタンプは、アーカイブ・コマンドが実行された時刻を示します。したがって、この時刻は、必ずしもログの最後のレコードの時刻を正確に反映しているわけではありません。アーカイブされた監査ログ・ファイルには、ログ・ファイル名のタイム・スタンプよりも数秒後のタイム・スタンプを持つレコードが含まれていることがあります。これは次のような理由によります。

- アーカイブ・コマンドが発行される際、監査機能は、すべての処理中のレコードの書き込みが完了するまで、アーカイブされたログ・ファイルの作成を待機します。
- 複数のマシンがある環境では、リモート・マシンのシステム時刻とアーカイブ・コマンドが発行されたマシンのシステム時刻が同期化されていないことがあります。

DPF 環境の場合は、アーカイブの実行時にサーバーが稼働していれば、パーティション間でタイム・スタンプが整合され、アーカイブが実行されたパーティションで生成されたタイム・スタンプがその他のパーティションにも反映されます。

DB2 監査データを保持する表の作成:

監査データをデータベース表で操作する前に、データを保持するための表を作成する必要があります。表のデータを無許可ユーザーから保護するために、これらの表を 1 つの別個のスキーマで作成することを考慮してください。

- スキーマの作成に必要な権限および特権については、 CREATE SCHEMA ステートメントの説明を参照してください。
- 表の作成に必要な権限および特権については、 CREATE TABLE ステートメントの説明を参照してください。
- 表を保持するために、どの表スペースを使用するかを決定します。 (このトピックでは、表スペースの作成方法については説明しません。)

注: 監査データを保持するために作成する必要がある表の形式は、リリースによって変わることがあります。新しい列が追加されたり、既存の列のサイズが変更されたりする場合もあります。スクリプト db2audit.ddl は、監査レコードを格納するための正しい形式の表を作成します。

以下のいくつかの例は、区切りファイルに含まれるレコードを保持するための表を作成する方法を示しています。必要に応じて、これらの表を格納するための 1 つの別個のスキーマを作成することもできます。

ファイルに含まれるすべてのデータを必ずしも使用する必要がない場合には、表の定義から列を省略するか、必要に応じて、特定の表の作成を回避することができます。表の定義から列を省略した場合、これらの表にデータをロードするためのコマンドを変更する必要があります。

1. db2 コマンドを発行して、 DB2 コマンド・ウィンドウをオープンします。
2. オプション。表を保持するためのスキーマを作成します。この例では、スキーマの名前は AUDIT です。

   ```
   CREATE SCHEMA AUDIT
   ```

3. オプション。 AUDIT スキーマを作成した場合には、表を作成する前に、そのスキーマに切り替えます。

   ```
   SET CURRENT SCHEMA = 'AUDIT'
   ```

4. スクリプト db2audit.ddl を実行して、監査レコードを格納する表を作成します。

   スクリプト db2audit.ddl は、sqllib/misc ディレクトリー (Windows の場合は sqllib¥misc ディレクトリー) にあります。このスクリプトは、データベースへの接続が存在しており、8K の表スペースが使用可能であることを前提としていま

す。このスクリプトを実行するためのコマンドは db2 +o -tf sqllib/misc/db2audit.ddl です。スクリプトによって作成される表は AUDIT、CHECKING、OBJMAINT、SECMAINT、SYSADMIN、 VALIDATE、CONTEXT および EXECUTE です。

5. 表を作成した後、セキュリティー管理者なら SYSPROC.AUDIT_DELIM_EXTRACT ストアード・プロシージャーを使用して、あるいはシステム管理者なら db2audit extract コマンドを使用して、アーカイブされた監査ログ・ファイルから区切りファイルに監査レコードを抽出することができます。区切りファイルの監査データは、ここで作成したデータベース表にロードすることができます。

DB2 監査データの表へのロード:

監査ログ・ファイルを区切り文字付きファイルにアーカイブして抽出し、監査データを保持するためのデータベース表を作成した後に、監査データを分析のために区切り文字付きファイルからデータベース表にロードすることができます。

ロード・ユーティリティーを使用して、監査データを表にロードします。それぞれの表ごとに、別個のロード・コマンドを発行してください。表定義から 1 つまたは複数の列を省略した場合には、データを正常にロードするために、LOAD コマンドの内容を変更する必要があります。さらに、監査データの抽出時にデフォルト以外の区切り文字を指定した場合にもまた、使用する LOAD コマンドのバージョンを変更する必要があります。

1. db2 コマンドを発行して、 DB2 コマンド・ウィンドウをオープンします。
2. AUDIT 表をロードするために、以下のコマンドを発行します。

   ```
   LOAD FROM audit.del OF DEL MODIFIED BY DELPRIORITYCHAR LOBSINFILE
            INSERT INTO schema.AUDIT
   ```

   注: DELPRIORITYCHAR 修飾子を指定して、バイナリー・データが正しく構文解析されるようにします。

   注: LOAD コマンドの LOBSINFILE オプションを指定します (ラージ・オブジェクトのインライン・データは 32 KB に制限されるという制約があるため)。状況によっては、LOBS FROM オプションも使用する必要がある場合があります。

   注: ファイル名を指定するときには、絶対パスを使用してください。例えば、Windows ベースのコンピューターの C: ドライブに DB2 データベース・システムがインストールされている場合、audit.del ファイルの完全修飾ファイル名として、C:¥Program Files¥IBM¥SQLLIB¥*instance*¥security¥audit.del と指定します。
3. CHECKING 表をロードするために、以下のコマンドを発行します。

   ```
   LOAD FROM checking.del OF DEL MODIFIED BY DELPRIORITYCHAR LOBSINFILE
            INSERT INTO schema.CHECKING
   ```

4. OBJMAINT 表をロードするために、以下のコマンドを発行します。

   ```
   LOAD FROM objmaint.del OF DEL MODIFIED BY DELPRIORITYCHAR LOBSINFILE
            INSERT INTO schema.OBJMAINT
   ```

5. SECMAINT 表をロードするために、以下のコマンドを発行します。

```
LOAD FROM secmaint.del OF DEL MODIFIED BY DELPRIORITYCHAR LOBSINFILE
          INSERT INTO schema.SECMAINT
```

6. SYSADMIN 表をロードするために、以下のコマンドを発行します。

```
LOAD FROM sysadmin.del OF DEL MODIFIED BY DELPRIORITYCHAR LOBSINFILE
          INSERT INTO schema.SYSADMIN
```

7. VALIDATE 表をロードするために、以下のコマンドを発行します。

```
LOAD FROM validate.del OF DEL MODIFIED BY DELPRIORITYCHAR LOBSINFILE
          INSERT INTO schema.VALIDATE
```

8. CONTEXT 表をロードするために、以下のコマンドを発行します。

```
LOAD FROM context.del OF DEL MODIFIED BY DELPRIORITYCHAR LOBSINFILE
          INSERT INTO schema.CONTEXT
```

9. EXECUTE 表をロードするために、以下のコマンドを発行します。

```
LOAD FROM execute.del OF DEL MODIFIED BY DELPRIORITYCHAR LOBSINFILE
          INSERT INTO schema.EXECUTE
```

10. これらの表へのデータのロードを完了したら、sqllib ディレクトリーの security/auditdata サブディレクトリーから .del ファイルを削除します。
11. 監査データを表にロードしたら、これらの表から分析のためにデータを選択する準備ができました。

表にあらかじめデータが入っていて、再びデータを入れたい場合には、既存の表データに新しい表データを追加するために、 INSERT オプションを使用してください。以前の db2audit extract 操作によるレコードを表から除去する場合には、REPLACE オプションを使って表を再びロードします。

監査のアーカイブおよび抽出のストアード・プロシージャー:

セキュリティー管理者は、SYSPROC.AUDIT_ARCHIVE ストアード・プロシージャー、SYSPROC.AUDIT_DELIM_EXTRACT ストアード・プロシージャー、SYSPROC.AUDIT_LIST_LOGS 表関数を使用して、セキュリティー管理者が現在接続しているデータベースの監査ログをアーカイブする操作や、データを区切りファイルに抽出する操作を実行できます。

セキュリティー管理者は、これらのストアード・プロシージャーや表関数を使用してデータベースの監査ログをアーカイブまたはリストするためには、データベースに接続している必要があります。

アーカイブしたファイルを別のデータベース・システムにコピーする場合で、そのアクセスにストアード・プロシージャーを使用することを望む場合は、データベース名が同じになるようにするか、同じデータベース名を含むようにファイルの名前を変更してください。

これらのストアード・プロシージャーや表関数では、インスタンス・レベルの監査ログはアーカイブまたはリストされません。インスタンス・レベルの監査ログをアーカイブおよび抽出するためには、システム管理者は db2audit コマンドを使用する必要があります。

セキュリティー管理者は、これらのストアード・プロシージャーや表関数を使用して、以下の操作を実行できます。

表 5. 監査システムのストアード・プロシージャー

ストアード・プロシージャーおよび表関数	操作	コメント
AUDIT_ARCHIVE	現行監査ログをアーカイブします。	アーカイブのパスを入力として取ります。アーカイブ・パスが指定されない場合、このストアード・プロシージャーは監査構成ファイルからアーカイブ・パスを取得します。 アーカイブは各ノードで実行され、同期化されたタイム・スタンプが監査ログ・ファイルのファイル名に付加されます。
AUDIT_LIST_LOGS	現行データベースの指定されたパスにある、アーカイブされた監査ログのリストを戻します。	

表 5. 監査システムのストアード・プロシージャー (続き)

ストアード・プロシージャーおよび表関数	操作	コメント
AUDIT_ DELIM_EXTRACT	バイナリー・アーカイブ・ログからデータを抽出し、区切りファイルにロードします。	抽出された監査レコードは、DB2 データベース表にロードするのに適した区切り形式のファイルに置かれます。出力は区分ごとに 1 つずつ独立したファイルに置かれます。加えて、監査データに含まれているラージ・オブジェクトを保持するためのファイル **auditlobs** が作成されます。ファイル名は次のとおりです。 • **audit.del** • **checking.del** • **objmaint.del** • **secmaint.del** • **sysadmin.del** • **validate.del** • **context.del** • **execute.del** • **auditlobs** ファイルが既に存在する場合は、そこに出力が追加されます。 **auditlobs** ファイルは、CONTEXT または EXECUTE 区分が抽出される場合に作成されます。抽出できるのは、現行データベースのアーカイブされた監査ログだけです。また、コーディネーター・ノードから見えるファイルだけが抽出されます。 アーカイブされた監査ログは、インスタンスの所有者にしか削除できません。

SQL ステートメントを監査するための EXECUTE 区分

EXECUTE 区分では、ユーザーが発行する SQL ステートメントを的確にトラッキングすることができます (バージョン 9.5 よりも前は、CONTEXT 区分を使用してこの情報を探さなければなりませんでした)。

この EXECUTE 区分は、SQL ステートメントのテキストに加えて、コンパイル環境や、後でステートメントを再現するのに必要なその他の値もキャプチャーします。例えば、ステートメントを再現すると、SELECT ステートメントがどの行を戻

したのかを正確に知ることができます。ステートメントを再実行するためには、まず、データベース表をステートメント発行時の状態にリストアする必要があります。

EXECUTE 区分を使用して監査を行う場合は、静的 SQL と動的 SQL の両方のステートメント・テキストが記録され、入力パラメーター・マーカーやホスト変数も記録されます。入力値を使用するかどうかにかかわらず、EXECUTE 区分を構成して監査できます。

注: グローバル変数は監査されません。

EXECUTE イベントの監査は、イベントの完了時に行われます (SELECT ステートメントの場合は、カーソルのクローズ時に行われます)。イベント完了時の状況も保管されます。EXECUTE イベントは完了時に監査されるため、長時間にわたって実行される照会は、すぐには監査ログに現れません。

注: ステートメントの準備は、実行の一部とは見なされません。ほとんどの許可検査は準備の際に実行されます (例えば SELECT 特権)。つまり、許可エラーが原因となって準備の途中で失敗するステートメントの場合は、EXECUTE イベントは生成されません。

特定の EXECUTE レコードごとに、「Statement Value Index」、「Statement Value Type」、および「Statement Value Data」フィールドが繰り返される場合があります。抽出によって生成されるレポート形式の場合は、各レコードに複数の値がリストされます。区切りファイル形式の場合は、複数の行が使用されます。最初の行には、STATEMENT のイベント・タイプが示され、値はありません。続く行には、SQL ステートメントに関連付けられているデータ値ごとに 1 行ずつ、DATA のイベント・タイプが示されます。STATEMENT 行と DATA 行は、イベント相関関係子とアプリケーション ID のフィールドを使用してリンクさせることができます。「Statement Text」、「Statement Isolation Level」、および「Compilation Environment Description」の列は、DATA イベントにはありません。

監査されるステートメント・テキストと入力データ値は、ディスクに保管される際、データベース・コード・ページに変換されます (監査されたフィールドはすべてデータベース・コード・ページで保管されます)。入力データのコード・ページとデータベース・コード・ページに互換性がなかった場合は、エラーは戻されず、変換されないままのデータが代わりにログとして記録されます。各データベースは独自の監査ログを持っているため、異なるコード・ページを持つデータベースがあっても、問題は起こりません。

ROLLBACK および COMMIT は、アプリケーションによって実行されたときに監査されます。また、BIND などの別のコマンドの一部として暗黙的に発行されたときも監査されます。

監査されている表へのアクセスがあったために EXECUTE イベントが監査された後、作業単位内で他にどのステートメントが実行されるかに影響を与えるステートメントがすべて監査されます。 COMMIT、ROLLBACK、ROLLBACK TO SAVEPOINT、および SAVEPOINT がこのようなステートメントに該当します。

「Savepoint ID」フィールド

どのステートメントが ROLLBACK TO SAVEPOINT ステートメントの影響を受けたかは、「Savepoint ID」フィールドを使用してトラッキングできます。通常の DML ステートメント (SELECT、INSERT など) では、現行のセーブポイント ID が監査されています。しかし、ROLLBACK TO SAVEPOINT ステートメントの場合は、代わりにロールバック先のセーブポイント ID が監査されます。したがって、次の例が示すように、ロールバック先の ID と同じかそれよりも大きなセーブポイント ID を持っているステートメントは、すべてロールバックされます。表は、ステートメントの実行の順序を示しています。2 以上のセーブポイント ID を持つイベントはすべてロールバックされます。値 3 (最初の INSERT ステートメントより) だけが、表 T1 に挿入されます。

表 6. ROLLBACK TO SAVEPOINT ステートメントの影響を示すためのステートメントのシーケンス

ステートメント	Savepoint ID
INSERT INTO T1 VALUES (3)	1
SAVEPOINT A	2
INSERT INTO T1 VALUES (5)	2
SAVEPOINT B	3
INSERT INTO T1 VALUES (6)	3
ROLLBACK TO SAVEPOINT A	2
COMMIT	

WITH DATA オプション

WITH DATA オプションが指定されている場合は、すべての入力値が監査されるわけではありません。LOB、LONG、XML、および構造化タイプのパラメーターは、NULL として示されます。

日付、時刻、およびタイム・スタンプのフィールドは、ISO 形式で記録されます。

あるポリシーで WITH DATA が指定されていて、SQL ステートメントの実行に関係するオブジェクトに関連付けられている別のポリシーで WITHOUT DATA が指定されている場合は、WITH DATA が優先され、その特定のステートメントに関してデータが監査されます。例えば、ユーザーに関連付けられている監査ポリシーでは WITHOUT DATA が指定されていて、表に関連付けられているポリシーでは WITH DATA が指定されている場合、そのユーザーがその表にアクセスするときは、ステートメントに使用される入力データが監査されます。

位置指定更新ステートメントや位置指定削除ステートメントでどの行が変更されたかは、判別できません。ログに記録されるのは基礎となる SELECT ステートメントの実行だけで、個々の FETCH は記録されません。ステートメントが発行されたときにカーソルがどの行にあったかを EXECUTE レコードから判別することは不可能です。後でステートメントを再現する際に唯一可能なのは、SELECT ステートメントを発行して、どの範囲の行が影響を受けている可能性があるかを確認することだけです。

過去のアクティビティーの再現の例

この例では、ある企業が、自社の包括的なセキュリティー・ポリシーの一環として、7 年前までさかのぼってデータベース内の特定の表に対する特定の要求の影響を分析できるようにしておく必要があるとします。これを行うために、この企業は、選択した任意の時点のデータベースを再構成できるような、週単位のバックアップをアーカイブするポリシーとそれに関連付けるログ・ファイルを設けます。関連する、リストアされたデータベースに対するあらゆる要求を再現し、分析できるようにしておくために、データベース監査には、データベースに対して行われたすべての要求に関する十分な情報をキャプチャーさせる必要があります。この要件は、静的 SQL ステートメントと動的 SQL ステートメントの両方を含みます。

この例は、SQL ステートメントの発行時に実施される必要のある監査ポリシーと、監査ログをアーカイブし、後でそれを抽出し、分析するためのステップを示しています。

1. EXECUTE 区分を監査する監査ポリシーを作成し、このポリシーをデータベースに適用します。

   ```
   CREATE AUDIT POLICY STATEMENTS CATEGORIES EXECUTE WITH DATA
      STATUS BOTH ERROR TYPE AUDIT
   COMMIT

   AUDIT DATABASE USING POLICY STATEMENTS
   COMMIT
   ```

2. 定期的に監査ログをアーカイブし、アーカイブ・コピーを作成します。

   一定の間隔 (ログに記録されるデータの量に応じて、週に一度、一日に一度など) で、セキュリティー管理者によって以下のステートメントが実行される必要があります。これらのアーカイブされたファイルは、どれほどの期間でも必要なだけ保持しておくことができます。2 つの入力パラメーターを使用して、プロシージャー SYSPROC.AUDIT_ARCHIVE が呼び出されます。1 つはアーカイブ・ディレクトリーのパス、もう 1 つはアーカイブをすべてのノードで実行することを示す -2 です。

   ```
   CALL SYSPROC.AUDIT_ARCHIVE( '/auditarchive', -2 )
   ```

3. セキュリティー管理者は、SYSPROC.AUDIT_LIST_LOGS 表関数を使用し、2006 年 4 月以降の入手可能な監査ログをすべて調べて、どのログに必要なデータが含まれている可能性があるかを判別します。

   ```
   SELECT FILE FROM TABLE(SYSPROC.AUDIT_LIST_LOGS('/auditarchive'))
      AS T WHERE FILE LIKE 'db2audit.dbname.log.0.200604%'
   FILE
   ----------------------------------------
   ...
   db2audit.dbname.log.0.20060418235612
   db2audit.dbname.log.0.20060419234937
   db2audit.dbname.log.0.20060420235128
   ```

4. この出力から、セキュリティー管理者は、必要なログが 1 つのファイル db2audit.dbname.log.20060419234937 に含まれていることを観察します。タイム・スタンプは、このファイルが、監査員が確認したいと思っている日の終わりにアーカイブされたファイルであることを示しています。

   セキュリティー管理者は、このファイル名を SYSPROC.AUDIT_DELIM_EXTRACT ストアード・プロシージャーの入力として

使用して、監査データを区切りファイルに抽出します。これらのファイルの監査データは、DB2 データベース表にロードできます。監査員は、そのデータベース表でデータを分析し、調べている特定のステートメントを見つけることができます。監査員が調べているのは 1 つの SQL ステートメントだけであったとしても、そのステートメントに何らかの影響を与えるステートメントがある場合は、その作業単位内の複数のステートメントを調べることが必要になることもあります。

5. ステートメントを再現するためには、セキュリティー管理者は以下のアクションを行う必要があります。
   - 発行する抽出ステートメントを監査レコードから判別する。
   - ステートメントを発行したユーザーを監査レコードから判別する。
   - ユーザーがステートメントを発行したときのユーザーの許可 (すべての LBAC 保護を含む) を正確に再作成する。
   - 監査レコードのコンパイル環境に関する列を SET COMPILATION ENVIRONMENT ステートメントに組み合わせて使用し、コンパイル環境を再現する。
   - ステートメントが発行されたときの正確な状態に、データベースをリストアする。

   実動システムに支障が出ないようにするため、データベースのリストアやステートメントの再現は、2 次データベース・システム上で行います。ステートメントを発行したユーザーとして実行中のセキュリティー管理者は、「Statement Value Data」エレメントに指定された任意の入力変数をもとにステートメント・テキストから検出されたステートメントを再発行できます。

監査機能の管理

監査機能の動作

このトピックでは、監査レコードをログに書き込むタイミングがデータベースのパフォーマンスにどのように影響を与えることがあるか、監査機能内で起こるエラーをどのように管理するか、およびさまざまな状況においてどのように監査レコードが生成されるかを理解するのに役立つ、背景情報を提供します。

アクティブ・ログに監査レコードを書き込むタイミングの制御

アクティブ・ログへの監査レコードの書き込みは、そのレコードの生成の原因となるイベントの発生と同期的に、または非同期的に行われます。 *audit_buf_sz* データベース・マネージャーの構成パラメーター値は、いつ監査レコードが書き込まれるかを決定します。

audit_buf_sz の値がゼロ (0) の場合、書き込みは同期的に行われます。監査レコードを生成しているイベントは、そのレコードがディスクに書き込まれるまで待機します。それぞれのレコードに関連した待機のために、DB2 データベースのパフォーマンスが低下します。

audit_buf_sz の値がゼロより大きい場合、レコードの書き込みは非同期で行われます。ゼロより大きいとき *audit_buf_sz* の値は、内部バッファーの作成に使用される 4 KB ページの数となります。ディスクに書き込む前の複数の監査レコードを維持

するために、内部バッファーが使用されます。監査イベントの結果として監査レコードを生成するステートメントは、レコードがディスクに書き込まれるまで待機することなく、動作を継続できます。

非同期の場合、空きがあるバッファーに監査レコードがしばらく保持される可能性があります。長時間にわたってこれが発生しないようにするために、データベース・マネージャーは定期的に監査レコードの書き込みを強制します。さらに、監査機能の許可ユーザーもまた、明示的な要求により監査バッファーをフラッシュすることができます。また、アーカイブ操作中は、バッファーが自動的にフラッシュされます。

エラーが発生したときには、同期のレコード書き込みか、非同期のレコード書き込みかによって、違いが出てきます。非同期モードでは、監査レコードがディスクに書き込まれる前にバッファーに入れられるので、いくつかのレコードが失われる可能性があります。同期モードでは、エラーのために書き込みできない監査レコードは多くて 1 つなので、失われる可能性があるレコードは 1 つだけです。

監査機能のエラーの管理

ERRORTYPE 監査機能パラメーターの設定により、どのように DB2 データベース・システムと監査機能の間でエラーを管理するかを制御します。監査機能がアクティブであり、監査機能パラメーター ERRORTYPE の設定が AUDIT であるとき、監査機能は DB2 データベースの他の部分と同じように扱われます。監査レコードを (同期モードの場合はディスクに、非同期モードの場合は監査バッファーに) 書き込まないと、ステートメントに関連した監査イベントは正常終了したものと見なされません。このモードの実行時にエラーが検出されると、監査レコードを生成するステートメントに関する負の SQLCODE がアプリケーションに戻されます。

エラー・タイプが NORMAL に設定されると、 db2audit からのエラーはすべて無視され、操作の SQLCODE が戻されます。

さまざまな状況で生成される監査レコード

API または照会ステートメントや監査設定に応じて、特定のイベントに対して監査レコードが何も生成されなかったり、1 つまたは複数の監査レコードが生成されたりします。例えば、SELECT 副照会による SQL UPDATE ステートメントの結果として、表の UPDATE 特権に対する許可検査の結果を含む 1 つの監査レコードと、表の SELECT 特権に対する許可検査の結果を含む別の監査レコードが生成される可能性があります。

動的なデータ操作言語 (DML) のステートメントの場合、ステートメントが準備される時点ですべての許可検査の監査レコードが生成されます。同一ユーザーによるステートメントの再使用では許可検査されないため、再度監査されることはありません。ただし、特権情報を含んでいるカタログ表のいずれかが変更された場合には、次の作業単位において、キャッシュされた動的 SQL または XQuery ステートメントのステートメント特権が再び検査され、1 つ以上の新規監査レコードが作成されます。

静的 DML ステートメントだけを含んでいるパッケージの場合、監査レコードを生成する可能性のある唯一の監査可能イベントは、ユーザーがそのパッケージを実行する特権を持っているかどうかの許可検査です。パッケージ内の静的 SQL または

XQuery ステートメント用に必要な許可検査および監査レコードの作成 (作成される場合) は、パッケージがプリコンパイルまたはバインドされるときに実行されます。パッケージ内部の静的 SQL または XQuery ステートメントの実行は、EXECUTE 区分を使用して監査可能です。ユーザーにより明示的に、またはシステムにより暗黙的に 1 つのパッケージが再びバインドされるとき、静的 SQL または XQuery ステートメントにより要求される許可検査のために複数の監査レコードが生成されます。

許可検査が実行時に行われるステートメント (例えば、データ定義言語 (DDL)、GRANT、および REVOKE ステートメント) の場合、これらのステートメントが使用されるときにはいつでも監査レコードが生成されます。

注: DDL を実行するとき、監査レコード内の (コンテキスト・イベントを除く) すべてのイベント用に記録されるセクション番号は、ステートメントの実際のセクション番号にかかわらずゼロ (0) にリセットされます。

監査機能のヒントと技法

監査ログのアーカイブ

監査ログは定期的にアーカイブする必要があります。監査ログをアーカイブすると、現行監査ログはアーカイブ・ディレクトリーに移され、サーバーは新しいアクティブ監査ログの書き込みを開始します。各アーカイブ・ログの名前には、後で分析するために情報を必要とするログ・ファイルを識別するのに役立つタイム・スタンプが含まれています。

長期保管の場合、アーカイブ・ファイルのグループを zip 圧縮できます。

必要でなくなったアーカイブ監査ログについては、インスタンス所有者はファイルをオペレーティング・システムから削除するだけで済みます。

エラー処理

監査ポリシーを作成するときに、単にテスト監査ポリシーを作成しているのでなければ、エラー・タイプ AUDIT を使用する必要があります。例えば、エラー・タイプが AUDIT に設定されているときにディスク・スペースの不足などのエラーが発生すると、エラーが戻されます。エラー条件を修正してからでなければ、監査可能アクションをそれ以上続行できません。一方、エラー・タイプが NORMAL に設定されている場合、ロギングが失敗するだけで、エラーはユーザーに戻されません。エラーが発生しなかったかのように操作は続行します。

アーカイブの途中で、アーカイブ・パスのディスク・スペースがいっぱいになってしまった、あるいはアーカイブ・パスが存在しない、などの問題が発生した場合は、アーカイブ・プロセスは失敗し、監査ログ・データのパスに .bk という拡張子を持つ中間ログ・ファイルが生成されます (例えば db2audit.instance.log.0.20070508172043640941.bk)。 (アーカイブ・パスに十分なディスク・スペースを割り振ることによって、またはアーカイブ・パスを作成することによって) 問題が解決されたなら、この中間ログをアーカイブ・パスに移動させてください。アーカイブ・パスに移動させた後は、この中間ログは正常にアーカイブされたログと同じ方法で扱うことができます。

DDL ステートメントの制限

AUDIT 排他 SQL ステートメントと呼ばれるいくつかのデータ定義言語 (DDL) ステートメントは、次の作業単位まで有効ではありません。そのため、以下の各ステートメントの直後に COMMIT ステートメントを使用することが推奨されています。

AUDIT 排他 SQL ステートメントは以下のとおりです。

- AUDIT
- CREATE AUDIT POLICY、ALTER AUDIT POLICY、および DROP AUDIT POLICY
- DROP ROLE および DROP TRUSTED CONTEXT (ドロップされるロールまたはトラステッド・コンテキストが監査ポリシーに関連付けられている場合)

アーカイブ・データを保持するための表形式は変わることがある

セキュリティー管理者なら SYSPROC.AUDIT_DEL_EXTRACT ストアード・プロシージャーを使用して、あるいはシステム管理者なら db2audit extract コマンドを使用して、アーカイブされた監査ログ・ファイルから区切りファイルに監査レコードを抽出することができます。区切りファイルのデータは、分析用の DB2 データベース表にロードできます。監査データを保持するために作成する必要がある表の形式は、リリースによって変わることがあります。

重要: スクリプト db2audit.ddl は、監査レコードを格納するための正しい形式の表を作成します。列が追加されたり、既存の列のサイズが変更されたりする場合もあるので、リリースごとに db2audit.ddl を実行するようにしなければなりません。

CHECKING イベントの使用

CHECKING イベントを操作しているとき、ほとんどの場合、監査レコードのオブジェクト・タイプのフィールドは、オブジェクトにアクセスしようとするユーザー ID が必要な特権または権限を持っているか検査する対象のオブジェクトとなります。例えば、ユーザーが 1 列を追加することによって表を ALTER しようとする場合、CHECKING イベントの監査レコードは、試みたアクセスが 『ALTER』 であり、検査されているオブジェクト・タイプが 『TABLE』 (検査されているのは表特権なので、列ではない) であったことを示します。

ただし、オブジェクトの CREATE、BIND、または削除をユーザー ID に許可するデータベース権限があるかどうか検査する場合には、データベースに対して検査が行われますが、オブジェクト・タイプ・フィールドは (データベース自体ではなく) 作成、バインド、またはドロップする対象のオブジェクトを指定します。

表に索引を作成するとき、索引の作成特権が必要です。このため、 CHECKING イベントの監査レコードのアクセス試行タイプは 『CREATE』 ではなく 『INDEX』 になります。

パッケージのバインドについて作成される監査レコード

既存のパッケージをバインドしているとき、パッケージの DROP に対して OBJMAINT イベントの監査レコードが作成されます。さらに、パッケージの新しい

コピーの CREATE に対して別の OBJMAINT イベントの監査レコードが作成されます。

ROLLBACK 後の CONTEXT イベント情報の使用

データ定義言語 (DDL) は、正常であるとしてログに記録される OBJMAINT または SECMAINT イベントを生成する可能性があります。しかし、イベントのロギングの後、エラーが原因で ROLLBACK が発生するかもしれません。その場合、オブジェクトが作成されないか、 GRANT または REVOKE 操作が不完全になります。このような場合に、CONTEXT イベントの使用が重要となります。このような CONTEXT イベントの監査レコード (特にイベントを終了するステートメント) は、操作試行の完了状態を示します。

ロード区切り文字

監査レコードを DB2 データベース表にロードするためにふさわしい区切り形式に抽出しているとき、ステートメントのテキスト・フィールド内で使用される区切り文字を明確に指定しなければなりません。区切りファイルを抽出する際、これは、次のコマンドを使用して行うことができます。

```
db2audit extract delasc delimiter <load delimiter>
```

この *load delimiter* は、単一文字 (例えば ″) または 16 進数の値で示される 4 バイト・ストリング (例えば 『0xff』) です。有効なコマンドの例は次のとおりです。

```
db2audit extract delasc
db2audit extract delasc delimiter !
db2audit extract delasc delimiter 0xff
```

抽出時に区切り文字としてデフォルトのロード区切り文字以外を使用した場合、LOAD コマンドでは MODIFIED BY オプションを使用する必要があります。区切り文字として使用される ″0xff″ を指定した LOAD コマンド例の一部分を次に示します。

```
db2 load from context.del of del modified by chardel0xff replace into ...
```

これにより、デフォルトのロード文字ストリング区切り文字、″ (二重引用符) がオーバーライドされます。

第 2 章 ロール

ロールは、特権の管理を簡素化します。つまり、グループと同等の機能は提供されますが、同じ制約事項は設けられません。ロールは、1 つ以上の特権をまとめたデータベース・オブジェクトですが、これを、GRANT ステートメントを使用してユーザー、グループ、PUBLIC、またはその他のロールに割り当てるか、あるいは、CREATE TRUSTED CONTEXT または ALTER TRUSTED CONTEXT ステートメントを使用して、トラステッド・コンテキストに割り当てることができます。ワークロード定義内で、SESSION_USER ROLE 接続属性用のロールを指定することができます。

ロールは、次のように、データベース・システム内での特権の管理がより簡単になるという利点を備えています。

- セキュリティー管理者は、組織の構造を映し出すような方法で、そのデータベースへのアクセスを制御することができます (組織における役職や担当業務に対応したロールをデータベース内に作成できます)。
- ユーザーには、その役職や担当業務に応じたロールに対するメンバーシップが付与されます。ユーザーの役職や担当業務の変更に応じて、ロールに対するメンバーシップを簡単に付与または取り消すことができます。
- 特権の割り当てが簡素化されます。管理者は、特定の役職や担当業務に該当する個々のユーザーに一連の同じ特権を付与するのではなく、その役職や担当業務に応じたロールに対してこの一連の特権を付与してから、その役職や担当業務に該当する各ユーザーにそのロールを付与することができます。
- そのロールを付与されたすべてのユーザーに対し、更新が適用されます。つまり管理者は、個人ごとに各ユーザーの特権を更新する必要はありません。
- ビュー、トリガー、マテリアライズ照会表 (MQT)、静的 SQL、および SQL ルーチンの作成時には、ロールに対して付与された特権および権限が常に使用されます。この場合、グループに付与された特権および権限は (直接でも間接にでも) 使用されません。

  その理由は、グループはサード・パーティー・ソフトウェア (例えば、オペレーティング・システムまたは LDAP ディレクトリー) によって管理されるので、DB2 データベース・システムは、グループ内のメンバーシップがいつ変更になったかを判別できないからです。ロールはデータベース内部で管理されるので、DB2 データベース・システムは、許可がいつ変更されたかを判別して、それに応じたアクションをとることができます。グループが検討の対象にならないのと同じ理由で、グループに付与されたロールも検討の対象にはなりません。
- ユーザーに割り当てられたすべてのロールは、ユーザーが接続を確立したときに有効になるので、ロールに付与されたすべての特権と許可も、ユーザーが接続するときに有効となります。ロールを明示的に有効または無効にすることはできません。
- セキュリティー管理者は、ロールの管理を他人に委任することができます。

セキュリティー管理者 (SECADM) 権限を例外として、データベース内で付与できるどの DB2 特権および権限でも、ロールに付与することができます。例えば、以下のどの権限および特権でも、ロールに付与することができます。

- DBADM、LOAD、および IMPLICIT_SCHEMA データベース権限
- CONNECT、CREATETAB、CREATE_NOT_FENCED、BINDADD CREATE_EXTERNAL_ROUTINE、または QUIESCE_CONNECT データベース権限
- 任意のデータベース・オブジェクト特権 (CONTROL を含む)

ユーザーがデータベースに接続したとき、そのユーザーのロールは自動的に有効になり、許可の検討の対象になります。つまり、SET ROLE ステートメントを使用してロールを活動化する必要はありません。例えば、ビュー、マテリアライズ照会表 (MQT)、トリガー、パッケージ、または SQL ルーチンを作成すると、ロールを通して取得した特権が適用されます。ただし、自分がメンバーとして所属するグループに付与されたロールを通して取得した特権は適用されません。

ロールには所有者はいません。セキュリティー管理者は、GRANT ステートメントの WITH ADMIN OPTION 節を使用して、ロールの管理を別のユーザーに委任することができます。それによって、他のユーザーがロールのメンバーシップを制御できるようになります。

制約事項

ロールの使用に関しては、次のようないくつかの制約事項があります。

- ロールはデータベース・オブジェクトを所有できません。
- ロールには、セキュリティー管理者 (SECADM) 権限を付与できません。
- 以下のデータベース・オブジェクトの作成時には、グループに付与された許可およびロールは検討の対象にはなりません。
  - 静的 SQL を格納するパッケージ。
  - ビュー
  - マテリアライズ照会表 (MQT)
  - トリガー
  - SQL ルーチン

  オブジェクトを作成するユーザーに対してか、または PUBLIC に対して直接または間接的に (例えばロール階層を介して) 付与されたロールだけが、上記のオブジェクトの作成時に検討の対象になります。

ロールのメンバーシップの作成と付与

セキュリティー管理者は、ロールの作成、ドロップ、付与、取り消し、およびコメント作成の権限を保有します。セキュリティー管理者は GRANT (ロール) ステートメントを使用して、ロール内のメンバーシップを許可 ID に付与し、REVOKE (ロール) ステートメントを使用して、ロール内のメンバーシップを許可 ID から取り消します。

セキュリティー管理者は、WITH ADMIN OPTION を持ったロール内のメンバーシップを許可 ID に付与することで、ロール内のメンバーシップの管理をその許可 ID

に委任することができます。 GRANT (ロール) ステートメントで WITH ADMIN OPTION 節を使用すれば、別のユーザーが以下を行えるようになります。

- ロールを他人に付与する。
- 他人のロールを取り消す。
- ロールに関するコメントを作成する。

WITH ADMIN OPTION 節を使って、以下を行うことはできません。

- ロールをドロップする。
- ロールの WITH ADMIN OPTION を許可 ID から取り消す。
- 他の誰かに WITH ADMIN OPTION を付与する (SECADM 権限の保有者でない場合)。

セキュリティー管理者がロールを作成した後、データベース管理者は GRANT ステートメントを使用して、権限および特権をそのロールに割り当てることができます。 SECADM 権限を例外として、データベース内で付与できるどの DB2 特権および権限でも、ロールに付与することができます。 SYSADM 権限などのインスタンスのレベルの権限をロールに割り当てることはできません。

セキュリティー管理者、またはあるロールに対しセキュリティー管理者から WITH ADMIN OPTION 付きでメンバーシップを付与されているすべてのユーザーは、GRANT (ロール) ステートメントを使用して、そのロールに対するメンバーシップを他のユーザー、グループ、PUBLIC、またはロールに付与することができます。ユーザーに対して直接、WITH ADMIN OPTION 付きであるロールのメンバーシップを付与することも、または PUBLIC、グループ、またはロールを通して間接的に付与することもできます。

ユーザーに割り当てられたすべてのロールは、そのユーザーがセッションを確立したときに有効になります。 DB2 データベース・システムでの許可の検査の際、ユーザーのロールに関連付けられたすべての特権と権限が考慮の対象になります。一部のデータベース・システムは、SET ROLE ステートメントを使用して特定のロールを活動化します。 DB2 データベース・システムは、SET ROLE ステートメントを使用する製品との互換性への配慮から、SET ROLE をサポートしています。 DB2 データベース・システムでは、セッション・ユーザーがロールのメンバーかどうかが SET ROLE ステートメントによって検査されて、メンバーでない場合は、エラーが戻されます。

あるロールでユーザーのメンバーシップを取り消す場合は、セキュリティー管理者、またはそのロールに対し WITH ADMIN OPTION 特権を保有するユーザーが、REVOKE (ロール) ステートメントを使用してこれを行います。

例

ロールは一連の特権を持っており、そのロールに対するメンバーシップを付与されたユーザーは、そのような特権を継承します。特権をこのように継承することによって、あるユーザーの特権を別のユーザーに再度割り当てるときに、特権を個別に管理する手間を省くことができます。ロールの使用により、必要な操作は、あるユーザーのロールに対するメンバーシップを取り消して、そのロールに対するメンバーシップを他のユーザーに付与するということだけで済みます。

例えば、DEV 部署に配属されている社員 BOB および ALICE は、表 SERVER、CLIENT、および TOOLS に対する SELECT 特権を持っているとします。ある日、上司が彼らを QA という別の部署に異動することを決定したので、データベース管理者は、表 SERVER、CLIENT、および TOOLS での選択を行う彼らの特権を取り消す必要が生じました。その後、部署 DEV には TOM という新入社員が配属されたので、データベース管理者は、表 SERVER 、CLIENT、および TOOLS に対する SELECT 特権を TOM に付与しなければなりません。

ロールを使用する場合、次のようなステップを行います。

1. セキュリティー管理者は、ロール DEVELOPER を次のようにして作成します。

   ```
   CREATE ROLE DEVELOPER
   ```

2. データベース管理者 (DBADM 権限の保有者) は、次のようにして、表 SERVER、CLIENT、および TOOLS に対する SELECT をロール DEVELOPER に付与します。

   ```
   GRANT  SELECT ON TABLE SERVER TO ROLE DEVELOPER
   GRANT  SELECT ON TABLE CLIENT TO ROLE DEVELOPER
   GRANT  SELECT ON TABLE TOOLS TO ROLE DEVELOPER
   ```

3. セキュリティー管理者は、次のようにして、部署 DEV 内のユーザー BOB と ALICE にロール DEVELOPER を付与します。

   ```
   GRANT ROLE DEVELOPER TO USER BOB, USER ALICE
   ```

4. BOB および ALICE が部署 DEV 外に配置換えになった場合、セキュリティー管理者は次のようにして、ロール DEVELOPER をユーザー BOB と ALICE から取り消します。

   ```
   REVOKE ROLE DEVELOPER FROM USER BOB, USER ALICE
   ```

5. TOM が部署 DEV に配属された場合、セキュリティー管理者は次のようにして、ロール DEVELOPER をユーザー TOM に付与します。

   ```
   GRANT ROLE DEVELOPER TO USER TOM
   ```

ロールの階層

ある 1 つのロールが別のロール内のメンバーシップを付与されると、ロール階層が形成されます。

ある 1 つのロールに対して、他のロールが付与されると、前者のロールの中に後者のロールが格納されます。後者のロールは、前者のロールのすべての特権を継承します。例えば、ロール DOCTOR がロール SURGEON に対して付与された場合、SURGEON には DOCTOR が格納されていることになります。ロール SURGEON は、ロール DOCTOR のすべての特権を継承します。

ロール階層内で循環を行うことはできません。*循環* が起きるのは、ある 1 つのロールが別のロールに付与されてから、後者のロールが前者のロールに付与されるという、循環方式でロールが付与される場合です。例えば、ロール DOCTOR がロール SURGEON に対して付与された後、ロール SURGEON が元のロール DOCTOR に戻って付与されることになります。ロール階層内で循環を確立した場合、エラーが戻されます (SQLSTATE 428GF)。

ロール階層の作成例

以下の例は、病院内の医療レベルを表すロール階層を作成する方法を示しています。

DOCTOR、SPECIALIST、および SURGEON という各ロールについて考察します。ロール階層を作成するには、循環を作成しないで、ロールを別のロールに付与します。ロール DOCTOR がロール SPECIALIST に付与され、ロール SPECIALIST がロール SURGEON に付与されます。

ロール SURGEON をロール DOCTOR に付与すると、循環が作成されるので、許可されません。

セキュリティー管理者は、次のような SQL ステートメントを実行して、ロール階層を作成します。

```
CREATE ROLE DOCTOR
CREATE ROLE SPECIALIST
CREATE ROLE SURGEON

GRANT ROLE DOCTOR TO ROLE SPECIALIST

GRANT ROLE SPECIALIST TO ROLE SURGEON
```

ロールからの特権の取り消しの効果

特権を取り消した場合、場合によっては、ビュー、パッケージ、またはトリガーなどの従属データベース・オブジェクトが無効になるかまたは作動不能になることがあります。

許可 ID から特定の特権を取り消す一方で、ロールを介するかまたは別の手段を講じて特権を維持した場合に、データベース・オブジェクトに何が起きるかを以下の例で示します。

ロールからの特権の取り消しの例

1. セキュリティー管理者は、次のように、ロール DEVELOPER を作成し、ユーザー BOB にこのロール内のメンバーシップを付与します。

   ```
   CREATE ROLE DEVELOPER
   GRANT ROLE DEVELOPER TO USER BOB
   ```

2. ユーザー ALICE が、次のように表 WORKITEM を作成します。

   ```
   CREATE TABLE WORKITEM (x int)
   ```

3. データベース管理者は、次のようにして、表 WORKITEM に対する SELECT および INSERT 特権を PUBLIC とロール DEVELOPER に付与します。

   ```
   GRANT SELECT ON TABLE ALICE.WORKITEM TO PUBLIC
   GRANT INSERT ON TABLE ALICE.WORKITEM TO PUBLIC
   GRANT SELECT ON TABLE ALICE.WORKITEM TO ROLE DEVELOPER
   GRANT INSERT ON TABLE ALICE.WORKITEM TO ROLE DEVELOPER
   ```

4. ユーザー BOB は、次のようにして、ビュー PROJECT を作成します。このビューは、表 WORKITEM と、表 WORKITEM に従属するパッケージ PKG1 を使用します。

   ```
   CREATE VIEW PROJECT AS SELECT * FROM ALICE.WORKITEM
   PREP emb001.sqc BINDFILE PACKAGE USING PKG1 VERSION 1
   ```

5. データベース管理者が、次のように、表 ALICE.WORKITEM に対する PUBLIC の SELECT 特権を取り消しても、ビュー BOB.PROJECT は操作可能のままで、パッケージ PKG1 も有効のままになります。なぜなら、ビューを定義した BOB はこれまでどおり、ロール DEVELOPER 内の自分のメンバーシップを通して必要な特権を保持しているからです。

   ```
   REVOKE SELECT ON TABLE ALICE.WORKITEM FROM PUBLIC
   ```

6. データベース管理者が、次のように、表 ALICE.WORKITEM に対するロール DEVELOPER の SELECT 特権を取り消した場合、ビュー BOB.PROJECT は作動不能になり、パッケージ PKG1 は無効になります。なぜなら、ビューとパッケージを定義した BOB は、他の手段を通して必要な特権を保有していないからです。

   ```
   REVOKE SELECT ON TABLE ALICE.WORKITEM FROM ROLE DEVELOPER
   ```

DBADM 権限の取り消しの例

以下の例では、ロール DEVELOPER は、DBADM 権限を保有していて、ユーザー BOB に付与されます。

1. セキュリティー管理者は、ロール DEVELOPER を次のようにして作成します。

   ```
   CREATE ROLE DEVELOPER
   ```

2. システム管理者は、次のように、DBADM 権限をロール DEVELOPER に付与します。

   ```
   GRANT DBADM ON DATABASE TO ROLE DEVELOPER
   ```

3. セキュリティー管理者は、次のようにして、ユーザー BOB にこのロール内のメンバーシップを付与します。

   ```
   GRANT ROLE DEVELOPER TO USER BOB
   ```

4. ユーザー ALICE が、次のように表 WORKITEM を作成します。

   ```
   CREATE TABLE WORKITEM (x int)
   ```

5. ユーザー BOB は、次のようにして、ビュー PROJECT を作成します。このビューは、表 WORKITEM、表 WORKITEM に従属するパッケージ PKG1、およびやはり表 WORKITEM に従属するトリガー TRG1 を使用します。

   ```
   CREATE VIEW PROJECT AS SELECT * FROM ALICE.WORKITEM
   PREP emb001.sqc BINDFILE PACKAGE USING PKG1 VERSION 1
   CREATE TRIGGER TRG1 AFTER DELETE ON ALICE.WORKITEM
                  FOR EACH STATEMENT MODE DB2SQL
          INSERT INTO ALICE.WORKITEM VALUES (1)
   ```

6. セキュリティー管理者は、次のように、ユーザー BOB のロール DEVELOPER を取り消します。

   ```
   REVOKE ROLE DEVELOPER FROM USER BOB
   ```

   ロール DEVELOPER を取り消すと、ユーザー BOB は DBADM 権限を喪失します。なぜなら、この権限の根拠であったロールが取り消されるからです。ビュー、パッケージ、およびトリガーは、次のような影響を受けます。

   - ビュー BOB.PROJECT はこれまでどおり有効です。
   - パッケージ PKG1 は無効になります。
   - トリガー BOB.TRG1 はこれまでどおり有効です。

ビュー BOB.PROJECT およびトリガー BOB.TRG1 は使用可能であるのに対して、パッケージ PKG1 は使用不可になります。 DBADM 権限が喪失しても、その DBADM 権限を保有していた許可 ID で作成されたビューおよびトリガー・オブジェクトは影響を受けません。

WITH ADMIN OPTION 節を使用したロール保守のデリゲート

セキュリティー管理者は、GRANT (ロール) SQL ステートメントで WITH ADMIN OPTION 節を使用すれば、ロール内のメンバーシップの管理を他の誰かに委任 (デリゲート) することができます。 WITH ADMIN OPTION 節を使えば、ロール内のメンバーシップを他のユーザーに付与する権限、ロールのメンバーのそのロール内のメンバーシップを取り消す権限、およびロールをドロップしないでそのロールに関するコメントを作成する権限を、別のユーザーに与えることができます。

WITH ADMIN OPTION 節は、ロールに対する WITH ADMIN OPTION を別のユーザーに付与する権限をどのユーザーにも与えません。また、ロールに対して別の許可 ID が持つ WITH ADMIN OPTION を取り消す権限を与えることもありません。

WITH ADMIN OPTION 節の使用を例示する例

1. セキュリティー管理者はロール DEVELOPER を作成し、次のように WITH ADMIN OPTION 節を使用して、この新しいロールをユーザー BOB に付与します。

   ```
   CREATE ROLE DEVELOPER
   GRANT ROLE DEVELOPER TO USER BOB WITH ADMIN OPTION
   ```

2. ユーザー BOB は、例えば他のユーザー ALICE を対象として、次のようにしてこのロール内のメンバーシップの付与および取り消しを行うことができます。

   ```
   GRANT ROLE DEVELOPER TO USER ALICE
   REVOKE ROLE DEVELOPER FROM USER ALICE
   ```

3. ユーザー BOB は、別のユーザーを対象としてロールのドロップや、WITH ADMIN OPTION の付与を行うことはできません (この 2 つの操作を実行できるのはセキュリティー管理者のみです)。 BOB が以下のコマンドを発行すると、失敗します。

   ```
   DROP ROLE DEVELOPER  - FAILURE!
                        - セキュリティー管理者のみが、ロールをドロップする
                          ことができます。
   GRANT ROLE DEVELOPER TO USER ALICE WITH ADMIN OPTION - FAILURE!
                        - セキュリティー管理者のみが、WITH ADMIN OPTION を
                          付与することができます。
   ```

4. ユーザー BOB は、ロール DEVELOPER のユーザーのロール管理特権 (WITH ADMIN OPTION で与えられる) を取り消すことはできません。なぜなら、当人はセキュリティー管理者 (SECADM) 権限を持っていないからです。 BOB が以下のコマンドを発行すると、次のように失敗します。

   ```
   REVOKE ADMIN OPTION FOR ROLE DEVELOPER FROM USER SANJAY - FAILURE!
   ```

5. セキュリティー管理者は、ユーザー BOB のロール DEVELOPER (WITH ADMIN OPTION で与えられる) のロール管理特権を取り消すことはできますが、ユーザー BOB はこれまでどおりロール DEVELOPER を付与されたままになります。

   ```
   REVOKE ADMIN OPTION FOR ROLE DEVELOPER FROM USER BOB
   ```

ただし、セキュリティー管理者が、次のようにユーザー BOB のロール DEVELOPER を単純に取り消した場合、BOB は、ロール DEVELOPER のメンバーであることによって与えられたすべての特権と、WITH ADMIN OPTION 節を通して与えられたロールに対する権限を喪失します。

```
REVOKE ROLE DEVELOPER FROM USER BOB
```

グループに対するロールの比較

グループに付与された特権および権限は、ビュー、マテリアライズ照会表 (MQT)、SQL ルーチン、トリガー、および静的 SQL を格納したパッケージの作成時には検討の対象にはされません。この制約事項が適用されないようにするには、グループではなくロールを使用します。

ユーザーはロールを介して、DB2 データベース・システムで制御されるロールを通して取得した特権を使ってデータベース・オブジェクトを作成することができます。グループとユーザーは、例えば、オペレーティング・システムや LDAP サーバーなどによって、DB2 データベース・システムから外部的に制御されます。

グループの使用をロールに置き換える例

以下の例は、ロールを使用してグループを置き換える方法を示しています。

DEVELOPER_G、TESTER_G、および SALES_G という 3 つのグループがあると仮定します。以下の表に示されているとおり、ユーザー BOB、ALICE、および TOM は、これらのグループのメンバーであるとします。

表 7. グループおよびユーザーの例

グループ	このグループに所属するユーザー
DEVELOPER_G	BOB
TESTER_G	ALICE、TOM
SALES_G	ALICE、BOB

1. セキュリティー管理者は、グループの代わりに使用するロール DEVELOPER、TESTER、および SALES を作成します。

   ```
   CREATE ROLE DEVELOPER
   CREATE ROLE TESTER
   CREATE ROLE SALES
   ```

2. セキュリティー管理者は、次のように、これらのロール内のメンバーシップをユーザーに付与します (グループ内でユーザーのメンバーシップを設定するのは、システム管理者の責務です)。

   ```
   GRANT  ROLE DEVELOPER TO USER BOB
   GRANT  ROLE TESTER TO USER ALICE, USER TOM
   GRANT  ROLE SALES TO USER BOB, USER ALICE
   ```

3. データベース管理者は、次のように、グループが保有していたものに似た特権または権限をこのロールに付与することができます。

   ```
   GRANT <privilege> ON <object> TO ROLE DEVELOPER
   ```

   データベース管理者は次に、グループのこれらの特権を取り消すと同時に、グループをシステムからも除去するようシステム管理者に依頼することができます。

ロールを通して取得した特権を使用したトリガーの作成例

以下の例は、ユーザー BOB が、ロール DEVELOPER を通して必要な特権を保有しているときに、トリガー TRG1 を正常に作成できるという例を示しています。

1. まず、ユーザー ALICE が次のように表 WORKITEM を作成します。

   ```
   CREATE TABLE WORKITEM (x int)
   ```

2. 次に、ALICE の表を変更する特権が、データベース管理者によってロール DEVELOPER に付与されます。

   ```
   GRANT ALTER ON ALICE.WORKITEM TO ROLE DEVELOPER
   ```

3. ユーザー BOB は、ロール DEVELOPER のメンバーであるので、トリガー TRG1 を正常に作成できます。

   ```
   CREATE TRIGGER TRG1 AFTER DELETE ON ALICE.WORKITEM
             FOR EACH STATEMENT MODE DB2SQL INSERT INTO ALICE.WORKITEM VALUES (1)
   ```

IBM Informix Dynamic Server からのマイグレーション後のロールの使用

IBM Informix® Dynamic Server から DB2 データベース・システムにマイグレーションしてロールを使用する場合、気を付ける必要のある点がいくつかあります。

Informix Dynamic Server (IDS) の SQL ステートメント GRANT ROLE は、節 WITH GRANT OPTION を提供します。 DB2 データベース・システムの GRANT ROLE ステートメントは、同じ機能を備えた節 WITH ADMIN OPTION (これは SQL 標準に準拠します) を提供します。 IDS から DB2 データベース・システムへのマイグレーションでは、dbschema ツールが CREATE ROLE および GRANT ROLE ステートメントを生成した後、dbschema ツールが WITH GRANT OPTION のすべてのオカレンスを WITH ADMIN OPTION に置き換えます。

IDS データベース・システムでは、SET ROLE ステートメントは特定のロールを活動化します。 DB2 データベース・システムは、SET ROLE ステートメントをサポートしますが、その目的は、この SQL ステートメントを使用する他の製品との互換性を保つことに限られます。 SET ROLE ステートメントは、セッション・ユーザーがロールのメンバーかどうかを検査し、メンバーでなければ、エラーを戻します。

dbschema の出力例

ロール DEVELOPER、TESTER、および SALES が IDS データベースに格納されていると仮定します。ユーザー BOB、ALICE、および TOM は、それぞれ別々のロールを付与されています。つまり、ロール DEVELOPER は BOB に付与され、ロール TESTER は ALICE に付与され、ロール TESTER および SALES が TOM に付与されています。 DB2 データベース・システムにマイグレーションするには、dbschema ツールを使用して、次のようにこのデータベース用の CREATE ROLE および GRANT ROLE ステートメントを作成します。

```
CREATE ROLE DEVELOPER
CREATE ROLE TESTER
CREATE ROLE SALES

GRANT  DEVELOPER TO BOB
GRANT  TESTER TO ALICE, TOM
GRANT  SALES TO TOM
```

DB2 データベース・システム内にデータベースを作成する必要があります。その後、そのデータベース内で上記のステートメントを実行し、ロールとそのロールの割り当てを再作成することができます。

第 3 章 トラステッド・コンテキストおよびトラステッド接続の使用

DB2 への接続が確立されている場合、アプリケーション内で明示的要求を行うことにより、明示的トラステッド接続を確立できます。これに成功するためには、確立する接続の属性と一致する属性を指定した CREATE TRUSTED CONTEXT SQL ステートメントを使用して、セキュリティー管理者がトラステッド・コンテキストを定義しておくことが必要です (後述のステップ 1 を参照)。

接続を確立する時の明示的トラステッド接続を要求するために使用する API は、使用するアプリケーションのタイプによって異なります (ステップ 2 の表を参照)。明示的トラステッド接続を確立した後、アプリケーションで接続のユーザーを切り替えることができます。ただしこれは、トラステッド・コンテキスト定義によりそのような切り替えに適用される制約に従い、アプリケーションのタイプに合った適切な API を使用して行われます (ステップ 3 の表を参照)。

1. セキュリティー管理者は、CREATE TRUSTED CONTEXT ステートメントを使用してサーバーにトラステッド・コンテキストを定義します。 例:

```
CREATE TRUSTED CONTEXT MYTCX
       BASED UPON CONNECTION USING SYSTEM AUTHID NEWTON
       ATTRIBUTES (ADDRESS   '192.0.2.1')
       WITH USE FOR PUBLIC WITHOUT AUTHENTICATION
       ENABLE
```

2. トラステッド接続を確立するには、アプリケーションで以下のいずれかの API を使用します。

オプション	説明
アプリケーション	**API**
CLI/ODBC	SQLConnect、SQLSetConnectAttr
XA CLI/ODBC	Xa_open
JAVA	getDB2TrustedPooledConnection、getDB2TrustedXAConnection

3. 別のユーザーに切り替えるには、認証のあるなしに関係なく、アプリケーションで以下のいずれかの API を使用します。

オプション	説明
アプリケーション	**API**
CLI/ODBC	SQLSetConnectAttr
XA CLI/ODBC	SQLSetConnectAttr
JAVA	getDB2Connection、reuseDB2Connection

明示的トラステッド接続の確立とユーザーの切り替えの例

以下の例では、中間層サーバーは、エンド・ユーザーの代わりにいくつかのデータベース要求を発行する必要がありますが、そのエンド・ユーザーに代わってデータベース接続を確立するためのエンド・ユーザーの資格情報へのアクセス権限がありません。

中間層サーバーがデータベースへの明示的トラステッド接続を確立することを許可するトラステッド・コンテキスト・オブジェクトをデータベース・サーバー上に作成できます。明示的トラステッド接続の確立後、中間層サーバーは、その接続の現行ユーザー ID を、データベース・サーバーで新規ユーザー ID を認証する必要なく新規ユーザー ID に切り替えることができます。以下の CLI コード・スニペットは、前述のステップ 1 で定義されたトラステッド・コンテキスト MYTCX を使用してトラステッド接続を確立する方法と、認証なしでトラステッド接続のユーザーを切り替える方法を示しています。

```
int main(int argc, char *argv[])
{
 SQLHANDLE henv;         /* environment handle */
 SQLHANDLE hdbc1;       /* connection handle */
 char origUserid[10] = "newton";
  char password[10] = "test";
   char switchUserid[10] = "zurbie";
 char dbName[10] = "testdb";

// Allocate the handles
SQLAllocHandle( SQL_HANDLE_ENV, &henv );
SQLAllocHandle( SQL_HANDLE_DBC, &hdbc1 );

// Set the trusted connection attribute
SQLSetConnectAttr( hdbc1, SQL_ATTR_USE_TRUSTED_CONTEXT,
SQL_TRUE, SQL_IS_INTEGER );

// Establish a trusted connection
SQLConnect( hdbc1, dbName, SQL_NTS, origUserid, SQL_NTS,
password, SQL_NTS );

//Perform some work under user ID "newton"
. . . . . .

 // Commit the work
SQLEndTran(SQL_HANDLE_DBC, hdbc1, SQL_COMMIT);


// Switch the user ID on the trusted connection
SQLSetConnectAttr( hdbc1,
SQL_ATTR_TRUSTED_CONTEXT_USERID, switchUserid,
SQL_IS_POINTER
);

//Perform new work using user ID "zurbie"
. . . . . .

//Commit the work
SQLEndTranSQL_HANDLE_DBC, hdbc1, SQL_COMMIT);

// Disconnect from database
SQLDisconnect( hdbc1 );

 return 0;

}  /* end of main */
```

問題判別

明示的トラステッド接続は、トラステッド接続の明示的で固有の要求により正常に確立された接続です。明示的トラステッド接続を要求した時にトラステッド接続を使用する資格がない場合には、通常の接続が確立され、警告が出されます (+20360)。ユーザーがトラステッド接続を確立できなかった理由を判別するため、セキュリティー管理者は、システム・カタログにあるトラステッド・コンテキスト定義と、接続属性を調べる必要があります。特に、接続を確立した IP アドレス、データ・ストリームまたはネットワークの暗号化レベル、および接続を行うシステム許可 ID を調べます。 db2pd ユーティリティーの -application オプションはこの情報に加えて、以下の情報も戻します。

- 接続信頼タイプ: 接続がトラステッドであるかそうでないかを示します。接続がトラステッドである場合、これが明示的トラステッド接続であるか暗黙的トラステッド接続であるかも示します。
- トラステッド・コンテキスト名: トラステッド接続に関連付けられたトラステッド・コンテキストの名前。
- 継承したロール: トラステッド接続を介して継承したロール。

注:

- TCP/IP は、トラステッド接続を確立するために使用できる DB2 サーバーとクライアント・アプリケーションが通信するためにサポートされる唯一のプロトコルです。
- データベース・サーバー認証タイプが CLIENT に設定されている場合、明示的トラステッド接続は確立できません。
- トラステッド接続が一度確立されると、接続の継続期間においてそのトラステッド状態が保持されます。

トラステッド・コンテキストおよびトラステッド接続

トラステッド・コンテキストとは、データベースと外部エンティティー (アプリケーション・サーバーなど) の間の接続における信頼関係を定義するデータベース・オブジェクトのことをいいます。

信頼関係は、以下の属性のセットに基づいています。

- システム許可 ID: データベース接続を確立するユーザーを表します
- IP アドレス (またはドメイン・ネーム): データベース接続を確立するホストを表します
- データ・ストリーム暗号化: データベース・サーバーとデータベース・クライアントの間のデータ通信のための暗号化設定がある場合にはそれを表します

ユーザーがデータベース接続を確立するときに、DB2 データベース・システムは、接続がデータベース内のトラステッド・コンテキスト・オブジェクトの定義と一致するかどうかを検査します。一致していた場合、データベース接続は信頼できると見なされます。

トラステッド接続を使用すると、このトラステッド接続の起動側では、トラステッド接続の有効範囲外では使用できない追加機能を取得することができます。追加機能は、トラステッド接続が明示的であるか暗黙的であるかによって異なります。

明示的トラステッド接続の起動側には以下の機能があります。

- その接続の現行ユーザー ID を、認証のあるなしに関係なく別のユーザー ID に切り替える
- トラステッド・コンテキストのロール継承フィーチャーにより追加の特権を取得する

暗黙的トラステッド接続は、明示的に要求されていないトラステッド接続で、明示的トラステッド接続要求ではなく通常の接続要求により確立されます。暗黙接続を取得するためにアプリケーション・コードを変更する必要はありません。また、暗黙的トラステッド接続を取得するかどうかは、接続戻りコードには影響はありません (明示的トラステッド接続を要求する場合は、接続戻りコードは要求が成功したかどうかを示します)。暗黙的トラステッド接続の起動側は、トラステッド・コンテキストのロール継承フィーチャーにより追加の特権を取得できるだけで、ユーザー ID を切り替えることはできません。

トラステッド・コンテキストを使用するとどのようにセキュリティーが向上するか

3 層アプリケーション・モデルは、クライアント・アプリケーションとデータベース・サーバーの間に中間層を置くことにより、標準的な 2 層クライアントおよびサーバー・モデルを拡張します。このモデルは、Web ベースのテクノロジーや Java™ 2 Enterprise Edition (J2EE) プラットフォームの登場により、近年特に大きな人気を得ています。3 層アプリケーション・モデルをサポートするソフトウェア・プロダクトの一例として、IBM WebSphere Application Server (WAS) があります。

3 層アプリケーション・モデルでは、クライアント・アプリケーションを実行するユーザーの認証、およびデータベース・サーバーとの相互作用の管理は、中間層が処理します。従来の方法では、データベース・サーバーとのすべての相互作用は、データベース・サーバーに対して中間層を識別するユーザー ID と資格情報の組み合わせを使用して、その中間層により確立されたデータベース接続を介して行われます。言い換えると、データベース・サーバーは、中間層のユーザー ID に関連付けられたデータベース特権を使用して、すべてのデータベース・アクセスで行う必要がある許可検査および監査を行います。これには、ユーザーの代わりに中間層により実行されるアクセスも含まれます。

3 層アプリケーション・モデルには多くの利点がありますが、データベース・サーバーとのすべての相互作用 (例えば、ユーザー要求) を中間層の許可 ID で行うようにすると、いくつかのセキュリティー上の問題が生じます。これらを要約すると以下のようになります。

- ユーザー ID の消失

  企業によっては、アクセス制御の目的で、データベースにアクセスしている実際のユーザーの ID を知りたい場合があります。

- ユーザーの説明責任の減少

  監査による説明責任は、データベース・セキュリティーにおける基本原則です。ユーザーの ID が不明であると、中間層の固有の目的のために中間層により実行されるトランザクションと、ユーザーのために中間層により実行されるトランザクションを区別することが難しくなります。

