承認ガイド

Version 2023.1
2024-01-02

InterSystems Corporation　1 Memorial Drive　Cambridge MA 02142　www.intersystems.com

承認ガイド
InterSystems IRIS Data Platform　Version 2023.1　2024-01-02

インターシステムズ社の製品に関するサポートやご質問は、以下にお問い合わせください：

InterSystems Worldwide Response Center (WRC)
Tel:　+1-617-621-0700
Tel:　+44 (0) 844 854 2917
Email:　support@InterSystems.com

目次

テーブル一覧

1

インターシステムズの承認について

ユーザが認証された後、セキュリティに関連する次の手順は、そのユーザに使用、閲覧、または変更が認められている資源が何であるかを判断することです。**アセット**には、以下のものが含まれます。

- データベース ー データまたはコードが含まれる物理ファイル。
- サービス ー InterSystems IRIS に接続するためのツール (例 : クライアント・サーバ・サービス、Telnet)。
- アプリケーション ー InterSystems IRIS プログラム (例 : Web アプリケーション)。
- 管理アクション ー タスクのセット (例: InterSystems IRIS の開始および停止、バックアップの作成)。

組織のすべてのユーザがシステム上のあらゆる資源を表示して変更できる状況は望ましくありません。資源へのアクセスの決定と制御を承認と呼びます。

ユーザと資源との関係を承認で管理します。InterSystems IRIS® のデータ・プラットフォームでは、この関係をリソースとして表現します。InterSystems IRIS では、その承認モデルとしてロールベースのアクセス制御 (RBAC) を採用しています。このモデルでは、システム管理者がタスクベースの 1 つ以上のロールにユーザを割り当てます。各ロールには、リソースの特定の組み合わせを使用して、アクティビティの特定の組み合わせを実行することが認められています。アプリケーションで、ユーザが持つロールを一時的に拡張できます。

このページでは、InterSystems IRIS に実装されている RBAC 承認モデルの概要について説明します。実際の操作を通じたインターシステムズの RBAC の説明は、"Configuring Role-Based Access" を参照してください。

1.1 リソース、許可、および特権

セキュリティの最大の目的は、情報や機能である資源を何らかの形式で保護することです。InterSystems IRIS データ・プラットフォームでは、データベース、サービス、アプリケーション、ツールなどのほか、管理アクションも資源と捉えることができます。

InterSystems IRIS では各資源はリソースで表現され、1 つのリソースが複数の資源を表すこともあります。

システム管理者は、リソースに許可を割り当てることで、資源へのアクセスを制御します。許可を付与または取り消すことで、リソースが表現している資源に対して実行できるアクティビティへのアクセスを有効または無効にします。データベースの場合、許可は Read と Write です。その他ほとんどのリソース・タイプで関連する許可は Use です。

リソースとそれに関連する許可の組み合わせを特権といいます。これは、多くの場合、以下の省略表現を使用して記述されます。`Resource-Name:Permission`。例えば、EmployeeInfo データベースに対する読み取り許可と書き込み許可を付与する特権は、以下のように表現できます。`%DB_EmployeeInfo:Read,Write` または `%DB_EmployeeInfo:RW`。

詳細は、"リソースの使用による資源の保護" と "特権および許可" を参照してください。

1.2 ユーザとロール

インターシステムズのロールベースのアクセス制御モデルでは、以下のようにユーザがリソースを操作できます。

1. 前のセクションの説明のとおり、許可にリソースを関連付けて特権を確立します。
2. 特権の集合をロールに関連付けます。
3. ロールに、ユーザなどのメンバを割り当てます。

ユーザは InterSystems IRIS に接続して一連のタスクを実行します。ロールは、ユーザが持つ一連の特権を記述するものであり、したがってユーザが実行できるタスクを表すと言えます。

ロールは、ユーザと特権を仲介する機能を提供します。ユーザの人数に応じて数多くの特権のセットを作成する代わりに、ロールを使用すれば、タスク固有の特権のセットを作成できます。ロールで保持された特権を付与、変更、削除できます。この内容は、そのロールに関連するすべてのユーザに自動的に伝播されます。特権のセットを個人ユーザや全ユーザごとに管理するのではなく、ロールを極めて少ない数に抑えて管理することになります。

例えば、病院向けのアプリケーションには、巡回を担当する医師 (**RoundsDoctor**) のロールと救急処置室に勤務する医師 (**ERDoctor**) のロールがあり、それぞれのロールに適切な特権が関連付けられています。

個々のユーザを複数のロールのメンバとすることができます。上記の例を使用すると、病院の医長は、すべての診療科の医師が使用する機能を必要とすることがあります。このユーザには、**RoundsDoctor** ロールと **ERDoctor** ロールの両方を割り当てることが考えられます。また、これら両方のロールのメンバであると同時に、それに応じて特権を継承した **MedicalDirector** ロールをシステム管理者が作成することもできます。

ロールベースのアクセス制御によるネイティブな InterSystems の実装は、InterSystems IRIS がサポートするすべてのタイプの認証メカニズム (LDAP、Kerberos、OS ベースなど) で使用できます。また、LDAP や代行認証を使用してロールを割り当てることもできます。詳細は、"ロール" および "ユーザ・アカウント" を参照してください。

1.3 アプリケーション

インターシステムズのセキュリティは、柔軟なアプリケーション・セキュリティ・モデルを提供します。アプリケーションを使用する機能も 1 つのリソースであるため、特定のアプリケーションの使用を特定のユーザ・グループに限定することも、すべてのユーザが使用できるようにすることもできます。アプリケーションを使用できるユーザについては、セキュリティ・モデルではロール・エスカレーション・モデルがサポートされます。つまり、ユーザはアプリケーションを使用している間は、通常であればアクセスできない特定のリソースにアクセスできます。

複数タイプのアプリケーションの詳細は、"アプリケーション" を参照してください。

2
リソースの使用による資源の保護

承認によって、InterSystems IRIS® データ・プラットフォームのコンポーネントが不適切な使用から保護されます。インターシステムズでは、これらのコンポーネントを取り上げる場合、以下の用語を使用します。

- 資源とは、保護の対象になるものです。例えば、InterSystems IRIS データベースは資源であり、SQL を使用して InterSystems IRIS に接続する機能も資源です。また、バックアップを実行する機能も資源です。
- リソースは資源を保護します。資源とリソースの中には一対一で対応するものがあり、この場合は 1 つのリソースによって単一の資源（データベースなど）が保護されます。これ以外の場合は、セキュリティ管理を簡素化するために、単一のリソースで複数の資源が保護されます。例えば、さまざまなシステム管理機能が単一のリソースで保護されます。
- 特権によって、あるリソースで保護されている 1 つ以上の資源に対して何らかの処理を実行する許可が付与されます。例えば、注文データベースを読み取ることができるようになります。特権は、`%DB_Sales:Read` のように、リソース名の後にコロンで区切って許可を記す書式で記述します。

  特権の詳細は、"特権および許可" を参照してください。

ここでは、リソースとリソースが保護する資源に関する問題を扱います。InterSystems IRIS には、資源を保護する一連のリソースが含まれており、ユーザが保持している権限に基づいて、ユーザに資源へのアクセスが提供されます。ユーザが独自のリソースを定義することもできます。

2.1 リソースのタイプ

リソースには、以下のようないくつかのタイプがあります。

- システム・リソース ー InterSystems IRIS インスタンスに対してさまざまなアクションを実行する機能を制御します。これらのリソースの詳細は、"システム・リソース" を参照してください。

  このようなリソースとして
  %Admin_Manage:Use、**%Admin_Operate:Use**、**%Admin_Secure:Use**、**%Admin_Task:Use**、**%Admin_Wsgateway:Use**、**%DB_ENSLIB:RW**、**%DB_IRISAUDIT:RW**、**%DB_IRISLIB:RW**、**%DB_IRISLOCALDATA:RW**、**%DB_IRISSYS:RW**、**%DB_IRISTEMP:RW**、**%Development:Use**、**%Service_Bindings:Use**、**%Service_CallIn:Use**、**%Service_ComputeNode:Use**、**%Service_Console:Use**、**%Service_Mirror:Use**、**%Service_Monitor:Use**、**%Service_Shadow:Use**、**%Service_Terminal:Use**、**%Service_WebGateway:Use**
  があります。
- データベース・リソース ー InterSystems IRIS データベースに対する読み取りと書き込みのアクセスを制御します。これらのリソースの詳細は、"データベース・リソース" を参照してください。

  新しくインストールされた InterSystems IRIS インスタンスのデータベース・リソースは、**%DB_IRISLOCALDATA**、**%DB_IRISAUDIT**、**%DB_IRISLIB**、**%DB_IRISLOCALDATA**、**%DB_IRISSYS**、**%DB_IRISTEMP**、**%DB_ENSLIB** です。

- ゲートウェイ・リソース ー 外部言語サーバへのアクセスを制御します。これらのリソースの詳細は、"ゲートウェイ・リソース" を参照してください。

  新たにインストールした InterSystems IRIS インスタンスのゲートウェイ・リソースは、**%Gateway_Object**、**%Gateway_SQL**、**%Gateway_ML** です。

- サービス・リソース ー InterSystems のさまざまな接続テクノロジを使用して InterSystems IRIS に接続する機能を制御します。これらのリソースとそれらが制御する機能の詳細は、"サービス" を参照してください。

  すべてのサービスが特権に関連付けられているわけではありません。InterSystems IRIS によってユーザ・ベースのアクセスが提供されるサービスのみにこのような関連付けがあります。データ・チェックなどのこれ以外のサービスはユーザ・ベースではないので、関連付けられたセキュリティ・リソースが存在しません。サービスの管理の詳細は、"サービス" を参照してください。

  サービス・リソースとは、**%Service_CallIn**、**%Service_ECP**、**%Service_Console**、**%Service_DocDB**、**%Service_Login**、**%Service_Native**、**%Service_Object**、**%Service_SQL**、**%Service_Telnet**、**%Service_Terminal**、**%Service_WebGateway** です。

- アプリケーション・リソース ー ユーザ定義アプリケーションの全体を制御するか、ユーザ・コードの任意の場所において承認を行います。これらのリソースの一般的な情報は、"アプリケーション・リソース" を参照してください。これらのリソースの作成に関する詳細は、"リソースの作成または編集" を参照してください。

2.2 システム・リソース

InterSystems IRIS には、インストール済みの InterSystems IRIS インスタンスに関するアクションを制御する組み込みリソースのセットが付属しています。システム・リソースには以下のものがあります。

- 管理リソース
- %Development リソース
- %DocDB_Admin リソース
- %IAM リソース
- %System_Callout リソース
- %System_Attach リソース
- %Secure_Break リソース
- %Service_Native リソース

システム・リソースには、リソース・ベース・サービスに関連付けられたリソースも含まれます。サービスの詳細は、"サービス" を参照してください。

2.2.1 管理リソース

管理リソースは以下のとおりです。

- %Admin_ExternalLanguageServerEdit
- %Admin_Journal
- %Admin_Manage
- %Admin_Operate
- %Admin_RoleEdit

- %Admin_Secure
- %Admin_Tasks
- %Admin_UserEdit

注釈　**%Admin_*** リソースに対する特権があると、ユーザは、データベースの特権 (**%DB_<database-name>:R/W**) をまったく持っていなくても管理機能を実行できます。例えば、**%Admin_Operate:Use** 特権を持つシステム運用管理者などのユーザは、データベースに対する特権がなくても、そのデータベースを含むバックアップを実行できます。これは、InterSystems IRIS のデータベース・バックアップ・システムのようなアプリケーション以外の方法で、オペレータがデータベースの内容にアクセスする必要はないからです。

2.2.1.1 %Admin_ExternalLanguageServerEdit

このリソースは、外部言語サーバ (ゲートウェイともいいます) を作成、変更、削除する機能を制御します。ゲートウェイに関連付けたゲートウェイ・リソースの変更も対象となります。

このリソースには Use 許可が必要です。

既定では、**%Manager** ロールは **%Admin_ExternalLanguageServerEdit:USE** 特権を保持します。

2.2.1.2 %Admin_Journal

このリソースにより、ユーザは、ターミナルのプログラマ・モードで、ジャーナリングなしプロセス・フラグを DISABLE^%SYS.NOJRN と ENABLE^%SYS.NOJRN のエントリ・ポイントによって、それぞれ設定およびクリアできます。このリソースにより、Use 許可を **%Admin_Manage** リソースで付与 (この場合、必要以上の特権が付与される場合があります) しなくても、このアクションを実行可能なユーザを確立できます。

このリソースには Use 許可が必要です (Read 許可または Write 許可ではありません)。

2.2.1.3 %Admin_Manage

このリソースは、複数の特権セットを制御します。

- [システム管理] ページを始めとする、管理ポータルのさまざまなページへのアクセスを制御します。
- 以下の機能を制御します。
  - InterSystems IRIS 構成の作成、変更、および削除
  - バックアップ定義の作成、変更、および削除
  - データベースの追加、データベースの特性の変更、およびデータベースの削除
  - ネームスペース・マップの変更
  - データベースとジャーナルのリストア
  - ジャーナリングなしプロセス・フラグの設定およびクリア。このフラグは、ターミナルのプログラマ・モードで、ENABLE^%SYS.NOJRN と DISABLE^%SYS.NOJRN のエントリ・ポイントを使用して設定およびクリアします。他の管理特権なしでユーザがこのタスクを実行できるようにする場合、**%Admin_Journal** リソースを使用します。

このリソースには Use 許可が必要です。

2.2.1.4 %Admin_Operate

このリソースは、複数の特権セットを制御します。

- [システムオペレーション] ページを始めとする、管理ポータルのさまざまなページへのアクセスを制御します。

- 以下の機能を制御します。
  - InterSystems IRIS の開始と停止
  - プロセスの検証と終了
  - データベースのマウントとディスマウント
  - 整合性の確認
  - ジャーナルの開始、停止、および切り換え
  - データベースのバックアップ
  - ロックの検証と削除
  - ログの検証
  - サービスの開始と停止

データベースをマウントするには、**%Admin_Operate:Use** 特権が必要です。これは、明示的なマウント (ObjectScript ユーティリティを使用した場合など) でも、暗黙的なマウント (マウントしていないデータベースへのグローバル参照を作成した場合など) でも同様です。

このリソースには Use 許可が必要です。

2.2.1.5 %Admin_RoleEdit

以下の特権を制御します。

- SQL の場合は、以下の機能を制御します。
  - ロールの作成または削除。

このリソースには Use 許可が必要です。

2.2.1.6 %Admin_Secure

このリソースは、複数の特権セットを制御します。

- 管理ポータルのさまざまなページへのアクセスを制御します。
- RBAC セキュリティ・モデルを操作している場合は、以下の機能を制御します。
  - ユーザの作成、変更、または削除。
  - ロールの作成、変更、または削除。
  - アプリケーション定義とアプリケーション・リソースの作成、変更、または削除。
  - 監査設定の変更
  - サービスの変更
- SQL の場合は、以下の機能を制御します。
  - ユーザの作成、変更、削除。
  - ロールの作成、変更、削除。
  - ユーザに付与された特権の参照。
  - ロールに付与された特権の参照。
  - 別のユーザから付与された SQL 特権の取り消し。

このリソースには Use 許可が必要です。

2.2.1.7 %Admin_Tasks

このリソースの特権には、管理ポータルのタスク・マネージャ（**[システムオペレーション]** > **[タスクマネージャ]**）などを使用してタスクを生成、変更、および実行する機能が含まれます。

このリソースには Use 許可が必要です。

2.2.1.8 %Admin_UserEdit

以下の特権を制御します。

- SQL の場合は、以下の機能を制御します。
  - ユーザの作成、変更、削除。

このリソースには Use 許可が必要です。

2.2.2 %Development リソース

%Development リソースは、InterSystems IRIS の開発機能と管理ポータルのさまざまなページへのアクセスを制御します。特に、以下の機能を制御します。

- ダイレクト・モードへの切り換え
- スタジオの使用**%Development:Use** 特権は、スタジオがサーバに接続するたびにチェックされます。
- InterSystems IRIS システム・マネージャ・ユーティリティのグローバル、ルーチン、クラス、テーブル、または SQL のいずれかの機能の使用(この機能にプログラム的にアクセスする API を呼び出す場合も、この特権が必要です)。
- InterSystems IRIS のデバッグ機能の使用。BREAK コマンドと ZBREAK コマンド、InterSystems IRIS システム・マネージャ・ユーティリティでのプロセス表示のデバッグ・オプションなどがあります。

%Development:Use 特権は、データベース特権との組み合わせで動作し、以下のように InterSystems IRIS への開発者のアクセスを制御します。

- スタジオの場合は、スタジオがサーバに接続するたびに **%Development:Use** 特権があるかどうかがチェックされます。接続するには、そのサーバに対する **%Development:Use** 特権を持ち、ネームスペースの既定のグローバル・データベースを読み取ることができる（つまり、**%DB_<database-name>:R** 特権を持つ）必要があります。ルーチン、クラス、または他の定義を開くには、それが格納されているデータベース（既定のルーチン・データベースであるかどうかは問いません）に対する Read 特権を持っている必要があります。定義をコンパイルまたは保存するには、そのデータベースに対する Write 特権を持っている必要があります。
- InterSystems IRIS システム・マネージャ・ユーティリティのグローバル、ルーチン、またはクラスの各機能の場合、グローバルにアクセスするユーザまたはグローバルを変更するユーザには、それぞれ該当のデータベースに対する Read 特権または Write 特権が必要です。
- InterSystems IRIS システム・マネージャ・ユーティリティの SQL 機能の場合、テーブル、ビュー、ストアド・プロシージャ、または他の SQL アセットに対する適切な SQL 特権がユーザに必要です。データベース・テーブルに対する何らかの SQL アクセスがユーザに許可されると、そのデータベースに対する Read アクセスまたは Write アクセスも付与されます。

InterSystems IRIS アプリケーションをデバッグするために、特定のデータベース特権が必要になることはありません。システムに対する **%Development:Use** 特権があれば、そのシステム上のあらゆるデータベースに格納されているあらゆるルーチンにブレークポイントを設定できます。ただし、以下の操作ではデータベースに対する Read 特権が必要です。

- デバッガを使用してルーチンのソースを表示する

- ルーチンを実行する

2.2.3 %DocDB_Admin リソース

%DocDB_Admin リソースは、ドキュメント・データベース・アプリケーションを管理する機能を制御します。この機能の詳細は、"ドキュメント・データベースの使用法" を参照してください。

2.2.4 %IAM リソース

%IAM リソースは、InterSystems API Manager (IAM) を実行するためのライセンスを InterSystems IRIS から取得する機能を制御します。

2.2.5 %System_Callout リソース

%System_Callout リソースは、InterSystems IRIS 外でアクションを実行するさまざまなツールへのアクセスを制御します。これには、以下のものがあります。

- ObjectScript での $ZF(-100) 関数の使用。これは、ObjectScript コード内からのオペレーティング・システム・コマンドの実行をサポートしています。また、"オペレーティング・システム・コマンドの発行" も参照してください。ここには、**%System_Callout:Use** 特権を追加する詳細手順が記載されています。
- ターミナルでの、オペレーティング・システムにアクセスする制御文字としての "!" および "$" の使用。詳細は、$ZF(-100) のドキュメントを参照してください。
- ObjectScript を使用したローカル・プロセス間通信における、Q モードでのプロセス間通信デバイスのオープン。詳細は、"プロセス間通信パイプの OPEN のみのコマンド・キーワード" の表を参照してください。

注釈　**%System_Callout** は、非推奨の $ZF(-1) 関数および $ZF(-2) 関数とのやり取りも制御します。

2.2.6 %System_Attach リソース

%System_Attach リソースでは、実行中のプロセスにスタジオ・デバッガをアタッチできます。

2.2.7 %Secure_Break リソース

%Secure_Break リソースは、保護されたデバッグ・シェルの使用に適用されます。これは、<BREAK> プロンプトでのプログラマのアクセスを制限します。保護されたデバッグ・シェルの詳細は、"保護されたデバッグ・シェル" を参照してください。

2.2.8 %Service_Native リソース

%Service_Native リソースは、Python、Java、.NET、Node.js のどれを通じてユーザが Native SDK を発行できるかを制御します。ユーザには **Use** 許可が必要です。システム定義のロールである **%Developer** と **%Manager** には既定でこの特権があります。

2.3 データベース・リソース

データベース・リソースは、InterSystems IRIS データベースのコンテンツへのアクセスを制御します。データベースへのアクセスを制御するデータベース・リソースの名前は、そのデータベースのラベル・ブロックに格納されます。

データベース・リソースの名前はすべて、文字列“%DB_”で始める必要があります。また、カスタム・リソースの場合、アンダースコアの次の文字にパーセント文字を使用することはできません。既定のデータベース・リソース名は、**%DB_<** **>** です。管理ポータルを使用して、データベースに割り当てられたリソース名を変更できます。

2.3.1 データベース・リソースの特権

利用できるデータベースの特権は、以下のとおりです。

テーブル 2-1: データベースの特権

許可	可能な操作
Read	データへのアクセスとルーチンの実行
Write	データ（実行可能コードも含む）の変更と削除

Read 許可および Write 許可では、データベースのすべてのコンテンツへのアクセスが可能です。このコンテンツには、データのほか、ソース・コードと実行可能コードも含まれます。インターシステムズのセキュリティ管理ユーティリティでは、Write アクセスが可能なデータベース・リソースには、自動的に Read 許可が与えられます。

データベースの特権では、ルーチンやグローバルなど、データベースに含まれる項目に対して個別に保護が可能になるわけではありません。データベースにあるリソースのすべての項目に、同一の保護が適用されます。独立したデータベースにグローバルとルーチンを格納することで、高度に細分化した保護を確立できます。InterSystems IRIS ネームスペースのマッピングを使用すれば、アプリケーション・レベルを変更せずに、このような保護を実現できます。

注釈　SQL セキュリティではテーブル・レベルのアクセスが与えられ、SELECT や UPDATE など、実行可能な特定のアクションが指定されます。SQL とセキュリティに関する詳細は、″SQL のユーザ、ロール、および特権″を参照してください。

2.3.2 共有データベース・リソース

多くの場合、データベースとそのデータベースの保護に使用するリソースは、一対一で対応します。例えば、IRISSYS データベースの保護を指定するには、**%DB_IRISSYS** リソースを使用します。ただし、これは必ずしも必須ではありません。複数のデータベースで同じセキュリティ定義を共有している場合には、それらデータベースで同じセキュリティ・リソースを共有できます。

3 つのデータベースを使用する営業アプリケーションを考えてみます。システム管理者は、データベースごとにアクセスを定義する代わりに、以下のような処理を行うことができます。

1. **%DB_SALES** などの新しいデータベース・リソースを作成します。
2. このリソースを 3 つのデータベースに割り当てます。
3. **%DB_SALES** への適切なアクセスを指定します。これにより、3 つのデータベースすべてに対するアクセスが、このリソースで制御されます。

2.3.3 既定のデータベース・リソース

データベース・リソース名を持たない既存データベースをマウントしたときは、既定リソースの **%DB_%DEFAULT** がデータベースに割り当てられます。既定では、**%DB_%DEFAULT** には以下の許可が付与されています。

テーブル 2-2: %DB_%DEFAULT の特権

ロール	許可
%Developer	Read、Write
%Manager	Read、Write

%DB_%DEFAULT リソースに関連付けられた特権は変更できますが、名前のないデータベースがマウントされた場合に利用できる必要があるため、**%DB_%DEFAULT** リソース自体は削除できません。

2.3.4 認識されないリソース名、または無効なリソース名

1 つの例外 (以下を参照) を除き、認識されないリソース名または無効なリソース名を設定したデータベースをマウントしようとしても、マウントできません(異なる InterSystems IRIS インスタンス間でデータベースを移動すると、このようなエラーが発生する場合があります)。自動的にマウントしようとするとエラーで失敗し、明示的にマウントしようとすると、データベース・ラベルにある名前を持つリソースを作成するか、使用するリソースを有効なものに変更するか選択するように求められます。

