認証ガイド

Version 2023.1
2024-01-02

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

認証ガイド
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

図一覧

テーブル一覧

1

認証メカニズムの概要

1.1 認証について

認証は、InterSystems IRIS® に接続しようとしているあらゆるユーザやその他のエンティティの身元を確認するプロセスです。よく言われるように、認証は、自分が自身で主張するとおりの人物であることを証明する方法です。

認証されたユーザは、InterSystems IRIS との接続を確立し、そのデータとツールを使用できるようになります。信頼できる認証がないと、あるユーザが他人になりすまして不正に入手した特権を利用できるため、承認が無意味なものになります。

1.2 認証メカニズム

ユーザを認証するための方法がいくつかあり、それぞれを認証メカニズムといいます。InterSystems IRIS では、以下のようなさまざまな認証メカニズムがサポートされています。

- Kerberos ー Kerberos プロトコルは、安全でないネットワーク上でサービスに対する安全な認証を提供することを目的として設計されました。Kerberos では、チケットを使用してユーザが認証され、ネットワーク上でのパスワードの交換が回避されます。
- オペレーティング・システム・ベース ー OS ベースの認証では、オペレーティング・システムでユーザごとに割り当てられている身元情報を使用して、InterSystems IRIS 向けにユーザを識別します。
- インスタンス認証 ー インスタンス認証では、ユーザにパスワードを要求し、ユーザが入力したパスワードのハッシュ値を格納値と比較します。
- Lightweight Directory Access Protocol (LDAP) ー LDAP により、InterSystems IRIS では LDAP サーバとして知られる一元管理リポジトリにある情報に基づいてユーザを認証します。
- 代行認証 ー 代行認証により、カスタマイズされた認証メカニズムを作成する方法が実現します。アプリケーション開発者は、代行認証コードのコンテンツを完全に制御します。

認証を一切実行せずに、すべてのユーザが InterSystems IRIS に接続できるようにすることも可能です。これを、認証なしアクセスといいます。認証なしアクセス・オプションは、外部との境界が強力に保護されている組織や、アプリケーションとデータの両方が攻撃の対象としてまったく興味を引かない場合に適用できます。

通常、認証なしアクセスを許可するようにインターシステムズの製品やサービスを構成する場合は、認証なしアクセスのみを使用することをお勧めします。認証メカニズムと、認証失敗時に適用する認証なしアクセスの両方をサポートする場合、この手法をカスケード認証といいます。詳細は、"カスケード認証" を参照してください。複数の認証メカニズムを使

用する状況は、"複数の認証メカニズムの使用" を参照してください。通常、InterSystems IRIS はこれら認証メカニズムのうち 1 つのみを使用するように構成されます。

1.3 認証の仕組み

認証メカニズムは接続ツールで使用されます。これらのツールは、ユーザが InterSystems IRIS との接続を確立するための手段を指定します。それぞれの接続ツール (ターミナル、Java、Web など) はインターシステムズのサービスを使用しますが、このサービスは、サポートされる認証メカニズムを指定するために管理者が使用できるものです (インターシステムズのサービスは、InterSystems IRIS への接続を許可または拒否する機能を持ちます。サービスの詳細は、"サービス" を参照してください)。

接続ツールは 3 種類に分類でき、それぞれをアクセス・モードといいます。アクセス・モードごとに、独自の特性と独自のサポート対象のサービスがあります。アクセス・モードには以下のものがあります。

- ローカル ー ユーザは、InterSystems IRIS の実行可能プログラムが実行されているマシン上で、その実行可能プログラムを直接操作します。
- クライアント・サーバ ー ユーザは、InterSystems IRIS に接続する独立した実行可能プログラムを操作します。
- Web ー ユーザは Web ブラウザを使用して、Web ベースのアプリケーションを通じて InterSystems IRIS を操作します。

エンドユーザは接続ツールを使用し、特定の認証メカニズムによって特定のアクセス・モードで InterSystems IRIS を操作します。この章で説明しているプロセスそのものでは、認証されたアクセスは確立されません。これらのプロセスで確立されるのは、特定のアクセス・モードで特定の機構を通じてユーザを認証するときにアプリケーションで使用されるインフラストラクチャです。