- 中間層の許可 ID に特権を付与しすぎる

  中間層の許可 ID には、すべてのユーザーからのすべての要求を実行するために必要なすべての特権が含まれていなければなりません。これには、特定の情報にアクセスする必要がないユーザーがアクセス権限を取得できてしまうというセキュリティー問題があります。
- 弱いセキュリティー

  前述の特権の問題に加えて、現行方式では、接続するために中間層により使用される許可 ID には、ユーザー要求によりアクセスされる可能性があるすべてのリソースへの特権を付与する必要があります。中間層の許可 ID の暗号漏えいが発生すると、それらすべてのリソースは公開されてしまいます。
- 同じ接続を使用するユーザー間で相互に影響を及ぼす

  直前のユーザーによる変更が現行ユーザーに影響を与える場合があります。

明らかに、実際のユーザーの ID およびデータベース特権が、そのユーザーに代わって中間層により実行されるデータベース要求で使用されるようなメカニズムが必要です。この目標を達成するための最も簡単な方法は、中間層がユーザーの ID とパスワードを使用して新規接続を確立した後、ユーザーの要求をその接続を介して送信するというものです。この方法は単純ですが、以下に挙げるようないくつかの欠点があります。

- 特定の中間層では不適当。多くの中間層サーバーには、接続を確立するために必要なユーザー認証資格情報がありません。
- パフォーマンス上のオーバーヘッド。新しい物理接続を作成し、データベース・サーバーでユーザーを再認証することに関連した、パフォーマンス上の明らかなオーバーヘッドがあります。
- 保守上のオーバーヘッド。一元的なセキュリティー・セットアップ、またはシングル・サインオンを使用していない状況では、2 つのユーザー定義 (1 つは中間層上、もう 1 つはサーバー上) を持つことによる保守上のオーバーヘッドがあります。この状況では、異なる場所にあるパスワードを変更することが必要です。

トラステッド・コンテキスト機能は、この問題を解決します。セキュリティー管理者は、データベースと中間層の間の信頼関係を定義するトラステッド・コンテキスト・オブジェクトをデータベースに作成できます。その後、中間層ではデータベースへの明示的トラステッド接続を確立できますが、この接続では、接続の現行ユーザー ID を、認証のあるなしに関係なく別のユーザー ID に切り替える機能が中間層に付与されます。トラステッド・コンテキストは、エンド・ユーザーの ID アサーション問題を解決するだけでなく、別の利点もあります。それは、データベース・ユーザーが特権を使用できるようになる時期を制御する機能です。ユーザーが特権を使用できる時期を制御できないと、全体的なセキュリティーの低下につながります。例えば、特権が、最初に意図した目的以外で使用される場合があります。セキュリティー管理者は、1 つ以上の特権を 1 つのロールに割り当て、そのロールをトラステッド・コンテキスト・オブジェクトに割り当てることができます。そのトラステッド・コンテキストの定義と一致するトラステッド・データベース接続 (明示的または暗黙的) のみがそのロールに関連付けられた特権を利用できます。

パフォーマンスの向上

トラステッド接続を使用すると以下の利点があるため、パフォーマンスを最大限に発揮します。

- 接続の現行ユーザー ID が切り替わる時に新規接続は確立されません。
- トラステッド・コンテキスト定義が切り替え先のユーザー ID の認証を必要としない場合には、データベース・サーバーで新規ユーザーを認証することに関連したオーバーヘッドは発生しません。

トラステッド・コンテキストの作成例

セキュリティー管理者が以下のトラステッド・コンテキスト・オブジェクトを作成すると想定します。

```
CREATE TRUSTED CONTEXT CTX1
  BASED UPON CONNECTION USING SYSTEM AUTHID USER2
  ATTRIBUTES (ADDRESS   '192.0.2.1')
  DEFAULT ROLE managerRole
   ENABLE
```

ユーザー *user1* が IP アドレス 192.0.2.1 からトラステッド接続を要求した場合、DB2 データベース・システムは、トラステッド接続を確立できなかったためユーザー *user1* が非トラステッド接続を取得したことを示す警告 (SQLSTATE 01679、SQLCODE +20360) を戻します。しかし、ユーザー *user2* が IP アドレス 192.0.2.1 からトラステッド接続を要求した場合には、接続属性はトラステッド・コンテキスト CTX1 により条件が満たされるため、要求は受け入れられます。ユーザー *user2* はトラステッド接続を確立したため、そのユーザーはトラステッド・コンテキストのロール managerRole に関連付けられたすべての特権および権限を取得できます。このトラステッド接続の有効範囲外では、ユーザー *user2* はこれらの特権および権限を使用できません。

トラステッド・コンテキストを使用したロール・メンバーシップの継承

トラステッド接続の現行ユーザーは、トラステッド・コンテキストを使用したロールの自動継承により追加の特権を取得できます。ただしこれは、関連したトラステッド・コンテキスト定義の一部としてセキュリティー管理者により指定されている場合のみ該当します。

デフォルトでは、ロールはトラステッド接続のすべてのユーザーが継承できます。セキュリティー管理者は、トラステッド・コンテキスト定義を使用して特定のユーザーが継承するロールを指定することもできます。

トラステッド接続を使用している時にセッション許可 ID が保持できるアクティブなロールには以下のものがあります。

- セッション許可 ID が通常はメンバーであると見なされるロール。加えて
- トラステッド・コンテキストのデフォルトのロールまたはトラステッド・コンテキストのユーザー固有のロール (定義されている場合)

注:

- 接続に成功した時にセキュリティー・プラグインにより作成されるシステム許可 ID とセッション許可 ID が互いに異なるように作成されたカスタム・セキュリテ

ィー・プラグインを使用して、ユーザー認証を構成する場合には、トラステッド・コンテキストのロールは、その接続がトラステッド接続である場合であってもその接続を介して継承することはできません。

- ロールを使用して取得したトラステッド・コンテキスト特権は、動的 DML 操作においてのみ有効です。これらは、以下においては無効です。
  - DDL 操作
  - 非動的 SQL (BIND、REBIND、暗黙的な再バインド、追加バインドなどの静的 SQL ステートメントに関係した操作)

トラステッド・コンテキストのユーザー固有の特権の取得

セキュリティー管理者は、以下のようにするため、トラステッド・コンテキスト定義を使用してロールをトラステッド・コンテキストに関連付けることができます。

- デフォルトで、トラステッド接続のすべてのユーザーが、指定されたロールを継承できる
- トラステッド接続の特定のユーザーが、指定されたロールを継承できる

トラステッド接続のユーザーが新しい許可 ID に切り替わり、この新しい許可 ID にトラステッド・コンテキストのユーザー固有のロールが存在する場合、例で示されているように、ユーザー固有のロールがトラステッド・コンテキストのデフォルトのロールをオーバーライドします (トラステッド・コンテキストが存在する場合)。

デフォルトのロールおよびユーザー固有のロールを割り当てるトラステッド・コンテキストの作成例

セキュリティー管理者が以下のトラステッド・コンテキスト・オブジェクトを作成すると想定します。

```
CREATE TRUSTED CONTEXT CTX1
  BASED UPON CONNECTION USING SYSTEM AUTHID USER1
 ATTRIBUTES (ADDRESS   '192.0.2.1')
  WITH USE FOR USER2 WITH AUTHENTICATION,
               USER3 WITHOUT AUTHENTICATION
  DEFAULT ROLE AUDITOR
  ENABLE
```

USER1 がトラステッド接続を確立すると、ロール AUDITOR に付与された特権はこの許可 ID に継承されます。同様に、これらの同じ特権は、トラステッド接続の現行許可 ID が USER3 のユーザー ID に切り替わった時にこのユーザーにも継承されます。 (接続のユーザー ID がある時点で USER2 に切り替わった場合には、USER2 もトラステッド・コンテキストのデフォルトのロールである AUDITOR を継承します。) セキュリティー管理者は、トラステッド・コンテキストのデフォルトのロールとは異なるロールを USER3 に継承させるように選択することができます。これは、以下のように、このユーザーに固有のロールを割り当てることにより行えます。

```
CREATE TRUSTED CONTEXT CTX1
 BASED UPON CONNECTION USING SYSTEM AUTHID USER1
 ATTRIBUTES (ADDRESS   '192.0.2.1')
  WITH USE FOR USER2 WITH AUTHENTICATION,
               USER3 WITHOUT AUTHENTICATION ROLE OTHER_ROLE
  DEFAULT ROLE AUDITOR
  ENABLE
```

トラステッド接続の現行ユーザー ID が USER3 に切り替わっても、このユーザーはトラステッド・コンテキストのデフォルトのロールを継承しなくなります。代わりに、セキュリティー管理者によってそのユーザーに割り当てられた固有のロールである OTHER_ROLE を継承します。

明示的トラステッド接続でのユーザー ID の切り替え

明示的トラステッド接続では、接続のユーザー ID を別のユーザー ID に切り替えることができます。この際、認証があるかないかは、この明示的トラステッド接続に関連付けられたトラステッド・コンテキスト・オブジェクトの定義によって異なります。

トラステッド接続のユーザーを切り替える場合、アプリケーション内で以下のいずれかの API を使用して、要求を明示的に行うことが必要です。

表 8. トラステッド接続でのユーザー切り替え要求のためのアプリケーション API

アプリケーション	API
CLI/ODBC	SQLSetConnectAttr
JAVA	getDB2Connection、reuseDB2Connection
XA	SQLSetConnectAttr

ユーザー ID を切り替えるための規則

1. 切り替え要求が明示的トラステッド接続から出されず、切り替え要求が処理のためにサーバーに送信された場合、その接続はシャットダウンされ、エラー・メッセージが戻されます (SQLSTATE 08001、SQLCODE-30082、理由コード 41)。
2. 切り替え要求がトランザクション境界で出されず、トランザクションがロールバックされ、さらに切り替え要求が処理のためにサーバーに送信された場合、その接続は非接続状態になり、エラー・メッセージが戻されます (SQLSTATE 58009、SQLCODE -30020)。
3. 切り替え要求がストアード・プロシージャー内から出された場合、エラー・メッセージが戻され (SQLCODE -30090、理由コード 29)、これはこの環境では正しくない操作であることを示します。接続状態は維持され、接続は非接続状態にはなりません。以降の要求は処理できます。
4. 切り替え要求が (データベース接続ではなく) インスタンス接続のサーバーに送信された場合、その接続はシャットダウンされ、エラー・メッセージが戻されます (SQLCODE -30005)。
5. 切り替え要求がトラステッド接続で許可されない許可 ID で出された場合、エラー (SQLSTATE 42517、SQLCODE -20361) が戻され、接続は非接続状態になります。
6. トラステッド接続での切り替え要求が認証付きで許可される (WITH AUTHENTICATION が指定されている) 許可 ID で出されているのに適切な認証トークンが提供されていない場合、エラー (SQLSTATE 42517、SQLCODE -20361) が戻され、接続は非接続状態になります。

7. トラステッド接続に関連付けられたトラステッド・コンテキスト・オブジェクトが使用不可になり、そのトラステッド接続への切り替え要求が出された場合、エラー (SQLSTATE 42517、SQLCODE -20361) が戻され、接続は非接続状態になります。

   この場合、受け入れられる唯一のユーザー切り替え要求は、トラステッド接続を確立するユーザー ID または NULL ユーザー ID を指定する要求です。トラステッド接続を確立したユーザー ID への切り替えが行われた場合、このユーザー ID は、トラステッド・コンテキストのロール (トラステッド・コンテキストのデフォルトのロールもトラステッド・コンテキストのユーザー固有のロールも) を継承しません。

8. トラステッド接続に関連付けられたトラステッド・コンテキスト・オブジェクトのシステム許可 ID 属性が変更され、そのトラステッド接続への切り替え要求が出された場合、エラー (SQLSTATE 42517、SQLCODE -20361) が戻され、接続は非接続状態になります。

   この場合、受け入れられる唯一のユーザー切り替え要求は、トラステッド接続を確立するユーザー ID または NULL ユーザー ID を指定する要求です。トラステッド接続を確立したユーザー ID への切り替えが行われた場合、このユーザー ID は、トラステッド・コンテキストのロール (トラステッド・コンテキストのデフォルトのロールもトラステッド・コンテキストのユーザー固有のロールも) を継承しません。

9. トラステッド接続に関連付けられたトラステッド・コンテキスト・オブジェクトがドロップされ、そのトラステッド接続への切り替え要求が出された場合、エラー (SQLSTATE 42517、SQLCODE -20361) が戻され、接続は非接続状態になります。

   この場合、受け入れられる唯一のユーザー切り替え要求は、トラステッド接続を確立するユーザー ID または NULL ユーザー ID を指定する要求です。トラステッド接続を確立したユーザー ID への切り替えが行われた場合、このユーザー ID は、トラステッド・コンテキストのロール (トラステッド・コンテキストのデフォルトのロールもトラステッド・コンテキストのユーザー固有のロールも) を継承しません。

10. トラステッド接続での切り替え要求が許可されたユーザー ID で出されたにもかかわらず、そのユーザー ID がデータベースに対する CONNECT 特権を保持していない場合、その接続は非接続状態になり、エラー・メッセージが戻されます (SQLSTATE 08004、SQLCODE -1060)。
11. トラステッド・コンテキストのシステム許可 ID が WITH USE FOR 節に出現する場合、DB2 データベース・システムは、そのシステム許可 ID に戻すためのユーザー切り替え要求のシステム許可 ID の認証設定を受け入れます。トラステッド・コンテキストのシステム許可 ID が WITH USE FOR 節に出現しない場合には、そのシステム許可 ID に戻すためのユーザー切り替え要求は認証なしの場合でも常に許可されます。

注: 接続が非接続状態になったときに、以下の要求だけは受け入れられ、「アプリケーションの状態にエラーがあります。データベース接続が存在しません。」(SQLCODE -900) エラーが戻されません。

- ユーザー切り替え要求

- COMMIT または ROLLBACK ステートメント
- DISCONNECT、CONNECT RESET、または CONNECT 要求

注: トラステッド接続のユーザー ID が新しいユーザー ID に切り替わると、古いユーザーの接続環境の痕跡はすべてなくなります。言い換えると、ユーザー ID の切り替えにより、環境は新しい接続環境と全く同一になります。例えば、接続における古いユーザー ID で TEMPORARY 表または WITH HOLD カーソルをオープンしている場合、その接続のユーザー ID が新しいユーザー ID に切り替わると、これらのオブジェクトは完全に失われます。

特定のユーザーに認証を指定するトラステッド接続の作成例

セキュリティー管理者が以下のトラステッド・コンテキスト・オブジェクトを作成すると想定します。

```
CREATE TRUSTED CONTEXT CTX1
 BASED UPON CONNECTION USING SYSTEM AUTHID USER1
 ATTRIBUTES (ADDRESS   '192.0.2.1')
  WITH USE FOR USER2 WITH AUTHENTICATION,
               USER3 WITHOUT AUTHENTICATION
  ENABLE
```

さらに、明示的トラステッド接続が確立されたと仮定します。 USER3 は、認証を必要としないトラステッド・コンテキスト CTX1 のユーザーとして定義されているので、トラステッド接続のユーザー ID を認証情報を提供せずに USER3 に切り替える要求は許可されます。ただし、USER2 は、認証情報を提供する必要があるトラステッド・コンテキスト CTX1 のユーザーとして定義されているので、トラステッド接続のユーザー ID を認証情報を提供せずに USER2 に切り替える要求は失敗します。

第 4 章 ラベル・ベースのアクセス制御 (LBAC)

ラベル・ベースのアクセス制御 (LBAC) は、データにどのユーザーがアクセスできるかに対する制御を大きく向上させます。 LBAC を使用すると、個々の行および個々の列に対して、どのユーザーに書き込みアクセスがあり、どのユーザーに読み取りアクセスがあるのかを厳密に決定することができます。

LBAC の動作

LBAC 機能は非常に構成しやすく、特定の安全保護環境と一致するように調整することができます。すべての LBAC 構成はセキュリティー管理者 により実行されます。セキュリティー管理者は、システム管理者により SECADM 権限が付与されているユーザーです。

セキュリティー管理者は、セキュリティー・ポリシーを作成して LBAC システムを構成します。セキュリティー・ポリシー では、どのデータに誰がアクセスできるかを判断するために使用される基準を記述します。任意の 1 つの表を保護するためには、1 つのセキュリティー・ポリシーしか使用できませんが、異なる表には、異なるセキュリティー・ポリシーを用いて保護することができます。

セキュリティー・ポリシーを作成した後、セキュリティー管理者は、そのポリシーの一部であるセキュリティー・ラベル というオブジェクトを作成します。セキュリティー・ラベルに厳密に何が含まれるかはセキュリティー・ポリシーにより決定され、特定のデータ項目にアクセスできるユーザーを決定するために組織が使用する基準を示すように構成することができます。たとえば、ある人の会社内での立場とその人がどのプロジェクトに参加しているかを参照して、その人が表示することができるデータを判断する場合には、各ラベルにその情報が含まれるようにセキュリティー・ラベルを構成することができます。 LBAC は柔軟であるため、非常に複雑な基準だけでなく、各ラベルが ″high″ または ″low″ のいずれかの信頼レベルを示すだけであるような非常に単純なシステムに至るまで、自由にセットアップできます。

作成が完了すると、セキュリティー・ラベルを表の個々の列と行に関連付けてそこに保持されているデータを保護することができます。セキュリティー・ラベルにより保護されるデータは、保護データ と呼ばれます。セキュリティー管理者は、ユーザーにセキュリティー・ラベルを付与することにより、保護データへのアクセスを許可します。ユーザーが保護データへのアクセスを試行すると、そのユーザーのセキュリティー・ラベルが、データを保護しているセキュリティー・ラベルと比較されます。セキュリティー・ラベルには、保護ラベルによってブロックされるものと、そうでないものがあります。

ユーザー、ロール、またはグループは、複数のセキュリティー・ポリシーに対する (複数の) セキュリティー・ラベルを同時に保持することが許可されています。ただし、どのセキュリティー・ポリシーに対しても、ユーザー、ロール、またはグループは読み取りアクセス用に最大 1 つのラベル、書き込みアクセス用に最大 1 つのラベルしか保持することができません。

セキュリティー管理者はユーザーに免除を付与することもできます。*免除* があれば、本来はセキュリティー・ラベルによってアクセスできない保護データにアクセスすることができます。セキュリティー・ラベルと免除をまとめて、*LBAC 信用証明情報* といいます。

LBAC 信用証明情報がアクセスを許可しない保護列にアクセスしようとすると、アクセスは失敗し、エラー・メッセージを受け取ります。

LBAC 信用証明情報が読み取りを許可しない保護行の読み取りを試行すると、DB2 はそれらの行が存在しないかのように動作します。それらの行は、実行するすべての SQL ステートメント (SELECT、UPDATE、DELETE を含む) において、その一部として選択することはできません。集約関数であっても、LBAC 信用証明情報が読み取りを許可しない行は無視します。たとえば、COUNT(*) 関数は、読み取りアクセスを持つ行のみのカウントを戻します。

ビューと LBAC

ビューを、無保護の表にビューを定義する際と同様に、保護された表に定義することができます。そのようなビューにアクセスする際には、基礎表に対する LBAC 保護が施行されます。使用される LBAC 信用証明情報は、セッション許可 ID の LBAC 信用証明情報となります。同じビューに 2 人のユーザーがアクセスすると、それぞれの LBAC 信用証明情報により異なる行が表示される可能性があります。

参照保全制約と LBAC

以下の規則は、参照保全制約がある場合に LBAC 規則が施行される方法を説明しています。

- **規則 1**: LBAC 読み取りアクセス規則は、子表の内部で生成されたスキャンには適用されません。これは、孤立した子ができないようにするためです。
- **規則 2**: LBAC 読み取りアクセス規則は、親表の内部で生成されたスキャンには適用されません。
- **規則 3**: 子表に対して CASCADE 操作が実行される際に LBAC 書き込み規則が適用されます。たとえば、ユーザーが親を削除したものの、LBAC 書き込み規則違反となるためにどの子も削除できない場合には、削除をロールバックする必要があり、エラーが出されます。

LBAC を使用したストレージ・オーバーヘッド

LBAC を使用して表を行レベルで保護する場合、追加のストレージ・コストは行セキュリティー・ラベル列のコストです。このコストは、選択したセキュリティー・ラベルのタイプによって異なります。例えば、表を保護するために 2 つのコンポーネントを持つセキュリティー・ポリシーを作成する場合、そのセキュリティー・ポリシーからのセキュリティー・ラベルは 16 バイト (コンポーネントごとに 8 バイト) になります。行セキュリティー・ラベル列は NULL 不可 VARCHAR 列として扱われるため、この場合の合計コストは行ごとに 20 バイトになります。通常、行ごとの合計コストは (*N**8 + 4) バイトです。ここで、*N* はセキュリティー・ポリシーの表を保護するコンポーネントの数です。

LBAC を使用して表を列レベルで保護する場合、列セキュリティー・ラベルはメタデータです (つまり、列のメタデータとともに SYSCOLUMNS カタログ表に格納さ

れます)。このメタデータは、列を保護するセキュリティー・ラベルの ID に過ぎません。この場合、ユーザー表はストレージ・オーバーヘッドの影響を受けません。

LBAC が行わない動作

- LBAC は、任意アクセス制御により禁止されているデータへのアクセスは、決して許可しません。

  例: 表からの読み取りの許可がない場合には、その表からのデータの読み取りは許可されません。普通なら LBAC によってアクセスが許可されるはずの行および列に関しても同様です。
- LBAC 信用証明情報は、保護データへのアクセスのみを制限します。無保護のデータへのアクセスには影響がありません。
- 表またはデータベースをドロップする場合、その表またはデータベースに保護データが含まれている場合であっても、LBAC 信用証明情報はチェックされません。
- データをバックアップする際には LBAC 信用証明情報はチェックされません。表のバックアップを実行できる場合、どの行がバックアップされるかについて、データの LBAC 保護により制限されることはまったくありません。また、バックアップ・メディア上のデータは LBAC により保護されません。データベース上のデータのみが保護されます。
- LBAC は、次のタイプの表を保護するために使用することはできません。
  - マテリアライズ照会表 (MQT)
  - マテリアライズ照会表 (MQT) が依存する表
  - ステージング表
  - ステージング表が依存する表
  - 型付き表
- LBAC 保護はニックネームには適用できません。

LBAC チュートリアル

LBAC の使用の基本を手引きするチュートリアルを、オンラインで利用することができます。このチュートリアルは、IBM developerWorks® Web サイト (http://www.ibm.com/developerworks/db2) にあり、『DB2 Label-Based Access Control, a practical guide』というタイトルです。

LBAC セキュリティー・ポリシー

セキュリティー管理者は、セキュリティー・ポリシーを使用して、表の個々の行および個々の列ごとに、誰に書き込みアクセスがあり、誰に読み取りアクセスがあるかを規定する基準を定義します。

セキュリティー・ポリシーには、次の情報が含まれます。

- ポリシーの一部であるセキュリティー・ラベルにおいて、どのセキュリティー・ラベル・コンポーネントが使用されるか
- それらのセキュリティー・ラベル・コンポーネントを比較する際に、どの規則が使用されるか

- ポリシーにより保護されるデータにアクセスする際に、どのオプションの動作が使用されるか
- セキュリティー・ポリシーで保護されているデータへのアクセス権の行使時に、どのような追加のセキュリティー・ラベルおよび免除を考慮の対象とするか。例えば、ロールおよびグループに付与されたセキュリティー・ラベルを考慮の対象とするかどうかのオプションは、セキュリティー・ポリシーを通して制御されます。

すべての保護されている表は、関連付けられたセキュリティー・ポリシーを 1 つだけ持たなければなりません。その表の行および列は、そのセキュリティー・ポリシーの一部であるセキュリティー・ラベルでのみ保護することができ、保護データの全アクセスは、そのポリシーの規則に従います。単一のデータベースで複数のセキュリティー・ポリシーを持つことができますが、特定の表を保護するセキュリティー・ポリシーを複数持つことはできません。

セキュリティー・ポリシーの作成

セキュリティー・ポリシーを作成する人は、セキュリティー管理者でなければなりません。セキュリティー・ポリシーは、SQL ステートメントの CREATE SECURITY POLICY で作成します。セキュリティー・ポリシーでリストされるセキュリティー・ラベル・コンポーネントは、CREATE SECURITY POLICY ステートメントを実行する前に作成する必要があります。セキュリティー・ポリシーが作成される際にコンポーネントがリストされる順序は、コンポーネント間の何らかの優先順位またはその他の関係を示すものではありませんが、SECLABEL のような組み込み関数でセキュリティー・ラベルを作成する際にその順序を知っておくことは重要です。

セキュリティー・ポリシーの変更

セキュリティー管理者は、ALTER SECURITY POLICY ステートメントを使用して、セキュリティー・ポリシーを変更することができます。

セキュリティー・ポリシーのドロップ

セキュリティー・ポリシーをドロップする人は、セキュリティー管理者でなければなりません。セキュリティー・ポリシーは、SQL ステートメントの DROP を使用してドロップします。

セキュリティー・ポリシーは、表に関連付けられている (追加されている) 場合は、ドロップできません。

LBAC セキュリティー・ラベル・コンポーネントの概要

セキュリティー・ラベル・コンポーネント は、ラベル・ベースのアクセス制御 (LBAC) の一部であるデータベース・オブジェクトです。セキュリティー・ラベル・コンポーネントは、組織のセキュリティー構造をモデル化するために使用します。

セキュリティー・ラベル・コンポーネントは、ユーザーがデータの特定の部分にアクセスする必要があるかどうかを判別するために使用できる任意の基準を表すことができます。そのような基準の代表的な例として、以下のものがあります。

- ユーザーに対する信頼の程度
- ユーザーの所属部門
- ユーザーが特定のプロジェクトに参加しているかどうか

例: あるユーザーが所属する部門によって、どのデータにアクセスできるかに影響を与えるようにする場合、dept という名前のコンポーネントを作成し、そのコンポーネントのエレメントで会社のさまざまな部門の名前を定義することができます。その後、コンポーネント dept をセキュリティー・ポリシーに組み込みます。

セキュリティー・ラベル・コンポーネントのエレメント は、そのコンポーネントに対して許可される 1 つの特定の「設定値」です。

例: 信頼のレベルを表すセキュリティー・ラベル・コンポーネントとして、Top Secret、Secret、Classified、および Unclassified の 4 つのエレメントを設けることができます。

セキュリティー・ラベル・コンポーネントの作成

セキュリティー・ラベル・コンポーネントを作成する人は、セキュリティー管理者でなければなりません。セキュリティー・ラベル・コンポーネントは、SQL ステートメントの CREATE SECURITY LABEL COMPONENT で作成します。

セキュリティー・ラベル・コンポーネントを作成する際には、以下を指定する必要があります。

- コンポーネントの名前
- そのコンポーネントのタイプ (ARRAY、TREE、または SET)
- 許可されるエレメントの完全なリスト
- タイプ ARRAY および TREE では、各エレメントがコンポーネントの構造に収まる方法を記述する必要があります。

コンポーネントのタイプ

次の 3 つのタイプのセキュリティー・ラベル・コンポーネントがあります。

- TREE: 各エレメントはツリー構造内のノードを表します
- ARRAY: 各エレメントは線形スケール上の点を表します
- SET: 各エレメントはある集合の 1 人のメンバーを表します

エレメントが互いに関連し合うことができるさまざまな方法をモデル化するためにタイプを使用することができます。たとえば、会社内の 1 つ以上の部門を記述するコンポーネントを作成する場合、おそらく TREE のコンポーネント・タイプを使用するはずです。なぜなら、ほとんどのビジネス構造はツリーの形式になっているからです。ユーザーが持つ信頼のレベルを表すコンポーネントを作成する場合、おそらくタイプ ARRAY のコンポーネントを使用するはずです。なぜなら、信頼の 2 つのレベルがある場合、一方は常に他方より高いからです。

エレメントが互いに持つことができる関係の詳細記述などの、各タイプの詳細は、タイプごとのセクションで説明されています。

セキュリティー・ラベル・コンポーネントの変更

セキュリティー管理者は、ALTER SECURITY LABEL COMPONENT ステートメントを使用して、セキュリティー・ラベル・コンポーネントを変更することができます。

セキュリティー・ラベル・コンポーネントのドロップ

セキュリティー・ラベル・コンポーネントをドロップする人は、セキュリティー管理者でなければなりません。セキュリティー・ラベル・コンポーネントは、SQL ステートメントの DROP でドロップします。

LBAC セキュリティー・ラベル・コンポーネント・タイプ: SET

SET は、ラベル・ベースのアクセス制御 (LBAC) セキュリティー・ポリシーで使用できるセキュリティー・ラベル・コンポーネントのタイプの 1 つです。

タイプ SET のコンポーネントは、エレメントの順不同のリストです。このタイプのコンポーネントのエレメントに対して行うことができる比較は、特定のエレメントがリストにあるかどうかだけです。

LBAC セキュリティー・ラベル・コンポーネント・タイプ: ARRAY

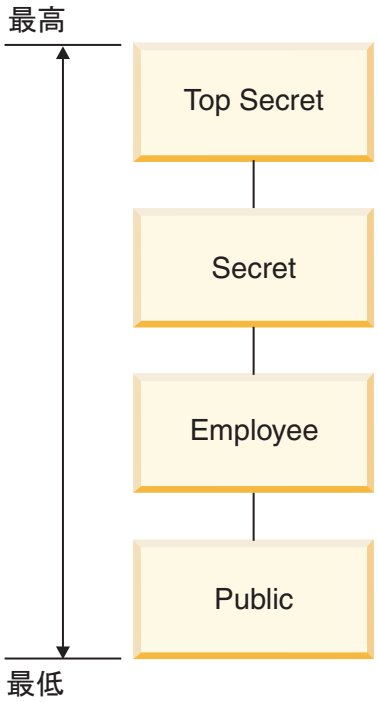
ARRAY は、セキュリティー・ラベル・コンポーネントのタイプの 1 つです。

ARRAY タイプのコンポーネントでは、コンポーネントが作成されるときにエレメントがリストされている順序によりスケールが定義されます。そこでは、リストされている最初のエレメントが最高の値となり、最後のエレメントが最低になります。

例: コンポーネント mycomp が次のように定義されているとします。

```
CREATE SECURITY LABEL COMPONENT mycomp
  ARRAY [ 'Top Secret', 'Secret', 'Employee', 'Public' ]
```

この場合、エレメントは、以下のような構造に編成されているかのようにして処理されます。



タイプ ARRAY のコンポーネントでは、エレメントは互いに以下の種類の関係を持つことができます。

より高い

ARRAY 節の中でエレメント A がエレメント B より前にリストされている場合には、エレメント A はエレメント B より高くなります。

より低い

ARRAY 節の中でエレメント A がエレメント B より後にリストされている場合には、エレメント A はエレメント B より低くなります。

LBAC セキュリティー・ラベル・コンポーネント・タイプ: TREE

TREE は、ラベル・ベースのアクセス制御 (LBAC) セキュリティー・ポリシーで使用できるセキュリティー・ラベル・コンポーネントのタイプの 1 つです。

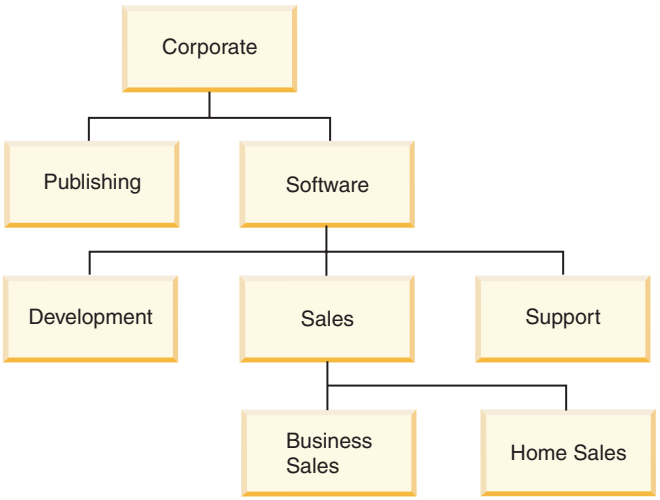
TREE タイプのコンポーネントでは、エレメントは、ツリー構造に配置されているかのようにして処理されます。タイプ TREE のコンポーネントの一部であるエレメントを指定する際、それが、他のどのエレメントの下位に置かれるかも指定する必要があります。唯一の例外は、ツリーの ROOT であると指定する必要のある最初のエレメントです。これにより、エレメントをツリー構造に編成することができます。

例: コンポーネント mycomp が次のように定義されているとします。

```
CREATE SECURITY LABEL COMPONENT mycomp
TREE (
    'Corporate'      ROOT,
    'Publishing'     UNDER 'Corporate',
    'Software'       UNDER 'Corporate',
    'Development'    UNDER 'Software',
    'Sales'          UNDER 'Software',
```

```
    'Support'        UNDER 'Software'
    'Business Sales' UNDER 'Sales'
    'Home Sales'     UNDER 'Sales'
)
```

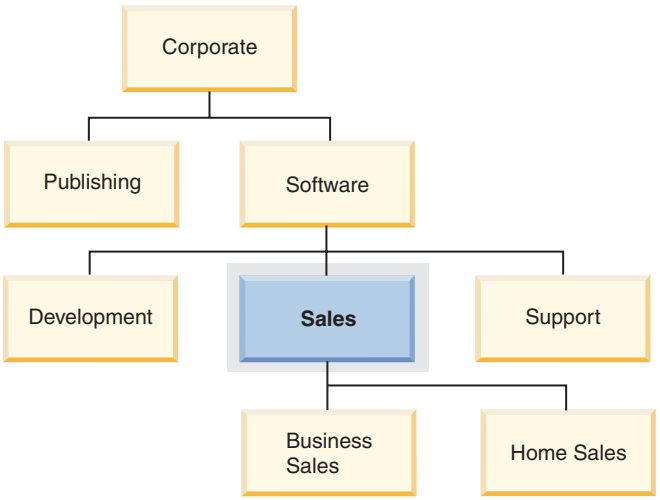
この場合、エレメントは、以下のようなツリー構造に編成されているかのようにして処理されます。



タイプ TREE のコンポーネントでは、エレメントは互いに以下のタイプの関係を持つことができます。

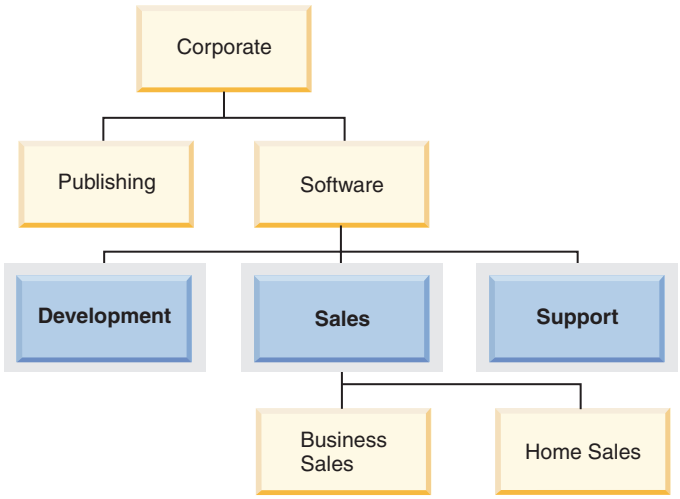
親 エレメント B がエレメント A の下にある場合には、エレメント A はエレメント B の親となります。

例: この図は、Business Sales エレメントの親を示しています。



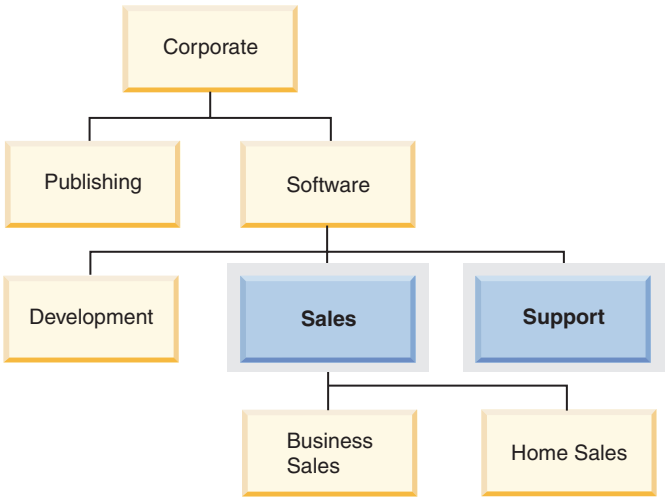
子 エレメント A がエレメント B の下にある場合には、エレメント A はエレメント B の子となります。

例: この図は、Software エレメントの子を示しています。



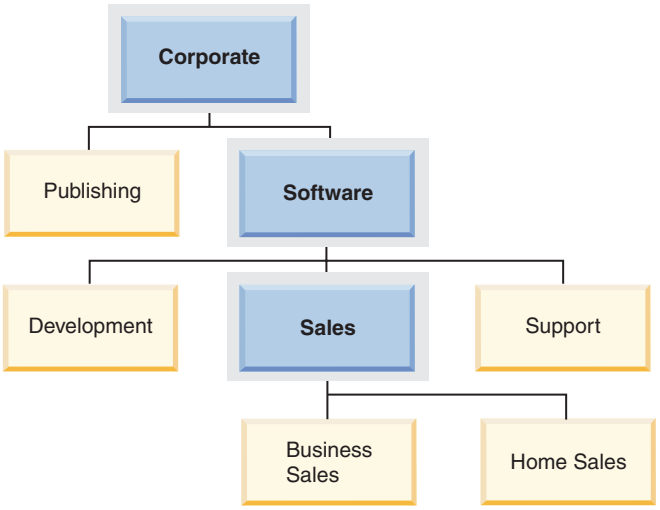
兄弟　2 つのエレメントが同じ親を持つ場合には、それらは互いに兄弟です。

例: この図は、Development エレメントの兄弟を示しています。



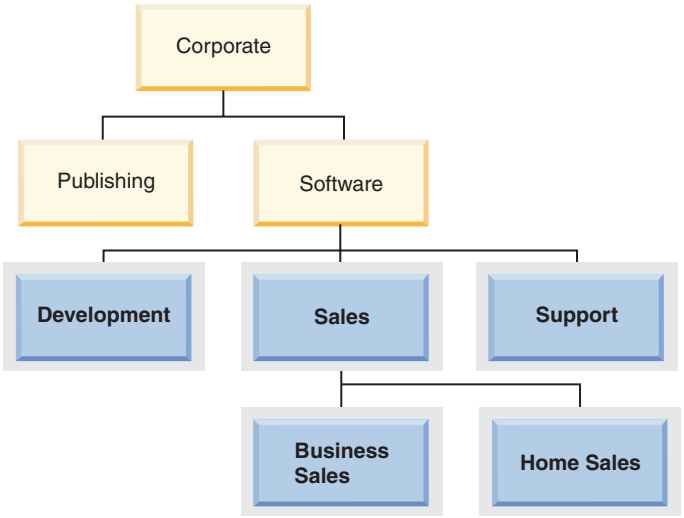
上位　エレメント A が B の親である場合、またはエレメント A が B の親の親である場合 (以下同様の条件が続く) には、エレメント A はエレメント B の上位となります。ルート・エレメントは、ツリー内の他のすべてのエレメントの上位です。

例: この図は、Home Sales エレメントの上位を示しています。



下層　エレメント A が B の子である場合、またはエレメント A が B の子の子である場合 (以下同様の条件が続く) には、エレメント A はエレメント B の下層となります。

例: この図は、Software エレメントの下層を示しています。



LBAC セキュリティー・ラベル

ラベル・ベースのアクセス制御 (LBAC) において、セキュリティー・ラベル は、セキュリティー基準の特定のセットを記述するデータベース・オブジェクトです。セキュリティー・ラベルは、データを保護するためにデータに適用されます。ユーザーが保護データにアクセスすることを許可するために、セキュリティー・ラベルは、ユーザーに付与されます。

ユーザーが保護データへのアクセスを試行すると、ユーザーのセキュリティー・ラベルが、データを保護しているセキュリティー・ラベルと比較されます。セキュリティー・ラベルには、保護しているセキュリティー・ラベルによってブロックされるものと、そうでないものがあります。あるユーザーのセキュリティー・ラベルがブロックされると、そのユーザーはデータにアクセスすることができません。

どのセキュリティー・ラベルもただ 1 つのセキュリティー・ポリシーの一部となっており、そのセキュリティー・ポリシーのコンポーネントごとに 1 つの値が含まれます。セキュリティー・ラベル・コンポーネントについて言及する場合の *値* とは、そのコンポーネントにより許可されるゼロ個以上のエレメントが含まれるリストのことを言います。 ARRAY タイプのコンポーネントの値には、ゼロ個または 1 個のエレメントを含めることができ、他のタイプの値には、ゼロ個以上のエレメントを含めることができます。エレメントが含まれない値は、*空の値* と呼ばれます。

例: TREE タイプのコンポーネントに、人的資源、販売、および配送という 3 つのエレメントが含まれる場合には、以下のものはそのコンポーネントの有効な値の一部となります。

- 人的資源 (または任意のエレメントがそれ自体で有効です)
- 人的資源、配送 (またはエレメントの他の任意の組み合わせ。ただし同じエレメントを複数回含めることはできません)
- *空の値*

特定のセキュリティー・ラベルが別のセキュリティー・ラベルをブロックするかどうかは、ラベル内の各コンポーネントの値と、表のセキュリティー・ポリシーで指定される LBAC 規則セットにより決定されます。比較をする方法の詳細は、LBAC セキュリティー・ラベルの比較方法を取り上げているトピックに説明されています。

セキュリティー・ラベルがテキスト・ストリングに変換されるときは、セキュリティー・ラベル値のフォーマットを取り上げているトピックに説明されているフォーマットを使用します。

セキュリティー・ラベルの作成

セキュリティー・ラベルを作成する人は、セキュリティー管理者でなければなりません。セキュリティー・ラベルは、SQL ステートメントの CREATE SECURITY LABEL で作成します。セキュリティー・ラベルを作成する際には、以下を指定します。

- ラベルの名前
- そのラベルが含まれるセキュリティー・ポリシー
- セキュリティー・ポリシーに含まれる 1 つ以上のコンポーネントの値

値が指定されていないコンポーネントでは、空の値を持つと想定されます。セキュリティー・ラベルには、少なくとも 1 つの空以外の値が含まれていなければなりません。

セキュリティー・ラベルの変更

セキュリティー・ラベルは、変更することができません。セキュリティー・ラベルを変更する唯一の方法は、それをドロップし、再作成することです。しかし、セキュリティー・ラベルのコンポーネント であれば、セキュリティー管理者が変更できます (ALTER SECURITY LABEL COMPONENT ステートメントを使用)。

セキュリティー・ラベルのドロップ

セキュリティー・ラベルをドロップする人は、セキュリティー管理者でなければなりません。セキュリティー・ラベルは、SQL ステートメントの DROP でドロップします。データベース内のどこかにあるデータを保護するために使用されているか、1 人以上のユーザーにより現在保持されているセキュリティー・ラベルは、ドロップできません。

セキュリティー・ラベルの付与

ユーザー、グループ、またはロールにセキュリティー・ラベルを付与する人は、セキュリティー管理者でなければなりません。セキュリティー・ラベルを付与するには、SQL ステートメントの GRANT SECURITY LABEL を使用します。セキュリティー・ラベルを付与する際、読み取りアクセス、書き込みアクセス、または読み取りと書き込み両方のアクセスに対して付与することができます。ユーザー、グループ、またはロールは、同じタイプのアクセスの場合に同じセキュリティー・ポリシーの複数のセキュリティー・ラベルを保有することはできません。

セキュリティー・ラベルの取り消し

ユーザー、グループ、またはロールのセキュリティー・ラベルを取り消す人は、セキュリティー管理者でなければなりません。セキュリティー・ラベルを取り消すには、SQL ステートメントの REVOKE SECURITY LABEL を使用します。

セキュリティー・ラベルと互換性のあるデータ・タイプ

セキュリティー・ラベルは、SYSPROC.DB2SECURITYLABEL のデータ・タイプを持っています。 SYSPROC.DB2SECURITYLABEL と VARCHAR(128) FOR BIT DATA の間で、データ変換がサポートされています。

セキュリティー・ラベル値の形式

セキュリティー・ラベルの値を、文字ストリングの形式で表すことがあります (たとえば組み込み関数 SECLABEL を使用するとき)。セキュリティー・ラベルの値をストリングで表すときは、次の形式を使用します。

- コンポーネントの値は、CREATE SECURITY POLICY ステートメントでのセキュリティー・ポリシーのコンポーネントのリストと同じ順序で、左から右にリストする。
- エレメントは、そのエレメントの名前で表す。
- コンポーネントが異なるエレメントは、コロン (:) で分離する。
- 同一コンポーネントに複数のエレメントを指定する場合は、エレメントを括弧 (()) で囲み、コンマ (,) で分離する。
- 空の値は、一組の空括弧 (()) で表す。

例: セキュリティー・ラベルは、Level、Department、Projects という 3 つのコンポーネントをこの順序で持つセキュリティー・ポリシーの一部です。このセキュリティー・ラベルは次の値を持ちます。

表 9.

コンポーネント	値
Level	Secret
Department	*空の値*
Projects	• Epsilon 37 • Megaphone • Cloverleaf

このセキュリティー・ラベル値をストリングで表すと、次のようになります。

```
'Secret:():(Epsilon 37,Megaphone,Cloverleaf)'
```

LBAC セキュリティー・ラベルが比較される方法

ラベル・ベースのアクセス制御 (LBAC) により保護されたデータへのアクセスを試行すると、そのユーザーの LBAC 信用証明情報が 1 つ以上のセキュリティー・ラベルと比較され、アクセスがブロックされるかどうかが確認されます。 LBAC 信用証明情報には、保持しているすべてのセキュリティー・ラベルに加え、保持しているすべての免除が含まれます。

行うことができる比較は、2 つのタイプしかありません。LBAC 信用証明情報と読み取りアクセス用の単一のセキュリティー・ラベルとの比較、または LBAC 信用証明情報と書き込みアクセス用の単一のセキュリティー・ラベルとの比較をすることができます。更新および削除は、読み取りの後に書き込みをするものとして処理されます。操作において、複数の比較を行う必要がある場合、それぞれの比較は別個に実行されます。

どのセキュリティー・ラベルが使用されるか

複数のセキュリティー・ラベルを保持することが可能ですが、1 つだけが保護セキュリティー・ラベルと比較されます。使用されるラベルは、以下の基準を満たすものです。

- アクセスされている表を保護しているセキュリティー・ポリシーの一部である。
- アクセスのタイプ (読み取りまたは書き込み) 用に付与されたものである。

これらの基準を満たすセキュリティー・ラベルを持っていない場合には、すべてのコンポーネントに対して空の値を持つデフォルトのセキュリティー・ラベルが想定されます。

比較が行われる方法

セキュリティー・ラベルは、コンポーネントごとに比較されます。セキュリティー・ラベルにいずれかのコンポーネントの値が含まれていない場合には、空の値が想定されます。各コンポーネントを調べる際、そのコンポーネントのユーザーの値に含まれるエレメントが保護ラベル内の同じコンポーネントの値に含まれるエレメントによりブロックされるかどうかを決定するために LBAC 規則セットの該当する規則が使用されます。ユーザーのいずれかの値がブロックされる場合には、LBAC 信用証明情報は保護セキュリティー・ラベルによりブロックされます。

比較で使用される LBAC 規則セットは、セキュリティー・ポリシーで指定されます。どんな規則か、および各規則がいつ使用されるかについて調べるには、その規則セットの説明を参照してください。

免除が比較に与える影響

2 つの値を比較するために使用している規則に対する免除を保持している場合には、その比較は行われず、保護値はセキュリティー・ラベル内の値をブロックしないと想定されます。

例: LBAC 規則セットは DB2LBACRULES で、セキュリティー・ポリシーには 2 つのコンポーネントがあります。一方のコンポーネントはタイプ ARRAY で、他方はタイプ TREE です。ユーザーには、規則 DB2LBACREADTREE に対する免除が付与されています。この規則は、タイプ TREE のコンポーネントの値同士を比較する際に読み取りアクセス用に使用される規則です。ユーザーが保護データの読み取りを試行する場合には、その規則は使用されないため、TREE コンポーネント用にユーザーが持っている値はいずれも (それが空の値である場合であっても)、アクセスをブロックしません。ユーザーがデータを読み取ることができるかどうかは、ラベルの ARRAY コンポーネントの値に完全に依存しています。

LBAC 規則セットの概要

LBAC 規則セットは、セキュリティー・ラベルを比較する際に使用される事前定義された規則のセットです。 2 つのセキュリティー・ラベルの値が比較される際、一方の値が別の値をブロックするかどうかを判別するために、規則セット内の 1 つ以上の規則が使用されます。

それぞれの LBAC 規則セットは、固有の名前で識別されます。セキュリティー・ポリシーを作成する際、そのポリシーで使用される LBAC 規則セットを指定する必要があります。そのポリシーの一部であるセキュリティー・ラベル同士を比較する際には、その LBAC 規則セットを使用します。

規則セット内のそれぞれの規則も、固有の名前で識別されます。その規則に免除を付与する際には規則の名前を使用します。

1 つのセットに含まれる規則の数や、それぞれの規則が使用される時期は、規則セットごとに異なる場合があります。

サポートされる LBAC 規則セットは、現在 1 つだけです。その規則セットの名前は DB2LBACRULES です。

LBAC 規則セット: DB2LBACRULES

DB2LBACRULES LBAC 規則セットは、セキュリティー・ラベル・コンポーネントの値を比較するための従来型の規則のセットを提供します。これは、ライトアップおよびライトダウン両方から保護します。

ライトアップおよびライトダウンの説明

ライトアップおよびライトダウンは、タイプ ARRAY のコンポーネントの書き込みアクセスのみに適用されます。ライトアップは、書き込み対象のデータを保護する

値が自分の値より高い場合に発生します。ライトダウンは、そのデータを保護する値が自分の値より低い場合に発生します。デフォルトではライトアップもライトダウンも許可されません。つまり、自分が持っている値と同じ値で保護されているデータしか書き込むことはできません。

同じコンポーネントに対する 2 つの値を比較する際、どちらの規則が使用されるかは、コンポーネントのタイプ (ARRAY、SET、または TREE) およびどのタイプのアクセス (読み取りまたは書き込み) が試行されているかにより異なります。次の表は、規則をリストし、それぞれの規則がいつ使用されるかを示し、アクセスがブロックされるかどうかを規則が判別する方法を説明しています。

表 10. DB2LBACRULES 規則のサマリー

規則名	次のタイプのコンポーネントの値を比較する場合に使用	次のタイプのアクセスを試行する場合に使用	次の条件が満たされた場合にアクセスをブロック
DB2LBACREADARRAY	ARRAY	読み取り	ユーザーの値が、保護値より低い。
DB2LBACREADSET	SET	読み取り	ユーザーが保持しない保護値が 1 つ以上ある。
DB2LBACREADTREE	TREE	読み取り	ユーザーの値がいずれも、保護値のいずれとも等しくないか、その上位でない。
DB2LBACWRITEARRAY	ARRAY	書き込み	ユーザーの値が保護値より高いか、保護値より低い。[1]
DB2LBACWRITESET	SET	書き込み	ユーザーが保持しない保護値が 1 つ以上ある。
DB2LBACWRITETREE	TREE	書き込み	ユーザーの値がいずれも、保護値のいずれとも等しくないか、その上位でない。

注:

1. DB2LBACWRITEARRAY 規則は、2 つの異なる規則が結合したものであると考えることができます。一方は自分のレベルより高いデータに書き込むこと (ライトアップ) を防ぎ、他方は自分のレベルより低いデータに書き込む (ライトダウン) ことを防ぎます。この規則に免除を付与すると、ユーザーをこれらの規則の一方または両方から免除することができます。

規則が空の値を処理する方法

すべての規則は空の値を同じ方法で処理します。空の値は他の値をブロックせず、空以外のすべての値によりブロックされます。

DB2LBACREADSET および DB2LBACWRITESET の例

以下の例は、保護データの読み取りまたは書き込みを試行しているユーザーに対して有効です。ここでは、値は、one two three four というエレメントを持つタイプ SET のコンポーネント用であると想定します。

表 11. DB2LBACREADSET および DB2LBACWRITESET 規則を適用する例

ユーザーの値	保護値	アクセスのブロック
'one'	'one'	ブロックされません。値は同じです。
'(one,two,three)'	'one'	ブロックされません。ユーザーの値にはエレメント 'one' が含まれています。

表 11. *DB2LBACREADSET および DB2LBACWRITESET 規則を適用する例 (続き)*

ユーザーの値	保護値	アクセスのブロック
’(one,two)’	’(one,two,four)’	ブロックされます。エレメント ’four’ は保護値には含まれていますが、ユーザーの値には含まれていません。
’()’	’one’	ブロックされます。空の値は空以外のすべての値によりブロックされます。
’one’	’()’	ブロックされません。空の値ではどの値もブロックされません。
’()’	’()’	ブロックされません。空の値ではどの値もブロックされません。

DB2LBACREADTREE および DB2LBACWRITETREE

以下の例は、読み取りアクセスと書き込みアクセスの両方に有効です。ここでは、TREE タイプのコンポーネントの値が以下の方法で定義されたと想定します。

```
CREATE SECURITY LABEL COMPONENT mycomp
TREE (
    'Corporate'      ROOT,
    'Publishing'     UNDER 'Corporate',
    'Software'       UNDER 'Corporate',
    'Development'    UNDER 'Software',
    'Sales'          UNDER 'Software',
    'Support'        UNDER 'Software'
        'Business Sales' UNDER 'Sales'
    'Home Sales'     UNDER 'Sales'
)
```

これは、エレメントが次のように配置されていることを意味します。

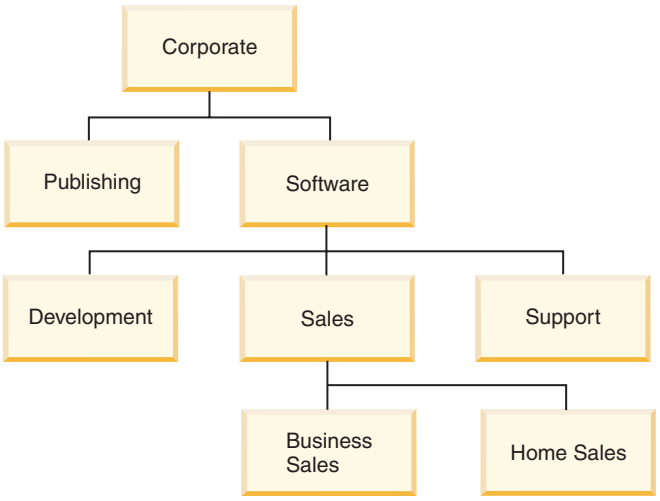


表 12. DB2LBACREADTREE および DB2LBACWRITETREE 規則を適用する例

ユーザーの値	保護値	アクセスのブロック
’(Support,Sales)’	’Development’	ブロックされます。エレメント ’Development’ はユーザーの値の 1 つではなく、’Support’ および ’Sales’ のいずれも ’Development’ の上位ではありません。
’(Development,Software)’	’(Business Sales,Publishing)’	ブロックされません。エレメント ’Software’ は ’Business Sales’ の上位です。
’(Publishing,Sales)’	’(Publishing,Support)’	ブロックされません。エレメント ’Publishing’ は両方の値のセットに含まれています。
’Corporate’	’Development’	ブロックされません。ルート値は他のすべての値の上位です。
’()’	’Sales’	ブロックされます。空の値は空以外のすべての値によりブロックされます。
’Home Sales’	’()’	ブロックされません。空の値ではどの値もブロックされません。
’()’	’()’	ブロックされません。空の値ではどの値もブロックされません。

DB2LBACREADARRAY の例

以下の例は読み取りアクセス専用です。ここでは、値は、以下の配置における以下のエレメントが含まれるタイプ ARRAY のコンポーネント用であると想定します。

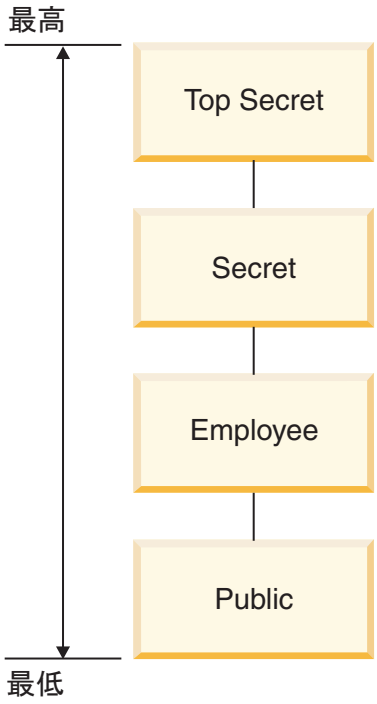


表 13. DB2LBACREADARRAY 規則を適用する例

ユーザーの値	保護値	読み取りアクセスのブロック
'Secret'	'Employee'	ブロックされません。エレメント 'Secret' はエレメント 'Employee' より高位にあります。
'Secret'	'Secret'	ブロックされません。値は同じです。
'Secret'	'Top Secret'	ブロックされます。エレメント 'Top Secret' はエレメント 'Secret' より高位にあります。
'()'	'Public'	ブロックされます。空の値は空以外のすべての値によりブロックされます。
'Public'	'()'	ブロックされません。空の値ではどの値もブロックされません。
'()'	'()'	ブロックされません。空の値ではどの値もブロックされません。

DB2LBACWRITEARRAY の例

以下の例は書き込みアクセス専用です。ここでは、値は、以下の配置における以下のエレメントが含まれるタイプ ARRAY のコンポーネント用であると想定します。

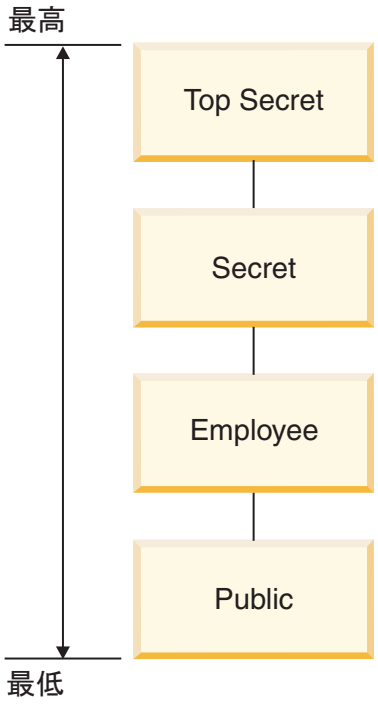


表 14. DB2LBACWRITEARRAY 規則を適用する例

ユーザーの値	保護値	書き込みアクセスのブロック
'Secret'	'Employee'	ブロックされます。エレメント 'Employee' はエレメント 'Secret' より下位にあります。
'Secret'	'Secret'	ブロックされません。値は同じです。
'Secret'	'Top Secret'	ブロックされます。エレメント 'Top Secret' はエレメント 'Secret' より高位にあります。
'()'	'Public'	ブロックされます。空の値は空以外のすべての値によりブロックされます。
'Public'	'()'	ブロックされません。空の値ではどの値もブロックされません。
'()'	'()'	ブロックされません。空の値ではどの値もブロックされません。

LBAC 規則の免除

特定のセキュリティー・ポリシーの特定の規則に関する LBAC 規則免除を保有していれば、そのセキュリティー・ポリシーによって保護されているデータにアクセスしようとした場合、その規則は施行されません。

付与されたセキュリティー・ポリシー以外のセキュリティー・ポリシーのセキュリティー・ラベルを比較する場合には、免除による影響はありません。

例:

T1 および T2 という 2 つの表があります。 T1 はセキュリティー・ポリシー P1 によって保護されており、T2 はセキュリティー・ポリシー P2 によって保護されて

います。両方のセキュリティー・ポリシーは 1 つのコンポーネントを持っています。それぞれのコンポーネントのタイプは ARRAY です。T1 および T2 にはそれぞれ、1 行のデータのみが含まれます。セキュリティー・ポリシー P1 のもとで読み取りアクセスに対して保持しているセキュリティー・ラベルにより、T1 の行にアクセスすることはできません。セキュリティー・ポリシー P2 のもとで読み取りアクセスに対して保持しているセキュリティー・ラベルにより、T2 の行に読み取りアクセスすることはできません。

ここで、P1 の下の DB2LBACREADARRAY に対する免除が付与されます。T1 から行を読み取ることはできますが、T2 から読み取ることはできません。これは、T2 が別のセキュリティー・ポリシーによって保護されており、そのポリシー内の DB2LBACREADARRAY 規則に対する免除が保持されていないためです。

複数の免除を保持することができます。セキュリティー・ポリシーによって使用されるすべての規則に対する免除を保持している場合、そのセキュリティー・ポリシーによって保護されているすべてのデータに対する完全なアクセス権を持ちます。

LBAC 規則の免除の付与

LBAC 規則の免除を付与する人は、セキュリティー管理者でなければなりません。LBAC 規則の免除を付与するには、SQL ステートメント GRANT EXEMPTION ON RULE を使用します。

LBAC 規則を付与する場合、以下の情報を提供します。

- 免除の対象となる 1 つ以上の規則
- 免除の対象となるセキュリティー・ポリシー
- 免除を付与する対象のユーザー、グループ、またはロール

重要: LBAC 規則の免除により、非常に強力なアクセス権が提供されます。アクセス権を付与する場合には、注意深く考慮してください。

LBAC 規則の免除の取り消し

LBAC 規則の免除を取り消す人は、セキュリティー管理者でなければなりません。LBAC 規則の免除を取り消すには、SQL ステートメント REVOKE EXEMPTION ON RULE を使用します。

LBAC セキュリティー・ラベルを管理するための組み込み関数

ラベル・ベースのアクセス制御 (LBAC) セキュリティー・ラベルを管理するために、組み込み関数 SECLABEL、SECLABEL_BY_NAME、および SECLABEL_TO_CHAR が提供されています。

それぞれの組み込み関数の概要はここで説明されており、詳細については「*SQL リファレンス*」で説明されています。

SECLABEL

この組み込み関数は、セキュリティー・ポリシーと、ラベル内の各コンポーネントの値を指定することによりセキュリティー・ラベルを作成するために使用されます。戻り値は DB2SECURITYLABEL のデータ・タイプを持っているセキュリティ

ー・ラベルであり、そのセキュリティー・ラベルは指示されたセキュリティー・ポリシーの一部で、コンポーネント用の指示された値を持っています。指示された値を持つセキュリティー・ラベルがすでに存在している必要はありません。

例: 表 T1 には 2 つの列があり、最初の列のデータ・タイプは DB2SECURITYLABEL で、2 番目の列のデータ・タイプは INTEGER です。T1 はセキュリティー・ポリシー P1 により保護され、P1 には level、departments、および groups という 3 つのセキュリティー・ラベル・コンポーネントがあります。UNCLASSIFIED がコンポーネント level のエレメントであり、ALPHA および SIGMA が両方ともコンポーネント departments のエレメントであり、G2 がコンポーネント groups のエレメントである場合には、セキュリティー・ラベルは次のように挿入できます。

```
INSERT INTO T1 VALUES
   ( SECLABEL( 'P1', 'UNCLASSIFIED:(ALPHA,SIGMA):G2' ), 22 )
```

SECLABEL_BY_NAME

この組み込み関数は、セキュリティー・ポリシーの名前と、そのセキュリティー・ポリシーの一部であるセキュリティー・ラベルの名前を受け入れます。この組み込み関数はその後、指示されたセキュリティー・ラベルを DB2SECURITYLABEL として戻します。DB2SECURITYLABEL のデータ・タイプを持つ列に既存のセキュリティー・ラベルを挿入する際にこの関数を使用する必要があります。

例: 表 T1 には 2 つの列があり、最初の列のデータ・タイプは DB2SECURITYLABEL で、2 番目の列のデータ・タイプは INTEGER です。L1 という名前のセキュリティー・ラベルは、セキュリティー・ポリシー P1 の一部です。次の SQL は、セキュリティー・ラベルを挿入します。

```
INSERT INTO T1 VALUES ( SECLABEL_BY_NAME( 'P1', 'L1' ), 22 )
```

次の SQL ステートメントは作動しません。

```
INSERT INTO T1 VALUES ( P1.L1, 22 )     // Syntax Error!
```

SECLABEL_TO_CHAR

この組み込み関数は、セキュリティー・ラベルを構成する値のストリング表記を戻します。

例: 表 T1 の列 C1 のデータ・タイプは DB2SECURITYLABEL です。T1 はセキュリティー・ポリシー P1 により保護され、P1 には level、departments、および groups という 3 つのセキュリティー・ラベル・コンポーネントがあります。T1 には 1 つの行があり、列 C1 には各コンポーネントに以下のエレメントを持つ値があります。

コンポーネント	エレメント
level	SECRET
departments	DELTA および SIGMA
グループ	G3

行の読み取りを許可する LBAC 信用証明情報を持つユーザーは、次の SQL ステートメントを実行します。

```
SELECT SECLABEL_TO_CHAR( 'P1', C1 ) AS C1 FROM T1
```

出力は次のようになります。

```
C1

'SECRET:(DELTA,SIGMA):G3'
```

LBAC を使用したデータの保護

ラベル・ベースのアクセス制御 (LBAC) を使用すると、データの行またはデータの列の一方または両方を保護することができます。表内のデータは、表を保護するセキュリティー・ポリシーの一部であるセキュリティー・ラベルによってのみ保護できます。データ保護 (セキュリティー・ポリシーの追加を含む) は、表の作成時に、またはそれ以降に表を変更することによって行うことができます。

セキュリティー・ポリシーを表に追加し、同じ CREATE TABLE または ALTER TABLE ステートメントの一部としてその表のデータを保護することができます。

一般規則として、現行の LBAC 信用証明情報がデータへの書き込みを許可しないようにそのデータを保護することは、許可されません。

表へのセキュリティー・ポリシーの追加

CREATE TABLE ステートメントの SECURITY POLICY 節を使用して表を作成するときに、セキュリティー・ポリシーを表に追加することができます。 ALTER TABLE ステートメントの ADD SECURITY POLICY 節を使用して、セキュリティー・ポリシーを既存の表に追加することができます。表にセキュリティー・ポリシーを追加するために SECADM 権限または LBAC 信用証明情報は必要ではありません。

セキュリティー・ポリシーは、LBAC により保護できない表のタイプには追加できません。LBAC で保護できない表タイプのリストについては、LBAC の概説を参照してください。

どの表にも複数のセキュリティー・ポリシーを追加することはできません。

行の保護

表を作成する際に DB2SECURITYLABEL のデータ・タイプの列を組み込むことにより、新しい表内の保護された行を許可することができます。さらに CREATE TABLE ステートメントでは、セキュリティー・ポリシーを表に追加する必要があります。そのような表を作成するために SECADM 権限または LBAC 信用証明情報は必要ではありません。

DB2SECURITYLABEL のデータ・タイプを持つ列を追加することにより、既存の表内の保護された行を許可することができます。そのような列を追加するためには、表がすでにセキュリティー・ポリシーで保護されているか、列を追加する ALTER TABLE ステートメントもセキュリティー・ポリシーを表に追加するかのいずれかでなければなりません。列が追加されると、既存のすべての行を保護するために、書き込みアクセス用に保持しているセキュリティー・ラベルが使用されます。表を保

護するセキュリティー・ポリシーの一部である書き込みアクセス用のセキュリティー・ラベルを保持していない場合には、DB2SECURITYLABEL のデータ・タイプを持つ列を追加することはできません。

表にタイプ DB2SECURITYLABEL の列が組み込まれた後、その列にセキュリティー・ラベルを保管することにより、それぞれの新しいデータの行を保護します。これがどのようになるかの詳細は、LBAC 保護データの挿入および更新に関するトピックに説明されています。タイプ DB2SECURITYLABEL の列を持つ表に行を挿入するには、LBAC 信用証明情報が必要です。

DB2SECURITYLABEL のデータ・タイプを持つ列は、ドロップできず、他のデータ・タイプに変更できません。

列の保護

CREATE TABLE ステートメントの SECURED WITH 列オプションを使用して表を作成するときに、列を保護することができます。 ALTER TABLE ステートメントの SECURED WITH オプションを使用して、既存の列に保護を追加することができます。

特定のセキュリティー・ラベルで列を保護するには、そのセキュリティー・ラベルにより保護されるデータへの書き込みを許可する LBAC 信用証明情報がなければなりません。 SECADM 権限を持っている必要はありません。

列は、表を保護するセキュリティー・ポリシーの一部であるセキュリティー・ラベルによってのみ保護できます。セキュリティー・ポリシーを持たない表の列を保護することはできません。セキュリティー・ポリシーで表を保護して、同じステートメントで 1 つ以上の列を保護することが許可されています。

表内の任意の数の列を保護することができますが、1 つの列を複数のセキュリティー・ラベルで保護することはできません。

LBAC 保護データの読み取り

ラベル・ベースのアクセス制御 (LBAC) により保護されたデータの読み取りを試行すると、読み取り用の LBAC 信用証明情報が、データを保護しているセキュリティー・ラベルと比較されます。保護ラベルが信用証明情報をブロックしない場合、データを読み取ることが許可されます。

保護された列の場合、保護セキュリティー・ラベルは、表のスキーマで定義されます。その列の保護セキュリティー・ラベルは、表のすべての行において同じです。保護された行の場合、保護セキュリティー・ラベルは、タイプ DB2SECURITYLABEL の列の行に保管されます。それは、表内の行ごとに異なる場合があります。

LBAC 信用証明情報がセキュリティー・ラベルと比較される方法の詳細は、LBAC セキュリティー・ラベルの比較方法に関するトピックで説明されています。

保護された列の読み取り

保護された列からの読み取りを試行する際、LBAC 信用証明情報はその列を保護するセキュリティー・ラベルと比較されます。この比較を基にして、アクセスはブロックまたは許可されます。アクセスがブロックされる場合にはエラーが戻され、ステートメントは失敗します。そうでない場合は、ステートメントは通常通り進行します。

LBAC 信用証明情報が読み取りを許可しない列の読み取りを試行すると、ステートメント全体が失敗します。

例:

表 T1 には保護された列が 2 つあります。列 C1 はセキュリティー・ラベル L1 により保護されています。列 C2 はセキュリティー・ラベル L2 により保護されています。

ユーザー Jyoti は、セキュリティー・ラベル L1 へのアクセスを許可する読み取り用 LBAC 信用証明情報を持っているものの、L2 に対するものは持っていないと想定します。Jyoti が次の SQL ステートメントを発行すると、ステートメントは失敗します。

```
SELECT * FROM T1
```

SELECT 節に、ワイルドカード (*) の一部として列 C2 が含まれているために、ステートメントは失敗します。

Jyoti が次の SQL ステートメントを発行すると、それは成功します。

```
SELECT C1 FROM T1
```

SELECT 節の中で保護されている列は C1 のみで、Jyoti の LBAC 信用証明情報は Jyoti がその列を読み取ることを許可しています。

保護された行の読み取り

ある行を読み取ることを許可する LBAC 信用証明情報をあるユーザーが持っていない場合、そのユーザーにとっては、その行は存在していないかのようになります。

保護された行を読み取る際、LBAC 信用証明情報が読み取りアクセスを許可する行のみが戻されます。タイプ DB2SECURITYLABEL の列が SELECT 節の一部でない場合でも、そのように処理されます。

ユーザーの LBAC 信用証明情報に応じて、保護された行を持つ表では、異なるユーザーには異なる行が表示される可能性があります。たとえば、T1 に保護された行があり、2 人のユーザーが異なる LBAC 信用証明情報を持っている場合、ステートメント `SELECT COUNT(*) FROM T1` を実行するそれらユーザーは、異なる結果を受け取る可能性があります。

LBAC 信用証明情報は、SELECT ステートメントだけでなく、UPDATE、DELETE のような他の SQL ステートメントにも影響します。ある行を読み取ることを許可する LBAC 信用証明情報を持っていない場合、その行に影響を与えることはできません。

例:

表 T1 には以下のような行と列があります。列 ROWSECURITYLABEL のデータ・タイプは DB2SECURITYLABEL です。

表 15.