この規則の唯一の例外として、**%All** ロールのメンバであるユーザは、リソースがないデータベースをマウントできます (リソースが削除された場合や、データベースが以前は別のシステム上にあった場合など)。

2.3.5 ネームスペース

ユーザとアプリケーションは、ネームスペースを通じて InterSystems IRIS データベースと対話します。ネームスペースに関連付けられた特権は存在しませんが、その基盤となっているデータベースに関連付けられた特権に基づいて、ネームスペースへのアクセスが付与または拒否されます。具体的には、ネームスペースにアクセスするには、そのネームスペースに関連付けられた既定のグローバル・データベースに対する Read 特権を保持している必要があります。この要件は、以下の場合に確認されます。

- プロセスが、$NAMESPACE 特殊変数、ZNSPACE コマンド、%CD ユーティリティなどを使用することによって、別のネームスペースに変更しようとする場合。
- SQL 接続やオブジェクト接続など、ネームスペースに接続するサービスを使用して InterSystems IRIS に接続しようとする場合。

注釈　ネームスペースに対して暗黙的または明示的に、グローバルまたはルーチンによる参照を作成する場合、この要件は確認されません。

ネームスペースの特権はその基盤となっているデータベースの特権に依存するため、予期しない動作が発生する可能性があります。例えば、ネームスペース NSCust が、DBCust1、DBCust2、および DBCust3 という 3 つのデータベース内のデータを参照するとします。さらにロール **AverageUser** があり、特権 **%DB_DBCust1:R** と **%DB_DBCust3:R** を持つとします。このロールには DBCust2 に関連付けられた特権がないため、そのデータベース内のデータへのアクセスの試行はすべて失敗します (ネームスペースを介したアクセスを含む)。

2.3.6 IRISSYS (マネージャ・データベース)

InterSystems IRIS には、管理ルーチンおよびグローバルのリポジトリを提供するデータベースが付属しています。**IRISSYS** データベースがそのデータベースで、マネージャ・データベースとも呼ばれます。

このデータベースには、パーセント記号で始まる名前を持つグローバルとルーチンのグループがあります (これらはそれぞれ“パーセント・グローバル”および“グローバル・ルーチン”と呼ばれます)。これらのグローバルとルーチンは、InterSystems IRIS サイトの管理で特別な役目を持っており、以下のような特別な適用規則を備えています。

- すべてのユーザは、パーセント・ルーチンおよびパーセント・グローバルに対する Read 許可を持っています。

  マッピングを使用すれば、これらの項目を格納する場所を変更できますが、これらの表示には影響しません。パーセント・ルーチンおよびパーセント・グローバルはすべてのネームスペースで常に表示されます。

- すべてのパーセント・ルーチンは、同じデータベースに存在するすべてのグローバル (パーセント・グローバルのほか、非パーセントのグローバルも含む) に対する Write 許可を持っています。例えば、**IRISSYS** データベースにあるパーセント・ルーチンは、そのデータベースに格納されているグローバルに対して Write アクセス許可を持っていますが、他のデータベースにあるグローバルに対してはこの許可を持っていません。同時に、他のデータベースにあるパーセント・ルーチンは、その同じデータベースに格納されているグローバルに対して暗黙の Write アクセス許可を持ちますが、**IRISSYS** にあるパーセント・グローバルに対してはこの許可を持っていません。この暗黙の Write 許可は、通常のルーチンを実行しているときにのみ有効です。ルーチンが変更されていて、XECUTE コマンドでも引数による間接指定でもそのルーチンを使用できない場合、この許可は無効になります。

- **[システムワイドセキュリティパラメータ]** ページ (**[システム管理]→[セキュリティ]→[システム・セキュリティ]→[システムワイドセキュリティパラメータ]**) の **[パーセントで始まるグローバルへの書き込みを有効に]** フィールドを使用して、パーセント・グローバルに対する Write アクセス許可を非パーセント・ルーチンから制御できます。このページの説明は、"システム規模のセキュリティ・パラメータ" を参照してください。

注意　　**IRISSYS** データベースの移動、置換、削除は行わないでください。

2.3.6.1 特別な機能

IRISSYS データベースにあるコードで使用できる特別な機能があります。これらの機能を、“制限付きシステム機能”と呼ぶことがあります。この機能の働きは、以下のとおりです。

- 保護された VIEW コマンドおよび $VIEW 関数を呼び出す。
- 保護されたクラス・メソッドを使用する。
- `SET $ROLES = ...` 呼び出しを使用してプロセスのロールを変更する。
- 引数を 1 つ使用した形式で $SYSTEM.Security.Login (**%SYSTEM.Security** クラスの Login メソッド) 関数を呼び出す。
- 引数を 2 つ使用した形式で $SYSTEM.Security.ChangePassword (**%SYSTEM.Security** クラスの ChangePassword メソッド) 関数を呼び出す (新しいパスワードは、"ユーザ・アカウントのプロパティ" で説明されている一般的なパスワード制約および "パスワードの強固さとパスワードのポリシー" で説明されているインスタンス固有のパスワード制約に従う必要があります)。
- $ZF 関数の 1 つを呼び出す。これにより、ObjectScript ではないプログラムまたは関数を ObjectScript ルーチンから呼び出すことができます。

注釈　　VIEW コマンドを使用してデータベースのブロックに対する読み取りまたは書き込みを実行するとき、データベース特権は不要です。

以下のコードのみが、上記のアクションを実行できます。

- IRISSYS データベースに格納されているルーチン (″通常″ のルーチンの実行中のみ)。現在のルーチンへの ZINSERT によりこのルーチンが変更されている場合、これらの機能は無効になり、XECUTE コマンドでも引数による間接指定でも利用できなくなります。
- **%DB_IRISSYS** リソースに対する Write 許可を持っているプロセス。

2.4 ゲートウェイ・リソース

ゲートウェイ・リソースは、InterSystems IRIS に用意されている外部言語サーバ (ゲートウェイともいいます) へのアクセスを制御します。ゲートウェイ・リソースとそれが既定で関連付けられている外部言語サーバの種類を以下に挙げます。

- **%Gateway_ML** — IntegratedML
- **%Gateway_Object** — .NET、Java、Python、R、XSLT
- **%Gateway_SQL** — JDBC

これらのリソースには Use 許可が必要です。

%Admin_ExternalLanguageServerEdit リソースは、外部言語サーバを作成、削除、変更する機能を制御します。外部言語サーバに関連付けたゲートウェイ・リソースの変更も対象となります。ゲートウェイに関連付けられた既定のゲートウェイ・リソースを、ユーザ定義のリソースに置き換えることができます。また、置き換えずに削除することもできますが、その場合、関連付けられていたゲートウェイはパブリックになり、誰でも使用できるようになります。

重要　　ゲートウェイ・リソースに関連付けることにより、すべてのゲートウェイを保護することを強くお勧めします。

2.5 アプリケーション・リソース

InterSystems IRIS では、数種類の形式のカスタム承認をサポートしています。それらのすべては、アプリケーション・リソースとして知られるユーザ定義のリソースに依存しています。これには、以下のものがあります。

- ポータル・ページのための補足的な承認の確認 — 詳細は、″管理ポータルによるカスタム・リソースの使用法″ を参照してください。
- アプリケーションでの特定の時点における承認の確認 — 詳細は、次の ″リソースの作成または編集″ を参照してください。
- アプリケーション全体の承認

アプリケーション全体に対して、InterSystems IRIS はユーザ定義のアプリケーションと関連付けられたアプリケーション定義を作成できます (これ自体は、実行可能なコードで構成される名前付きエンティティとして定義されます)。アプリケーション・リソースによりアプリケーションの承認を確認できます。アプリケーションには、以下のようないくつかのタイプがあります。

- Web アプリケーション定義
- 特権ルーチン・アプリケーション定義
- クライアント・アプリケーション定義
- ドキュメント・データベース定義

アプリケーション・リソースは、アプリケーションに対するアクセスを制御する手段を提供します。この機能を使用するには、カスタム・リソースを作成し（"リソースの作成または編集" を参照）、アプリケーションに関連付けて使用します（詳細は、"Web アプリケーションの編集：[一般] タブ" または "特権ルーチン・アプリケーション、クライアント・アプリケーション、またはドキュメント・データベース・アプリケーションの編集：[一般] タブ" を参照）。

例えば Web アプリケーションに関連付けられたリソースがある場合、ユーザは、そのリソースに対して Use 許可を持つ場合にのみ、そのアプリケーションを実行できます。アプリケーションでリソースを規制するその他のエンティティ（データベースなど）が使用されている場合は、ユーザにもこれらのリソースに対する適切な許可を与えて、アプリケーションを効率的に操作する必要があります。アプリケーションの詳細は、"アプリケーション" を参照してください。

2.6 リソースの作成または編集

新規のリソースを作成するには、**[リソース]** ページ（**[システム管理]**→**[セキュリティ]**→**[リソース]**）で、**[新規リソース作成]** をクリックします。

既存のリソースを編集するには、**[リソース]** ページ（**[システム管理]** > **[セキュリティ]** > **[リソース]**）で、編集するリソースの右にある **[編集]** ボタンをクリックします。

[リソースの編集] ページが表示されます。**[リソースの編集]** ページには、以下のフィールドがあります。

- リソース名 ー リソースを識別するための文字列。リソース名の詳細は、"リソースの名前付け規約" を参照してください。リソースを作成する場合に、このフィールドは編集可能になります。既存のリソースを編集する場合は、編集不可能な文字列が表示されます。
- 説明 ー リソースに関連するオプション・テキスト。
- パブリック許可 ー
  - Read ー チェックが付いている場合は、すべてのユーザがこのリソースを参照できることを表します。
  - Write ー チェックが付いている場合は、すべてのユーザがこのリソースを参照または変更できることを表します。
  - Use ー チェックが付いている場合、すべてのユーザがこのリソースを実行または使用できることを表します。

リソースを追加すると、これがリソースのテーブルに表示され、アプリケーション・タイプになります。これは、アプリケーション固有の承認の一部として使用できます。詳細は、"プロセスの特権の確認" を参照してください。

2.6.1 リソースの名前付け規約

InterSystems IRIS リソースの名前はパーセント記号の文字で始まります。アプリケーション定義リソースの名前は、パーセント記号文字で始めないでください。

リソース名では大文字と小文字が区別されません。したがって、以下の点に注意します。

- 大文字と小文字が混在した名前を定義すると、入力したとおりの文字の組み合わせで名前が保持されます。
- 大文字と小文字の違いのみで別の名前とすることはできません。
- 名前の検索では、大文字と小文字の違いは無視されます。

例えば、**Accounting** という名前のリソースがある場合、**ACCOUNTING** という名前のリソースを作成することはできません。**Accounting** リソースを参照する場合、**accounting** や **ACCOUNTING** のようにすべて大文字または小文字を使用しても正常に実行されます。

2.7 管理ポータルによるカスタム・リソースの使用法

既定では、**%Admin_Manage**、**%Admin_Operate**、**%Admin_Secure**、および **%Development** のシステム・リソースにより、管理ポータルへのアクセスが制御されます。よりきめ細かくポータルのセキュリティを実現できるようにこれらを補足すると、さらにカスタム・リソースを各ポータル・ページと関連付けることができます。関連付けられるカスタム・リソースがポータル・ページにある場合、そのページを表示するためには、ユーザがそのページのシステム・リソースとカスタム・リソースの両方を保持する必要があります。

例えば、[**ロックテーブル**] ページにアクセスするには、**%Operator** ロールが必要です。カスタム・リソース (例えば、**MyLockTable**) を [**ロックテーブル**] ページに関連付けることもできます。この関連付けを作成したら、[**ロックテーブル**] ページを表示するには、ユーザは **%Operator** ロールのメンバであり、さらに **MyLockTable:Use** 特権がある必要があります。これにより、**%Operator** ロールでアクセス権が付与されるページの数が、既定の設定のインスタンスよりも少なくなります。そして、**[ ]** ページや **%Operator** ロールでアクセス権が付与されるその他のすべてのページを表示できる、新しいロールを定義できます。複数のカスタム・リソースも作成できます。これによって、事前定義済みロールが既定で用意しているもののさまざまなサブセットにさまざまなロールでアクセスできます。

ここでは以下について説明します。

- カスタム・リソースの定義とページへの適用
- ページからのカスタム・リソースの削除

重要　さまざまなページ、リソース、およびロール間における相互作用は複雑な場合があるので、管理ポータルのカスタム・リソースを実装する前に、システム管理者は注意深く計画する必要があります。

2.7.1 カスタム・リソースの定義とページへの適用

カスタム・リソースを定義して適用する手順は以下のとおりです。

1. **%Admin_Secure:Use** 特権のあるユーザ、または **%All** ロールのメンバとしてログインします。
2. カスタム・リソースを作成します。このためには、[**リソース**] ページ ([**システム管理**] > [**セキュリティ**]、[**リソース**]) で、[**新規リソース作成**] をクリックします。リソースを作成する際、インスタンスのニーズに従ってパブリック許可を適切に設定してあることを確認します。
3. カスタム・リソースを使用する特権をロールに関連付けます。既存のロールの場合、[**ロール**] ページ ([**システム管理**] > [**セキュリティ**] > [**ロール**]) で、特権をロールに追加するだけです。または、(同じく [**ロール**] ページで) 新しいロールを作成してからその直後に特権を追加します。どちらの方法でも、特権はカスタム・リソースと Use 許可で構成されます。
4. カスタム・リソースをページに割り当てます。以下はその方法です。
   a. ポータルの検索機能を使用して、ページを選択します。ページの名前をクリックすると直接そのページに移動します。ページのアクション・ペインを表示するには (名前自体ではなく) ボックス内をクリックしてください。
   b. ページのアクション・ペインの一番下にある [**割り当てる**] をクリックします。[**カスタム・リソースの割り当て**] ダイアログが表示されます。
   c. そのダイアログで、適切なリソースを [**カスタム・リソース名**] リストで選択し、[OK] をクリックします。

2.7.2 ページからのカスタム・リソースの削除

カスタム・リソースの関連付けをページから削除するには、以下の手順に従います。

1. **%Admin_Secure:Use** 特権のあるユーザ、または **%All** ロールのメンバとしてログインします。
2. ポータルの検索機能を使用して、ページを選択します。ページの名前をクリックすると直接そのページに移動します。ページのアクション・ペインを表示するには (名前自体ではなく) ボックス内をクリックしてください。
3. ページのアクション・ペインの一番下にある [**割り当てる**] をクリックします。[**カスタム・リソースの割り当て**] ダイアログが表示されます。
4. そのダイアログで、空の項目を [**カスタム・リソース名**] リストで選択し、[OK] をクリックします。

3

特権および許可

許可によって、ユーザは、データの読み取りや書き込み、ツールの使用など、何らかのアクションを実行できます。許可はリソースと関連付けられて、特権を形成します。特権は、`%DB_Sales:Read` のように、リソース名の後にコロンで区切って許可を記す書式で記述します。これは、ユーザが実行できるアクションを示します。

特権のグループは、ロールと呼ばれます。アクションを実行するユーザは、適切な特権を保有するロールのメンバであることが必要です。このモデルにより、ユーザ（またはユーザのグループ）が実行できる操作を指定する際に、精度が確保されます。調整を行うには、そのユーザのロールで特権を調整するだけです。

3.1 特権の機能

特権によって、リソースが許可に関連付けられます。この関連付けにより、特権を保持しているロールは、データベースに対する読み取りや書き込み、アプリケーションの使用などの特定のアクションを実行できます。使用できる許可は以下のとおりです。

- Read ー データベースなどのリソースのコンテンツを閲覧できますが、それを変更することはできません。
- Write ー データベースなどのリソースのコンテンツを閲覧および変更できます。
- Use ー アプリケーションや InterSystems サービスなどの実行可能プログラムやツールを実行、またはその他の方法で使用します。

それぞれの許可の意味は、それと組み合わせるリソースによって異なります。許可の名前は、そのフル・ネームまたは先頭の文字で表すことができます。この名前では、大文字と小文字が区別されません。

3.2 パブリック許可

リソースごとに、許可をパブリックとして指定できます。これは、リソースに対してこの許可を保持しているすべてのユーザが実質的に同等の扱いになるということです。例えば、**`%DB_SALES:Read`** 特権がパブリックであれば、どのユーザでも %DB_SALES リソースで保護されているデータベースをすべて読み取ることができます。ただし、この例では **`%DB_SALES:Write`** 特権がパブリックではないので、これらのデータベースにすべてのユーザが書き込めるわけではありません。

以下のデータベースは既定でパブリックになっています。

テーブル 3-1: 既定のパブリック特権

リソース	許可
%DB_IRIS	Read
%DB_IRISLIB	Read
%DB_IRISTEMP	Read、Write

3.3 プロセスの特権の確認

InterSystems IRIS® データ・プラットフォームには、$SYSTEM.Security.Check という、現在のプロセスで保持されている特権を確認するためのメソッドが用意されています。1 つの引数で構成した形式でこの関数を使用すると、プロセスが特定のリソースに対して保持している特権が示されます。2 つの引数で構成した形式では、特定のリソースに対してプロセスが特権を保持しているかどうかが返されます(次のセクションで説明する、組み込みの権限確認機能を持つメソッドもあります)。

1 つの引数で構成した形式では、リソースに対してプロセスが保持している許可が、コンマ区切りリストで返されます。以下はその例です。

```
$SYSTEM.Security.Check("%DB_TESTDATABASE")
```

プロセスが %DB_TESTDATABASE に対して Read 許可および Write 許可を保持している場合は、READ,WRITE が返されます。許可名は、必ずすべて大文字のフルネームで返されます。プロセスがリソースに対して許可を保持していない場合、この関数では空文字列が返されます。

2 つの引数で構成した形式では、プロセスが特定の特権を保持しているかどうかを示す True または False の値 (1 または 0) が返されます。以下はその例です。

```
$SYSTEM.Security.Check("%DB_TESTDATABASE", "WRITE")
```

プロセスが %DB_TESTDATABASE リソースに対して Write 許可を保持している場合は 1 が返されます。

以下のように、許可のリストを指定してこの関数を呼び出すこともできます。

```
$SYSTEM.Security.Check("%DB_TESTDATABASE", "WRITE,READ")
```

ここで要求されたすべての許可をプロセスが保持している場合は 1、それ以外の場合は 0 が返されます。以下のように、確認する特権の最初の文字のみを使用することもできます。

```
$SYSTEM.Security.Check("%DB_TESTDATABASE", "W,R")
```

このメソッドの一般的な動作は以下のとおりです。

- パブリックなリソース特権に対しては、プロセスがその特権を明示的に保持しているかどうかに関係なく、必ず 1 が返されます。
- 許可名では、大文字と小文字が区別されません。
- 許可名は、上の例のようにフルネームで記述できるほか、先頭の文字のみに省略して記述することもできます。また、許可名では大文字と小文字が区別されません。したがって、“WRITE,READ”、“W,R”、および “R,Write” はすべて同じ許可を指定していることになります。

3.4 組み込みの特権確認機能を持つメソッドの使用

InterSystems IRIS では、メソッドを呼び出すプロセスが特定の特権を持っていることをメソッドで要求できます。

この機能では、Requires メソッド・キーワードを使用します。Requires メソッド・キーワードは、特権のコンマ区切りリストである、引用符で囲まれた文字列値を持ちます。各特権では、リソースとそれに関連する許可 (Use、Read、または Write) を標準形式で指定します。

例えば、MyAction メソッドに **Service_FileSystem:Use** 特権が必要な場合、そのシグニチャは以下のようになります。

```
ClassMethod MyAction() [ Requires="Service_FileSystem:Use"]
{
 // Method content
}
```

Requires キーワードに値がある場合、メソッドは、呼び出し側プロセスがそのメソッドを呼び出す際に必要な特権を持っている場合にのみ実行されます。プロセスに必要な特権がない場合は、<PROTECT> エラーが生成されます。

このキーワードを継承するメソッドは、キーワードをオーバーライドして新しい値を設定することによって、追加のリソースを要求できます。この要件を削除することはできません。

3.5 特権の変更が有効になるタイミング

InterSystems IRIS は、セキュリティ設定が格納されたデータベースを永続的に保持しています。InterSystems IRIS が起動されると、その情報が抽出され共有メモリのセグメントに配置されるため、統合化された設定への迅速なアクセスが可能になります。プロセスが実行されている間、そのプロセスに付与されている特権のキャッシュが独自に保持されます。これは、新しい特権が必要になり認可されると更新されます。

ロールや特権などを編集すると、その情報の永続コピーに変更が反映されます。この変更は、ユーザまたはアプリケーションの次回の認証時に認識されます。

4

ロール

ロールとは、特権の名前付き集合です。複数のユーザが特権の同じセットを必要とすることが多いため、ロールの使用は便利です。例えば、アプリケーションのすべてのユーザや特定のプロジェクトに携わっているすべての開発者が、特権の共通のセットを必要とすることがあります。ロールを使用すれば、特権のこのようなセットを 1 回定義しておくことで、関係するユーザにそれを共通で適用でき、将来発生する変更にも極めて容易に対応できます。

特権はロールに排他的に割り当てられます。また、ユーザに直接割り当てられることはありません。1 人のユーザにいくつかの特権を割り当てるには、そのような目的のロールを作成します。

注釈　テーブルのデータに対する SQL アクセスでは、InterSystems IRIS® データ・プラットフォームは行レベル・セキュリティをサポートしています。これを設定する方法の詳細は、"行レベル・セキュリティの追加" を参照してください。

4.1 ロールについて

各ロールには、以下のプロパティがあります。

テーブル 4-1: ロールのプロパティ

プロパティ名	プロパティの説明
名前	ロールの一意の識別子。有効な名前の詳細は、"名前付け規約" を参照してください。
説明	任意のテキスト。
特権	リソースと、ロールに関連付けられた許可との組み合わせ。ロールにはゼロ個以上の特権を保持できます。
メンバ	ロールに割り当てられているユーザまたはロール ([ロール編集] ページの [メンバ] タブに表示されます)。

これらは、**[ロール編集]** ページの **[一般]** タブに表示されます。このページは、[ロール] ページ (**[システム管理]** > **[セキュリティ]** > **[ロール]**) のテーブルにあるロールの行で [編集] を選択するとアクセスできます。

各ロールは、それに割り当てられたメンバを持つことができるほか、その割り当て先となっている他のロールをメンバとして持つこともできます。これらの関係については、"ロール、ユーザ、メンバ、および割り当て" で説明しています。

4.1.1 ロール割り当てについて

InterSystems IRIS は、さまざまなロール割り当てメカニズムもサポートしています。ロール割り当てメカニズムによって、特定の認証されたユーザに特定のロールを関連付けできるようになります。InterSystems IRIS は、こうした関連付けを使用して、そのユーザに許可する操作を決定します。それぞれのロール割り当てメカニズムは、1 つ以上の認証メカニズムに関連付けられています。InterSystems IRIS を構成する際、認証メカニズムとロール割り当てメカニズムのサポート対象組み合わせを指定します。

使用可能なロール割り当てメカニズムは、以下のとおりです。

テーブル 4-2: 認証メカニズムとロール割り当てメカニズム

認証メカニズム	ロール割り当てメカニズム
代行認証 (ZAUTHENTICATE)	ZAUTHENTICATE
インスタンス認証	ネイティブ認証 (このドキュメントで説明している主なアプローチ)
LDAP	LDAP
Kerberos	このオプションを含む認証 · 代行認証 (ZAUTHORIZE) · ネイティブ認証
オペレーティング・システム認証	このオプションを含む認証 · 代行認証 (ZAUTHORIZE) · LDAP · ネイティブ認証

認証なしアクセスをサポートするインスタンスでは、すべてのユーザは、UnknownUser アカウントと _PUBLIC アカウントに関連付けられている特権を持ちます。これらのアカウントの詳細は、"UnknownUser アカウント" と "_PUBLIC アカウント をそれぞれ参照してください。