LASTNAME	DEPTNO	ROWSECURITYLABEL
Rjaibi	55	L2
Miller	77	L1
Fielding	11	L3
Bird	55	L2

ユーザー Dan は、セキュリティー・ラベル L1 により保護されるデータの読み取りを許可する LBAC 信用証明情報を持っているものの、L2 または L3 により保護されるデータに対するものは持っていないと想定します。

Dan は次の SQL ステートメントを発行します。

```
SELECT * FROM T1
```

SELECT ステートメントは Miller の行のみを戻します。エラー・メッセージや警告は戻されません。

表 T1 の Dan のビューは次のようになります。

表 16.

LASTNAME	DEPTNO	ROWSECURITYLABEL
Miller	77	L1

Rjaibi、Fielding、および Bird の行は戻されません。なぜなら読み取りアクセスはそれらのセキュリティー・ラベルによりブロックされるからです。 Dan はこれらの行を削除または更新することはできません。これらの行は、集約関数に組み込むこともできません。Dan にとっては、これらの行は存在していないかのようになります。

Dan は次の SQL ステートメントを発行します。

```
SELECT COUNT(*) FROM T1
```

Miller の行しかユーザー Dan は読み取ることができないため、ステートメントは 1 の値を戻します。

保護された列を含む保護された行の読み取り

列のアクセスは、行のアクセスの前にチェックされます。選択している列のいずれかを保護しているセキュリティー・ラベルによって読み取りアクセス用の LBAC 信用証明情報がブロックされる場合には、ステートメント全体が失敗します。ブロックされない場合、ステートメントは継続し、LBAC 信用証明情報が読み取りアクセスを許可する対象であるセキュリティー・ラベルにより保護される行のみが戻されます。

例

表 T1 の列 LASTNAME はセキュリティー・ラベル L1 で保護されています。列 DEPTNO はセキュリティー・ラベル L2 で保護されています。列 ROWSECURITYLABEL のデータ・タイプは DB2SECURITYLABEL です。T1 (データを含む) は次のようになります。

表 17.

LASTNAME *L1 により保護される*	**DEPTNO** *L2 により保護される*	**ROWSECURITYLABEL**
Rjaibi	55	L2
Miller	77	L1
Fielding	11	L3

ユーザー Sakari は、セキュリティー・ラベル L1 により保護されるデータの読み取りを許可する LBAC 信用証明情報を持っているものの、L2 または L3 に対するものは持っていないと想定します。

Sakari は次の SQL ステートメントを発行します。

```
SELECT * FROM T1
```

SELECT 節は列 DEPTNO が含まれるワイルドカード (*) を使用しているため、ステートメントは失敗します。列 DEPTNO は、Sakari の LBAC 信用証明情報が Sakari に読み取りを許可しないセキュリティー・ラベル L2 により保護されています。

Sakari は今度は次の SQL ステートメントを発行します。

```
SELECT LASTNAME, ROWSECURITYLABEL FROM T1
```

SELECT 節には Sakari が読み取りできない列が含まれていないため、ステートメントは継続します。ただし、他の各行はセキュリティー・ラベル L2 または L3 により保護されるため、1 つの行のみが戻されます。

表 18.

LASTNAME	**ROWSECURITYLABEL**
Miller	L1

LBAC 保護データの挿入

保護された列への挿入

保護された列に明示的にデータの挿入を試行する際、書き込み用の LBAC 信用証明情報はその列を保護するセキュリティー・ラベルと比較されます。この比較を基にして、アクセスはブロックまたは許可されます。

2 つのセキュリティー・ラベルが比較される方法の詳細は、LBAC セキュリティー・ラベルの比較方法に関するトピックで説明されています。

アクセスが許可される場合、ステートメントは通常通り進行します。アクセスがブロックされる場合には挿入は失敗し、エラーが戻されます。

行を挿入しているものの、保護される列の値を指定しない場合には、デフォルト値がある場合にはそれが挿入されます。これは、LBAC 信用証明情報がその列への書き込みアクセスを許可しない場合であっても、そのように処理されます。デフォルトは次の場合に使用可能です。

- 列が WITH DEFAULT オプションで宣言された
- 列が生成された列である
- 列には、BEFORE トリガーにより指定されるデフォルト値がある
- 列のデータ・タイプは DB2SECURITYLABEL で、その場合、書き込みアクセス用に保持するセキュリティー・ラベルはデフォルト値である

保護された行への挿入

保護された行を持つ表に新しい行を挿入する際、タイプ DB2SECURITYLABEL の列には値を指定する必要はありません。その列に値を指定しない場合、列には、書き込みアクセス用にユーザーに付与されたセキュリティー・ラベルが自動的に取り込まれます。書き込みアクセス用のセキュリティー・ラベルがユーザーに付与されていない場合、エラーが戻され、挿入は失敗します。

SECLABEL のような組み込み関数を使用することにより、タイプ DB2SECURITYLABEL の列に挿入されるセキュリティー・ラベルを明示的に指定することができます。ただし、指定したセキュリティー・ラベルは、挿入を試行しているセキュリティー・ラベルで保護されるデータへの書き込みを LBAC 信用証明情報が許可している場合にのみ、使用されます。

書き込みできないセキュリティー・ラベルを指定した場合には、行われる処理は、表を保護しているセキュリティー・ポリシーにより異なります。セキュリティー・ポリシーに RESTRICT NOT AUTHORIZED WRITE SECURITY LABEL オプションがある場合は、挿入が失敗し、エラーが戻されます。セキュリティー・ポリシーに RESTRICT NOT AUTHORIZED WRITE SECURITY LABEL オプションがない場合や、代わりに OVERRIDE NOT AUTHORIZED WRITE SECURITY LABEL オプションがある場合は、指定したセキュリティー・ラベルが無視され、書き込みアクセス用に保持しているセキュリティー・ラベルがあれば、そのセキュリティー・ラベルが代わりに使用されます。書き込みアクセス用のセキュリティー・ラベルを保持していない場合は、エラーが戻されます。

例

表 T1 は、RESTRICT NOT AUTHORIZED WRITE SECURITY LABEL オプションを付けずに作成された P1 という名前のセキュリティー・ポリシーにより保護されています。表 T1 には 2 つの列がありますが、行はありません。列は LASTNAME と LABEL です。列 LABEL のデータ・タイプは DB2SECURITYLABEL です。

ユーザー Joe は、書き込みアクセス用のセキュリティー・ラベル L2 を保持しています。セキュリティー・ラベル L2 は、セキュリティー・ラベル L2 により保護されるデータへの書き込みを Joe に許可するものの、セキュリティー・ラベル L1 または L3 により保護されるデータに対しては許可しないと想定します。

Joe は次の SQL ステートメントを発行します。

```
INSERT INTO T1 (LASTNAME, DEPTNO) VALUES ('Rjaibi', 11)
```

INSERT ステートメントにはセキュリティー・ラベルが含まれていなかったため、Joe の書き込みアクセス用セキュリティー・ラベルは LABEL 行に挿入されます。

表 T1 は次のようになります。

表 19.

LASTNAME	LABEL
Rjaibi	L2

Joe は以下の SQL ステートメントを発行します。その中で、列 LABEL に挿入するセキュリティー・ラベルを明示的に指定します。

```
INSERT INTO T1 VALUES ('Miller', SECLABEL_BY_NAME('P1', 'L1') )
```

ステートメント内の SECLABEL_BY_NAME 関数は、セキュリティー・ポリシー P1 の一部で L1 という名前のセキュリティー・ラベルを戻します。Joe は L1 で保護されるデータへの書き込みが許可されていないため、L1 を列 LABEL に挿入することは許可されません。

T1 を保護するセキュリティー・ポリシーは RESTRICT NOT AUTHORIZED WRITE SECURITY LABEL オプションを付けずに作成されたため、書き込み用に Joe が保持するセキュリティー・ラベルが代わりに挿入されます。エラーまたはメッセージは戻されません。

表は次のようになります。

表 20.

LASTNAME	LABEL
Rjaibi	L2
Miller	L2

もし表を保護しているセキュリティー・ポリシーが RESTRICT NOT AUTHORIZED WRITE SECURITY LABEL オプションを付けて作成されていたとすると、挿入は失敗し、エラーが戻されたはずです。

次に Joe は LBAC 規則の 1 つに対する免除が付与されます。Joe の新しい LBAC 信用証明情報は、セキュリティー・ラベル L1 および L2 で保護されるデータへの書き込みを許可すると想定します。書き込みアクセス用に Joe に付与されたセキュリティー・ラベルは変更されず、L2 のままです。

Joe は次の SQL ステートメントを発行します。

```
INSERT INTO T1 VALUES ('Bird', SECLABEL_BY_NAME('P1', 'L1') )
```

Joe の新しい LBAC 信用証明情報のため、Joe はセキュリティー・ラベル L1 により保護されるデータに書き込むことができます。そのため、L1 の追加は許可されます。表は次のようになります。

表 21.

LASTNAME	LABEL
Rjaibi	L2
Miller	L2
Bird	L1

LBAC 保護データの更新

LBAC 信用証明情報が、データへの書き込みアクセスを許可していなければ、データを更新することはできません。保護された行を更新する場合、LBAC 信用証明情報が行への読み取りアクセスも許可していなければなりません。

保護された列の更新

保護された列にあるデータの更新を試行する際、LBAC 信用証明情報はその列を保護するセキュリティー・ラベルと比較されます。行われる比較は書き込みアクセスに対するものです。書き込みアクセスがブロックされる場合にはエラーが戻され、ステートメントは失敗します。ブロックされない場合、更新は継続します。

LBAC 信用証明情報がセキュリティー・ラベルと比較される方法の詳細は、LBAC セキュリティー・ラベルの比較方法に関するトピックで説明されています。

例:

列 DEPTNO がセキュリティー・ラベル L2 により保護され、列 PAYSCALE がセキュリティー・ラベル L3 により保護される表 T1 があると想定します。T1 (そのデータを含む) は次のようになります。

表 22. 表 T1

EMPNO	LASTNAME	DEPTNO *保護ラベルは* L2	PAYSCALE *保護ラベルは* L3
1	Rjaibi	11	4
2	Miller	11	7
3	Bird	11	9

ユーザー Lhakpa には LBAC 信用証明情報がありません。Lhakpa は次の SQL ステートメントを発行します。

```
UPDATE T1 SET EMPNO = 4
   WHERE LASTNAME = "Bird"
```

このステートメントは、保護された列を更新しないため、エラーなく実行されます。 T1 は次のようになります。

表 23. 更新後の表 T1

EMPNO	LASTNAME	DEPTNO *保護ラベルは* L2	PAYSCALE *保護ラベルは* L3
1	Rjaibi	11	4
2	Miller	11	7
4	Bird	11	9

Lhakpa は今度は次の SQL ステートメントを発行します。

```
UPDATE T1 SET DEPTNO = 55
   WHERE LASTNAME = "Miller"
```

DEPTNO は保護されていて Lhakpa には LBAC 信用証明情報がないため、このステートメントは失敗し、エラーが戻されます。

Lhakpa に LBAC 信用証明情報が付与されていて、それが以下の表で要約されているアクセスを許可すると想定します。それらの信用証明情報がどんなもので、セキュリティー・ラベルにどんなエレメントが入っているかは、この例では重要ではありません。

データを保護しているセキュリティー・ラベル	読み取りの可否	書き込みの可否
L2	いいえ	はい
L3	いいえ	いいえ

Lhakpa は次の SQL ステートメントを再度発行します。

```
UPDATE T1 SET DEPTNO = 55
   WHERE LASTNAME = "Miller"
```

今度は、Lhakpa の LBAC 信用証明情報が、列 DEPTNO を保護しているセキュリティー・ラベルにより保護されるデータへの書き込みを許可しているため、ステートメントはエラーなく実行されます。その同じ列から読み取りができないことは関係ありません。 T1 のデータは次のようになります。

表 24. 2 回目の更新後の表 T1

EMPNO	LASTNAME	DEPTNO *保護ラベルは* L2	PAYSCALE *保護ラベルは* L3
1	Rjaibi	11	4
2	Miller	55	7
4	Bird	11	9

今度は Lhakpa は次の SQL ステートメントを発行します。

```
UPDATE T1 SET DEPTNO = 55, PAYSCALE = 4
   WHERE LASTNAME = "Bird"
```

列 PAYSCALE はセキュリティー・ラベル L3 により保護され、Lhakpa の LBAC 信用証明情報は Lhakpa がその列に書き込むことを許可しません。 Lhakpa はその列に書き込むことができないため、更新は失敗し、データは変更されません。

保護された行の更新

ユーザーの LBAC 信用証明情報が、ある行の読み取りを許可していない場合には、そのユーザーにとってはその行は存在していないかのようになるため、そのユーザーがその行を更新する方法はありません。読み取ることができる行においては、その更新を行うためには、行への書き込みもできなければなりません。

行の更新を試行する際、書き込み用の LBAC 信用証明情報はその行を保護するセキュリティー・ラベルと比較されます。書き込みアクセスがブロックされる場合、更新は失敗し、エラーが戻されます。書き込みアクセスがブロックされない場合には、更新は継続します。

実行される更新は、DB2SECURITYLABEL のデータ・タイプを持つ列の処理を除き、無保護の行への更新と同様に行われます。その列の値を明示的に設定しない場合、そこには書き込みアクセス用に保持しているセキュリティー・ラベルが自動的に設定されます。書き込みアクセス用のセキュリティー・ラベルを持っていない場合、エラーが戻され、ステートメントは失敗します。

更新で DB2SECURITYLABEL のデータ・タイプを持つ列を明示的に設定した場合には、LBAC 信用証明情報は再度チェックされます。実行しようとしている更新で、現行の LBAC 信用証明情報では書き込みが許可されない行が作成されることになる場合の処理は、表を保護しているセキュリティー・ポリシーによって異なります。セキュリティー・ポリシーに RESTRICT NOT AUTHORIZED WRITE SECURITY LABEL オプションがある場合は、更新が失敗し、エラーが戻されます。セキュリティー・ポリシーに RESTRICT NOT AUTHORIZED WRITE SECURITY LABEL オプションがない場合や、代わりに OVERRIDE NOT AUTHORIZED WRITE SECURITY LABEL オプションがある場合は、指定したセキュリティー・ラベルが無視され、書き込みアクセス用に保持しているセキュリティー・ラベルがあれば、そのセキュリティー・ラベルが代わりに使用されます。書き込みアクセス用のセキュリティー・ラベルを保持していない場合は、エラーが戻されます。

例:

表 T1 が、P1 という名前のセキュリティー・ポリシーにより保護され、データ・タイプが DB2SECURITYLABEL である LABEL という名前の列を持つと想定します。

T1 (そのデータを含む) は次のようになります。

表 25. 表 T1

EMPNO	LASTNAME	DEPTNO	LABEL
1	Rjaibi	11	L1
2	Miller	11	L2
3	Bird	11	L3

ユーザー Jenni は、セキュリティー・ラベル L0 および L1 により保護されるデータへの読み取りおよび書き込みを許可するものの、他のセキュリティー・ラベルにより保護されるデータに対する読み取りおよび書き込みは許可しない LBAC 信用証明情報を持っていると想定します。 Jenni が読み取りおよび書き込み両方のために保持しているセキュリティー・ラベルは L0 です。Jenni の信用証明情報全体の詳細、およびラベルにどんなエレメントが入っているかは、この例では重要ではありません。

Jenni は次の SQL ステートメントを発行します。

```
SELECT * FROM T1
```

Jenni の表には 1 行だけ表示されます。

表 26. Jenni の SELECT 照会の結果

EMPNO	LASTNAME	DEPTNO	LABEL
1	Rjaibi	11	L1

ラベル L2 および L3 により保護される行は結果セットには組み込まれません。なぜなら Jenni の LBAC 信用証明情報は、これらの行の読み取りを Jenni に許可しないからです。 Jenni にとっては、これらの行は存在していないかのようになります。

Jenni は以下の SQL ステートメントを発行します。

```
UPDATE T1 SET DEPTNO = 44 WHERE DEPTNO = 11;
SELECT * FROM T1;
```

照会により戻される結果セットは、以下のようになります。

表 27. Jenni の UPDATE & SELECT 照会の結果

EMPNO	LASTNAME	DEPTNO	LABEL
1	Rjaibi	44	L0

表の実際のデータは次のようになります。

表 28. 表 T1

EMPNO	LASTNAME	DEPTNO	LABEL
1	Rjaibi	44	L0
2	Miller	11	L2
3	Bird	11	L3

ステートメントはエラーなく実行されましたが、最初の行のみが影響を受けました。 2 番目および 3 番目の行を Jenni は読み取ることができないため、それらの行が WHERE 節の条件を満たす場合であっても、それらはステートメントによる更新で選択されません。

LABEL 列が UPDATE ステートメントで明示的に設定されていなかったにもかかわらず、更新された行のその列の値が変更されたことに注目してください。その列には、Jenni が書き込み用に保持しているセキュリティー・ラベルが設定されます。

今度は Jenni は、どのセキュリティー・ラベルによって保護されるデータへの読み取りをも許可する LBAC 信用証明情報が付与されます。 Jenni の書き込み用 LBAC 信用証明情報は変更されません。依然として、Jenni は L0 および L1 により保護されるデータのみに書き込むことができます。

Jenni は再度次の SQL ステートメントを発行します。

```
UPDATE T1 SET DEPTNO = 44 WHERE DEPTNO = 11
```

今度は、2 番目と 3 番目の行のために更新は失敗します。 Jenni はそれらの行の読み取りができるため、それらはステートメントによる更新で選択されます。しかし、それらはセキュリティー・ラベル L2 および L3 により保護されているため、Jenni はそれらに書き込むことはできません。更新は行われず、エラーが戻されます。

Jenni はここで次の SQL ステートメントを発行します。

```
UPDATE T1
 SET DEPTNO = 55, LABEL = SECLABEL_BY_NAME( 'P1', 'L2' )
 WHERE LASTNAME = "Rjaibi"
```

ステートメント内の SECLABEL_BY_NAME 関数は、L2 という名前のセキュリティー・ラベルを戻します。Jenni は、最初の行を保護するセキュリティー・ラベルの明示的設定を試行します。 Jenni の LBAC 信用証明情報は、最初の行の読み取りを許可するため、その行は更新のために選択されます。 Jenni の LBAC 信用証明情報は、セキュリティー・ラベル L0 により保護されている行への書き込みを許可するため、Jenni はその行を更新することが許可されます。しかし、Jenni の LBAC 信用証明情報は、セキュリティー・ラベル L2 により保護されている行への書き込みを許可しないため、Jenni は列 LABEL にその値を設定することは許可されません。ステートメントは失敗し、エラーが戻されます。その行内の列は更新されません。

Jenni はここで次の SQL ステートメントを発行します。

```
UPDATE T1 SET LABEL = SECLABEL_BY_NAME( 'P1', 'L1' ) WHERE LASTNAME = "Rjaibi"
```

Jenni はセキュリティー・ラベル L1 により保護されている行への書き込みができるため、ステートメントは成功します。

T1 は次のようになります。

表 29. 表 T1

EMPNO	LASTNAME	DEPTNO	LABEL
1	Rjaibi	44	L1
2	Miller	11	L2
3	Bird	11	L3

保護された列を含む保護された行の更新

保護された行を持つ表内の保護された列の更新を試行する場合には、LBAC 信用証明情報は、更新による影響を受けるすべての保護された列の書き込みを許可していなければなりません。それ以外の場合、更新は失敗し、エラーが戻されます。これは、保護された列の更新に関する前のセクションで説明されているとおりです。更

新による影響を受けるすべての保護された列の更新が許可されている場合でも、LBAC 信用証明情報が読み取りおよび書き込みの両方を許可する行しか更新できません。これは、保護された行の更新に関する前のセクションで説明されているとおりです。DB2SECURITYLABEL のデータ・タイプを持つ列の処理は、保護された列が更新による影響を受けるかどうかに関わりなく同じです。

DB2SECURITYLABEL のデータ・タイプを持つ列がそれ自体保護された列である場合には、LBAC 信用証明情報は、その列への書き込みを許可しなければなりません。そうでない場合、その表のどの行も更新できません。

LBAC 保護データの削除またはドロップ

ユーザーの LBAC 信用証明情報が、ある行の読み取りを許可しない場合には、そのユーザーにとってはその行は存在していないかのようになるため、そのユーザーがそれを削除する方法はありません。読み取ることができる行を削除するには、LBAC 信用証明情報がその行への書き込みも許可していなければなりません。保護された列を持つ表の任意の行を削除するには、表内の保護されたすべての列への書き込みを許可する LBAC 信用証明情報を持っていなければなりません。

保護された行の削除

行の削除を試行する際、書き込み用の LBAC 信用証明情報はその行を保護するセキュリティー・ラベルと比較されます。保護セキュリティー・ラベルが LBAC 信用証明情報による書き込みアクセスをブロックする場合、DELETE ステートメントは失敗し、エラーが戻され、行は削除されません。

例

保護された表 T1 には以下の行があります。

LASTNAME	DEPTNO	LABEL
Rjaibi	55	L2
Miller	77	L1
Bird	55	L2
Fielding	77	L3

Pat は、アクセスが次の表に要約されるような LBAC 信用証明情報を持っていると想定します。

セキュリティー・ラベル	読み取りアクセス	書き込みアクセス
L1	はい	はい
L2	はい	いいえ
L3	いいえ	いいえ

Pat の LBAC 信用証明情報とセキュリティー・ラベルの厳密な詳細は、この例では重要ではありません。

Pat は次の SQL ステートメントを発行します。

```
SELECT * FROM T1 WHERE DEPTNO != 999
```

ステートメントが実行され、以下の結果セットを戻します。

LASTNAME	DEPTNO	LABEL
Rjaibi	55	L2
Miller	77	L1
Bird	55	L2

T1 の最後の行は結果に含まれません。なぜなら Pat はその行への読み取りアクセスがないからです。 Pat にとっては、その行は存在していないかのようになります。

Pat は次の SQL ステートメントを発行します。

```
DELETE FROM T1 WHERE DEPTNO != 999
```

Pat は、最初と 3 番目の行への書き込みアクセスがありません。その 2 つの行は L2 により保護されています。そのため、Pat は行の読み取りはできたとしても、それらの削除はできません。 DELETE ステートメントは失敗し、行は削除されません。

Pat は次の SQL ステートメントを発行します。

```
DELETE FROM T1 WHERE DEPTNO = 77;
```

Pat は、LASTNAME 列が Miller である行に書き込むことができるため、このステートメントは成功します。それは、このステートメントにより選択される唯一の行です。LASTNAME 列が Fielding である行は選択されません。なぜなら Pat の LBAC 信用証明情報はその行への読み取りを許可しないからです。その行は削除の対象としては決して考慮されないため、エラーは発生しません。

表の実際の行は、次のようになります。

LASTNAME	DEPTNO	LABEL
Rjaibi	55	L2
Bird	55	L2
Fielding	77	L3

保護された列を持つ行の削除

保護された列を持つ表の任意の行を削除するには、表内の保護されたすべての列への書き込みを許可する LBAC 信用証明情報を持っていなければなりません。LBAC 信用証明情報が書き込みを許可しない行が表にある場合には、削除は失敗し、エラーが戻されます。

表に保護された列および保護された行がある場合、特定の行を削除するには、表の保護されたすべての列への書き込みと、さらに削除したい行に対する読み取りおよび書き込みを許可する LBAC 信用証明情報を持っていなければなりません。

例

保護された表 T1 では、列 DEPTNO はセキュリティー・ラベル L2 により保護されています。T1 には以下の行が含まれます。

LASTNAME	DEPTNO *L2 により保護される*	LABEL
Rjaibi	55	L2
Miller	77	L1
Bird	55	L2
Fielding	77	L3

ユーザー Benny は次の表で要約されるアクセスを許可する LBAC 信用証明情報を持っていると想定します。

セキュリティー・ラベル	読み取りアクセス	書き込みアクセス
L1	はい	はい
L2	はい	いいえ
L3	いいえ	いいえ

Benny の LBAC 信用証明情報とセキュリティー・ラベルの厳密な詳細は、この例では重要ではありません。

Benny は次の SQL ステートメントを発行します。

```
DELETE FROM T1 WHERE DEPTNO = 77
```

Benny は列 DEPTNO への書き込みアクセスがないため、このステートメントは失敗します。

ここで、Benny が次の表で要約されているアクセスを持つように、Benny の LBAC 信用証明情報は変更されます。

セキュリティー・ラベル	読み取りアクセス	書き込みアクセス
L1	はい	はい
L2	はい	はい
L3	はい	いいえ

Benny は次の SQL ステートメントを再度発行します。

```
DELETE FROM T1 WHERE DEPTNO = 77
```

今回、Benny には、列 DEPTNO への書き込みアクセスがあるため、削除は継続します。 DELETE ステートメントは、LASTNAME 列に Miller の値がある行のみを選択します。 LASTNAME 列に Fielding の値がある行は選択されません。なぜなら Benny の LBAC 信用証明情報はその行の読み取りを許可しないからです。ステートメントによる削除でその行が選択されていないため、Benny がその行に書き込むことができなくても関係ありません。

選択された 1 行はセキュリティー・ラベル L1 により保護されます。Benny の LBAC 信用証明情報は、L1 により保護されるデータへの書き込みを許可するため、削除は成功します。

表 T1 の実際の行は、次のようになります。

LASTNAME	**DEPTNO** **L2** *により保護される*	**LABEL**
Rjaibi	55	L2
Bird	55	L2
Fielding	77	L3

保護データのドロップ

LBAC 信用証明情報が、セキュリティー・ラベルにより保護される列への書き込みを許可していなければ、その列をドロップできません。

DB2SECURITYLABEL のデータ・タイプを持つ列は、表からドロップできません。これを除去するには、最初に表からセキュリティー・ポリシーをドロップする必要があります。セキュリティー・ポリシーをドロップするとき、表は LBAC で保護されなくなり、列のデータ・タイプは自動的に DB2SECURITYLABEL から VARCHAR(128) FOR BIT DATA に変更されます。その後、列をドロップできます。

LBAC 信用証明情報は、保護データが含まれる表全体またはデータベース全体のドロップを妨げることはありません。ある表またはデータベースをドロップするための正常な権限を持っている場合には、データベースに保護データが含まれている場合であっても、ドロップするために LBAC 信用証明情報は必要ありません。

データからの LBAC 保護の除去

表からセキュリティー・ポリシーを除去するには、SECADM 権限がなければなりません。表からセキュリティー・ポリシーを除去するには、ALTER TABLE ステートメントの DROP SECURITY POLICY 節を使用します。これにより、表のすべての行と列の保護も自動的に除去されます。

行から保護を除去する

保護される行を持つ表では、すべての行がセキュリティー・ラベルによって保護されている必要があります。個別の行から LBAC 保護を除去することはできません。

表からセキュリティー・ポリシーを除去するという例外を除いては、タイプ DB2SECURITYLABEL の列を変更したり除去することはできません。

列から保護を除去する

列の保護は、SQL ステートメント ALTER TABLE の DROP COLUMN SECURITY 節を使用して除去することができます。列から保護を除去するには、表を変更するのに必要な通常の特権および権限に加えて、その列に対する読み取り/書き込みを行

うための LBAC 信用証明情報を持っている必要があります。

第 5 章 セキュリティー情報のためのシステム・カタログの使用

それぞれのデータベースについての情報は、 (データベース作成時に作成される) システム・カタログという 1 組のビュー内に自動的に維持されます。このシステム・カタログには、表、列、索引、プログラム、特権、その他のオブジェクトが含まれています。

次のビューおよび表関数は、ユーザーが保持する特権、特権を認可するユーザーの ID、およびオブジェクト所有者に関する情報をリストします。

SYSCAT.DBAUTH
データベースの特権のリスト

SYSCAT.TABAUTH
表とビューの特権のリスト

SYSCAT.COLAUTH
列の特権のリスト

SYSCAT.PACKAGEAUTH
パッケージの特権のリスト

SYSCAT.INDEXAUTH
索引の特権のリスト

SYSCAT.SCHEMAAUTH
スキーマの特権のリスト

SYSCAT.PASSTHRUAUTH
サーバーの特権のリスト

SYSCAT.ROUTINEAUTH
ルーチン (関数、メソッド、およびストアード・プロシージャー) の特権のリスト

SYSCAT.SURROGATEAUTHIDS
別の許可 ID が代理を務めることのできる許可 ID をリストします。

システムによってユーザーに付与される特権の付与者は、SYSIBM になります。SYSADM、SYSMAINT SYSCTRL、および SYSMON はシステム・カタログにリストされません。

CREATE ステートメントと GRANT ステートメントを使うと、システム・カタログの中に特権が入れられます。 SYSADM および DBADM 権限を持つユーザーは、システム・カタログ・ビューに対する SELECT 特権の GRANT と取り消しを行うことができます。

付与された特権を持つ許可名の検索

PRIVILEGES および他の管理ビューを使用して、データベースに特権を付与された許可名に関する情報を検索することができます。

たとえば次の照会では、付与された明示特権および許可 ID に加えて、PRIVILEGES 管理ビューから他の情報を検索します。

```
SELECT AUTHID, PRIVILEGE, OBJECTNAME, OBJECTSCHEMA, OBJECTTYPE FROM SYSIBMADM.PRIVILEGES
```

次の照会は、AUTHORIZATIONIDS 管理ビューを使用して、特権または権限を付与されたすべての許可 ID を検索し、それらのタイプを示します。

```
SELECT AUTHID, AUTHIDTYPE FROM SYSIBMADM.AUTHORIZATIONIDS
```

SYSIBMADM.OBJECTOWNERS 管理ビューおよび SYSPROC.AUTH_LIST_GROUPS_FOR_AUTHID 表関数を使用してセキュリティーに関する情報を検索することもできます。

バージョン 9.1 より前には、すべての特権に関する情報が、いずれか 1 つのシステム・カタログ・ビューに含まれることはありませんでした。バージョン 9.1 より前のリリースでは、以下のステートメントは、特権を持つすべての許可名を検索します。

```
 SELECT DISTINCT GRANTEE, GRANTEETYPE, 'DATABASE' FROM SYSCAT.DBAUTH
UNION
SELECT DISTINCT GRANTEE, GRANTEETYPE, 'TABLE   ' FROM SYSCAT.TABAUTH
UNION
SELECT DISTINCT GRANTEE, GRANTEETYPE, 'PACKAGE ' FROM SYSCAT.PACKAGEAUTH
UNION
SELECT DISTINCT GRANTEE, GRANTEETYPE, 'INDEX   ' FROM SYSCAT.INDEXAUTH
UNION
SELECT DISTINCT GRANTEE, GRANTEETYPE, 'COLUMN  ' FROM SYSCAT.COLAUTH
UNION
SELECT DISTINCT GRANTEE, GRANTEETYPE, 'SCHEMA  ' FROM SYSCAT.SCHEMAAUTH
UNION
SELECT DISTINCT GRANTEE, GRANTEETYPE, 'SERVER  ' FROM SYSCAT.PASSTHRUAUTH
ORDER BY GRANTEE, GRANTEETYPE, 3
```

時折、このステートメントによって検索されたリストと、システムのセキュリティー機能で定義されているユーザー名とグループ名のリストとを比較してみる必要があります。これによって、有効でなくなった許可名を識別できます。

注: リモート・データベース・クライアントをサポートする場合、許可名をリモート・クライアントだけに定義し、データベースのサーバー・マシンには定義しないことも可能です。

DBADM 権限を持つすべての名前の検索

以下のステートメントは、DBADM 権限が直接付与されている、すべての許可名を検索します。

```
SELECT DISTINCT GRANTEE, GRANTEETYPE FROM SYSCAT.DBAUTH
   WHERE DBADMAUTH = 'Y'
```

注: この照会は、SYSADM 権限を持つことで DBADM 権限を暗黙的に獲得した許可名に関する情報を戻しません。

表へのアクセスを許可されている名前の検索

PRIVILEGES および他の管理ビューを使用して、データベースに特権を付与された許可名に関する情報を検索することができます。

以下のステートメントは、修飾子 JAMES を持つ表 EMPLOYEE にアクセスすることが直接許可されている、すべての許可名 (およびそのタイプ) を検索します。

```
SELECT DISTINCT AUTHID, AUTHIDTYPE FROM SYSIBMADM.PRIVILEGES
    WHERE OBJECTNAME = 'EMPLOYEE' AND OBJECTSCHEMA = 'JAMES'
```

バージョン 9.1 より前のリリースでは、次の照会は同じ情報を検索します。

```
    SELECT DISTINCT GRANTEETYPE, GRANTEE FROM SYSCAT.TABAUTH
       WHERE TABNAME = 'EMPLOYEE'
         AND TABSCHEMA = 'JAMES'
    UNION
    SELECT DISTINCT GRANTEETYPE, GRANTEE FROM SYSCAT.COLAUTH
       WHERE TABNAME = 'EMPLOYEE'
         AND TABSCHEMA = 'JAMES'
```

だれが修飾子 JAMES を持つ表 EMPLOYEE を更新できるかを調べるためには、以下のステートメントを出します。

```
     SELECT DISTINCT GRANTEETYPE, GRANTEE FROM SYSCAT.TABAUTH
       WHERE TABNAME = 'EMPLOYEE' AND TABSCHEMA = 'JAMES' AND
          (CONTROLAUTH  = 'Y' OR
           UPDATEAUTH IN ('G','Y'))
    UNION
    SELECT DISTINCT GRANTEETYPE, GRANTEE FROM SYSCAT.DBAUTH
       WHERE DBADMAUTH = 'Y'
    UNION
    SELECT DISTINCT GRANTEETYPE, GRANTEE FROM SYSCAT.COLAUTH
       WHERE TABNAME = 'EMPLOYEE' AND TABSCHEMA = 'JAMES' AND
       PRIVTYPE = 'U'
```

これは、DBADM 権限をもつ許可名があればすべて検索し、さらに CONTROL または UPDATE 特権が直接付与されている許可名も検索します。ただし、SYSADM 権限だけを保持しているユーザーの許可名は戻しません。

一部の許可名は、個別のユーザーだけでなく、グループである場合もあることに注意してください。

ユーザーに付与されたすべての特権の検索

ユーザーは、システム・カタログ・ビューについての照会を行うことにより、自ら持っている特権のリストと、他のユーザーに付与した特権のリストを作成できます。

PRIVILEGES および他の管理ビューを使用して、データベースに特権を付与された許可名に関する情報を検索することができます。たとえば、次の照会は現行セッションの許可 ID に付与されたすべての特権を検索します。

```
SELECT * FROM SYSIBMADM.PRIVILEGES
 WHERE AUTHID = SESSION_USER AND AUTHIDTYPE = 'U'
```

このステートメント内のキーワード SESSION_USER は、現行ユーザーの許可名の値と等しい特殊レジスターです。

バージョン 9.1 より前のリリースでは、次の例は類似の情報を検索します。たとえば、以下のステートメントは、個々の許可名 JAMES に直接付与されているデータベース特権のリストを検索します。

```
     SELECT * FROM SYSCAT.DBAUTH
      WHERE GRANTEE = 'JAMES' AND GRANTEETYPE = 'U'
```

表の特権のうちユーザー JAMES によって直接付与されたものを検索するには、次のようなステートメントを使います。

```
SELECT * FROM SYSCAT.TABAUTH
 WHERE GRANTOR  = 'JAMES'
```

以下のステートメントは、ユーザー JAMES によって直接付与された、個別の列特権のリストを検索します。

```
SELECT * FROM SYSCAT.COLAUTH
   WHERE GRANTOR  = 'JAMES'
```

システム・カタログ・ビューのセキュリティー

システム・カタログ・ビューはデータベース内のすべてのオブジェクトを記述するので、ユーザーが機密データを持つとき、それらのアクセスを制限する場合があります。

CREATE DATABASE ... RESTRICTIVE コマンドを使用して、特権が PUBLIC に自動的に付与されないデータベースを作成することができます。この場合、次の通常のデフォルト認可アクションはいずれも発生しません。

- CREATETAB
- BINDADD
- CONNECT
- IMPLSCHEMA
- スキーマ SQLJ 中のすべてのプロシージャーに関する GRANT 付きの EXECUTE
- スキーマ SYSPROC 中のすべての関数とプロシージャーに関する GRANT 付きの EXECUTE
- NULLID スキーマ内で作成されたすべてのパッケージに対する BIND
- NULLID スキーマ中に作成されたすべてのパッケージに関する EXECUTE
- スキーマ SQLJ に関する CREATEIN
- スキーマ NULLID に関する CREATEIN
- 表スペース USERSPACE1 に関する USE
- SYSIBM カタログ表への SELECT アクセス
- SYSCAT カタログ・ビューへの SELECT アクセス
- SYSIBMADM 管理ビューへの SELECT アクセス
- SYSSTAT カタログ・ビューへの SELECT アクセス
- SYSSTAT カタログ・ビューへの UPDATE アクセス

RESTRICTIVE オプションでデータベースを作成し、PUBLIC に付与された許可が限定されていることを確認する場合、次の照会を発行して PUBLIC がアクセスできるスキーマを検証することができます。

```
SELECT DISTINCT OBJECTSCHEMA FROM SYSIBMADM.PRIVILEGES WHERE AUTHID='PUBLIC'

OBJECTSCHEMA
------------
SYSFUN
SYSIBM
SYSPROC
```

PUBLIC が依然 SYSIBM に対して持っているアクセスを確認するには、次の照会を発行して SYSIBM に付与されている特権を検証することができます。 結果には、特定のプロシージャーと関数について EXECUTE のみが付与されていることが示されています。

```
SELECT * FROM SYSIBMADM.PRIVILEGES WHERE OBJECTSCHEMA = 'SYSIBM'

AUTHID       AUTHIDTYPE  PRIVILEGE   GRANTABLE  OBJECTNAME         OBJECTSCHEMA    OBJECTTYPE
---------... ----------  ----------  ---------  ---------------... ------------... ----------
PUBLIC       G           EXECUTE     N          SQL060207192129400 SYSPROC         FUNCTION
PUBLIC       G           EXECUTE     N          SQL060207192129700 SYSPROC         FUNCTION
PUBLIC       G           EXECUTE     N          SQL060207192129701 SYSPROC
...
PUBLIC       G           EXECUTE     Y          TABLES             SYSIBM          PROCEDURE
PUBLIC       G           EXECUTE     Y          TABLEPRIVILEGES    SYSIBM          PROCEDURE
PUBLIC       G           EXECUTE     Y          STATISTICS         SYSIBM          PROCEDURE
PUBLIC       G           EXECUTE     Y          SPECIALCOLUMNS     SYSIBM          PROCEDURE
PUBLIC       G           EXECUTE     Y          PROCEDURES         SYSIBM          PROCEDURE
PUBLIC       G           EXECUTE     Y          PROCEDURECOLS      SYSIBM          PROCEDURE
PUBLIC       G           EXECUTE     Y          PRIMARYKEYS        SYSIBM          PROCEDURE
PUBLIC       G           EXECUTE     Y          FOREIGNKEYS        SYSIBM          PROCEDURE
PUBLIC       G           EXECUTE     Y          COLUMNS            SYSIBM          PROCEDURE
PUBLIC       G           EXECUTE     Y          COLPRIVILEGES      SYSIBM          PROCEDURE
PUBLIC       G           EXECUTE     Y          UDTS               SYSIBM          PROCEDURE
PUBLIC       G           EXECUTE     Y          GETTYPEINFO        SYSIBM          PROCEDURE
PUBLIC       G           EXECUTE     Y          SQLCAMESSAGE       SYSIBM          PROCEDURE
PUBLIC       G           EXECUTE     Y          SQLCAMESSAGECCSID  SYSIBM          PROCEDURE
```

注: DB2 データベース・マネージャーのバージョン 9.1 からは、SYSIBMADM.PRIVILEGES 管理ビューが使用できるようになりました。

DB2 データベース・マネージャーのバージョン 9.1 より前のリリースでは、データベースの生成時に、システム・カタログ・ビューに対する SELECT 特権が PUBLIC に付与されます。多くの場合、これによってセキュリティー上の問題が生じることはありません。しかし、これらの表にはデータベース内のすべてのオブジェクトが含まれているため、非常に重要なデータの場合は適切でないことがあります。このような場合には、PUBLIC から SELECT 特権を取り消してから、必要に応じて、特定のユーザーに対して SELECT 特権を付与することを考慮してください。システム・カタログ・ビューについての SELECT 特権の付与と取り消しは、他のビューの場合と同じ方法で行いますが、そのためには SYSADM または DBADM 権限が必要です。

いずれのユーザーも他のユーザーがアクセスするオブジェクトを知ることができないようにする場合には、最低でも、次のカタログおよび管理ビューへのアクセスを制限することを検討すべきです。

- SYSCAT.COLAUTH
- SYSCAT.DBAUTH
- SYSCAT.INDEXAUTH
- SYSCAT.PACKAGEAUTH
- SYSCAT.PASSTHRUAUTH
- SYSCAT.ROUTINEAUTH
- SYSCAT.SCHEMAAUTH
- SYSCAT.SECURITYLABELACCESS
- SYSCAT.SECURITYPOLICYEXEMPTIONS

- SYSCAT.SEQUENCEAUTH
- SYSCAT.SURROGATEAUTHIDS
- SYSCAT.TABAUTH
- SYSCAT.TBSPACEAUTH
- SYSCAT.XSROBJECTAUTH
- SYSIBMADM.AUTHORIZATIONIDS
- SYSIBMADM.OBJECTOWNERS
- SYSIBMADM.PRIVILEGES

それによって、ユーザー特権についての情報がデータベースにアクセスできる人全員で利用できるような事態を避けることができます。

各列にどの統計が集められているかも調べてください。システム・カタログに記録される統計には、ご使用の環境では機密情報となりうるデータ値が含まれることがあります。この統計に機密データが含まれている場合には、 SYSCAT.COLUMNS および SYSCAT.COLDIST カタログ・ビューについての SELECT 特権を、PUBLIC から取り消すことができます。

システム・カタログ・ビューに対するアクセスを限定する場合は、それぞれの許可名が自分自身の特権にかかわる情報だけを検索できるようにするビューを定義することができます。

たとえば、次のビュー MYSELECTS には、ユーザーの許可名に SELECT 特権が直接付与されている表の所有者と名前が含まれます。

```
CREATE VIEW MYSELECTS AS
   SELECT TABSCHEMA, TABNAME FROM SYSCAT.TABAUTH
   WHERE GRANTEETYPE = 'U'
     AND GRANTEE = USER
     AND SELECTAUTH = 'Y'
```

このステートメントの中の USER というキーワードは、現行セッションの許可名の値と等しくなります。

以下のステートメントは、このビューをそれぞれの許可名から利用可能にするものです。

```
GRANT SELECT ON TABLE MYSELECTS TO PUBLIC
```

最後に、次の 2 つのステートメントの発行によってビューおよび基本表についての SELECT 特権を必ず取り消すようにしてください。

```
REVOKE SELECT ON TABLE SYSCAT.TABAUTH FROM PUBLIC

REVOKE SELECT ON TABLE SYSIBM.SYSTABAUTH FROM PUBLIC
```

セキュリティーについての考慮事項

セキュリティーを正常に管理するには、ユーザーがデータへのアクセスを取得できる間接的な方法を認識しておく必要があります。また、データベースの作成時に付与される特定のシステム表に対するデフォルトの特権も認識しておく必要があります。

間接的な方法によるデータへのアクセス権の取得

ユーザーが許可されていないデータへのアクセス権を取得するために使用できる間接的な方法を以下に示します。

- **カタログ・ビュー:** DB2 データベース・システムのカタログ・ビューは、データベース・オブジェクトに関するメタデータと統計を保管します。カタログ・ビューに対する SELECT アクセス権を持つユーザーは、自分には資格のないデータに関する一部の情報を取得できます。セキュリティーを向上させるには、資格あるユーザーのみがカタログ・ビューに対するアクセス権を持っていることを確認してください。

  注: DB2 Universal Database™ バージョン 8 以前では、カタログ・ビューに関する SELECT アクセス権はデフォルトで PUBLIC に付与されました。 DB2 バージョン 9.1 以降のデータベース・システムでは、ユーザーは、カタログ・ビューに対する SELECT アクセス権を PUBLIC に付与するか、それとも CREATE DATABASE コマンドで新規の RESTRICTIVE オプションを使用して付与しないかを選択できます。

- **Visual Explain:** Visual Explain は、特定の照会に関する照会オプティマイザーによって選択されたアクセス・プランを示します。Visual Explain の情報には、照会で参照される列に関する統計も含まれます。これらの統計は、表の内容に関する情報を開示する可能性があります。
- **Explain スナップショット:** Explain スナップショットとは、SQL または XQuery ステートメントの EXPLAIN 時に収集される、圧縮された情報のことです。Explain スナップショットは、EXPLAIN_STATEMENT 表にバイナリー・ラージ・オブジェクト (BLOB) として保管され、表データに関する情報を開示する可能性のある列統計が含まれます。セキュリティーを向上させるには、Explain 表に対するアクセス権を資格あるユーザーのみに付与する必要があります。
- **ログ・リーダー関数:** ログを読み取る関数の実行を許可されているユーザーは、ログ・レコードの形式を理解できれば、許可されていないデータに対するアクセス権を取得できます。以下の関数がログを読み取ります。

関数	関数の実行に必要な権限
db2ReadLog	SYSADM または DBADM
db2ReadLogNoConn	なし。

- **複製:** データを複製する際には、保護データであっても宛先に再作成されます。セキュリティーを向上させるには、宛先がソースの場所と少なくとも同じほど安全であることを確認してください。
- **例外表:** データを表にロードしている際に例外表を指定すると、例外表に対するアクセス権を持つユーザーは許可されていない情報を取得できます。セキュリティーを向上させるには、例外表に対するアクセス権を許可ユーザーのみに付与し、例外表を処理し終えたらできるだけ早くドロップしてください。
- **表スペースまたはデータベースのバックアップ:** バックアップ・コマンドを実行する権限を持つユーザーは、保護データを含むデータベースまたは表スペースのバックアップを取り、そのデータを他の場所にリストアできます。ユーザーが本来アクセス権を持てないデータをバックアップに組み込むことができます。

SYSADM、SYSCTRL、または SYSMAINT 権限を持つユーザーがバックアップ・コマンドを実行できます。

- **セッション許可の設定:** DB2 Universal Database バージョン 8 以前では、DBADM 権限を持つユーザーは SET SESSION AUTHORIZATION SQL ステートメントを使用してセッション許可 ID をデータベース・ユーザーに設定できました。 DB2 バージョン 9.1 以降のデータベース・システムでは、ユーザーがセッション許可 ID を設定できるようにするには、その前に GRANT SETSESSIONUSER ステートメントを使用して明示的に許可されていなければなりません。

  しかし、既存のバージョン 8 データベースを DB2 バージョン 9.1 以降のデータベース・システムに移行する際には、(例えば、SYSCAT.DBAUTH で付与されて) 既存の明示 DBADM 権限を持つユーザーは、引き続きセッション許可をデータベース・ユーザーに設定できます。それが許可されているのは、既存のアプリケーションが引き続き作動するようにするためです。セッション許可を設定できるということは、すべての保護データにアクセスできる可能性があるということになります。セキュリティーの制限を増すには、REVOKE SETSESSIONUSER SQL ステートメントを実行して、この設定をオーバーライドできます。
- **ステートメントおよびデッドロックのモニター:** WITH VALUES 節が指定されていると、DB2 データベース管理システムのデッドロック・モニター・アクティビティーの一部として、パラメーター・マーカーに関連した値がモニター出力に書き込まれます。モニター出力に対するアクセス権を持つユーザーは、許可されていない情報に対するアクセス権を取得できます。
- **トレース:** トレースに表データを含めることができます。この種のトレースに対するアクセス権を持つユーザーは、許可されていない情報に対するアクセス権を取得できます。
- **ダンプ・ファイル:** DB2 データベース製品は、特定の問題のデバッグに役立つように、sqllib¥db2dump ディレクトリーにメモリー・ダンプ・ファイルを生成することがあります。これらのメモリー・ダンプ・ファイルに表データが含まれることがあります。その場合、このファイルに対するアクセス権を持つユーザーは、許可されていない情報に対するアクセス権を取得できます。セキュリティーを向上させるには、sqllib¥db2dump ディレクトリーに対するアクセスを制限する必要があります。
- **db2dart:** db2dart ツールは、データベースを調べ、検出した設計上のエラーを報告します。このツールは表データにアクセスでき、DB2 はこのアクセスに関するアクセス制御を施行しません。db2dart ツールを実行する権限を持つユーザーや、db2dart 出力に対するアクセス権を持つユーザーは、許可されていない情報に対するアクセス権を取得できます。
- **REOPT バインド・オプション:** REOPT バインド・オプションを指定すると、再最適化可能な追加バインド SQL ステートメントごとの Explain スナップショット情報が実行時に Explain 表に入れられます。EXPLAIN は入力データ値も表示します。
- **db2cat:** db2cat ツールは、表のパックされた記述子のダンプに使用します。表のパックされた記述子には、表の内容に関する情報を開示する可能性のある統計が含まれます。db2cat ツールを実行するユーザーや、この出力に対するアクセス権を持つユーザーは、許可されていない情報に対するアクセス権を取得できます。

データベースの作成時に付与されるデフォルト特権

データベースの作成時に付与される特定のシステム表に対するデフォルトの特権を以下に示します。

1. SYSIBM.SYSDBAUTH
   - データベースの作成者には以下の特権が付与されます。
     - DBADM
     - CREATETAB
     - CREATEROLE
     - BINDADD
     - CONNECT
     - NOFENCE
     - IMPLSCHEMA
     - LOAD
     - EXTERNALROUTINE
     - QUIESCECONNECT
   - 特殊なグループである PUBLIC には以下の特権が付与されます。
     - CREATETAB
     - BINDADD
     - CONNECT
     - IMPLSCHEMA
2. SYSIBM.SYSTABAUTH
   - 特殊なグループである PUBLIC には以下の特権が付与されます。
     - すべての SYSCAT および SYSIBM 表に関する SELECT
     - すべての SYSSTAT 表に関する SELECT および UPDATE
3. SYSIBM.SYSROUTINEAUTH
   - 特殊なグループである PUBLIC には以下の特権が付与されます。
     - スキーマ中のすべてのプロシージャーに関する GRANT 付きの EXECUTE
     - スキーマ SYSFUN 中のすべての関数とプロシージャーに関する GRANT 付きの SQLJ EXECUTE
     - スキーマ SYSPROC 中のすべての関数とプロシージャーに関する GRANT 付きの EXECUTE
     - スキーマ SYSIBM 中のすべての表関数に関する EXECUTE
     - スキーマ SYSIBM 中の他のすべてのプロシージャーに関する EXECUTE
4. SYSIBM.SYSPACKAGEAUTH
   - データベースの作成者には以下の特権が付与されます。
     - NULLID スキーマ中に作成されたすべてのパッケージに関する CONTROL
     - NULLID スキーマ中に作成されたすべてのパッケージに関する GRANT 付きの BIND
     - NULLID スキーマ中に作成されたすべてのパッケージに関する GRANT 付きの EXECUTE

–

- 特殊なグループである PUBLIC には以下の特権が付与されます。
  - NULLID スキーマ内で作成されたすべてのパッケージに対する BIND
  - NULLID スキーマ中に作成されたすべてのパッケージに関する EXECUTE

5. SYSIBM.SCHEMAAUTH
   - 特殊なグループである PUBLIC には以下の特権が付与されます。
     - スキーマ SQLJ に関する CREATEIN
     - スキーマ NULLID に関する CREATE IN
6. SYSIBM.TBSPACEAUTH
   - 特殊なグループである PUBLIC には以下の特権が付与されます。
     - 表スペース USERSPACE1 に関する USE

第 6 章 ファイアウォール・サポート

ファイアウォール は、ネットワーク・ゲートウェイ・サーバーに位置し、システムまたはネットワークへの無許可アクセスを防ぐために使用される、関連したプログラムのセットです。

ファイアウォールには、以下の 4 つのタイプがあります。

1. ネットワーク・レベル、パケット・フィルター、またはスクリーニング・ルーター・ファイアウォール
2. 従来のアプリケーション・レベルのプロキシー・ファイアウォール
3. 回線レベル、または透過性のプロキシー・ファイアウォール
4. Stateful Multi-Layer Inspection (SMLI) ファイアウォール

上記のファイアウォールのタイプのいずれかに入る既存のファイアウォール製品もあります。他にも、上記のタイプの組み合わせとなる、多くのファイアウォール製品があります。

スクリーニング・ルーター・ファイアウォール

スクリーニング・ルーター・ファイアウォールは、ネットワーク・レベルまたはパケット・フィルター・ファイアウォールとも呼ばれます。このようなファイアウォールは、着信パケットをプロトコル属性によってスクリーニングすることにより動作します。スクリーニングされるプロトコル属性には、送信元または宛先アドレス、プロトコルのタイプ、送信元または宛先のポート、または他のプロトコルに特有の属性が含まれます。

すべてのファイアウォール・ソリューション (SOCKS を除く) では、 DB2 データベースによって使用されるすべてのポートを着信および発信パケットのために開けておく必要があります。 DB2 データベースは、DB2 データベース・ツールによって使用される DB2 Administration Server (DAS) 用として、ポート 523 を使用します。サービス・ファイルを使用して、サーバー・データベース・マネージャー構成ファイル内のサービス名をそのポート番号にマップし、すべてのサーバー・インスタンスが使用するポートを判別してください。

アプリケーション・プロキシー・ファイアウォール

プロキシーまたはプロキシー・サーバーは、 Web クライアントと Web サーバーの間の中継地点として動作する技術です。プロキシー型ファイアウォールは、クライアントからの要求に対するゲートウェイの役割を果たします。

クライアントの要求がファイアウォールで受信されると、最終のサーバー宛先アドレスがプロキシー・ソフトウェアによって判別されます。アプリケーション・プロキシーがアドレスを変換し、必要に応じてさらにアクセス・コントロール検査およびロギングを行い、クライアントに代わってサーバーに接続します。

ファイアウォール・マシン上の DB2 Connect 製品は、宛先サーバーに対するプロキシーの役割を果たすことができます。また、最終的な宛先サーバーに対するホップ・サーバーとして動作する、ファイアウォール上の DB2 データベース・サーバーは、アプリケーション・プロキシーに似た役割を果たします。

回線レベルのファイアウォール

回線レベルのファイアウォールは、透過性プロキシー・ファイアウォールとも呼ばれます。

透過性プロキシー・ファイアウォールは、プロキシー認証および識別に必要とされる以上には、要求または応答を変更しません。透過性プロキシー・ファイアウォールとしては、 SOCKS があります。

DB2 データベース・システムは SOCKS バージョン 4 をサポートします。

Stateful Multi-Layer Inspection (SMLI) ファイアウォール

Stateful Multi-Layer Inspection (SMLI) ファイアウォールは、オープン・システム間相互接続 (OSI) モデルの 7 層すべてを調べる、パケット・フィルター操作の高性能な形式を使用します。

各パケットが調べられ、類似したパケットの既知の状態と比較されます。スクリーニング・ルーター・ファイアウォールがパケット・ヘッダーのみを調べるのに対し、 SMLI ファイアウォールはデータも含むパケット全体を調べます。

第 7 章 セキュリティー・プラグイン

DB2 データベース・システムでの認証は、セキュリティー・プラグイン を使用して行われます。セキュリティー・プラグインは、動的にロード可能なライブラリーであり、認証セキュリティー・サービスを提供します。

DB2 データベース・システムは、以下のタイプのプラグインを提供します。

- グループ検索プラグイン: 特定のユーザーのグループ・メンバーシップ情報を検索します。
- クライアント認証プラグイン: DB2 クライアント上で認証を管理します。
- サーバー認証プラグイン: DB2 サーバー上で認証を管理します。

DB2 は、次の 2 つのプラグイン認証メカニズムをサポートしています。

ユーザー ID/パスワード認証

これには、ユーザー ID とパスワードを使用する認証が関係します。以下の認証タイプは、ユーザー ID/パスワード認証プラグインを使用してインプリメントされます。

- CLIENT
- SERVER
- SERVER_ENCRYPT
- DATA_ENCRYPT
- DATA_ENCRYPT_CMP

上記のような認証タイプによって、ユーザー認証がどこでどのように行われるかが決まります。使用される認証タイプは、*authentication* データベース・マネージャー構成パラメーターで指定した認証タイプによって異なります。 SRVCON_AUTH パラメーターを指定した場合、接続またはアタッチの操作の処理時には、このパラメーターのほうが AUTHENTICATION よりも優先されます。

GSS-API 認証

GSS-API の正式名称は、*Generic Security Service Application Program Interface Version 2* (IETF RFC2743) および *Generic Security Service API Version 2: C-Bindings* (IETF RFC2744) です。 Kerberos 認証も、GSS-API を使用してインプリメントされます。以下の認証タイプは、GSS-API 認証プラグインを使用してインプリメントされます。

- KERBEROS
- GSSPLUGIN
- KRB_SERVER_ENCRYPT
- GSS_SERVER_ENCRYPT

KRB_SERVER_ENCRYPT および GSS_SERVER_ENCRYPT は、GSS-API 認証とユーザー ID/パスワード認証の両方をサポートしますが、GSS-API 認証のほうが望ましい認証タイプです。

注: 認証タイプによって、ユーザーがどこでどのように認証されるかが決まります。特定の認証タイプを使用するには、認証データベース・マネージャーの構成パラメーターを更新します。

各プラグインは独立して使用するか、1 つ以上の他のプラグインと併せて使用することができます。例えば、サーバー認証プラグインだけを使用して、クライアントおよびグループの認証に対しては DB2 のデフォルトをとることができます。または、グループまたはクライアントの認証プラグインのみを使用することもできます。クライアントとサーバーの両方のプラグインが必要となるのは、GSS-API 認証プラグインの場合のみです。

デフォルトの動作として、オペレーティング・システム・レベルの認証メカニズムをインプリメントするユーザー ID/パスワード・プラグインが使用されます。これより前のリリースでは、デフォルト動作として、プラグインのインプリメンテーションなしで、オペレーティング・システム・レベルの認証が直接使用されます。クライアント・サイドの Kerberos サポートは、Solaris、AIX、Windows、および Linux オペレーティング・システムで使用できます。 Windows プラットフォームでは、デフォルトで Kerberos サポートが使用可能になっています。

DB2 データベース・システムには、グループ検索用、ユーザー ID/パスワード認証用、および Kerberos 認証用のプラグインのセットが組み込まれています。セキュリティー・プラグイン・アーキテクチャーを用いる場合は、独自のプラグインを作成するか、またはサード・パーティーからプラグインを購入することによって、DB2 のクライアントおよびサーバーの認証動作をカスタマイズすることができます。