注釈　ロール割り当ての方法に関係なく、ロール管理 (つまり、特定のロールへの特定の特権の関連付け) は、InterSystems IRIS 内部で行われます。

4.1.2 ロールの最大数

InterSystems IRIS の各インスタンスには、最大で 10,240 のロールを設定できます。

4.2 ロール、ユーザ、メンバ、および割り当て

ロールは 1 つ以上の特権を保持するコンテナです。ユーザをロールに関連付けると、そのユーザは関連付けられたロールの特権を実行できます。ユーザとロールの関連付けに使用される用語は以下のとおりです。

· ユーザはロールに割り当てられます。

· ユーザはロールのメンバです。

- ユーザはロールに含まれます。

これらの表現はすべて、互いに同等の意味を持ちます。

各ユーザを複数のロールに割り当てることができます。また、各ロールにはそのメンバとして複数のユーザを指定できます。同様に、各ロールを複数のロールに割り当てることができ、各ロールにはそのメンバとして複数のロールを指定できます。ロールには、そのメンバとしてユーザと他のロールの両方を指定できます。

ある 1 つのロールを別のロールに割り当てるとします。この場合、ロール A をロール B に割り当てるには、ロール A をロール B の“メンバ”として記述します。これで、ロール A をロール B に割り当てるように指定した、またはロール B がロール A に含まれるように指定したことと同じになります。

あるロールを他のロールに割り当てることで、最初のロールでは、2 番目のロールに関連付けられている特権を保持できます。これは、ロールに対するユーザの割り当ての関係に似ています。この場合は、ロールに関連付けられている特権をユーザが保持できます。したがって、ユーザがロールのメンバであり、そのロールが別のロールのメンバである場合、そのユーザは両方のロールに関連付けられている特権を保持します。

例えば、学生に対して **UndergraduateStudent**、**GraduateStudent**、および **GeneralStudent** の 3 つのロールを用意している大学があるとします。各学生は **UndergraduateStudent** または **GraduateStudent** のいずれかに割り当てられ、この 2 つのロールは両方とも **GeneralStudent** に割り当てられます。**GraduateStudent** に割り当てられた Elizabeth は、**GraduateStudent** に割り当てられた特権と **GeneralStudent** に割り当てられた特権の両方を保持します。また、**UndergraduateStudent** に割り当てられた James は、**UndergraduateStudent** に割り当てられた特権と **GeneralStudent** に割り当てられた特権の両方を保持します。

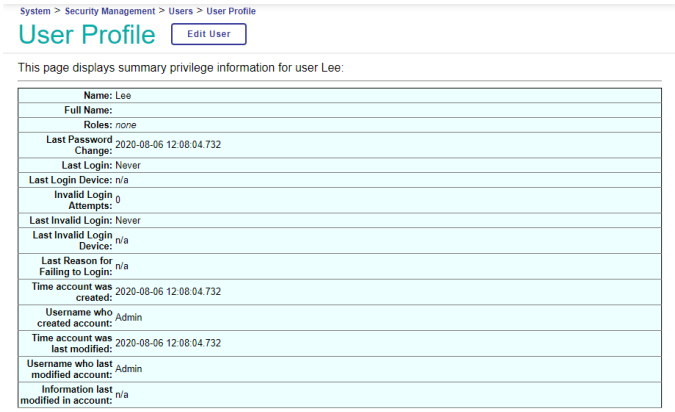
ロールのメンバは、[**ロール編集**] ページの [**メンバ**] タブに表示されます。このタブでは、ロールに新しいメンバを割り当てることもできます。ロールを他のロールに割り当てると、その割り当て先ロールは [**ロール編集**] ページの [**割り当て先**] タブに表示されます。このタブでは、ロールの割り当て先ロールを追加することもできます。

4.2.1 複数ロールの割り当て例

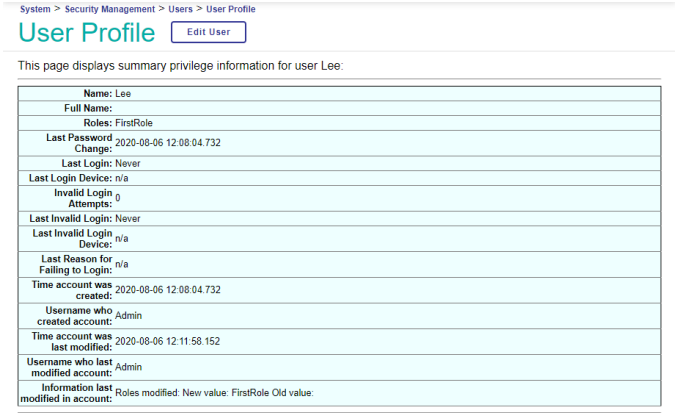
このセクションでは、InterSystems IRIS におけるユーザとロールとの関係の例について説明します。

Lee というユーザと、**FirstRole** および **SecondRole** という 2 つのロールがあるとします。**FirstRole** は **FirstResource** というリソースを保護し、**SecondRole** は **SecondResource** というリソースを保護します。

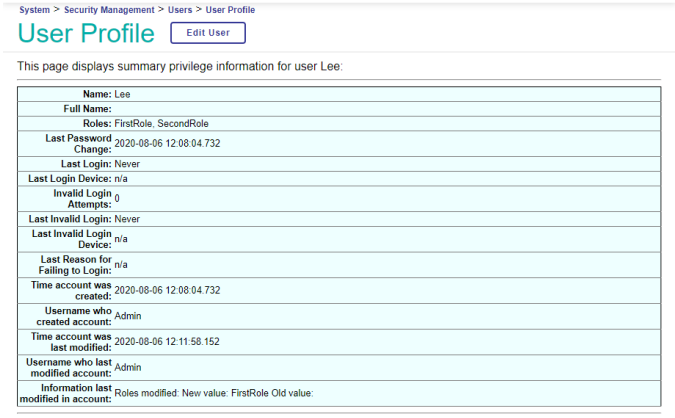
最初の作成時、Lee はどのロールのメンバでもありません。これは以下のように Lee のプロファイルに反映されます。



Lee がロール **FirstRole** に割り当てられると、Lee のプロファイルが変更されます。



ロール FirstRole がロール **SecondRole** に割り当てられる場合も、以下のように Lee のプロファイルが変更されます。



Lee の特権リストには、その特権がどのロールのものであるかが示されます。

Asset		Privileges			Source	
Resource	Protects	R	W	U	Granted by role	Granted by public resource
FirstResource		R	W	U	FirstRole:RWU	
SecondResource		R	W	U	SecondRole:RWU	

4.3 ロールの作成

開発者、オペレータ、システム管理者、およびその他のユーザ・クラスが使用するためにロールを定義できます。ロールを作成すると、ロールを編集するさまざまな機能が使用可能となります。

新規ロールを作成する方法は、以下のとおりです。

1. 管理ポータルのホーム・ページで、**[ロール]** ページ (**[システム管理]** > **[セキュリティ]** > **[ロール]**) に移動します。
2. **[ロール]** ページで、**[新規ロール作成]** をクリックします。**[ロール編集]** ページが表示されます。
3. **[ロール編集]** ページには **[一般]** タブがあります。ここで、以下のプロパティの値を入力します。
   - **[名前]** (必須) – 新規ロールの名前を指定します。名前付けのルールは、"名前付け規約" を参照してください。
   - **[説明]** (オプション) – ロールを説明する情報を指定します。

   ロールのリソースが一覧表示されますが、作成済みのロールでないとリソースを受け取ることができないので、この次の手順に記されている条件に該当しない限り、この段階ではリソースは空です。

4. 類似の特性を持つ複数のロールを簡単に作成するには、[**ロール**] ページにある [**コピー元**] フィールドを使用して新規ロールの作成を開始します。このフィールドのドロップダウン・メニューから既存のロールを選択すると、そのユーザのすべての特権がリソースのリストに表示されます。ここで、必要に応じて特権の追加や削除、特権の [説明] プロパティの変更などができます。
5. [**保存**] をクリックすると、指定したロールが作成されます。

ロールを作成すると、"ロールの管理" の説明に従ってそのロールを編集できます。例えば、[**リソース**] テーブルで [**追加**] をクリックすることにより、このロールに新しい特権を追加できます。

注釈　事前定義のロールを変更しないことをお勧めします。

4.3.1 名前付け規約

ユーザ定義ロールの名前は、文字の使用に関して以下の規則に従う必要があります。

- あらゆる英数字を使用できます。
- 禁止されている文字 (“,” (コンマ)、“:” (コロン)、“/” (スラッシュ)) を除く記号を使用できます。
- 先頭に “%” (パーセント記号) は使用できません。これは InterSystems IRIS の事前定義ロール用に予約されています。
- Unicode 文字を使用できます。
- 既存のユーザ名と同じユーザ名は指定できません。

また、ロール名では大文字と小文字が区別されません。したがって、以下の点に注意します。

- 大文字と小文字が混在した名前を定義すると、入力したとおりの文字の組み合わせで名前が保持されます。
- 大文字と小文字の違いのみで別の名前とすることはできません。
- 名前の検索では、大文字と小文字の違いは無視されます。

ロール名の最大長は 64 文字です。

例えば、**BasicUser** という名前のロールがある場合、**BASICUSER** という名前のロールを作成することはできません。**BasicUser** ロールを参照する場合、**basicuser** や **BASICUSER** のようにすべて大文字または小文字を使用しても正常に実行されます。

4.4 ロールの管理

ロールを作成すると、さまざまな方法でそのロールを変更できます。以下でそれぞれの方法について説明します。このアクションは以下のいくつかのカテゴリに分類できます。

- 一般的なタスク。これには以下が含まれます。
  - 既存のロールの確認
  - ロールの削除
- ロールの特権の作成、変更、および削除。これには以下が含まれます。
  - ロールに対する新しい特権の付与
  - ロールの特権の変更

  - ロールからの特権の削除

- ロールとユーザ間の割り当ての作成と削除。これには以下が含まれます。
  - 現在のロールに対するユーザまたはロールの割り当て
  - 現在のロールからのユーザまたはロールの削除
  - 他のロールに対する現在のロールの割り当て
  - 他のロールからの現在のロールの削除
- ロールの SQL 関連オプションの変更

注釈　ユーザのロールの変更またはロールの特権の変更は、ユーザの既存のプロセスと関連付けられている割り当て済みの特権には影響しません。新しい特権を有効にするには、ユーザはログアウトして再度ログインし、プロセスを再開するか、同等のアクションを実行する必要があります。

4.4.1 既存のロールの確認

現在の既存ロールのリストを表示するには、ポータルで **[ロール]** ページ (**[システム管理]**→**[セキュリティ]**→**[ロール]**) を表示します。このページでは、以下のフィールドに情報が表示されます。

- [名前] – ロールの名前 (これは編集できません)。
- [説明] – ロールの説明として指定されているテキスト。
- [作成者] – ロールを作成したユーザの名前。

それぞれのロールについて以下の操作が可能です。

- ロールのプロパティの編集。特権の管理、割り当ての管理、および SQL 関連オプションのすべてのアクションも対象となります。
- ロールの削除

4.4.2 ロールの削除

ロールを削除する方法は、以下のとおりです。

1. **[ロール]** ページ (**[システム管理]** > **[セキュリティ]** > **[ロール]**) で、削除するロールの行で **[削除]** をクリックします。
2. 操作を確認するダイアログが表示されます。そのロールを削除する場合は [OK] をクリックし、それ以外の場合は **[キャンセル]** をクリックします。

4.4.3 ロールに対する新しい特権の付与

ロールに新しい特権を付与する手順は、以下のとおりです。

1. 既存のロールの **[ロール編集]** ページ (**[システム管理]** > **[セキュリティ]** > **[ロール]** > **[ロール編集]**) で、**[権限]** テーブルにある **[追加]** ボタンをクリックします。
2. リソースをすべて一覧表示したページが表示されます。目的のロールに割り当てるリソースを選択するには、そのリソースをクリックします。Ctrl キーまたは Shift キーを使用すると、複数のリソースを同時に選択できます。
3. 選択したリソースをロールに追加するには、**[保存]** をクリックします。これによって、そのリソースで使用できるすべての許可がロールに与えられます。これで、このリソースで使用できる許可を変更できるようになります (例えば、データベースに対する許可を Read-Write から Read のみに変更できます)。

4.4.4 ロールの特権の変更

ロールに保持されている特権を変更する手順は、以下のとおりです。

1. 管理ポータルのホーム・ページで、[**ロール**] ページ ([**システム管理**] > [**セキュリティ**] > [**ロール**]) に移動します。
2. [**ロール**] ページで、編集するロールの [**編集**] をクリックします。[**ロール編集**] ページが表示されます。
3. [**ロール編集**] ページにある [**リソース**] テーブルで、変更対象の特権を持つリソースの [**編集**] をクリックします。
4. 選択したリソースに対する許可を編集するためのページが表示されます。必要に応じて、各許可のチェック・ボックスにチェックを付けるか、チェックを外します。

   注釈　このページでは、個々のリソースについてすべての許可のチェック・ボックスのチェックを外すことはできません。リソースからロールのすべての許可を削除することは、そのリソースからロールの特権を削除することと同じであるためです。

5. [**保存**] をクリックすると、新しい状態で特権が保存されます。

これらの変更内容は、[**ロール**] ページの [**リソース**] テーブルに反映されます。

4.4.5 ロールからの特権の削除

ロールから特権を削除する方法は、以下のとおりです。

1. 管理ポータルのホーム・ページで、[**ロール**] ページ ([**システム管理**] > [**セキュリティ**] > [**ロール**]) に移動します。
2. [**ロール**] ページで、編集するロールの [**編集**] をクリックします。[**ロール編集**] ページが表示されます。
3. [**ロール編集**] ページにある [**リソース**] テーブルで、[**削除**] をクリックします。ロールから、リソースの特権が削除されます。
4. [**保存**] をクリックすると、新しい状態で特権が保存されます。

4.4.6 現在のロールに対するユーザまたはロールの割り当て

ロールは、割り当てられたメンバとしてユーザまたは他のロールを持つことができます。ユーザをロールに割り当てると、そのユーザは割当先ロールに関連付けられている特権を保持します。あるロールを他のロールに割り当てると、最初のロールに割り当てられているユーザは、2 番目のロールに関連付けられている特権を保持します。

編集中のロールは “現在の” ロールとして認識されます。現在のロールに割り当てられているユーザとロールは、[**ロール編集**] ページの [**メンバ**] タブに表示されます (これらのユーザとロールはメンバとして認識されます)。

現在のロールにユーザまたはロールを割り当てる手順は以下のとおりです。

1. 管理ポータルのホーム・ページで、[**ロール**] ページ ([**システム管理**] > [**セキュリティ**] > [**ロール**]) に移動します。
2. [**ロール**] ページで、編集するロールの [**編集**] をクリックします。[**ロール編集**] ページが表示されます。
3. [**ロール編集**] ページで、[**メンバ**] タブを選択します。
4. [**メンバ**] タブで、[ユーザ] オプションまたは [ロール] オプションを選択して、このロールにユーザを割り当てるか、ロールを割り当てるかを指定します(既定は [ユーザ] です)。
5. 現在のロールに割り当てることができるユーザまたはロールのリストが、[**利用可能**] リストに表示されます。[**選択済**] リストとの間にある矢印ボタンを使用して、2 つのリストの間でこれらを移動できます。
6. 追加するユーザまたはロールの選択が完了した後、[**割り当てる**] または [Grant 付で割り当て] をクリックします。[**割り当てる**] をクリックすると、編集中のロールに新しいメンバ (ユーザまたはロール) が単純に割り当てられます。[Grant

付で割り当て] をクリックすると、新しいメンバで SQL コマンドを使用して、現在のロールに他のユーザまたはロールを割り当てることもできるようになります。

4.4.7 現在のロールからのユーザまたはロールの削除

現在のロールにユーザまたはロールを割り当てておくと、それはそのロールのメンバとして認識されます。ロールからメンバを削除する手順は以下のとおりです。

1. 管理ポータルのホーム・ページで、[**ロール**] ページ ([**システム管理**] > [**セキュリティ**] > [**ロール**]) に移動します。
2. [**ロール**] ページで、編集するロールの [**編集**] をクリックします。[**ロール編集**] ページが表示されます。
3. [**ロール編集**] ページで、[**メンバ**] タブを選択します。
4. [**メンバ**] タブには、現在のロールに割り当てられているユーザとロールのテーブルが表示されています。削除するメンバを示す行の最も右の列にある [**削除**] ボタンをクリックします。
5. 削除の処理を確認するプロンプトが表示されます。[OK] をクリックします。

指定したユーザまたはロールが現在のロールから削除されます。

4.4.8 他のロールに対する現在のロールの割り当て

ある 1 つのロールを別のロールに割り当てることができます。あるロールを他のロールに割り当てると、最初のロールに割り当てられているユーザは、2 番目のロールに関連付けられている特権を保持します。

編集中のロールは “現在の” ロールとして認識されます。現在のロールの割り当て先となっているロールは、[**ロール編集**] ページの [**割り当て先**] タブに表示されます。

現在のロールを他のロールに割り当てる手順は以下のとおりです。

1. 管理ポータルのホーム・ページで、[**ロール**] ページ ([**システム管理**] > [**セキュリティ**] > [**ロール**]) に移動します。
2. [**ロール**] ページで、編集するロールの [**編集**] をクリックします。[**ロール編集**] ページが表示されます。
3. [**ロール編集**] ページで、[**割り当て先**] タブを選択します。
4. 現在のロールを割り当てることができるロールのリストが、[**利用可能**] リストに表示されます。[**選択済**] リストとの間にある矢印ボタンを使用して、2 つのリストの間でこれらを移動できます。
5. 割り当て先とするロールの選択が完了した後、[**割り当てる**] または [Grant **付で割り当て**] をクリックします。[**割り当てる**] をクリックすると、選択したロールに現在のロールが単純に割り当てられます。[Grant **付で割り当て**] をクリックすると、現在のロールで SQL コマンドを使用して、選択したロールに他のユーザまたはロールを割り当てることもできるようになります。

4.4.9 他のロールからの現在のロールの削除

現在のロールを他のロールに割り当てておくと、現在のロールは割り当て先のロールのメンバとして認識されます。他のロールから現在のロールを削除する手順は以下のとおりです。

1. 管理ポータルのホーム・ページで、[**ロール**] ページ ([**システム管理**] > [**セキュリティ**] > [**ロール**]) に移動します。
2. [**ロール**] ページで、編集するロールの [**編集**] をクリックします。[**ロール編集**] ページが表示されます。
3. [**ロール編集**] ページで、[**割り当て先**] タブを選択します。
4. [**割り当て先**] タブには、現在のロールの割り当て先となっているロールのテーブルが表示されています。これらのロールのいずれかから現在のロールを削除するには、そのロールを示す行の最も右の列にある [**削除**] ボタンを選択します。

5. 削除の処理を確認するプロンプトが表示されます。[OK] をクリックします。

指定したロールから現在のロールが削除されます。

4.4.10 ロールの SQL 関連オプションの変更

どのロールに対しても、以下の SQL 関連の特性を付与または削除できます。

- 一般的な SQL 特権
- テーブルの特権
- ビューに対する特権
- ストアド・プロシージャの特権

4.4.10.1 一般的な SQL 特権

[ロール編集] ページの **[SQL 権限]** タブで、ロールの SQL 特権を追加または削除できます。

- ロールに特権を追加するには、まずその特権を **[利用可能]** リストから **[選択済]** リストに移動します (特権をダブルクリックするか、特権を選択して単線の右矢印をクリックします)。次に **[割り当てる]** をクリックすると、その特権がロールに付与されます。追加した特権を他のロールに付与できる特権も追加するには、**[利用可能]** リストの下にある該当のチェック・ボックスにチェックを付けます。
- ロールにすべての特権を追加するには、**[利用可能]** リストから **[選択済]** リストを指している二重線の矢印をクリックします。次に **[割り当てる]** をクリックすると、これらの特権がロールに付与されます。追加した特権を他のロールに付与できる特権も追加するには、**[利用可能]** リストの下にある該当のチェック・ボックスにチェックを付けます。
- ロールから特権を削除するには、特権の名前の右にある **[削除]** をクリックします。
- すべての特権をロールから削除するには、現在割り当てられている特権が表示されたテーブルの下にある **[すべて削除]** をクリックします。

利用できる特権は、以下のとおりです。

- %ALTER_TABLE – 指定されたネームスペースについて、ロールのメンバは ALTER TABLE コマンドを実行できます。
- %ALTER_VIEW – 指定されたネームスペースについて、ロールのメンバは ALTER VIEW コマンドを実行できます。
- %CREATE_FUNCTION – 指定されたネームスペースについて、ロールのメンバは CREATE FUNCTION コマンドを実行できます。
- %CREATE_METHOD – 指定されたネームスペースについて、ロールのメンバは CREATE METHOD コマンドを実行できます。
- %CREATE_PROCEDURE – 指定されたネームスペースについて、ロールのメンバは CREATE PROCEDURE コマンドを実行できます。
- %CREATE_QUERY – 指定されたネームスペースについて、ロールのメンバは CREATE QUERY コマンドを実行できます。
- %CREATE_TABLE – 指定されたネームスペースについて、ロールのメンバは CREATE TABLE コマンドを実行できます。
- %CREATE_TRIGGER – 指定されたネームスペースについて、ロールのメンバは CREATE TRIGGER コマンドを実行できます。
- %CREATE_VIEW – 指定されたネームスペースについて、ロールのメンバは CREATE VIEW コマンドを実行できます。

- %DROP_FUNCTION – 指定されたネームスペースについて、ロールのメンバは DROP FUNCTION コマンドを実行できます。
- %DROP_METHOD – 指定されたネームスペースについて、ロールのメンバは DROP METHOD コマンドを実行できます。
- %DROP_PROCEDURE – 指定されたネームスペースについて、ロールのメンバは DROP PROCEDURE コマンドを実行できます。
- %DROP_QUERY – 指定されたネームスペースについて、ロールのメンバは DROP QUERY コマンドを実行できます。
- %DROP_TABLE – 指定されたネームスペースについて、ロールのメンバは DROP TABLE コマンドを実行できます。
- %DROP_TRIGGER – 指定されたネームスペースについて、ロールのメンバは DROP TRIGGER コマンドを実行できます。
- %DROP_VIEW – 指定されたネームスペースについて、ロールのメンバは DROP VIEW コマンドを実行できます。

4.4.10.2 テーブルの特権

[ロール編集] ページの **[SQLテーブル]** タブで、ロールのテーブル関連 SQL 特権を追加または削除できます。

1. ページ上部近くにあるドロップダウンから、該当のネームスペースを選択します。そのネームスペースのテーブルのリストが表示されます。
2. テーブルの特権を変更するには、そのテーブルの行にある **[編集]** をクリックします。特権を変更するためのウィンドウが表示されます。
3. このウィンドウで、以下の項目のチェック・ボックスにチェックを付けるか、チェックを外します。
   - ALTER
   - DELETE
   - INSERT
   - REFERENCES
   - SELECT
   - UPDATE
   - このロールに付与特権オプションを付与
4. 以上の選択の後、**[適用]** をクリックすると、テーブルに新しい特権が設定されます。

ロールにテーブルに対する特権がある場合は、このページのテーブルにリストされます。テーブルに対するロールの特権を削除するには、ロールの行の一番右側にある **[削除]** をクリックします。これをクリックすると、“SAMPLES Sample.Company” のように、ネームスペースとテーブルの正式名称 (スキーマを含む) を含むメッセージが表示されます。

4.4.10.3 ビューに対する特権

[ロール編集] ページの **[SQLビュー]** タブで、ロールに対してビュー関連 SQL 特権を追加または削除できます。

ビューの特権を追加する手順は、以下のとおりです。

1. ページ上部近くにあるドロップダウンから、該当のネームスペースを選択します。そのネームスペースのビューのリストが表示されます。
2. ビューの特権を変更するには、そのビューの行にある **[編集]** をクリックします。特権を変更するためのウィンドウが表示されます。

3. このウィンドウで、以下の項目のチェック・ボックスにチェックを付けるか、チェックを外します。
   - ALTER
   - DELETE
   - INSERT
   - REFERENCES
   - SELECT
   - UPDATE
   - このロールに付与特権オプションを付与
4. 以上の選択の後、**[適用]** をクリックすると、テーブルに新しい特権が設定されます。