DB2 クライアント上のセキュリティー・プラグインのデプロイメント

DB2 クライアントは 1 つのグループ・プラグイン、1 つのユーザー ID/パスワード認証プラグインをサポートでき、特定の GSS-API プラグインについて DB2 サーバーと折衝します。この折衝では、クライアントが DB2 サーバー側のインプリメントされている GSS-API プラグインのリストをスキャンして、クライアントにインプリメントされている認証プラグインと一致する認証プラグイン名を探します。サーバー側のプラグインのリストは、サーバー上にインプリメントされているすべてのプラグインの名前を収めている *srvcon_gssplugin_list* データベース・マネージャー構成パラメーター値に指定されます。以下の図は、DB2 クライアント上のセキュリティー・プラグイン・インフラストラクチャーを描写しています。

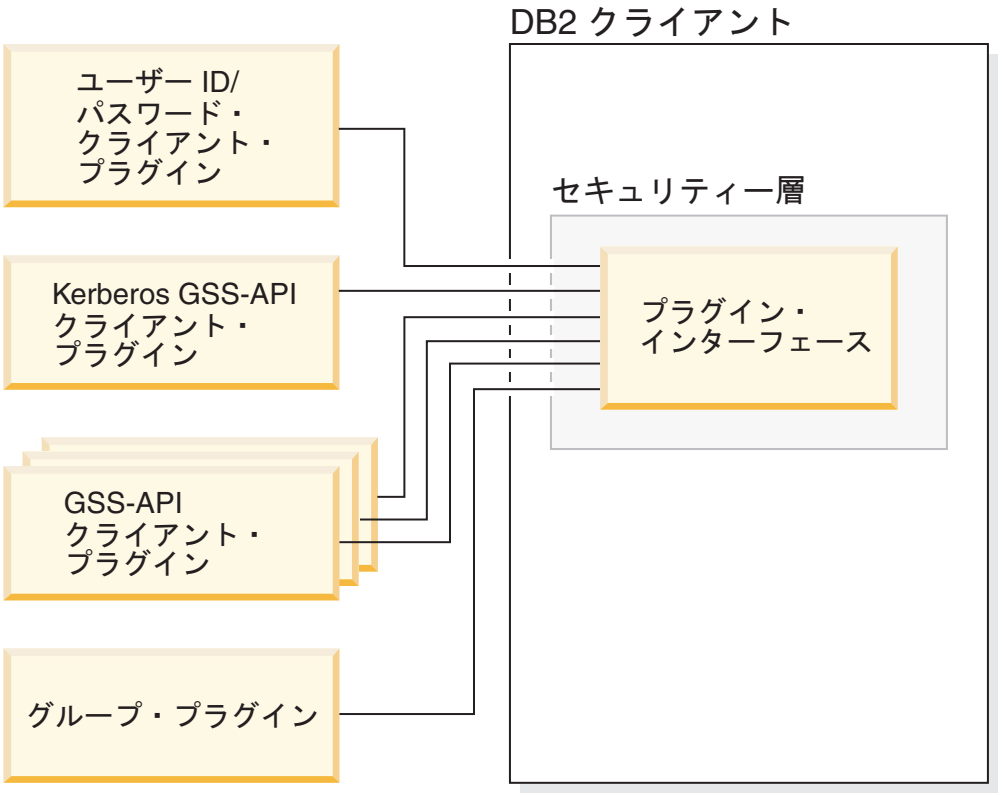


図 3. DB2 クライアント上のセキュリティー・プラグインのデプロイメント

DB2 サーバー上のセキュリティー・プラグインのデプロイメント

DB2 サーバーは 1 つのグループ・プラグイン、1 つのユーザー ID/パスワード認証プラグイン、および複数の GSS-API プラグインをサポートできます。複数の GSS-API プラグインは、*srvcon_gssplugin_list* データベース・マネージャー構成パラメーター値内にリストとして指定されます。このリストの中の 1 つの GSS-API プラグインのみ、Kerberos プラグインにすることができます。

サーバー・サイドのセキュリティー・プラグインに加えて、データベース・サーバー上にもクライアント許可プラグインをデプロイする必要があるかもしれません。db2start および db2trc などのインスタンス・レベルの操作の実行時には、DB2 データベース・マネージャーはクライアント認証プラグインを使用して、その操作に対する許可検査を実行します。したがって、*authentication* データベース・マネージャー構成パラメーターで指定したサーバー・プラグインに対応するクライアント認証プラグインをインストールしておく必要があります。 *authentication* と *srvcon_auth* には主な違いがあります。特に、これらを別々の異なる値に設定することで、データベース接続の認証用に一方のメカニズムを使用し、ローカル許可用にもう一方のメカニズムを使用することができます。最も一般的な使用法としては、*srvcon_auth* を GSSPLUGIN として設定し、*authentication* を SERVER として設定します。データベース・サーバー上でクライアント認証プラグインを使用しない場合、db2start などのインスタンス・レベルの操作は失敗します。例えば、認証タイプが SERVER で、ユーザー指定のクライアント・プラグインが使用されていない場合、DB2 データベース・システムは、IBM 提供のデフォルト・クライアント・オペレーティング・システム・プラグインを使用します。以下の図は、DB2 サーバー上のセキュリ

ティー・プラグイン・インフラストラクチャーを描写しています。

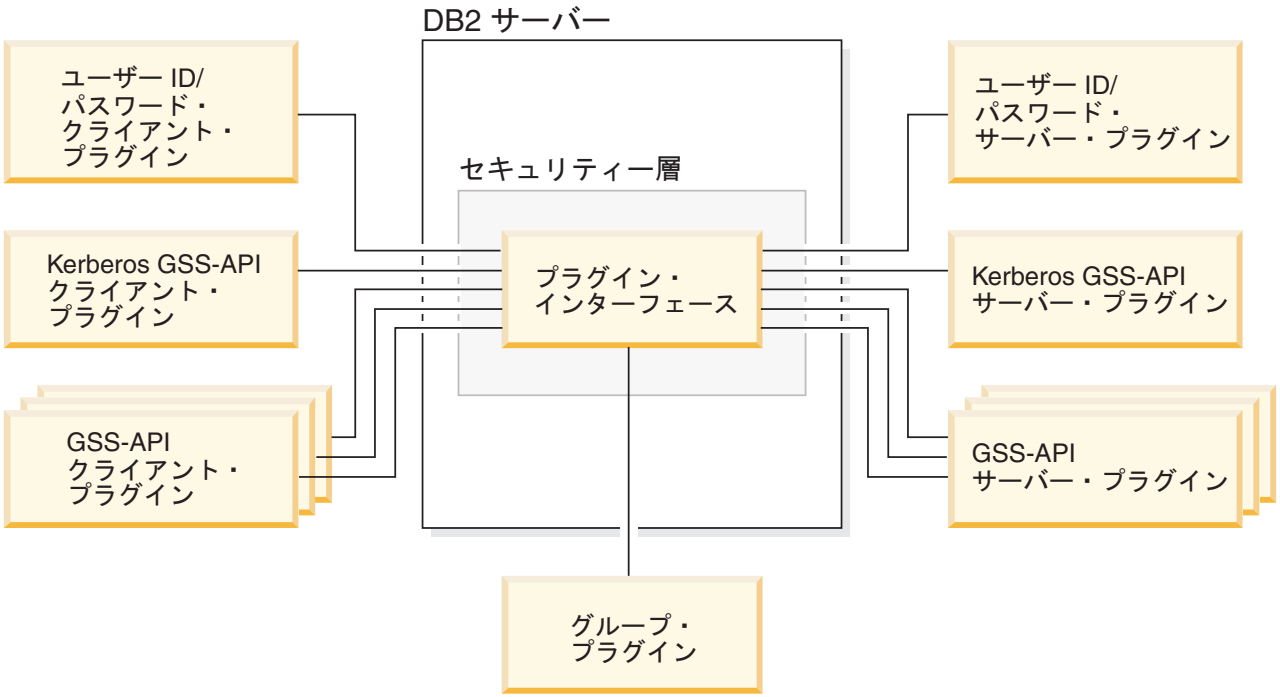


図 4. DB2 サーバー上のセキュリティー・プラグインのデプロイメント

注: セキュリティー・プラグインのデプロイメントのコーディング、確認、およびテストが十分に行われないと、インストールされている DB2 データベース・システムの保全性が損なわれることがあります。 DB2 データベース・システムでは、多数の一般的なタイプの障害に対する予防措置がとられていますが、ユーザー作成のセキュリティー・プラグインのデプロイ時には、完全な保全性が確保されるとは限りません。

セキュリティー・プラグインの有効化

システム管理者は、プラグインに関係した特定のデータベース・マネージャー構成パラメーターを更新することにより、各認証メカニズムに使用するプラグインの名前を指定できます。これらのパラメーターがヌルの場合、DB2 提供のグループ検索、ユーザー ID/パスワード管理、または Kerberos (サーバー上で、認証に Kerberos が設定されている場合) 用のプラグインがデフォルトになります。 DB2 ではデフォルト GSS-API プラグインは用意されていません。したがって、システム管理者は、*authentication* パラメーターに認証タイプ GSSPLUGIN を指定する場合は、*srvcon_gssplugin_list* に GSS-API 認証プラグインも指定しなければなりません。

DB2 によるセキュリティー・プラグインのロード方法

データベース・マネージャーの始動時に、データベース・マネージャー構成パラメーターで識別されたサポートされるすべてのプラグインがロードされます。

接続またはアタッチの操作中にサーバーとの折衝済みのセキュリティー・メカニズムに適したプラグインが、DB2 クライアントによってロードされます。クライアン

ト・アプリケーションが原因で、複数のセキュリティー・プラグインが並行してロードされて使用される可能性もあります。このような事態が発生するのは、例えば、さまざまなインスタンスからのそれぞれ異なるデータベースへの同時接続をもったスレッド化されたプログラムにおいてです。

接続またはアタッチの操作以外のアクションでは、許可も必要になります (データベース・マネージャー構成の更新、データベース・マネージャーの開始と停止、DB2 トレースのオン/オフなど)。そのようなアクションの場合、DB2 クライアント・プログラムは、別のデータベース・マネージャー構成パラメーターに指定されているプラグインをロードします。 *authentication* が GSSPLUGIN に設定されている場合、DB2 データベース・マネージャーは、*local_gssplugin* で指定されたプラグインを使用します。 *authentication* が KERBEROS に設定されている場合、DB2 データベース・マネージャーは、*clnt_krb_plugin* で指定されたプラグインを使用します。その他の場合は、DB2 データベース・マネージャーは *clnt_pw_plugin* で指定されたプラグインを使用します。

セキュリティー・プラグイン API は、IPv4 プラットフォームまたは IPv6 プラットフォームから呼び出すことができます。 IPv4 アドレスは、a.b.c.d という可読形式の 32 ビット・アドレスです。a から d はそれぞれ、0 から 255 までの 10 進数を表します。 IPv6 アドレスは、a:b:c:d:e:f:g:h の形式の 128 ビット・アドレスです。a から h はそれぞれ、4 桁の 16 進数字を表します。

セキュリティー・プラグインの作成

セキュリティー・プラグインを作成する場合は、DB2 データベース・マネージャーが使用する標準認証機能をインプリメントする必要があります。自分独自のカスタマイズしたセキュリティー・プラグインを使用する場合、CLP または動的 SQL ステートメントを介して発行する接続ステートメント上で、最大 255 文字までのユーザー ID を使用することができます。使用できるプラグイン・タイプに関してインプリメントする必要がある機能は、以下のとおりです。

グループ検索
ユーザーが所属するグループのリストを取得します。

ユーザー ID/パスワード認証
- デフォルトのセキュリティー・コンテキストを識別します (クライアントのみ)。
- パスワードの検証と、任意選択でパスワードの変更を行います。
- 指定されたストリングが有効なユーザーを表すかどうかを判別します (サーバーのみ)。
- クライアント上で提示されたユーザー ID またはパスワードを、サーバーへの送信前に修正します (クライアントのみ)。
- 特定のユーザーに関連した DB2 許可 ID を戻します。

GSS-API 認証
- 必要な GSS-API 関数をインプリメントします。
- デフォルトのセキュリティー・コンテキストを識別します (クライアントのみ)。

- ユーザー ID とパスワードに基づいて初期信用証明情報を生成し、任意選択でパスワードを変更します (クライアントのみ)。
- セキュリティー・チケットの作成と受け入れを行います。
- 特定の GSS-API セキュリティー・コンテキストに関連した DB2 許可 ID を戻します。

セキュリティー・プラグイン・ライブラリーの位置

セキュリティー・プラグインを (自分で作成するか、またはサード・パーティーから購入して) 取得したら、データベース・サーバー上の特定の場所にコピーします。

DB2 クライアントは、クライアント・サイドのユーザー認証プラグインを次のディレクトリーで探します。

- UNIX 32 ビット: `$DB2PATH/security32/plugin/client`
- UNIX 64 ビット: `$DB2PATH/security64/plugin/client`
- WINDOWS 32 ビットおよび 64 ビット: `$DB2PATH¥security¥plugin¥`*instance name*`¥client`

注: Windows ベースのプラットフォームの場合、サブディレクトリーの *instance name* および *client* は自動的には作成されません。これらは、インスタンスの所有者が手動で作成しなければなりません。

DB2 データベース・マネージャーは、サーバー・サイドのユーザー認証プラグインを次のディレクトリーで探します。

- UNIX 32 ビット: `$DB2PATH/security32/plugin/server`
- UNIX 64 ビット: `$DB2PATH/security64/plugin/server`
- WINDOWS 32 ビットおよび 64 ビット: `$DB2PATH¥security¥plugin¥`*instance name*`¥server`

注: Windows ベースのプラットフォームの場合、サブディレクトリーの *instance name* および *server* は自動的には作成されません。これらは、インスタンスの所有者が手動で作成しなければなりません。

DB2 データベース・マネージャーは、グループ・プラグインを次のディレクトリーで探します。

- UNIX 32 ビット: `$DB2PATH/security32/plugin/group`
- UNIX 64 ビット: `$DB2PATH/security64/plugin/group`
- WINDOWS 32 ビットおよび 64 ビット: `$DB2PATH¥security¥plugin¥`*instance name*`¥group`

注: Windows ベースのプラットフォームの場合、サブディレクトリーの *instance name* および *group* は自動的には作成されません。これらは、インスタンスの所有者が手動で作成しなければなりません。

セキュリティー・プラグインの命名規則

セキュリティー・プラグインのライブラリーには、プラットフォーム固有のファイル名拡張子が付いていなければなりません。 C または C++ で書かれたセキュリティー・プラグイン・ライブラリーには、プラットフォーム固有のファイル名拡張子が付いていなければなりません。

- Windows: .dll
- AIX: .a または .so。両方の拡張子が存在する場合、.a 拡張子が使用されます。
- Linux、 HP IPF、および Solaris: .so
- PA-RISC 上の HPUX: .sl または .so。両方の拡張子が存在する場合、.sl 拡張子が使用されます。

注: また、ユーザーは、DB2 Universal JDBC ドライバーをもったセキュリティー・プラグインを開発することもできます。
例えば、MyPlugin というセキュリティー・プラグイン・ライブラリーがあるとします。サポートされている各オペレーティング・システムに対する適切なライブラリー・ファイル名は、次のとおりです。

- Windows 32 ビット: MyPlugin.dll
- Windows 64 ビット: MyPlugin64.dll
- AIX 32 または 64 ビット: MyPlugin.a またはMyPlugin.so
- SUN 32 または 64-bit、Linux 32 または 64 ビット、IPF 上の HP 32 または 64 ビット: MyPlugin.so
- PA-RISC 上の HP-UX 32 または 64 ビット: MyPlugin.sl または MyPlugin.so

注: 接尾部 ″64″ は、64 ビットWindows のセキュリティー・プラグインのライブラリー名にのみ必要です。

セキュリティー・プラグインの名前を使ってデータベース・マネージャー構成を更新する際は、接尾部 ″64″ のないライブラリーの絶対パス名を使用し、ファイル拡張子と名前の修飾パス部分を省略してください。オペレーティング・システムが何であっても、MyPlugin というセキュリティー・プラグイン・ライブラリーは、次のように登録されます。

```
UPDATE DBM CFG USING CLNT_PW_PLUGIN MyPlugin
```

セキュリティー・プラグイン名は、大文字小文字の区別があり、ライブラリー名と完全に一致している必要があります。 DB2 データベース・システムは、関連するデータベース・マネージャー構成パラメーターの値を使用してライブラリー・パスを組み立て、そのライブラリー・パスを使用してセキュリティー・プラグイン・ライブラリーをロードします。

セキュリティー・プラグイン名の競合を防ぐために、使用する認証方式、およびそのプラグインを作成した会社の識別シンボルを使って、プラグインに名前を付けてください。例えば、Foo, Inc. という会社が FOOsomemethod という認証方式をインプリメントするプラグインを作成した場合、そのプラグインには FOOsomemethod.dll といった名前を付けることができます。

プラグイン名の最大長 (ファイル拡張子および接尾部の 64 を含まない) は、 32 バイトまでに制限されています。データベース・サーバーによってサポートされるプ

ラグインの最大数はありませんが、データベース・マネージャー構成内のコンマで区切られたプラグインのリストの最大長は 255 バイトです。次のように、この 2 つの制限を識別する 2 つの定義が、インクルード・ファイル sqlenv.h にあります。

```
#define SQL_PLUGIN_NAME_SZ     32     /* plug-in name */
#define SQL_SRVCON_GSSPLUGIN_LIST_SZ 255 /* GSS API plug-in list */
```

セキュリティー・プラグイン・ライブラリー・ファイルには、以下のファイル許可がなければなりません。

- インスタンス所有者によって所有される。
- システム上のすべてのユーザーが読み取れる。
- システム上のすべてのユーザーが実行できる。

セキュリティー・プラグインの 2 部構成ユーザー ID のサポート

Windows 上の DB2 データベース・マネージャーは、2 部構成ユーザー ID の使用と、2 部構成ユーザー ID から 2 部構成許可 ID へのマッピングをサポートしています。

例えば、ドメインとユーザー ID からなる Windows オペレーティング・システムの 2 部構成ユーザー ID (例えば MEDWAY¥pieter) について考えます。この例では、MEDWAY はドメインであり、pieter はユーザー名です。 DB2 データベース・システムでは、この 2 部構成ユーザー ID を、1 部構成許可 ID と 2 部構成許可 ID のどちらにマップするかを指定することができます。

2 部構成ユーザー ID から 2 部構成許可 ID へのマッピングもサポートされていますが、デフォルト動作ではありません。デフォルトでは、 1 部構成ユーザー ID と 2 部構成ユーザー ID はどちらも 1 部構成許可 ID にマップされます。 2 部構成ユーザー ID から 2 部構成許可 ID へのマッピングもサポートされていますが、デフォルト動作ではありません。

2 部構成ユーザー ID から 1 部構成ユーザー ID へのデフォルト・マッピングを使用すれば、ユーザーは次のようにしてデータベースに接続することができます。

```
db2 connect to db user MEDWAY¥pieter using pw
```

この状況下でデフォルト動作を使用すると、ユーザー ID MEDWAY¥pieter は許可 ID PIETER に解決されます。 2 部構成ユーザー ID から 2 部構成許可 ID へのマッピングのサポートが有効になっている場合、許可 ID は MEDWAY¥PIETER となります。

2 部構成ユーザー ID から 2 部構成許可 ID へのマップをDB2 で有効にするために、以下の 2 セットの認証プラグインが DB2 に用意されています。

- 一方のセットは、1 部構成ユーザー ID を 1 部構成許可 ID にマップし、2 部構成ユーザー ID を 1 部構成許可 ID にマップするだけです。
- もう一方のセットは、1 部構成ユーザー ID または 2 部構成ユーザー ID の両方を、2 部構成許可 ID にマップします。

作業環境におけるユーザー名を、さまざまな場所で定義された複数のアカウント (ローカル・アカウント、ドメイン・アカウント、およびトラステッド・ドメイン・

アカウントなど) にマップできる場合は、2 部構成許可 ID のマッピングを有効にするプラグインを指定することができます。

PIETER などの 1 部構成許可 ID と、ドメインとユーザー ID を結合した MEDWAY¥PIETERのような 2 部構成許可 ID は、機能的に個別の許可 ID であることに注意してください。このような許可 ID の一方に関連付けられている特権セットは、他方の許可 ID に関連付けられている特権セットとは完全に異なります。 1 部構成許可 ID と 2 部構成許可 ID を扱う際には注意が必要です。

次の表は、DB2 データベース・システムに用意されているプラグインの種類と、特定の認証インプリメンテーション用のプラグイン名を示しています。

表 30. DB2 セキュリティー・プラグイン

認証タイプ	1 部構成のユーザー ID のプラグインの名前	2 部構成のユーザー ID のプラグインの名前
ユーザー ID/パスワード (クライアント)	IBMOSauthclient	IBMOSauthclientTwoPart
ユーザー ID/パスワード (サーバー)	IBMOSauthserver	IBMOSauthserverTwoPart
Kerberos	IBMkrb5	IBMkrb5TwoPart

注: Windows 64 ビット・プラットフォームでは、ここにリストされているプラグイン名に ″64″ という文字が付加されます。

ユーザー ID/パスワード・プラグインまたは Kerberos プラグインを必要とする認証タイプを指定している場合は、上記の表の「1 部構成ユーザー ID のプラグインの名前」列にリストされているプラグインがデフォルトで使用されます。

2 部構成ユーザー ID を 2 部構成許可 ID にマップするには、2 部構成プラグイン (これはデフォルトのプラグインではありません) の使用を指定する必要があります。セキュリティー・プラグインは、セキュリティー関連のデータベース・マネージャー構成パラメーターを設定することによって、インスタンス・レベルで指定します。それは次のようにします。

2 部構成ユーザー ID を 2 部構成許可 ID にマップするサーバー認証の場合は、以下のように設定する必要があります。

- srvcon_pw_plugin を IBMOSauthserverTwoPart に設定
- clnt_pw_plugin を IBMOSauthclientTwoPart に設定

2 部構成ユーザー ID を 2 部構成許可 ID にマップするクライアント認証の場合は、以下のように設定する必要があります。

- srvcon_pw_plugin を IBMOSauthserverTwoPart に設定
- clnt_pw_plugin を IBMOSauthclientTwoPart に設定

2 部構成ユーザー ID を 2 部構成許可 ID にマップする Kerberos 認証の場合は、以下のように設定する必要があります。

- srvcon_gssplugin_list を IBMOSkrb5TwoPart に設定
- clnt_krb_plugin を IBMkrb5TwoPart に設定

セキュリティー・プラグイン・ライブラリーは、Microsoft Windows Security Account Manager 互換形式で指定された 2 パーツのユーザー ID を受け入れます。これは、例えば *domain¥user ID* の形式です。接続時の DB2 の認証プロセスおよび許可プロセスでは、ドメインとユーザー ID の両方の情報が使用されます。

新規データベースを作成する際は、既存データベース内の 1 部構成許可 ID と競合しないよう、2 部構成プラグインをインプリメントすることを検討するようお勧めします。 2 部構成許可 ID を使用する新規のデータベースは、1 部構成許可 ID を使用するデータベースとは別のインスタンス内で作成する必要があります。

セキュリティー・プラグイン API のバージョン管理

DB2 データベース・システムは、セキュリティー・プラグイン API のバージョン番号をサポートします。そのようなバージョン番号は、DB2 UDB、バージョン 8.2 では、1 で始まる整数になります。

DB2 がセキュリティー・プラグイン API に渡すバージョン番号は、DB2 がサポートできる最高のバージョン番号であり、構造のバージョン番号に対応します。プラグインは、それより高い API バージョンをサポートできる場合、DB2 が要求したバージョン用の関数ポインターを戻さなければなりません。プラグインがそれより低いバージョンの API しかサポートしない場合、プラグインはその低いバージョン用の関数ポインターを入れる必要があります。いずれのケースでも、セキュリティー・プラグイン API は、サポートしている API のバージョン番号を、関数構造のバージョン・フィールドに入れて戻す必要があります。

DB2 の場合、セキュリティー・プラグインのバージョン番号は、必要な場合にのみ変化します (例えば、API のパラメーターが変更された場合など)。バージョン番号が DB2 のリリース番号とともに自動的に変わるわけではありません。

セキュリティー・プラグインの 32 ビットと 64 ビットに関する考慮事項

通常、32 ビット DB2 インスタンスは、32 ビット・セキュリティー・プラグインを使用し、64 ビット DB2 インスタンスは 64 ビット・セキュリティー・プラグインを使用します。しかし、64 ビット・インスタンス上では、DB2 は、32 ビット・プラグイン・ライブラリーを必要とする 32 ビット・アプリケーションをサポートします。

32 ビットと 64 ビットの両方のアプリケーションが実行できるデータベース・インスタンスは、ハイブリッド・インスタンスといいます。ハイブリッド・インスタンスがあり、32 ビット・アプリケーションを実行しようとしている場合は、必要な 32 ビット・セキュリティー・プラグインが、 32 ビット・プラグイン・ディレクトリー内に用意されていることを確認してください。 Linux および UNIX オペレーティング・システム (Linux on IPF を除く) 上の 64 ビット DB2 インスタンスでは、`security32`と `security64` のディレクトリーが現れます。 X64 または IPF 上での Windows における 64 ビット DB2 インスタンスの場合、32 ビットと 64 ビットの両方のセキュリティー・プラグインが同一のディレクトリー内にありますが、64 ビット・プラグイン名には、接尾部の 64 が付いています。

32 ビット・インスタンスを 64 ビット・インスタンスに移行する予定の場合は、64 ビット用に再コンパイルされたセキュリティー・プラグイン用のバージョンを取得する必要があります。

64 ビット・プラグイン・ライブラリーを提供しないベンダーからセキュリティー・プラグインを購入した場合、32 ビット・アプリケーションを実行する 64 ビット・スタブをインプリメントできます。この場合、セキュリティー・プラグインは、ライブラリーではなく外部プログラムになります。

セキュリティー・プラグインの問題判別

セキュリティー・プラグインの問題は、SQL エラー経由と管理通知ログ経由の 2 とおりの方法で報告されます。

以下は、セキュリティー・プラグインに関係した SQLCODE 値です。

- SQLCODE -1365 は、db2start または db2stop の間にプラグイン・エラーが発生すると戻されます。
- SQLCODE -1366 は、ローカル許可の問題がある場合に戻されます。
- SQLCODE -30082 は、すべての接続に関係したプラグイン・エラーで戻されます。

管理通知ログは、セキュリティー・プラグインのデバッグおよび管理のための良い情報源です。 UNIX 上で管理通知ログを参照するには、sqllib/db2dump/*instance name*.nfy を調べます。 Windows オペレーティング・システム上で管理通知ログを参照するには、「イベント ビューア」ツールを使用します。「イベント ビューア」ツールは、Windows オペレーティング・システムの「スタート」ボタンから「設定」->「コントロール パネル」->「管理ツール」->「イベント ビューア」の順にナビゲートすると見つかります。以下は、セキュリティー・プラグインに関係した管理通知ログ値です。

- 13000 は、GSS-API セキュリティー・プラグイン API の呼び出しがエラーによって失敗し、オプションのエラー・メッセージが戻されたことを示します。

  ```
  SQLT_ADMIN_GSS_API_ERROR (13000)
  Plug-in "plug-in name" received error code "error code" from
  GSS API  "gss api name" with the error message "error message"
  ```

- 13001 は、DB2 セキュリティー・プラグイン API の呼び出しがエラーで失敗し、オプションのエラー・メッセージが戻されたことを示します。

  ```
  SQLT_ADMIN_PLUGIN_API_ERROR(13001)
  Plug-in "plug-in name" received error code "error code" from DB2
  security plug-in API "gss api name" with the error message
  "error message"
  ```

- 13002 は、DB2 がプラグインをアンロードできなかったことを示します。

  ```
  SQLT_ADMIN_PLUGIN_UNLOAD_ERROR (13002)
  Unable to unload plug-in "plug-in name". No further action required.
  ```

- 13003 は、無効なプリンシパル名を示します。

  ```
  SQLT_ADMIN_INVALID_PRIN_NAME (13003)
  The principal name "principal name" used for "plug-in name"
  is invalid. Fix the principal name.
  ```

- 13004 は、プラグイン名が無効なことを示します。パス区切り記号 (UNIX の場合は ″/″、Windows の場合は ″¥″) をプラグイン名の一部として使用することはできません。

  ```
  SQLT_ADMIN_INVALID_PLGN_NAME (13004)
  The plug-in name "plug-in name" is invalid. Fix the plug-in name.
  ```

- 13005 は、セキュリティー・プラグインがロードできなかったことを示します。プラグインを正しいディレクトリーに入れ、該当するデータベース・マネージャー構成パラメーターを更新してください。

  ```
  SQLT_ADMIN_PLUGIN_LOAD_ERROR (13005)
  Unable to load plug-in "plug-in name". Verify the plug-in existence and
  directory where it is located is correct.
  ```

- 13006 は、セキュリティー・プラグインによって予期しないエラーが検出されたことを示します。すべての db2support 情報を収集し、可能であれば db2trc を取り込んでから、IBM サポートに問い合わせてください。

  ```
  SQLT_ADMIN_PLUGIN_UNEXP_ERROR (13006)
  Plug-in encountered unexpected error. Contact IBM Support for further assistance.
  ```

注: Windows 64 ビットのデータベース・サーバー上でセキュリティー・プラグインを使用しているときに、セキュリティー・プラグインのロード・エラーが表示された場合は、32 ビットおよび 64 ビットの場合の考慮事項とセキュリティー・プラグインの命名規則に関するトピックを参照してください。 64 ビット・プラグイン・ライブラリーには、ライブラリー名に 64 という接尾部が付いていなければなりませんが、セキュリティー・プラグインのデータベース・マネージャー構成パラメーターへの入力にはこの接尾部は含めません。

プラグインの使用可能化

グループ検索プラグインの展開

DB2 セキュリティー・システムのグループ検索動作をカスタマイズするには、独自のグループ検索プラグインを開発するか、または第三者からこれを購入できます。

データベース管理システムに適したグループ検索プラグインを入手した後、それを展開することができます。

- グループ検索プラグインをデータベース・サーバー上に展開するには、以下のステップを実行します。
  1. グループ検索プラグイン・ライブラリーをサーバーのグループ・プラグイン・ディレクトリーにコピーします。
  2. データベース・マネージャー構成パラメーター *group_plugin* をプラグインの名前で更新します。
- グループ検索プラグインをデータベース・クライアント上に展開するには、以下のステップを実行します。
  1. グループ検索プラグイン・ライブラリーをクライアントのグループ・プラグイン・ディレクトリーにコピーします。
  2. データベース・クライアント上で、データベース・マネージャー構成パラメーター *group_plugin* をプラグインの名前で更新します。

ユーザー ID/パスワード・プラグインのデプロイ

DB2 セキュリティー・システムのユーザー ID/パスワード認証動作をカスタマイズするには、独自のユーザー ID/パスワード認証プラグインを開発するか、または第三者からこれを購入できます。

すべてのユーザー ID/パスワード・ベース認証プラグインは、意図しているそれらプラグインの使用法に応じて、クライアントのプラグイン・ディレクトリーか、サーバーのプラグイン・ディレクトリーのいずれかに置く必要があります。プラグインがクライアントのプラグイン・ディレクトリーに置かれる場合、それはローカル許可検査のためと、クライアントがサーバーと接続しようとするときのクライアントの妥当性検査のために使用されます。プラグインがサーバーのプラグイン・ディレクトリーに置かれる場合、それはサーバーへの着信接続の処理のためと、 USER または GROUP キーワードが指定されずに GRANT ステートメントが発行された場合の許可 ID の存在とその有効性の検査のために使用されます。ほとんどの場合、ユーザー ID/パスワード認証で必要となるのは、サーバー・サイドのプラグインのみです。一般にそれほど役に立ちませんが、クライアントのユーザー ID/パスワード・プラグインのみを使用することも可能です。非常に稀なことですが、クライアントとサーバーの両方に、一致するユーザー ID/パスワード・プラグインが必要になることもあります。

注: *既存の プラグインの新しい* バージョンをデプロイする場合は、DB2 サーバー、またはプラグインを使用するすべてのアプリケーションをまず停止する必要があります。新しいバージョンを (同じ名前で) 上書きコピーするときに、プラグインを使用するプロセスがまだ実行中になっていると、未定義の動作 (トラップなど) が発生します。この制約事項は、プラグインを初めてデプロイする場合や、プラグインが使用中になっていない場合には適用されません。

データベース管理システムに適したユーザー ID/パスワード認証プラグインを入手した後、それらをデプロイすることができます。

- ユーザー ID/パスワード認証プラグインをデータベース・サーバー上にデプロイするには、データベース・サーバー上で以下のステップを実行します。
  1. ユーザー ID/パスワード認証プラグイン・ライブラリーをサーバーのプラグイン・ディレクトリーにコピーします。
  2. データベース・マネージャー構成パラメーター *srvcon_pw_plugin* をサーバー・プラグインの名前で更新します。このプラグインは、サーバーが接続 (CONNECT) 要求およびアタッチ (ATTACH) 要求を処理する際に使用します。
  3. 次のいずれかを行ってください。
     - データベース・マネージャー構成パラメーター *srvcon_auth* に、CLIENT、SERVER、SERVER_ENCRYPT、DATA_ENCRYPT、または DATA_ENCRYPT_CMP の認証タイプを設定します。あるいは、
     - データベース・マネージャー構成パラメーター *srvcon_auth* に NOT_SPECIFIED を設定し、 *authentication* に CLIENT、 SERVER、SERVER_ENCRYPT、 DATA_ENCRYPT、または DATA_ENCRYPT_CMP の認証タイプを設定します。
- ユーザー ID/パスワード認証プラグインをデータベース・クライアント上にデプロイするには、各クライアント上で以下のステップを実行します。

1. ユーザー ID/パスワード認証プラグイン・ライブラリーをクライアントのプラグイン・ディレクトリーにコピーします。
2. データベース・マネージャー構成パラメーター *clnt_pw_plugin* をクライアント・プラグインの名前で更新します。このプラグインは、認証がどこで行われているかに関係なく、つまり、データベース構成パラメーター *authentication* が CLIENT に設定されているとき以外にもロードされ、呼び出されます。

- ユーザー ID/パスワード認証プラグインを使用したクライアント、サーバー、またはゲートウェイでのローカル許可に関しては、各クライアント、サーバー、またはゲートウェイ上で以下のステップを実行します。
  1. ユーザー ID/パスワード認証プラグイン・ライブラリーをクライアント、サーバー、またはゲートウェイ上のクライアント・プラグイン・ディレクトリーにコピーします。
  2. データベース・マネージャー構成パラメーター *clnt_pw_plugin* をプラグインの名前で更新します。
  3. *authentication* データベース・マネージャー構成パラメーターに、 CLIENT、 SERVER、 SERVER_ENCRYPT、 DATA_ENCRYPT、または DATA_ENCRYPT_CMP を設定します。

GSS-API プラグインのデプロイ

DB2 セキュリティー・システムの認証動作をカスタマイズするには、 GSS-API を使用して独自の認証プラグインを開発するか、または第三者からこれを購入できます。

Kerberos 以外のプラグイン・タイプの場合は、クライアントとサーバー上に、同じプラグイン・タイプだけでなく、一致するプラグイン名がなければなりません。クライアントとサーバー上のプラグインは、同一のベンダーのものである必要はありませんが、これらは互換性のある GSS-API トークンを生成して使用する必要があります。 Kerberos プラグインは標準化されているので、クライアントとサーバー上にはどのような組み合わせの Kerberos プラグインをデプロイしても構いません。しかし、それほど標準化されていない *x.509* 証明書などの種々の GSS-API メカニズムのインプリメンテーション間には、DB2 データベース・システムとの部分的な互換性しかない可能性があります。すべての GSS-API 認証プラグインは、意図しているそれらプラグインの使用法に応じて、クライアントのプラグイン・ディレクトリーか、サーバーのプラグイン・ディレクトリーのいずれかに置く必要があります。プラグインがクライアントのプラグイン・ディレクトリーに置かれる場合、それはローカル許可検査のためと、クライアントがサーバーと接続しようとするときに使用されます。プラグインがサーバーのプラグイン・ディレクトリーに置かれる場合、それはサーバーへの着信接続の処理のためと、 USER または GROUP キーワードが指定されずに GRANT ステートメントが発行された場合の許可 ID の存在とその有効性の検査のために使用されます。

注: *既存の* プラグインの *新しい* バージョンをデプロイする場合は、DB2 サーバー、またはプラグインを使用するすべてのアプリケーションをまず停止する必要があります。新しいバージョンを (同じ名前で) 上書きコピーするときに、プラグインを使用するプロセスがまだ実行中になっていると、未定義の動作 (トラップなど) が発生します。この制約事項は、プラグインを初めてデプロイする場合や、プラグインが使用中になっていない場合には適用されません。

データベース管理システムに適した GSS-API 認証プラグインを入手した後、それらをデプロイすることができます。

- GSS-API 認証プラグインをデータベース・サーバー上にデプロイするには、サーバー上で以下のステップを実行します。
  1. GSS-API 認証プラグイン・ライブラリーをサーバー・プラグイン・ディレクトリーにコピーします。このディレクトリーには、多数の GSS-API プラグインをコピーすることができます。
  2. データベース・マネージャー構成パラメーター *srvcon_gssplugin_list* を、GSS-API プラグイン・ディレクトリーにインストールされたプラグインの名前の順番のコンマ区切りのリストで更新します。
  3. 次のいずれかを行ってください。
     - データベース・マネージャー構成パラメーター *srvcon_auth* を GSSPLUGIN または GSS_SERVER_ENCRYPT に設定することは、サーバーが GSSAPI PLUGIN 認証方式を使用できるようにする一つの方法です。あるいは、
     - データベース・マネージャー構成パラメーター *srvcon_auth* を NOT_SPECIFIED に設定し、*authentication* を GSSPLUGIN または GSS_SERVER_ENCRYPT に設定することは、サーバーが GSSAPI PLUGIN 認証方式を使用できるようにする一つの方法です。
- GSS-API 認証プラグインをデータベース・クライアント上にデプロイするには、各クライアント上で以下のステップを実行します。
  1. GSS-API 認証プラグイン・ライブラリーをクライアントのプラグイン・ディレクトリーにコピーします。このディレクトリーには、多数の GSS-API プラグインをコピーすることができます。クライアントは、クライアント上で使用できる、サーバーのプラグインのリストに含まれている最初の GSS-API を選出することによって、接続 (CONNECT) またはアタッチ (ATTACH) 操作中の認証用の GSS-API プラグインを選択します。
  2. オプション: クライアントがアクセスするデータベースをカタログし、クライアントが GSS-API 認証プラグインのみを認証メカニズムとして受け入れることを示します。例:

     ```
     CATALOG DB testdb AT NODE testnode AUTHENTICATION GSSPLUGIN
     ```
- GSS-API 認証プラグインを使用したクライアント、サーバー、またはゲートウェイでのローカル許可に関しては、以下のステップを実行します。
  1. GSS-API 認証プラグイン・ライブラリーをクライアント、サーバー、またはゲートウェイ上のクライアント・プラグイン・ディレクトリーにコピーします。
  2. データベース・マネージャー構成パラメーター *local_gssplugin* をプラグインの名前で更新します。
  3. *authentication* データベース・マネージャー構成パラメーターに、GSSPLUGIN、または GSS_SERVER_ENCRYPT を設定します。

Kerberos プラグインのデプロイ

DB2 セキュリティー・システムの Kerberos 認証動作をカスタマイズするには、独自の Kerberos 認証プラグインを開発するか、または第三者からこれを購入できます。 Kerberos セキュリティー・プラグインは IPv6 をサポートしないことに注意してください。

注: *既存の プラグインの新しい* バージョンをデプロイする場合は、DB2 サーバー、またはプラグインを使用するすべてのアプリケーションをまず停止する必要があります。新しいバージョンを (同じ名前で) 上書きコピーするときに、プラグインを使用するプロセスがまだ実行中になっていると、未定義の動作 (トラップなど) が発生します。この制約事項は、プラグインを初めてデプロイする場合や、プラグインが使用中になっていない場合には適用されません。

データベース管理システムに適した Kerberos 認証プラグインを入手した後、それらをデプロイすることができます。

- Kerberos 認証プラグインをデータベース・サーバー上にデプロイするには、サーバー上で以下のステップを実行します。
  1. Kerberos 認証プラグイン・ライブラリーをサーバー・プラグイン・ディレクトリーにコピーします。
  2. 順番のコンマ区切りのリストとして指定されるデータベース・マネージャー構成パラメーター *srvcon_gssplugin_list* を更新して、 Kerberos サーバー・プラグイン名を含めます。このリストの中の 1 つのプラグインのみを Kerberos プラグインにすることができます。このリストがブランクで、 *authentication* に KERBEROS または KRB_SVR_ENCRYPT が設定されている場合は、デフォルトの DB2 Kerberos プラグイン IBMkrb5 が使用されます。
  3. 次のいずれかを行ってください。
     - データベース・マネージャー構成パラメーター *srvcon_auth* に、KERBEROS または KRB_SERVER_ENCRYPT の認証タイプを設定します。 (KERBEROS プラグインをデプロイし、引き続き GSSPLUGIN または GSS_SERVER_ENCRYPT を使用することができます。) あるいは、
     - データベース・マネージャー構成パラメーター *srvcon_auth* に NOT_SPECIFIED を設定し、 *authentication* に KERBEROS または KRB_SERVER_ENCRYPT の認証タイプを設定します。
- Kerberos 認証プラグインをデータベース・クライアント上にデプロイするには、各クライアント上で以下のステップを実行します。
  1. Kerberos 認証プラグイン・ライブラリーをクライアントのプラグイン・ディレクトリーにコピーします。
  2. データベース・マネージャー構成パラメーター *clnt_krb_plugin* を Kerberos プラグインの名前で更新します。 *clnt_krb_plugin* がブランクの場合、 DB2 はクライアントは Kerberos 認証を使用できないとみなします。サーバーがプラグインをサポートできない場合にのみ、この設定は適切です。サーバーとクライアントの両方がセキュリティー・プラグインをサポートする場合、デフォルトのサーバー・プラグイン *IBMkrb5* が *clnt_krb_plugin* のクライアントの値よりも優先して使用されます。 Kerberos 認証プラグインを使用したクライアント、サーバー、またはゲートウェイでのローカル許可に関しては、以下のステップを実行します。
     a. Kerberos 認証プラグイン・ライブラリーをクライアント、サーバー、またはゲートウェイ上のクライアント・プラグイン・ディレクトリーにコピーします。
     b. データベース・マネージャー構成パラメーター *clnt_krb_plugin* をプラグインの名前で更新します。

c. *authentication* データベース・マネージャー構成パラメーターに、KERBEROS または KRB_SERVER_ENCRYPT を設定します。

3. オプション: クライアントがアクセスするデータベースをカタログし、クライアントが Kerberos 認証プラグインのみを使用することを示します。例:

```
CATALOG DB testdb AT NODE testnode AUTHENTICATION KERBEROS
        TARGET PRINCIPAL service/host@REALM
```

注: Kerberos をサポートするプラットフォームの場合、 IBMkrb5 ライブラリーはクライアント・プラグイン・ディレクトリーに置かれます。 Kerberos プラグインは GSS-API プラグインを使用してインプリメントされているので、 DB2 はこのライブラリーを有効な GSS-API プラグインとして認識します。

LDAP ベースの認証とグループ検索サポート

DB2 データベース・マネージャーと DB2 Connect は、LDAP セキュリティー・プラグイン・モジュールを使用することによって LDAP ベースの認証とグループ検索機能をサポートしています。

DB2 データベース・マネージャーは、LDAP セキュリティー・プラグイン・モジュールを使用して、LDAP ディレクトリーに定義されているユーザーを認証します。その場合は、ユーザーとグループをオペレーティング・システムに対して定義する必要がありません。サポートされているプラットフォームは、AIX、IA32 版 Linux、x64 版 Linux、zSeries 版 Linux、Solaris、Windowsです。サポートされているプラットフォームに対応したコンパイル済みのバイナリー・プラグイン・モジュールは、それぞれに該当するディレクトリー (例えば、aix64、win32 など) にあります。

セキュリティー・プラグイン・モジュールの使用に対応した LDAP サーバーは、以下のとおりです。

- IBM Tivoli® Directory Server (ITDS) バージョン 5.2、6.0 以降
- Microsoft Active Directory (MSAD) バージョン 2000、2003 以降
- Sun Java System Directory Server Enterprise Edition バージョン 5.2 以降
- Novell eDirectory バージョン 8.7 以降
- IBM Lotus® Domino® LDAP Server バージョン 7.0 以降
- z/OS Integrated Security Services LDAP Server バージョン V1R6 以降

注: LDAP プラグイン・モジュールを使用する場合は、データベースに関連付けられているすべてのユーザーを LDAP サーバーで定義する必要があります。その中には、DB2 インスタンス所有者 ID と fenced ユーザーの両方が含まれます。(これらのユーザーは通常、オペレーティング・システムで定義されていますが、LDAP でも定義しなければなりません。)さらに、LDAP グループ・プラグイン・モジュールを使用する場合は、許可に必要なグループを LDAP サーバーで定義することも必要です。その中には、データベース・マネージャー構成で定義されている SYSADM、SYSMAINT、SYSCTRL、SYSMON の各グループが含まれます。

DB2 セキュリティー・プラグイン・モジュールは、後で取り上げるサーバー・サイドの認証、クライアント・サイドの認証、グループ検索のために使用できます。それぞれの環境に応じて、それらのタイプのうち、1 つか 2 つまたはすべてのプラグインを使用する必要があります。

DB2 セキュリティー・プラグイン・モジュールを使用するには、以下の手順を実行します。

1. 必要なのは、サーバー・プラグイン・モジュールか、クライアント・プラグイン・モジュールか、グループ・プラグイン・モジュールか、それらの組み合わせかを決定します。
2. プラグイン・モジュールを構成するために、IBM LDAP セキュリティー構成ファイル (デフォルトの名前は IBMLDAPSecurity.ini) の値を設定します。適切な値については、LDAP 管理者に問い合わせる必要があります。
3. プラグイン・モジュールを使用可能にします。
4. さまざまな LDAP ユーザー ID で接続をテストします。

サーバー認証プラグイン

サーバー認証プラグイン・モジュールは、CONNECT ステートメントと ATTACH ステートメントでクライアントから渡されるユーザー ID とパスワードのサーバー検証を実行します。必要に応じて、LDAP ユーザー ID を DB2 許可 ID に対応付けることも可能です。通常、サーバー・プラグイン・モジュールが必要になるのは、LDAP ユーザー ID とパスワードを使用して、DB2 データベース・マネージャーに対してユーザーを認証する場合です。

クライアント認証プラグイン

クライアント認証プラグイン・モジュールを使用するのは、クライアント・システムでユーザー ID とパスワードの検証を実行する場合、つまり、DB2 サーバーの SRVCON_AUTH 設定または AUTHENTICATION 設定が CLIENT になっている場合です。クライアントは、CONNECT ステートメントまたは ATTACH ステートメントに指定されているユーザー ID とパスワードを検証してから、そのユーザー ID を DB2 サーバーに送信します。ただし、クライアント認証は、セキュリティーの確保が難しいので、通常はお勧めできません。

クライアント認証プラグイン・モジュールは、データベース・サーバーのローカル・オペレーティング・システム・ユーザー ID と、そのユーザーに関連付けられている DB2 許可 ID が異なる場合にも必要になることがあります。クライアント・サイド・プラグインを使用すれば、データベース・サーバーで `db2start` などのローカル・コマンドの許可検査を実行する前に、ローカル・オペレーティング・システム・ユーザー ID と DB2 許可 ID を対応付けることが可能になります。

グループ検索プラグイン

グループ検索プラグイン・モジュールを使用すれば、LDAP サーバーから特定ユーザーのグループ・メンバーシップ情報を取得できます。LDAP を使用してグループ定義を格納する場合は、このモジュールが必要です。最も一般的なのは、以下のようなシナリオです。

- すべてのユーザーとグループが LDAP サーバーで定義されている場合

- データベース・サーバーでローカルに定義されているユーザーが、LDAP サーバーでも同じユーザー ID で定義されている場合 (インスタンス所有者と fenced ユーザーを含む)
- DB2 サーバーでパスワードを検証する場合 (つまり、DBM 構成ファイルで、AUTHENTICATION または SRVCON_AUTH の値が、SERVER、SERVER_ENCRYPT、DATA_ENCRYPT のいずれかに設定されている場合)

通常は、サーバーにサーバー認証プラグイン・モジュールとグループ検索プラグイン・モジュールをインストールするだけで十分です。DB2 クライアントには基本的に、LDAP プラグイン・モジュールをインストールする必要はありません。

LDAP グループ検索プラグイン・モジュールだけをインストールして、他の形式の認証プラグイン (Kerberos など) と組み合わせて使用することも可能です。その場合は、LDAP グループ検索プラグイン・モジュールに、ユーザーに関連付けられている DB2 許可 ID が渡されます。そのプラグイン・モジュールは、LDAP ディレクトリーで、AUTHID_ATTRIBUTE が一致するユーザーを検索し、そのユーザー・オブジェクトに関連付けられているグループを取得します。

LDAP プラグイン・モジュールの構成

LDAP プラグイン・モジュールを構成するには、それぞれの環境に合わせて、IBM LDAP セキュリティー・プラグイン構成ファイルを更新する必要があります。ほとんどの場合は、LDAP 管理者に問い合わせて、適切な構成値を確認することが必要です。

IBM LDAP セキュリティー・プラグイン構成ファイルのデフォルトの名前と場所は、以下のとおりです。

- UNIX: INSTHOME/sqllib/cfg/IBMLDAPSecurity.ini
- Windows: %DB2PATH%¥cfg¥IBMLDAPSecurity.ini

オプションとして、DB2LDAPSecurityConfig 環境変数を使用して、このファイルの場所を指定することも可能です。Windows では、グローバル・システム環境で DB2LDAPSecurityConfig を設定し、DB2 サービスからその値を参照できるようにしておく必要があります。

適切な構成値を確認するときに役立つ情報を以下の表にまとめます。

表 31. サーバー関連の値

パラメーター	説明
LDAP_HOST	LDAP サーバーの名前。 これは、LDAP サーバーのホスト名または IP アドレスのスペース区切りリストであり、オプションとしてそれぞれのポート番号を組み込むこともできます。 例えば、`host1[:port] [host2:[port2] ... ]` のようになります。 デフォルトのポート番号は 389 です。SSL が有効になっている場合は 636 になります。
ENABLE_SSL	SSL サポートを使用可能にするには、ENABLE_SSL を TRUE に設定します (GSKit をインストールしておく必要があります)。これは、オプション・パラメーターであり、デフォルトでは FALSE (SSL サポートなし) になります。

表 31. サーバー関連の値 (続き)

パラメーター	説明
SSL_KEYFILE	SSL 鍵リングのパス。 鍵ファイルが必要になるのは、GSKit のインストールで自動的にトラステッドに設定されない証明書を LDAP サーバーで使用している場合に限られます。 例えば、`SSL_KEYFILE = /home/db2inst1/IBMLDAPSecurity.kdb` のようになります。
SSL_PW	SSL 鍵リングのパスワード。例えば、`SSL_PW = keyfile-password` のようになります。

表 32. ユーザー関連の値

パラメーター	説明
USER_ OBJECTCLASS	ユーザー用の LDAP オブジェクト・クラス。 通常は、USER_OBJECTCLASS を `inetOrgPerson` (Microsoft Active Directory の場合はそのユーザー) に設定します。 例えば、`USER_OBJECTCLASS = inetOrgPerson` のようになります。
USER_BASEDN	ユーザー検索時に使用する LDAP 基底 DN。 指定しない場合は、ユーザー検索が LDAP ディレクトリーのルートから始まります。いくつかの LDAP サーバーでは、このパラメーターの値を指定することが必須になっています。 例えば、`USER_BASEDN = o=ibm` のようになります。

USERID_
ATTRIBUTE
	ユーザー ID に対応する LDAP ユーザー属性。
	ユーザーが非修飾ユーザー ID で DB2 CONNECT ステートメントを実行すると、USERID_ATTRIBUTE 属性、USER_OBJECTCLASS 属性、USER_BASEDN 属性 (指定されている場合) の組み合わせによって LDAP 検索フィルターが構成されます。
	例えば、`USERID_ATTRIBUTE = uid` の場合に、以下のステートメントを実行するとします。
	`db2 connect to MYDB user bob using bobpass`
	結果として、以下のような検索フィルターが生成されます。
	`&(objectClass=inetOrgPerson)(uid=bob)`

AUTHID_ ATTRIBUTE	DB2 許可 ID に対応する LDAP ユーザー属性。 通常は、USERID_ATTRIBUTE と同じです。 例えば、`AUTHID_ATTRIBUTE = uid` のようになります。

表 33. グループ関連の値

パラメーター	説明
GROUP_ OBJECTCLASS	グループ用の LDAP オブジェクト・クラス。 通常は、`groupOfNames` または `groupOfUniqueNames` (Microsoft Active Directory の場合は `group`) です。 例えば、`GROUP_OBJECTCLASS = groupOfNames` のようになります。
GROUP_BASEDN	グループ検索時に使用する LDAP 基底 DN。 指定しない場合は、グループ検索が LDAP ディレクトリーのルートから始まります。いくつかの LDAP サーバーでは、このパラメーターの値を指定することが必須になっています。 例えば、`GROUP_BASEDN = o=ibm` のようになります。

表 33. グループ関連の値 (続き)

パラメーター	説明
GROUPNAME_ ATTRIBUTE	グループの名前に対応する LDAP グループ属性。 例えば、GROUPNAME_ATTRIBUTE = cn のようになります。
GROUP_LOOKUP_ METHOD	ユーザーのグループ・メンバーシップを確認するための方法を指定します。可能な値は以下のとおりです。 • SEARCH_BY_DN: ユーザーをメンバーとしてリストするグループを検索します。メンバーシップは、GROUP_LOOKUP_ATTRIBUTE として定義するグループ属性 (通常は member または uniqueMember) によって示します。 • USER_ATTRIBUTE: この場合は、ユーザーの所属するグループがユーザー・オブジェクト自体の属性としてリストされます。この設定では、GROUP_LOOKUP_ATTRIBUTE として定義するユーザー属性を検索して、ユーザーの所属するグループを取得します (基本的に、Microsoft Active Directory の場合は memberOf、IBM Tivoli Directory Server の場合は ibm-allGroups です)。 例: GROUP_LOOKUP_METHOD = SEARCH_BY_DN GROUP_LOOKUP_METHOD = USER_ATTRIBUTE
GROUP_LOOKUP_ ATTRIBUTE	グループ・メンバーシップを確認するために使用する属性の名前 (GROUP_LOOKUP_METHOD の説明を参照)。 例: GROUP_LOOKUP_ATTRIBUTE = member GROUP_LOOKUP_ATTRIBUTE = ibm-allGroups
NESTED_GROUPS	NESTED_GROUPS を TRUE にすると、DB2 データベース・マネージャーは、グループ・メンバーシップを再帰的に検索するために、検出されるすべてのグループのグループ・メンバーシップを検索しようとします。 A が B に所属し、B が A に所属するといった循環も、正しく処理されます。 このパラメーターはオプションであり、デフォルトでは FALSE になります。

表 34. その他の値

パラメーター	説明
SEARCH_DN, SEARCH_PW	LDAP サーバーが匿名アクセスをサポートしていない場合や、ユーザーまたはグループの検索時に匿名アクセスでは不十分な場合は、検索実行時に使用する DN とパスワードをオプションとして指定できます。 例: SEARCH_DN = cn=root SEARCH_PW = rootpassword
DEBUG	DEBUG を TRUE に設定すると、LDAP 関連の問題のデバッグに役立つ追加情報が db2diag.log に書き込まれます。 ほとんどの追加情報は、DIAGLEVEL 4 (INFO) でログに記録されます。DEBUG は、デフォルトで false になります。

LDAP プラグイン・モジュールの使用可能化

DB2 インスタンスの中でそれぞれの LDAP プラグイン・モジュールが格納されている場所を以下の表にまとめます。

表 35. 64 ビットの UNIX システムと Linux システム

プラグイン・モジュールのタイプ	場所
サーバー	/sqllib/security64/plugin/IBM/server
クライアント	/sqllib/security64/plugin/IBM/client
グループ	/sqllib/security64/plugin/IBM/group

表 36. 32 ビットの UNIX システムと Linux システム

プラグイン・モジュールのタイプ	場所
サーバー	/sqllib/security32/plugin/IBM/server
クライアント	/sqllib/security32/plugin/IBM/client
グループ	/sqllib/security32/plugin/IBM/group

表 37. Windows システム (64 ビットと 32 ビットの両方)

プラグイン・モジュールのタイプ	場所
サーバー	%DB2PATH%¥security¥plugin¥IBM¥*instance-name*¥server
クライアント	%DB2PATH%¥security¥plugin¥IBM¥*instance-name*¥client
グループ	%DB2PATH%¥security¥plugin¥IBM¥*instance-name*¥group

注: 64 ビットの Windowsプラグイン・モジュールの場合は、ファイル名に 64 という数字が入っています。

DB2 コマンド行プロセッサーを使用して、データベース・マネージャー構成を更新し、必要なプラグイン・モジュールを使用可能にします。

- サーバー・プラグイン・モジュール:

  ```
  UPDATE DBM CFG USING SRVCON_PW_PLUGIN IBMLDAPauthserver
  ```

- クライアント・プラグイン・モジュール:

  ```
  UPDATE DBM CFG USING CLNT_PW_PLUGIN IBMLDAPauthclient
  ```

- グループ・プラグイン・モジュール:

  ```
  UPDATE DBM CFG USING GROUP_PLUGIN IBMLDAPgroups
  ```

db2 terminate コマンドを使用して、DB2 コマンド行プロセッサーの実行中のすべてのバックエンド・プロセスを終了してから、db2stop コマンドと db2start コマンドを使用してインスタンスを停止して再始動します。

LDAP ユーザー ID による接続

LDAP ディレクトリー内のオブジェクトの位置は、識別名 (DN) によって定義します。

DN は基本的に、ある種の階層を反映した複数パーツの名前です。例えば、次のようになります。

```
cn=John Smith, ou=Sales, o=WidgetCorp
```

LDAP プラグイン・モジュールを使用可能にして構成すると、ユーザーは、さまざまなストリングを使用して DB2 データベースに接続できるようになります。

- 完全 DN。例:

  ```
  connect to MYDB user 'cn=John Smith, ou=Sales, o=WidgetCorp'
  ```

- 部分 DN。部分 DN を使用して LDAP ディレクトリーを検索するときに、適切な検索基底 DN が定義されていれば、1 つの一致項目が検出されます。例:

  ```
  connect to MYDB user 'cn=John Smith' connect to MYDB user uid=jsmith
  ```

- 単純なストリング (等号が含まれていないストリング)。そのストリングは、USERID_ATTRIBUTE で修飾されて、部分 DN として処理されます。例:

  ```
  connect to MYDB user jsmith
  ```

注: CONNECT ステートメントまたは ATTACH ステートメントに指定するストリングにスペースや特殊文字が含まれている場合は、そのストリングを単一引用符で区切っておく必要があります。

ユーザー ID と DB2 許可 ID

ユーザーのユーザー *ID* は、ユーザー・オブジェクトに関連した属性 (通常は uid 属性) によって定義します。単純なストリング (例えば jsmith) の場合もあれば、組織階層の一部を反映した E メール・アドレス (例えば jsmith@sales.widgetcorp.com) のような場合もあります。

ユーザーの DB2 *許可 ID* は、DB2 データベース内のユーザーに関連した名前です。

以前は、サーバーのホスト・オペレーティング・システムでユーザーを定義するのが普通で、ユーザー ID と許可 ID は同じでした (ただし、許可 ID は基本的に大文字で表記します)。DB2 LDAP プラグイン・モジュールを使用すれば、LDAP ユーザー・オブジェクトのさまざまな属性をユーザー ID と許可 ID に関連付けることが可能になります。ほとんどの場合は、ユーザー ID と許可 ID を同じストリングにして、USERID_ATTRIBUTE と AUTHID_ATTRIBUTE の両方に同じ属性名を使用できます。ただし、許可 ID には組み込みたくない余分の情報がユーザー ID 属性に含まれていることが多い環境では、プラグインの初期設定ファイルで別の AUTHID_ATTRIBUTE を構成することも可能です。サーバーは、その AUTHID_ATTRIBUTE 属性の値を取り出して、ユーザーの内部 DB2 表記としてその値を使用します。

例えば、LDAP ユーザー ID が E メール・アドレス (例えば jsmith@sales.widgetcorp.com) のようになっているとします。DB2 許可 ID としてユーザーの部分 (jsmith) だけを使用する場合は、以下のようにします。

1. 短縮名が含まれている新しい属性を LDAP サーバーのすべてのユーザー・オブジェクトに関連付けます。
2. AUTHID_ATTRIBUTE にその新しい属性の名前を設定します。

ユーザーは、完全な LDAP ユーザー ID とパスワードを使用して、DB2 データベースに接続できます。例えば、次のようにします。

```
db2 connect to MYDB user 'jsmith@sales.widgetcorp.com' using 'pswd'
```

ところが、内部では、DB2データベース・マネージャーは、AUTHID_ATTRIBUTE で取得した短縮名 (この場合は jsmith) を使用してユーザーを参照します。

グループ検索に関する考慮事項

LDAP サーバーでは通常、ユーザー・オブジェクトの属性またはグループ・オブジェクトの属性としてグループ・メンバーシップ情報を表します。

- ユーザー・オブジェクトの属性として

  各ユーザー・オブジェクトには、GROUP_LOOKUP_ATTRIBUTE という属性があります。この属性に対して照会を実行すれば、そのユーザーのすべてのグループ・メンバーシップを取得できます。
- グループ・オブジェクトの属性として

  各グループ・オブジェクトにも、GROUP_LOOKUP_ATTRIBUTE という属性があります。その属性を使用すれば、そのグループのメンバーになっているすべてのユーザーをリストできます。特定ユーザーが属しているすべてのグループを列挙するために、ユーザー・オブジェクトがメンバーとして含まれているすべてのグループを検索することも可能です。

そのどちらかの方法で構成できる LDAP サーバーは多くありますし、両方の方法を同時にサポートしている LDAP サーバーもいくつかあります。ご使用の LDAP サーバーの構成方法については、LDAP 管理者に問い合わせてください。

LDAP プラグイン・モジュールを構成するときには、GROUP_LOOKUP_METHOD パラメーターを使用して、グループ検索の実行方法を指定できます。

- ユーザー・オブジェクトの GROUP_LOOKUP_ATTRIBUTE 属性を使用してグループ・メンバーシップを確認する必要がある場合は、GROUP_LOOKUP_METHOD = USER_ATTRIBUTE を設定します。
- グループ・オブジェクトの GROUP_LOOKUP_ATTRIBUTE 属性を使用してグループ・メンバーシップを確認する必要がある場合は、GROUP_LOOKUP_METHOD = SEARCH_BY_DN を設定します。

多くの LDAP サーバーは、グループ・オブジェクトの GROUP_LOOKUP_ATTRIBUTE 属性を使用してメンバーシップを確認します。その場合は、以下の例のように構成できます。

```
GROUP_LOOKUP_METHOD = SEARCH_BY_DN
GROUP_LOOKUP_ATTRIBUTE = groupOfNames
```

Microsoft Active Directory は通常、ユーザー属性としてグループ・メンバーシップを格納します。その場合は、以下の例のように構成できます。

```
GROUP_LOOKUP_METHOD = USER_ATTRIBUTE
GROUP_LOOKUP_ATTRIBUTE = memberOf
```

IBM Tivoli Directory Server は、両方の方法を同時にサポートしています。ユーザーのグループ・メンバーシップを照会するには、特別なユーザー属性 ibm-allGroups を以下の例のように使用できます。

```
GROUP_LOOKUP_METHOD = USER_ATTRIBUTE
GROUP_LOOKUP_ATTRIBUTE = ibm-allGroups
```

他の LDAP サーバーでも、グループ・メンバーシップを取得するために、それと似たような特別な属性が用意されている場合があります。一般に、ユーザー属性によってメンバーシップを取得する方が、ユーザーをメンバーとしてリストするグループを検索するよりも高速です。

LDAP ユーザーの認証とグループの検索に関するトラブルシューティング

LDAP ユーザーの認証やグループの検索で問題が発生した場合は、DB2 診断ログ db2diag.log と管理ログがトラブルシューティングのための情報源になります。

LDAP プラグイン・モジュールは通常、障害発生時の LDAP 戻りコード、検索フィルター、その他の役立つデータをログに記録します。LDAP プラグイン・モジュールの構成ファイルで DEBUG オプションを有効にすると、さらに多くの情報が db2diag.log ログに書き込まれます。その情報は確かにトラブルシューティングに役立ちますが、すべての追加データを 1 つのファイルに書き込む処理に関連したオーバーヘッドを考えると、実動システムでそのオプションを広範囲に使用することはお勧めできません。

データベース・マネージャーで DIAGLEVEL 構成パラメーターを 4 に設定して、LDAP プラグイン・モジュールから送られてくるすべてのメッセージをキャプチャーできるようにしてください。

セキュリティー・プラグインの作成

DB2 によるセキュリティー・プラグインのロード方法

各プラグイン・ライブラリーには、以下のような、プラグイン・タイプに応じた特定の名前を持つ初期化関数が含まれていなければなりません。

- サーバー・サイド認証プラグイン: `db2secServerAuthPluginInit()`
- クライアント・サイド認証プラグイン: `db2secClientAuthPluginInit()`
- グループ・プラグイン: `db2secGroupPluginInit()`

この関数は、プラグイン初期化関数と呼ばれます。プラグイン初期化関数は、指定されたプラグインを初期化し、プラグインの関数を呼び出すために必要な情報を DB2 に提供します。プラグイン初期化関数は、以下のパラメーターを受け入れます。

- プラグインを呼び出す DB2 インスタンスがサポートできる関数ポインター構造の最高バージョン番号
- インプリメンテーションを必要とするすべての API を指すポインターを収めた構造を指すポインター
- db2diag.log ファイルにログ・メッセージを追加する関数を指すポインター

- エラー・メッセージ・ストリングを指すポインター
- エラー・メッセージの長さ

以下は、グループ検索プラグインの初期化関数の関数シグニチャーです。

```
SQL_API_RC SQL_API_FN db2secGroupPluginInit(
  db2int32 version,
  void *group_fns,
  db2secLogMessage *logMessage_fn,
  char **errormsg,
  db2int32  *errormsglen);
```

注: プラグイン・ライブラリーを C++ でコンパイルする場合は、 extern "C" を使用してすべての関数を宣言する必要があります。 DB2 は、基礎オペレーティング・システムの動的ローダーを利用して、C++ のユーザー作成プラグイン・ライブラリーの内部で使用されている C++ コンストラクターおよびデストラクターを処理します。

初期化関数は、規定の関数名を使用しなければならない、プラグイン・ライブラリー内の唯一の関数です。その他のプラグイン関数は、初期化関数から戻された関数ポインターを通して参照されます。サーバー・プラグインは、DB2 サーバーの始動時にロードされます。クライアント・プラグインは、クライアント上で必要とされるときにロードされます。 DB2 は、プラグイン・ライブラリーをロードするとすぐに、この初期化関数の位置を解決して呼び出します。この関数固有のタスクは、以下のとおりです。

- 関数ポインターを、適切な関数構造を指すポインターにキャストする
- ライブラリー内の他の関数を指すポインターに入力する
- 戻される関数ポインター構造のバージョン番号に入力する

DB2 はプラグイン初期化関数を複数回呼び出すことがあります。このことが起こるのは、アプリケーションが動的に DB2 クライアント・ライブラリーをロードしてからこれをアンロードして再ロードし、再ロードの前と後の両方にプラグインから認証関数を実行した場合です。このような場合は、プラグイン・ライブラリーがアンロードされず、したがって再ロードもされないことがあります。ただし、この動作はオペレーティング・システムによって異なります。

別の例として、データベース・サーバー自身がクライアントとして振る舞うことがある、ストアード・プロシージャーやフェデレーテッド・システム呼び出しの実行時にも、DB2 がプラグイン初期化関数を複数回呼び出すことがあります。データベース・サーバー上のクライアント・プラグインとサーバー・プラグインが同じファイル内にある場合、 DB2 はプラグイン初期化関数を 2 回呼び出す可能性があります。

db2secGroupPluginInit が複数回呼び出されたことを検出した場合、プラグインは、プラグイン・ライブラリーを終了して再初期化するよう指示されたものとして、このイベントを処理する必要があります。したがって、プラグイン初期化関数は、db2secPluginTerm を呼び出すと実行されるクリーンアップ・タスクをすべて実行してから、再び関数ポインターのセットを戻す必要があります。

UNIX または Linux ベースのオペレーティング・システムが稼働している DB2 サーバーでは、DB2 は異なるプロセスでプラグイン・ライブラリーを複数回ロードして再初期化することがあります。

セキュリティー・プラグイン・ライブラリーの開発に関する制約事項

以下は、プラグイン・ライブラリーの作成に関連した制約事項です。

C-linkage
プラグイン・ライブラリーは、C-linkage とリンクされていなければなりません。プロトタイプ、プラグインのインプリメントに必要なデータ構造、およびエラー・コード定義を規定するヘッダー・ファイルは、C/C++ の場合にのみ準備されます。プラグイン・ライブラリーが C++ としてコンパイルされている場合は、 DB2 がロード時に解決する関数を extern ″C″ を用いて宣言する必要があります。

.NET 共通言語ランタイムはサポートされていません。
プラグイン・ライブラリーのソース・コードのコンパイルおよびリンクにおいて、 .NET 共通言語ランタイム (CLR) はサポートされません。

シグナル・ハンドラー
プラグイン・ライブラリーは、シグナル・ハンドラーをインストールしたり、シグナル・マスクを変更したりしてはなりません。なぜなら、これをすると、DB2 のシグナル・ハンドラーが妨げられるからです。 DB2 のシグナル・ハンドラーが妨げられると、プラグイン・コード自体にあるトラップを含めたエラーを報告してリカバリーする DB2 の機能が著しく妨げられます。さらに、プラグイン・ライブラリーは、C++ 例外を出してはなりません。なぜなら、これも DB2 のエラー処理を妨げるからです。

スレッド・セーフ
プラグイン・ライブラリーは、スレッド・セーフおよび再入可能でなければなりません。プラグイン初期化関数は、再入可能でなくてもよい唯一の API です。プラグイン初期化関数は異なるプロセスから複数回呼び出される可能性がありますが、その場合は、プラグインがすべての使用済みリソースをクリーンアップして、プラグイン自体を再初期化します。

終了ハンドラー、および標準 C ライブラリーとオペレーティング・システム呼び出しのオーバーライド
プラグイン・ライブラリーは、標準 C ライブラリーやオペレーティング・システム呼び出しをオーバーライドしてはなりません。さらに、プラグイン・ライブラリーは、終了ハンドラーや `pthread_atfork` ハンドラーをインストールしてはなりません。終了ハンドラーはプログラムが終了する前にアンロードされる可能性があるので、終了ハンドラーを使用することはお勧めしません。

ライブラリーの従属関係
Linux または UNIX では、プラグイン・ライブラリーをロードするプロセスは、`setuid` か `setgid` になります。このことは、プロセスが `$LD_LIBRARY_PATH`、 `$SHLIB_PATH`、または `$LIBPATH` 環境変数を利用して従属ライブラリーを検索できないことを意味します。したがって、従属ライブ

ラリーが次のような他の方法でアクセス可能にされていない限り、プラグイン・ライブラリーが追加のライブラリーに従属していてはなりません。

- /lib または /usr/lib の中に入れる。
- それらが常駐するディレクトリーを OS ワイド (Linux 上の ld.so.conf ファイル内など) で指定する。
- プラグイン・ライブラリー自体の RPATH で指定する。

この制限は、Windows オペレーティング・システムには当てはまりません。

シンボルの重複

可能であれば、プラグイン・ライブラリーは、シンボルの重複の可能性を減らすオプションとして使用できるオプション (アンバインドされた外部シンボル参照を削減するオプションなど) を用いてコンパイルおよびリンクする必要があります。例えば、HP、Solaris、および Linux 上で ″-Bsymbolic″ リンカー・オプションを使用するなら、シンボルの重複に関係した問題を防ぐことができます。ただし、AIX で作成されたプラグインの場合は、"-brtl" リンカー・オプションは明示的にも暗黙的にも使用しないでください。

32 ビット・アプリケーションと 64 ビット・アプリケーション

32 ビット・アプリケーションは、32 ビット・プラグインを使用する必要があります。 64 ビット・アプリケーションは、64 ビット・プラグインを使用する必要があります。詳細については、32 ビットと 64 ビットの考慮事項に関するトピックを参照してください。

テキスト・ストリング

入力テキスト・ストリングがヌル終了になっているという保証はなく、出力ストリングがヌル終了である必要はありません。その代わり、すべての入力ストリングに対して整数の長さが指定され、戻される長さとして整数を指すポインターが指定されます。

許可 ID パラメーターの引き渡し

DB2 がプラグインに渡す許可 ID (authid) パラメーター (入力 authid パラメーター) には、埋め込みブランクが除かれた大文字の authid が含まれます。プラグインが DB2 に戻す authid パラメーター (出力 authid パラメーター) には特別な処理は必要ありませんが、DB2 は authid を大文字に変換し、内部 DB2 規格に準じてブランクを埋め込みます。

パラメーターのサイズ制限

プラグイン API は、パラメーターの長さ制限として以下を使用します。

```
#define DB2SEC_MAX_AUTHID_LENGTH 255
#define DB2SEC_MAX_USERID_LENGTH 255
#define DB2SEC_MAX_USERNAMESPACE_LENGTH 255
#define DB2SEC_MAX_PASSWORD_LENGTH 255
#define DB2SEC_MAX_DBNAME_LENGTH 128
```

特定のプラグイン・インプリメンテーションでは、許可 ID、ユーザー ID、およびパスワードの最大長は、小さくする必要があるか、あるいは強制的に小さくされる可能性があります。特に、DB2 データベース・システムに付属しているオペレーティング・システム認証プラグインは、オペレーティング・システムの限界が上記の限界より低い場合、オペレーティング・システムが施行する最大ユーザー長、最大グループ長、および最大ネーム・スペース長の限界の制約を受けます。

AIX でのセキュリティー・プラグイン・ライブラリーの拡張子

AIX システムでは、セキュリティー・プラグイン・ライブラリーは、*.a* または *.so* というファイル名拡張子を持つことができます。プラグイン・ライブラリーをロードするのに使用するメカニズムは、次のように、どの拡張子が使用されているかによって異なります。

- ファイル名拡張子が *.a* のプラグイン・ライブラリーは、共用オブジェクト・メンバーを含むアーカイブであると想定されます。そのようなメンバーには、*shr.o* (32 ビット) または *shr64.o* (64 ビット) という名前を付けなければなりません。 32 ビットおよび 64 ビットの両方のメンバーを 1 つのアーカイブに収容することができ、それによって、両方のタイプのプラットフォームにデプロイすることができます。

  例えば、32 ビット・アーカイブ・スタイルのプラグイン・ライブラリーを作成するには、次のようにします。

  ```
  xlc_r -qmkshrobj -o shr.o MyPlugin.c -bE:MyPlugin.exp
  ar rv MyPlugin.a shr.o
  ```

- ファイル名拡張子が *.so* のプラグイン・ライブラリーは、動的にロード可能な共用オブジェクトであると想定されます。そのようなオブジェクトは、その作成時に使用したコンパイラーおよびリンカーのオプションに応じて、32 ビットまたは 64 ビットのどちらかになります。例えば、32 ビットのプラグイン・ライブラリーを作成するには、次のようにします。

  ```
  xlc_r -qmkshrobj -o MyPlugin.so MyPlugin.c -bE:MyPlugin.exp
  ```

AIX 以外のすべてのプラットフォームでは、セキュリティー・プラグイン・ライブラリーは、常に動的にロード可能な共有オブジェクトであるとみなされます。

セキュリティー・プラグインに関する制約事項

以下は、セキュリティー・プラグインの使用に関する制限事項です。

DB2 データベース・ファミリーのサポートに関する制約事項

GSS-API プラグインを使用して、Linux、UNIX、およびWindows 上の DB2 クライアントと、DB2 for z/OS などの別の DB2 ファミリー・サーバーとの間の接続を認証することはできません。また、クライアントとして機能する他の DB2 データベース・ファミリー製品から Linux、UNIX、または Windows 上の DB2 サーバーへの接続も認証できません。

ただし、Linux、UNIX、または Windows 上の DB2 クライアントを使用して他の DB2 データベース・ファミリー・サーバーに接続する場合には、クライアント・サイドのユーザー ID/パスワード・プラグイン (IBM 提供のオペレーティング・システム認証プラグインなど) を使用したり、独自のユーザー ID/パスワード・プラグインを作成したりすることができます。また、組み込みの Kerberos プラグインの使用や、自分独自のプラグインのインプリメントを行ってもかまいません。

Linux、 UNIX、または Windows 上の DB2 クライアントでは、GSSPLUGIN 認証タイプを使用してデータベースをカタログしてはなりません。

AUTHID ID に関する制限DB2 データベース・システムのバージョン 9.5 以降では、128 バイトの許可 ID を持つことができますが、その許可 ID がオペレーティ

ング・システムのユーザー ID またはグループ名として解釈される場合、オペレーティング・システムの命名上の制約が適用されます (例えば、ユーザー ID の制限は 8 または 30 文字、グループ名の制限は 30 文字です)。このため、128 バイトの許可 ID を付与できますが、この許可 ID を持つユーザーとしては接続することができません。独自のセキュリティー・プラグインを作成した場合は、許可 ID の拡張されたサイズを最大限に活用することができます。例えば、セキュリティー・プラグインに 30 バイトのユーザー ID を与えて、接続可能な認証中に、セキュリティー・プラグインが 128 バイトの許可 ID を返すことができます。

WebSphere® フェデレーション・サーバーのサポートに関する制約事項

DB2 II は、GSS_API プラグインからの委任証明書を使用して、データ・ソースへのアウトバウンド接続を確立することをサポートしていません。データ・ソースへの接続には、引き続き CREATE USER MAPPING コマンドを使用する必要があります。

データベース管理サーバーのサポートに関する制約事項

DB2 Administration Server (DAS) はセキュリティー・プラグインをサポートしていません。DAS はオペレーティング・システムの認証メカニズムのみをサポートします。

DB2 クライアントでのセキュリティー・プラグインに関する問題および制約事項 (Windows)

Windows オペレーティング・システム上の DB2 クライアント内でデプロイする予定のセキュリティー・プラグインの開発時には、プラグイン終了関数の中でどの補助ライブラリーもアンロードしないでください。この制約事項は、グループ、ユーザー ID とパスワード、Kerberos、および GSS-API プラグインを含む、すべてのタイプのクライアント・セキュリティー・プラグインに対して適用されます。このような、db2secPluginTerm、db2secClientAuthPluginTerm、および db2secServerAuthPluginTerm といった終了 API は、どの Windows プラットフォームでも呼び出されないので、該当するリソース・クリーンアップを行う必要があります。

この制約事項は、Windows での DLL のアンロードに関連したクリーンアップ問題に関係しています。

AIX 上での .a または .so の拡張子の付いたプラグイン・ライブラリーのロード

AIX では、セキュリティー・プラグイン・ライブラリーには、.a または .so のファイル名拡張子をつけることができます。プラグイン・ライブラリーをロードするのに使用するメカニズムは、次のように、どの拡張子が使用されているかによって異なります。

- .a のファイル名拡張子の付いたプラグイン・ライブラリー

  .a のファイル名拡張子の付いたプラグイン・ライブラリーは、共有オブジェクト・メンバーを収容するアーカイブであるとみなされます。そのようなメンバーには、shr.o (32 ビット) または shr64.o (64 ビット) という名前を付けなけれ

ばなりません。 32 ビットおよび 64 ビットの両方のメンバーを 1 つのアーカイブに収容することができ、それによって、両方のタイプのプラットフォームにデプロイすることができます。

例えば、32 ビット・アーカイブ・スタイルのプラグイン・ライブラリーを作成するには、次のようにします。

```
xlc_r -qmkshrobj -o shr.o MyPlugin.c -bE:MyPlugin.exp
ar rv MyPlugin.a shr.o
```

- .so のファイル名拡張子の付いたプラグイン・ライブラリー

.so のファイル名拡張子の付いたプラグイン・ライブラリーは、動的にロード可能な共有オブジェクトであるとみなされます。そのようなオブジェクトは、その作成時に使用したコンパイラーおよびリンカーのオプションに応じて、32 ビットまたは 64 ビットのどちらかになります。例えば、32 ビットのプラグイン・ライブラリーを作成するには、次のようにします。

```
xlc_r -qmkshrobj -o MyPlugin.so MyPlugin.c -bE:MyPlugin.exp
```

AIX 以外のすべてのプラットフォームでは、セキュリティー・プラグイン・ライブラリーは、常に動的にロード可能な共有オブジェクトであるとみなされます。

GSS-API セキュリティー・プラグインでは、メッセージの暗号化と署名はサポートされない

メッセージの暗号化および署名は、GSS-API セキュリティー・プラグインでは使用できません。

セキュリティー・プラグインの戻りコード

すべてのセキュリティー・プラグイン API は、API の実行の成功や失敗を示すために整数の値を戻す必要があります。戻りコード値 0 は、API が正常に実行したことを示します。 -3、-4、および -5 以外のすべての負の戻りコードは、API がエラーを検出したことを示します。

-3、-4、または -5 が付く戻りコードを除き、セキュリティー・プラグイン API から戻されるすべての負の戻りコードは、SQLCODE -1365、SQLCODE -1366、または SQLCODE -30082 にマップされます。 -3、-4、および -5 の値は、許可 ID が有効なユーザーまたはグループを表しているかどうかを示すために使用されます。

すべてのセキュリティー・プラグイン API の戻りコードは、DB2 の組み込みディレクトリー SQLLIB/include にある db2secPlugin.h で定義されます。

すべてのセキュリティー・プラグインの戻りコードに関する詳細については、以下の表で説明しています。

表 38. セキュリティー・プラグインの戻りコード

戻りコード	定義値	意味	関連 API
0	DB2SEC_PLUGIN_OK	プラグイン API が正常に実行されました。	すべて
-1	DB2SEC_PLUGIN_UNKNOWNERROR	プラグイン API で想定外のエラーが発生しました。	すべて

表 38. セキュリティー・プラグインの戻りコード (続き)

戻りコード	定義値	意味	関連 API
-2	DB2SEC_PLUGIN_BADUSER	入力として渡されたユーザー ID が定義されていません。	db2secGenerateInitialCred db2secValidatePassword db2secRemapUserid db2secGetGroupsForUser
-3	DB2SEC_PLUGIN _INVALIDUSERORGROUP	このユーザーまたはグループがありません。	db2secDoesAuthIDExist db2secDoesGroupExist
-4	DB2SEC_PLUGIN _USERSTATUSNOTKNOWN	ユーザー状況が不明です。これは DB2 ではエラーとして扱われません。これは、GRANT ステートメントが、 authid がユーザーまたはオペレーティング・システム・グループのどちらを表しているか判別するために使用します。	db2secDoesAuthIDExist
-5	DB2SEC_PLUGIN _GROUPSTATUSNOTKNOWN	グループ状況が不明です。これは DB2 ではエラーとして扱われません。これは、GRANT ステートメントが、 authid がユーザーまたはオペレーティング・システム・グループのどちらを表しているか判別するために使用します。	db2secDoesGroupExist
-6	DB2SEC_PLUGIN_UID_EXPIRED	ユーザー ID が期限切れです。	db2secValidatePassword db2GetGroupsForUser db2secGenerateInitialCred
-7	DB2SEC_PLUGIN_PWD_EXPIRED	パスワードが期限切れです。	db2secValidatePassword db2GetGroupsForUser db2secGenerateInitialCred
-8	DB2SEC_PLUGIN_USER_REVOKED	ユーザーが失効しています。	db2secValidatePassword db2GetGroupsForUser
-9	DB2SEC_PLUGIN _USER_SUSPENDED	ユーザーが一時失効しています。	db2secValidatePassword db2GetGroupsForUser
-10	DB2SEC_PLUGIN_BADPWD	パスワードが無効です。	db2secValidatePassword db2secRemapUserid db2secGenerateInitialCred
-11	DB2SEC_PLUGIN _BAD_NEWPASSWORD	新規パスワードが無効です。	db2secValidatePassword db2secRemapUserid
-12	DB2SEC_PLUGIN _CHANGEPASSWORD _NOTSUPPORTED	パスワード変更はサポートされていません。	db2secValidatePassword db2secRemapUserid db2secGenerateInitialCred

表 38. セキュリティー・プラグインの戻りコード (続き)

戻りコード	定義値	意味	関連 API
-13	DB2SEC_PLUGIN_NOMEM	メモリー不足のため、プラグインがメモリーを割り振れませんでした。	すべて
-14	DB2SEC_PLUGIN_DISKERROR	プラグインがディスク・エラーを検出しました。	すべて
-15	DB2SEC_PLUGIN_NOPERM	ファイルの許可が不適切なため、プラグインがファイルにアクセスできませんでした。	すべて
-16	DB2SEC_PLUGIN_NETWORKERROR	プラグインがネットワーク・エラーを検出しました。	すべて
-17	DB2SEC_PLUGIN_CANTLOADLIBRARY	プラグインが必要なライブラリーをロードできません。	db2secGroupPluginInit db2secClientAuthPluginInit db2secServerAuthPluginInit
-18	DB2SEC_PLUGIN_CANT_OPEN_FILE	欠落ファイルや不適切なファイル許可以外の理由のために、プラグインがファイルをオープンして読み取ることができません。	すべて
-19	DB2SEC_PLUGIN_FILENOTFOUND	ファイル・システムにファイルがないために、プラグインがファイルをオープンして読み取ることができません。	すべて
-20	DB2SEC_PLUGIN_CONNECTION_DISALLOWED	接続できるデータベース、または特定のデータベースに接続できない TCP/IP アドレスについての制約事項のために、プラグインが接続を拒否しています。	すべてのサーバー・サイドのプラグイン API。
-21	DB2SEC_PLUGIN_NO_CRED	GSS API プラグインのみ: 初期クライアント証明書がありません。	db2secGetDefaultLoginContext db2secServerAuthPluginInit
-22	DB2SEC_PLUGIN_CRED_EXPIRED	GSS API プラグインのみ: クライアント証明書が期限切れです。	db2secGetDefaultLoginContext db2secServerAuthPluginInit
-23	DB2SEC_PLUGIN_BAD_PRINCIPAL_NAME	GSS API プラグインのみ: プリンシパル名が無効です。	db2secProcessServerPrincipalName
-24	DB2SEC_PLUGIN_NO_CON_DETAILS	この戻りコードは、db2secGetConDetails コールバック (例えば、DB2 からプラグインへの) によって戻され、 DB2 がクライアントの TCP/IP アドレスを判別できないことを示します。	db2secGetConDetails
-25	DB2SEC_PLUGIN_BAD_INPUT_PARAMETERS	プラグイン API を呼び出すとき、いくつかのパラメーターが無効か、または欠落しています。	すべて

表 38. セキュリティー・プラグインの戻りコード (続き)

戻りコード	定義値	意味	関連 API
-26	DB2SEC_PLUGIN _INCOMPATIBLE_VER	プラグインによって報告された API のバージョンに、DB2 との互換性がありません。	db2secGroupPluginInit db2secClientAuthPluginInit db2secServerAuthPluginInit
-27	DB2SEC_PLUGIN_PROCESS_LIMIT	プラグインが新規プロセスを作成するために、十分なリソースを使用できません。	すべて
-28	DB2SEC_PLUGIN_NO_LICENSES	プラグインがユーザー・ライセンスの問題を検出しました。基礎メカニズムのライセンスが限界に達している可能性があります。	すべて

セキュリティー・プラグインのエラー・メッセージ処理

セキュリティー・プラグイン API でエラーが発生すると、 API は errormsg フィールドに ASCII テキスト・ストリングを戻して、戻りコードよりも具体的な問題の説明を提示することがあります。

例えば、errormsg ストリングに、 "File /home/db2inst1/mypasswd.txt does not exist." などのメッセージが含められます。 DB2 はこのストリングをまるごと DB2 管理通知ログに書き込み、さらに、短縮版をいくつかの SQL メッセージにトークンとして組み込みます。 SQL メッセージ内のトークンは限られた長さにしかできないため、これらのメッセージは短くし、これらのメッセージの重要な変数の部分がストリングの先頭に来るようにしてください。デバッグに役立てるため、エラー・メッセージにはセキュリティー・プラグインの名前を追加することを考慮してください。

パスワード期限切れエラーなどの緊急でないエラーに関しては、errormsg ストリングは、 DIAGLEVEL データベース・マネージャー構成パラメーターに 4 が設定されている場合にのみダンプされます。

これらのエラー・メッセージ用のメモリーは、セキュリティー・プラグインによって割り振られる必要があります。したがって、プラグインは、このメモリーを解放するための API である db2secFreeErrormsg を備えていなければなりません。

errormsg フィールドは、 API がゼロ以外の値を戻した場合にのみ DB2 によってチェックされます。そのため、プラグインは、エラーがない場合は、この戻りエラー・メッセージ用のメモリーを割り振るべきではありません。

初期化時には、メッセージ・ロギング関数ポインター logMessage_fn が、グループ、クライアント、およびサーバーのプラグインに渡されます。プラグインはこの関数を使用してデバッグ情報を db2diag.log に記録できます。例:

```
// Log an message indicate init successful
(*(logMessage_fn))(DB2SEC_LOG_CRITICAL,
                   "db2secGroupPluginInit successful",
                    strlen("db2secGroupPluginInit successful"));
```

db2secLogMessage 関数の各パラメーターについて詳しくは、各プラグイン・タイプの初期化 API を参照してください。

セキュリティー・プラグイン API の呼び出し順序

DB2 データベース・マネージャーがセキュリティー・プラグイン API を呼び出す主なシナリオを以下に示します。

- クライアントでのデータベース接続 (暗黙的および明示的)
  - CLIENT
  - サーバー・ベース (SERVER、SERVER_ENCRYPT、DATA_ENCRYPT)
  - GSSAPI および Kerberos
- クライアント、サーバー、またはゲートウェイでのローカル許可
- サーバーでのデータベース接続
- サーバーでの GRANT ステートメント
- サーバーで許可 ID が所属するグループのリストを取得する

注: DB2 データベース・サーバーは、ローカル許可が必要な db2start、db2stop、および db2trc などのデータベース・アクションを、クライアント・アプリケーションと同様に取り扱います。

DB2 データベース・マネージャーがセキュリティー・プラグイン API を呼び出す順序は、これらの各操作ごとに異なります。これらの各シナリオにおいて、DB2 データベース・マネージャーが呼び出す API の順序を以下に示します。

CLIENT - 暗黙的

ユーザー構成認証タイプが CLIENT の場合、DB2 クライアント・アプリケーションは以下のセキュリティー・プラグイン API を呼び出します。

- db2secGetDefaultLoginContext();
- db2secValidatePassword();
- db2secFreetoken();

暗黙的な認証の場合、すなわち、特定のユーザー ID やパスワードを指定せずに接続する場合は、ユーザー ID/パスワード・プラグインを使用していると、db2secValidatePassword API が呼び出されます。必要に応じ、プラグイン作成者はこの API によって暗黙的な認証を禁止することができます。

CLIENT - 明示的

明示的な認証の場合、すなわち、ユーザー ID とパスワードの両方が指定されているデータベースに接続する場合は、*authentication* データベース・マネージャー構成パラメーターが CLIENT に設定されていると、DB2 クライアント・アプリケーションは、インプリメンテーションが必要とする場合には、以下のセキュリティー・プラグイン API を複数回呼び出します。

- db2secRemapUserid();
- db2secValidatePassword();
- db2secFreeToken();

サーバー・ベース (SERVER、SERVER_ENCRYPT、DATA_ENCRYPT) - 暗黙的

暗黙的な認証の場合、クライアントとサーバーがユーザーID/パスワードの

認証を折衝している場合 (例えば、サーバー側の *srvcon_auth* パラメーターが SERVER、SERVER_ENCRYPT、DATA_ENCRYPT、または DATA_ENCRYPT_CMP に設定されている場合)、クライアント・アプリケーションは以下のセキュリティー・プラグイン API を呼び出します。

- `db2secGetDefaultLoginContext();`
- `db2secFreeToken();`

サーバー・ベース (SERVER、SERVER_ENCRYPT、DATA_ENCRYPT) - 明示的
明示的な認証の場合、クライアントとサーバーがユーザーID/パスワードの認証を折衝している場合 (例えば、サーバー側の srvcon_auth パラメーターが SERVER、SERVER_ENCRYPT、DATA_ENCRYPT、または DATA_ENCRYPT_CMP に設定されている場合)、クライアント・アプリケーションは以下のセキュリティー・プラグイン API を呼び出します。

- `db2secRemapUserid();`

GSSAPI および Kerberos - 暗黙的
暗黙的な認証の場合、クライアントとサーバーが GSS-API または Kerberos 認証を折衝している場合 (例えば、サーバー側の *srvcon_auth* パラメーターが KERBEROS、KRB_SERVER_ENCRYPT、GSSPLUGIN、または GSS_SERVER_ENCRYPT に設定されている場合)、クライアント・アプリケーションは以下のセキュリティー・プラグイン API を呼び出します。 (gss_init_sec_context() を呼び出すときは、GSS_C_NO_CREDENTIAL が入力証明書として使用されます。)

- `db2secGetDefaultLoginContext();`
- `db2secProcessServerPrincipalName();`
- `gss_init_sec_context();`
- `gss_release_buffer();`
- `gss_release_name();`
- `gss_delete_sec_context();`
- `db2secFreeToken();`

マルチフロー GSS-API サポートを使用すると、インプリメンテーションが必要とする場合には、 `gss_init_sec_context()` を複数回呼び出すことができます。

GSSAPI および Kerberos - 明示的
折衝された認証タイプが GSS-API または Kerberos の場合は、クライアント・アプリケーションが GSS-API プラグイン用に、以下のセキュリティー・プラグイン API をこの順序で呼び出します。特に記述されていない場合、これらの API は暗黙的な認証と明示的な認証の両方に使用されます。

- `db2secProcessServerPrincipalName();`
- `db2secGenerateInitialCred();` (明示的な認証の場合のみ)
- `gss_init_sec_context();`
- `gss_release_buffer ();`
- `gss_release_name();`
- `gss_release_cred();`
- `db2secFreeInitInfo();`

- gss_delete_sec_context();
- db2secFreeToken();

サーバーから相互認証トークンが戻され、インプリメンテーションが必要とする場合には、API gss_init_sec_context() が複数回呼び出されることがあります。

クライアント、サーバー、またはゲートウェイでのローカル許可

ローカル許可の場合は、使用される DB2 コマンドが、以下のセキュリティー・プラグイン API を呼び出します。

- db2secGetDefaultLoginContext();
- db2secGetGroupsForUser();
- db2secFreeToken();
- db2secFreeGroupList();

これらの API が、ユーザー ID/パスワードと GSS-API の両方の認証メカニズム用に呼び出されます。

サーバーでのデータベース接続

データベース・サーバー上でのデータベース接続の場合は、DB2 エージェント・プロセスまたはスレッドが、ユーザー ID/パスワード認証メカニズム用に以下のセキュリティー・プラグイン API を呼び出します。

- db2secValidatePassword(); (*authentication* データベース構成パラメーターが CLIENT でない場合のみ)
- db2secGetAuthIDs();
- db2secGetGroupsForUser();
- db2secFreeToken();
- db2secFreeGroupList();

データベースへの接続の場合は、DB2 エージェント・プロセスまたはスレッドが、GSS-API 認証メカニズム用に以下のセキュリティー・プラグイン API を呼び出します。

- gss_accept_sec_context();
- gss_release_buffer();
- db2secGetAuthIDs();
- db2secGetGroupsForUser();
- gss_delete_sec_context();
- db2secFreeGroupListMemory();

サーバーでの GRANT ステートメント

USER または GROUP キーワードを指定しない GRANT ステートメント (例えば、 "GRANT CONNECT ON DATABASE TO user1") の場合、 DB2 エージェント・プロセスは user1 がユーザー、グループ、またはその両方のいずれであるかを判別できなければなりません。そのため、DB2 エージェント・プロセスまたはスレッドは以下のセキュリティー・プラグイン API を呼び出します。

- db2secDoesGroupExist();
- db2secDoesAuthIDExist();

サーバーで authid が所属するグループのリストを取得する

データベース・サーバーで、許可 ID が所属するグループのリストを取得する必要がある場合、DB2 エージェント・プロセスまたはスレッドは以下のセキュリティー・プラグイン API を、許可 ID のみを入力として呼び出します。

- `db2secGetGroupsForUser();`

他のセキュリティー・プラグインからのトークンはありません。

第 8 章 セキュリティー・プラグイン API

ユーザーが DB2 データベース・システムの認証およびグループ・メンバーシップの検索の動作をカスタマイズできるように、既存のプラグイン・モジュールの変更や、新規セキュリティー・プラグイン・モジュールの作成の際に使用できる API が DB2 データベース・システムに用意されています。

セキュリティー・プラグイン・モジュールを作成するときは、DB2 データベース・マネージャーが呼び出す標準の認証またはグループ・メンバーシップの検索関数をインプリメントする必要があります。使用できる 3 つのタイプのプラグイン・モジュールに関してインプリメントする必要がある機能は、以下のとおりです。

グループ検索
特定のユーザーのグループ・メンバーシップ情報を検索し、指定されたストリングが有効なグループ名を表しているかどうかを判別します。

ユーザー ID/パスワード認証
この認証は、デフォルトのセキュリティー・コンテキストを識別し (クライアントのみ)、パスワードを検証して必要があれば変更し、指定されたストリングが有効なユーザーを表しているかどうか判別し (サーバーのみ)、クライアントで規定されているユーザー ID またはパスワードをサーバーへの送信の前に変更し (クライアントのみ)、指定されたユーザーに関連付けられた DB2 許可 ID を戻します。

GSS-API 認証
この認証は、必要な GSS-API 関数をインプリメントし、デフォルトのセキュリティー・コンテキストを識別し (クライアント・サイドのみ)、ユーザー ID およびパスワードを基に初期証明書を生成し、必要があればパスワードを変更し (クライアント・サイドのみ)、セキュリティー・チケットを作成して受け入れ、指定された GSS-API セキュリティー・コンテキストに関連付けられた DB2 許可 ID を戻します。

以下は、プラグイン API の説明に使用される用語の定義です。

プラグイン
DB2 が、ユーザー作成の認証またはグループ・メンバーシップの検索関数にアクセスするためにロードする動的にロード可能なライブラリー。

暗黙的な認証
ユーザー ID またはパスワードが指定されないデータベースへの接続。

明示的な認証
ユーザー ID とパスワードの両方が指定されるデータベースへの接続。

Authid データベース内での権限および特権が付与された個人またはグループを表す内部 ID。内部では、DB2 authid は大文字に変換されます。これは、8 文字以上です (8 文字になるようブランクが埋め込まれます)。現在のところ、DB2 は、7 ビット ASCII で表記できる authid、ユーザー ID、パスワード、グループ名、ネーム・スペース、およびドメイン名を必要とします。

ローカル許可
許可をインプリメントしているサーバーまたはクライアントでのローカルな許可です。これは、データベース・マネージャーの開始と停止、DB2 トレースのオン/オフ、データベース・マネージャー構成の更新などのアクション (データベース接続以外のアクション) を実行する権限がユーザーにあるかどうかを検査します。

ネーム・スペース
ユーザーの集合またはグループ。この中で個々のユーザー ID はユニークでなければなりません。一般的な例としては、Windows ドメインと Kerberos レルムがあります。例えば、Windows ドメイン ″usa.company.com″ では、すべてのユーザー名がユニークでなければなりません。例えば、″user1@usa.company.com″ などとなります。他のドメインにある同一のユーザー ID (例えば、″user1@canada.company.com″) は、別のユーザーを表します。完全修飾ユーザー ID には、ユーザー ID とネーム・スペースのペア (例えば ″user@domain.name″ または ″domain¥user″) が含まれます。

入力 DB2 が、セキュリティー・プラグイン API パラメーターに値を入力することを示します。

出力 セキュリティー・プラグイン API が API パラメーターの値を入力することを示します。

グループ検索プラグイン用の API

グループ検索プラグイン・モジュール用には、以下の API をインプリメントする必要があります。

- db2secGroupPluginInit

  注: db2secGroupPluginInit API は、以下のプロトタイプを持つ API を指すポインター *logMessage_fn を入力としてとります。

  ```
  SQL_API_RC (SQL_API_FN db2secLogMessage)
  (
  db2int32 level,
  void    *data,
  db2int32 length
  );
  ```

  db2secLogMessage API により、プラグインはデバッグまたは通知の目的で、メッセージを db2diag.log に記録することができます。この API は DB2 データベース・システムによって提供されるため、インプリメントする必要はありません。
- db2secPluginTerm
- db2secGetGroupsForUser
- db2secDoesGroupExist
- db2secFreeGroupListMemory
- db2secFreeErrormsg
- 外部で解決できなければならない唯一の API は、db2secGroupPluginInit です。この API は、void * パラメーターをとり、それは以下のタイプにキャストする必要があります。

```
typedef struct db2secGroupFunctions_1
{
db2int32 version;
db2int32 plugintype;
SQL_API_RC (SQL_API_FN * db2secGetGroupsForUser)
(
const char *authid,
db2int32    authidlen,
const char *userid,
db2int32    useridlen,
const char *usernamespace,
db2int32    usernamespacelen,
db2int32    usernamespacetype,
const char *dbname,
db2int32    dbnamelen,
const void *token,
db2int32    tokentype,
db2int32    location,
const char *authpluginname,
db2int32    authpluginnamelen,
void      **grouplist,
db2int32   *numgroups,
char      **errormsg,
db2int32   *errormsglen
);

SQL_API_RC (SQL_API_FN * db2secDoesGroupExist)
(
const char *groupname,
db2int32    groupnamelen,
char      **errormsg,
db2int32   *errormsglen
);

SQL_API_RC (SQL_API_FN * db2secFreeGroupListMemory)
(
void     *ptr,
char    **errormsg,
db2int32 *errormsglen
);

SQL_API_RC (SQL_API_FN * db2secFreeErrormsg)
(
char *msgtobefree
);

SQL_API_RC (SQL_API_FN * db2secPluginTerm)
(
char    **errormsg,
db2int32 *errormsglen
);

} db2secGroupFunctions_1;
```

db2secGroupPluginInit API は、外部で使用できる残りの関数のアドレスを割り当てます。

注: _1 はこれが API のバージョン 1 に対応する構造であることを示します。後続のインターフェース・バージョンの拡張子は _2、_3 というようになります。

db2secDoesGroupExist API - グループの存在のチェック

authid がグループを表すかどうかを判断します。

グループ名が存在する場合、API は、正常に完了したことを示すために値 DB2SEC_PLUGIN_OK を戻すことができなければなりません。グループ名が有効でない場合は、値 DB2SEC_PLUGIN_INVALIDUSERORGROUP も戻されなければなりません。入力が有効なグループかどうか判別できない場合は、API が値 DB2SEC_PLUGIN_GROUPSTATUSNOTKNOWN を戻すこともできます。無効なグループ (DB2SEC_PLUGIN_INVALIDUSERORGROUP) や不明なグループ (DB2SEC_PLUGIN_GROUPSTATUSNOTKNOWN) の値が戻される場合、 DB2 は USER キーワードおよび GROUP キーワードのない GRANT ステートメントを発行するときに、 authid がグループかユーザーかを判別できない可能性があり、その結果 SQLCODE -569、SQLSTATE 56092 のエラーがユーザーに戻されます。

API とデータ構造構文

```
SQL_API_RC ( SQL_API_FN *db2secDoesGroupExist)
                      ( const char *groupname,
                        db2int32 groupnamelen,
                        char      **errormsg,
                        db2int32 *errormsglen );
```

db2secDoesGroupExist API パラメーター

groupname
入力。末尾ブランクなしの大文字の authid。

groupnamelen
入力。 groupname パラメーター値のバイト単位の長さ。

errormsg
出力。db2secDoesGroupExist API が正常に実行されない場合にこのパラメーターで戻されることのある、プラグインによって割り振られた ASCII エラー・メッセージ・ストリングのアドレスを指すポインター。

errormsglen
出力。errormsg パラメーターのエラー・メッセージ・ストリングのバイト単位の長さを示す整数を指すポインター。

db2secFreeErrormsg API - エラー・メッセージのメモリーの解放

直前の API 呼び出しのエラー・メッセージを保持するために使用されているメモリーを解放します。これは、エラー・メッセージを一緒に戻さない唯一の API です。この API がエラーを戻す場合、DB2 はそれをログに記録して続行します。

API とデータ構造構文

```
SQL_API_RC ( SQL_API_FN *db2secFreeErrormsg)
                      ( char *errormsg );
```

db2secFreeErrormsg API パラメーター

msgtofree
入力。以前の API 呼び出しで割り振られたエラー・メッセージを指すポインター。

db2secFreeGroupListMemory API - グループ・リストのメモリーの解放

直前の db2secGetGroupsForUser API の呼び出しのグループのリストを保持するのに使用されているメモリーを解放します。

API とデータ構造構文

```
SQL_API_RC ( SQL_API_FN *db2secFreeGroupListMemory)
                      ( void *ptr,
                        char      **errormsg,
                        db2int32 *errormsglen );
```

db2secFreeGroupListMemory API パラメーター

ptr 入力。解放されるメモリーを指すポインター。

errormsg
出力。プラグインによって割り振られた ASCII エラー・メッセージ・ストリングのアドレスを指すポインター。このエラー・メッセージ・ストリングは、db2secFreeGroupListMemory API が正常に実行されない場合にこのパラメーターに戻されることがあります。

errormsglen
出力。errormsg パラメーターのエラー・メッセージ・ストリングのバイト単位の長さを示す整数を指すポインター。

db2secGetGroupsForUser API - ユーザーのグループのリストの取得

ユーザーが所属するグループのリストを戻します。

API とデータ構造構文

```
SQL_API_RC ( SQL_API_FN *db2secGetGroupsForUser)
                      ( const char *authid,
                        db2int32 authidlen,
                        const char *userid,
                        db2int32 useridlen,
                        const char *usernamespace,
                        db2int32 usernamespacelen,
                        db2int32 usernamespacetype,
                        const char *dbname,
                        db2int32 dbnamelen,
                        void *token,
                        db2int32 tokentype,
                        db2int32 location,
                        const char *authpluginname,
                        db2int32 authpluginnamelen,
                        void      **grouplist,
                        db2int32 *numgroups,
                        char      **errormsg,
                        db2int32 *errormsglen );
```

db2secGetGroupsForUser API パラメーター

authid 入力。このパラメーター値は SQL authid です。これは、その値が DB2 により大文字ストリングに変換され、末尾ブランクは付かないという意味です。 DB2 は常に、authid パラメーターに対して非ヌル値を提供します。API は、他の入力パラメーターに関係なく、 authid が所属するグループの

リストを戻せなければなりません。これが判別できない場合は、短縮されたリストまたは空のリストを戻しても差し支えありません。

ユーザーが存在しない場合、この API は戻りコード DB2SEC_PLUGIN_BADUSER を戻す必要があります。 authid には関連するグループがなくても差し支えないため、 DB2 は存在しないユーザーのケースをエラーとして扱いません。これには、db2secGetAuthids API がオペレーティング・システムに存在しない authid を戻す可能性があります。この authid にはグループが関連付けられていませんが、それでもこれには直接特権を割り当てることができます。

API がその authid を使用するだけでは完全なグループのリストを戻せない場合、グループ・サポートに関連した特定の SQL 関数になんらかの制限が生じる可能性があります。考えられる問題シナリオのリストについて詳しくは、このトピックの「使用上の注意」セクションを参照してください。

authidlen
入力。 authid パラメーター値のバイト単位の長さ。 DB2 データベース・マネージャーは常に、authidlen パラメーターに対してゼロ以外の値を提供します。

userid 入力。これは authid に対応するユーザー ID です。非接続のシナリオで、サーバー上でこの API が呼び出されたときは、DB2 はこのパラメーターに値を入れません。

useridlen
入力。 userid パラメーター値のバイト単位の長さ。

usernamespace
入力。取得されたユーザー ID が属するネーム・スペース。ユーザー ID が使用できない場合、DB2 データベース・マネージャーはこのパラメーターに値を入れません。

usernamespacelen
入力。 usernamespace パラメーター値のバイト単位の長さ。

usernamespacetype
入力。ネーム・スペースのタイプ。 usernamespacetype パラメーターの有効な値 (db2secPlugin.h で定義されている) は以下のとおりです。

- DB2SEC_NAMESPACE_SAM_COMPATIBLE は domain¥myname などのユーザー名スタイルに対応します。
- DB2SEC_NAMESPACE_USER_PRINCIPAL は myname@domain.ibm.com などのユーザー名スタイルに対応します。

現在のところ、DB2 データベース・システムは値 DB2SEC_NAMESPACE_SAM_COMPATIBLE しかサポートしていません。ユーザー ID がない場合、usernamespacetype パラメーターの値は DB2SEC_USER_NAMESPACE_UNDEFINED (db2secPlugin.h で定義された) に設定されます。

dbname
入力。接続先のデータベースの名前。このパラメーターは、非接続シナリオでは NULL にすることができます。

dbnamelen
入力。 dbname パラメーター値のバイト単位の長さ。非接続シナリオでは、dbname パラメーターが NULL の場合、このパラメーターは 0 に設定されます。

token 入力。認証プラグインによって提供されるデータを指すポインター。これは DB2 では使用されません。これを使用することにより、プラグイン作成者はユーザーおよびグループ情報を調整することができるようになります。このパラメーターは、必ずしもすべての事例で使用できない可能性があり (例えば、非接続シナリオで)、その場合のパラメーターの値は NULL になります。使用されている認証プラグインが GSS-API ベースの場合、このトークンには GSS-API コンテキスト・ハンドル (gss_ctx_id_t) が設定されます。

tokentype
入力。認証プラグインによって提供されるデータのタイプを示します。使用されている認証プラグインが GSS-API ベースの場合、このトークンには GSS-API コンテキスト・ハンドル (gss_ctx_id_t) が設定されます。使用されている認証プラグインがユーザー ID/パスワード・ベースの場合、これは汎用タイプになります。 tokentype パラメーターの有効な値 (db2secPlugin.h で定義されている) は以下のとおりです。

- DB2SEC_GENERIC: トークンがユーザー ID/パスワード・ベースのプラグインからのものであることを示します。
- DB2SEC_GSSAPI_CTX_HANDLE: トークンが GSS-API (Kerberos を含む) ベースのプラグインからのものであることを示します。

location
入力。 DB2 がクライアント・サイドとサーバー・サイドのどちらでこの API を呼び出すかを示します。 location パラメーターの有効な値 (db2secPlugin.h で定義されている) は以下のとおりです。

- DB2SEC_SERVER_SIDE: API はデータベース・サーバーで呼び出されます。
- DB2SEC_CLIENT_SIDE: API はクライアントで呼び出されます。

authpluginname
入力。トークンのデータを提供した認証プラグインの名前。db2secGetGroupsForUser API は、正しいグループ・メンバーシップを判別するためにこの情報を使用することがあります。 authid が認証されない場合 (例えば、authid が現行接続ユーザーと一致しない場合) には、このパラメーターには DB2 によって値が入力されないことがあります。

authpluginnamelen
入力。 authpluginname パラメーター値のバイト単位の長さ。

grouplist
出力。ユーザーが所属するグループのリスト。グループのリストは、連結された varchar (varchar とは、最初のバイトが後続のバイトの数を示す文字配列です) が含まれている、プラグインによって割り振られたメモリーのセクションを指すポインターとして戻されなければなりません。長さは unsigned char (1 バイト) であり、このためグループ名の最大長は 255 文字までに制限されます。例えば、「¥006GROUP1¥007MYGROUP¥008MYGROUP3」などです。各グループ名は、有効な DB2 authid でなければなりません。この配列のメモリーは、プラグインによって割り振られる必要があります。した

がって、プラグインは、 DB2 がメモリーを解放するために呼び出す db2secFreeGroupListMemory API などの API を備えている必要があります。

numgroups
出力。grouplist パラメーターに含まれるグループの数。

errormsg
出力。db2secGetGroupsForUser API が正常に実行されない場合にこのパラメーターで戻されることのある、プラグインによって割り振られた ASCII エラー・メッセージ・ストリングのアドレスを指すポインター。

errormsglen
出力。errormsg パラメーターのエラー・メッセージ・ストリングのバイト単位の長さを示す整数を指すポインター。

使用上の注意

以下は、この API によって DB2 に不完全なグループのリストが戻された場合に、問題が生じる可能性のあるシナリオのリストです。

- DYNAMICRULES BIND (あるいは、パッケージがスタンドアロン・アプリケーションとして実行している場合は、 DEFINEDBIND または INVOKEDBIND) が指定された組み込み SQL アプリケーション。 DB2 は SYSADM メンバーシップをチェックします。そして、アプリケーションが、 SYSADM のメンバーであることによって付与される暗黙的な DBADM 権限に依存している場合、このアプリケーションは失敗します。
- CREATE SCHEMA ステートメントで代替許可が提供される。 CREATE SCHEMA ステートメント内にネストされた CREATE ステートメントがある場合、 AUTHORIZATION NAME パラメーターに対してグループ検索が実行されます。
- DYNAMICRULES DEFINERUN/DEFINEBIND が指定された組み込み SQL アプリケーションがあり、そのパッケージがルーチン・コンテキストで実行している。 DB2 はルーチン定義者の SYSADM メンバーシップをチェックします。そして、アプリケーションが、 SYSADM のメンバーであることによって付与される暗黙的な DBADM 権限に依存している場合、このアプリケーションは失敗します。
- MPP 環境での jar ファイルの処理。 MPP 環境では、jar 処理要求が、セッション authid とともにコーディネーター・ノードから送信されます。カタログ・ノードは要求を受信すると、セッション authid (jar 処理要求を実行するユーザー) の特権に基づいて jar ファイルを処理します。
  - jar ファイルのインストール。セッション authid は、SYSADM、DBADM、または CREATEIN のいずれかの (jar スキーマに対する暗黙的または明示的な) 権限を有している必要があります。セッション authid の含まれるグループに対しては上記の権限が付与されているが、セッション authid に明示的には付与されていない場合や、データベース構成パラメーターによって定義されたグループのメンバーシップによって SYSADM メンバーシップが判別されたために、 SYSADM のみが保持されている場合は、操作は失敗します。
  - jar ファイルの除去。セッション authid は、SYSADM、DBADM、または DROPIN のいずれかの (jar スキーマに対する暗黙的または明示的な) 権限を有

しているか、 jar ファイルの定義者である必要があります。セッション authid の含まれるグループに対しては上記の権限が付与されているが、セッション authid に明示的には付与されておらず、セッション authid が Jar ファイルの定義者でもない場合や、データベース構成パラメーターによって定義されたグループのメンバーシップによって SYSADM メンバーシップが判別されたために、 SYSADM のみが保持されている場合は、操作は失敗します。

  - jar ファイルの置き換え。これは、jar ファイルを除去した後に、jar ファイルをインストールするのと同じことです。上記の両方が当てはまります。

- ビューの再生成。これは ALTER TABLE、ALTER COLUMN、SET DATA TYPE VARCHAR/VARGRAPHIC ステートメントによって、またはマイグレーションの際に起動されます。 DB2 データベース・マネージャーはビュー定義者の SYSADM メンバーシップをチェックします。アプリケーションが、 SYSADM グループのメンバーであることによって付与される暗黙的な DBADM 権限に依存している場合、このアプリケーションは失敗します。
- SET SESSION_USER ステートメントが発行される場合。その後の DB2 操作は、このステートメントで指定された authid のコンテキストの下で実行されます。必要な特権が SESSION_USER のグループのいずれかによって所有されているものの、 SESSION_USER authid に明示的に付与されてはいない場合、それらの操作は失敗します。

db2secGroupPluginInit API - グループ・プラグインの初期化

プラグインのロードの直後に DB2 データベース・マネージャーが呼び出す、グループ検索プラグイン用の初期化 API。

API とデータ構造構文

```
SQL_API_RC SQL_API_FN db2secGroupPluginInit
                              ( db2int32 version,
                                void *group_fns,
                                db2secLogMessage *logMessage_fn,
                                char     **errormsg,
                                db2int32 *errormsglen );
```

db2secGroupPluginInit API パラメーター

version

入力。そのプラグインをロードするインスタンスによってサポートされる API の最上位バージョン。値 DB2SEC_API_VERSION (db2secPlugin.h 内) には、DB2 データベース・マネージャーが現在サポートしている API の最新のバージョン番号が含まれます。

group_fns

出力。db2secGroupFunctions_<version_number> (group_functions_<version_number> としても知られる) 構造を指すポインター。db2secGroupFunctions_<version_number> 構造には、グループ検索プラグイン用にインプリメントされた API を指すポインターが含まれます。将来、これらの API には異なるバージョンが存在する可能性があるので (例えば、db2secGroupFunctions_<version_number>)、group_fns パラメーターは、プラグインがインプリメントしているバージョンに対応する db2secGroupFunctions_<version_number> 構造を指すポインターとしてキャストされます。 group_functions_<version_number> 構造の最初のパラメーター

は、プラグインがインプリメントしている API のバージョンを DB2 に知らせます。注: DB2 のバージョンが、プラグインがインプリメントしている API のバージョンと同じかそれより大きい場合に限り、キャストが行われます。バージョン番号は、プラグインがインプリメントしている API のバージョンを表しており、pluginType は DB2SEC_PLUGIN_TYPE_GROUP に設定されていなければなりません。

logMessage_fn

入力。 DB2 データベース・システムによってインプリメントされる db2secLogMessage API を指すポインター。 db2secGroupPluginInit API は、db2secLogMessage API を呼び出して、デバッグまたは通知の目的でメッセージを db2diag.log に記録することができます。 db2secLogMessage API の最初のパラメーター (level) は、db2diag.log ファイルに記録される診断エラーのタイプを指定し、最後の 2 つのパラメーターはそれぞれメッセージ・ストリングとその長さです。 (db2secPlugin.h で定義された) dbesecLogMessage API の最初のパラメーターの有効な値は以下のとおりです。

- DB2SEC_LOG_NONE: (0) ロギングなし
- DB2SEC_LOG_CRITICAL: (1) 重大エラーを検出した
- DB2SEC_LOG_ERROR: (2) エラーを検出した
- DB2SEC_LOG_WARNING: (3) 警告
- DB2SEC_LOG_INFO: (4) 通知

メッセージ・テキストが diag.log に表示されるのは、db2secLogMessage API の 'level' パラメーターの値が diaglevel データベース・マネージャー構成パラメーターの値以下である場合だけです。そのため、例えば DB2SEC_LOG_INFO 値を使用する場合、メッセージ・テキストは diaglevel データベース・マネージャー構成パラメーターに 4 が設定されている場合にのみ db2diag.log に表示されます。

errormsg

出力。プラグインによって割り振られた ASCII エラー・メッセージ・ストリングのアドレスを指すポインター。db2secGroupPluginInit API が正常に実行されない場合にこのパラメーターに戻されることがあります。

errormsglen

出力。errormsg パラメーターのエラー・メッセージ・ストリングのバイト単位の長さを示す整数を指すポインター。

db2secPluginTerm - グループ・プラグイン・リソースのクリーンアップ

グループ検索プラグインによって使用されるリソースを解放します。

この API は、DB2 データベース・マネージャーがグループ検索プラグインをアンロードする直前に呼び出されます。これは、プラグイン・ライブラリーが保持しているリソースの適切なクリーンアップを実行する、という方法でインプリメントされる必要があります。例えば、プラグインによって割り振られたメモリーを解放し、まだオープンしているファイルをクローズし、ネットワーク接続をクローズし

ます。これらのリソースを解放するためにその記録を保持することは、プラグインが行います。この API は Windows プラットフォームでは呼び出されません。

API とデータ構造構文

```
SQL_API_RC ( SQL_API_FN *db2secPluginTerm)
                      ( char      **errormsg,
                        db2int32 *errormsglen );
```

db2secPluginTerm API parameters

errormsg
出力。プラグインによって割り振られた ASCII エラー・メッセージ・ストリングのアドレスを指すポインター。db2secPluginTerm API が正常に実行されない場合にこのパラメーターに戻されることがあります。

errormsglen
出力。errormsg パラメーターのエラー・メッセージ・ストリングのバイト単位の長さを示す整数を指すポインター。

ユーザー ID/パスワード認証プラグインの API

ユーザー ID/パスワード・プラグイン・モジュール用には、以下のクライアント・サイド API をインプリメントする必要があります。

- db2secClientAuthPluginInit

  注: db2secClientAuthPluginInit API は、以下のプロトタイプを持つ API を指すポインター *logMessage_fn を入力としてとります。

  ```
  SQL_API_RC (SQL_API_FN db2secLogMessage)
  (
  db2int32 level,
  void    *data,
  db2int32 length
  );
  ```

  db2secLogMessage API により、プラグインはデバッグまたは通知の目的で、メッセージを db2diag.log に記録することができます。この API は DB2 データベース・システムによって提供されるため、インプリメントする必要はありません。
- db2secClientAuthPluginTerm
- db2secGenerateInitialCred (gssapi 専用)
- db2secRemapUserid (オプション)
- db2secGetDefaultLoginContext
- db2secValidatePassword
- db2secProcessServerPrincipalName (これは GSS-API 専用)
- db2secFreeToken (DLL で保持されているメモリーを解放するための関数)
- db2secFreeErrormsg
- db2secFreeInitInfo
- 外部で解決できなければならない唯一の API は、db2secClientAuthPluginInit です。この API は void * パラメーターをとり、それは以下のいずれかにキャストする必要があります。

```
typedef struct db2secUseridPasswordClientAuthFunctions_1
{
db2int32 version;
db2int32 plugintype;

SQL_API_RC (SQL_API_FN * db2secGetDefaultLoginContext)
(
char        authid[DB2SEC_MAX_AUTHID_LENGTH],
db2int32   *authidlen,
char        userid[DB2SEC_MAX_USERID_LENGTH],
db2int32   *useridlen,
db2int32    useridtype,
char        usernamespace[DB2SEC_MAX_USERNAMESPACE_LENGTH],
db2int32   *usernamespacelen,
db2int32   *usernamespacetype,
const char *dbname,
db2int32    dbnamelen,
void      **token,
char      **errormsg,
db2int32   *errormsglen
);
/* Optional */
SQL_API_RC (SQL_API_FN * db2secRemapUserid)
(
char        userid[DB2SEC_MAX_USERID_LENGTH],
db2int32   *useridlen,
char        usernamespace[DB2SEC_MAX_USERNAMESPACE_LENGTH],
db2int32   *usernamespacelen,
db2int32   *usernamespacetype,
char        password[DB2SEC_MAX_PASSWORD_LENGTH],
db2int32   *passwordlen,
char        newpassword[DB2SEC_MAX_PASSWORD_LENGTH],
db2int32   *newpasswordlen,
const char *dbname,
db2int32    dbnamelen,
char      **errormsg,
db2int32   *errormsglen
);

SQL_API_RC (SQL_API_FN * db2secValidatePassword)
(
const char *userid,
db2int32    useridlen,
const char *usernamespace,
db2int32    usernamespacelen,
db2int32    usernamespacetype,
const char *password,
db2int32    passwordlen,
const char *newpassword,
db2int32    newpasswordlen,
const char *dbname,
db2int32    dbnamelen,
db2Uint32   connection_details,
void      **token,
char      **errormsg,
db2int32   *errormsglen
);

SQL_API_RC (SQL_API_FN * db2secFreeToken)
(
void     **token,
char     **errormsg,
db2int32  *errormsglen
);

SQL_API_RC (SQL_API_FN * db2secFreeErrormsg)
(
```

```
char *errormsg
);

SQL_API_RC (SQL_API_FN * db2secClientAuthPluginTerm)
(
char     **errormsg,
db2int32  *errormsglen
);
}
```

または

```
typedef struct db2secGssapiClientAuthFunctions_1
{
db2int32 version;
db2int32 plugintype;

SQL_API_RC (SQL_API_FN * db2secGetDefaultLoginContext)
(
char        authid[DB2SEC_MAX_AUTHID_LENGTH],
db2int32   *authidlen,
char        userid[DB2SEC_MAX_USERID_LENGTH],
db2int32   *useridlen,
db2int32    useridtype,
char        usernamespace[DB2SEC_MAX_USERNAMESPACE_LENGTH],
db2int32   *usernamespacelen,
db2int32   *usernamespacetype,
const char *dbname,
db2int32    dbnamelen,
void      **token,
char      **errormsg,
db2int32   *errormsglen
);

SQL_API_RC (SQL_API_FN * db2secProcessServerPrincipalName)
(
const void *data,
gss_name_t *gssName,
char      **errormsg,
db2int32   *errormsglen
);

SQL_API_RC (SQL_API_FN * db2secGenerateInitialCred)
(
const char    *userid,
db2int32       useridlen,
const char    *usernamespace,
db2int32       usernamespacelen,
db2int32       usernamespacetype,
const char    *password,
db2int32       passwordlen,
const char    *newpassword,
db2int32       newpasswordlen,
const char    *dbname,
db2int32       dbnamelen,
gss_cred_id_t *pGSSCredHandle,
void         **initInfo,
char         **errormsg,
db2int32      *errormsglen
);

SQL_API_RC (SQL_API_FN * db2secFreeToken)
(
void      *token,
char     **errormsg,
db2int32  *errormsglen
);
```

```
SQL_API_RC (SQL_API_FN * db2secFreeErrormsg)
(
char *errormsg
);


SQL_API_RC (SQL_API_FN * db2secFreeInitInfo)
(
void      *initInfo,
char     **errormsg,
db2int32  *errormsglen
);

SQL_API_RC (SQL_API_FN * db2secClientAuthPluginTerm)
(
char     **errormsg,
db2int32  *errormsglen
);

/* GSS-API specific functions -- refer to db2secPlugin.h
   for parameter list*/

  OM_uint32 (SQL_API_FN * gss_init_sec_context )(<parameter list>);
  OM_uint32 (SQL_API_FN * gss_delete_sec_context )(<parameter list>);
  OM_uint32 (SQL_API_FN * gss_display_status )(<parameter list>);
  OM_uint32 (SQL_API_FN * gss_release_buffer )(<parameter list>);
  OM_uint32 (SQL_API_FN * gss_release_cred )(<parameter list>);
  OM_uint32 (SQL_API_FN * gss_release_name )(<parameter list>);
  }
```

ユーザー ID/パスワード・プラグインを作成する場合は、db2secUseridPasswordClientAuthFunctions_1 構造を使用する必要があります。GSS-API (Kerberos を含む) プラグインを作成する場合は、db2secGssapiClientAuthFunctions_1 構造を使用する必要があります。

ユーザー ID/パスワード・プラグイン・ライブラリー用には、以下のサーバー・サイド API をインプリメントする必要があります。

- db2secServerAuthPluginInit

  db2secServerAuthPluginInit API は、以下のプロトタイプを持つ、db2secLogMessage API を指すポインター *logMessage_fn、および db2secGetConDetails API を指すポインター *getConDetails_fn を入力としてとります。

```
SQL_API_RC (SQL_API_FN db2secLogMessage)
(
db2int32 level,
void    *data,
db2int32 length
);

SQL_API_RC (SQL_API_FN db2secGetConDetails)
(
db2int32    conDetailsVersion,
const void *pConDetails
);
```

  db2secLogMessage API により、プラグインはデバッグまたは通知の目的で、メッセージを db2diag.log に記録することができます。 db2secGetConDetails API により、プラグインは、データベース接続を持とうとしているクライアントに関する

詳細を取得することができます。 db2secLogMessage API と db2secGetConDetails API はどちらも DB2 データベース・システムによって提供されるので、インプリメントする必要はありません。同様に、db2secGetConDetails API は、その 2 番目のパラメーター pConDetails として、以下の構造の 1 つを指すポインターをとります。

db2sec_con_details_1:

```
typedef struct db2sec_con_details_1
{
  db2int32  clientProtocol;
  db2Uint32 clientIPAddress;
  db2Uint32 connect_info_bitmap;
  db2int32  dbnameLen;
  char      dbname[DB2SEC_MAX_DBNAME_LENGTH + 1];
} db2sec_con_details_1;
```

db2sec_con_details_2:

```
typedef struct db2sec_con_details_2
{
  db2int32  clientProtocol;     /* See SQL_PROTOCOL_ in sqlenv.h */
  db2Uint32 clientIPAddress;    /* Set if protocol is TCPIP4     */
  db2Uint32 connect_info_bitmap;
  db2int32  dbnameLen;
  char dbname[DB2SEC_MAX_DBNAME_LENGTH + 1];
  db2Uint32 clientIP6Address[4];/* Set if protocol is TCPIP6     */
} db2sec_con_details_2;
```

db2sec_con_details_3:

```
typedef struct db2sec_con_details_3
{
  db2int32 clientProtocol; /* See SQL_PROTOCOL_ in sqlenv.h */
  db2Uint32 clientIPAddress; /* Set if protocol is TCPIP4 */
  db2Uint32 connect_info_bitmap;
  db2int32 dbnameLen;
  char dbname[DB2SEC_MAX_DBNAME_LENGTH + 1];
  db2Uint32 clientIP6Address[4];/* Set if protocol is TCPIP6 */
  db2Uint32 clientPlatform; /* SQLM_PLATFORM_* from sqlmon.h */
  db2Uint32 _reserved[16];
} db2sec_con_details_3;
```

conDetailsVersion の考えられる値は、API のバージョンを表す DB2SEC_CON_DETAILS_VERSION_1、DB2SEC_CON_DETAILS_VERSION_2 および DB2SEC_CON_DETAILS_VERSION_3 です。

注: db2sec_con_details_1、db2sec_con_details_2 または db2sec_con_details_3 を使用しているときには、以下の事柄を考慮してください。