ロールにビューに対する特権がある場合は、このページのテーブルにリストされます。ビューに対するロールの特権を削除するには、ロールの行の一番右側にある **[削除]** をクリックします。これをクリックすると、ネームスペースとビューの正式名称 (スキーマを含む) を含むメッセージが表示されます。

4.4.10.4 ストアド・プロシージャの特権

[ロール編集] ページの **[SQLプロシージャ]** タブで、ストアド・プロシージャに関連する、ロールの SQL 特権を追加または削除できます。

ストアド・プロシージャの特権を追加する手順は、以下のとおりです。

1. ページ上部近くにあるドロップダウンから、該当のネームスペースを選択します。そのネームスペースのストアド・プロシージャのリストが表示されます。
2. このウィンドウの下にある **[追加]** をクリックすると、**[プロシージャ権限を...に付与]** ダイアログが表示されます。
3. このダイアログの上部にあるドロップダウンから、追加するプロシージャを含むスキーマを選択します。ページの左側部分にある **[使用可能]** ウィンドウに、このスキーマのプロシージャのリストが表示されます。
4. 1 つ以上のプロシージャを、**[選択済み]** ウィンドウに移動します。[EXECUTE] チェック・ボックスにチェックが付いていることを確認します。これによって、ストアド・プロシージャを実行する特権を、このロールで保持できるようになります。
5. 必要に応じて、この特権を他のロールに与える機能をこのロールに与えることができます。この処理を実行するには、このページの下部近くにある **[このユーザに Grant 権限を付与する場合はここをチェックします。]** ボックスにチェックを付けます。
6. **[適用]** をクリックすると、目的の特権がロールに与えられます。

ロールにストアド・プロシージャに対する特権がある場合、このページのテーブルにリストされます。ストアド・プロシージャに対するロールの特権を削除するには、ロールの行の一番右側にある **[削除]** をクリックします。これをクリックすると、ネームスペースとストアド・プロシージャの正式名称 (スキーマを含む) を含むメッセージが表示されます。

4.5 事前定義のロール

InterSystems IRIS には、以下のような多数の事前定義のロールが付属しています。これには、以下のものがあります。

- **%All** – すべてのオペレーションを実行する機能です。
- **%Developer** – 一般にアプリケーション開発に関連付けられる特権です。ほとんどがポータルの[システム開発] メニューに関連付けられる特権です。この特権として、管理ポータルで [システムエクスプローラ] ページ、[WebStress]

ページ、[UnitTest] ページを使用する機能のほか、ドキュメント・クラス・リファレンス (Documatic ともいいます) を使用する機能もあります。

- **%Manager** ― 一般にシステム管理に関連付けられる特権です。ほとんどがポータルの [システム管理] および [システム処理] メニューに関連付けられる特権です。
- **%Operator** ― 一般にシステム処理に関連付けられる特権です。ほとんどがポータルの [システム処理] メニューに関連付けられる特権です。
- **%SQL** ― 一般に SQL 関連タスクに関連付けられる特権です。
- **%SecureBreak** ― **%Secure_Break:Use** 特権です。保護されたデバッグ・シェルの使用に適用されます。保護されたデバッグ・シェルの詳細は、"保護されたデバッグ・シェル" を参照してください。

注釈　事前定義のロールを変更しないことをお勧めします。それよりも、事前定義のロールに基づいて新しいロールを作成し、その作成したロールを変更してください。

以下のテーブルには、ロールごとの列があります。テーブルの各行には、システム定義リソースと、そのリソースでロールが保持する特権がある場合はその特権が示されています。

テーブル 4-3: 事前定義ロールとその特権

リソース	%Developer	%Manager	%Operator	%SQL	%SecureBreak
%Admin_Manage		Use			
%Admin_Operate		Use	Use		
%Admin_Secure		Use			
%Admin_Task		Use			
%DB_IRISLOCALDATA	Read	Read	Read		
%DB_IRISAUDIT		Read			
%DB_IRISLIB	Read	Read、Write			
%DB_IRISSYS		Read、Write	Read、Write		
%DB_IRISTEMP	Read、Write	Read、Write	Read、Write		
%DB_%DEFAULT	Read、Write	Read、Write			
%Development	Use	Use			
%DocDB_Admin	Use	Use			
%Secure_Break					Use
%Service_Console					
%Service_DocDB	Use	Use	Use		
%System_Native	Use	Use			
%Service_Object	Use	Use			
%Service_SQL	Use	Use		Use	
%Service_Telnet	Use	Use			
%Service_Terminal	Use	Use			

リソース	%Developer	%Manager	%Operator	%SQL	%SecureBreak
%Service_WebGateway	Use	Use	Use		

これら事前定義されたロールの定義は、InterSystems IRIS を新規にインストールするときに設定され、以降のアップグレード・インストールでは変更されません。**%All** を除き、事前定義のロールの使用はオプションです。

%Admin_Secure リソースは、必要なすべてのセキュリティ・アセットを、ひとまとめの単位として使用可能にしたり、制限したりするために設計されています。これにより、セキュリティ管理者が使用できるように、これらのリソースを容易に分離しておくことができます。

注釈　既定では、**%Operator** ロールは **%Admin_Task:Use** 特権を持ちません。このロールのメンバがタスクを管理できるようにする場合、ロールの特権に **%Admin_Task:Use** を組み込んでください。さらに、**%Operator** に基づくカスタム・ロールはいずれも、ポータルの [**オペレータ**] メニューを使用するために **%DB_IRISSYS:RW** 特権を加える必要があります。また、それらのロールがタスクを管理できるように、**%Admin_Task:Use** 特権を加えることができます。

4.5.1 %All

事前定義のロール **%All** は、システム上のすべてのリソースに対するすべての特権を必ず保持しています。このため、**%All** ロールに属するユーザは、例えば使用可能なリソースがないデータベースをマウントすることもできます。(例外として **%Secure_Break:Use** 特権があります。これは、常に明示的に付与される必要があります。)

このロールは削除も変更もできません。また、**%All** ロールを保持しているユーザ・アカウントが必ず 1 つ以上存在している必要があります。そのようなアカウントが 1 つのみである場合、そのアカウントは削除することも、無効にすることもできません。これは、不注意な操作によって、唯一の InterSystems IRIS システム管理者がシステムからロックアウトされないようにするための設計です。

重要　**%All** ロールに割り当てられたユーザには、行レベル・セキュリティによって保護されているテーブルの行に対するアクセス権が自動的には付与されません。アプリケーション側で明示的にこのような行へのアクセス権を **%All** ロールに与える必要があります。これを行う方法の詳細は、"行レベル・セキュリティの追加" を参照してください。

4.5.2 既定のデータベース・リソース・ロール

データベース・リソースを作成すると、そのリソースに対する Read 許可と Write 許可を持つロール (**%DB_<database-resource-name>** という名前) が自動的に作成されます。**%DB_<database-resource-name>** ロールは読み取り専用であるため、変更できません。したがって、これらの各ロールに対して、ロールが指定されているデータベース・リソースへの RW アクセス以外に他のリソースに対する特権を追加することはできません。

4.6 ログイン・ロールおよび追加ロール

どのような時点でも、各 InterSystems IRIS プロセスには、そのプロセスに対する現在の特権を決定するロールのセットがあります。ロールのセットには、ログイン・ロールと追加ロールの両方があります。ログイン・ロールはユーザの定義に基づくもので、ログイン時に得られます。追加ロールは現在実行しているアプリケーションに基づくもので、アプリケーション・ロールのエスカレーションによって得られます。セキュリティの観点からいえば、ロールが何を基にしているかは問題ではありません。重要なことは、必要な特権がプロセスにあるかどうかという点です。

アプリケーションが起動すると、プロセスで現在保持されているロールがテーブル内で 1 つずつ検索され、関連付けられているアプリケーション・ロールがあれば追加されます。

例えば、通常のユーザと管理者という 2 つのクラスのユーザが使用する発注入力アプリケーションがあるとします。通常のユーザに割り当てられているロールは **OrderEntryUser** で、管理者に割り当てられているロールは **OrderEntryManager** です。どちらのロールを使用しても、この発注入力アプリケーションを実行できます (つまり、両方のロールに **%Application_OrderEntry:Use** 特権が割り当てられています)。ただし、アプリケーションでロールのエスカレーションが発生すると、別のロールが使用され (**OrderEntryAppNormal**、**OrderEntryAppSpecial** および **OrderEntryAppReporting**)、このアプリケーションでこれらのユーザ・クラスの代わりにさまざまな機能が実行できるようになります。

一致ロール	追加ロール
OrderEntryUser	**OrderEntryAppNormal**
OrderEntryManager	**OrderEntryAppSpecial**、**OrderEntryAppReporting**

一致を検索するシーケンスでは、プロセスで保持されている各ロールは、一致が既に見つかっている場合でも検索の対象となります。言い換えれば、複数のロールが一致し、その結果、新しいロールの複数のセットが追加されることがあります。ただし、このプロセスが繰り返されることはありません。つまり、一致プロセスの結果によって追加されたロールは、以降の一致検索では対象外となります。

注釈　　ユーザのロールをログイン・ロールより少なくするよう制限する方法はありません。

4.6.1 追加されたロールと管理ポータルでのアクセスに関するメモ

ユーザが新しいポータル・ページに移動すると、ポータルはユーザのログイン・ロールだけを保持するようにプロセスをリセットします。次に、ポータルはそのページのアプリケーションにリソースが必要かどうかをチェックし、リソースが必要な場合は、ユーザがそのリソースに対する適切な権限を持っているかどうかをチェックします。必要な特権がユーザの特権に含まれていない場合、そのページは使用できなくなります。

ユーザが必要な特権を持っている場合、ポータルはアプリケーション・ロールと該当するターゲット・ロールを追加します。次に、ポータルはページ上のリンクにカスタム・リソースが必要かどうかをチェックし、ユーザが適切なリソースを持っている場合は、それらのリンクを表示します。

4.7 プログラムで管理するロール

ルーチンによっては、$ROLES システム変数を以下のように設定することにより、実行中のプロセスのアプリケーション・ロールを直接変更できるものがあります。

```
SET $ROLES = "Payroll"
```

$ROLES には、現行プロセスに割り当てられたロール名のコンマ区切りのリストが格納されます。リスト内のすべてのロールに付与されているすべての特権の集合によって、プロセスが持つ特権が決められます。$ROLES には、認証時に割り当てられたロール (つまり ログイン・ロール) が初期ロールとして格納されています。

このコマンドを実行するには、IRISSYS データベースに組み込んだルーチンから呼び出すか、現在保持している特権に IRISSYS データベースに対する Write 許可 (**%DB_IRISSYS:W**) が含まれている必要があります。

$ROLES を設定することで変更されるのはプロセスの追加ロールのみであって、ログイン・ロールは変更されません。プロセスで現在保持されているログイン・ロールが Employee および Manager で、追加ロールが **Payroll** であるとします。

```
SET $ROLES = "Accounting"
```

上の文を実行した後、$ROLES の値は“Employee,Manager,Payroll,Accounting”になります。

プロセスの現在のロールの後にロールを続けて記述することで、現在のロールにロールを追加できます。これには以下のようなコールを使用します。

```
SET $ROLES = $ROLES _ ",Payroll"
```

以下の文を考えます。

```
SET $ROLES = ""
```

この文では、すべての追加ロールが削除されます。

$ROLES に NEW コマンドを組み合わせて使用し、ロール（ログイン・ロールと追加ロール）の現在のセットと $USERNAME の現在の値をスタックできます。これによりコードでリストを変更でき、制御によってその格納ブロックが正常または異常な状態で放置されても、終了時に変更が元に戻されます。

引数が NULL 文字である場合を除き、SET $ROLES = < > という処理はシステム機能です。NEW $ROLES および SET $ROLES = "" は、どのようなコードでも実行できます。

5
ユーザ・アカウント

ユーザ・アカウントは、InterSystems IRIS® データ・プラットフォームの実際のユーザを表します。ユーザ・アカウントがメンバになっているロールにより、そのユーザがアクセスできるリソースが決まります。ユーザ・アカウントに特定の SQL 特権を付与することもできます。

5.1 ユーザ・アカウントのプロパティ

InterSystems IRIS の各ユーザ・アカウントには、多数のプロパティがあります。アカウントのプロパティを確認するには、管理ポータルの [**ユーザ**] ページ ([**システム管理**]→[**セキュリティ**]→[**ユーザ**]) に移動して、確認するユーザ・アカウントを選択します。

ユーザ・アカウントの [**一般**] タブには、以下のプロパティが含まれます。その他のタブの詳細は、"ユーザのロールの変更" および "ユーザの SQL 関連オプションの変更" を参照してください。

テーブル 5-1: ユーザ・アカウントのプロパティ

プロパティ名	プロパティの説明
名前	128 文字以内の一意のユーザ識別子。ユーザ・アカウントを作成した後に、名前を変更することはできません。 この識別子には、“@” と “*” を除くすべての文字を使用できます。名前は大文字と小文字が区別されません。すべてのユーザ名に Unicode 文字を使用することができます。ユーザ名に既存のロールと同じ文字列を指定することはできません。
フルネーム	このユーザ・アカウントの表示可能な名前。
コメント	任意のテキスト。
パスワード	新しいパスワードの値。このページを閲覧するユーザの特権に関係なく、この値を見ることはできません。**%Admin_Secure:Use** 特権を持つユーザ、または **%All** ロールに割り当てられているユーザであれば、他のユーザのパスワードを変更できます。この機能は、ユーザがパスワードを忘れた場合や、パスワードを紛失した場合などに使用します。 パスワードでは、大文字と小文字が区別され、Unicode 文字を使用できます。また、[システムセキュリティ設定] ページ ([システム管理]→[セキュリティ]→[システム・セキュリティ]→[システムワイドセキュリティパラメータ]) の [パスワードパターン] フィールドで指定されているパターン (文字のタイプおよび長さ) に適合している必要があります。

プロパティ名	プロパティの説明
パスワード確認	新しいパスワードの値の確認。
次回ログイン時にパスワード変更	次回ユーザがログインしたときに、パスワードの変更を求めるかどうかを指定するチェック・ボックス。
パスワードを有効期限切れにしない	システム規模のパスワードの有効期限をこのユーザに適用するかどうかを指定するチェック・ボックス。チェックを付けると、システムの制限より長い期間パスワードを変更しなくてもユーザのパスワードは期限切れになりません。パスワードの有効期限を設定するには、"システム規模のセキュリティ・パラメータ" のページを参照してください。
ユーザ有効	このアカウントが現在有効であるかどうかを指定するチェック・ボックス。
アカウントを有効期限切れにしない	システム規模のアカウント不活動上限をこのユーザに適用するかどうかを指定するチェック・ボックス。チェックを付けると、システムの制限より長い期間アカウントが不活動でもユーザのアカウントは期限切れになりません。不活動上限を設定するには、"システム規模のセキュリティ・パラメータ" のページを参照してください。
アカウントの有効期限	このアカウントを使用できる最後の日付。
開始ネームスペース	ターミナル・タイプのサービスまたはポータルからログインした後、実行を開始する場所となるネームスペース。このプロパティは、InterSystems IRIS を呼び出すコマンドで指定されたネームスペースの値より優先して適用されます。
Tag^Routine の起動	ターミナル・タイプのサービスからログインした後、自動的に実行されるルーチン。このプロパティは、InterSystems IRIS を呼び出すコマンドで指定されたルーチンの値より優先して適用されます。
メールアドレス	このアカウントに関連付けられているメール・アドレス。
携帯電話サービスプロバイダ	2 要素認証用のユーザの携帯電話サービス・プロバイダ。 ユーザの携帯電話サービス・プロバイダがこのリストに表示されていない場合は、[新規プロバイダ作成] をクリックして新規プロバイダを追加できます。この操作を行うと、新規携帯電話サービス・プロバイダを追加するためのフィールドが表示され、追加したプロバイダが表示されるようになります。
携帯電話番号	2 要素認証用の携帯電話番号。2 つ目の認証トークン (要素) を含むテキスト・メッセージをユーザが受信する携帯電話番号です。
タイプ ([ユーザ] ページにのみ表示されます)	ユーザの種類。これは使用している認証メカニズムとロール割り当てメカニズムによって決まります。値は、 、 、Kerberos 、LDAP 、または OS のいずれかです。ユーザ・タイプの詳細は、次の "ユーザ・タイプについて" を参照してください。

5.1.1 ユーザ・タイプについて

ユーザ・アカウントの Type は、以下のいずれかにできます。

- パスワード・ユーザ ー このユーザは、インスタンス認証、Kerberos 認証 (代行承認を使用していない場合)、またはオペレーティング・システム認証 (代行承認を使用していない場合) により認証されます。ユーザを編集したり何らかの方法で変更したりするための InterSystems IRIS ツールは、パスワード・ユーザに対して使用します。
- 代行ユーザ ー ユーザ定義の認証メカニズムを通じて認証されます。InterSystems IRIS ツールは、このタイプのユーザのプロパティを表示するためにのみ使用できます。ユーザのプロパティは、ツール以外の外的な手段を使って編集する必要があります。

- Kerberos ユーザ ー このユーザは、代行承認を使用している場合に、Kerberos を使用して認証されます。代行承認では、このタイプのユーザのプロパティを表示する目的でのみ InterSystems IRIS ツールを使用できます。このユーザのプロパティは、InterSystems IRIS ツール以外の外部手段を使用して編集し、ZAUTHORIZE ルーチンで指定する必要があります ("代行承認" を参照)。代行承認を使用せずに Kerberos でユーザを認証する場合、ユーザのタイプはパスワード・ユーザです。
- LDAP ユーザ ー LDAP を通じて認証されます。InterSystems IRIS ツールは、このタイプのユーザのプロパティを表示するためにのみ使用できます。ユーザのプロパティは、ツール以外の外的な手段を使って編集する必要があります。
- OS ユーザ ー このユーザは、代行承認を使用している場合に、オペレーティング・システム (OS) を使用して認証されます。代行承認では、このタイプのユーザのプロパティを表示する目的でのみ InterSystems IRIS ツールを使用できます。このユーザのプロパティは、InterSystems IRIS ツール以外の外部手段を使用して編集し、ZAUTHORIZE ルーチンで指定する必要があります ("代行承認" を参照)。代行承認を使用せずにオペレーティング・システムでユーザを認証する場合、ユーザのタイプはパスワード・ユーザです。

重要　1 人のユーザが持つタイプは 1 つのみです。あるタイプのユーザは、別のタイプに関連付けられている認証メカニズムを使用してログインすることはできません。

5.2 ユーザ・アカウントの管理

既存のユーザ・アカウントのリストを確認するには、ポータルで **[ユーザ]** ページ (**[システム管理]→[セキュリティ]→[ユーザ]**) を表示します。このページでは、"ユーザ・アカウントのプロパティ" で詳しく説明している以下のフィールドに情報が表示されます。

- **[ユーザ]** ー このユーザ・アカウントの一意の識別子。
- **[フルネーム]** ー このアカウントの表示可能な名前。
- **[有効]** ー このユーザ・アカウントが現在有効化されているかどうか。
- **[ネームスペース]** (既定のネームスペース) ー ターミナル・タイプの接続で使用される最初のネームスペース。
- **[ルーチン]** (既定のルーチン) ー ターミナル・タイプの接続で実行される最初のルーチン。
- **[タイプ]** ー ユーザ・アカウントの種類。これは使用している認証メカニズムとロール割り当てメカニズムによって決まります。

[ユーザ] ページでは次のアクションを実行できます。

- ユーザ・アカウントの新規作成
- 既存のユーザ・アカウントの編集
- ユーザ・プロファイルの表示
- ユーザ・アカウントの無効化/有効化
- ユーザ・アカウントの削除

5.2.1 ユーザ・アカウントの新規作成

ユーザ・アカウントを新規作成するには、以下の手順に従います。

1. 管理ポータルのホーム・ページで、**[ユーザ]** ページ (**[システム管理]→[セキュリティ]→[ユーザ]**) に移動します。

2. **[ユーザ]** ページで、**[新規ユーザ作成]** を選択します。**[ユーザ編集]** ページの **[一般]** タブが表示され、ここでユーザ・アカウントを作成および構成します。
3. **[ユーザ編集]** ページで、"ユーザ・アカウントのプロパティ" で説明したユーザ・プロパティの値を設定します。

   注釈　類似の特性を持つ複数のアカウントを簡単に作成するには、**[コピー元]** フィールドを使用してプロセスを開始します。**[コピー元]**ドロップダウン・メニューから既存のユーザ・アカウントを選択して、以下のフィールドに、選択したアカウントの値を入力します。

   - [フルネーム]
   - [有効期限]
   - [デフォルト・ネームスペース]
   - [デフォルト Tag^Routine]

4. **[保存]** ボタンをクリックすると、新規ユーザ・アカウントが作成されます。

ユーザ・アカウントを作成した後、その特性を編集できます。

5.2.2 既存のユーザ・アカウントの編集

ユーザ・アカウントを作成した後、その基本プロパティを編集できます。

1. 管理ポータルのホーム・ページで、**[ユーザ]** ページ (**[システム管理]**→**[セキュリティ]**→**[ユーザ]**) に移動します。
2. **[ユーザ]** ページには、ユーザ・アカウントのテーブルがあります。既存のアカウントを編集するには、テーブルからアカウントの名前を選択します。**[ユーザ編集]** ページの **[一般]** タブが表示され、ここでユーザ・アカウントを作成および構成します。
3. **[ユーザ編集]** ページで、"ユーザ・アカウントのプロパティ" で説明したプロパティの値を変更できます。
4. **[保存]** ボタンをクリックすると、ユーザ・アカウントの新しい値が保存されます。

このページにある他のタブでは、ユーザ・アカウントの以下の特性を変更することもできます。

- ロール – そのユーザ・アカウントが現在保持しているロールが一覧表示されます。このページでは、ユーザ・アカウントに新しいロールを与えることもできます。また、ユーザ・アカウントからロールを削除することもできます。
- SQL プロパティ – 以下のプロパティがあります。
  - **[SQL権限]** – ユーザ・アカウントが現在保持しているすべての SQL 特権をネームスペース単位で一覧表示します。このページで SQL 特権を割り当て、または削除することもできます。
  - **[SQLテーブル]** – ユーザ・アカウントに与えられた特権 (%ALTER、DELETE、INSERT、REFERENCES、SELECT、および UPDATE) の対象となっているテーブルがネームスペースごとに一覧表示されます。このページで SQL テーブルの特権を割り当て、または削除することもできます。
  - **[SQLビュー]** – ユーザ・アカウントに与えられた特権 (%ALTER、DELETE、INSERT、REFERENCES、SELECT、および UPDATE) の対象となっているビューがネームスペースごとに一覧表示されます。このページで SQL ビューの特権を割り当て、または削除することもできます。
  - **[SQLプロシージャ]** – ユーザ・アカウントが実行できるストアド・プロシージャがネームスペースごとに一覧表示されます。このページで、プロシージャを実行する権限を割り当て、または削除できます。

注釈　ユーザ・アカウントに対する変更は、ユーザがログアウトし、改めてログインした後で初めて有効になります。

5.2.2.1 ユーザのロールの変更

[**ユーザ編集**] ページの [**ロール**] タブでは、以下の手順でユーザ・アカウントをロールに割り当てたり、ユーザ・アカウントをロールから削除したりできます。

- ユーザ・アカウントをロールに割り当てるには、まずそのロールを [**利用可能**] リストから [**選択済み**] リストに移動します (ロールをダブルクリックするか、ロールを選択して一重の右矢印をクリックします)。次に [**割り当てる**] ボタンをクリックすると、そのユーザ・アカウントがロールに割り当てられます。
- すべてのロールにユーザ・アカウントを割り当てるには、[**利用可能**] リストから [**選択済み**] リストを指している二重矢印をクリックします。次に [**割り当てる**] ボタンをクリックすると、このユーザ・アカウントがすべてのロールに割り当てられます。

  注釈　すべてのロールにユーザ・アカウントを割り当てる場合、事前定義の %SecureBreak ロールも含まれます。これはユーザ・アカウントの機能を限定します (拡張しません)。ユーザ・アカウントを **%SecureBreak** ロールに割り当てると、InterSystems IRIS の保護されたデバッグ・シェルが有効になります。これは、ユーザが発行できるコマンドを制限します。これは他の領域でも予期しない結果をもたらす場合があります。

- ロールからユーザ・アカウントを削除するには、ロール名の右側にある [**削除**] ボタンをクリックします。
- すべてのロールからユーザ・アカウントを削除するには、現在割り当てられているロールが表示されたテーブルの下にある [**すべて削除**] をクリックします (このボタンは、ユーザ・アカウントが複数のロールに割り当てられている場合のみ表示されます)。

5.2.2.2 ユーザの SQL 関連オプションの変更

どのユーザ・アカウントに対しても、以下の SQL 関連の特性を付与または削除できます。

- 一般的な SQL 特権
- テーブルの特権
- ビューに対する特権
- ストアド・プロシージャの特権

一般的な SQL 特権

[**ユーザ編集**] ページの [**SQL権限**] タブで、ユーザ・アカウントの SQL 特権を追加または削除できます。

- ユーザ・アカウントに特権を追加するには、まずその特権を [**利用可能**] リストから [**選択済み**] リストに移動します (特権をダブルクリックするか、特権を選択して一重の右矢印をクリックします)。次に [**割り当てる**] ボタンをクリックすると、その特権がアカウントに与えられます。追加された特権を他のユーザ・アカウントに付与できる特権も追加するには、[**利用可能**] リストの下にある該当のボタンをクリックします。
- ユーザ・アカウントにすべての特権を追加するには、[**利用可能**] リストから [**選択済み**] リストを指している二重矢印をクリックします。次に [**割り当てる**] ボタンをクリックすると、これらの特権がユーザ・アカウントに与えられます。追加された特権を他のユーザ・アカウントに付与できる特権も追加するには、[**利用可能**] リストの下にある該当のボタンをクリックします。
- ユーザ・アカウントから特権を削除するには、特権の名前の右にある [**削除**] リンクをクリックします。
- すべての特権をユーザ・アカウントから削除するには、現在割り当てられている特権が表示されたテーブルの下にある [**すべて削除**] ボタンをクリックします。

利用できる特権は、以下のとおりです。

- %ALTER _TABLE ー 指定されたネームスペースについて、ユーザは ALTER TABLE コマンドを実行できます。
- %ALTER _VIEW ー 指定されたネームスペースについて、ユーザは ALTER VIEW コマンドを実行できます。

- %CREATE_FUNCTION – 指定されたネームスペースについて、ユーザは CREATE FUNCTION コマンドを実行できます。
- %CREATE_METHOD – 指定されたネームスペースについて、ユーザは CREATE METHOD コマンドを実行できます。
- %CREATE_PROCEDURE – 指定されたネームスペースについて、ユーザは CREATE PROCEDURE コマンドを実行できます。
- %CREATE_QUERY – 指定されたネームスペースについて、ユーザは CREATE QUERY コマンドを実行できます。
- %CREATE_TABLE – 指定されたネームスペースについて、ユーザは CREATE TABLE コマンドを実行できます。
- %CREATE_TRIGGER – 指定されたネームスペースについて、ユーザは CREATE TRIGGER コマンドを実行できます。
- %CREATE_VIEW – 指定されたネームスペースについて、ユーザは CREATE VIEW コマンドを実行できます。
- %DROP_FUNCTION – 指定されたネームスペースについて、ユーザは DROP FUNCTION コマンドを実行できます。
- %DROP_METHOD – 指定されたネームスペースについて、ユーザは DROP METHOD コマンドを実行できます。
- %DROP_PROCEDURE – 指定されたネームスペースについて、ユーザは DROP PROCEDURE コマンドを実行できます。
- %DROP_QUERY – 指定されたネームスペースについて、ユーザは DROP QUERY コマンドを実行できます。
- %DROP_TABLE – 指定されたネームスペースについて、ユーザは DROP TABLE コマンドを実行できます。
- %DROP_TRIGGER – 指定されたネームスペースについて、ユーザは DROP TRIGGER コマンドを実行できます。
- %DROP_VIEW – 指定されたネームスペースについて、ユーザは DROP VIEW コマンドを実行できます。

テーブルの特権

[ユーザ編集] ページの **[SQLテーブル]** タブで、ユーザ・アカウントに対してテーブル関連 SQL 特権を追加または削除できます。

1. ページ上部近くにあるドロップダウンから、該当のネームスペースを選択します。そのネームスペースのテーブルのリストが表示されます。
2. テーブルの特権を変更するには、そのテーブルの行にある **[編集]** ボタンをクリックします。特権を変更するためのウィンドウが表示されます。
3. このウィンドウで、以下の項目のチェック・ボックスにチェックを付けるか、チェックを外します。
   - ALTER
   - SELECT
   - INSERT
   - UPDATE
   - DELETE
   - REFERENCES
4. 以上の選択の後、**[適用]** ボタンをクリックすると、テーブルに新しい特権が設定されます。

ビューに対する特権

[ユーザ編集] ページの **[SQLビュー]** タブで、ユーザ・アカウントに対してビュー関連 SQL 特権を追加または削除できます。

ビューの特権を追加する手順は、以下のとおりです。

1. ページ上部近くにあるドロップダウンから、該当のネームスペースを選択します。そのネームスペースのビューのリストが表示されます。
2. ビューの特権を変更するには、そのビューの行にある **[編集]** ボタンをクリックします。特権を変更するためのウィンドウが表示されます。
3. このウィンドウで、以下の項目のチェック・ボックスにチェックを付けるか、チェックを外します。
   - ALTER
   - SELECT
   - INSERT
   - UPDATE
   - DELETE
   - REFERENCES
4. 以上の選択の後、**[適用]** ボタンをクリックすると、テーブルに新しい特権が設定されます。

ストアド・プロシージャの特権

[ユーザ編集] ページの **[SQLプロシージャ]** タブで、ストアド・プロシージャに関連する、ユーザ・アカウントの SQL 特権を追加または削除できます。

ストアド・プロシージャの特権を追加する手順は、以下のとおりです。

1. ページ上部近くにあるドロップダウンから、該当のネームスペースを選択します。そのネームスペースのストアド・プロシージャのリストが表示されます。
2. このウィンドウの下にある **[追加]** ボタンをクリックすると、**[プロシージャ権限を...に付与]** ダイアログが表示されます。
3. このダイアログの上部にあるドロップダウンから、追加するプロシージャを含むスキーマを選択します。ページの左側部分にある **[使用可能]** ウィンドウに、このスキーマのプロシージャのリストが表示されます。
4. 1 つ以上のプロシージャを、**[選択済み]** ウィンドウに移動します。[EXECUTE] チェック・ボックスにチェックが付いていることを確認します。これによって、ストアド・プロシージャを実行する特権を、このユーザ・アカウントが保持できるようになります。
5. 必要に応じ、この特権を他のユーザ・アカウントに与える機能をこのユーザに与えることができます。この処理を実行するには、このページの下部近くにある **[このユーザに Grant 権限を付与する場合はここをチェックします。]** ボックスにチェックを付けます。
6. **[適用]** ボタンをクリックすると、目的の特権がユーザ・アカウントに与えられます。

ユーザ・アカウントのストアド・プロシージャの特権を削除する手順は、以下のとおりです。

1. ページ上部近くにあるドロップダウンから、該当のネームスペースを選択します。そのネームスペースのストアド・プロシージャのリストが表示されます。
2. ストアド・プロシージャの特権を変更するには、そのテーブルの行にある **[編集]** ボタンをクリックします。特権を変更するためのページが表示されます。
3. 表示されたページで、[EXECUTE] チェック・ボックスのチェックを外し、**[このユーザに Grant 権限を付与する場合はここをチェックします。]** チェック・ボックスを目的に応じて設定します。
4. **[適用]** ボタンをクリックすると、ユーザ・アカウントの目的の特権が変更されます。

5.2.3 ユーザ・プロファイルの表示

ユーザ・プロファイルには、ユーザ・アカウントが割り当てられているロールやユーザが最後にログインした日時など、ユーザ・アカウントに関するセキュリティ情報が含まれます。ユーザ・プロファイルを表示する手順は以下のとおりです。

1. 管理ポータルのホーム・ページで、[**ユーザ**] ページ ([**システム管理**]→[**セキュリティ**]→[**ユーザ**]) に移動します。
2. [**ユーザ**] ページで、そのユーザに対応する行の [**プロファイル**] をクリックします。ユーザ・プロファイルが表示されます。

また、ユーザ・アカウントの [**ユーザ編集**] ページが表示されている場合は、ページの左上にある [**プロファイル**] をクリックします。

ユーザ・プロファイルの一部として、以下のプロパティが表示されます。

テーブル 5-2: ユーザ・プロファイルのプロパティ

プロパティ名	プロパティの説明
名前	一意のユーザ識別子。@ を除くあらゆる文字を使用できます。@ はドメインを識別するために使用します。これは [ユーザ編集] ページで編集できます。
フルネーム	このユーザ・アカウントの表示可能な名前。これは [ユーザ編集] ページで編集できます。
ロール	ユーザ・アカウントに割り当てられているロールのコンマ区切りリスト。これらは [ユーザ編集] ページの [ロール] タブで編集できます。
最終パスワード変更	ユーザ・アカウントが最後にパスワードを変更した日付と時刻。
最終ログイン	前回正常にログインした日時。まだ正常にログインしていない場合は 0 です。読み取り専用。
最終ログイン・デバイス	ユーザが前回ログインしたログイン元のホストの IP アドレス。
不正ログイン試行	前回の正常なログイン以降に発生した、不正なログインの試行回数。読み取り専用。
最終不正ログイン	前回行われた不正なログイン試行の日時。読み取り専用。
最終不正ログイン・デバイス	ユーザが前回のログインの試行に失敗したときのログイン元ホストの IP アドレス。
前回のログイン失敗の理由	前回行われた無効なログイン試行で発生したエラー。読み取り専用。
アカウント作成日時	ユーザ・アカウントが作成された日時。読み取り専用。
アカウント作成者のユーザ名	アカウントを作成したユーザに関連付けられたアカウント名。読み取り専用。
アカウントの最終変更日時	前回行われたアカウント変更の日時。読み取り専用。
最終アカウント変更ユーザ名	前回アカウントを変更したユーザに関連付けられたアカウント名。読み取り専用。
アカウントの最終変更情報	アカウントに関して前回変更したプロパティのリスト。読み取り専用。

5.2.4 ユーザ・アカウントの無効化/有効化

ユーザ・アカウントを無効化/有効化できます。例えば、アカウントを無効化して一時的に利用不可にすることができます。こうすると、後でそのアカウントを有効にする際にプロパティやロールなどを再構築する必要がありません。

ユーザ・アカウントを無効化または有効化するには、以下の手順を実行します。

1. 管理ポータルのホーム・ページで、[**ユーザ**] ページ ([**システム管理**]→[**セキュリティ**]→[**ユーザ**]) に移動します。
2. [**ユーザ**] ページで、無効化/有効化するユーザ・アカウントの名前をクリックします。そのユーザの [**ユーザ編集**] ページの [**一般**] タブが表示されます。
3. [**ユーザ編集**] ページで、[**ユーザ有効**] フィールドをクリアします。
4. [**保存**] をクリックすると、新しい状態でユーザ・アカウントが保存されます。

5.2.5 ユーザ・アカウントの削除

ユーザ・アカウントを削除する方法は、以下のとおりです。

1. 管理ポータルのホーム・ページで、[**ユーザ**] ページ ([**システム管理**]→[**セキュリティ**]→[**ユーザ**]) に移動します。
2. [**ユーザ**] ページで、削除するユーザ・アカウントの行にある [**削除**] ボタンを選択します。
3. 操作を確認するダイアログが表示されます。ユーザ・アカウントを削除する場合は [OK] をクリックし、それ以外の場合は [**キャンセル**] をクリックします。

5.3 事前定義のユーザ・アカウント

InterSystems IRIS の各インスタンスには、自動的に以下のアカウントが作成されます。

テーブル 5-3: 事前定義のユーザ・アカウント

ユーザ名	割り当てられたロール	目的
Admin	%Manager	既定の管理者アカウント。このアカウントは、すべてのインターシステムズ製品のすべてのインスタンスに対して存在し、インスタンスの管理をサポートします。プロダクションに移行する前にこのアカウントのパスワードを初期値から変更することをお勧めします。
CSPSystem	(なし)	通常のインスタンスおよびロック・ダウン・インスタンスのためにインスタンス認証を通じて InterSystems IRIS に接続するときの Web ゲートウェイを表す既定のアカウント。プロダクションに移行する前にこのアカウントのパスワードを初期値から変更することをお勧めします。 注釈　インスタンスでこのユーザのパスワードを変更する場合は、Web ゲートウェイでもパスワードを変更する必要があります。
IAM	IAM_API	InterSystems API Manager (IAM) のライセンスを InterSystems IRIS から取得するために必要な既定のアカウント。IAM ユーザを使用するには、IAM ユーザを有効化してパスワードを変更する必要があります。IAM ユーザの設定は、一般的に IAM の設定の一部です。
SuperUser	%All	利用できるすべての特権を持つ既定のアカウント。このアカウントは、すべてのインターシステムズ製品のすべてのインスタンスに対して存在し、製品のすべての側面に完全にアクセスすることが可能になります。プロダクションに移行する前にこのアカウントのパスワードを初期値から変更することをお勧めします。
UnknownUser	%All (最小のセキュリティ)、またはなし (通常のセキュリティ、またはロック・ダウン・セキュリティ)	非ログイン・ユーザの既定のアカウント
_PUBLIC	(なし)	すべてのユーザ (ログイン・アカウント以外) に与えられる特権のセット
_SYSTEM	%All	既定の SQL アカウント。このアカウントは、すべてのインターシステムズ製品のすべてのインスタンスに対して存在し、SQL アクセスを提供します。プロダクション・システムではこのアカウントを無効にすることをお勧めします。
_Ensemble	%All	相互運用性マネージャ (ログイン・アカウントではない)。InterSystems IRIS インスタンスのみ。
HS_Services	%HS_ServiceRole	このアカウントは既定で無効化されており、医療製品で内部用に使用されます。FHIR® ベースの IHE プロファイルを使用する前に、このアカウントを有効にする必要があります。InterSystems IRIS for Health™ および HealthShare® Health Connect にのみ適用されます。

“特権ユーザ・アカウント” というアカウントもあり、これは通常のインストールまたはロック・ダウン・インストールで作成されます。このアカウントに、ユーザ名とパスワードを指定します。

以下のアカウントは削除できません。

- _Ensemble
- _PUBLIC
- _SYSTEM
- UnknownUser

注意 ″ユーザ・アカウントの初期パスワード″ で説明しているように、InterSystems IRIS の新規インストレーションではすべての事前定義アカウントに対して同一のパスワードが使用されます。特に最小のセキュリティによるインストールでは、既定のパスワードはセキュリティ面で脆弱です。この問題を解決するには、そのアカウントを無効にするか、パスワードを変更します。普通はアカウントを無効にすることをお勧めします。

これは特に、コンテナ化されたインスタンスでは重要な問題です。問題を解決する方法を含む詳細は、″認証とパスワード″ を参照してください。

また、%System というユーザ・アカウントがあります。このユーザ・アカウントは、表示されず、InterSystems IRIS で内部的に使用されます。このアカウントに対してログイン、編集、または削除を行うことはできません。このアカウントは **%All** ロールを持って実行されます。%ZSTART や %ZSTOP など特定のルーチンは、このユーザ・アカウントとして実行されます。このようなルーチンを別のユーザとして実行するには、$SYSTEM.Security.Login() を呼び出します。

推奨されていませんが、事前定義のユーザ・アカウントを削除することはできます。ただし、%System の他に、**%All** ロールを持つアカウントが少なくとも 1 つ必要です。

5.3.1 さまざまなアカウントに関するメモ

5.3.1.1 UnknownUser アカウント

特定のアプリケーションやアプリケーションの特定の部分では、承認されないユーザに対しても、InterSystems IRIS を使用する妥当な理由が存在することがあります。例えば、小売システムでユーザが購入手続きを開始する前に、商品の在庫を確認する場合です。このような状況を考慮して、InterSystems IRIS では UnknownUser アカウントがサポートされています。認証されていないユーザが接続すると、特別な名前である UnknownUser が $USERNAME に割り当てられ、このユーザに定義されているロールが $ROLES に割り当てられます。

認証されていないアクセスは、認証に失敗したときに使用されるものではありません。 例えば、ターミナル経由で InterSystems IRIS に接続しようとしているユーザがユーザ名とパスワードを入力したが、認証に失敗したとします。この場合は、認証されていないアクセスが許可されていても、このユーザは InterSystems IRIS に接続されません。一方、認証されていないアクセスが許可されているときに、同じユーザがユーザ名を指定せずに (例えば、ユーザ名のプロンプトで Enter キーを押します) InterSystems IRIS に接続すると、このユーザは認証されていないユーザ UnknownUser として接続されます。同様に、ODBC クライアントがユーザ名とパスワードに NULL 文字列を使用して接続しようとすると、認証されていない接続が該当のサービスに対して許可されていれば、この接続は受け入れられます。同じ ODBC クライアントが、空文字列ではないユーザ名とパスワードを指定すると認証に失敗し、その結果、認証されていないアクセスが許可されていても、このクライアントは接続されません。

5.3.1.2 _PUBLIC アカウント

事前定義のユーザ・アカウント _PUBLIC は、ログインでは使用できない特殊なアカウントです。このアカウントはロールのセットを保持するものです。これらのロールは、システムに接続するすべてのユーザに対して既定のロールとして指定されています。これにより、どのようなユーザにも最低限のロールのセットを与えることができます。例えば、%Operator ロールを _PUBLIC ユーザに関連付けると、$Roles の値には必ず %Operator が含まれます。

5.4 ユーザ・アカウントの検証

アプリケーション・コードでユーザ・アカウントを検証する必要がある場合は、$SYSTEM.Security.Login メソッドの引数を 1 つ使用する形式を使って、このユーザのログインを試行するシンプルなルーチンを作成して実行します。ログインに成功したら、このユーザ・アカウントは正当なユーザです。ログインに失敗した場合は不正です。ログインが成功するかどうかには関係なく、このルーチンが存在する場合、現在のユーザ・アカウントがこのルーチンを呼び出したユーザ・アカウントになります。

以下はこのタスクを実行するサンプル・ルーチン ValidateUser です。

ObjectScript

```
ValidateUser(TestUser) {
    Write "Validating ",TestUser,"...",!
    New $Roles
    Set sc = $SYSTEM.Security.Login(TestUser)
    If sc = 1 {
        Write $Username," is a valid user.",!
        Write $Username," belongs to the following login roles: ",$Roles,!
    } Else {
        Write TestUser," is not a valid user.",!
    }
    Quit sc
}
```

このルーチンは、検証の対象となるユーザ・アカウントの名前を表す文字列を唯一の引数として取ります。その後、以下のアクションを実行します。

1. `New $Roles` を呼び出し、$Roles 変数と $Username 変数の両方をスタックします。$Roles の詳細は、リファレンスの ″$Roles″ を参照してください。
2. その後、引数を 1 つ取る形式の $SYSTEM.Security.Login メソッドを呼び出します。このメソッドはユーザのログインを試行します。ユーザのパスワードは必要ありません。ログインに成功すると、このメソッドは 1 を返します。これにより、ルーチンが表示する情報とルーチンの返り値が決まります。
3. このルーチンが存在する場合は、ログインに成功することで、ユーザが暗黙的にログアウトされます。

重要　このルーチンは、引数を 1 つ取る形式の $SYSTEM.Security.Login メソッドを使用します。引数を 1 つ取る形式の $SYSTEM.Security.Login を正常に呼び出すには、ユーザ・アカウントに **IRISSYS:Write** 特権と **%Service_Login:Use** 特権が必要です。$SYSTEM.Security.Login の詳細は、リファレンスの ″%SYSTEM.Security″ クラスを参照してください。

以下は、ValidateUser の呼び出し方法を示すサンプル・ルーチン VUTest です。これは、ValidUser と NonexistentUser という 2 人のユーザをテストするようにハード・コード化されています。

ObjectScript

```
VUTest() {
    Write $Username," is the current user.",!,!

    Set sc = $$^ValidateUser("ValidUser")
    Write !

    Write "Exited validation code. ",$Username," is the current user.",!,!

    Set sc = $$^ValidateUser("NonexistentUser")
    Write !

    Write "Testing complete.",!
    Write $Username," is the current user."
    Quit 1
}
```

VUTest ルーチンは InterSystems IRIS インスタンスの **User** ネームスペースで作成され、PrivilegedUser アカウントは **%All** ロールのメンバであり、ValidUser のみが存在するとします。ターミナル・プロンプトで VUTest を呼び出した結果は以下のとおりです。

```
Username: PrivilegedUser
Password: ***********
USER>d ^VUTest
PrivilegedUser is the current user.

Validating ValidUser...
ValidUser is a valid user.
ValidUser belongs to the following login roles: %Manager

Exited validation code. PrivilegedUser is the current user.

Validating NonexistentUser...
NonexistentUser is not a valid user.

Testing complete.
PrivilegedUser is the current user.
USER>
```

6
アプリケーション

ほとんどのユーザは、主にアプリケーションによって InterSystems IRIS® データ・プラットフォームとの対話操作を実行します。また、ユーザ・アクセスとユーザ・アクションを制御するための一連の重要なツールは、アプリケーションのセキュリティによって提供されます。アプリケーション・セキュリティでは、インターシステムズの承認ツールを使用して、適切なユーザのみにアプリケーションの使用が許可されます。アプリケーションでは、そのアプリケーションを使用するユーザの特権をエスカレートすることもできます。

ここでは、以下のトピックについて説明します。

- アプリケーション、およびそのプロパティと特権
- アプリケーション・タイプ
- アプリケーションの作成および編集
- 組み込みアプリケーション

6.1 アプリケーション、およびそのプロパティと特権

セキュリティの観点から捉えた場合、アプリケーションは次のように定義されます。

- ユーザによるアクションの実行を可能にするエンティティです。
- 1 つ以上のリソースに関連付けられています。
- その動作を制御するプロパティを持ちます。
- 実行中に実行ユーザの特権を強化することができます。
- プログラムによる特権チェックを内包することができます。

これらの特性および機能はすべてインターシステムズの承認ツールの一部であり、これらの承認ツールが、アプリケーションとそのユーザがインターシステムズ製品および他のセキュリティ・リソースと対話する方法を管理します。アプリケーションには、以下のようないくつかの種類があります。

- Web アプリケーション
- 特権ルーチン・アプリケーション
- クライアント・アプリケーション
- ドキュメント・データベース・アプリケーション

このセクションでは以下について説明します。

- アプリケーションとそのプロパティ
- リソースへのアプリケーションの関連付け
- アプリケーションおよび特権のエスカレーション
- プログラムによる特権チェック

6.1.1 アプリケーションとそのプロパティ

アプリケーションでは、データベースに対する読み書きや他のアセットの使用など、ユーザに許可するアクションのセットを指定できます。この制限を可能にするため、InterSystems IRIS では、アプリケーション定義という機能をサポートしています。この定義は、InterSystems IRIS 内でアプリケーションを表現するために使用される一連の情報です(アプリケーションとアプリケーション定義の関係は、アセットとリソースの関係に似ています)。アプリケーション定義を設定することによって、アプリケーションを制御および管理できます。

重要　アプリケーションとアプリケーション定義は、しばしば同じ意味で使用されます。この 2 つの区別が重要になるのは、実行可能コードまたはそのコードのユーザ・エクスペリエンスが、InterSystems IRIS 内でのそのコードの表現と異なる設定に限られます。この場合、前者 (実行可能コード) はアプリケーションそのものであり、後者 (InterSystems IRIS 内でのそのコードの表現) はアプリケーション定義になります。