– db2sec_con_details_1 構造と DB2SEC_CON_DETAILS_VERSION_1 値を使用している既存のプラグインは、db2GetConDetails API を呼び出すと、バージョン 8.2 で行っていたように作業を続けます。この API が IPv4 プラットフォームで呼び出される場合、クライアント IP アドレスが構造の clientIPAddress フィールドで戻されます。この API が IPv6 プラットフォームで呼び出される場合、値 0 が clientIPAddress フィールドで戻されます。クライアント IP アドレスを IPv6 プラットフォームで取り出すには、db2sec_con_details_2 構造と DB2SEC_CON_DETAILS_VERSION_2 値を使用するように、または db2sec_con_details_3 構造と DB2SEC_CON_DETAILS_VERSION_3 値を使用するようにセキュリティー・プラグイン・コードを変更する必要があります。

– 新規プラグインは db2sec_con_details_3 構造と DB2SEC_CON_DETAILS_VERSION_3 値を使用する必要があります。db2secGetConDetails API が IPv4 プラットフォームで呼び出される場合、クライアント IP アドレスが db2sec_con_details_3 構造の clientIPAddress フィールドで戻され、この API が IPv6 プラットフォームで呼び出される場合、クライアント IP アドレスが db2sec_con_details_3 構造の clientIP6Address フィールドで戻されます。接続詳細構造の *clientProtocol* フィールドは、SQL_PROTOCOL_TCPIP (IPv4 で v1 の構造を持つ)、SQL_PROTOCOL_TCPIP4 (IPv4 で v2 または v3 の構造を持つ)、または SQL_PROTOCOL_TCPIP6 (IPv6で v2 の構造を持つ) のいずれかに設定されます。
– 構造 db2sec_con_details_3 は、SQLM_PLATFORM_AIX のような sqlmon.h で定義されるプラットフォーム・タイプ定数を使用するクライアント・プラットフォーム・タイプ (通信層でレポートされる) を特定する追加フィールド (*clientPlatform*) を含んでいる点を除けば、構造 db2sec_con_details_2 と同じです。

- db2secServerAuthPluginTerm
- db2secValidatePassword
- db2secGetAuthIDs
- db2secDoesAuthIDExist
- db2secFreeToken
- db2secFreeErrormsg
- 外部で解決できなければならない唯一の API は、db2secServerAuthPluginInit です。この API は void * パラメーターをとり、それは以下のいずれかにキャストする必要があります。

```
typedef struct db2secUseridPasswordServerAuthFunctions_1
{
db2int32 version;
db2int32 plugintype;

  /* parameter lists left blank for readability
    see above for parameters */
SQL_API_RC (SQL_API_FN * db2secValidatePassword)(<parameter list>);
SQL_API_RC (SQL_API_FN * db2secGetAuthIDs)(<parameter list);
SQL_API_RC (SQL_API_FN * db2secDoesAuthIDExist)(<parameter list>);
SQL_API_RC (SQL_API_FN * db2secFreeToken)(<parameter list>);
SQL_API_RC (SQL_API_FN * db2secFreeErrormsg)(<parameter list>);
SQL_API_RC (SQL_API_FN * db2secServerAuthPluginTerm)();
} userid_password_server_auth_functions;
```

または

```
typedef struct db2secGssapiServerAuthFunctions_1
{
db2int32 version;
db2int32 plugintype;
gss_buffer_desc serverPrincipalName;
gss_cred_id_t ServerCredHandle;
SQL_API_RC (SQL_API_FN * db2secGetAuthIDs)(<parameter list);
SQL_API_RC (SQL_API_FN * db2secDoesAuthIDExist)(<parameter list>);
SQL_API_RC (SQL_API_FN * db2secFreeErrormsg)(<parameter list>);
SQL_API_RC (SQL_API_FN * db2secServerAuthPluginTerm)();

/* GSS-API specific functions
```

```
refer to db2secPlugin.h for parameter list*/
OM_uint32 (SQL_API_FN * gss_accept_sec_context )(<parameter list>);
OM_uint32 (SQL_API_FN * gss_display_name )(<parameter list>);
OM_uint32 (SQL_API_FN * gss_delete_sec_context )(<parameter list>);
OM_uint32 (SQL_API_FN * gss_display_status )(<parameter list>);
OM_uint32 (SQL_API_FN * gss_release_buffer )(<parameter list>);
OM_uint32 (SQL_API_FN * gss_release_cred )(<parameter list>);
OM_uint32 (SQL_API_FN * gss_release_name )(<parameter list>);

} gssapi_server_auth_functions;
```

ユーザー ID/パスワード・プラグインを作成する場合は、db2secUseridPasswordServerAuthFunctions_1 構造を使用する必要があります。GSS-API (Kerberos を含む) プラグインを作成する場合は、db2secGssapiServerAuthFunctions_1 構造を使用する必要があります。

db2secClientAuthPluginInit API - クライアント認証プラグインの初期化

プラグインのロードの直後に DB2 データベース・マネージャーが呼び出す、クライアント認証プラグイン用の初期化 API。

API とデータ構造構文

```
SQL_API_RC SQL_API_FN db2secClientAuthPluginInit
                            ( db2int32 version,
                              void *client_fns,
                              db2secLogMessage *logMessage_fn,
                              char    **errormsg,
                              db2int32 *errormsglen );
```

db2secClientAuthPluginInit API パラメーター

version

入力。 DB2 データベース・マネージャーが現在サポートしている API の最大のバージョン番号。 DB2SEC_API_VERSION 値 (db2secPlugin.h 内) には、DB2 が現在サポートしている API の最新のバージョン番号が含まれます。

client_fns

出力。GSS-API 認証が使用される場合は、db2secGssapiClientAuthFunctions_<version_number> 構造 (gssapi_client_auth_functions_<version_number> としても知られる) のために DB2 データベース・マネージャーによって提供されたメモリーを指すポインター。ユーザー ID/パスワード認証が使用される場合は、db2secUseridPasswordClientAuthFunctions_<version_number> 構造 (userid_password_client_auth_functions_<version_number> としても知られる) のために DB2 データベース・マネージャーによって提供されたメモリーを指すポインター。 db2secGssapiClientAuthFunctions_<version_number> 構造は、GSS-API 認証プラグイン用にインプリメントされた API を指すポインターを含んでおり、db2secUseridPasswordClientAuthFunctions_<version_number> 構造は、ユーザー ID/パスワード認証プラグイン用にインプリメントされた API を指すポインターを含んでいます。 DB2 の将来のバージョンでは、異なるバージョンの API が存在している可能性があるので、client_fns パラメーターは、プ

ラグインがインプリメントしているバージョンに対応する gssapi_client_auth_functions_<version_number> 構造を指すポインターとしてキャストします。

gssapi_client_auth_functions_<version_number> 構造または userid_password_client_auth_functions_<version_number> 構造の最初のパラメーターは、プラグインがインプリメントしている API のバージョンを DB2 データベース・マネージャーに知らせます。

注: DB2 のバージョンが、プラグインがインプリメントしている API のバージョンと同じかそれより大きい場合に限り、キャストが行われます。gssapi_server_auth_functions_<version_number> または userid_password_server_auth_functions_<version_number> 構造内では、plugintype パスワードを DB2SEC_PLUGIN_TYPE_USERID_PASSWORD、DB2SEC_PLUGIN_TYPE_GSSAPI、または DB2SEC_PLUGIN_TYPE_KERBEROS のいずれかに設定する必要があります。将来のバージョンの API では、他の値も定義される可能性があります。

logMessage_fn

入力。 DB2 データベース・マネージャーによってインプリメントされる db2secLogMessage API を指すポインター。 db2secClientAuthPluginInit API は、db2secLogMessage API を呼び出して、デバッグまたは通知の目的でメッセージを db2diag.log に記録することができます。 db2secLogMessage API の最初のパラメーター (level) は、db2diag.log ファイルに記録される診断エラーのタイプを指定し、最後の 2 つのパラメーターはそれぞれメッセージ・ストリングとその長さです。 (db2secPlugin.h で定義された) dbesecLogMessage API の最初のパラメーターの有効な値は以下のとおりです。

- DB2SEC_LOG_NONE (0) ロギングなし
- DB2SEC_LOG_CRITICAL (1) 重大エラーを検出した
- DB2SEC_LOG_ERROR (2) エラーを検出した
- DB2SEC_LOG_WARNING (3) 警告
- DB2SEC_LOG_INFO (4) 通知

メッセージ・テキストが db2diag.log に表示されるのは、db2secLogMessage API の 'level' パラメーターの値が diaglevel データベース・マネージャー構成パラメーターの値以下である場合だけです。例えば DB2SEC_LOG_INFO 値を使用する場合、メッセージ・テキストは diaglevel データベース・マネージャー構成パラメーターに 4 が設定されている場合にのみ db2diag.log に表示されます。

errormsg

出力。プラグインによって割り振られた ASCII エラー・メッセージ・ストリングのアドレスを指すポインター。db2secClientAuthPluginInit API が正常に実行されない場合にこのパラメーターに戻されることがあります。

errormsglen

出力。errormsg パラメーターのエラー・メッセージ・ストリングのバイト単位の長さを示す整数を指すポインター。

db2secClientAuthPluginTerm API - クライアント認証プラグイン・リソースのクリーンアップ

クライアント認証プラグインによって使用されるリソースを解放します。

この API は、DB2 データベース・マネージャーがクライアント認証プラグインをアンロードする直前に呼び出します。これは、プラグイン・ライブラリーが保持しているリソースの適切なクリーンアップを実行する、という方法でインプリメントされる必要があります。例えば、プラグインによって割り振られたメモリーを解放し、まだオープンしているファイルをクローズし、ネットワーク接続をクローズします。これらのリソースを解放するためにその記録を保持することは、プラグインが行います。この API は Windows プラットフォームでは呼び出されません。

API とデータ構造構文

```
SQL_API_RC ( SQL_API_FN *db2secClientAuthPluginTerm)
                      ( char     **errormsg,
                        db2int32 *errormsglen);
```

db2secClientAuthPluginTerm API パラメーター

errormsg
出力。db2secClientAuthPluginTerm API が正常に実行されない場合にこのパラメーターで戻されることのある、プラグインによって割り振られた ASCII エラー・メッセージ・ストリングのアドレスを指すポインター。

errormsglen
出力。errormsg パラメーターのエラー・メッセージ・ストリングのバイト単位の長さを示す整数を指すポインター。

db2secDoesAuthIDExist - 認証 ID の存在の検査

authid が個々のユーザーを表しているかどうか (例えば、この API がこの authid を外部ユーザーにマップできるかどうか) を判別します。

この API は、これが正常 (authid が有効) な場合は値 DB2SEC_PLUGIN_OK を、無効な場合は DB2SEC_PLUGIN_INVALID_USERORGROUP を、authid の存在を判別できない場合は DB2SEC_PLUGIN_USERSTATUSNOTKNOWN を戻す必要があります。

API とデータ構造構文

```
SQL_API_RC ( SQL_API_FN *db2secDoesAuthIDExist)
                      ( const char *authid,
                        db2int32 authidlen,
                        char       **errormsg,
                        db2int32 *errormsglen );
```

db2secDoesAuthIDExist API パラメーター

authid 入力。検証する authid。これは、末尾ブランクなしの大文字になります。

authidlen
入力。 authid パラメーター値のバイト単位の長さ。

errormsg
出力。db2secDoesAuthIDExist API が正常に実行されない場合にこのパラメ

ーターで戻されることのある、プラグインによって割り振られた ASCII エラー・メッセージ・ストリングのアドレスを指すポインター。

errormsglen
出力。errormsg パラメーターのエラー・メッセージ・ストリングの長さを示す整数を指すポインター。

db2secFreeInitInfo API - db2secGenerateInitialCred が保持しているリソースのクリーンアップ

db2secGenerateInitialCred API によって割り振られたすべてのリソースを解放します。これには、例えば、基礎メカニズム・コンテキストのハンドルや、 GSS-API 証明書キャッシュ用に作成された証明書キャッシュが含まれます。

API とデータ構造構文

```
SQL_API_RC ( SQL_API_FN *db2secFreeInitInfo)
                      ( void *initinfo,
                        char     **errormsg,
                        db2int32 *errormsglen);
```

db2secFreeInitInfo API パラメーター

initinfo
入力。 DB2 データベース・マネージャーに認識されていないデータを指すポインター。プラグインはこのメモリーを使用して、証明書ハンドルの生成プロセスで割り振られたリソースのリストを保守できます。これらのリソースは、この API を呼び出すことによって解放されます。

errormsg
出力。プラグインによって割り振られた ASCII エラー・メッセージ・ストリングのアドレスを指すポインター。db2secFreeInitInfo API が正常に実行されない場合にこのパラメーターに戻されることがあります。

errormsglen
出力。errormsg パラメーターのエラー・メッセージ・ストリングのバイト単位の長さを示す整数を指すポインター。

db2secFreeToken API - トークンが保持しているメモリーの解放

トークンによって保持されたメモリーを解放します。この API は、DB2 データベース・マネージャーが token パラメーターによって保持されているメモリーを必要としなくなったときに呼び出します。

API とデータ構造構文

```
SQL_API_RC ( SQL_API_FN *db2secFreeToken)
                      ( void *token,
                        char       **errormsg,
                        db2int32 *errormsglen );
```

db2secFreeToken API パラメーター

token 入力。解放されるメモリーを指すポインター。

errormsg
出力。プラグインによって割り振られた ASCII エラー・メッセージ・スト

リングのアドレスを指すポインター。db2secFreeToken API が正常に実行されない場合にこのパラメーターに戻されることがあります。

errormsglen
出力。errormsg パラメーターのエラー・メッセージ・ストリングのバイト単位の長さを示す整数を指すポインター。

db2secGenerateInitialCred API - 初期証明書の生成

渡されるユーザー ID およびパスワードを基に初期 GSS-API 証明書を取得します。 Kerberos の場合、これは発券許可証 (TGT) になります。pGSSCredHandle パラメーターに戻される証明書ハンドルは、 gss_init_sec_context API で使用されるハンドルであり、INITIATE 証明書か BOTH 証明書のいずれかでなければなりません。 db2secGenerateInitialCred API は、ユーザー ID、そしておそらくパスワードが指定されている場合にのみ呼び出されます。それ以外の場合、DB2 データベース・マネージャーは gss_init_sec_context API を呼び出すときに値 GSS_C_NO_CREDENTIAL を指定して、現行ログイン・コンテキストから取得されるデフォルト証明書が使用されることを示します。

API とデータ構造構文

```
SQL_API_RC ( SQL_API_FN *db2secGenerateInitialCred)
                       ( const char *userid,
                         db2int32 useridlen,
                         const char *usernamespace,
                         db2int32 usernamespacelen,
                         db2int32 usernamespacetype,
                         const char *password,
                         db2int32 passwordlen,
                         const char *newpassword,
                         db2int32 newpasswordlen,
                         const char *dbname,
                         db2int32 dbnamelen,
                         gss_cred_id_t *pGSSCredHandle,
                         void          **InitInfo,
                         char          **errormsg,
                         db2int32 *errormsglen );
```

db2secGenerateInitialCred API パラメーター

userid 入力。データベース・サーバー上でパスワードが検証されるユーザー ID。

useridlen
入力。 userid パラメーター値のバイト単位の長さ。

usernamespace
入力。取得されたユーザー ID が属するネーム・スペース。

usernamespacelen
入力。 usernamespace パラメーター値のバイト単位の長さ。

usernamespacetype
入力。ネーム・スペースのタイプ。

password
入力。検証されるパスワード。

passwordlen
入力。 password パラメーター値のバイト単位の長さ。

newpassword
入力。パスワードが変更される場合の新規パスワード。変更が要求されない場合、newpassword パラメーターは NULL に設定されます。これが非ヌルの場合、API は、旧パスワードを新規パスワードに設定する前に検証する必要があります。 API は、パスワードの変更要求を受け入れなくても構いませんが、受け入れない場合は、旧パスワードを検証せずに即時に戻り値 DB2SEC_PLUGIN_CHANGEPASSWORD_NOTSUPPORTED を戻す必要があります。

newpasswordlen
入力。 newpassword パラメーター値のバイト単位の長さ。

dbname
入力。接続先のデータベースの名前。この API はこのパラメーターを無視しても差し支えありません。あるいは、特定のデータベースへのアクセスを、有効なパスワードを特別に持つユーザーのみに限定する方針をとっている場合は、この関数は値 DB2SEC_PLUGIN_CONNECTION_DISALLOWED を戻すことができます。

dbnamelen
入力。 dbname パラメーター値のバイト単位の長さ。

pGSSCredHandle
出力。GSS-API 証明書ハンドルを指すポインター。

InitInfo
出力。DB2 に認識されていないデータを指すポインター。プラグインはこのメモリーを使用して、証明書ハンドルの生成プロセスで割り振られたリソースのリストを保守できます。 DB2 データベース・マネージャーは認証プロセスの最後に db2secFreeInitInfo API を呼び出し、その時点でこれらのリソースは解放されます。 db2secGenerateInitialCred API は、このようなリストを保守する必要がない場合は、NULL を戻す必要があります。

errormsg
出力。db2secGenerateInitialCred API が正常に実行されない場合にこのパラメーターで戻されることのある、プラグインによって割り振られた ASCII エラー・メッセージ・ストリングのアドレスを指すポインター。

注: この API では、戻り値が無効なユーザー ID またはパスワードを示している場合には、エラー・メッセージは作成されるべきではありません。エラー・メッセージは、API の中に、この API が正しく完了することを妨げる内部エラーがある場合にのみ戻される必要があります。

errormsglen
出力。errormsg パラメーターのエラー・メッセージ・ストリングのバイト単位の長さを示す整数を指すポインター。

db2secGetAuthIDs API - 認証 ID の取得

認証ユーザーの SQL authid を戻します。この API は、ユーザー ID/パスワードと GSS-API の両方の認証方式において、データベース接続時に呼び出されます。

API とデータ構造構文

```
SQL_API_RC ( SQL_API_FN *db2secGetAuthIDs)
                       ( const char *userid,
                         db2int32 useridlen,
                         const char *usernamespace,
                         db2int32 usernamespacelen,
                         db2int32 usernamespacetype,
                         const char *dbname,
                         db2int32 dbnamelen,
                         void      **token,
                         char SystemAuthID[DB2SEC_MAX_AUTHID_LENGTH],
                         db2int32 *SystemAuthIDlen,
                         char InitialSessionAuthID[DB2SEC_MAX_AUTHID_LENGTH],
                         db2int32 *InitialSessionAuthIDlen,
                         char username[DB2SEC_MAX_USERID_LENGTH],
                         db2int32 *usernamelen,
                         db2int32 *initsessionidtype,
                         char      **errormsg,
                         db2int32 *errormsglen );
```

db2secGetAuthIDs API パラメーター

userid 入力。認証ユーザー。 GSS-API 認証では、通常は使用されません。ただし、認証なしのユーザー切り替え操作を認めるトラステッド・コンテキストが定義されている場合は例外です。そのような場合は、ユーザーの切り替え要求で指定されているユーザー名がこのパラメーターで渡されます。

useridlen
入力。 userid パラメーター値のバイト単位の長さ。

usernamespace
入力。取得されたユーザー ID が属するネーム・スペース。

usernamespacelen
入力。 usernamespace パラメーター値のバイト単位の長さ。

usernamespacetype
入力。ネーム・スペース・タイプ値。現時点でサポートされる唯一のネーム・スペース・タイプ値は、DB2SEC_NAMESPACE_SAM_COMPATIBLE です (domain¥myname などのユーザー名スタイルに相当します)。

dbname
入力。接続先のデータベースの名前。 API はこれを無視しても差し支えありません。あるいはプラグインは、同一のユーザーが異なるデータベースに接続する際に別々の authid を戻すこともできます。このパラメーターは、NULL にすることができます。

dbnamelen
入力。 dbname パラメーター値のバイト単位の長さ。 dbname パラメーターが NULL の場合、このパラメーターは 0 に設定されます。

token 入力または出力。プラグインが db2secGetGroupsForUser API に渡すデータ。 GSS-API の場合、これはコンテキスト・ハンドル (gss_ctx_id_t) です。通常、トークンは入力のみのパラメーターであり、この値は db2secValidatePassword から取り込まれます。認証がクライアントで行われ、そのために db2secValidatePassword API が呼び出されない場合には、これは出力パラメーターにもなります。認証なしのユーザー切り替え操作を認めるトラステッド・コンテキストが定義されている環境では、

db2secGetAuthIDs API でこのトークン・パラメーターの NULL 値を受け付け、上記の userid 入力パラメーターと useridlen 入力パラメーターに基づいて、システム許可 ID を派生させることができるように設定しなければなりません。

SystemAuthID
出力。認証ユーザーの ID に対応するシステム許可。サイズは 255 バイトですが、DB2 データベース・マネージャーは現在最大 30 バイトまで使用します。

SystemAuthIDlen
出力。SystemAuthID パラメーター値のバイト単位の長さ。

InitialSessionAuthID
出力。この接続セッションに使用される authid。これは通常 SystemAuthID パラメーターと同じですが、SET SESSION AUTHORIZATION ステートメントを発行する場合などのある特定の場合には異なることがあります。サイズは 255 バイトですが、DB2 データベース・マネージャーは現在最大 30 バイトまで使用します。

InitialSessionAuthIDlen
出力。InitialSessionAuthID パラメーター値のバイト単位の長さ。

username
出力。認証ユーザーと authid に対応するユーザー名。これは監査のためにのみ使用され、CONNECT ステートメントの監査記録内の「ユーザー ID」フィールドに記録されます。 API が username パラメーターに記入していない場合、DB2 データベース・マネージャーは userid からそれをコピーします。

usernamelen
出力。username パラメーター値のバイト単位の長さ。

initsessionidtype
出力。InitialSessionAuthid パラメーターがロールか authid かを示すセッション authid タイプ。 API は、以下の値のいずれか (db2secPlugin.h で定義された) を戻さなければなりません。

- DB2SEC_ID_TYPE_AUTHID (0)
- DB2SEC_ID_TYPE_ROLE (1)

errormsg
出力。db2secGetAuthIDs API が正常に実行されない場合にこのパラメーターで戻されることのある、プラグインによって割り振られた ASCII エラー・メッセージ・ストリングのアドレスを指すポインター。

errormsglen
出力。errormsg パラメーターのエラー・メッセージ・ストリングのバイト単位の長さを示す整数を指すポインター。

db2secGetDefaultLoginContext API - デフォルト・ログイン・コンテキストの取得

デフォルト・ログイン・コンテキストに関連したユーザーを判別します。すなわち、ユーザー ID を明示的に指定しない (データベースに対する暗黙的な認証か、

ローカル許可) で DB2 コマンドを呼び出すユーザーの DB2 authid を判別します。この API は、authid とユーザー ID の両方を戻さなければなりません。

API とデータ構造構文

```
SQL_API_RC ( SQL_API_FN *db2secGetDefaultLoginContext)
                       ( char authid[DB2SEC_MAX_AUTHID_LENGTH],
                         db2int32 *authidlen,
                         char userid[DB2SEC_MAX_USERID_LENGTH],
                         db2int32 *useridlen,
                         db2int32 useridtype,
                         char usernamespace[DB2SEC_MAX_USERNAMESPACE_LENGTH],
                         db2int32 *usernamespacelen,
                         db2int32 *usernamespacetype,
                         const char *dbname,
                         db2int32 dbnamelen,
                         void      **token,
                         char      **errormsg,
                         db2int32 *errormsglen );
```

db2secGetDefaultLoginContext API パラメーター

authid 出力。authid が戻されるパラメーター。戻り値は DB2 authid の命名規則に準拠していなければなりません。そうでなければ、ユーザーは要求されたアクションの実行を許可されません。

authidlen
出力。authid パラメーター値のバイト単位の長さ。

userid 出力。デフォルト・ログイン・コンテキストに関連したユーザー ID を戻すパラメーター。

useridlen
出力。userid パラメーター値のバイト単位の長さ。

useridtype
入力。プロセスの実ユーザー ID、または有効ユーザー ID が指定されているかどうかを示します。 Windowsの場合、実ユーザー ID のみ存在します。 UNIX および Linux では、アプリケーションの `uid` ユーザー ID がプロセスを実行しているユーザーの ID と異なる場合、実ユーザー ID と有効ユーザー ID が異なることがあります。 userid パラメーターの有効な値 (db2secPlugin.h で定義されている) は以下のとおりです。

DB2SEC_PLUGIN_REAL_USER_NAME
実ユーザー ID が指定されていることを示します。

DB2SEC_PLUGIN_EFFECTIVE_USER_NAME
有効ユーザー ID が指定されていることを示します。

注: 一部のプラグイン・インプリメンテーションによっては、実ユーザー ID と有効ユーザー ID を区別しないものがあります。特に、DB2 許可 ID を設定するためにユーザーの UNIX または Linux の ID を使用しないプラグインは、この区別を無視しても支障がありません。

usernamespace
出力。ユーザー ID のネーム・スペース。

usernamespacelen

出力。usernamespace パラメーター値のバイト単位の長さ。 usernamespacetype パラメーターが値 DB2SEC_NAMESPACE_SAM_COMPATIBLE (db2secPlugin.h で定義されている) に設定されていなければならないという制限のもとでは、現在サポートされる最大長は 15 バイトになります。

usernamespacetype

出力。ネーム・スペース・タイプ値。現時点でサポートされる唯一のネーム・スペース・タイプ値、DB2SEC_NAMESPACE_SAM_COMPATIBLE です (domain¥myname などのユーザー名スタイルに相当します)。

dbname

入力。データベース接続のコンテキストでこの呼び出しが使用される場合に、接続先のデータベースの名前が入ります。ローカル許可アクションやインスタンス接続の場合、このパラメーターは NULL に設定されます。

dbnamelen

入力。 dbname パラメーター値のバイト単位の長さ。

token 出力。これはプラグインが、そのプラグインでの後の認証呼び出しや、またはグループ検索プラグインに渡す、プラグインによって割り振られたデータを指すポインターです。このデータの構造は、プラグイン作成者によって決定されます。

errormsg

出力。db2secGetDefaultLoginContext API が正常に実行されない場合にこのパラメーターで戻されることのある、プラグインによって割り振られた ASCII エラー・メッセージ・ストリングのアドレスを指すポインター。

errormsglen

出力。errormsg パラメーターのエラー・メッセージ・ストリングのバイト単位の長さを示す整数を指すポインター。

db2secProcessServerPrincipalName API - サーバーから戻されたサービス・プリンシパル名の処理

サーバーから戻されたサービス・プリンシパル名を処理し、gss_init_sec_context API で使用される gss_name_t 内部形式のプリンシパル名を戻します。 db2secProcessServerPrincipalName API は、Kerberos 認証の使用時に、データベース・ディレクトリーでカタログされたサービス・プリンシパル名も処理します。通常、この変換では gss_import_name API が使用されます。コンテキストが確立されると、gss_name_t オブジェクトは gss_release_name API の呼び出しによって解放されます。 db2secProcessServerPrincipalName API は、gssName パラメーターが有効な GSS 名を指していれば値 DB2SEC_PLUGIN_OK を戻します。プリンシパル名が無効な場合は DB2SEC_PLUGIN_BAD_PRINCIPAL_NAME エラー・コードが戻されます。

API とデータ構造構文

```
SQL_API_RC ( SQL_API_FN *db2secProcessServerPrincipalName)
                        ( const char *name,
                          db2int32 namelen,
```

```
                        gss_name_t *gssName,
                        char       **errormsg,
                        db2int32 *errormsglen );
```

db2secProcessServerPrincipalName API パラメーター

name 入力。 GSS_C_NT_USER_NAME 形式のサービス・プリンシパルのテキスト名 (例: service/host@REALM)。

namelen
入力。 name パラメーター値のバイト単位の長さ。

gssName
出力。GSS-API 内部形式の出力サービス・プリンシパル名を指すポインター。

errormsg
出力。プラグインによって割り振られた ASCII エラー・メッセージ・ストリングのアドレスを指すポインター。db2secProcessServerPrincipalName API が正常に実行されない場合にこのパラメーターに戻されることがあります。

errormsglen
出力。errormsg パラメーターのエラー・メッセージ・ストリングのバイト単位の長さを示す整数を指すポインター。

db2secRemapUserid API - ユーザー ID およびパスワードの再マップ

この API は DB2 データベース・マネージャーによってクライアント・サイドで呼び出され、特定のユーザー ID およびパスワード (そしておそらく新規パスワードおよび usernamespace) を、接続時に指定された値とは異なる値に再マップします。DB2 データベース・マネージャーは、接続時にユーザー ID およびパスワードが指定されている場合にのみ、この API を呼び出します。これは、プラグインがユーザー ID を自らユーザー ID/パスワードのペアに再マップすることを防止します。この API はオプションであり、セキュリティー・プラグインによって提供あるいはインプリメントされていなければ呼び出されません。

API とデータ構造構文

```
SQL_API_RC ( SQL_API_FN *db2secRemapUserid)
                       ( char userid[DB2SEC_MAX_USERID_LENGTH],
                         db2int32 *useridlen,
                         char usernamespace[DB2SEC_MAX_USERNAMESPACE_LENGTH],
                         db2int32 *usernamespacelen,
                         db2int32 *usernamespacetype,
                         char password[DB2SEC_MAX_PASSWORD_LENGTH],
                         db2int32 *passwordlen,
                         char newpasswd[DB2SEC_MAX_PASSWORD_LENGTH],
                         db2int32 *newpasswdlen,
                         const char *dbname,
                         db2int32 dbnamelen,
                         char      **errormsg,
                         db2int32 *errormsglen);
```

db2secRemapUserid API パラメーター

userid 入力または出力。再マップされるユーザー ID。入力ユーザー ID 値がある場合は、API は出力ユーザー ID 値を提供しなければならず、それは入力ユ

ーザー ID 値と同じか、あるいは異なる値になる可能性があります。入力ユーザー ID 値がない場合、API は出力ユーザー ID 値を戻すべきではありません。

useridlen
入力または出力。userid パラメーター値のバイト単位の長さ。

usernamespace
入力または出力。ユーザー ID のネーム・スペース。この値は、オプションで再マップすることができます。入力パラメーター値が指定されず、出力値が戻されている場合、usernamespace は CLIENT タイプ認証の場合にのみ DB2 データベース・マネージャーによって使用され、他の認証タイプでは無視されます。

usernamespacelen
入力または出力。usernamespace パラメーター値のバイト単位の長さ。usernamespacetype パラメーターが値 DB2SEC_NAMESPACE_SAM_COMPATIBLE (db2secPlugin.h で定義されている) に設定されていなければならないという制限のもとでは、現在サポートされる最大長は 15 バイトになります。

usernamespacetype
入力または出力。namespacetype の古い値と新しい値。現時点でサポートされる唯一のネーム・スペース・タイプ値は、DB2SEC_NAMESPACE_SAM_COMPATIBLE です (domain¥myname などのユーザー名スタイルに相当します)。

password
入力または出力。入力の場合、これは再マップ対象のパスワードになります。出力の場合、これは再マップ済みパスワードになります。このパラメーターで入力値が指定されている場合、API は入力値とは異なる出力値を戻すことができなければなりません。入力値が指定されていない場合、API は出力 password 値を戻してはなりません。

passwordlen
入力または出力。password パラメーター値のバイト単位の長さ。

newpasswd
入力または出力。入力の場合、これは設定される新規パスワードになります。出力の場合、これは確認済みの新規パスワードになります。

注: これは、DB2 データベース・マネージャーがクライアント上またはサーバー上のどちらか (認証データベース・マネージャー構成パラメーターの値に応じる) の db2secValidatePassword API の newpassword パラメーターに渡す新規パスワードです。新規パスワードが入力として渡された場合は、API は出力値を戻すことができなければなりませんが、出力値が別の新規パスワードである可能性もあります。入力として渡された新規パスワードがない場合、API は出力の新規パスワードを戻すべきではありません。

newpasswdlen
入力または出力。newpasswd パラメーター値のバイト単位の長さ。

dbname
入力。クライアントの接続先のデータベースの名前。

dbnamelen

入力。 dbname パラメーター値のバイト単位の長さ。

errormsg

出力。db2secRemapUserid API が正常に実行されない場合にこのパラメーターで戻されることのある、プラグインによって割り振られた ASCII エラー・メッセージ・ストリングのアドレスを指すポインター。

errormsglen

出力。errormsg パラメーターのエラー・メッセージ・ストリングのバイト単位の長さを示す整数を指すポインター。

db2secServerAuthPluginInit - サーバー認証プラグインの初期化

プラグインのロードの直後に DB2 データベース・マネージャーが呼び出す、サーバー認証プラグイン用の初期化 API。GSS-API の場合は、プラグインが、初期化時に gssapi_server_auth_functions 構造内部の serverPrincipalName パラメーターにサーバーのプリンシパル名を入れ、gssapi_server_auth_functions 構造内部の serverCredHandle パラメーターにサーバーの証明書ハンドルを提供します。プリンシパル名および証明書ハンドルを保持するために割り振られているメモリーを解放するのは、gss_release_name および gss_release_cred API を呼び出すことによって、db2secServerAuthPluginTerm API が行うべき事柄です。

API とデータ構造構文

```
SQL_API_RC SQL_API_FN db2secServerAuthPluginInit
                            ( db2int32 version,
                              void *server_fns,
                              db2secGetConDetails *getConDetails_fn,
                              db2secLogMessage *logMessage_fn,
                              char               **errormsg,
                              db2int32 *errormsglen );
```

db2secServerAuthPluginInit API パラメーター

version

入力。 DB2 データベース・マネージャーが現在サポートしている API の最大のバージョン番号。 DB2SEC_API_VERSION 値 (db2secPlugin.h 内) には、DB2 データベース・マネージャーが現在サポートしている API の最新のバージョン番号が含まれます。

server_fns

出力。GSS-API 認証が使用される場合は、db2secGssapiServerAuthFunctions_<version_number> 構造 (gssapi_server_auth_functions_<version_number> としても知られる) のために DB2 データベース・マネージャーによって提供されたメモリーを指すポインター。ユーザー ID/パスワード認証が使用される場合は、db2secUseridPasswordServerAuthFunctions_<version_number> 構造 (userid_password_server_auth_functions_<version_number> としても知られる) のために DB2 データベース・マネージャーによって提供されたメモリーを指すポインター。 db2secGssapiServerAuthFunctions_<version_number> 構造は、GSS-API 認証プラグイン用にインプリメントされた API を指すポインターを含んでおり、

db2secUseridPasswordServerAuthFunctions_<version_number> 構造は、ユーザー ID/パスワード認証プラグイン用にインプリメントされた API を指すポインターを含んでいます。

server_fns パラメーターは、プラグインがインプリメントしているバージョンに対応する gssapi_server_auth_functions_<version_number> 構造を指すポインターとしてキャストします。
gssapi_server_auth_functions_<version_number> 構造または
userid_password_server_auth_functions_<version_number> 構造の最初のパラメーターは、プラグインがインプリメントしている API のバージョンを DB2 データベース・マネージャーに知らせます。

注: DB2 のバージョンが、プラグインがインプリメントしている API のバージョンと同じかそれより大きい場合に限り、キャストが行われます。
gssapi_server_auth_functions_<version_number> または
userid_password_server_auth_functions_<version_number> 構造内では、
plugintype パスワードを DB2SEC_PLUGIN_TYPE_USERID_PASSWORD、
DB2SEC_PLUGIN_TYPE_GSSAPI、または
DB2SEC_PLUGIN_TYPE_KERBEROS のいずれかに設定する必要があります。将来のバージョンの API では、他の値も定義される可能性があります。

getConDetails_fn

入力。 DB2 によってインプリメントされる db2secGetConDetails API を指すポインター。 db2secServerAuthPluginInit API は、いずれかの他の認証 API で db2secGetConDetails API を呼び出して、データベース接続に関する詳細を取得することができます。これらの詳細には、接続に関連した通信メカニズム (TCP/IP の場合は IP アドレスなど) についての情報が含まれ、これは、プラグイン作成者が認証についての決定をする際に参照しなければならない可能性があります。例えば、プラグインは、特定のユーザーが特定の IP アドレスから接続しようとしているのでない場合、そのユーザーの接続を禁止できます。 db2secGetConDetails API の使用はオプションです。

データベース接続に関係しない状況で db2secGetConDetails API が呼び出された場合、これは値 DB2SEC_PLUGIN_NO_CON_DETAILS を戻し、それ以外の場合は、正常なら 0 を戻します。

db2secGetConDetails API は、db2sec_con_details_<version_number> 構造を指すポインターである pConDetails と、使用される db2sec_con_details 構造を示すバージョン番号である conDetailsVersion という 2 つの入力パラメーターをとります。可能な値は、db2sec_con_details1 が使用される場合は DB2SEC_CON_DETAILS_VERSION_1 で、db2sec_con_details2 が使用される場合は DB2SEC_CON_DETAILS_VERSION_2 です。使用するように推奨されているバージョン番号は DB2SEC_CON_DETAILS_VERSION_2 です。

正常に戻る場合、db2sec_con_details 構造 (db2sec_con_details1 または db2sec_con_details2) には以下の情報が含まれます。

- サーバーへの接続に使用されるプロトコル。プロトコル定義のリストは、ファイル sqlenv.h (include ディレクトリーにある) (SQL_PROTOCOL_*) にあります。この情報は clientProtocol パラメーターに書き込まれます。

- clientProtocol が SQL_PROTOCOL_TCPIP または SQL_PROTOCOL_TCPIP4 の場合、サーバーへのインバウンド接続の TCP/IP アドレス。この情報は clientIPAddress パラメーターに書き込まれます。
- クライアントが接続しようとしているデータベースの名前。これは、インスタンス接続の場合は設定されません。この情報は dbname および dbnameLen パラメーターに書き込まれます。
- db2secValidatePassword API の connection_details パラメーター内に記述されるのと同じ詳細を含む、接続情報のビットマップ。この情報は connect_info_bitmap パラメーターに書き込まれます。
- clientProtocol が SQL_PROTOCOL_TCPIP6 の場合、サーバーへのインバウンド接続の TCP/IP アドレス。この情報は clientIP6Address パラメーターに書き込まれ、db2secGetConDetails API 呼び出しに DB2SEC_CON_DETAILS_VERSION_2 が使用される場合にのみ使用できます。

logMessage_fn

入力。 DB2 データベース・マネージャーによってインプリメントされる db2secLogMessage API を指すポインター。 db2secClientAuthPluginInit API は、db2secLogMessage API を呼び出して、デバッグまたは通知の目的でメッセージを db2diag.log に記録することができます。 db2secLogMessage API の最初のパラメーター (level) は、db2diag.log ファイルに記録される診断エラーのタイプを指定し、最後の 2 つのパラメーターはそれぞれメッセージ・ストリングとその長さです。 (db2secPlugin.h で定義された) dbesecLogMessage API の最初のパラメーターの有効な値は以下のとおりです。

DB2SEC_LOG_NONE (0)
ロギングなし

DB2SEC_LOG_CRITICAL (1)
重大エラーが検出された

DB2SEC_LOG_ERROR (2)
エラーが検出された

DB2SEC_LOG_WARNING (3)
警告

DB2SEC_LOG_INFO (4)
通知

メッセージ・テキストが db2diag.log に表示されるのは、db2secLogMessage API の 'level' パラメーターの値が diaglevel データベース・マネージャー構成パラメーターの値以下である場合だけです。

そのため、例えば DB2SEC_LOG_INFO 値を使用する場合、メッセージ・テキストは diaglevel データベース・マネージャー構成パラメーターに 4 が設定されている場合にのみ db2diag.log に表示されます。

errormsg
出力。プラグインによって割り振られた ASCII エラー・メッセージ・スト

リングのアドレスを指すポインター。db2secServerAuthPluginInit API が正常に実行されない場合にこのパラメーターに戻されることがあります。

errormsglen
出力。errormsg パラメーターのエラー・メッセージ・ストリングのバイト単位の長さを示す整数を指すポインター。

db2secServerAuthPluginTerm API - サーバー認証プラグイン・リソースのクリーンアップ

サーバー認証プラグインによって使用されるリソースを解放します。この API は、DB2 データベース・マネージャーがサーバー認証プラグインをアンロードする直前に呼び出します。これは、プラグイン・ライブラリーが保持しているリソースの適切なクリーンアップを実行する、という方法でインプリメントされる必要があります。例えば、プラグインによって割り振られたメモリーを解放し、まだオープンしているファイルをクローズし、ネットワーク接続をクローズします。これらのリソースを解放するためにその記録を保持することは、プラグインが行います。この API は Windows プラットフォームでは呼び出されません。

API とデータ構造構文

```
SQL_API_RC ( SQL_API_FN *db2secServerAuthPluginTerm)
                      ( char     **errormsg,
                        db2int32 *errormsglen );
```

db2secServerAuthPluginTerm API パラメーター

errormsg
出力。db2secServerAuthPluginTerm API が正常に実行されない場合にこのパラメーターで戻されることのある、プラグインによって割り振られた ASCII エラー・メッセージ・ストリングのアドレスを指すポインター。

errormsglen
出力。errormsg パラメーターのエラー・メッセージ・ストリングのバイト単位の長さを示す整数を指すポインター。

db2secValidatePassword API - パスワードの検証

データベース接続の操作時に、ユーザー ID およびパスワードのスタイル認証を実行する方法について説明します。

注: API がクライアント・サイドで実行されている場合、API コードは、CONNECT ステートメントを実行するユーザーの特権で実行されます。この API は、認証構成パラメーターに CLIENT が設定されている場合にのみ、クライアント・サイドで呼び出されます。

API がサーバー・サイドで実行されている場合、API コードはインスタンス所有者の特権で実行されます。

認証に特別な特権 (UNIX 上のルート・レベルのシステム・アクセスなど) が必要な場合、プラグイン作成者は、上記の点を考慮する必要があります。

この API は、パスワードが正常な場合は値 DB2SEC_PLUGIN_OK (正常) を、また、パスワードが無効な場合は DB2SEC_PLUGIN_BADPWD などのエラー・コードを戻す必要があります。

API とデータ構造構文

```
SQL_API_RC ( SQL_API_FN *db2secValidatePassword)
                       ( const char *userid,
                         db2int32 useridlen,
                         const char *usernamespace,
                         db2int32 usernamespacelen,
                         db2int32 usernamespacetype,
                         const char *password,
                         db2int32 passwordlen,
                         const char *newpasswd,
                         db2int32 newpasswdlen,
                         const char *dbname,
                         db2int32 dbnamelen,
                         db2Uint32 connection_details,
                         void      **token,
                         char      **errormsg,
                         db2int32 *errormsglen );
```

db2secValidatePassword API パラメーター

userid 入力。パスワードが検証されるユーザー ID。

useridlen
入力。 userid パラメーター値のバイト単位の長さ。

usernamespace
入力。取得されたユーザー ID が属するネーム・スペース。

usernamespacelen
入力。 usernamespace パラメーター値のバイト単位の長さ。

usernamespacetype
入力。ネーム・スペースのタイプ。 usernamespacetype パラメーターの有効な値 (db2secPlugin.h で定義されている) は以下のとおりです。

- DB2SEC_NAMESPACE_SAM_COMPATIBLE は domain¥myname などのユーザー名スタイルに対応します。
- DB2SEC_NAMESPACE_USER_PRINCIPAL は myname@domain.ibm.com などのユーザー名スタイルに対応します。

現在のところ、DB2 データベース・システムは値 DB2SEC_NAMESPACE_SAM_COMPATIBLE しかサポートしていません。ユーザー ID がない場合、usernamespacetype パラメーターの値は DB2SEC_USER_NAMESPACE_UNDEFINED (db2secPlugin.h で定義された) に設定されます。

password
入力。検証されるパスワード。

passwordlen
入力。 password パラメーター値のバイト単位の長さ。

newpasswd
入力。パスワードが変更される場合の新規パスワード。変更が要求されない場合、このパラメーターは NULL に設定されます。このパラメーターが非

ヌルの場合、この API は、旧パスワードを新規パスワードに変更する前に検証する必要があります。 API は、パスワードの変更要求を受け入れなくても構いませんが、受け入れない場合は、旧パスワードを検証せずに即時に戻り値 DB2SEC_PLUGIN_CHANGEPASSWORD_NOTSUPPORTED を戻す必要があります。

newpasswdlen
入力。 newpasswd パラメーター値のバイト単位の長さ。

dbname
入力。接続先のデータベースの名前。この API は dbname パラメーターを無視しても差し支えありません。あるいは、特定のデータベースへのアクセスを、有効なパスワードを特別に持つユーザーのみに限定する方針をとっている場合は、この関数は値 DB2SEC_PLUGIN_CONNECTIONREFUSED を戻すことができます。このパラメーターは、NULL にすることができます。

dbnamelen
入力。 dbname パラメーター値のバイト単位の長さ。 dbname パラメーターが NULL の場合、このパラメーターは 0 に設定されます。

connection_details
入力。 32 ビットのパラメーター。そのうちの 3 ビットが、以下の情報を保管するために現在使用されています。

- 右端のビットは、ユーザー ID のソースが db2secGetDefaultLoginContext のデフォルトであるか、それとも接続時に明示的に指定されているかを示します。
- 右から 2 番目のビットは、接続がローカル (Inter Process Communication (IPC) を使用しているか、あるいはパーティション・データベース環境の db2nodes.cfg 内にあるノードのいずれかからの接続である) か、リモート (ネットワークまたはループバックを経由) かを示します。これによって、API は同一のマシン上のクライアントがパスワードなしで DB2 サーバーに接続できるかどうか判別できます。デフォルトのオペレーティング・システム・ベースのユーザー ID/パスワード・プラグインにより、同一のマシン上のクライアントからのパスワードなしのローカル接続が常時許可されます (ユーザーが接続特権を持つ場合)。
- 右から 3 番目のビットは、DB2 データベース・マネージャーがサーバー・サイドまたはクライアント・サイドのどちらで API を呼び出しているかを示します。

ビット値は、db2secPlugin.h 内で次のように定義されています。

- DB2SEC_USERID_FROM_OS は、ユーザー ID は OS から取得され、接続ステートメントで明示的には指定されていないことを示します。
- DB2SEC_CONNECTION_ISLOCAL はローカル接続を示します。
- DB2SEC_VALIDATING_ON_SERVER_SIDE は、 DB2 データベース・マネージャーがサーバー・サイドまたはクライアント・サイドのどちらからパスワードの検証を呼び出しているかを示します。このビット値が設定されている場合、DB2 データベース・マネージャーはサーバー・サイドから呼び出しています。それ以外の場合は、クライアント・サイドから呼び出しています。

暗黙的な認証での DB2 データベース・システムのデフォルト動作では、パスワード検証なしの接続が許可されます。しかし、プラグイン開発者には、DB2SEC_PLUGIN_BADPASSWORD エラーを戻すことによって暗黙的な認証を禁止するという選択肢もあります。

token 入力。現行接続中の後続の API 呼び出しに渡されるデータを指すポインター。呼び出される可能性のある API は、db2secGetAuthIDs API と db2secGetGroupsForUser API です。

errormsg
出力。db2secValidatePassword API が正常に実行されない場合にこのパラメーターで戻されることのある、プラグインによって割り振られた ASCII エラー・メッセージ・ストリングのアドレスを指すポインター。

errormsglen
出力。errormsg パラメーターのエラー・メッセージ・ストリングのバイト単位の長さを示す整数を指すポインター。

GSS-API 認証プラグインに必要な API および定義

以下に、DB2 セキュリティー・プラグイン・インターフェースに必要な GSS-API の完全なリストを提示します。

サポートされている API は、*Generic Security Service Application Program Interface, Version 2* (IETF RFC2743) および *Generic Security Service API Version 2: C-Bindings* (IETF RFC2744) の仕様に準拠しています。 GSS-API ベース・プラグインをインプリメントする前に、これらの仕様について完全に理解しておく必要があります。

表 39. GSS-API 認証プラグインに必要な API および定義

名前		説明
クライアント・サイド API	gss_init_sec_context	対等アプリケーションとのセキュリティー・コンテキストを開始します。
サーバー・サイド API	gss_accept_sec_context	対等アプリケーションによって開始されたセキュリティー・コンテキストを受け入れます。
サーバー・サイド API	gss_display_name	内部フォーマットの名前をテキストに変換します。
共通 API	gss_delete_sec_context	確立されたセキュリティー・コンテキストを削除します。
共通 API	gss_display_status	GSS-API 状況コードに関連したテキスト・エラー・メッセージを取得します。
共通 API	gss_release_buffer	バッファーを削除します。
共通 API	gss_release_cred	GSS-API 証明書に関連したローカル・データ構造を解放します。
共通 API	gss_release_name	内部フォーマットの名前を削除します。
必要な定義	GSS_C_DELEG_FLAG	委任を要求します。
必要な定義	GSS_C_EMPTY_BUFFER	gss_buffer_desc にデータが含まれていないことを意味します。
必要な定義	GSS_C_GSS_CODE	GSS メジャー状況コードを示します。

表 39. GSS-API 認証プラグインに必要な API および定義 (続き)

名前		説明
必要な定義	GSS_C_INDEFINITE	メカニズムがコンテキストの有効期限をサポートしていないことを示します。
必要な定義	GSS_C_MECH_CODE	GSS マイナー状況コードを示します。
必要な定義	GSS_C_MUTUAL_FLAG	相互認証が要求されました。
必要な定義	GSS_C_NO_BUFFER	gss_buffer_t 変数が有効な gss_buffer_desc 構造を指していないことを示します。
必要な定義	GSS_C_NO_CHANNEL_BINDINGS	通信チャネル・バインディングがありません。
必要な定義	GSS_C_NO_CONTEXT	gss_ctx_id_t 変数が有効なコンテキストを指していないことを示します。
必要な定義	GSS_C_NO_CREDENTIAL	gss_cred_id_t 変数が有効な証明書ハンドルを指していないことを示します。
必要な定義	GSS_C_NO_NAME	gss_name_t 変数が有効な内部名を指していないことを示します。
必要な定義	GSS_C_NO_OID	デフォルト認証メカニズムを使用します。
必要な定義	GSS_C_NULL_OID_SET	デフォルト・メカニズムを使用します。
必要な定義	GSS_S_COMPLETE	API が正常に完了しました。
必要な定義	GSS_S_CONTINUE_NEEDED	プロセスが完了しておらず、対等機能から受け取った応答トークンを指定して API をもう一度呼び出す必要があります。

GSS-API 認証プラグインに関する制約事項

以下は、GSS-API 認証プラグインに関する制約事項のリストです。

- デフォルト・セキュリティー・メカニズムが常時採用されるため、OID についての考慮事項はありません。
- gss_init_sec_context() で要求される GSS サービスは、相互認証および委任だけです。 DB2 データベース・マネージャーは常に委任のためのチケットを要求しますが、そのチケットを使用して新規チケットを生成することはありません。
- デフォルト・コンテキスト時刻のみが要求されます。
- gss_delete_sec_context() からのコンテキスト・トークンは、クライアントからサーバー、またはその逆には送信されません。
- 匿名はサポートされていません。
- チャネル・バインディングはサポートされていません。
- 初期証明書の有効期限が切れた場合、DB2 データベース・マネージャーはそれを自動的に更新しません。
- GSS-API 仕様は、gss_init_sec_context() または gss_accept_sec_context() が失敗しても、対等機能に送信するトークンがいずれかの関数によって戻されなければならないことを規定しています。しかし、DRDA の制限のため、DB2 データベース・マネージャーは gss_init_sec_context() が失敗し、かつ最初の呼び出しでトークンを生成した場合にのみトークンを送信できます。

第 9 章 監査機能のレコード・レイアウト

監査ログから監査レコードを抽出する際、各レコードは、以下の表に示されているいずれかのフォーマットになります。各表の前には、サンプル・レコードを示します。

レコードの各項目の記述は、関連する表において一度に 1 つの行で示されます。表の中で、各項目は、抽出操作後に区切りファイルに出力されるときと同じ順序で示してあります。

注:

1. サンプル・レコードのすべてのフィールドが、必ずしも値を持っているとは限りません。
2. 中には、『Access Attempted』 のようにビットマップとして区切り文字付き ASCII 形式で保管されるフィールドもあります。ただし、現在のフラットなレポート・ファイルにおいて、それらのフィールドはビットマップ値を表す一連のストリングとして表示されます。

監査レコード・オブジェクト・タイプ

次の表は、監査レコード・オブジェクト・タイプ別に、各タイプで CHECKING、OBJMAINT、および SECMAINT イベントが生成される可能性があるかどうかを示しています。

表 40. 監査イベントに基づく監査レコード・オブジェクト・タイプ

オブジェクト・タイプ	CHECKING イベント	OBJMAINT イベント	SECMAINT イベント
ACCESS_RULE			X
ALIAS	X	X	
ALL	X		
AUDIT_POLICY	X	X	
BUFFERPOOL	X	X	
CHECK_CONSTRAINT		X	
DATABASE	X		X
DATA TYPE		X	
EVENT_MONITOR	X	X	
FOREIGN_KEY		X	
FUNCTION	X	X	X
FUNCTION MAPPING	X	X	
GLOBAL_VARIABLE	X	X	X
HISTOGRAM TEMPLATE	X	X	
INDEX	X	X	X
INDEX EXTENSION		X	
INSTANCE	X		

表 40. 監査イベントに基づく監査レコード・オブジェクト・タイプ (続き)

オブジェクト・タイプ	CHECKING イベント	OBJMAINT イベント	SECMAINT イベント
JAR_FILE		X	
METHOD_BODY	X	X	X
NICKNAME	X	X	X
NODEGROUP	X	X	
NONE	X	X	X
OPTIMIZATION PROFILE	X		
PACKAGE	X	X	X
PACKAGE CACHE	X		
PRIMARY_KEY		X	
REOPT_VALUES	X		
ROLE	X	X	X
SCHEMA	X	X	X
SECURITY LABEL		X	X
SECURITY LABEL COMPONENT		X	
SECURITY POLICY		X	X
SEQUENCE	X	X	
SERVER	X	X	X
SERVER OPTION	X	X	
SERVICE CLASS	X	X	
STORED_PROCEDURE	X	X	X
SUMMARY TABLES	X	X	X
TABLE	X	X	X
TABLESPACE	X	X	X
THRESHOLD	X	X	
TRIGGER		X	
TRUSTED CONTEXT	X	X	X
TYPE MAPPING	X	X	
TYPE&TRANSFORM	X	X	
UNIQUE_CONSTRAINT		X	
USER MAPPING	X	X	
VIEW	X	X	X
WORK ACTION SET	X	X	
WORK CLASS SET	X	X	
WORKLOAD	X	X	X
WRAPPER	X	X	
XSR オブジェクト	X	X	X

AUDIT イベントの監査レコード設計

次の表は、AUDIT イベントの監査レコード設計を示しています。

以下に監査レコードのサンプルを示します。

```
timestamp=2007-04-10-08.29.52.000001;
category=AUDIT;
audit event=START;
event correlator=0;
event status=0;
userid=newton;
authid=NEWTON;
application id=*LOCAL_APPLICATION;
application name=db2audit.exe;
```

表 41. AUDIT イベントの監査レコード設計

名前	フォーマット	説明
Timestamp	CHAR(26)	監査イベントの日付と時刻。
Category	CHAR(8)	監査イベントの区分。可能な値は以下のとおりです。 AUDIT
Audit Event	VARCHAR(32)	特定の監査イベント。 可能な値は以下のとおりです。 • CONFIGURE • DB2AUD • EXTRACT • FLUSH • PRUNE (バージョン 9.5 以降では生成されません) • START • STOP • UPDATE_ADMIN_CFG • ARCHIVE • LIST_LOGS • CREATE_AUDIT_POLICY • ALTER_AUDIT_POLICY • DROP_AUDIT_POLICY • AUDIT_USING • AUDIT_REPLACE • AUDIT_REMOVE
Event Correlator	INTEGER	監査対象の操作の相関 ID。単一イベントにどの監査レコードが関連しているかを識別するために使用できます。
Event Status	INTEGER	監査イベントの状況、次のような 1 つの SQLCODE で表されます。 成功イベント > = 0 失敗イベント < 0
User ID	VARCHAR(1024)	監査イベントの時刻でのユーザー ID。
Authorization ID	VARCHAR(128)	監査イベントの時刻での許可 ID。

表 41. AUDIT イベントの監査レコード設計 (続き)

名前	フォーマット	説明
Database Name	CHAR(8)	どのイベントが生成されたかを示すデータベース名。インスタンス・レベルの監査イベントであった場合にはブランクとなります。
Origin Node Number	SMALLINT	監査イベントが発生したノード番号。
Coordinator Node Number	SMALLINT	コーディネーター・ノードのノード番号。
Application ID	VARCHAR(255)	監査イベントが発生した時刻で使用していたアプリケーション ID。
Application Name	VARCHAR(1024)	監査イベントが発生した時刻で使用していたアプリケーション名。
Package Schema	VARCHAR(128)	監査イベントの時刻で使用していたパッケージのスキーマ。
Package Name	VARCHAR(128)	監査イベントが発生した時刻で使用していたパッケージ名。
Package Section	SMALLINT	監査イベントが発生した時刻で使用されていたパッケージのセクション番号。
Package Version	VARCHAR(64)	監査イベントが発生した時刻で使用していたパッケージのバージョン。
Local Transaction ID	VARCHAR(10) FOR BIT DATA	監査イベントが発生した時刻で使用していたローカル・トランザクション ID。これは、トランザクション・ログの一部となる SQLU_TID 構造体です。
Global Transaction ID	VARCHAR(30) FOR BIT DATA	監査イベントが発生した時刻で使用していたグローバル・トランザクション ID。これは、トランザクション・ログの一部となる SQLP_GXID 構造体のデータ・フィールドです。
Client User ID	VARCHAR(255)	監査イベントが発生した時刻の CURRENT CLIENT USERID 特殊レジスターの値。
Client Workstation Name	VARCHAR(255)	監査イベントが発生した時刻の CURRENT CLIENT_WRKSTNNAME 特殊レジスターの値。
Client Application Name	VARCHAR(255)	監査イベントが発生した時刻の CURRENT CLIENT_APPLNAME 特殊レジスターの値。
Client Accounting String	VARCHAR(255)	監査イベントが発生した時刻の CURRENT CLIENT_ACCTNG 特殊レジスターの値。
Trusted Context Name	VARCHAR(128)	トラステッド接続に関連付けられたトラステッド・コンテキストの名前。
Connection Trust Type	INTEGER	可能な値は以下のとおりです。 IMPLICIT_TRUSTED_CONNECTION EXPLICIT_TRUSTED_CONNECTION
Role Inherited	VARCHAR(128)	トラステッド接続を介して継承したロール。
Policy Name	VARCHAR(128)	監査ポリシー名。
Policy Association Object Type	CHAR(1)	監査ポリシーを関連付けるオブジェクトのタイプ。可能な値は以下のとおりです。 • N = ニックネーム • S = MQT • T = 表 (非型付き) • i = 許可 ID • g = 権限 • x = トラステッド・コンテキスト • ブランク = データベース

表 41. AUDIT イベントの監査レコード設計 (続き)

名前	フォーマット	説明
Policy Association Subobject Type	CHAR(1)	監査ポリシーを関連付けるサブオブジェクトのタイプ。オブジェクト・タイプが i (許可 ID) の場合、可能な値は以下のとおりです。 • U = ユーザー • G = グループ • R = ロール
Policy Association Object Name	VARCHAR(128)	監査ポリシーを関連付けるオブジェクトの名前。
Policy Association Object Schema	VARCHAR(128)	監査ポリシーを関連付けるオブジェクトのスキーマ名。「Policy Association Object Type」で、スキーマが適用されないオブジェクトが識別されている場合は、NULL になります。
Audit Status	CHAR(1)	監査ポリシーの AUDIT 区分の状況。可能な値は以下のとおりです。 • B- 両方 • F- 失敗 • N- なし • S- 成功
Checking Status	CHAR(1)	監査ポリシーの CHECKING 区分の状況。可能な値は以下のとおりです。 • B- 両方 • F- 失敗 • N- なし • S- 成功
Context Status	CHAR(1)	監査ポリシーの CONTEXT 区分の状況。可能な値は以下のとおりです。 • B- 両方 • F- 失敗 • N- なし • S- 成功
Execute Status	CHAR(1)	監査ポリシーの EXECUTE 区分の状況。可能な値は以下のとおりです。 • B- 両方 • F- 失敗 • N- なし • S- 成功
Execute With Data	CHAR(1)	監査ポリシーの EXECUTE 区分の WITH DATA オプション。可能な値は以下のとおりです。 • Y - WITH DATA • N - WITHOUT DATA

表 41. AUDIT イベントの監査レコード設計 (続き)

名前	フォーマット	説明
Objmaint Status	CHAR(1)	監査ポリシーの OBJMAINT 区分の状況。可能な値は以下のとおりです。 • B- 両方 • F- 失敗 • N- なし • S- 成功
Secmaint Status	CHAR(1)	監査ポリシーの SECMAINT 区分の状況。可能な値については、「Audit Status」フィールドを参照してください。
Sysadmin Status	CHAR(1)	監査ポリシーの SYSADMIN 区分の状況。可能な値は以下のとおりです。 • B- 両方 • F- 失敗 • N- なし • S- 成功
Validate Status	CHAR(1)	監査ポリシーの VALIDATE 区分の状況。可能な値は以下のとおりです。 • B- 両方 • F- 失敗 • N- なし • S- 成功
Error Type	CHAR(8)	監査ポリシーのエラー・タイプ。可能な値は AUDIT および NORMAL です。
Data Path	VARCHAR(1024)	db2audit configure コマンドで指定されたアクティブ監査ログのパス。
Archive Path	VARCHAR(1024)	db2audit configure コマンドで指定されたアーカイブされた監査ログのパス。

CHECKING イベントの監査レコード設計

次の表は、CHECKING イベントの監査レコードの形式を示しています。

以下に監査レコードのサンプルを示します。

```
timestamp=1998-06-24-08.42.11.622984;
category=CHECKING;
audit event=CHECKING_OBJECT;
event correlator=2;
event status=0;
database=FOO;
userid=boss;
authid=BOSS;
application id=*LOCAL.newton.980624124210;
application name=testapp;
package schema=NULLID;
package name=SYSSH200;
package section=0;
object schema=GSTAGER;
```

```
object name=NONE;
object type=REOPT_VALUES;
access approval reason=DBADM;
access attempted=STORE;
```

表 42. CHECKING イベントの監査レコード設計

名前	フォーマット	説明
Timestamp	CHAR(26)	監査イベントの日付と時刻。
Category	CHAR(8)	監査イベントの区分。可能な値は以下のとおりです。 CHECKING
Audit Event	VARCHAR(32)	特定の監査イベント。 可能な値は CHECKING_OBJECT、 CHECKING_FUNCTION、CHECKING_TRANSFER、および CHECKING_MEMBERSHIP_IN_ROLES です。
Event Correlator	INTEGER	監査対象の操作の相関 ID。単一イベントにどの監査レコードが関連しているかを識別するために使用できます。
Event Status	INTEGER	監査イベントの状況、次のような 1 つの SQLCODE で表されます。 成功イベント > = 0 失敗イベント < 0
Database Name	CHAR(8)	どのイベントが生成されたかを示すデータベース名。インスタンス・レベルの監査イベントであった場合にはブランクとなります。
User ID	VARCHAR(1024)	監査イベントの時刻でのユーザー ID。
Authorization ID	VARCHAR(128)	監査イベントの時刻での許可 ID。
Origin Node Number	SMALLINT	監査イベントが発生したノード番号。
Coordinator Node Number	SMALLINT	コーディネーター・ノードのノード番号。
Application ID	VARCHAR(255)	監査イベントが発生した時刻で使用していたアプリケーション ID。
Application Name	VARCHAR(1024)	監査イベントが発生した時刻で使用していたアプリケーション名。
Package Schema	VARCHAR(128)	監査イベントの時刻で使用していたパッケージのスキーマ。
Package Name	VARCHAR(128)	監査イベントが発生した時刻で使用していたパッケージ名。
Package Section Number	SMALLINT	監査イベントが発生した時刻で使用されていたパッケージのセクション番号。
Object Schema	VARCHAR(128)	監査イベントの生成対象となったオブジェクトのスキーマ。
Object Name	VARCHAR(128)	監査イベントの生成対象となったオブジェクトの名前。
Object Type	VARCHAR(32)	監査イベントの生成対象となったオブジェクトのタイプ。可能な値については、『監査レコード・オブジェクト・タイプ』というトピックに示されています。
Access Approval Reason	CHAR(18)	アクセスがこの監査イベントで承認された理由を示します。可能な値については、『有効な CHECKING アクセス承認理由のリスト』というトピックに示されています。
Access Attempted	CHAR(18)	試みられたアクセスのタイプを示します。可能な値については、『有効な CHECKING アクセス試行タイプのリスト』というトピックに示されています。

表 42. CHECKING イベントの監査レコード設計 (続き)

名前	フォーマット	説明
Package Version	VARCHAR (64)	監査イベントが発生した時点で使用されていたパッケージのバージョン。
Checked Authorization ID	VARCHAR(128)	許可 ID は監査イベント時の許可 ID と異なる場合にチェックされます。例えば、これは TRANSFER OWNERSHIP ステートメントのターゲット所有者などです。 監査イベントが SWITCH_USER である場合、このフィールドには、切り替え後の許可 ID が表示されます。
Local Transaction ID	VARCHAR(10) FOR BIT DATA	監査イベントが発生した時刻で使用していたローカル・トランザクション ID。これは、トランザクション・ログの一部となる SQLU_TID 構造体です。
Global Transaction ID	VARCHAR(30) FOR BIT DATA	監査イベントが発生した時刻で使用していたグローバル・トランザクション ID。これは、トランザクション・ログの一部となる SQLP_GXID 構造体のデータ・フィールドです。
Client User ID	VARCHAR(255)	監査イベントが発生した時刻の CURRENT CLIENT USERID 特殊レジスターの値。
Client Workstation Name	VARCHAR(255)	監査イベントが発生した時刻の CURRENT CLIENT_WRKSTNNAME 特殊レジスターの値。
Client Application Name	VARCHAR(255)	監査イベントが発生した時刻の CURRENT CLIENT_APPLNAME 特殊レジスターの値。
Client Accounting String	VARCHAR(255)	監査イベントが発生した時刻の CURRENT CLIENT_ACCTNG 特殊レジスターの値。
Trusted Context Name	VARCHAR(128)	トラステッド接続に関連付けられたトラステッド・コンテキストの名前。
Connection Trust Type	INTEGER	可能な値は以下のとおりです。 IMPLICIT_TRUSTED_CONNECTION EXPLICIT_TRUSTED_CONNECTION
Role Inherited	VARCHAR(128)	トラステッド接続を介して継承したロール。

CHECKING アクセス承認理由

次のリストは、有効な CHECKING アクセス承認理由を示しています。

0x0000000000000001 ACCESS DENIED
アクセスは承認されません。その上、拒否されました。

0x0000000000000002 SYSADM
アクセスは承認されます。アプリケーションまたはユーザーは SYSADM 権限を持ちます。

0x0000000000000004 SYSCTRL
アクセスは承認されます。アプリケーションまたはユーザーは SYSCTRL 権限を持ちます。

0x0000000000000008 SYSMAINT
アクセスは承認されます。アプリケーションまたはユーザーは SYSMAINT 権限を持ちます。

0x0000000000000010 DBADM
アクセスは承認されます。アプリケーションまたはユーザーは DBADM 権限を持ちます。

0x0000000000000020 DATABASE PRIVILEGE
アクセスは承認されます。アプリケーションまたはユーザーはこのデータベースに関して明示的な権限を持ちます。

0x0000000000000040 OBJECT PRIVILEGE
アクセスは承認されます。アプリケーションまたはユーザーは、オブジェクトまたは関数に関する特権を持ちます。

0x0000000000000080 DEFINER
アクセスは承認されます。アプリケーションまたはユーザーは、オブジェクトまたは関数の定義をするものとなります。

0x0000000000000100 OWNER
アクセスは承認されます。アプリケーションまたはユーザーは、オブジェクトまたは関数の所有者となります。

0x0000000000000200 CONTROL
アクセスは承認されます。アプリケーションまたはユーザーは、オブジェクトまたは関数に関する CONTROL 権限を持ちます。

0x0000000000000400 BIND
アクセスは承認されます。アプリケーションまたはユーザーは、パッケージに関するバインド権限を持ちます。

0x0000000000000800 SYSQUIESCE
アクセスは承認されます。インスタンスまたはデータベースが静止モードにある場合は、アプリケーションまたはユーザーは接続またはアタッチを行うことができます。

0x0000000000001000 SYSMON
アクセスは承認されます。アプリケーションまたはユーザーは SYSMON 権限を持ちます。

0x0000000000002000 SECADM
アクセスは承認されます。アプリケーションまたはユーザーは SECADM 権限を持ちます。

0x0000000000004000 SETSESSIONUSER
アクセスは承認されます。アプリケーションまたはユーザーは SETSESSIONUSER 権限を持ちます。

0x0000000000008000 TRUSTED_CONTEXT_MATCH
接続属性が、DB2 サーバーで定義されている固有のトラステッド・コンテキストの属性と一致しました。

0x0000000000010000 TRUSTED_CONTEXT_USE
トラステッド・コンテキストを使用するためのアクセスが承認されます。

CHECKING アクセス試行タイプ

次のリストは、有効な CHECKING アクセス試行タイプを示しています。

監査イベントが CHECKING_TRANSFER の場合、監査項目は特権が保留されるかどうかを反映します。

0x0000000000000001 CONTROL
CONTROL 特権が保留されるかどうかを検査しようとします。

0x0000000000000002 ALTER
オブジェクトを変更しようとするか、または監査イベントが CHECKING_TRANSFER の場合に ALTER 特権が保留されるかどうかを検査しようとします。

0x0000000000000004 DELETE
オブジェクトを削除しようとするか、または監査イベントが CHECKING_TRANSFER の場合に DELETE 特権が保留されるかどうかを検査しようとします。

0x0000000000000008 INDEX
索引を使用しようとするか、または監査イベントが CHECKING_TRANSFER の場合に INDEX 特権が保留されるかどうかを検査しようとします。

0x0000000000000010 INSERT
オブジェクトに挿入しようとするか、または監査イベントが CHECKING_TRANSFER の場合に INSERT 特権が保留されるかどうかを検査しようとします。

0x0000000000000020 SELECT
表またはビューを照会しようとするか、または監査イベントが CHECKING_TRANSFER の場合に SELECT 特権が保留されるかどうかを検査しようとします。

0x0000000000000040 UPDATE
オブジェクト内のデータを更新しようとするか、または監査イベントが CHECKING_TRANSFER の場合に UPDATE 特権が保留されるかどうかを検査しようとします。

0x0000000000000080 REFERENCE
オブジェクト間の参照制約を確立しようとするか、または監査イベントが CHECKING_TRANSFER の場合に REFERENCE 特権が保留されるかどうかを検査しようとします。

0x0000000000000100 CREATE
オブジェクトを作成しようとします。

0x0000000000000200 DROP
オブジェクトをドロップしようとします。

0x0000000000000400 CREATEIN
別のスキーマ内にオブジェクトを作成しようとします。

0x0000000000000800 DROPIN
別のスキーマ内に見いだされるオブジェクトをドロップしようとします。

0x0000000000001000 ALTERIN
別のスキーマ内に見いだされるオブジェクトを変更しようとします。

0x0000000000002000 EXECUTE
アプリケーションを実行、またはルーチンを呼び出そうとします。ルーチンからのソース関数を作成する (関数のみに適用されます) か、何らかの DDL ステートメントのルーチンを参照するか、または監査イベントが CHECKING_TRANSFER の場合に EXECUTE 特権が保留されるかどうかを検査しようとします。

0x0000000000004000 BIND
アプリケーションをバインドまたは準備しようとします。

0x0000000000008000 SET EVENT MONITOR
イベント・モニターのスイッチをセットしようとします。

0x0000000000010000 SET CONSTRAINTS
オブジェクトに関する制約をセットしようとします。

0x0000000000020000 COMMENT ON
オブジェクトに関する注釈を作成しようとします。

0x0000000000040000 GRANT
あるオブジェクトに対する特権またはロールを別の許可 ID に付与しようとします。

0x0000000000080000 REVOKE
あるオブジェクトに対する特権またはロールを許可 ID から取り消そうとします。

0x0000000000100000 LOCK
オブジェクトをロックしようとします。

0x0000000000200000 RENAME
オブジェクトを名前変更しようとします。

0x0000000000400000 CONNECT
オブジェクトに接続しようとします。

0x0000000000800000 Member of SYS Group
SYS グループのメンバーをアクセスまたは使用しようとします。

0x0000000001000000 Access All
保持されているオブジェクトに対して必要なすべての特権を使用して、ステートメントを実行しようとします (DBADM/SYSADM でのみ使用されます)。

0x0000000002000000 Drop All
複数のオブジェクトをドロップしようとします。

0x0000000004000000 LOAD
表スペースに表をロードしようとします。

0x0000000008000000 USE
表スペースに表を作成しようとするか、または監査イベントが CHECKING_TRANSFER の場合に USE 特権が保留されるかどうかを検査しようとします。

0x0000000010000000 SET SESSION_USER
SET SESSION_USER ステートメントを実行しようとします。

0x0000000020000000 FLUSH
FLUSH ステートメントを実行しようとします。

0x0000000040000000 STORE
EXPLAIN_PREDICATE 表内の再び最適化されたステートメントの値を表示しようとします。

0x0000000400000000 TRANSFER
オブジェクトを転送しようとします。

0x0000000800000000 ALTER_WITH_GRANT
GRANT 特権の付いた ALTER が保留かどうかを検査しようとします。

0x0000001000000000 DELETE_WITH_GRANT
GRANT 特権の付いた DELETE が保留かどうかを検査しようとします。

0x0000002000000000 INDEX_WITH_GRANT
GRANT 特権の付いた INDEX が保留かどうかを検査しようとします。

0x0000004000000000 INSERT_WITH_GRANT
GRANT 特権の付いた INSERT が保留かどうかを検査しようとします。

0x0000008000000000 SELECT_WITH_GRANT
GRANT 特権の付いた SELECT が保留かどうかを検査しようとします。

0x0000010000000000 UPDATE_WITH_GRANT
GRANT 特権の付いた UPDATE が保留かどうかを検査しようとします。

0x0000020000000000 REFERENCE_WITH_GRANT
GRANT 特権の付いた REFERENCE が保留かどうかを検査しようとします。

0x0000040000000000 USAGE
シーケンスまたは XSR オブジェクトを使用しようとするか、または監査イベントが CHECKING_TRANSFER の場合に USAGE 特権が保留されるかどうかを検査しようとします。

0x0000080000000000 SET ROLE
ロールを設定しようとします。

0x0000100000000000 EXPLICIT_TRUSTED_CONNECTION
明示的なトラステッド接続を確立しようとします。

0x0000200000000000 IMPLICIT_TRUSTED_CONNECTION
暗黙的なトラステッド接続を確立しようとします。

0x0000400000000000 READ
グローバル変数を読み取ろうとします。

0x0000800000000000 WRITE
グローバル変数を書き込もうとします。

0x0001000000000000 SWITCH_USER
明示的なトラステッド接続でユーザー ID を切り替えようとします。

0x0002000000000000 AUDIT_USING
オブジェクトに監査ポリシーを関連付けようとします。

0x0004000000000000 AUDIT_REPLACE
オブジェクトへの監査ポリシーの関連付けを置き換えようとします。

0x0008000000000000 AUDIT_REMOVE
オブジェクトへの監査ポリシーの関連付けを除去しようとします。

0x0010000000000000 AUDIT_ARCHIVE
監査ログをアーカイブしようとします。

0x0020000000000000 AUDIT_EXTRACT
監査ログを抽出しようとします。

0x0040000000000000 AUDIT_LIST_LOGS
監査ログをリストしようとします。

OBJMAINT イベントの監査レコード設計

次の表は、OBJMAINT イベントの監査レコードの形式を示しています。

以下に監査レコードのサンプルを示します。

```
timestamp=1998-06-24-08.42.41.957524;
category=OBJMAINT;
audit event=CREATE_OBJECT;
event correlator=3;
event status=0;
database=FOO;
userid=boss;
authid=BOSS;
application id=*LOCAL.newton.980624124210;
application name=testapp;
package schema=NULLID;
package name=SQLC28A1;
package section=0;
object schema=BOSS;
object name=AUDIT;
object type=TABLE;
```

表 43. OBJMAINT イベントの監査レコード設計

名前	フォーマット	説明
Timestamp	CHAR(26)	監査イベントの日付と時刻。
Category	CHAR(8)	監査イベントの区分。可能な値は以下のとおりです。 OBJMAINT
Audit Event	VARCHAR(32)	特定の監査イベント。 有効な値は、 CREATE_OBJECT、RENAME_OBJECT、DROP_OBJECT、および ALTER_OBJECT です。 ALTER_OBJECT イベントは、保護表を変更するときにのみ生成されます。
Event Correlator	INTEGER	監査対象の操作の相関 ID。単一イベントにどの監査レコードが関連しているかを識別するために使用できます。
Event Status	INTEGER	監査イベントの状況、次のような 1 つの SQLCODE で表されます。 成功イベント > = 0 失敗イベント < 0

表 43. *OBJMAINT* イベントの監査レコード設計 (続き)

名前	フォーマット	説明
Database Name	CHAR(8)	どのイベントが生成されたかを示すデータベース名。インスタンス・レベルの監査イベントであった場合にはブランクとなります。
User ID	VARCHAR(1024)	監査イベントの時刻でのユーザー ID。
Authorization ID	VARCHAR(128)	監査イベントの時刻での許可 ID。
Origin Node Number	SMALLINT	監査イベントが発生したノード番号。
Coordinator Node Number	SMALLINT	コーディネーター・ノードのノード番号。
Application ID	VARCHAR(255)	監査イベントが発生した時刻で使用していたアプリケーション ID。
Application Name	VARCHAR(1024)	監査イベントが発生した時刻で使用していたアプリケーション名。
Package Schema	VARCHAR(128)	監査イベントの時刻で使用していたパッケージのスキーマ。
Package Name	VARCHAR(256)	監査イベントが発生した時刻で使用していたパッケージ名。
Package Section Number	SMALLINT	監査イベントが発生した時刻で使用されていたパッケージのセクション番号。
Object Schema	VARCHAR(128)	監査イベントの生成対象となったオブジェクトのスキーマ。
Object Name	VARCHAR(128)	監査イベントの生成対象となったオブジェクトの名前。
Object Type	VARCHAR(32)	監査イベントの生成対象となったオブジェクトのタイプ。可能な値については、『監査レコード・オブジェクト・タイプ』というトピックに示されています。
Package Version	VARCHAR (64)	監査イベントが発生した時刻で使用していたパッケージのバージョン。
Security Policy Name	VARCHAR(128)	オブジェクト・タイプが TABLE でその表がセキュリティー・ポリシーに関連している場合、そのセキュリティー・ポリシーの名前。
Alter Action	VARCHAR(32)	特定の変更操作 可能な値は以下のとおりです。 • ADD_PROTECTED_COLUMN • ADD_COLUMN_PROTECTION • DROP_COLUMN_PROTECTION • ADD_ROW_PROTECTION • ADD_SECURITY_POLICY • ADD_ELEMENT • ADD COMPONENT • USE GROUP AUTHORIZATIONS • IGNORE GROUP AUTHORIZATIONS • USE ROLE AUTHORIZATIONS • IGNORE ROLE AUTHORIZATIONS • OVERRIDE NOT AUTHORIZED WRITE SECURITY LABEL • RESTRICT NOT AUTHORIZED WRITE SECURITY LABEL
Protected Column Name	VARCHAR(128)	Alter Action が ADD_COLUMN_PROTECTION または DROP_COLUMN_PROTECTION の場合、これは影響される列の名前です。

表 43. OBJMAINT イベントの監査レコード設計 (続き)

名前	フォーマット	説明
Column Security Label	VARCHAR(128)	フィールド Column Name で指定された列を保護するセキュリティー・ラベル。
Security Label Column Name	VARCHAR(128)	行を保護するセキュリティー・ラベルを含む列の名前。
Local Transaction ID	VARCHAR(10) FOR BIT DATA	監査イベントが発生した時刻で使用していたローカル・トランザクション ID。これは、トランザクション・ログの一部となる SQLU_TID 構造体です。
Global Transaction ID	VARCHAR(30) FOR BIT DATA	監査イベントが発生した時刻で使用していたグローバル・トランザクション ID。これは、トランザクション・ログの一部となる SQLP_GXID 構造体のデータ・フィールドです。
Client User ID	VARCHAR(255)	監査イベントが発生した時刻の CURRENT CLIENT USERID 特殊レジスターの値。
Client Workstation Name	VARCHAR(255)	監査イベントが発生した時刻の CURRENT CLIENT_WRKSTNNAME 特殊レジスターの値。
Client Application Name	VARCHAR(255)	監査イベントが発生した時刻の CURRENT CLIENT_APPLNAME 特殊レジスターの値。
Client Accounting String	VARCHAR(255)	監査イベントが発生した時刻の CURRENT CLIENT_ACCTNG 特殊レジスターの値。
Trusted Context Name	VARCHAR(128)	トラステッド接続に関連付けられたトラステッド・コンテキストの名前。
Connection Trust Type	INTEGER	可能な値は以下のとおりです。 IMPLICIT_TRUSTED_CONNECTION EXPLICIT_TRUSTED_CONNECTION
Role Inherited	VARCHAR(128)	トラステッド接続を介して継承したロール。

SECMAINT イベントの監査レコード設計

次の表は、SECMAINT イベントの監査レコードの形式を示しています。

以下に監査レコードのサンプルを示します。

```
timestamp=1998-06-24-11.57.45.188101;
category=SECMAINT;
audit event=GRANT;
event correlator=4;
event status=0;
database=FOO;
userid=boss;
authid=BOSS;
application id=*LOCAL.boss.980624155728;
application name=db2bp;
package schema=NULLID;
package name=SQLC28A1;
package section=0;
object schema=BOSS;
object name=T1;
object type=TABLE;
grantor=BOSS;
grantee=WORKER;
grantee type=USER;
privilege=SELECT;
```

表 44. SECMAINT イベントの監査レコード設計

名前	フォーマット	説明
Timestamp	CHAR(26)	監査イベントの日付と時刻。
Category	CHAR(8)	監査イベントの区分。可能な値は以下のとおりです。 SECMAINT
Audit Event	VARCHAR(32)	特定の監査イベント。 可能な値は GRANT、REVOKE、IMPLICIT_GRANT、IMPLICIT_REVOKE、SET_SESSION_USER、 UPDATE_DBM_CFG、TRANSFER_OWNERSHIP、 ADD_DEFAULT_ROLE、DROP_DEFAULT_ROLE、 ALTER_DEFAULT_ROLE、ADD_USER、 DROP_USER、 ALTER_USER_ADD_ROLE、ALTER_USER_DROP_ROLE、 ALTER_USER_AUTHENTICATION、および ALTER SECURITY POLICY です。
Event Correlator	INTEGER	監査対象の操作の相関 ID。単一イベントにどの監査レコードが関連しているかを識別するために使用できます。
Event Status	INTEGER	監査イベントの状況、次のような 1 つの SQLCODE で表されます。 成功イベント > = 0 失敗イベント < 0
Database Name	CHAR(8)	どのイベントが生成されたかを示すデータベース名。インスタンス・レベルの監査イベントであった場合にはブランクとなります。
User ID	VARCHAR(1024)	監査イベントの時刻でのユーザー ID。
Authorization ID	VARCHAR(128)	監査イベントの時刻での許可 ID。
Origin Node Number	SMALLINT	監査イベントが発生したノード番号。
Coordinator Node Number	SMALLINT	コーディネーター・ノードのノード番号。
Application ID	VARCHAR(255)	監査イベントが発生した時刻で使用していたアプリケーション ID。
Application Name	VARCHAR(1024)	監査イベントが発生した時刻で使用していたアプリケーション名。
Package Schema	VARCHAR(128)	監査イベントの時刻で使用していたパッケージのスキーマ。
Package Name	VARCHAR(128)	監査イベントが発生した時刻で使用していたパッケージ名。
Package Section Number	SMALLINT	監査イベントが発生した時刻で使用されていたパッケージのセクション番号。
Object Schema	VARCHAR(128)	監査イベントの生成対象となったオブジェクトのスキーマ。 オブジェクト・タイプ・フィールドが ACCESS_RULE なら、このフィールドには規則に関連したセキュリティー・ポリシー名が含まれます。 規則の名前はオブジェクト名のフィールドに格納されます。 オブジェクト・タイプ・フィールドが SECURITY_LABEL なら、このフィールドにはセキュリティー・ラベルがその一部であるセキュリティー・ポリシーの名前が含まれます。 セキュリティー・ラベルの名前はオブジェクト名のフィールドに格納されます。

表 44. SECMAINT イベントの監査レコード設計 (続き)

名前	フォーマット	説明
Object Name	VARCHAR(128)	監査イベントの生成対象となったオブジェクトの名前。 監査イベントが ADD_DEFAULT_ROLE、DROP_DEFAULT_ROLE、 ALTER_DEFAULT_ROLE、 ADD_USER, DROP_USER、 ALTER_USER_ADD_ROLE、ALTER_USER_DROP_ROLE、または ALTER_USER_AUTHENTICATION のいずれかである場合に、ロール名を示します。 オブジェクト・タイプ・フィールドが ACCESS_RULE なら、このフィールドには規則の名前が含まれます。 規則に関連するセキュリティー・ポリシーの名前はオブジェクト・スキーマのフィールドに格納されます。 オブジェクト・タイプ・フィールドが SECURITY_LABEL なら、このフィールドにはセキュリティー・ラベルの名前が含まれます。 セキュリティー・ポリシーの一部である名前はオブジェクト・スキーマのフィールドに格納されます。
Object Type	VARCHAR(32)	監査イベントの生成対象となったオブジェクトのタイプ。可能な値については、『監査レコード・オブジェクト・タイプ』というトピックに示されています。 監査イベントが ADD_DEFAULT_ROLE、DROP_DEFAULT_ROLE、 ALTER_DEFAULT_ROLE、ADD_USER、 DROP_USER、 ALTER_USER_ADD_ROLE、ALTER_USER_DROP_ROLE、および ALTER_USER_AUTHENTICATION のいずれかである場合、値は ROLE です。
Grantor	VARCHAR(128)	特権や権限の付与または取り消しを行う ID。
Grantee	VARCHAR(128)	特権または権限が付与または取り消された被認可者の ID。 監査イベントが ADD_DEFAULT_ROLE、DROP_DEFAULT_ROLE、 ALTER_DEFAULT_ROLE、ADD_USER、 DROP_USER、 ALTER_USER_ADD_ROLE、ALTER_USER_DROP_ROLE、または ALTER_USER_AUTHENTICATION のいずれかである場合に、トラステッド・コンテキスト・オブジェクトを示します。
Grantee Type	VARCHAR(32)	付与または取り消された被認可者のタイプ。可能な値は USER、GROUP、ROLE、AMBIGUOUS、または、監査イベントが ADD_DEFAULT_ROLE、 DROP_DEFAULT_ROLE、ALTER_DEFAULT_ROLE、 ADD_USER、 DROP_USER、ALTER_USER_ADD_ROLE、 ALTER_USER_DROP_ROLE、または ALTER_USER_AUTHENTICATION のいずれかである場合には TRUSTED_CONTEXT です。

表 44. SECMAINT イベントの監査レコード設計 (続き)

名前	フォーマット	説明
Privilege または Authority	CHAR(18)	付与または取り消された特権または権限のタイプを示します。有効な値は、『有効な SECMAINT 特権または権限のリスト』というトピックに示されています。 監査イベントが ADD_DEFAULT_ROLE、DROP_DEFAULT_ROLE、 ALTER_DEFAULT_ROLE、ADD_USER、 DROP_USER、 ALTER_USER_ADD_ROLE、ALTER_USER_DROP_ROLE、または ALTER_USER_AUTHENTICATION のいずれかである場合、値は ROLE MEMBERSHIP です。
Package Version	VARCHAR (64)	監査イベントが発生した時刻で使用していたパッケージのバージョン。
アクセス・タイプ	VARCHAR(32)	セキュリティー・ラベルが付与されるアクセス・タイプ。 可能な値: • READ • WRITE • ALL セキュリティー・ポリシーが変更されるアクセス・タイプ。可能な値: • USE GROUP AUTHORIZATIONS • IGNORE GROUP AUTHORIZATIONS • USE ROLE AUTHORIZATIONS • IGNORE ROLE AUTHORIZATIONS • OVERRIDE NOT AUTHORIZED WRITE SECURITY LABEL • RESTRICT NOT AUTHORIZED WRITE SECURITY LABEL
Assumable Authid	VARCHAR(128)	付与される特権が SETSESSIONUSER 特権のとき、これは被認可者がセッション・ユーザーとして設定されることが可能な許可 ID です。
Local Transaction ID	VARCHAR(10) FOR BIT DATA	監査イベントが発生した時刻で使用していたローカル・トランザクション ID。これは、トランザクション・ログの一部となる SQLU_TID 構造体です。
Global Transaction ID	VARCHAR(30) FOR BIT DATA	監査イベントが発生した時刻で使用していたグローバル・トランザクション ID。これは、トランザクション・ログの一部となる SQLP_GXID 構造体のデータ・フィールドです。
Grantor Type	VARCHAR(32)	付与者のタイプ。可能な値は、USER です。
Client User ID	VARCHAR(255)	監査イベントが発生した時刻の CURRENT CLIENT USERID 特殊レジスターの値。
Client Workstation Name	VARCHAR(255)	監査イベントが発生した時刻の CURRENT CLIENT_WRKSTNNAME 特殊レジスターの値。
Client Application Name	VARCHAR(255)	監査イベントが発生した時刻の CURRENT CLIENT_APPLNAME 特殊レジスターの値。
Client Accounting String	VARCHAR(255)	監査イベントが発生した時刻の CURRENT CLIENT_ACCTNG 特殊レジスターの値。

表 44. SECMAINT イベントの監査レコード設計 (続き)

名前	フォーマット	説明
Trusted Context User	VARCHAR(128)	監査イベントが ADD_USER または DROP_USER であるときに、トラステッド・コンテキスト・ユーザーを識別する。
Trusted Context User Authentication	INTEGER	監査イベントが ADD_USER、DROP_USER、または ALTER_USER_AUTHENTICATION であるときに、トラステッド・コンテキスト・ユーザーの認証設定を示す。 1 : 認証が必要 0 : 認証は不要
Trusted Context Name	VARCHAR(128)	トラステッド接続に関連付けられたトラステッド・コンテキストの名前。
Connection Trust Type	INTEGER	可能な値は以下のとおりです。 IMPLICIT_TRUSTED_CONNECTION EXPLICIT_TRUSTED_CONNECTION
Role Inherited	VARCHAR(128)	トラステッド接続を介して継承したロール。

SECMAINT 特権または権限

次のリストは、有効な SECMAINT 特権または権限を示しています。

0x0000000000000001 Control Table
表またはビューに関して付与または取り消されたコントロール特権。

0x0000000000000002 ALTER
表またはシーケンスを変更するために付与または取り消された特権。

0x0000000000000004 ALTER with GRANT
特権の付与が許可されている表またはシーケンスを変更するために付与または取り消された特権。

0x0000000000000008 DELETE TABLE
表またはビューをドロップするために付与または取り消された特権。

0x0000000000000010 DELETE TABLE with GRANT
特権の付与が許可されている表をドロップするために付与または取り消された特権。

0x0000000000000020 Table Index
索引に関して付与または取り消された特権。

0x0000000000000040 Table Index with GRANT
特権の付与が許可されている索引に関して付与または取り消された特権。

0x0000000000000080 Table INSERT
表またはビューへの挿入に関して付与または取り消された特権。

0x0000000000000100 Table INSERT with GRANT
特権の付与が許可されている表への挿入に関して付与または取り消された特権。

0x0000000000000200 Table SELECT
表での選択に関して付与または取り消された特権。

0x0000000000000400 Table SELECT with GRANT
特権の付与が許可されている表での選択に関して付与または取り消された特権。

0x0000000000000800 Table UPDATE
表またはビューの更新に関して付与または取り消された特権。

0x0000000000001000 Table UPDATE with GRANT
特権の付与が許可されている表またはビューの更新に関して付与または取り消された特権。

0x0000000000002000 Table REFERENCE
表への参照に関して付与または取り消された特権。

0x0000000000004000 Table REFERENCE with GRANT
特権の付与が許可されている表への参照に関して付与または取り消された特権。

0x0000000000020000 CREATEIN Schema
スキーマに関して付与または取り消された CREATEIN 特権。

0x0000000000040000 CREATEIN Schema with GRANT
特権の付与が許可されているスキーマに関して付与または取り消された CREATEIN 特権。

0x0000000000080000 DROPIN Schema
スキーマに関して付与または取り消された DROPIN 特権。

0x0000000000100000 DROPIN Schema with GRANT
特権の付与が許可されているスキーマに関して付与または取り消された DROPIN 特権。

0x0000000000200000 ALTERIN Schema
スキーマに関して付与または取り消された ALTERIN 特権。

0x0000000000400000 ALTERIN Schema with GRANT
特権の付与が許可されているスキーマに関して付与または取り消された ALTERIN 特権。

0x0000000000800000 DBADM Authority
付与または取り消された DBADM 権限。

0x0000000001000000 CREATETAB Authority
付与または取り消された CREATETAB 権限。

0x0000000002000000 BINDADD Authority
付与または取り消された BINDADD 権限。

0x0000000004000000 CONNECT Authority
付与または取り消された CONNECT 権限。

0x0000000008000000 Create not fenced Authority
付与または取り消された fenced でない作成権限。

0x0000000010000000 Implicit Schema Authority
付与または取り消された暗黙的スキーマ権限。

0x0000000020000000 Server PASSTHRU
このサーバー (フェデレーテッド・データベースのデータ・ソース) でパススルー機能を使用するために、付与または取り消された特権。

0x0000000040000000 ESTABLISH TRUSTED CONNECTION
トラステッド接続が作成された。

0x0000000100000000 Table Space USE
表スペースに表を作成するために付与または取り消された特権。

0x0000000200000000 Table Space USE with GRANT
特権の付与が許可されている表スペースに表を作成するために付与または取り消された特権。

0x0000000400000000 Column UPDATE
表の 1 つ以上の特定の列への更新に関して付与または取り消された特権。

0x0000000800000000 Column UPDATE with GRANT
特権の付与が許可されている表の 1 つ以上の特定の列への更新に関して付与または取り消された特権。

0x0000001000000000 Column REFERENCE
表の 1 つ以上の特定の列への参照に関して付与または取り消された特権。

0x0000002000000000 Column REFERENCE with GRANT
特権の付与が許可されている表の 1 つ以上の特定の列への参照に関して付与または取り消された特権。

0x0000004000000000 LOAD Authority
付与または取り消された LOAD 権限。

0x0000008000000000 Package BIND
パッケージに関して付与または取り消された BIND 特権。

0x0000010000000000 Package BIND with GRANT
特権の付与が許可されているパッケージに関して付与または取り消された BIND 特権。

0x0000020000000000 EXECUTE
パッケージまたはルーチンに関して付与または取り消された EXECUTE 特権。

0x0000040000000000 EXECUTE with GRANT
特権の付与が許可されているパッケージまたはルーチンに関して付与または取り消された EXECUTE 特権。

0x0000080000000000 EXECUTE IN SCHEMA
スキーマ内のすべてのルーチンに関して付与または取り消された EXECUTE 特権。

0x0000100000000000 EXECUTE IN SCHEMA with GRANT
特権の付与が許可されているスキーマ内のすべてのルーチンに関して付与または取り消された EXECUTE 特権。

0x0000200000000000 EXECUTE IN TYPE
タイプ内のすべてのルーチンに関して付与または取り消された EXECUTE 特権。

0x0000400000000000 EXECUTE IN TYPE with GRANT
特権の付与が許可されているタイプ内のすべてのルーチンに関して付与または取り消された EXECUTE 特権。

0x0000800000000000 CREATE EXTERNAL ROUTINE
付与または取り消された CREATE EXTERNAL ROUTINE 特権。

0x0001000000000000 QUIESCE_CONNECT
付与または取り消された QUIESCE_CONNECT 特権。

0x0004000000000000 SECADM Authority
付与または取り消された SECADM 権限。

0x0008000000000000 USAGE Authority
シーケンスに関して付与または取り消された USAGE 特権。

0x0010000000000000 USAGE with GRANT Authority
特権の付与が許可されているシーケンスに関して付与または取り消された USAGE 特権。

0x0020000000000000 WITH ADMIN Option
ロールに関して WITH ADMIN オプションが付与または取り消された。

0x0040000000000000 SETSESSIONUSER Privilege
付与または取り消された SETSESSIONUSER 特権。

0x0080000000000000 Exemption
付与または取り消された免除。

0x0100000000000000 Security label
付与または取り消されたセキュリティー・ラベル。

0x0200000000000000 WRITE with GRANT
特権の付与が許可されているグローバル変数を書き込むために付与または取り消された特権。

0x0400000000000000 Role Membership
付与または取り消されたロールのメンバーシップ。

0x0800000000000000 Role Membership with ADMIN Option
付与または取り消された ADMIN オプション付きのロールのメンバーシップ。

0x1000000000000000 READ
グローバル変数を読み取るために付与または取り消された特権。

0x2000000000000000 READ with GRANT
特権の付与が許可されているグローバル変数を読み取るために付与または取り消された特権。

0x4000000000000000 WRITE
グローバル変数を書き込むために付与または取り消された特権。

SYSADMIN イベントの監査レコード設計

次の表は、SYSADMIN イベントの監査レコード設計を示しています。

以下に監査レコードのサンプルを示します。

```
timestamp=1998-06-24-11.54.04.129923;
category=SYSADMIN;
audit event=DB2AUDIT;
event correlator=1;
event status=0;
userid=boss;authid=BOSS;
application id=*LOCAL.boss.980624155404;
application name=db2audit;
```

表 45. SYSADMIN イベントの監査レコード設計

名前	フォーマット	説明
Timestamp	CHAR(26)	監査イベントの日付と時刻。
Category	CHAR(8)	監査イベントの区分。可能な値は以下のとおりです。 SYSADMIN
Audit Event	VARCHAR(32)	特定の監査イベント。 有効な値を、この表に続くリストに示しています。
Event Correlator	INTEGER	監査対象の操作の相関 ID。単一イベントにどの監査レコードが関連しているかを識別するために使用できます。
Event Status	INTEGER	監査イベントの状況、次のような 1 つの SQLCODE で表されます。 成功イベント > = 0 失敗イベント < 0
Database Name	CHAR(8)	どのイベントが生成されたかを示すデータベース名。インスタンス・レベルの監査イベントであった場合にはブランクとなります。
User ID	VARCHAR(1024)	監査イベントの時刻でのユーザー ID。
Authorization ID	VARCHAR(128)	監査イベントの時刻での許可 ID。
Origin Node Number	SMALLINT	監査イベントが発生したノード番号。
Coordinator Node Number	SMALLINT	コーディネーター・ノードのノード番号。
Application ID	VARCHAR(255)	監査イベントが発生した時刻で使用していたアプリケーション ID。
Application Name	VARCHAR(1024)	監査イベントが発生した時刻で使用していたアプリケーション名。
Package Schema	VARCHAR(128)	監査イベントの時刻で使用していたパッケージのスキーマ。
Package Name	VARCHAR(128)	監査イベントが発生した時刻で使用していたパッケージ名。
Package Section Number	SMALLINT	監査イベントが発生した時刻で使用されていたパッケージのセクション番号。
Package Version	VARCHAR (64)	監査イベントが発生した時刻で使用していたパッケージのバージョン。
Local Transaction ID	VARCHAR(10) FOR BIT DATA	監査イベントが発生した時刻で使用していたローカル・トランザクション ID。これは、トランザクション・ログの一部となる SQLU_TID 構造体です。
Global Transaction ID	VARCHAR(30) FOR BIT DATA	監査イベントが発生した時刻で使用していたグローバル・トランザクション ID。これは、トランザクション・ログの一部となる SQLP_GXID 構造体のデータ・フィールドです。
Client User ID	VARCHAR(255)	監査イベントが発生した時刻の CURRENT CLIENT USERID 特殊レジスターの値。

表 45. SYSADMIN イベントの監査レコード設計 (続き)

名前	フォーマット	説明
Client Workstation Name	VARCHAR(255)	監査イベントが発生した時刻の CURRENT CLIENT_WRKSTNNAME 特殊レジスターの値。
Client Application Name	VARCHAR(255)	監査イベントが発生した時刻の CURRENT CLIENT_APPLNAME 特殊レジスターの値。
Client Accounting String	VARCHAR(255)	監査イベントが発生した時刻の CURRENT CLIENT_ACCTNG 特殊レジスターの値。
Trusted Context Name	VARCHAR(128)	トラステッド接続に関連付けられたトラステッド・コンテキストの名前。
Connection Trust Type	INTEGER	可能な値は以下のとおりです。 IMPLICIT_TRUSTED_CONNECTION EXPLICIT_TRUSTED_CONNECTION
Role Inherited	VARCHAR(128)	トラステッド接続を介して継承したロール。

SYSADMIN 監査イベント

次の表は、有効な SYSADMIN 監査イベントのリストです。

表 46. SYSADMIN 監査イベント

START_DB2
STOP_DB2
CREATE_DATABASE
ALTER_DATABASE
DROP_DATABASE
UPDATE_DBM_CFG
UPDATE_DB_CFG
CREATE_TABLESPACE
DROP_TABLESPACE
ALTER_TABLESPACE
RENAME_TABLESPACE
CREATE_NODEGROUP
DROP_NODEGROUP
ALTER_NODEGROUP
CREATE_BUFFERPOOL
DROP_BUFFERPOOL
ALTER_BUFFERPOOL
CREATE_EVENT_MONITOR
DROP_EVENT_MONITOR
ENABLE_MULTIPAGE
MIGRATE_DB_DIR
DB2TRC
DB2SET
ACTIVATE_DB
ADD_NODE
BACKUP_DB
CATALOG_NODE
CATALOG_DB
CATALOG_DCS_DB
CHANGE_DB_COMMENT
DEACTIVATE_DB
DROP_NODE_VERIFY
FORCE_APPLICATION
GET_SNAPSHOT
LIST_DRDA_INDOUBT_TRANSACTIONS
MIGRATE_DB
RESET_ADMIN_CFG
RESET_DB_CFG
RESET_DBM_CFG
RESET_MONITOR
RESTORE_DB
ROLLFORWARD_DB
SET_RUNTIME_DEGREE
SET_TABLESPACE_CONTAINERS
UNCATALOG_DB
UNCATALOG_DCS_DB
UNCATALOG_NODE
UPDATE_ADMIN_CFG
UPDATE_MON_SWITCHES
LOAD_TABLE
DB2AUDIT
SET_APPL_PRIORITY
CREATE_DB_AT_NODE
KILLDBM
MIGRATE_SYSTEM_DIRECTORY
DB2REMOT
DB2AUD
MERGE_DBM_CONFIG_FILE
UPDATE_CLI_CONFIGURATION
OPEN_TABLESPACE_QUERY
SINGLE_TABLESPACE_QUERY
CLOSE_TABLESPACE_QUERY
FETCH_TABLESPACE
OPEN_CONTAINER_QUERY
FETCH_CONTAINER_QUERY
CLOSE_CONTAINER_QUERY
GET_TABLESPACE_STATISTICS
DESCRIBE_DATABASE
ESTIMATE_SNAPSHOT_SIZE
READ_ASYNC_LOG_RECORD
PRUNE_RECOVERY_HISTORY
UPDATE_RECOVERY_HISTORY
QUIESCE_TABLESPACE
UNLOAD_TABLE
UPDATE_DATABASE_VERSION
CREATE_INSTANCE
DELETE_INSTANCE
SET_EVENT_MONITOR
GRANT_DBADM
REVOKE_DBADM
GRANT_DB_AUTHORITIES
REVOKE_DB_AUTHORITIES
REDISTRIBUTE_NODEGROUP

VALIDATE イベントの監査レコード設計

次の表は、VALIDATE イベントの監査レコードの形式を示しています。

以下に監査レコードのサンプルを示します。

```
timestamp=2007-05-07-10.30.51.585626;
category=VALIDATE;
audit event=AUTHENTICATION;
event correlator=1;
event status=0;
userid=newton;
authid=NEWTON;
execution id=gstager;
application id=*LOCAL.gstager.070507143051;
application name=db2bp;
auth type=SERVER;
plugin name=IBMOSauthserver;
```

表 47. VALIDATE イベントの監査レコード設計

名前	フォーマット	説明
Timestamp	CHAR(26)	監査イベントの日付と時刻。
Category	CHAR(8)	監査イベントの区分。可能な値は以下のとおりです。 VALIDATE
Audit Event	VARCHAR(32)	特定の監査イベント。 可能な値は AUTHENTICATE および GET_USERMAPPING_FROM_PLUGIN です。 **注:** バージョン 9.5 以降では、GET_GROUPS、GET_USERID、および CHECK_GROUP_MEMBERSHIP は生成されません。
Event Correlator	INTEGER	監査対象の操作の相関 ID。単一イベントにどの監査レコードが関連しているかを識別するために使用できます。
Event Status	INTEGER	監査イベントの状況、次のような 1 つの SQLCODE で表されます。 成功イベント > = 0 失敗イベント < 0
Database Name	CHAR(8)	どのイベントが生成されたかを示すデータベース名。インスタンス・レベルの監査イベントであった場合にはブランクとなります。
User ID	VARCHAR(1024)	監査イベントの時刻でのユーザー ID。
Authorization ID	VARCHAR(128)	監査イベントの時刻での許可 ID。
Execution ID	VARCHAR(1024)	監査イベントの時刻で使用していた実行 ID。
Origin Node Number	SMALLINT	監査イベントが発生したノード番号。
Coordinator Node Number	SMALLINT	コーディネーター・ノードのノード番号。
Application ID	VARCHAR(255)	監査イベントが発生した時刻で使用していたアプリケーション ID。
Application Name	VARCHAR(1024)	監査イベントが発生した時刻で使用していたアプリケーション名。
Authentication Type	VARCHAR(32)	監査イベントの時刻での認証タイプ。
Package Schema	VARCHAR(128)	監査イベントの時刻で使用していたパッケージのスキーマ。
Package Name	VARCHAR(128)	監査イベントが発生した時刻で使用していたパッケージ名。
Package Section Number	SMALLINT	監査イベントが発生した時刻で使用されていたパッケージのセクション番号。
Package Version	VARCHAR (64)	監査イベントが発生した時刻で使用していたパッケージのバージョン。
Plug-in Name	VARCHAR(32)	監査イベントが発生した時点で使用されていたプラグインの名前。

表 47. *VALIDATE* イベントの監査レコード設計 (続き)

名前	フォーマット	説明
Local Transaction ID	VARCHAR(10) FOR BIT DATA	監査イベントが発生した時刻で使用していたローカル・トランザクション ID。これは、トランザクション・ログの一部となる SQLU_TID 構造体です。
Global Transaction ID	VARCHAR(30) FOR BIT DATA	監査イベントが発生した時刻で使用していたグローバル・トランザクション ID。これは、トランザクション・ログの一部となる SQLP_GXID 構造体のデータ・フィールドです。
Client User ID	VARCHAR(255)	監査イベントが発生した時刻の CURRENT CLIENT USERID 特殊レジスターの値。
Client Workstation Name	VARCHAR(255)	監査イベントが発生した時刻の CURRENT CLIENT_WRKSTNNAME 特殊レジスターの値。
Client Application Name	VARCHAR(255)	監査イベントが発生した時刻の CURRENT CLIENT_APPLNAME 特殊レジスターの値。
Client Accounting String	VARCHAR(255)	監査イベントが発生した時刻の CURRENT CLIENT_ACCTNG 特殊レジスターの値。
Trusted Context Name	VARCHAR(128)	トラステッド接続に関連付けられたトラステッド・コンテキストの名前。
Connection Trust Type	INTEGER	可能な値は以下のとおりです。 IMPLICIT_TRUSTED_CONNECTION EXPLICIT_TRUSTED_CONNECTION
Role Inherited	VARCHAR(128)	トラステッド・コンテキストを介して継承したロールの名前。

CONTEXT イベントの監査レコード設計

次の表は、CONTEXT イベントの監査レコード設計を示しています。

以下に監査レコードのサンプルを示します。

```
timestamp=1998-06-24-08.42.41.476840;
category=CONTEXT;
audit event=EXECUTE_IMMEDIATE;
event correlator=3;
database=FOO;
userid=boss;
authid=BOSS;
application id=*LOCAL.newton.980624124210;
application name=testapp;
package schema=NULLID;
package name=SQLC28A1;
package section=203;
text=create table audit(c1 char(10), c2 integer);
```

表 48. *CONTEXT* イベントの監査レコード設計

名前	フォーマット	説明
Timestamp	CHAR(26)	監査イベントの日付と時刻。
Category	CHAR(8)	監査イベントの区分。可能な値は以下のとおりです。 CONTEXT
Audit Event	VARCHAR(32)	特定の監査イベント。 有効な値を、この表に続くリストに示しています。

表 48. CONTEXT イベントの監査レコード設計 (続き)

名前	フォーマット	説明
Event Correlator	INTEGER	監査対象の操作の相関 ID。単一イベントにどの監査レコードが関連しているかを識別するために使用できます。
Database Name	CHAR(8)	どのイベントが生成されたかを示すデータベース名。インスタンス・レベルの監査イベントであった場合にはブランクとなります。
User ID	VARCHAR(1024)	監査イベントの時刻でのユーザー ID。 監査イベントが SWITCH_USER である場合、このフィールドには、切り替え後のユーザー ID が表示されます。
Authorization ID	VARCHAR(128)	監査イベントの時刻での許可 ID。 監査イベントが SWITCH_USER である場合、このフィールドには、切り替え後の許可 ID が表示されます。
Origin Node Number	SMALLINT	監査イベントが発生したノード番号。
Coordinator Node Number	SMALLINT	コーディネーター・ノードのノード番号。
Application ID	VARCHAR(255)	監査イベントが発生した時刻で使用していたアプリケーション ID。
Application Name	VARCHAR(1024)	監査イベントが発生した時刻で使用していたアプリケーション名。
Package Schema	VARCHAR(128)	監査イベントの時刻で使用していたパッケージのスキーマ。
Package Name	VARCHAR(128)	監査イベントが発生した時刻で使用していたパッケージ名。
Package Section Number	SMALLINT	監査イベントが発生した時刻で使用されていたパッケージのセクション番号。
Statement Text	CLOB(8M)	適用できる場合には、SQL または XQuery ステートメントのテキストです。 SQL または XQuery ステートメントのテキストが使用可能でない場合、NULL となります。
Package Version	VARCHAR (64)	監査イベントが発生した時刻で使用していたパッケージのバージョン。
Local Transaction ID	VARCHAR(10) FOR BIT DATA	監査イベントが発生した時刻で使用していたローカル・トランザクション ID。これは、トランザクション・ログの一部となる SQLU_TID 構造体です。
Global Transaction ID	VARCHAR(30) FOR BIT DATA	監査イベントが発生した時刻で使用していたグローバル・トランザクション ID。これは、トランザクション・ログの一部となる SQLP_GXID 構造体のデータ・フィールドです。
Client User ID	VARCHAR(255)	監査イベントが発生した時刻の CURRENT CLIENT USERID 特殊レジスターの値。
Client Workstation Name	VARCHAR(255)	監査イベントが発生した時刻の CURRENT CLIENT_WRKSTNNAME 特殊レジスターの値。
Client Application Name	VARCHAR(255)	監査イベントが発生した時刻の CURRENT CLIENT_APPLNAME 特殊レジスターの値。
Client Accounting String	VARCHAR(255)	監査イベントが発生した時刻の CURRENT CLIENT_ACCTNG 特殊レジスターの値。
Trusted Context Name	VARCHAR(128)	トラステッド接続に関連付けられたトラステッド・コンテキストの名前。
Connection Trust Type	INTEGER	可能な値は以下のとおりです。 IMPLICIT_TRUSTED_CONNECTION EXPLICIT_TRUSTED_CONNECTION

表 48. CONTEXT イベントの監査レコード設計 (続き)

名前	フォーマット	説明
Role Inherited	VARCHAR(128)	トラステッド接続を介して継承したロール。

CONTEXT 監査イベント

次の表は、有効な CONTEXT 監査イベントのリストです。

表 49. CONTEXT 監査イベント

CONNECTCONNECT_RESET	SET_APPL_PRIORITY
ATTACH	RESET_DB_CFG
DETACH	GET_DB_CFG
DARI_START	GET_DFLT_CFG
DARI_STOP	UPDATE_DBM_CFG
BACKUP_DB	SET_MONITOR
RESTORE_DB	GET_SNAPSHOT
ROLLFORWARD_DB	ESTIMATE_SNAPSHOT_SIZE
OPEN_TABLESPACE_QUERY	RESET_MONITOR
FETCH_TABLESPACE	OPEN_HISTORY_FILE
CLOSE_TABLESPACE_QUERY	CLOSE_HISTORY_FILE
OPEN_CONTAINER_QUERY	FETCH_HISTORY_FILE
CLOSE_CONTAINER_QUERY	SET_RUNTIME_DEGREE
FETCH_CONTAINER_QUERY	UPDATE_AUDIT
SET_TABLESPACE_CONTAINERS	DBM_CFG_OPERATION
GET_TABLESPACE_STATISTIC	DISCOVER
READ_ASYNC_LOG_RECORD	OPEN_CURSOR
QUIESCE_TABLESPACE	CLOSE_CURSOR
LOAD_TABLE	FETCH_CURSOR
UNLOAD_TABLE	EXECUTEEXECUTE_IMMEDIATE
UPDATE_RECOVERY_HISTORY	PREPARE
PRUNE_RECOVERY_HISTORY	DESCRIBE
SINGLE_TABLESPACE_QUERY	BIND
LOAD_MSG_FILE	REBIND
UNQUIESCE_TABLESPACE	RUNSTATS
ENABLE_MULTIPAGE	REORG
DESCRIBE_DATABASE	REDISTRIBUTE
DROP_DATABASE	COMMIT
CREATE_DATABASE	ROLLBACK
ADD_NODE	REQUEST_ROLLBACK
FORCE_APPLICATION	IMPLICIT_REBIND
	EXTERNAL_CANCEL
	SWITCH_USER

EXECUTE イベントの監査レコード設計

次の表は、EXECUTE 区分の一部として監査されるすべてのフィールドを説明しています。

以下に監査レコードのサンプルを示します。

注: 他の監査区分とは異なり、EXECUTE 区分の場合は、監査ログが表形式で表示される際に 1 つのイベントの記述が複数行にわたって示されることがあります。1 行目のレコードは主要なイベントについて記述しており、イベント列にはキーワード STATEMENT が含まれています。残りの行は、パラメーター・マーカーやホスト変数について、パラメーターごとに 1 つの行を使用して記述します。これらの行のイベント列には、キーワード DATA が含まれています。監査ログがレポート形式で表示される際は、レコードは 1 つになりますが、「Statement Value」には複数の項目が表示されます。DATA キーワードは、表形式の場合にのみ表示されます。

```
timestamp=2006-04-10-13.20.51.029203;
  category=EXECUTE;
  audit event=STATEMENT;
  event correlator=1;
  event status=0;
  database=SAMPLE;
  userid=smith;
  authid=SMITH;
  session authid=SMITH;
  application id=*LOCAL.prodrig.060410172044;
  application name=myapp;
  package schema=NULLID;
  package name=SQLC2F0A;
  package section=201;
  uow id=2;
  activity id=3;
  statement invocation id=0;
  statement nesting level=0;
  statement text=SELECT * FROM DEPARTMENT WHERE DEPTNO = ? AND DEPTNAME = ?;
  statement isolation level=CS;
  compilation environment=
    isolation level=CS
    query optimization=5
    min_dec_div_3=NO
    degree=1
    sqlrules=DB2
    refresh age=+00000000000000.000000
    schema=SMITH
    maintained table type=SYSTEM
    resolution timestamp=2006-06-29-20.32.13.000000
    federated asynchrony=0;
  value index=0;
  value type=CHAR;
  value data=C01;
  value index=1;
  value type=VARCHAR;
  value index=INFORMATION CENTER;
```

表 50. EXECUTE イベントの監査レコード設計

名前	フォーマット	DESCRIPTION
Timestamp	CHAR(26)	監査イベントの日付と時刻。
Category	CHAR(8)	監査イベントの区分。可能な値は EXECUTE です。

表 50. EXECUTE イベントの監査レコード設計 (続き)

名前	フォーマット	DESCRIPTION
Audit Event	VARCHAR(32)	特定の監査イベント。 可能な値は以下のとおりです。 • STATEMENT: SQL ステートメントの実行。 • DATA: ステートメント用のホスト変数またはパラメーター・マーカーのデータ値。 このイベントは、ステートメントに含まれる各ホスト変数やパラメーター・マーカーごとに繰り返されます。DATA は、区切りファイルから監査ログを抽出するときにのみ表示されます。 • COMMIT: COMMIT ステートメントの実行。 • ROLLBACK: ROLLBACK ステートメントの実行。 • SAVEPOINT: SAVEPOINT ステートメントの実行。 • RELEASE SAVEPOINT: RELEASE SAVEPOINT ステートメントの実行。 • GLOBAL COMMIT: グローバル・トランザクション内での COMMIT の実行。 • GLOBAL ROLLBACK: グローバル・トランザクション内での ROLLBACK の実行。 • CONNECT: データベース接続の確立。 • CONNECT RESET: データベース接続の終了。 • SWITCH USER: トラステッド接続内でのユーザーの切り替え。
Event Correlator	INTEGER	監査対象の操作の相関 ID。単一イベントにどの監査レコードが関連しているかを識別するために使用できます。

表 50. EXECUTE イベントの監査レコード設計 (続き)

名前	フォーマット	DESCRIPTION
Event Status	INTEGER	監査イベントの状況、次のような 1 つの SQLCODE で表されます。成功イベント > = 0 失敗イベント < 0
Database Name	CHAR(8)	どのイベントが生成されたかを示すデータベース名。インスタンス・レベルの監査イベントであった場合にはブランクとなります。
User ID	VARCHAR(1024)	監査イベントの時刻でのユーザー ID。
Authorization ID	VARCHAR(128)	監査イベントの時刻でのステートメント許可 ID。
Session Authorization ID	VARCHAR(128)	監査イベントの時刻でのセッション許可 ID。
Origin Node Number	SMALLINT	監査イベントが発生したノード番号。
Coordinator Node Number	SMALLINT	コーディネーター・ノードのノード番号。
Application ID	VARCHAR(255)	監査イベントが発生した時刻で使用していたアプリケーション ID。
Application Name	VARCHAR(1024)	監査イベントが発生した時刻で使用していたアプリケーション名。
Client User ID	VARCHAR(255)	監査イベントが発生した時刻の CURRENT CLIENT USERID 特殊レジスターの値。
Client Accounting String	VARCHAR(255)	監査イベントが発生した時刻の CURRENT CLIENT_ACCTNG 特殊レジスターの値。
Client Workstation Name	VARCHAR(255)	監査イベントが発生した時刻の CURRENT CLIENT_WRKSTNNAME 特殊レジスターの値。
Client Application Name	VARCHAR(255)	監査イベントが発生した時刻の CURRENT CLIENT_APPLNAME 特殊レジスターの値。
Trusted Context Name	VARCHAR(128)	トラステッド接続に関連付けられたトラステッド・コンテキストの名前。

表 50. EXECUTE イベントの監査レコード設計 (続き)

名前	フォーマット	DESCRIPTION
Connection Trust type	INTEGER	可能な値は以下のとおりです。IMPLICIT_TRUSTED_CONNECTION および EXPLICIT_TRUSTED_CONNECTION
Role Inherited	VARCHAR(128)	トラステッド接続を介して継承したロール。
Package Schema	VARCHAR(128)	監査イベントの時刻で使用していたパッケージのスキーマ。
Package Name	VARCHAR(128)	監査イベントが発生した時刻で使用していたパッケージ名。
Package Section	SMALLINT	監査イベントが発生した時刻で使用されていたパッケージのセクション番号。
Package Version	VARCHAR(164)	監査イベントが発生した時刻で使用していたパッケージのバージョン。
Local Transaction ID	VARCHAR(10) FOR BIT DATA	監査イベントが発生した時刻で使用していたローカル・トランザクション ID。これは、トランザクション・ログの一部となる SQLU_TID 構造体です。
Global Transaction ID	VARCHAR(30) FOR BIT DATA	監査イベントが発生した時刻で使用していたグローバル・トランザクション ID。これは、トランザクション・ログの一部となる SQLP_GXID 構造体のデータ・フィールドです。
UOW ID	BIGINT	アクティビティーが発生した作業単位の ID。この値は、作業単位ごとにアプリケーション ID 内で固有です。
Activity ID	BIGINT	作業単位内で固有のアクティビティー ID。

表 50. EXECUTE イベントの監査レコード設計 (続き)

名前	フォーマット	DESCRIPTION
Statement Invocation ID	BIGINT	SQL ステートメントが実行されたルーチン呼び出しの ID。値は、アプリケーションで現行のネスティング・レベルがアクティブであったときに発生した、そのレベルでのルーチン呼び出しの数を示します。このエレメントを「Statement Nesting Level」と合わせて使用することにより、特定の SQL ステートメントの呼び出しを一意的に識別することができます。
Statement Nesting Level	BIGINT	ステートメントが実行されていたときに有効であったネスティングまたは再帰のレベル。ネスティングの各レベルは、ストアード・プロシージャーやユーザー定義関数 (UDF) のネストされた、または再帰可能な呼び出しに対応しています。
Activity Type	VARCHAR(32)	アクティビティーのタイプ。 可能な値は以下のとおりです。 • READ_DML • WRITE_DML • DDL • CALL • NONE
Statement Text	CLOB(8M)	適用できる場合には、SQL または XQuery ステートメントのテキストです。
Statement Isolation Level	CHAR(8)	ステートメントが実行されていたときに有効であった分離の値。 可能な値は以下のとおりです。 • NONE (分離の指定なし) • UR (非コミット読み取り) • CS (カーソル固定) • RS (読み取り固定) • RR (反復可能読み取り)

表 50. EXECUTE イベントの監査レコード設計 (続き)

名前	フォーマット	DESCRIPTION
Compilation Environment Description	BLOB(8K)	SQL ステートメントのコンパイル時に使用されたコンパイル環境。このエレメントは、COMPILATION_ENV 表関数または SET COMPILATION ENVIRONMENT SQL ステートメントに入力として渡すことができます。
Rows Modified	INTEGER	以下の両方の結果として削除、挿入、または更新された行の総数。 • 削除操作成功後の制約の強制 • アクティブにされたトリガーが起動した SQL ステートメントの処理 コンパウンド SQL が呼び出される場合は、すべてのサブステートメントの、これに該当する行の数の集計が含まれます。場合によっては、エラーが発生したときに、内部エラーを示す負の値がこのフィールドに表示されることがあります。この値は、SQLCA の sqlerrd(5) フィールドと等価です。
Rows Returned	BIGINT	ステートメントによって戻される行の総数。
Savepoint ID	BIGINT	ステートメントが実行されていたときにそのステートメントで有効であったセーブポイント ID。「Audit Event」が SAVEPOINT、RELEASE_SAVEPOINT、または ROLLBACK_SAVEPOINT である場合、「Savepoint ID」は、それぞれ設定、解放、またはロールバックされるセーブポイントになります。
Statement Value Index	INTEGER	SQL ステートメントで使用される入力パラメーター・マーカーまたはホスト変数の位置。

表 50. EXECUTE イベントの監査レコード設計 (続き)

名前	フォーマット	DESCRIPTION
Statement Value Type	CHAR(16)	SQL ステートメントに関連付けられているデータ値のタイプのストリング表現。可能な値の例としては、INTEGER や CHAR が挙げられます。
Statement Value Data	CLOB(128K)	SQL ステートメントへのデータ値のストリング表現。LOB、LONG、XML、および構造化タイプのパラメーターは表示されません。日付、時刻、およびタイム・スタンプのフィールドは、ISO 形式で記録されます。

第 10 章 オペレーティング・システム・セキュリティーの操作

オペレーティング・システムには、データベース・インストールのセキュリティーをサポートするために使用できるセキュリティー・フィーチャーが備えられています。

DB2 および Windowsセキュリティー

Windows ドメインは、特定の名前および固有の名前で参照されるクライアント・コンピューターおよびサーバー・コンピューターの配置であり、 Security Access Manager (SAM) と呼ばれる単一のユーザー・アカウント・データベースを共用します。ドメイン内のコンピューターのうちの 1 つが、ドメイン・コントローラーです。ドメイン・コントローラーは、ユーザー・ドメインの対話のすべての面を管理します。

ドメイン・コントローラーは、ドメイン・アカウントにログオンするユーザーを認証するために、ドメイン・ユーザー・アカウント・データベース内の情報を使用します。ドメインごとに、1 つのドメイン・コントローラーが 1 次ドメイン・コントローラー (PDC) になります。ドメイン内には、1 次ドメイン・コントローラーが存在しない場合、または 1 次ドメイン・コントローラーが使用不可の場合に、ユーザー・アカウントを認証するバックアップ・ドメイン・コントローラー (BDC) が存在する場合があります。バックアップ・ドメイン・コントローラーは、PDC のマスター・コピーと定期的に同期化される Windows Security Account Manager (SAM) データベースのコピーを保持しています。

ユーザー・アカウント、ユーザー ID、およびパスワードは、 1 次ドメイン・コントローラーに定義するだけで、ドメイン・リソースにアクセスできるようになります。

注: CONNECT ステートメントと ATTACH コマンドは、2 部構成のユーザー ID をサポートしています。 SAM 互換ユーザー ID の修飾子は、最大長 15 文字の 'Domain¥User' スタイルの名前です。

Windows サーバーのインストール時のセットアップ手順の中で、以下を作成するように選択できます。

- 1 次ドメイン・コントローラー (新しいドメイン内)
- バックアップ・ドメイン・コントローラー (既知のドメイン内)
- スタンドアロン・サーバー (既知のドメイン内)

新しいドメイン内で「コントローラー」を選択すると、そのサーバーは 1 次ドメイン・コントローラーになります。

ユーザーはローカル・マシンにログオンすることができます。あるいは、 Windows ドメイン中にマシンをインストールしているならば、ユーザーはそのドメインにログオンできます。ユーザーを認証するために、 DB2 は最初にローカル・マシンのリスト、次に現在のドメインのドメイン・コントローラー、最後にドメイン・コントローラーを認識する承認されたドメインをチェックします。

この動作の方法を説明するために、DB2 インスタンスがサーバー認証を必要とすると仮定します。構成は、以下のとおりです。

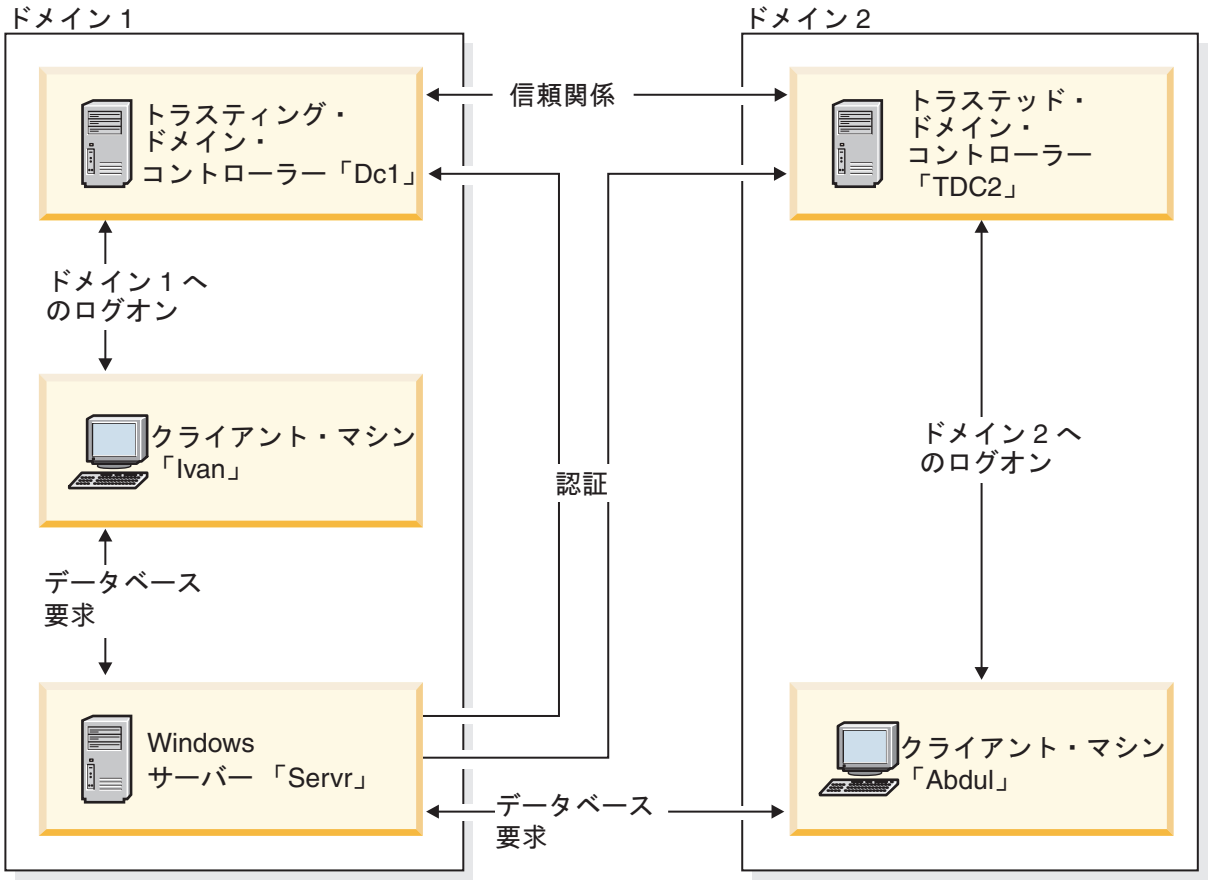


図 5. Windows のドメインを使用した認証

各マシンには、セキュリティー・データベース (セキュリティー・アクセス管理 (SAM)) があります。 DC1 はドメイン・コントローラーで、そのクライアント・マシンの Ivan、 DB2 サーバーの Servr が登録されています。 TDC2 は DC1 に承認されたドメインで、クライアント・マシンの Abdul は TDC2 のドメインのメンバーです。

認証シナリオ

サーバー認証を使用するシナリオ (Windows)