各アプリケーションは、それぞれのアプリケーション定義を通して、次のプロパティを持ちます。

名前

アプリケーション名。スラッシュ (“/”) で始まり、その後に英数字、あるいは

/、-、_、.、または % のいずれかの文字が続く必要があります。

アプリケーション定義の名前は、どのリソース名にも依存しません。

説明

アプリケーションの説明。

有効

アプリケーションが使用可能かどうかを示すスイッチ。有効になっていないアプリケーションは、**%All** ロールのメンバも含めて誰も実行できません。このプロパティによって各タイプのアプリケーションがどのように制御されるかについての詳細は、アプリケーションのタイプに応じて ″Web アプリケーション″、″特権ルーチン・アプリケーション″、または ″クライアント・アプリケーション″ を参照してください。

リソース

アプリケーションの動作を管理するリソース。リソースが及ぼす影響は、アプリケーションのタイプに応じて異なります。Web アプリケーションとクライアント・アプリケーションの場合は、このリソースによって、ユーザがアプリケーションにアクセスできるかどうかが制御され、特権ルーチン・アプリケーションの場合は、アプリケーションの特権のエスカレーションが制御されます。Web アプリケーションまたはクライアント・アプリケーションにリソースが割り当てられていない場合、すべてのユーザがそのアプリケーションを実行できます。特権ルーチン・アプリケーションにリソースが割り当てられていない場合、すべてのユーザに対して特権のエスカレーションがそのアプリケーションで実行されます。

各アプリケーション定義は単一のリソースにのみ関連付けることができます。このプロパティによって各タイプのアプリケーションがどのように制御されるかについての詳細は、アプリケーションのタイプに応じて ″Web アプリケーション″、″特権ルーチン・アプリケーション″、または ″クライアント・アプリケーション″ を参照してください。アプリケーションとリソースとの対話の詳細は、″リソースへのアプリケーションの関連付け″ を参照してください。

アプリケーション・ロール

アプリケーションのユーザが割り当てられる 1 つ以上のロール。アプリケーションの実行中、ユーザはそのアプリケーションのアプリケーション・ロールに割り当てられます。この割り当ては、$Roles 変数のロールのリストに、該当するアプリケーション・ロールを付加することで設定されます。アプリケーション・ロールの使用の詳細は、"アプリケーションおよび特権のエスカレーション" を参照してください。

マッチングロール

アプリケーションの実行中に、ユーザが何らかの追加ロール ("ターゲット・ロール") に割り当てられるように作用する 1 つ以上のロール。ユーザがいずれかのマッチングロールに割り当てられている場合、アプリケーションの使用中、そのユーザはターゲット・ロールにも割り当てられます。この割り当ては、$Roles 変数のロールのリストに、該当するロールを付加することで設定されます。例えば、マッチングロールが **%Admin_Manage** の場合、このロールのメンバがこのアプリケーションを使用すると、そのメンバはターゲット・ロール **%Admin_Secure** のメンバにもなります。マッチングロールの使用の詳細は、"アプリケーションおよび特権のエスカレーション" を参照してください。

すべてのアプリケーションはこれらのプロパティを持ちます。その他にも、アプリケーション・タイプのそれぞれが独自の特性を備えています。

6.1.2 リソースへのアプリケーションの関連付け

アプリケーション (および、結果としてそのアプリケーション定義) が 1 つの単体の全体である場合、そのアプリケーションのリソースは 1 つのみとなり、アプリケーションとリソースは一対一の関係になります。複数のアプリケーション (および、結果として複数のアプリケーション定義) が 1 つのリソースに関連付けられるように定義することもできます。この場合、アプリケーションとリソースの関係は多対一になります。アプリケーションの数がいくつでも、状況はこの 2 つの条件を何らかの形で組み合わせたものに帰着します。

アプリケーションが独立した複数の部分で構成される場合は、さらに状況が複雑になります。この各部分は、サブアプリケーションと呼ばれています。サブアプリケーションのグループで成立しているアプリケーションは、単体 (全体で 1 つ) のものとして動作するように設計されますが、それぞれのサブアプリケーションが異なるセキュリティの種類またはレベルを要求できます。このような状況では、サブアプリケーションごとに固有のアプリケーション定義を割り当て、それぞれを別のリソースに関連付けると便利です。この方法によって、各サブアプリケーションに独立したセキュリティ関連の動作を個別に割り当てることができます。アプリケーションの観点では、複数のサブアプリケーションによって 1 つの大きなアプリケーションが構成されることになりますが、インターシステムズのセキュリティの観点では、それぞれに独自のアプリケーション定義を持つ複数の個別アプリケーションが存在し、ユーザは意識することなくそれらのアプリケーション間を通過することになります。この場合も、一対一のケースと多対一のケースに帰着します。ただし、複数のアプリケーション定義は、エンドユーザには 1 つのアプリケーションに見えます。ユーザは既に認証されており、そのプロセスでロールが設定されているため、サブアプリケーション間の受け渡し時に認証は不要です。

例えば、経費レポートのアプリケーションがあるとします。このアプリケーションでは、全従業員が経費レポートを入力できますが、小切手を生成できるのは会計責任者だけです。この場合、アプリケーションは 1 つのアプリケーション全体のように見え、小切手を生成する機能は会計責任者を除くすべての従業員にグレー表示されます。このようなアプリケーションを作成するには、1 つはレポートの入力、もう 1 つは小切手の生成に使用する、2 つの独立したサブアプリケーションが必要になります。1 つ目のサブアプリケーションはすべてのユーザが使用できますが、2 つ目のアプリケーションを使用できるのは会計責任者だけです。2 つのサブアプリケーションは 1 つのアプリケーション内の独立した 2 つの画面にしか見えず、それ以上の外観上の違いはありません。

6.1.3 アプリケーションおよび特権のエスカレーション

ユーザのロールはアプリケーション・リソースを使用することでエスカレートできるため、それらのリソースには動的に推移する承認のニーズを満たすメカニズムが用意されています。アプリケーションの特権のエスカレーションを実行するには、以下の操作を実行します。

1. 既存のアプリケーションに対してリソースを作成し、そのリソースにアプリケーションを関連付けます。

2. そのリソースに対して Use 許可を持つロールを 1 つ以上作成します。
3. アプリケーションの実行に必要な特権のリストを決定します。そのアプリケーションが複数のサブアプリケーションで構成される場合、リストが複数になる場合があります。
4. 各特権リストを個々のロールに関連付けます。各ロールを、そのアプリケーションまたはサブアプリケーションのアプリケーション・ロールとして設定します。
5. そのアプリケーションまたはサブアプリケーションのマッチングロールを設定します。それぞれのマッチングロールには、1 つ以上のターゲット・ロールを関連付けます。
6. ユーザが正常にアプリケーションを起動すると、InterSystems IRIS は次の 2 つのアクションを実行します。
   - アプリケーションの使用中、ユーザをそのアプリケーションのロールに割り当てます (特権ルーチン・アプリケーションの場合、この割り当ては、AddRoles メソッドが正常に起動されたかどうかに依存します。詳細は、"特権ルーチン・アプリケーション" を参照してください)。
   - ユーザがいずれかのマッチングロールに割り当てられている場合、ユーザは、アプリケーションの使用中、アプリケーションによってターゲット・ロールに割り当てられます (これについても、特権ルーチン・アプリケーションの場合、この割り当ては、AddRoles メソッドが正常に起動されたかどうかに依存します。詳細は、"特権ルーチン・アプリケーション" を参照してください)。

例えば、**AppRsrc** という専用のリソースを持つアプリケーションがあるとします。**AppRsrc:Use** 特権を保持している、**AppUser** と **AppOperator** という 2 つのロールがあります。**AppOperator** はマッチングロールでもあり、そのターゲット・ロールは **%Manager** です。このシナリオでは、**AppUser** ロールに属するユーザがこのアプリケーションを呼び出しても、$Roles の値は変化しません。一方で、**AppOperator** ロールに属するユーザがこのアプリケーションを呼び出すと、**%Manager** を含むように $Roles の値が拡張されます。アプリケーションにアプリケーション・ロール **AppExtra** がある場合、**AppUser** ロールに属するユーザがこのアプリケーションを呼び出すと、このユーザは **AppExtra** ロールを受け取ります。マッチングロールのみの最初のシナリオでは、**AppOperator** ロールに属しているユーザで、特権のエスカレーションが発生します。マッチングロールとアプリケーション・ロールがある 2 番目のシナリオでは、どちらのロールに属しているユーザにも特権のエスカレーションが発生します。

6.1.3.1 ユーザ・ベースのセキュリティおよびアプリケーション・ベースのセキュリティ

インターシステムズのセキュリティ・モデルでは、ユーザ・ベース、アプリケーション・ベース、または両方をベースにした柔軟な特権の割り当てが可能です。アプリケーションの使用を特定のユーザのみに限定できるほか、すべてのユーザが使用できるようにすることもできます。アプリケーションの使用を承認されているユーザに対するそのアプリケーションの動作として、次の複数の動作が可能です。

- ユーザの特権のみでアプリケーションを実行できます。
- アプリケーションで一部のユーザの特権をエスカレートすることができます (マッチングロールとターゲット・ロールを使用)。
- アプリケーションですべてのユーザの特権をエスカレートすることができます (アプリケーション・ロールを使用)。
- 一部の特権については、アプリケーションですべてのユーザに対してエスカレーションを発生させ、他の特権については特定のユーザに対してのみエスカレーションを発生させることができます (マッチングロール/ターゲット・ロールとアプリケーション・ロールを組み合わせて使用)。

このように、アプリケーションの使用を、特定のユーザのみに限定して認めるか、すべてのユーザに認めるか制御できます。また、アプリケーションをユーザの特権で実行するか、アプリケーション自体の特権で実行するかも制御できます。これにより、InterSystems IRIS では以下のような極めて柔軟なモデルが実現します。

テーブル 6-1: 安全なアプリケーションの保護とエスカレーションのマトリックス

Privilege Level / Protection Level	Public Application	Restricted Application
With User-Dependent Privileges	1.どのユーザでもアプリケーションを実行できます。アプリケーションはユーザの特権で実行されます。	2.指定されたユーザのみがアプリケーションを実行できます。アプリケーションはユーザの特権で実行されます。
With Privilege Escalation	3.どのユーザでもアプリケーションを実行できます。アプリケーションは、アプリケーション・ロールとマッチングロールを通して、(拡張された) アプリケーション特権で実行されます。	4.指定されたユーザのみがアプリケーションを実行できます。アプリケーションは、アプリケーション・ロールとマッチングロールを通して、(拡張された) アプリケーション特権で実行されます。

上記のテーブルで説明した各シナリオは、以下のようなさまざまな承認モデルで広く使用されています。

1.ユーザ依存の特権を持つパブリック・アプリケーション

このモデルは、認証されたすべてのユーザが使用できるアプリケーションを表します。このアプリケーションは実行しても追加の特権は付与しません。例えば、企業の連絡先データベースの場合、その企業全体のロールに属するユーザであれば、任意の社員の所属先電話番号と電子メール・アドレスを知ることができます。管理職は、社員の自宅電話番号を知ることができる上位の特権を保持しています。人事担当のスタッフは、すべてのレコードを閲覧および更新できる、さらに上位の特権を保持しています。このアプリケーションにはすべての社員がアクセスできますが、その動作は、アプリケーションを呼び出したときに各社員が既に持っている特権で決まります。つまり、アプリケーションそのものからロールが与えられるわけではありません。

2.ユーザ依存の特権を持つ制限アプリケーション

このモデルは、指定のロールに属するユーザのみが使用できるアプリケーションを表します。このアプリケーションは実行しても追加の特権は付与しません。例えば、時給制の社員を扱う給与アプリケーションでは、勤務時間数や時給などが表示されます。このアプリケーションを実行するユーザは、**HourlyEmployee** ロールまたは **HourlyManager** ロールのメンバであることが必要です。このアプリケーションを実行すると、どのロールが指定されたかがチェックされます。このチェックにより、**HourlyEmployee** ロールのメンバは自身のデータを閲覧できますが、それを編集することはできません。一方、**HourlyManager** ロールのメンバは、報告の目的でデータの閲覧と編集が可能です。**HourlyEmployee** ロールのメンバである社員は、このアプリケーションを実行して個人データが正しいかどうかを確認できます。ここで要求されるロールのメンバではない月給制の社員などの他の社員は、このアプリケーションは実行することもできません。

3.特権のエスカレーションを伴うパブリック・アプリケーション

このモデルは、認証されたすべてのユーザが使用できるアプリケーションを表します。このアプリケーションは、実行するとユーザが属するロールに基づいて特権をエスカレートします(特定のロールについてのみ、特権をエスカレートすることもできます)。例えば、学生が自身のレコードを確認および更新できるアプリケーションが、大学にあるとします。ここでは、どの学生も認証されたユーザであり、自身の連絡先情報を編集できます。この機能をサポートするために、アプリケーションにはエントリを編集するためのコードがあります。このコードでは、編集されているエントリが認証ユーザと一致しているかどうかが確認され、一致していれば、レコードを更新できるようにコード自身の特権がエスカレートされます。更新の終了後は、特権が元の状態に戻されます。ある学生が他の学生のレコードを更新しようとしても、このエントリの確認に失敗するので、特権のエスカレートが発生せず、更新も行われません。このアプリケーションでは、ユーザが学籍係の業務ロールのメンバであるかどうかも確認されます。そのメンバである場合は、さらに広範囲に情報を更新できます。

4.特権のエスカレーションを伴う制限アプリケーション

このモデルは、指定のロールに属するユーザのみが使用できるアプリケーションを表します。このアプリケーションは、ユーザが属するロールに基づいて特権をエスカレートします(特定のロールについてのみ、特権をエスカレートすることもできます)。例えば、病院の救急処置室に置かれたアプリケーションでは、現在救急治療を受けている患者のレコードを閲覧できる、特別に幅広い特権を担当の医師に与えることが考えられます。救急処置室では高い緊急性が発生する

可能性があることから、この設定では、単純な巡回診療の場合よりも多くの情報を医師に提示できることが必要です。したがって、この場合は特権のエスカレートが発生します。

6.1.4 プログラムによる特権チェック

特定のアクションを実行するために必要な特権をユーザが保持しているかどうか確認するコードを、アプリケーションに組み込むこともできます。この操作には、$SYSTEM.Security.Check メソッドを使用します。これを呼び出すための構文は次のとおりです。

ObjectScript

```
Set status = $SYSTEM.Security.Check(app_resource, app_permission)
```

以下はその説明です。

- app_resource は、ユーザが許可を保持していなければならないリソースです。
- app_permission は、保持していなければならない許可です。
- status は、メソッドの返り値、True または False (1 または 0) です。

例えば、あるアプリケーションでユーザに **Application_Order_Customer** への Write 許可を付与する必要がある場合、Check 呼び出しは以下のようになります。

ObjectScript

```
Set status = $SYSTEM.Security.Check("Application_Order_Customer", "WRITE")
```

注釈　　$SYSTEM.Security.Check の呼び出しには、特権は必要ありません。

6.2 アプリケーション・タイプ

アプリケーションには、以下のようないくつかのタイプがあります。

- Web アプリケーション
- 特権ルーチン・アプリケーション
- クライアント・アプリケーション
- ドキュメント・データベース・アプリケーション

6.2.1 Web アプリケーション

これらのアプリケーションは、**%Service_WebGateway** サービスを使用して InterSystems IRIS に接続します。

Web アプリケーションでは、セキュリティ情報が Web セッションの一部として保持されます。つまり、$USERNAME および $ROLES の値が複数のページ要求にわたって保持されます (より具体的に言うと、ページに対する処理が開始されたとき、$ROLES にはユーザのロールに加えて、アプリケーションに定義されたロールも記述されます)。その前のページを処理したときに、SET $ROLES または $SYSTEM.Security.AddRoles を介して動的に追加されたロールは含まれません。これは、ステートレスおよび“ステートフル”のどちらのセッションにも当てはまります。

Web アプリケーションを使用すると、クライアント (つまりブラウザ) は通常、接続時にユーザ名とパスワードをサーバに送信することはありません。その代わりにユーザはページを要求し、それを受けたサーバはログイン・ページを送り返します。そのアプリケーションにさらにアクセスするには、ユーザはこのログイン・ページで必要な情報を入力する必要があり

ます。2 要素認証が有効な場合、ユーザがユーザ名とパスワードを入力すると、サーバではセキュリティ・コードを入力するためのページが表示されます。認証に成功すると、ユーザはアプリケーションにアクセスできるようになります。

注釈　2 要素認証では、ユーザ名とパスワードのペアが有効でない場合でも、サーバでは、1 回限りのセキュリティ・トークンを入力するためのページが必ず表示されます。1 回限りのセキュリティ・トークンをユーザが入力すると、サーバではアクセス拒否を示すメッセージが表示され、システムで使用可能な最小限の情報が提供されます。

CSP のセキュリティ処理は以下のように行われます。

1. ページ要求が受信されるたびに、そのアプリケーションは要求の URL によって決まります。そのアプリケーションが有効になっていない場合、接続は行われません。
2. そのアプリケーションが Web セッションで直前に処理されたページのアプリケーションと同じ場合は、既に接続されているため、それ以上のセキュリティ・チェックは要求されません。
3. **%Service_WebGateway** の Use 許可がパブリックではなく、この許可をユーザが保持していない場合、接続は行われません。
4. そのアプリケーションまたは **%Service_WebGateway** が認証を必要とし、ユーザがまだ認証済みでない場合は、要求に IRISUsername および IRISPassword パラメータが含まれているかどうかが InterSystems IRIS によって確認されます。
   a. IRISUsername と IRISPassword が含まれている場合、InterSystems IRIS はログインを試行し、ログインに成功すると、そのアプリケーション・リソースの Use 許可をユーザが保持しているかどうかを確認します。この両方に成功しないと、接続は行われません。
   b. IRISUsername と IRISPassword が含まれていない場合、その Web アプリケーションの構成でアプリケーション固有のログイン・ページが定義されていれば、そのページが表示されます(安全なアプリケーションでは、これがログイン前に使用できる唯一のページです)。アプリケーション固有のログイン・ページがないと、ユーザ名とパスワードによる認証に失敗します。また、ユーザがそのアプリケーション・リソースに対する Use 許可を保持していない場合、接続は行われません。

Web アプリケーションの編集は、以下を参照してください。

- アプリケーションの作成
- Web アプリケーションの編集：[一般] タブ
- アプリケーションの編集：[アプリケーションロール] タブ
- アプリケーションの編集：[マッチングロール] タブ

6.2.1.1 Web アプリケーションでロールをプログラムによってエスカレートする方法の例

以下は、Web アプリケーションのアプリケーション・ロールをエスカレートする方法の例です。この例では、以下のアクションが実行されます。

1. 制御を %SYS ネームスペースに変更します。これは、必要な呼び出しを実行するために行わなければなりません。ロールの管理とエスカレーションを実行するコードを使用できるのは、%SYS ネームスペース内のみだからです。
2. アプリケーションのターゲット・ロールを追加します。
   a. すべてのユーザに **MYAPP** ロールが割り当てられます。これは、行内のコロンの前に、matchroles の値を初期設定するマッチングロールがリストされていないためです。構文は、**matchrole:targetrole** です。**matchrole** の値が空の場合、すべてのユーザに **targetrole** が割り当てられます。
   b. **MYAPPSPECIAL** ロールが既に割り当てられているユーザに、**MYAPP2** ロールを追加します。ここの構文は、**matchrole1:targetrole1,matchrole2:targetrole2** です。つまり、**MYAPPSPECIAL** ロールが割り当てられているユーザには、アプリケーションによって **MYAPP2** ロールが追加されます (2 つ目以降のマッチング

ロールおよびターゲット・ロールの構文の後には、先行コンマが付きます。詳細は、Security.Applications.Create メソッドを参照してください)。

3. ロール・エスカレーション情報を保持しているローカル変数を使用して、アプリケーションの MatchRoles プロパティを更新します。Security.Applications.Modify メソッドは、値が指定されているプロパティのみを更新します。それ以外のプロパティは変更されないままです。
4. 最後に、ロール・エスカレーションが成功した場合に、これを通知します。

このコードは以下のとおりです。

Class Member

```
Method UpdateRoles() As %Status {
 // ******************** modify application roles ********************
 write !, "All users receive the added MYAPP role."
 write !, "Users who have MYAPP2 receive MYAPPSPECIAL also."

 // Change to the %SYS namespace.
 new $NAMESPACE
 set $NAMESPACE="%SYS"

 // Add roles for the application.
 //
 // Add the MYAPP role for all users.
 set matchroles=":MYAPP"
 // Also add MYAPP2 for users who already have the MYAPPSPECIAL role.
 set matchroles=matchroles_",MYAPPSPECIAL:MYAPP2"

 // Use the matchroles variable to
 // set the applications's MatchRoles property.
 set MyAppProps("MatchRoles")=matchroles
 set status=##class(Security.Applications).Modify("/csp/MyApp",.MyAppProps)

  // Announce success.
  if $$$ISOK(status) {
    write !, "Roles were successfully modified."
 }
}
```

6.2.2 特権ルーチン・アプリケーション

特権ルーチン・アプリケーションは、1 つ以上のクラスまたはルーチンにロールをエスカレートする特権を付与して、それらのクラスまたはルーチンのユーザのためにロールをエスカレートします。特権ルーチン・アプリケーションのクラスまたはルーチンは、ObjectScript で作成します。特権ルーチン・アプリケーションを使用するには、以下の操作を実行します。

1. 管理ポータルでアプリケーション定義を作成します。詳細は、"アプリケーションの作成" を参照してください。
2. そのアプリケーション定義にクラスまたはルーチンを追加します。詳細は、"アプリケーションの編集：[ルーチン/クラス] タブ" を参照してください。
3. ロールをエスカレートするように開発環境でアプリケーション定義のクラスまたはルーチンを編集します。詳細は、"特権ルーチン・アプリケーションでのロールのエスカレート：AddRoles メソッド" を参照してください。

ポータルには、特権ルーチン・アプリケーションを編集するための以下のページがあります (これには、前述した最初の 2 つのページが含まれます)。

- 特権ルーチン・アプリケーション、クライアント・アプリケーション、またはドキュメント・データベース・アプリケーションの編集：[一般] タブ
- アプリケーションの編集：[アプリケーションロール] タブ
- アプリケーションの編集：[マッチングロール] タブ
- アプリケーションの編集：[ルーチン/クラス] タブ

6.2.2.1 特権ルーチン・アプリケーションでのロールのエスカレート : AddRoles メソッド

特権ルーチン・アプリケーションでロールをエスカレートするには、%SYSTEM.Security クラスの AddRoles メソッドを呼び出します。AddRoles を呼び出すには、以下の構文を使用します。

ObjectScript

```
Set sc = $SYSTEM.Security.AddRoles("AppDefName")
```

AppDefName はアプリケーション定義の名前で、sc はステータス・コードです。アプリケーション定義に記述されているクラスまたはルーチンがあり、ユーザが適切な特権を持っている場合に、そのクラスまたはルーチンから AddRoles を呼び出すことで、あらゆるアプリケーション・ロール (“アプリケーションの編集 : [アプリケーションロール] タブ” を参照) および関連するあらゆるマッチングロール (“アプリケーションの編集 : [マッチングロール] タブ” を参照) を扱うことができるように特権がエスカレートされます。

重要　エントリ・ポイントでコードを区切る中括弧をルーチンに使用していない場合は、制御がエントリ・ポイント間で受け渡しされることがあります。その結果、ユーザに過度な特権が与えられ、意図しないレベルのアクセスが可能になる恐れがあります。ルーチンの構造化の詳細は、″ユーザ定義コード″ を参照してください。

AddRoles の呼び出しは、以下のように処理されます。

1. 呼び出しが特権クラスまたはルーチン以外からの場合、呼び出しは失敗します。
2. アプリケーション定義で指定された要求リソースがパブリックではないときに、このリソースに対する Use 権限を保持していないユーザがメソッドまたはルーチンを呼び出した場合、呼び出しは失敗します。
3. 上記以外の場合、この呼び出しは成功します。

Tipヒント　特権をエスカレートしたルーチンのスコープ外に制御が渡されたときに、ユーザからすべてのアプリケーション・ロールを剥奪してログイン・ロールに戻すには、AddRoles 呼び出しの前に以下のコマンドを含めます。

ObjectScript

```
New $Roles
```

これらのトピックの詳細は、″プログラムで管理するロール″ を参照してください。

6.2.2.2 特権ルーチン・アプリケーションの使用例

例えば、DB1 という名前のデータベースを使用するアプリケーションがあるとします。このアプリケーションの各ユーザは **%DB_DB1** ロールのみを所持していて、そのすべてのユーザが DB1 に対する特権を所持しています。このアプリケーションの一部のユーザは、一時的に別のデータベース (DB2) へのアクセスも必要とします。こうしたユーザは、PRATestClass クラスの PRAEscalateメソッド (“特権ルーチン・アプリケーション” の “PRA”) によって DB2 へのアクセス権を取得します。このメソッドは、該当するユーザの特権をエスカレートします。具体的には、PRAEscalate によって、**%DB_DB2** ロールを追加して DB2 へのアクセスを可能にします。

PRAEscalate メソッドによって該当するユーザに **%DB_DB2** ロールを追加できるようにするには、以下のセキュリティ項目が存在している必要があります。

- **PRATestResource** という名前のパブリックではないリソース。
- **PRA_DB2** という名前のロール (1 つの特権 **PRATestResource:Use** のみを持つロール)。
- **%DB_DB2** ロール (DB2 データベースの作成時に作成したロール)。
- PRATestApp という名前の特権ルーチン・アプリケーション。PRATestApp に関する説明は以下のとおりです。

- PRATestApp アプリケーションを実行するには、ユーザが **PRATestResource:Use** 特権を持っている必要があります。そのため、DB2 データベースへのアクセスを必要とするユーザは、**PRA_DB2** ロール (**PRATestResource:Use** 特権を付与するロール) を持っている必要があります。
- PRATestClass クラスは、PRATestApp アプリケーションに含まれます(このクラスをアプリケーションに含めるには、PRATestApp の [**編集**] ページにある [**ルーチン/クラス**] タブを使用します)。
- **%DB_DB2** ロールは、PRATestApp のアプリケーション・ロールです(アプリケーション・ロールを指定するには、PRATestApp の [**編集**] ページにある [**アプリケーションロール**] タブを使用します)。

この設定を実行したときに、PRATestBasicUser および PRATestDB2User というユーザが存在すると仮定すると、以下のようになります。

- PRATestBasicUser は、**%DB_DB1** 専用のユーザです。そのため、PRATestApp アプリケーションは、PRATestBasicUser のロールをエスカレートすることはありません。また、このユーザは、DB2 へのアクセスを必要とするアプリケーションの部分を使用できません。
- PRATestDB2User は、**%DB_DB1** ロールと **PRA_DB2** ロールのメンバです。そのため、PRATestApp アプリケーションは PRATestBasicUser のロールをエスカレートします。このユーザは、DB2 へのアクセスを必要とするアプリケーションの部分を使用できます。

PRAEscalate のコードは以下のとおりです。

Class Member

```
Method PRAEscalate()
 {
   Write "This method is a part of the privileged routine application ",!
   Write "called PRATestApp.",!
   Write "The user invoking this routine is ",$Username,!
   Write "The current value of $Roles is ",$Roles,!,!
   Write "Calling the AddRoles method...",!,!
   New $Roles
   Set sc = $SYSTEM.Security.AddRoles("PRATestApp")
   If sc = 1
   {
      Write "Application roles have been added.",!
      Write "$Roles now is ",$Roles,!,!
   } Else {
      Write "The call to AddRoles has failed.",!
      Do $system.Status.DecomposeStatus(sc,.Err)
      Write Err(Err),!
   }
 }
```

このルーチンを PRATestDB2User が実行した場合のターミナル・セッションは以下のとおりです。

```
Username: PRATestDB2User
Password: ********
USER>set x = ##class(PRATestClass).PRATest()
This method is a part of the privileged routine application
called PRATestApp.
The user invoking this routine is PRATestDB2User
The current value of $Roles is %DB_DB1, PRA_DB2

Calling the AddRoles method...

Application roles have been added.
The current value of $Roles is %DB_DB1, %DB_DB2, PRA_DB2
Removing %DB_DB2 from $Roles...
$Roles now is %DB_DB1, PRA_DB2

USER>
```

このルーチンを PRATestBasicUser が実行した場合のターミナル・セッションは以下のとおりです。

```
Username: PRATestBasicUser
Password: ********
USER>set x = ##class(PRATestClass).PRATestMethod()
This method is a part of the privileged routine application
called PRATestApp.
The user invoking this routine is PRATestUser
The current value of $Roles is %DB_DB1

Calling the AddRoles method...

The call to AddRoles has failed.
ERROR #862: User is restricted from running privileged application PRATestApp
-- cannot execute.

USER>
```

6.2.3 クライアント・アプリケーション

これは、InterSystems IRIS への接続にクライアント アプリケーション タイプを使用するアプリケーションです。

重要　　クライアント・アプリケーションに関して、以下の点に注意してください。

- Windows でのみサポートされます。このため、これらのアプリケーション用の管理ポータルのオプションは、Windows でのみ使用できます。
- アプリケーションの認証を設定するには、"認証" で説明されているツールを使用します。

クライアント・アプリケーションを編集するには、ポータルの以下のページを使用します。

- 特権ルーチン・アプリケーション、クライアント・アプリケーション、またはドキュメント・データベース・アプリケーションの編集：[一般] タブ
- アプリケーションの編集：[アプリケーションロール] タブ
- アプリケーションの編集：[マッチングロール] タブ

6.2.4 ドキュメント・データベース・アプリケーション

これらのアプリケーションは、ドキュメント・データベースを使用して InterSystems IRIS に接続します。

重要　　このようなアプリケーションでは、"認証" で説明されている認証ツールを使用してください。

ドキュメント・データベース・アプリケーションを編集するには、ポータルの以下のページを使用します。

- 特権ルーチン・アプリケーション、クライアント・アプリケーション、またはドキュメント・データベース・アプリケーションの編集：[一般] タブ
- アプリケーションの編集：[アプリケーションロール] タブ
- アプリケーションの編集：[マッチングロール] タブ

6.3 アプリケーションの作成および編集

ここでは、以下のトピックについて説明します。

- アプリケーションの作成

- Web アプリケーションの編集：[一般] タブ
- 特権ルーチン・アプリケーション、クライアント・アプリケーション、またはドキュメント・データベース・アプリケーションの編集：[一般] タブ
- アプリケーションの編集：[アプリケーションロール] タブ
- アプリケーションの編集：[マッチングロール] タブ
- アプリケーションの編集：[ルーチン/クラス] タブ
- Web アプリケーションの相互運用対応ネームスペース用の設定

6.3.1 アプリケーションの作成

アプリケーションを作成する手順は以下のとおりです。

1. 管理ポータルのメニューで、**[システム管理]→[セキュリティ]→[アプリケーション]** を選択します。これにより、さまざまなアプリケーション・タイプが表示されます。
2. **[Web アプリケーション]**、**[特権ルーチンアプリケーション]**、**[クライアントアプリケーション]**、または **[ドキュメント DB アプリケーション]** を選択します。選択したアプリケーション・タイプのページが表示されます。
3. アプリケーション・ページの左上隅で、新しいアプリケーションの作成ボタンをクリックします。選択した種類のアプリケーション編集ページが表示されます。以下の情報を使用して、既存のアプリケーションと同様に、アプリケーションを編集できます。
   - Web アプリケーションの編集：[一般] タブ
   - 特権ルーチン・アプリケーション、クライアント・アプリケーション、またはドキュメント・データベース・アプリケーションの編集：[一般] タブ

6.3.2 Web アプリケーションの編集：[一般] タブ

Web アプリケーションを編集する手順は以下のとおりです。

1. 管理ポータル・メニューで、**[システム管理]→[セキュリティ]→[アプリケーション]→[ウェブ・アプリケーション]** を選択します。

   構成済みの Web アプリケーションが一覧表示されます。**[タイプ]** 列には、アプリケーションがユーザ・アプリケーション (`CSP`) またはシステム・アプリケーション (`CSP,System`) として示されます。
2. アプリケーションを選択して **[編集]** をクリックした後、情報を入力または変更します。
3. 編集が完了したら、新規設定を有効にするために InterSystems IRIS を再起動します。

6.3.2.1 一般設定

[一般] タブの最初のセクションには、さまざまなオプションが表示されます。

注釈　ここでは、InterSystems IRIS REST ベースの Web アプリケーションに関連するフィールドについてのみ説明します。CSP/ZEN アプリケーションと呼ばれる他のタイプの Web アプリケーションは、従来のインターシステムズ・アプリケーション・タイプです。InterSystems IRIS をベースにした新しいアプリケーションは REST アプリケーションであるため、ここでは、CSP/ZEN アプリケーションにのみ関連するフィールドについては説明しません (別のインターシステムズ製品から CSP/ZEN アプリケーションを InterSystems IRIS に移行した場合は、その関連フィールドについてのドキュメントが用意されていますので参照してください)。

名前

アプリケーションの識別子。名前の先頭にはスラッシュ (/) を含める必要があります (例えば、/myorg/myapp アプリケーションのようにします)。

名前 /csp/docbook は予約されていることに注意してください。

説明

アプリケーションの説明テキスト。

ネームスペース

このアプリケーションが実行されるネームスペース。別のネームスペースを選択すると、そのネームスペースの既定アプリケーションがこのドロップダウン・メニューの右側に即座に表示されます。

ネームスペースの既定アプリケーション

アプリケーションがこのネームスペースの既定アプリケーションかどうかの指定。%System.CSP.GetDefaultApp メソッドは、ネームスペースの既定アプリケーションを返します。$system.OBJ.Load や $system.OBJ.ImportDir などの InterSystems IRIS のインポート関数は、関連付けられているアプリケーションなしでページをインポートする際に、この既定アプリケーションを使用します。

アプリケーション有効

アプリケーションを使用できるかどうかの指定。有効になっていれば、認証および承認されたユーザはアプリケーションを使用できます。無効の場合は使用できません。

[REST] または [CSP/ZEN] の有効化

これが REST アプリケーションであるか、CSP/ZEN アプリケーション (レガシ・インターシステムズ・アプリケーションのタイプ) であるかの指定。新しいアプリケーションの場合、サードパーティの最新のフロントエンド・テクノロジをサポートする [REST] を使用することをお勧めします。

CSP を使用する既存のアプリケーションがある場合、インターシステムズは、そのテクノロジを使用した継続的な開発をサポートしており、CSP 設定のドキュメントをアプリケーションの作成および編集：[一般] タブに用意しています。

ディスパッチ・クラス

REST サービスを実装するための %CSP.REST の対応するカスタム・サブクラス。詳細は、"手動による REST サービスの作成" を参照してください。

6.3.2.2 セキュリティの設定

セキュリティの設定を以下に示します。

必要なリソース

ユーザがアプリケーションを実行するために Use 許可を保持していなければならないリソース。これらのリソースおよび許可の詳細は、"リソースについて" を参照してください。

ID でグループ化

使用しません。このフィールドは、従来のアプリケーションを移行した場合にのみ使用します。詳細は、ドキュメントを参照してください。

許可された認証方法

アプリケーションでサポートされている認証メカニズム。ここで使用可能なオプションは、**[認証オプション]** ページ (**[システム管理]** > **[セキュリティ]** > **[システム・セキュリティ]** > **[認証/Web セッション・オプション]**) での選択内容によって異なります。アプリケーションが複数の認証メカニズムをサポートする場合、認証は以下のように行われます。

- **[認証なし]** を含む複数のオプションが有効になっている場合、ユーザはユーザ名とパスワードを入力せずにログインできます。
- 複数のオプションが有効になっている状態で、かつユーザ名とパスワードを入力した場合、InterSystems IRIS によってカスケード認証が試行されます。
- 選択したオプションが Kerberos 認証やインスタンス認証 (パスワード) であっても、**[認証なし]** を選択していない場合、ユーザ名とパスワードを入力する必要があります。InterSystems IRIS では、最初に Kerberos を使用して認証が実行され、次にインスタンス認証が実行されます。いずれかが成功すれば、ユーザは認証されます。両方の認証が失敗した場合、アクセスは拒否されます。

詳細は、"認証" を参照してください。

許可したクラス

使用しません。このフィールドは、従来のアプリケーションを移行した場合にのみ使用します。詳細は、ドキュメントを参照してください。

6.3.2.3 セッションの設定

このセクションのこの設定を使用して、Web アプリケーションのセッション・プロパティを管理できます。

重要　これらの設定を使用するには、まずアプリケーションのディスパッチ・クラスの UseSession パラメータをゼロ以外の値に設定する必要があります。そうしないと、これらの設定の値を変更しても効果はありません。詳細は、"手動による REST サービスの作成" を参照してください。

セッションの設定を以下に示します。

セッション・タイムアウト

既定のセッション・タイムアウトを秒単位で指定します。この値は、%CSP.Session オブジェクトの AppTimeout プロパティを使用してオーバーライドできます。

1 つのセッションでその有効期間中に Web アプリケーションを変更した場合、新しいアプリケーションでは、既定のタイムアウト値を使用してセッションのタイムアウト値が更新されません。例えば、既定のタイムアウト値が 900 秒の Web アプリケーション A でセッションが開始し、その後、既定のタイムアウト値が 1800 秒の Web アプリケーション B に移動した場合、セッションは 900 秒後にタイムアウトになります。

アプリケーションを変更するときにセッションのタイムアウト値が更新されるようにするには、セッション・イベント・クラスで OnApplicationChange コールバック・メソッドをオーバーライドし、%session オブジェクトの AppTimeout プロパティを更新するためのコードを追加します。

インターシステムズ管理ポータルの [Interoperability] ページで自動ログアウトを無効にした場合、これらのページにセッション・タイムアウトは適用されません。すなわち、ページはタイムアウトしません。自動ログアウトを無効にすることはお勧めしません。詳細は、"管理ポータルの自動ログアウト動作" を参照してください。

イベントクラス

クラス (%CSP.SessionEvents のサブクラス) の既定名。この既定名のメソッドは、タイムアウトやセッションの終了などの Web アプリケーション・イベントを呼び出します。この値を上書きするには、拡張子 (.cls など) を除いたクラス名を値として使用して、%CSP.Session オブジェクトの EventClass プロパティの値を指定します。

セッションにクッキーを使用する

アプリケーションでブラウザ・セッションを追跡するために cookie を使用するか、URL 書き換え手段 (各 URL に値を挿入する) を使用するかの設定。選択肢は以下のとおりです。

- **常時** – 既定。常に cookie を使用します。
- **なし** – cookie を使用しません。
- **自動検出** – クライアント・ブラウザで無効になっている場合を除き、cookie を使用します。ユーザが cookie を無効にしている場合、アプリケーションでは、URL 書き換えが使用されます。

このオプションは、アプリケーションが cookie を使用するどうかを設定するものではなく、ユーザの設定に従ってアプリケーションがどのようにセッションを管理するかを制御するものです。さらに、値が **[常時]** または **[自動検出]** であっても、アプリケーションは、コードが cookie を使用するように記述されている場合にのみ cookie を使用します。

セッションクッキーパス

このアプリケーションに関するセッション Cookie をブラウザから InterSystems IRIS に返送する際に使用する URL の一部。このフィールドの値を指定しない場合、アプリケーションは、**[名前]** フィールドの値の先頭と末尾にスラッシュを付けたものを既定のスコープとして使用します。したがって、ここで値を指定しない場合、myapp という名前のアプリケーションのスコープは /myapp/ になります。

アプリケーションは、指定されたスコープ内にあるページの cookie のみを送信します。スコープを 1 つの Web アプリケーションで必要なページに制限すると、このマシン上の他の Web アプリケーションがこのセッション cookie を使用するのを防ぐことができます。また、この Web サーバ上の他の Web アプリケーションが cookie を参照するのも防ぐことができます。

1 つのセッション cookie を同時に共有しながら、プライマリ・アプリケーションとそのサブアプリケーションとで異なるセキュリティ設定を使用できます (すべてのアプリケーションがプライマリ・アプリケーションのパスを使用している場合)。

Session Cookie Scope

Web アプリケーションに関連付けられているセッション Cookie の SameSite 属性の既定値を制御します。詳細は、以下の“SameSite 属性について”を参照してください。

User Cookie Scope

%CSP.Response.SetCookie を使用して作成したユーザ定義 Cookie の SameSite 属性の既定値を制御します。詳細は、以下の“SameSite 属性について”を参照してください。

SameSite 属性について

SameSite 属性では、サードパーティ・アプリケーションに関連する Cookie をアプリケーションでどのように処理するかを指定します (クロスサイト・リクエストともいいます)。

[Session Cookie Scope] フィールドおよび [User Cookie Scope] フィールドを使用して、アプリケーションの Cookie の SameSite を設定できます。[Session Cookie Scope] では、セッション、ログイン、CSRF、およびグループ ID の各 Cookie について SameSite 属性の値を設定し、[User Cookie Scope] では、ユーザ定義 Cookie について SameSite 属性の値を設定します。

SameSite には、以下の値を設定できます。

- [None] – アプリケーションはクロスサイト・リクエストに応じて Cookie を送信します。SameSite の値が [None] の場合、ブラウザにより、アプリケーションで HTTPS 接続を使用するよう要求されることがあります。
- [Lax] – アプリケーションは、安全な最上位のクロスサイト・ナビゲーションを使用して Cookie を送信します。

- [Strict] – アプリケーションはクロスサイト・リクエストに応じて Cookie を送信することはありません (システム Web アプリケーション、および新規またはアップグレードしたユーザ・アプリケーションの既定値)。

アプリケーションの Cookie をよりきめ細かく制御するには、%CSP.Response.SetCookie メソッドを使用します。これは、特定の Cookie の SameSite の既定値を上書きします。%CSP.Response.SetCookie を使用して、Cookie の **SameSite** の値を **[None]** に指定する場合は、HTTPS 接続を使用する必要があります。

SameSite 属性は IETF の計画の一環であり、IETF のいくつものドキュメントで扱われています。

6.3.3 特権ルーチン・アプリケーション、クライアント・アプリケーション、またはドキュメント・データベース・アプリケーションの編集 : [一般] タブ

以下はその方法です。

1. 管理ポータルのメニューで、**[システム管理]→[セキュリティ]→[アプリケーション]** を選択します。これにより、さまざまなアプリケーション・タイプが表示されます。
2. **[Web アプリケーション]**、**[特権ルーチンアプリケーション]**、**[クライアントアプリケーション]**、または **[ドキュメント DB アプリケーション]** を選択します。選択したアプリケーション・タイプのページが表示されます。
3. アプリケーションのページで、アプリケーションの名前をクリックして編集用に選択します。そのアプリケーションの **[編集]** ページが表示されます。
4. 既定では、**[一般]** タブが表示されます。特権ルーチン・アプリケーションとクライアント・アプリケーションの場合、このページには以下のフィールドが表示されます。

[名前] フィールド (アプリケーション・タイプによって異なります)

アプリケーションの識別子。

説明

アプリケーションの説明テキスト。

有効

アプリケーションが有効かどうかの指定。有効になっていれば、認証および承認されたユーザはアプリケーションを使用できます。無効の場合は使用できません。

アプリケーションの実行に必要なリソース

ユーザが特定のアクションを実行するために Use 許可 (ロールで特権の一部として有効化されたもの) を保持していなければならないリソース。Web アプリケーションおよびクライアント・アプリケーションの場合、このリソースはアプリケーションの単純な操作に必要です。特権ルーチン・アプリケーションの場合、このリソースは、アプリケーションでのロールのエスカレーションを可能にする AddRoles メソッドの呼び出し時に必要になります。

6.3.4 アプリケーションの編集 : [アプリケーションロール] タブ

Web アプリケーション、特権ルーチン・アプリケーション、またはクライアント・アプリケーションについては、すべてのユーザが特定のロールを取得するようにアプリケーションを構成できます。こうしたロールは、アプリケーション・ロールと呼ばれます。

アプリケーションのアプリケーション・ロールを指定する手順は以下のとおりです。

1. 管理ポータルのメニューで、**[システム管理]→[セキュリティ]→[アプリケーション]** を選択します。これにより、さまざまなアプリケーション・タイプが表示されます。

2. [**ウェブ・アプリケーション**]、[**特権ルーチンアプリケーション**]、または [**クライアントアプリケーション**] を選択します。選択したアプリケーション・タイプのページが表示されます。
3. アプリケーションのページで、アプリケーションの名前をクリックして編集用に選択します。そのアプリケーションの [**編集**] ページが表示されます。
4. [**編集**] ページで、[**アプリケーション・ロール**] タブを選択します。
5. 1 つ以上のアプリケーション・ロールを指定するには、[**使用可能**] リストに含まれているロールをクリックします。ロールを [**選択済み**] リストに移動します。
6. [**割り当てる**] をクリックして、アプリケーション・ロールを設定します。

6.3.5 アプリケーションの編集 : [マッチングロール] タブ

Web アプリケーション、特権ルーチン・アプリケーション、またはクライアント・アプリケーションについては、マッチングロールおよびターゲット・ロールをサポートするようにアプリケーションを構成できます。いずれかのマッチングロールに割り当てられているユーザがそのアプリケーションを実行すると、そのユーザは InterSystems IRIS によって関連するターゲット・ロールに割り当てられます。1 つのアプリケーションに複数のマッチングロールを含めることができます。マッチングロールごとに、複数のターゲット・ロールを含めることができます。また、複数のマッチングロールに同一のターゲット・ロールを含めることができます。

アプリケーションでマッチングロールとそのターゲット・ロールを設定する手順は以下のとおりです。

1. 管理ポータルのメニューで、[**システム管理**]→[**セキュリティ**]→[**アプリケーション**] を選択します。これにより、さまざまなアプリケーション・タイプが表示されます。
2. [**ウェブ・アプリケーション**]、[**特権ルーチンアプリケーション**]、または [**クライアントアプリケーション**] を選択します。選択したアプリケーション・タイプのページが表示されます。
3. アプリケーションのページで、アプリケーションの名前をクリックして編集用に選択します。そのアプリケーションの [**編集**] ページが表示されます。
4. [**編集**] ページで、[**マッチングロール**] タブを選択します。
5. [**マッチングロール**] タブで、[**マッチングロールの選択**] ドロップ・ダウンからマッチングロールとするロールを選択します。
6. 付随するターゲット・ロールを選択するには、[**使用可能**] リストに含まれているロールをクリックします。ロールを [**選択済み**] リストに移動します。
7. [**割り当てる**] をクリックして、マッチングロールとそのターゲット・ロールを設定します。

6.3.6 アプリケーションの編集 : [ルーチン/クラス] タブ

このタブは特権ルーチン・アプリケーションでのみ使用できます。このタブでは、特権ルーチン・アプリケーションに組み込むクラスまたはルーチンを指定できます。

特権ルーチン・アプリケーションにクラスまたはルーチンを追加する手順は以下のとおりです。

1. 管理ポータルのメニューで、[**特権ルーチンアプリケーション**] ページ ([**システム管理**] > [**セキュリティ**] > [**アプリケーション**] > [**特権ルーチンアプリケーション**]) に移動します。
2. [**特権ルーチン・アプリケーション**] ページには編集可能なアプリケーションのリストがあります。該当するアプリケーションの [**名前**] をクリックします。アプリケーションの [**特権ルーチンアプリケーション編集**] ページが表示されます。
3. [**特権ルーチンアプリケーション編集**] ページで、[**ルーチン/クラス**] タブを選択します。
4. [**ルーチン/クラス名**] フィールドに、アプリケーションに追加するルーチンまたはクラスの名前を入力します。

5. [ルーチン] と [クラス] のうちの追加する方を、該当するチェック・ボックスを選択して指定します。
6. [割り当てる] をクリックして、アプリケーションにルーチンまたはクラスを追加します。