1. Abdul は、TDC2 ドメインにログオンします (つまり、 TDC2 SAM データベースに認識されています)。
2. 次に Abdul は、次のように入力して SRV3 上に常駐するためにカタログされた DB2 データベースに接続します。

   ```
   db2 connect to remotedb user Abdul using fredpw
   ```

3. SRV3 は、Abdul が認識されている場所を判別します。この情報を検出するのに使われる API は、最初にローカル・マシン (SRV3)、次にドメイン・コントローラー (DC1) を検索し、最後に承認されたドメインを検索しようとします。ユー

ザー名 Abdul が TDC2 上で検出されます。この検索順序には、ユーザーとグループに単一ネーム・スペースが必要です。

4. 次に SRV3 は、次のように実行します。
   a. TDC2 を使ってユーザー名とパスワードの妥当性を検査します。
   b. TDC2 に照会することによって、Abdul が管理者かどうかを検出します。
   c. TDC2 に照会することによって、Abdul のグループすべてを列挙します。

クライアント認証および Windows クライアント・マシンを使用するシナリオ

1. 管理者の Dale は、SRV3 にログオンし、クライアントに対するデータベース・インスタンスの認証を変更します。

```
db2 update dbm cfg using authentication client
db2stop
db2start
```

2. Windows クライアント・マシンで、Ivan は DC1 ドメインにログオンします (つまり、 DC1 SAM データベースに認識されています)。
3. 次に Ivan は、次のように入力して SRV3 上に常駐するためにカタログされた DB2 データベースに接続します。

```
DB2 CONNECT to remotedb user Ivan using johnpw
```

4. Ivan のマシンは、ユーザー名とパスワードの妥当性を検査します。この情報を検出するのに使われる API は、最初にローカル・マシン (Ivan)、次にドメイン・コントローラー (DC1) を検索し、最後に承認されたドメインを検索しようとします。ユーザー名 Ivan が DC1 上で検出されます。
5. 次に Ivan のマシンは、DC1 を使ってユーザー名とパスワードの妥当性を検査します。
6. 次に SRV3 は、次のように実行します。
   a. Ivan が認識された場所を判別します。
   b. DC1 に照会することによって、Ivan は管理者かどうかを検出します。
   c. DC1 に照会することによって、Ivan のすべてのグループを列挙します。

注: DB2 データベースに接続してみる前に、必ず DB2 セキュリティー・サービスを始動してください。セキュリティー・サービスは、Windows のインストールの一部としてインストールされます。次いで DB2 がインストールされ、Windows サービスとして「登録」されますが、デフォルトでは、自動的に開始されません。DB2 セキュリティー・サービスを始動するには、 `NET START DB2NTSECSERVER` コマンドを入力してください。

Windows でのグローバル・グループのサポート

DB2 データベース・システムはグローバル・グループをサポートします。

グローバル・グループを使用するために、ローカル・グループ内にグローバル・グループを組み込む必要があります。あるユーザーがメンバーとなっているグループを DB2 データベース・マネージャーがすべて列挙するとき、そのユーザーが間接的にメンバーになっているローカル・グループもまたリストされます (そのグループは、1 つまたは複数のローカル・グループのメンバーになっているグローバル・グループ内にあるため)。

グローバル・グループは、以下の 2 つの状態で使用されます。

- ローカル・グループに組み込まれた状態。ローカル・グループに許可を付与する必要があります。
- ドメイン・コントローラーに組み込まれた状態。グローバル・グループに許可を付与する必要があります。

DB2 データベース・システムでのバックアップ・ドメイン・コントローラーの使用

DB2 データベース・システム用に使用しているサーバーがバックアップ・ドメイン・コントローラーとしても動作する場合、バックアップ・ドメイン・コントローラーを使用するよう DB2 データベース・システムを構成すれば、 DB2 データベースのパフォーマンスを向上させ、ネットワーク・トラフィックを削減することができます。

DB2DMNBCKCTLR レジストリー変数を設定することによって、 DB2 データベース・システムにバックアップ・ドメイン・コントローラーを指定します。

DB2 データベース・サーバーがバックアップ・ドメイン・コントローラーになっている場合、そのドメイン・ネームが分かれば、次を使用します。

```
db2dmnbckctlr=<domain_name>
```

この場合、domain_name は大文字でなければなりません。

ローカル・マシンがバックアップ・ドメイン・コントローラーになっている場合、DB2 データベース・システムにそのドメインを判別させるには、以下を使用します。

```
DB2DMNBCKCTLR=?
```

注: デフォルトでは、DB2 データベース・マネージャーは既存のバックアップ・ドメイン・コントローラーを使用しません。バックアップ・ドメイン・コントローラーは 1 次ドメイン・コントローラーと同期しないことがあり、機密漏れが生じることがあるためです。 1 次ドメイン・コントローラーのセキュリティー・データベースが更新されても、その変更内容がバックアップ・ドメイン・コントローラーに伝搬されていない場合に、ドメイン・コントローラーが同期しなくなります。この事態は、ネットワーク待ち時間が生じた場合や、コンピューターのブラウザー・サービスが作動可能でない場合に起こることがあります。

Windows での DB2 のユーザー認証

ユーザー名およびグループ名に関する制約事項 (Windows)

Windows 環境に固有の制約事項がいくつかあります。ただし、DB2 の一般的なオブジェクト命名規則も適用されます。

- Windows 環境では、ユーザー名に大文字小文字の区別はありません。ただし、パスワードには大文字小文字の区別があります。
- ユーザー名やグループ名には大文字と小文字の両方を含めることができます。ただし、DB2 データベース内で使用されるとき、通常は大文字に変換されます。たとえば、データベースに接続してから表 schema1.table1 を作成した場合、この表

はデータベース内に SCHEMA1.TABLE1 として保管されます。 (小文字のオブジェクト名を使用する場合は、コマンド行プロセッサーからコマンドを発行するときにオブジェクト名を引用符で囲むか、あるいはサード・パーティーの ODBC フロントエンド・ツールを使用します。)

- ユーザーが属すことのできるグループの数は 64 までです。
- DB2 データベース・マネージャーは単一のネーム・スペースをサポートします。つまり、トラステッド・ドメイン環境で実行する場合は、同じ名前のユーザー・アカウントを複数のドメインに置いたり、サーバー・マシンのローカル SAM や別のドメインに置いたりすることはできません。

Windows でのグループ認証およびユーザー認証

ユーザーは、「ユーザ マネージャ」という Windows 管理ツールを使用して、Windows 上に定義されます。他のアカウント (メンバーとも呼ばれる) が入っているアカウントは、グループです。

グループを使用すれば、Windows 管理者は、権利および許可をグループ内のユーザーに一度に付与できるようになり、各ユーザーを個別に保守する必要がなくなります。グループは、ユーザー・アカウントと同様、 Security Access Manager (SAM) データベース内で定義され保守されます。

グループには次の 2 つの種類があります。

- ローカル・グループ。ローカル・グループには、ローカル・アカウント・データベース内で作成されたユーザー・アカウントを入れることができます。ローカル・グループがドメインの一部であるマシン上に存在する場合は、ローカル・グループには、 Windows ドメインのドメイン・アカウントおよびグループを入れることもできます。ローカル・グループをワークステーション上に作成する場合は、ローカル・グループはそのワークステーション固有です。
- グローバル・グループ。グローバル・グループは、ドメイン・コントローラー上にのみ存在し、ドメインの SAM データベースのユーザー・アカウントが含まれています。つまり、グローバル・グループには、グローバル・グループが作成されたドメインのユーザー・アカウントだけを入れることができます。他のグループをメンバーとして入れることはできません。グローバル・グループは、グローバル・グループが所属するドメインのサーバーおよびワークステーションと、信用のあるドメインで使用できます。

Windows でのドメイン間の信頼関係

信頼関係は、2 つのドメイン間の管理および通信リンクです。 2 つのドメイン間に信頼関係があれば、ユーザー・アカウントおよびグローバル・グループを、アカウントが定義されたドメインではないドメインで使用できるようになります。

アカウント情報は、トラステッド・ドメイン内に認証されずに存在しているユーザー・アカウントおよびグローバル・グループの権利や許可を検証するために共用されます。信頼関係は、2 つ以上のドメインを単一の管理単位にまとめることにより、ユーザーの管理を単純化します。

信頼関係には次の 2 つのドメインがあります。

- トラスティング・ドメイン。このドメインは、ユーザーを認証してもらうために別のドメインを信頼しています。

- トラステッド・ドメイン。このドメインは、別のドメインに代わってユーザーを認証します。

信頼関係は他動的なものではありません。つまり、ドメイン間で双方向の信頼関係を明示的に確立する必要があります。たとえば、トラスティング・ドメインは、必ずしもトラステッド・ドメインであるとは限りません。

DB2 データベース・システムと Windows セキュリティー・サービス

DB2 データベース・システムでは、ユーザー名とパスワードの認証が DB2 システム・コントローラーに統合されています。

セキュリティー・サービスが必要になるのは、認証 CLIENT 用に構成されたサーバーにクライアントが接続するときだけです。

DB2 のバックアップ・ドメイン・コントローラーへのインストール

Windows 環境では、プライマリー・コントローラーとバックアップ・コントローラーのどちらでもユーザーの認証を実行できます。この機能は、どのサイトにも 1 つの中央 1 次ドメイン・コントローラー (PDC) と、1 つまたは複数のバックアップ・ドメイン・コントローラー (BDC) とが配置されているような大規模な分散 LAN では重要です。ユーザーは認証のために 1 次ドメイン・コントローラーを呼び出さなくても、現在のサイトにあるバックアップ・ドメイン・コントローラーで認証を実行できます。

この場合、バックアップ・ドメイン・コントローラーを設けることの利点は、ユーザーの認証を短時間で行えることと、 BDC がない場合よりも LAN が混雑しないで済むことです。

以下の条件にあてはまる場合、認証は BDC で行うことができます。

- DB2 サーバー (Windows 版) がバックアップ・ドメイン・コントローラーにインストールされている場合。
- DB2DMNBCKCTLR プロファイル・レジストリー変数が正しく設定されている場合。

DB2DMNBCKCTLR プロファイル・レジストリー変数を設定しなかったりブランクに設定したりすると、DB2 サーバーは 1 次ドメイン・コントローラーで認証を実行します。

DB2DMNBCKCTLR に有効な宣言設定は ″?″ またはドメイン・ネームだけです。

DB2DMNBCKCTLR プロファイル・レジストリー変数が疑問符に設定されていて (DB2DMNBCKCTLR=?)、かつ以下の条件にあてはまる場合、DB2 サーバーは認証をバックアップ・ドメイン・コントローラーで実行します。

- cachedPrimaryDomain レジストリー値が、このマシンが属しているドメインの名前に設定されている。 (この設定は、「**HKEY_LOCAL_MACHINE**」--**>**「**Software**」**-->**「**Microsoft**」**-->**「**Windows NT**」**-->**「**Current Version**」**-->**「**WinLogon**」で確認できます。)

- サーバー・マネージャーによって、バックアップ・ドメイン・コントローラーがアクティブかつ使用可能であることが示されている。 (つまり、このマシンを表すアイコンがグレー表示にはなっていない。)
- DB2 サーバーのレジストリーによって、そのシステムが、指定されたドメイン上のバックアップ・ドメイン・コントローラーであることが示されている。

通常の状態であれば DB2DMNBCKCTLR=? 設定はたいてい正常に機能しますが、あらゆる環境で正常に機能することが保証されているわけではありません。ドメイン上のサーバーについて提供される情報は動的であるため、コンピューター・ブラウザーを実行して、この情報を常に正確かつ最新のものに保つ必要があります。大規模な LAN ではコンピューター・ブラウザーを実行できないことがあるため、サーバー・マネージャーの情報が最新のものではなくなる場合もあります。この場合、DB2 サーバーに認証をバックアップ・ドメイン・コントローラーで実行させる別の方法があります。それは、DB2DMNBCKCTLR=*xxx* を設定することです (*xxx* は DB2 サーバーの Windows ドメイン・ネーム)。このように設定されている場合、以下の条件を満たしていれば、認証がバックアップ・ドメイン・コントローラーで行われます。

- cachedPrimaryDomain レジストリー値が、このマシンが属しているドメインの名前に設定されている。 (この設定は、「**HKEY_LOCAL_MACHINE**」-->「**Software**」-->「**Microsoft**」-->「**Windows NT**」-->「**Current Version**」-->「**WinLogon**」で確認できます。)
- マシンが、指定されたドメイン用のバックアップ・ドメイン・コントローラーとして構成されている。 (マシンがバックアップ・ドメイン・コントローラーとしてセットアップされていても、他のドメインのためのものである場合、この設定はエラーになります。)

認証でのグループおよびドメイン・セキュリティーの使用 (Windows)

DB2 データベース・システムでは、特権を付与するときまたは権限レベルを定義する時に、ローカル・グループかグローバル・グループを指定できます。

ユーザーがグループのメンバーであると判断されるのは、そのユーザーのアカウントが、ローカルまたはグローバル・グループ内で明示的に定義されている場合、またはローカル・グループのメンバーになるように定義されているグローバル・グループのメンバーになることによって暗黙的に定義されている場合です。

DB2 データベース・マネージャーは、以下のタイプのグループをサポートします。

- ローカル・グループ
- グローバル・グループ
- ローカル・グループのメンバーとしてのグローバル・グループ

  DB2 データベース・マネージャーは、ユーザーの情報が含まれているセキュリティー・データベースを使用して、そのユーザーがメンバーとなっているローカル・グループとグローバル・グループを列挙します。 DB2 データベース・システムは、ユーザー・アカウントがどこにあるかに関係なく、 DB2 データベースがインストールされているローカル Windows サーバーでのみグループが列挙されるように設定することができます。それには以下のコマンドを使用します。

  – グローバル設定の場合:

```
db2set -g DB2_GRP_LOOKUP=local
```

– インスタンス設定の場合:

```
db2set -i <instance_name> DB2_GRP_LOOKUP=local
```

このコマンドの発行後、変更を有効にするには、 DB2 データベース・インスタンスを停止して開始する必要があります。次に、ローカル・グループを作成し、ドメイン・アカウントまたはグローバル・グループをそのローカル・グループに入れます。

設定されているすべての DB2 プロファイル・レジストリー変数を表示するには、次のように入力します。

```
db2set -all
```

DB2_GRP_LOOKUP プロファイル・レジストリー変数が local に設定されている場合、DB2 データベースはローカル・マシン上のユーザーのグループのみを列挙しようとします。そのユーザーがローカルまたはグローバル・グループのメンバーとして定義されていない場合、グループの列挙は失敗します。DB2 は、同じドメインの他のマシンやドメイン・コントローラーからユーザーのグループを列挙しようとは**しません**。

DB2_GRP_LOOKUP プロファイル・レジストリー変数が設定されていない場合、以下が行われます。

1. DB2 データベース・システムは最初に同じマシン上でユーザーを探そうとします。
2. ユーザー名がローカルで定義されている場合、そのユーザーの認証はローカルで行われます。
3. ユーザーがローカルで見つからなかった場合、DB2 データベース・システムは同じドメインの中からユーザー名を探そうとし、それでも見つからない場合は、トラステッド・ドメインから探そうとします。

リソース・ドメイン内で 1 次ドメイン・コントローラーまたはバックアップ・ドメイン・コントローラーであるマシン上で DB2 データベース・マネージャーが実行されている場合は、任意のドメイン・コントローラーを任意のトラステッド・ドメインに置くことができます。トラステッド・ドメイン内のバックアップ・ドメイン・コントローラーのドメインの名前は、ドメイン・コントローラーでなければ知ることができないためです。

DB2 データベース・マネージャーがドメイン・コントローラー上で実行されていない場合は、以下を発行する必要があります。

```
db2set -g DB2_GRP_LOOKUP=DOMAIN
```

このコマンドは、DB2 データベース・システムが属するドメイン内のドメイン・コントローラーを使用して、アカウント・ドメイン内のドメイン・コントローラーの名前を検索するように DB2 データベース・システムに通知します。つまり、DB2 データベースは特定のユーザー・アカウントがドメイン x に定義されていることを検出すると、ドメイン x のドメイン・コントローラーを探そうとするのではなく、その要求を DB2 が属するドメイン内のドメイン・コントローラーに送信します。

アカウント・ドメイン内のドメイン・コントローラーの名前が検出され、 DB2 データベースが実行されているマシンに戻されます。この方法には、以下の 2 つの利点があります。

1. 1 次ドメイン・コントローラーが使用できない場合に、最も近くにあるドメイン・コントローラーが検出される。
2. 1 次ドメイン・コントローラーが地理的に離れている場合は、最も近くにあるドメイン・コントローラーが検出される。

順序付けドメイン・リストを使用した認証

1 つのトラステッド・ドメイン・フォレスト内では、ユーザー ID が複数回にわたって定義される場合があります。トラステッド・ドメイン・フォレストとは、ネットワークを介して互いに関連している複数のドメインからなる集合です。

1 つのドメインのユーザーが、別のドメイン内の別のユーザーと同じユーザー ID を持つ可能性があります。そのような場合、以下の操作が困難になります。

- 同じユーザー ID を持つ複数のユーザーをそれぞれ別のドメインで認証する。
- グループに基づいて特権を付与または取り消すための、グループ・ルックアップ。
- パスワードの検証。
- ネットワーク・トラフィックの制御。

同じユーザー ID を持つ複数のユーザーがドメイン・フォレスト内でアクセスする場合の問題を防ぐには、 db2set およびレジストリー変数 DB2DOMAINLIST を使って定義される、順序付けドメイン・リストを使用する必要があります。順序を設定するときには、リストに含める複数のドメインをコンマで区切ります。ユーザー認証時に複数のドメインを検索する順序を決定するときには、十分に考慮する必要があります。

ドメイン・リストの下の方にあるドメインに含まれるユーザー ID がアクセスのために認証されるには、それらを名前変更しなければなりません。

ドメイン・リストを介してアクセスを制御することができます。たとえば、ユーザーのドメインがリストに含まれない場合、そのユーザーは接続を許可されません。

注: DB2DOMAINLIST レジストリー変数が有効になるのは、データベース・マネージャー構成で CLIENT 認証が設定され、 Windows ドメイン環境の Windows デスクトップからのシングル・サインオンでこの認証が必要とされる場合のみです。DB2DOMAINLIST は、いくつかのバージョンの DB2 サーバーでサポートされていますが、クライアントもサーバーも Windows 環境に組み込まれていなければ、DB2DOMAINLIST は強制されません。

ドメイン・セキュリティーのサポート (Windows)

以下の例は、DB2 データベース管理システムが Windows ドメイン・セキュリティーをどのようにサポートするかを示しています。ユーザー名とローカル・グループが同じドメイン上にあるため、接続は機能します。

以下のシナリオでは、ユーザー名とローカルまたはグローバル・グループが同じドメイン上にあるため、接続は機能します。

必ずしもユーザー名とローカルまたはグローバル・グループを、データベース・サーバーが実行されているドメインに定義する必要はありません。しかし、ユーザー名とローカルまたはグローバル・グループを同じドメインに定義する必要はあります。

表 51. ドメイン・コントローラーを使用した接続が成功する場合

Domain1	**Domain2**
Domain2 との間に信頼関係が存在している。	• Domain1 との間に信頼関係が存在している。 • ローカルまたはグローバル・グループ grp2 が定義されている。 • ユーザー名 id2 が定義されている。 • ユーザー名 id2 が grp2 のメンバーとなっている。

DB2 サーバーがこのドメインで実行されている。以下の DB2 コマンドがこのサーバーから発行される。
`REVOKE CONNECT ON db FROM public`
`GRANT CONNECT ON db TO GROUP grp2`
`CONNECT TO db USER id2`	

ローカルまたはグローバル・ドメインがスキャンされるが、 id2 は見つからない。ドメイン・セキュリティーがスキャンされる。	
	ユーザー名 id2 がこのドメインで見つかる。 DB2 は、このユーザー名についての追加情報 (つまり、このユーザー名がグループ grp2 のメンバーであるということ) を入手する。
ユーザー名とローカルまたはグローバル・グループが同じドメイン上にあるため、接続は機能する。	

アクセス・トークンを使用してWindows ユーザーのグループ情報を取得する

アクセス・トークンは、プロセスまたはスレッドのセキュリティー・コンテキストを説明するオブジェクトです。アクセス・トークン内の情報には、プロセスまたはスレッドに関連したユーザー・アカウントの識別および特権が含まれます。

ログオンすると、システムはユーザーのパスワードをセキュリティー・データベースに保管されている情報と比較して、それを検証します。パスワードが認証されると、システムはアクセス・トークンを生成します。ユーザーのために実行されるすべてのプロセスは、このアクセス・トークンのコピーを使用します。

アクセス・トークンは、キャッシュされた証明書に基づいて取得することもできます。システムに認証されると、その証明書はオペレーティング・システムによってキャッシュに入れられます。ドメイン・コントローラーにアクセスできないときは、キャッシュ内にある前回のログオン時のアクセス・トークンを参照できます。

アクセス・トークンには、ローカル・グループおよびさまざまなドメイン・グループ (グローバル・グループ、ドメイン・ローカル・グループ、およびユニバーサル・グループ) など、ユーザーが所属するすべてのグループに関する情報が含まれています。

注: アクセス・トークン・サポートは使用可能ですが、リモート接続を使用する場合、クライアント認証を使用するグループ・ルックアップはサポートされていません。

アクセス・トークン・サポートを使用可能にするには、 db2set コマンドを使用して DB2_GRP_LOOKUP レジストリー変数を更新する必要があります。このレジストリー変数を更新する際の選択項目には、以下のものがあります。

- TOKEN

  この選択項目は、ユーザー・アカウントが定義されたロケーションだけでなく、ローカル・マシンでも、アクセス・トークン・サポートがユーザーの属するすべてのグループを検索できるようにします (アカウントがドメインで定義されている場合)。

- TOKENLOCAL

  この選択項目は、DB2 データベース・サーバー上で、アクセス・トークン・サポートがユーザーの所属するすべてのローカル・グループを検索できるようにします。

- TOKENDOMAIN

  この選択項目は、ユーザー・アカウントが定義されたロケーションで、アクセス・トークン・サポートがユーザーの所属するすべてのグループを検索できるようにします。ロケーションは通常、ドメインまたは DB2 データベース・サーバーに対してローカルな場所にあります。

DB2_GRP_LOOKUP レジストリー変数を使用することを検討し、従来型のグループ列挙方法を使用して、DB2 データベース・システムがグループを検索する場所を示すために、グループ・ルックアップ・ロケーションを指定してください。例:

```
db2set DB2_GRP_LOOKUP=LOCAL,TOKENLOCAL
```

これにより、ローカル・グループを列挙するためのアクセス・トークン・サポートが使用可能になります。

```
db2set DB2_GRP_LOOKUP=,TOKEN
```

これにより、ユーザー ID が定義されたロケーションだけでなく、ローカル・マシンでもグループを列挙するためのアクセス・トークン・サポートが使用可能になります (アカウントがドメインで定義されている場合)。

```
db2set DB2_GRP_LOOKUP=DOMAIN,TOKENDOMAIN
```

これにより、ユーザー ID が定義されたロケーションで、ドメイン・グループを列挙するためのアクセス・トークン・サポートが使用可能になります。

アクセス・トークン・サポートは、CLIENT 認証を除くすべての認証タイプによって使用可能になります。

Windows プラットフォームでのユーザーのセキュリティーに関する考慮事項

システム管理 (SYSADM) 権限は、そのアカウントが定義されているマシンのローカル管理者グループに属している、有効なあらゆる DB2 データベース・ユーザー・アカウントに付与されます。

Windows ドメイン環境のデフォルトでは、インスタンスに対する SYSADM 権限を付与されるのは、ドメイン・コントローラーの管理者グループに属しているドメイン・ユーザーだけです。DB2 は必ずアカウントが定義されているマシンで許可を行うので、サーバーのローカル管理者グループにドメイン・ユーザーを追加しても、そのグループにはドメイン・ユーザー SYSADM 権限は付与されません。

注: Windows にあるようなドメイン環境では、DB2 は、要件と制約事項を満たし、ユーザー ID が属する最初の 64 グループのみを認証します。グループは 64 より多いという可能性もあります。

1 次ドメイン・コントローラー (PDC) の管理者グループにドメイン・ユーザーが追加されないようにするには、グローバル・グループを作成し、SYSADM 権限を付与するユーザー (ドメインとローカルの両方) を追加します。これを行うには、以下のコマンドを入力します。

```
DB2STOP
   DB2 UPDATE DBM CFG USING SYSADM_GROUP global_group
DB2START
```

Windows ローカル・システム・アカウントのサポート

Windows プラットフォーム (Windows ME を除く) で、DB2 データベース・システムは、ローカル暗黙接続があるローカル・システム・アカウント (LSA) のコンテキストで実行するアプリケーションをサポートします。

このアカウントの下で実行するアプリケーションを作成する開発者は、「SYS」で始まるスキーマ名のオブジェクトに関して DB2データベース・システムに制約があることに注意する必要があります。そのため、DB2 データベース・オブジェクトを作成する DDL がアプリケーションに含まれる場合、それらのアプリケーションは以下のように作成する必要があります。

- 静的照会では、QUALIFIER オプションの値をデフォルト以外のものにしてバインドする必要があります。
- 動的照会では、作成するオブジェクトを DB2 データベース・マネージャーによってサポートされるスキーマ名で明示的に修飾するか、CURRENT SCHEMA レジスターをDB2 データベース・マネージャーによってサポートされるスキーマ名に設定する必要があります。

LSA のグループ情報は、DB2 データベース・インスタンスが開始した後の最初のグループ・ルックアップ要求で収集され、インスタンスが再起動するまで更新されません。

注: ローカル・システム・アカウント (LSA) のコンテキストで実行されるアプリケーションは、Windows ME 以外のすべての Windows プラットフォームでサポートされています。

DB2ADMNS と DB2USERS グループの使用による拡張 Windows セキュリティー

DB2 データベース・マネージャーのサーバー・バージョンでは、拡張セキュリティーが暗黙的にデフォルトで *有効*です。しかし、クライアント・バージョンでは、拡張セキュリティーは暗黙的にデフォルトで *無効*です。有効にするには、インストール時に明示的に拡張セキュリティーを選択する必要があります。

クライアントに DB2 をインストールするときに拡張セキュリティーを使用可能にするには、「**DB2 オブジェクトのためにオペレーティング・システム・セキュリティーを使用可能にする**」パネルで「**オペレーティング・システム・セキュリティーを使用可能にする**」チェック・ボックスを選択します。インストーラーは、DB2ADMNS と DB2USERS という 2 つの新規グループを作成します。DB2ADMNS と DB2USERS はデフォルトのグループ名です。任意で、インストール時にこれらのグループに別の名前を付けることもできます。サイレント・インストールを選択した場合、インストール応答ファイル内でこれらの名前を変更することができます。システム上に既に存在するグループを使用するように選択すると、それらのグループの特権が変更されるので注意してください。 必要に応じて、以下の表にリストされている特権が与えられます。これらのグループは、オペレーティング・システム レベルでの保護のため使用されるもので、SYSADM、SYSMAINT、および SYSCTRL のような DB2 権限レベルとは関連付けられていないということを理解する必要があります。しかし、インストーラーや管理者の判断により、デフォルトの Administrator グループを使用する代わりに、データベース管理者は 1 つまたはすべての DB2 権限レベルに DB2ADMNS グループを使用することができます。 SYSADM グループを指定する場合は、DB2ADMNS グループにしてください。この設定は、インストール時、またはそれ以降に管理者が実行できます。

注: DB2 管理者グループ (DB2ADMNS、またはインストール時に選択した名前) と DB2 ユーザー・グループ (DB2USERS、またはインストール時に選択した名前) は、ローカル・グループとしても、ドメイン・グループとしても指定できます。ただし、両方のグループを同じタイプにする必要があります (つまり、両方をローカルにするか、両方をドメインにするかのどちらかです)。

コンピューター名を変更する場合に、そのコンピューターのグループ DB2ADMNS と DB2USERS がローカル・コンピューター・グループであれば、グローバル・レジストリー DB2_ADMINGROUP と DB2_USERSGROUP を更新する必要があります。コンピューターの名前を変更してコンピューターを再始動した後にレジストリー変数を更新するには、以下のコマンドを実行します。

1. コマンド・プロンプトを開きます。
2. db2extsec コマンドを実行して、セキュリティー設定を更新します。

   ```
   db2extsec -a new computer name¥DB2ADMNS -u new computer name¥DB2USERS
   ```

注: Windows Vista にインストールした DB2 データベース製品で拡張セキュリティーが使用可能になっていると、グラフィカルな DB2 管理ツールを実行できるのは、DB2ADMNS グループに属するユーザーだけになります。さらに、DB2ADMNS グループのメンバーは、完全な管理者特権でそれらのツールを起動する必要があります。そのためには、ショートカットを右クリックして、「管理者として実行 (Run as administrator)」を選択します。

DB2ADMNS グループと DB2USERS グループによって取得される権限

DB2ADMNS と DB2USERS グループはメンバーに、以下の機能を提供します。

- DB2ADMNS

  すべての DB2 オブジェクトに対するフル・コントロール (保護されるオブジェクトについては以下のリストを参照)

- DB2USERS

  インストール・ディレクトリーとインスタンス・ディレクトリーに配置されたすべての DB2 オブジェクトに対する読み取りおよび実行アクセス。ただし、データベース・システム・ディレクトリー以下のオブジェクトにはアクセスできません。また IPC リソースに対しては限定されたアクセスになります。

  特定のオブジェクトに対して、必要に応じて追加的な特権が選択可能です (たとえば、書き込み特権、ファイルの追加特権、ファイルの更新特権など)。このグループのメンバーは、データベース・システム・ディレクトリー以下のオブジェクトにはアクセスできません。

  注: 実行アクセスの意味はオブジェクトにより異なります。たとえば、**.dll** や **.exe** ファイルに対する実行アクセスは、そのファイルを実行する権限があるという意味ですが、ディレクトリーに対する実行アクセスは、そのディレクトリーを全検索する権限があるという意味です。

すべての DB2 管理者は、DB2ADMNS グループのメンバー (かつローカル Administrators グループのメンバー) にするのが理想です。ただし、これは必須要件ではありません。 DB2 データベース・システムへのアクセス要求をする他のすべてのメンバーは、DB2USERS グループのメンバーである必要 があります。ユーザーをこれらのグループに追加するには、以下のようにします。

1. 「ユーザー/パスワード・マネージャー・ツール」を起動します。
2. ユーザー名を選択し、リストから追加します
3. 「プロパティー」をクリックします。「プロパティー」ウィンドウで、「グループ・メンバーシップ」タブをクリックします。
4. 「その他」ラジオ・ボタンを選択します。
5. ドロップダウン・リストから適切なグループを選択します。

インストール後の拡張セキュリティー追加 (db2extsec コマンド)

拡張セキュリティーを有効にせずに DB2 データベース・システムをインストールした場合、**db2extsec** コマンド (初期のリリースでは **db2secv82** コマンド) を実行して有効にすることができます。 **db2extsec** コマンドを実行するには、ローカルの Administrators グループのメンバーであり、保護されたオブジェクトの ACL を変更する権限を持つ必要があります。

必要に応じて、**db2extsec** コマンドを複数回実行することができます。ただしその場合、**db2extsec** を*実行*するたびに 直後に **db2extsec –r** コマンドを実行しない限り、拡張セキュリティーを使用不可に設定することはできません。

拡張セキュリティーの削除

注意:
拡張セキュリティーを使用可能にした後に削除する操作は、絶対に必要な場合以外は実行しないでください。

db2extsec -r コマンドを実行して拡張セキュリティーを削除できますが、削除が正常に完了するのは、拡張セキュリティーを有効にした後、データベースの作成、新規インスタンスの作成、表スペースの追加などの他のデータベース操作がされていない場合に限ります。拡張セキュリティー・オプションを削除する最も安全な方法は、DB2 データベース・システムをアンインストールし、データベース・ディレクトリーを含むすべての関連する DB2 ディレクトリーをすべて削除し、それから拡張セキュリティーを有効にしないで、DB2 データベース・システムを再インストールする方法です。

保護されたオブジェクト

DB2ADMNS と DB2USERS グループを使用して保護することができる*静的* オブジェクトには次のものがあります。

- ファイル・システム
  - ファイル
  - ディレクトリー
- サービス
- レジストリー・キー

DB2ADMNS と DB2USERS グループを使用して保護することができる*動的* オブジェクトには次のものがあります。

- 以下を含む IPC リソース
  - パイプ
  - セマフォー
  - イベント
- 共用メモリー

DB2ADMNS と DB2USERS グループの所有特権

DB2ADMNS と DB2USERS グループに割り当てられた特権を次の表に掲載します。

表 52. DB2ADMNS と DB2USERS グループの特権

特権	DB2ADMNS	DB2USERS	理由
トークン・オブジェクトの作成 (SeCreateTokenPrivilege)	Y	N	トークン操作 (一定のトークン操作が要求され、認証と許可に使用されます)
処理レベル・トークンの置換 (SeAssignPrimaryTokenPrivilege)	Y	N	他のユーザーとして処理の作成
割り当て量の引き上げ (SeIncreaseQuotaPrivilege)	Y	N	他のユーザーとして処理の作成

表 52. *DB2ADMNS と DB2USERS グループの特権 (続き)*

特権	DB2ADMNS	DB2USERS	理由
オペレーティング・システムの一部としての活動	Y	N	LogonUser (認証目的の Windows XP より前のバージョンでは、LogonUser API を実行するために必要です)
セキュリティー監査の生成 (SeSecurityPrivilege)	Y	N	監査とセキュリティー・ログの操作
ファイルと他のオブジェクトの所有権 (SeTakeOwnershipPrivilege)	Y	N	オブジェクト ACL の変更
スケジューリング優先順位の引き上げ (SeIncreaseBasePriorityPrivilege)	Y	N	処理作業セットの変更
ファイルとディレクトリーのバックアップ (SeBackupPrivilege)	Y	N	プロファイル/レジストリー操作 (次の特定のユーザー・プロファイルとレジストリー操作ルーチンが必要です。LoadUserProfile、RegSaveKey(Ex)、 RegRestoreKey、RegReplaceKey、 RegLoadKey(Ex))
ファイルとディレクトリーのリストア (SeRestorePrivilege)	Y	N	プロファイル/レジストリー操作 (次の特定のユーザー・プロファイルとレジストリー操作ルーチンが必要です。LoadUserProfile、RegSaveKey(Ex)、 RegRestoreKey、RegReplaceKey、 RegLoadKey(Ex))
デバッグ・プログラム (SeDebugPrivilege)	Y	N	トークン操作 (一定のトークン操作が要求され、認証と許可に使用されます)
監査とセキュリティー・ログの管理 (SeAuditPrivilege)	Y	N	監査ログ・エントリーの生成
サービスとしてログオン (SeServiceLogonRight)	Y	N	サービスとして DB2 を実行します。
ネットワークからコンピューターにアクセス (SeNetworkLogonRight)	Y	Y	ネットワーク・クリデンシャルを許可します。(DB2 データベース・マネージャーに LOGON32_LOGON_NETWORK オプションを認証のため使用することを許可します。これは、パフォーマンスに影響します。)
認証後クライアントの偽装 (SeImpersonatePrivilege)	Y	N	クライアントの偽装 (Windows で DB2 クライアントの偽名を使用するため、ImpersonateLoggedOnUser、 ImpersonateSelf、RevertToSelf などの特定の API の使用を許可する場合に必要)
メモリー内のロックされたページ (SeLockMemoryPrivilege)	Y	N	ラージ・ページのサポート
グローバル・オブジェクトの作成 (SeCreateGlobalPrivilege)	Y	Y	端末サーバー・サポート(Windows で必要)

Vista に関する考慮事項: ユーザー・アクセス制御フィーチャー

Windows Vista のユーザー・アクセス制御 (UAC) フィーチャーは、次のような点で DB2 データベース・システムに影響を与えます。

完全な管理特権によるアプリケーションの開始

Vista の場合、デフォルトでは、標準的なユーザー権限だけでアプリケーションを開始することになります。ユーザーがローカル管理者であっても、それは変わりません。さらに多くの特権を持った状態でアプリケーションを開始するには、完全な管理特権で実行するコマンド・ウィンドウからコマンドを起動する必要があります。DB2 のインストール・プロセスでは、Vista ユーザーに対応した「コマンド・ウィンドウ - 管理者」というショートカットが作成されます。管理コマンドを実行する場合は、このショートカットを起動することをお勧めします。

Windows Vista の場合、完全な管理特権を持っていない状態で、コマンド・プロンプトやグラフィック・ツールから DB2 管理タスクを実行しようとすると、さまざまなエラー・メッセージが生成される可能性があります。いずれも、アクセスが拒否され、タスクが正常に完了しなかった、という趣旨のメッセージです。

実行しようとしたアクションが管理タスクと見なされるかどうかを確認するために、以下のいずれかが当てはまるかどうかをチェックしてください。

- SYSADM、SYSCTRL、SYSMAINT のいずれかの権限が必要です。
- レジストリーの HKLM ブランチにあるレジストリー・キーが変更されます。
- Program Files ディレクトリーの下にあるディレクトリーに書き込まれます。

例えば、以下のようなアクションはすべて管理タスクと見なされます。

- DB2 インスタンスの作成とドロップ
- DB2 インスタンスの開始と停止
- データベースの作成
- データベース・マネージャーの構成パラメーターまたは DB2 Administration Server (DAS) の構成パラメーターの更新
- CLI 構成パラメーターの更新とシステム・データ・ソース名 (DSN) の構成
- DB2 トレース機能の開始
- db2pd ユーティリティーの実行
- DB2 プロファイル・レジストリー変数の変更

問題を解決するには、完全な管理者特権で実行するコマンド・プロンプトやグラフィック・ツールから DB2 管理タスクを実行する必要があります。完全な管理者特権でコマンド・プロンプトやグラフィック・ツールを起動するには、前述のショートカットを右クリックして、「**管理者として実行 (Run as administrator)**」を選択します。

注: 拡張セキュリティーが有効になっている場合は、グラフィカル管理ツール (コマンド・エディターやコントロール・センターなど) を起動するために、DB2ADMNS グループのメンバーになっていることも必要です。

ユーザー・データの場所

ユーザー・データ (インスタンス・ディレクトリーにあるファイルなど) は、ProgramData¥IBM¥DB2¥*copy_name* に格納されます (*copy_name* は DB2 コピーの名前で、デフォルトでは、インストール済みの最初のコピーの名前が DB2COPY1 になります)。Vista 以外の Windowsバージョンでは、Documents and Settings¥All

Users¥Application Data¥IBM¥DB2¥*copy_name* にユーザー・データが格納されます。

DB2 および UNIX セキュリティー

UNIX プラットフォームでのユーザーのセキュリティーに関する考慮事項

DB2 データベースは、ルートが直接データベース管理者として動作することをサポートしていません。データベース管理者としては su - <instance owner> を使用してください。

セキュリティーの理由で、インスタンス名を fenced ID として使用しないでください。ただし、fenced UDF またはストアード・プロシージャーを使用する計画がないならば、別のユーザー ID を作成する代わりに fenced ID をインスタンス名に設定することができます。

推奨は、このグループに関連付けられていると認識されるユーザー ID を作成することです。 fenced UDF およびストアード・プロシージャーのユーザーは、インスタンス作成スクリプト (db2icrt ... -u <FencedID>) のパラメーターとして指定されます。 DB2 クライアントまたは DB2 Software Developer's Kit をインストールする場合、これは必須ではありません。

インスタンス・ディレクトリーの場所

Linux および UNIX のルート・インストールでは、db2icrt コマンドはインスタンス所有者のホーム・ディレクトリーの下に、メイン SQL ライブラリー (sqllib) ディレクトリーを作成します。

Windows オペレーティング・システムでは、インスタンス・ディレクトリーは DB2 データベース・システムがインストールされたディレクトリーの /sqllib サブディレクトリーにあります。

DB2 および Linux セキュリティー

パスワード変更サポート (Linux)

DB2 データベース製品では、Linux オペレーティング・システムでパスワードを変更するためのサポートが用意されています。

このサポートをインプリメントするために、IBMOSchgpwdclient.so と IBMOSchgpwdserver.so というセキュリティー・プラグイン・ライブラリーが使用されています。

Linux でパスワード変更サポートを使用可能にするには、データベース・マネージャー構成パラメーター **CLNT_PW_PLUGIN** を IBMOSchgpwdclient に、**SRVCON_PW_PLUGIN** を IBMOSchgpwdserver にそれぞれ設定します。

さらに、/etc/pam.d ディレクトリーに db2 という PAM 構成ファイルを作成することも必要です。

パスワード変更プラグインのデプロイ (Linux)

Linux にインストールした DB2 データベース製品でパスワード変更サポートを有効にするには、セキュリティー・プラグイン IBMOSchgpwdclient と IBMOSchgpwdserver を使用するように DB2 インスタンスを構成する必要があります。

プラグイン・ライブラリーは、以下のディレクトリーにあります。

- *INSTHOME*/sqllib/*securityXX*/plugin/client/IBMOSchgpwdclient.so
- *INSTHOME*/sqllib/*securityXX*/plugin/server/IBMOSchgpwdserver.so

INSTHOME は、インスタンス所有者のホーム・ディレクトリー、*securityXX* は、インスタンスのビット幅によって、security32 または security64 のいずれかになります。

DB2 インスタンスにセキュリティー・プラグインをデプロイするには、以下の手順を実行します。

1. root 権限を持つユーザーとしてログインします。
2. PAM 構成ファイル /etc/pam.d/db2 を作成します。

   そのファイルに、システム管理者によって定義されている適切な規則のセットが含まれていることを確認します。例:

   ```
   auth     required pam_unix2.so    nullok
   account  required pam_unix2.so
   password required pam_pwcheck.so  nullok tries=1
   password required pam_unix2.so    nullok use_authtok use_first_pass
   session  required pam_unix2.so
   ```

3. DB2 インスタンスでセキュリティー・プラグインを有効にします。
   a. データベース・マネージャー構成パラメーター **SRVCON_PW_PLUGIN** を値 IBMOSchgpwdserver で更新します。

      ```
      db2 update dbm cfg using srvcon_pw_plugin IBMOSchgpwdserver
      ```

   b. データベース・マネージャー構成パラメーター **CLNT_PW_PLUGIN** を値 IBMOSchgpwdclient で更新します。

      ```
      db2 update dbm cfg using CLNT_PW_PLUGIN IBMOSchgpwdclient
      ```

   c. データベース・マネージャー構成パラメーター **SRVCON_AUTH** が、CLIENT、SERVER、SERVER_ENCRYPT、DATA_ENCRYPT、DATA_ENCRYPT_CMP のいずれかの値に設定されているか、データベース・マネージャー構成パラメーター **SRVCON_AUTH** が NOT_SPECIFIED の値に設定されていて、**AUTHENTICATION** が、CLIENT、SERVER、SERVER_ENCRYPT、DATA_ENCRYPT、DATA_ENCRYPT_CMP のいずれかの値に設定されていることを確認します。

付録 A. DB2 技術情報の概説

DB2 技術情報は、以下のツールと方法を介して利用できます。

- DB2 インフォメーション・センター
  - トピック (タスク、概念、およびリファレンス・トピック)
  - DB2 ツールのヘルプ
  - サンプル・プログラム
  - チュートリアル
- DB2 資料
  - PDF ファイル (ダウンロード可能)
  - PDF ファイル (DB2 PDF DVD に含まれる)
  - 印刷資料
- コマンド行ヘルプ
  - コマンド・ヘルプ
  - メッセージ・ヘルプ

注: DB2 インフォメーション・センターのトピックは、PDF やハードコピー資料よりも頻繁に更新されます。最新の情報を入手するには、資料の更新が発行されたときにそれをインストールするか、ibm.com® にある DB2 インフォメーション・センターを参照してください。

技術資料、ホワイト・ペーパー、IBM Redbooks® 資料などのその他の DB2 技術情報には、オンライン (ibm.com) でアクセスできます。 DB2 Information Management ソフトウェア・ライブラリー・サイト (http://www.ibm.com/software/data/sw-library/) にアクセスしてください。

資料についてのフィードバック

DB2 の資料についてのお客様からの貴重なご意見をお待ちしています。DB2 の資料を改善するための提案については、db2docs@ca.ibm.com まで E メールを送信してください。DB2 の資料チームは、お客様からのフィードバックすべてに目を通しますが、直接お客様に返答することはありません。お客様が関心をお持ちの内容について、可能な限り具体的な例を提供してください。特定のトピックまたはヘルプ・ファイルについてのフィードバックを提供する場合は、そのトピック・タイトルおよび URL を含めてください。

DB2 お客様サポートに連絡する場合には、この E メール・アドレスを使用しないでください。資料を参照しても、DB2 の技術的な問題が解決しない場合は、お近くの IBM サービス・センターにお問い合わせください。

DB2 テクニカル・ライブラリー (ハードコピーまたは PDF 形式)

以下の表は、DB2 ライブラリーについて説明しています。DB2 ライブラリーに関する詳細な説明については、www.ibm.com/shop/publications/order にある IBM Publications Center にアクセスしてください。英語の DB2 バージョン 9.5 のマニュアル (PDF 形式) とその翻訳版は、 www.ibm.com/support/docview.wss?rs=71&uid=swg2700947 からダウンロードできます。

この表には印刷資料が入手可能かどうかが示されていますが、国または地域によっては入手できない場合があります。

表 53. DB2 の技術情報

資料名	資料番号	印刷資料が入手可能かどうか
管理 API リファレンス	SC88-4431-00	入手可能
管理ルーチンおよびビュー	SC88-4435-00	入手不可
コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻	SC88-4433-00	入手可能
コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻	SC88-4434-00	入手可能
コマンド・リファレンス	SC88-4432-00	入手可能
データ移動ユーティリティー ガイドおよびリファレンス	SC88-4421-00	入手可能
データ・リカバリーと高可用性 ガイドおよびリファレンス	SC88-4423-00	入手可能
データ・サーバー、データベース、およびデータベース・オブジェクトのガイド	SC88-4259-00	入手可能
データベース・セキュリティー・ガイド	SC88-4418-00	入手可能
ADO.NET および OLE DB アプリケーションの開発	SC88-4425-00	入手可能
組み込み SQL アプリケーションの開発	SC88-4426-00	入手可能
Java アプリケーションの開発	SC88-4427-00	入手可能
Perl および PHP アプリケーションの開発	SC88-4428-00	入手不可
SQL および 外部ルーチンの開発	SC88-4429-00	入手可能
データベース・アプリケーション開発の基礎	GC88-4430-00	入手可能
DB2 インストールおよび管理概説 (Linux および Windows 版)	GC88-4439-00	入手可能
国際化対応ガイド	SC88-4420-00	入手可能

表 53. DB2 の技術情報 (続き)

資料名	資料番号	印刷資料が入手可能かどうか
メッセージ・リファレンス 第 1 巻	GI88-4109-00	入手不可
メッセージ・リファレンス 第 2 巻	GI88-4110-00	入手不可
マイグレーション・ガイド	GC88-4438-00	入手可能
Net Search Extender 管理およびユーザーズ・ガイド **注:** この資料の内容は、DB2 インフォメーション・センターには含まれていません。	SC88-4630-00	入手可能
パーティションおよびクラスタリングのガイド	SC88-4419-00	入手可能
Query Patroller 管理およびユーザーズ・ガイド	SC88-4611-00	入手可能
IBM データ・サーバー・クライアント機能 概説およびインストール	GC88-4441-00	入手不可
DB2 サーバー機能 概説およびインストール	GC88-4440-00	入手可能
Spatial Extender and Geodetic Data Management Feature ユーザーズ・ガイドおよびリファレンス	SC88-4629-00	入手可能
SQL リファレンス 第 1 巻	SC88-4436-00	入手可能
SQL リファレンス 第 2 巻	SC88-4437-00	入手可能
システム・モニター ガイドおよびリファレンス	SC88-4422-00	入手可能
テキスト検索ガイド	SC88-4424-00	入手可能
問題判別ガイド	GI88-4108-00	入手不可
データベース・パフォーマンスのチューニング	SC88-4417-00	入手可能
Visual Explain チュートリアル	SC88-4449-00	入手不可
新機能	SC88-4445-00	入手可能
ワークロード・マネージャー ガイドおよびリファレンス	SC88-4446-00	入手可能
pureXML ガイド	SC88-4447-00	入手可能
XQuery リファレンス	SC88-4448-00	入手不可

表 54. DB2 Connect 固有の技術情報

資料名	資料番号	印刷資料が入手可能かどうか
DB2 Connect Personal Edition 概説およびインストール	GC88-4443-00	入手可能

表 54. DB2 Connect 固有の技術情報 (続き)

資料名	資料番号	印刷資料が入手可能かどうか
DB2 Connect サーバー機能 概説およびインストール	GC88-4444-00	入手可能
DB2 Connect ユーザーズ・ガイド	SC88-4442-00	入手可能

表 55. Information Integration の技術情報

資料名	資料番号	印刷資料が入手可能かどうか
Information Integration: フェデレーテッド・システム 管理ガイド	SC88-4166-01	入手可能
Information Integration: レプリケーションおよびイベント・パブリッシングのための ASNCLP プログラム・リファレンス	SC88-4167-02	入手可能
Information Integration: フェデレーテッド・データ・ソース構成ガイド	SC88-4185-01	入手不可
Information Integration: SQL レプリケーション ガイドおよびリファレンス	SC88-4168-01	入手可能
Information Integration: レプリケーションとイベント・パブリッシング 概説	GC88-4187-01	入手可能

DB2 の印刷資料の注文方法

DB2 の印刷資料が必要な場合、オンラインで購入することができますが、すべての国および地域で購入できるわけではありません。 DB2 の印刷資料については、IBM 営業担当員にお問い合わせください。 *DB2 PDF ドキュメンテーション* DVD の一部のソフトコピー・ブックは、印刷資料では入手できないことに留意してください。例えば、「*DB2 メッセージ・リファレンス*」はどちらの巻も印刷資料としては入手できません。

DB2 PDF ドキュメンテーション DVD で利用できる DB2 の印刷資料の大半は、IBM に有償で注文することができます。国または地域によっては、資料を IBM Publications Center からオンラインで注文することもできます。お客様の国または地域でオンライン注文が利用できない場合、DB2 の印刷資料については、IBM 営業担当員にお問い合わせください。 DB2 PDF ドキュメンテーション DVD に収録されている資料の中には、印刷資料として提供されていないものもあります。

注: 最新で完全な DB2 資料は、DB2 インフォメーション・センター (http://publib.boulder.ibm.com/infocenter/db2luw/v9r5) で参照することができます。

DB2 の印刷資料は以下の方法で注文することができます。

- 日本 IBM 発行のマニュアルはインターネット経由でご購入いただけます。詳しくは http://www.ibm.com/shop/publications/order の「ご注文について」をご覧ください。資料の注文情報にアクセスするには、お客様の国、地域、または言語を選択してください。その後、各ロケーションにおける注文についての指示に従ってください。
- DB2 の印刷資料を IBM 営業担当員に注文するには、以下のようにします。
  1. 以下の Web サイトのいずれかから、営業担当員の連絡先情報を見つけてください。
     - IBM Directory of world wide contacts (www.ibm.com/planetwide)
     - IBM Publications Web サイト (http://www.ibm.com/shop/publications/order) 国、地域、または言語を選択し、お客様の所在地に該当する Publications ホーム・ページにアクセスしてください。このページから、「このサイトについて」のリンクにアクセスしてください。
  2. 電話をご利用の場合は、DB2 資料の注文であることをご指定ください。
  3. 担当者に、注文する資料のタイトルと資料番号をお伝えください。タイトルと資料番号は、282 ページの『DB2 テクニカル・ライブラリー (ハードコピーまたは PDF 形式)』でご確認いただけます。

コマンド行プロセッサーから SQL 状態ヘルプを表示する

DB2 は、SQL ステートメントの結果の原因になったと考えられる条件の SQLSTATE 値を戻します。 SQLSTATE ヘルプは、SQL 状態および SQL 状態クラス・コードの意味を説明します。

SQL 状態ヘルプを呼び出すには、コマンド行プロセッサーを開いて以下のように入力します。

```
? sqlstate or ? class code
```

ここで、*sqlstate* は有効な 5 桁の SQL 状態を、*class code* は SQL 状態の最初の 2 桁を表します。

例えば、`? 08003` を指定すると SQL 状態 08003 のヘルプが表示され、 `? 08` を指定するとクラス・コード 08 のヘルプが表示されます。

異なるバージョンの DB2 インフォメーション・センターへのアクセス

DB2 バージョン 9.5 のトピックを扱っている DB2 インフォメーション・センターの URL は、http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/です。

DB2 バージョン 9 のトピックを扱っている DB2 インフォメーション・センターの URL は http://publib.boulder.ibm.com/infocenter/db2luw/v9/です。

DB2 バージョン 8 のトピックについては、バージョン 8 のインフォメーション・センターの URL http://publib.boulder.ibm.com/infocenter/db2luw/v8/にアクセスしてください。

DB2 インフォメーション・センターにおける特定の言語でのトピックの表示

DB2 インフォメーション・センターでは、ブラウザーの設定で指定した言語でのトピックの表示が試みられます。トピックがその指定言語に翻訳されていない場合は、 DB2 インフォメーション・センターでは英語でトピックが表示されます。

- Internet Explorer Web ブラウザーで、指定どおりの言語でトピックを表示するには、以下のようにします。
  1. Internet Explorer の「**ツール**」 -> 「**インターネット オプション**」 -> 「**言語...**」ボタンをクリックします。「言語の優先順位」ウィンドウがオープンします。
  2. 該当する言語が、言語リストの先頭の項目に指定されていることを確認します。
     - リストに新しい言語を追加するには、「**追加...**」ボタンをクリックします。

       注: 言語を追加しても、特定の言語でトピックを表示するのに必要なフォントがコンピューターに備えられているとはかぎりません。
     - リストの先頭に新しい言語を移動するには、その言語を選択してから、その言語が言語リストに先頭に行くまで「**上に移動**」ボタンをクリックします。
  3. ブラウザー・キャッシュを消去してから、ページをリフレッシュし、使用する言語で DB2 インフォメーション・センターを表示します。
- Firefox または Mozilla Web ブラウザーの場合に、使いたい言語でトピックを表示するには、以下のようにします。
  1. 「**ツール**」 -> 「**オプション**」 -> 「**詳細**」 ダイアログの「**言語**」セクションにあるボタンを選択します。「設定」ウィンドウに「言語」パネルが表示されます。
  2. 該当する言語が、言語リストの先頭の項目に指定されていることを確認します。
     - リストに新しい言語を追加するには、「**追加...**」ボタンをクリックしてから、「言語を追加」ウィンドウで言語を選択します。
     - リストの先頭に新しい言語を移動するには、その言語を選択してから、その言語が言語リストに先頭に行くまで「**上に移動**」ボタンをクリックします。
  3. ブラウザー・キャッシュを消去してから、ページをリフレッシュし、使用する言語で DB2 インフォメーション・センターを表示します。

ブラウザーとオペレーティング・システムの組み合わせによっては、オペレーティング・システムの地域の設定も希望のロケールと言語に変更しなければならない場合があります。

コンピューターまたはイントラネット・サーバーにインストールされた DB2 インフォメーション・センターの更新

DB2 インフォメーション・センターをローカルにインストールしている場合は、IBM から提供される更新をダウンロードおよびインストールすることができます。

ローカルにインストールされた DB2 インフォメーション・センターを更新するには、以下のことを行う必要があります。

1. コンピューター上の DB2 インフォメーション・センターを停止し、インフォメーション・センターをスタンドアロン・モードで再始動します。インフォメーション・センターをスタンドアロン・モードで実行すると、ネットワーク上の他のユーザーがそのインフォメーション・センターにアクセスできなくなります。これで、更新をダウンロードして適用できるようになります。
2. 「更新」機能を使用することにより、どんな更新が利用できるかを確認します。インストールする更新がある場合は、「更新」機能を使用してそれをダウンロードおよびインストールできます。

   注: ご使用の環境において、インターネットに接続されていないマシンに DB2 インフォメーション・センターの更新をインストールする必要がある場合は、インターネットに接続されていて DB2 インフォメーション・センターがインストールされているマシンを使用して、更新サイトをローカル・ファイル・システムにミラーリングする必要があります。ネットワーク上の多数のユーザーが資料の更新をインストールする場合にも、更新サイトをローカルにミラーリングして、更新サイト用のプロキシーを作成することにより、個々のユーザーが更新を実行するのに要する時間を短縮できます。
   更新パッケージが入手可能な場合、「更新」機能を使用してパッケージをダウンロードします。ただし、「更新」機能は、スタンドアロン・モードでのみ使用できます。
3. スタンドアロンのインフォメーション・センターを停止し、コンピューター上の DB2 インフォメーション・センターを再開します。

注: Windows Vista の場合、下記のコマンドは管理者として実行する必要があります。完全な管理者特権でコマンド・プロンプトまたはグラフィカル・ツールを起動するには、ショートカットを右クリックしてから、 **「管理者として実行」**を選択します。

コンピューターまたはイントラネット・サーバーにインストール済みの DB2 インフォメーション・センターを更新するには、以下のようにします。

1. DB2 インフォメーション・センターを停止します。
   - Windows では、**「スタート」→「コントロール パネル」→「管理ツール」→「サービス」**をクリックします。次に、**「DB2 インフォメーション・センター」**サービスを右クリックして**「停止」**を選択します。
   - Linux では、以下のコマンドを入力します。

     ```
     /etc/init.d/db2icdv95 stop
     ```
2. インフォメーション・センターをスタンドアロン・モードで開始します。
   - Windows の場合:
     a. コマンド・ウィンドウを開きます。

b. インフォメーション・センターがインストールされているパスにナビゲートします。デフォルトでは、DB2 インフォメーション・センターは <Program Files>¥IBM¥DB2 Information Center¥Version 9.5 ディレクトリーにインストールされています (<Program Files> は「Program Files」ディレクトリーのロケーション)。

c. インストール・ディレクトリーの doc¥bin ディレクトリーにナビゲートします。

d. 次のように help_start.bat ファイルを実行します。

```
help_start.bat
```

- Linux の場合:

a. インフォメーション・センターがインストールされているパスにナビゲートします。デフォルトでは、DB2 インフォメーション・センターは /opt/ibm/db2ic/V9.5 ディレクトリーにインストールされています。

b. インストール・ディレクトリーの doc/bin ディレクトリーにナビゲートします。

c. 次のように help_start スクリプトを実行します。

```
help_start
```

システムのデフォルト Web ブラウザーが起動し、スタンドアロンのインフォメーション・センターが表示されます。

3. 「更新」ボタン () をクリックします。インフォメーション・センターの右側のパネルで、 「更新の検索 (Find Updates)」をクリックします。既存の文書に対する更新のリストが表示されます。
4. ダウンロード・プロセスを開始するには、ダウンロードする更新をチェックして選択し、「更新のインストール (Install Updates)」をクリックします。
5. ダウンロードおよびインストール・プロセスが完了したら、「完了」をクリックします。
6. スタンドアロンのインフォメーション・センターを停止します。
   - Windows の場合は、インストール・ディレクトリーの doc¥bin ディレクトリーにナビゲートしてから、次のように help_end.bat ファイルを実行します。

     ```
     help_end.bat
     ```

     注: help_end バッチ・ファイルには、help_start バッチ・ファイルを使用して開始したプロセスを安全に終了するのに必要なコマンドが含まれています。Ctrl-C または他の方法を使用して、help_start.bat を終了しないでください。
   - Linux の場合は、インストール・ディレクトリーの doc/bin ディレクトリーにナビゲートしてから、次のように help_end スクリプトを実行します。

     ```
     help_end
     ```

     注: help_end スクリプトには、help_start スクリプトを使用して開始したプロセスを安全に終了するのに必要なコマンドが含まれています。他の方法を使用して、help_start スクリプトを終了しないでください。
7. DB2 インフォメーション・センターを再開します。

- Windows では、「**スタート**」→「**コントロール パネル**」→「**管理ツール**」→「**サービス**」をクリックします。次に、「**DB2 インフォメーション・センター**」サービスを右クリックして「**開始**」を選択します。
- Linux では、以下のコマンドを入力します。

  ```
  /etc/init.d/db2icdv95 start
  ```

更新された DB2 インフォメーション・センターに、更新された新しいトピックが表示されます。

DB2 チュートリアル

DB2 チュートリアルは、DB2 製品のさまざまな機能について学習するのを支援します。この演習をとおして段階的に学習することができます。

はじめに

インフォメーション・センター (http://publib.boulder.ibm.com/infocenter/db2help/) から、このチュートリアルの XHTML 版を表示できます。

演習の中で、サンプル・データまたはサンプル・コードを使用する場合があります。個々のタスクの前提条件については、チュートリアルを参照してください。

DB2 チュートリアル

チュートリアルを表示するには、タイトルをクリックします。

「*pureXML ガイド* 」の『**pureXML™**』
: XML データを保管し、ネイティブ XML データ・ストアに対して基本的な操作を実行できるように、DB2 データベースをセットアップします。

「*Visual Explain チュートリアル*」の『**Visual Explain**』
: Visual Explain を使用して、パフォーマンスを向上させるために SQL ステートメントを分析し、最適化し、調整します。

DB2 トラブルシューティング情報

DB2 製品を使用する際に役立つ、トラブルシューティングおよび問題判別に関する広範囲な情報を利用できます。

DB2 ドキュメンテーション
: トラブルシューティング情報は、DB2 問題判別ガイド、または DB2 インフォメーション・センターの「サポートおよびトラブルシューティング」セクションにあります。ここには、DB2 診断ツールおよびユーティリティーを使用して、問題を切り分けて識別する方法、最も頻繁に起こる幾つかの問題に対するソリューションについての情報、および DB2 製品を使用する際に発生する可能性のある問題の解決方法についての他のアドバイスがあります。

DB2 Technical Support の Web サイト
: 現在問題が発生していて、考えられる原因とソリューションを検索したい場合は、DB2 Technical Support の Web サイトを参照してください。
Technical Support サイトには、最新の DB2 資料、TechNotes、プログラム

診断依頼書 (APAR またはバグ修正)、フィックスパック、およびその他のリソースへのリンクが用意されています。この知識ベースを活用して、問題に対する有効なソリューションを探し出すことができます。

DB2 Technical Support の Web サイト (http://www.ibm.com/software/data/db2/udb/support.html) にアクセスしてください。

ご利用条件

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で提供され、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。

付録 B. 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。

〒106-8711
東京都港区六本木 3-2-12
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で提供し、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。

本書は、IBM 以外の Web サイトおよびリソースへのリンクまたは参照を含む場合があります。 IBM は、本書より参照もしくはアクセスできる、または本書からリンクされた IBM 以外の Web サイトもしくは第三者のリソースに対して一切の責任を負いません。 IBM 以外の Web サイトにリンクが張られていることにより IBM が当該 Web サイトを推奨するものではなく、またその内容、使用もしくはサイトの所有者について IBM が責任を負うことを意味するものではありません。 また、IBM は、お客様が IBM Web サイトから第三者の存在を知ることになった場合にも (もしくは、IBM Web サイトから第三者へのリンクを使用した場合にも) 、お客様と第三者との間のいかなる取引に対しても一切責任を負いません。 従って、お客様は、IBM が上記の外部サイトまたはリソースの利用について責任を負うものではなく、また、外部サイトまたはリソースからアクセス可能なコンテンツ、サービス、

製品、またはその他の資料一切に対して IBM が責任を負うものではないことを承諾し、同意するものとします。 第三者により提供されるソフトウェアには、そのソフトウェアと共に提供される固有の使用条件が適用されます。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Canada Limited
Office of the Lab Director
8200 Warden Avenue
Markham, Ontario
L6G 1C7
CANADA

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性がありますが、その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。 IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確証できません。 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にお願いします。

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、単に目標を示しているものです。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。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。

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した創作物には、次のように、著作権表示を入れていただく必要があります。

© (*お客様の会社名*) (*西暦年*). このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。 © Copyright IBM Corp. _*年を入れる*_. All rights reserved.

商標

DB2 バージョン 9.5 ドキュメンテーション・ライブラリーの資料に記載されている会社名、製品名、またはサービス名は、 IBM Corporation の商標である可能性があります。 IBM Corporation の商標については、http://www.ibm.com/legal/copytrade.shtml を参照してください。

以下は、それぞれ各社の商標または登録商標です。

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。

Intel®、 Intel ロゴ、Intel Inside® ロゴ、Intel Centrino®、Intel Centrino ロゴ、Celeron®、Intel Xeon®、 Intel SpeedStep®、 Itanium® および Pentium® は、Intel Corporation の米国およびその他の国における商標です。

Java およびすべての Java 関連の商標およびロゴは、Sun Microsystems, Inc. の米国およびその他の国における商標です。

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。

Linux は、Linus Torvalds の米国およびその他の国における商標です。

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。

他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。

索引

日本語, 数字, 英字, 特殊文字の順に配列されています。なお, 濁音と半濁音は清音と同等に扱われています。

[ア行]

[カ行]

[サ行]

[タ行]

[ナ行]

[ハ行]

[マ行]

[ヤ行]

[ラ行]

A

B

C

D

E

G

I

K

L

N

P

Q

R

S

U

V

W

Printed in Japan

SC88-4418-00

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

Spine information:

DB2 Version 9.5 for Linux, UNIX, and Windows

データベース・セキュリティー・ガイド