6.3.7 Web アプリケーションの相互運用対応ネームスペース用の設定

InterSystems IRIS では、特定のインスタンス内で相互運用対応ネームスペースごとに異なる Web アプリケーションを使用できます。したがって、異なるユーザ・セットを有効にして、同じ InterSystems IRIS インスタンス内で異なる相互運用対応ネームスペースにアクセスすることができます。

Web アプリケーションを既存の相互運用対応ネームスペース用に設定する手順は以下のとおりです。

1. ネームスペースの最初の Web アプリケーションのコピーである Web アプリケーションを作成します。

   ネームスペースを作成すると、最初の Web アプリケーションが `/csp/namespace` という名前で作成されます。namespace はネームスペースの名前です。

   手順については、"アプリケーションの作成" を参照してください。[コピー元] フィールドを使用して、コピーするアプリケーションを指定できます。

2. `^%SYS("Ensemble","InstalledNamespace","namespace")` グローバル・ノードを、作成した Web アプリケーションの名前に設定します。namespace の値は、新しい Web アプリケーションを使用するネームスペースの名前です。

   例えば、`/csp/ensdemocopy` という名前の Web アプリケーションを作成し、この Web アプリケーションを ENSDEMO ネームスペースに使用したい場合は、ターミナルで以下のコマンドを実行します。

   ```
   set ^%SYS("Ensemble","InstalledNamespace","ENSDEMO")="/csp/ensdemocopy"
   ```

   ユーザがネームスペースの相互運用性のページに移動すると、新しい Web アプリケーションが表示されます。

6.4 組み込みアプリケーション

各 InterSystems IRIS インスタンスには、いくつかの組み込みアプリケーションが付属しています。これにはシステム・アプリケーションのグループが含まれ、**%Service_WebGateway** サービスが無効になっていても、これらのアプリケーションには常にアクセスできます。

テーブル 6-2: InterSystems IRIS の組み込み Web アプリケーション

名前	目的または管理された対話処理	関連付けられたリソース	システム・アプリケーション
/api/atelier	Atelier REST API (%Api.Atelier ディスパッチ・クラス)。	**%Development**	いいえ
/api/deepsee	DeepSee REST API (%Api.DeepSee ディスパッチ・クラス)。		いいえ
/api/docdb	DocDB REST API (%Api.DocDB ディスパッチ・クラス)。		いいえ

名前	目的または管理された対話処理	関連付けられたリソース	システム・アプリケーション
/api/iam	InterSystems API Manager (IAM) REST API (%Api.IAM ディスパッチクラス)。InterSystems IRIS から IAM ライセンスを取得する際に使用します。	**%IAM**	いいえ
/api/iknow	iKnow REST API (%Api.iKnow ディスパッチ・クラス)。		いいえ
/api/mgmnt	API 管理 REST API (%Api.Mgmnt ディスパッチ・クラス)。		いいえ
/api/monitor	Monitoring REST API (%Api.Monitor ディスパッチ・クラス)		いいえ
/api/uima	UIMA REST API (%Api.UIMA ディスパッチ・クラス)。		いいえ
/csp/broker	共通の静的ファイル・ストア。インターシステムズ内部での使用専用。		はい
/csp/documatic	インターシステムズのクラス・リファレンス・ドキュメント。	**%Development**	はい
/csp/sys	ポータルへの一般アクセス。		はい
/csp/sys/exp	ポータルのデータ管理オプション。	**%Development**	はい
/csp/sys/mgr	ポータルの構成およびライセンス・オプション。	**%Admin_Manage**	はい
/csp/sys/op	ポータルの操作オプション。	**%Admin_Operate**	はい
/csp/sys/sec	ポータルのセキュリティ管理および暗号化オプション。	**%Admin_Secure**	はい
/csp/user	USER ネームスペースの既定のアプリケーション。		いいえ
/isc/pki	インターシステムズ公開鍵基盤 (PKI)。		はい
/isc/studio/rules	CSP ルール・ファイルへのマッピング。		はい
/isc/studio/templates	システム定義のスタジオ・テンプレート・ファイルへのマッピング。	**%Development**	はい
/isc/studio/usertemplates	ユーザ定義のスタジオ・テンプレート・ファイルへのマッピング。		いいえ

7

代行承認の使用法

InterSystems IRIS® データ・プラットフォームでは、ユーザ定義の承認コードの使用がサポートされています。このメカニズムを、代行承認といいます。

- 代行承認の概要
- 代行 (ユーザ定義) 承認コードの作成
- 代行承認を使用するためのインスタンスの構成
- 承認後 – システムの状態

7.1 代行承認の概要

管理者は、代行承認により、インターシステムズのセキュリティの一部であるロール割り当ての動作に代わるカスタムのメカニズムを導入できます。例えば、ユーザ定義の承認コードが、外部データベースのユーザのロールを検索し、InterSystems IRIS にその情報を提供することができます。

代行承認を使用するには、以下の手順を実行します。

1. ZAUTHORIZE ルーチンで、代行 (ユーザ定義) 承認コードを作成します。
2. InterSystems IRIS インスタンスに対し、代行承認を使用するようインスタンスを構成します。

注釈　代行承認は、Kerberos およびオペレーティング・システム・ベースの承認でのみサポートされます。

代行認証と代行承認間の相互作用

ZAUTHORIZE.mac による代行承認は、代行認証と組み合わせて使用できません。代行認証のルーチン (″代行認証の使用法″ で説明している ZAUTHENTICATE) は、承認をサポートします。ZAUTHENTICATE を使用する場合、認証と承認のコードを分離することもできます。

重要　HealthShare® で認証を使用している場合は、インターシステムズが提供する ZAUTHENTICATE ルーチンを使用する必要があります。独自のルーチンは作成できません。

7.2 代行 (ユーザ定義) 承認コードの作成

代行承認コードの作成に関連するトピックは、以下のとおりです。

7.2.1 ZAUTHORIZE.mac テンプレートからの開始

インターシステムズが提供するサンプル・ルーチン ZAUTHORIZE.mac をコピーして変更することができます。このルーチンは GitHub の Samples-Security サンプルに含まれています (https://github.com/intersystems/Samples-Security)。"InterSystems IRIS で使用するサンプルのダウンロード" で説明されているようにサンプル全体をダウンロードすることもできますが、単に GitHub でファイルを開いて、その内容をコピーする方が簡単です。

独自の ZAUTHORIZE.mac を作成するには、以下の手順を実行します。

1. ZAUTHORIZE.mac をテンプレートとして使用するには、その内容を %SYS ネームスペースの ZAUTHORIZE.mac ルーチンにコピーして保存します。
2. ZAUTHORIZE.mac サンプル内のコメントを確認します。そこには、カスタム版のルーチンを実装する方法に関する重要なガイダンスが記載されています。
3. ユーザ・アカウントの特性を設定するには、カスタム承認コードと必要なコードを追加してルーチンを編集します。

注意　InterSystems IRIS では ZAUTHORIZE の承認コードに対する制約を行わないため、アプリケーション・プログラマは、コードが十分に安全であるかどうかを確認する必要があります。

代行承認コードのアップグレード

新しいバージョンの InterSystems IRIS にアップグレードする前に、ZAUTHORIZE.mac を確認し、現在の承認コードで新機能をサポートするための変更が必要かどうかを判断してください。

7.2.2 ZAUTHORIZE シグニチャ

代行承認用に構成されると、システムは承認が発生した後、自動的に ZAUTHORIZE を呼び出します。InterSystems IRIS は、必要に応じて ZAUTHORIZE シグニチャで定義されたパラメータの値を提供します。ZAUTHORIZE のシグニチャは以下のとおりです。

ObjectScript

```
ZAUTHORIZE(ServiceName, Namespace, Username, Password,
        Credentials, Properties) PUBLIC {

  // authorization code
  // optional code to specify user account properties and roles
}
```

以下はその説明です。

- ServiceName ー ユーザから InterSystems IRIS への接続を仲介しているサービスの名前を指定する文字列 (**%Service_Console** や **%Service_WebGateway** など)。
- Namespace ー 目的とする接続先の InterSystems IRIS サーバにあるネームスペースを指定する文字列。これは、スタジオや ODBC の接続など、**%Service_Bindings** のみで使用されます。その他のサービスの場合、この値は **""** (引用符で囲んだ空の文字列) とする必要があります。
- Username ー 特権が決定されるユーザを指定する文字列。

- Password ー 使用しているアカウントに関連付けられたパスワードを指定する文字列。これは、Kerberos K5API 認証メカニズムのみで使用されます。その他のメカニズムの場合、この値は "" (引用符で囲んだ空の文字列) とする必要があります。
- Credentials ー 参照渡し。今回のバージョンの InterSystems IRIS では、実装されていません。
- Properties ー 参照渡し。Username で指定したアカウントの特性を指定する返り値の配列。詳細は、"ZAUTHORIZE とユーザ・プロパティ" を参照してください。

7.2.3 ZAUTHORIZE による承認コード

ZAUTHORIZE の目的は、認証されたユーザのロールと他の特性を確立または更新することです。承認コードのコンテンツはアプリケーションに固有です。これには、ユーザが記述した ObjectScript コード、クラス・メソッド、または $ZF コールアウトを含めることができます。

ZAUTHORIZE は、Properties 配列の値を設定することによりロール情報を指定します。この配列は、ZAUTHORIZE に参照渡しされます。通常、設定する値のソースは、ZAUTHORIZE が使用できるユーザ情報のリポジトリです。

注意　InterSystems IRIS では ZAUTHORIZE の承認コードに対する制約を行わないため、アプリケーション・プログラマは、コードが十分に安全であるかどうかを確認する必要があります。

7.2.3.1 ZAUTHORIZE とユーザ・プロパティ

Properties 配列の要素は、Username パラメータで指定されたユーザに関連付けられた属性の値を指定します。通常、ZAUTHORIZE 内のコードにより、この要素の値が設定されます。Properties 配列の要素は以下のとおりです。

- Properties("Comment") ー 任意のテキスト。
- Properties("FullName") ー ユーザの名前と姓。
- Properties("NameSpace") ー ターミナル・ログインの既定のネームスペース。
- Properties("Roles") ー ユーザが InterSystems IRIS で保持するコンマ区切りのロール・リスト。
- Properties("Routine") ー ターミナル・ログインに対して実行されるルーチン。"" の値は、ターミナルがプログラマ・モードで実行されることを示します。
- Properties("Password") ー ユーザのパスワード。
- Properties("Username") ー ユーザのユーザ名。

各要素については、この後のセクションでそれぞれ詳しく説明します。

注釈　承認後に Properties 配列のメンバの値を操作することはできません。

Comment

ZAUTHORIZE が Properties("Comment") の値を返すと、その文字列が InterSystems IRIS におけるユーザ・アカウントの Comment プロパティの値になります (このプロパティは、"ユーザ・アカウントのプロパティ" で説明しています)。呼び出し元のルーチンに値が渡されない場合、ユーザ・アカウントの Comment の値は NULL 文字列になり、管理ポータルの関連フィールドにはコンテンツが表示されません。

FullName

ZAUTHORIZE が Properties("FullName") の値を返すと、その文字列が InterSystems IRIS におけるユーザ・アカウントの Full name プロパティの値になります (このプロパティは、"ユーザ・アカウントのプロパティ" で説明しています)。呼び出し元のルーチンに値が渡されない場合、ユーザ・アカウントの Full name の値は NULL 文字列になり、管理ポータルの関連フィールドにはコンテンツが表示されません。

Namespace

ZAUTHORIZE で Properties("Namespace") の値を設定すると、その文字列が InterSystems IRIS におけるユーザ・アカウントの Startup Namespace プロパティの値になります (このプロパティは、"ユーザ・アカウントのプロパティ" で説明しています)。呼び出し元のルーチンに値が渡されない場合、ユーザ・アカウントの Startup Namespace の値は NULL 文字列になり、管理ポータルの関連フィールドにはコンテンツが表示されません。

InterSystems IRIS に接続すると、Startup Namespace の値 (Properties("Namespace") の値で指定されたもの) は、ローカル・アクセス (コンソール、ターミナル、Telnet など) に認証されたユーザの最初のネームスペースを決定します。Startup Namespace に値が指定されない場合、ローカル・アクセスに認証されたユーザの最初のネームスペースは以下のように決定されます。

1. USER ネームスペースが存在する場合、これが最初のネームスペースになります。
2. USER ネームスペースが存在しない場合、最初のネームスペースは %SYS ネームスペースになります。

注釈　ユーザが最初のネームスペースに対する適切な特権を保持していない場合、アクセスは拒否されます。

Password

ZAUTHORIZE で Properties("Password") の値を設定すると、その文字列が InterSystems IRIS におけるユーザ・アカウントの Password プロパティの値になります (このプロパティは、"ユーザ・アカウントのプロパティ" で説明しています)。呼び出し元のルーチンに値が渡されない場合、ユーザ・アカウントの Password の値は NULL 文字列になり、管理ポータルの関連フィールドにはコンテンツが表示されません。

ZAUTHORIZE がパスワードを返す場合、パスワード認証が有効であれば、システムにログインできます。これは、代行認証からパスワード認証への移行を支援するために使用可能なメカニズムですが、複数の認証メカニズムを使用する点に関して通常と同じように注意が必要です。詳細は、"カスケード認証" を参照してください。

Roles

ZAUTHORIZE で Properties("Roles") の値を設定すると、その文字列によってユーザの割り当て先の Roles が指定されます。この値はコンマ区切りのロール・リストを含む文字列です。呼び出し元のルーチンに値が渡されない場合、ユーザ・アカウントに関連付けられたロールはありません。これは、管理ポータルに示されます。ユーザのロールに関する情報は、**[ユーザ編集]** ページの **[ロール]** タブおよびユーザのプロファイルで確認できます。

Properties("Roles") で返されるロールが定義されない場合、ユーザはロールに割り当てられません。

したがって、ログインしたユーザは以下のようにロールに割り当てられます。

- ロールが Properties("Roles") に示され、InterSystems IRIS インスタンスで定義される場合、ユーザはそのロールに割り当てられます。
- ロールが Properties("Roles") に示され、InterSystems IRIS インスタンスで定義されない場合、ユーザはそのロールに割り当てられません。
- ユーザは、_PUBLIC ユーザに関連付けられたロールには常に割り当てられます。また、すべてのパブリック・リソースにアクセスできます。_PUBLIC ユーザの詳細は、"_PUBLIC アカウント" を参照してください。パブリック・リソースの詳細は、"サービスとそのリソース" を参照してください。

Routine

ZAUTHORIZE で Properties("Routine") の値を設定すると、その文字列が InterSystems IRIS におけるユーザ・アカウントの Startup Tag^Routine プロパティの値になります (このプロパティは、"ユーザ・アカウントのプロパティ" で説明しています)。呼び出し元のルーチンに値が渡されない場合、ユーザ・アカウントの Startup Tag^Routine の値は NULL 文字列になり、管理ポータルの関連フィールドにはコンテンツが表示されません。

Properties("Routine") に値がある場合、この値によって、ターミナル・タイプのサービス (コンソール、ターミナル、Telnet など) でログイン後に自動的に実行するルーチンが指定されます。Properties("Routine") に値がない、または値が "" の場合、ログインは、プログラマ・モードへのアクセス権の有無に従って、プログラマ・モードでターミナル・セッションを開始します。

Username

ZAUTHORIZE で Properties("Username") の値を設定すると、その文字列が InterSystems IRIS におけるユーザ・アカウントの Name プロパティの値になります（このプロパティは、"ユーザ・アカウントのプロパティ" で説明しています）。これにより、アプリケーション・プログラマは、ログイン・プロンプトでエンドユーザが入力した内容を正規化できます（正規化されたユーザ名は大文字と小文字のみ異なります）。

Properties("Username") の値を呼び出し元のルーチンに渡す明示的な呼び出しがない場合、正規化は行われず、プロンプトでエンドユーザが入力する値が、変更されずに、そのユーザ・アカウントの Name プロパティの値としてそのまま使用されます。

7.2.3.2 ユーザ情報のリポジトリ

ZAUTHORIZE は、グローバルや外部ファイルなど、ユーザ情報のリポジトリであればどのような種類でも参照できます。認証されたユーザをこの情報で作成または更新できるように、ルーチンのコードによって Properties 配列で外部プロパティを設定します。例えば、あるリポジトリにロールやネームスペースなどの情報が含まれる一方で、ZAUTHORIZE コードは InterSystems IRIS がその情報を利用できるようにする必要があります。

リポジトリ内の情報が変更されると、このアクションを実行するコードが ZAUTHORIZE にある場合、この情報は InterSystems IRIS ユーザ情報に伝播されます。また、このようなコードがある場合、リポジトリでユーザのロールが変更されなければなりません。セッション中にユーザのロールを変更した場合、その変更は次のログインまで有効になりません。次のログインの時点で、そのユーザのロールは ZAUTHORIZE によってリセットされます。

7.2.4 ZAUTHORIZE の返り値とエラー・メッセージ

ルーチンは以下のいずれかの値を返します。

- 成功 ー $SYSTEM.Status.OK()。これは、ZAUTHORIZE が正常に実行されたことを示します。ルーチン内のコードにより、Username および Password に関連付けられたユーザ認証の成功、Username に関連付けられたユーザ承認の成功、またはその両方を示します。
- 失敗 ー $SYSTEM.Status.Error($$$ERRORMESSAGE)。承認が失敗したことを示します。ZAUTHORIZE がエラー・メッセージを返すと、LoginFailure イベント監査が有効であれば、このメッセージは監査ログに記録されます。エンドユーザには、$SYSTEM.Status.Error($$$AccessDenied) エラー・メッセージのみが表示されます。

ZAUTHORIZE は、システム定義またはアプリケーション固有のエラー・メッセージを返すことができます。これらのメッセージはすべて、**%SYSTEM.Status** クラスの Error メソッドを使用します。このメソッドは、$SYSTEM.Status.Error として呼び出され、エラーの状況に応じて、1 つまたは 2 つの引数を取ります。

使用可能なシステム定義のエラー・メッセージは以下のとおりです。

- $SYSTEM.Status.Error($$$AccessDenied) ー“アクセスが拒否されました”のエラー・メッセージ
- $SYSTEM.Status.Error($$$InvalidUsernameOrPassword) ー“ユーザ名またはパスワードが無効です”のエラー・メッセージ
- $SYSTEM.Status.Error($$$UserNotAuthorizedOnSystem,Username) ー“ユーザ username は許可されていません”のエラー・メッセージ
- $SYSTEM.Status.Error($$$UserAccountIsDisabled,Username) ー“ユーザ username アカウントが無効です”のエラー・メッセージ
- $SYSTEM.Status.Error($$$UserInvalidUsernameOrPassword,Username) ー“ユーザ username の名前またはパスワードは無効です”のエラー・メッセージ
- $SYSTEM.Status.Error($$$UserLoginTimeout) ー“ログインタイムアウト”のエラー・メッセージ
- $SYSTEM.Status.Error($$$UserCTRLC) ー“ログインは中止されました”のエラー・メッセージ

- $SYSTEM.Status.Error($$$UserDoesNotExist,Username) ー“ユーザ username は存在しません”のエラー・メッセージ
- $SYSTEM.Status.Error($$$UserInvalid,Username) ー“ユーザ名 username が無効です”のエラー・メッセージ
- $SYSTEM.Status.Error($$$PasswordChangeRequired) ー“パスワードの変更が必要です”のエラー・メッセージ
- $SYSTEM.Status.Error($$$UserAccountIsExpired,Username) ー“ユーザ username のアカウントは失効しました”のエラー・メッセージ
- $SYSTEM.Status.Error($$$UserAccountIsInactive,Username) ー“ユーザ username のアカウントはアクティブではありません”のエラー・メッセージ
- $SYSTEM.Status.Error($$$UserInvalidPassword) ー“無効なパスワードです”のエラー・メッセージ
- $SYSTEM.Status.Error($$$ServiceDisabled,ServiceName) ー“サービス username のログインは無効です”のエラー・メッセージ
- $SYSTEM.Status.Error($$$ServiceLoginsDisabled) ー“ログインは無効です”のエラー・メッセージ
- $SYSTEM.Status.Error($$$ServiceNotAuthorized,ServiceName) ー“ユーザはサービスを許可されていません”のエラー・メッセージ

これらのエラー・コードを使用するには、ZAUTHORIZE.mac にある `#Include %occErrors` 文をアンコメントします。

カスタム・メッセージを生成するには、$SYSTEM.Status.Error() メソッドを使用して、このメソッドに `$$$GeneralError` マクロを渡し、2 番目の引数として任意のカスタム・テキストを指定します。例えば以下のようになります。

```
$SYSTEM.Status.Error($$$GeneralError,"Any text here")
```

エラー・メッセージが呼び出し元に返されると、そのメッセージは監査データベース (LoginFailure イベント監査が有効な場合) にログとして記録されます。表示されるエラー・メッセージは $SYSTEM.Status.Error($$$AccessDenied) のみです。ただし、`$$$PasswordChangeRequired` エラーのメッセージも表示されます。現在のパスワードから新しいパスワードへの変更をユーザに要求する場合、このエラーを返します。

7.3 代行承認を使用するためのインスタンスの構成

カスタマイズされた ZAUTHORIZE ルーチンを作成したら、次に、インスタンスの関連サービスまたはアプリケーションでそのルーチンを有効にします。手順は以下のとおりです。

1. ターミナルまたはコンソール・ウィンドウの %SYS ネームスペースから、^SECURITY ルーチンを実行します。
2. ^SECURITY で **[システムパラメータの設定]** を選択し、その下で **[認証オプション編集]** を選択し、さらにその下で **[Kerberos認証を許可]** または **[オペレーティングシステム認証を許可]** を選択します (代行承認は、これら 2 つの認証メカニズムに対してのみサポートされています)。
3. **[オペレーティングシステム認証を許可]** を選択した場合、**[O/S 認証に代行承認を許可]** を選択します。**[Kerberos認証を許可]** を選択した場合、**[Kerberos に代行承認を許可]** を選択します。

これらのいずれかを選択すると、InterSystems IRIS は %SYS ネームスペースで ZAUTHORIZE.mac ルーチンを起動します (このルーチンが存在する場合)。

重要　　InterSystems IRIS が ZAUTHORIZE を呼び出すのは、ユーザ認証後のみです。

7.3.1 代行承認とユーザ・タイプ

代行承認を使用する認証メカニズムでユーザが最初に InterSystems IRIS にログインすると、InterSystems IRIS は OS (オペレーティング・システム) または Kerberos のタイプのユーザ・アカウントを作成します(この値は、**[ユーザ]** ページ (**[システム管理]** > **[セキュリティ]** > **[ユーザ]**) にあるユーザのテーブルの [タイプ] 列に表示されません。)ZAUTHORIZE ルーチンは、アカウント作成の時点で、その後のログインのため、そのユーザのロールを指定します。

代行承認を使用せずにログインを試みると、ログインは失敗します。これは、代行承認のみがユーザ・タイプを OS または Kerberos として指定するためです。代行承認なしでこれらの認証メカニズムを使用する場合、ユーザはパスワード・ユーザ・タイプとして認証されます。ユーザが持つことのできるタイプは 1 つのみで、あるタイプのユーザは別のタイプに関連付けられたメカニズムを使用してログインすることができないため、ログインは失敗します (代行認証および LDAP 認証も同じ理由で失敗します)。

ユーザ・タイプの一般情報は、"ユーザ・タイプについて" を参照してください。

7.4 承認後 – システムの状態

ユーザが正常に承認されると、InterSystems IRIS セキュリティ・データベースは次のいずれかの方法で更新されます。

1. そのユーザの最初のログインである場合、ZAUTHORIZE によって返されたプロパティを使用して、入力されたユーザ名に対するユーザ・レコードがセキュリティ・データベースに作成されます。
2. そのユーザが以前にログインしたことがある場合、この関数によって返されるプロパティを使用して、セキュリティ・データベースのユーザ・レコードが更新されます。

初めてのユーザかどうかに関係なく、ログインするプロセスは $ROLES システム変数の値を Properties("Roles") の値に設定します。ターミナル・ログインの場合、ネームスペースは Properties("NameSpace") の値に設定され、起動ルーチンは Properties("Routine") の値に設定されます。