1.3.1 さまざまなアクセス・モードについて

1.3.1.1 ローカル・アクセス・モード

ローカル・アクセスでは、InterSystems IRIS サーバと同じマシン上にエンドユーザが存在します。ユーザがデータにアクセスするには、共有メモリとの間で読み取りと書き込みを実行する InterSystems IRIS のプライベート・イメージを実行します。複数のローカル・ユーザが存在する場合、それぞれのユーザは InterSystems IRIS の実行可能プログラムの個人用コピーを使用し、これらすべての実行可能プログラムは同じ共有メモリを参照します。ユーザと実行可能プログラムが同じマシン上に存在することから、両者の間の通信を保護したり、暗号化したりする必要がありません。これは、この実行可能プログラムと他の実行可能プログラムとの間で情報が受け渡されることがないからです。ユーザと InterSystems IRIS との間の通信が単独のプロセスの範囲で処理されるので、この認証をプロセス内認証ともいいます。

ローカル・アクセスは以下の場合に利用できます。

- ターミナル ー Windows の場合は **%Service_Console**、その他のオペレーティング・システムの場合は **%Service_Terminal**
- コールイン ー **%Service_CallIn**

1.3.1.2 クライアント・サーバ・アクセス モード

クライアント・サーバ・アクセスでは、InterSystems IRIS の実行可能プログラムはサーバであり、そのサーバから独立したマシンに、クライアント側の実行可能プログラムを置くことができます。InterSystems IRIS は、多くの場合はネットワーク経由でこのクライアントとの接続を受け入れます。この接続では、InterSystems IRIS でサポートされているものであれば、どのような言語またはプロトコルも使用できます。これには、以下のものがあります。

- ComPort ― **%Service_ComPort**
- Java ― **%Service_Bindings**
- JDBC ― **%Service_Bindings**
- ODBC ― **%Service_Bindings**
- Telnet ― **%Service_Telnet**

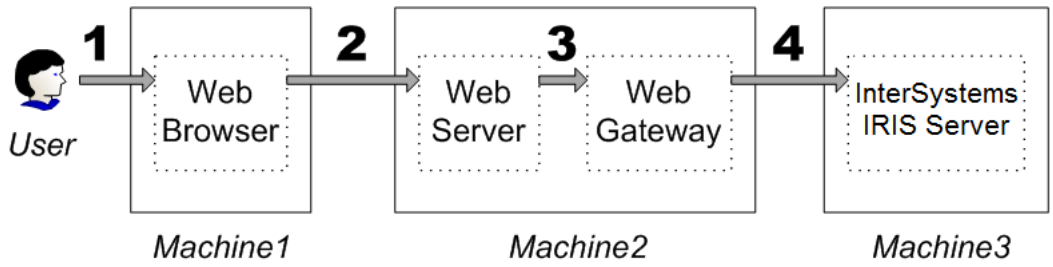
インスタンス認証を介した認証のみをサポートする **%Service_ComPort** を除くすべての接続ツールは、Kerberos またはインスタンス認証をサポートします。

どの場合でも、サポートされる認証タイプはサーバで指定されています。クライアントがサーバに対するアクセスを開始するときは、これらのサポートされている認証タイプのいずれかを使用する必要があります。他のタイプを使用すると、接続が拒否されます。接続ツールによっては、一部の認証タイプを使用できないことがあります。

1.3.1.3 Web アクセス・モード

Web アクセス・モードでは、以下の形式の接続がサポートされています。

図 1-1: Web 接続のアーキテクチャ



1. Web ブラウザで、ユーザがコンテンツまたはアクションを要求します。
2. ユーザの要求が Web ブラウザから Web サーバに渡されます。
3. Web ゲートウェイが Web サーバと同じ場所に置かれていて、ユーザの要求が Web ゲートウェイに渡されます。
4. ユーザの要求が Web ゲートウェイから InterSystems IRIS サーバに渡されます。

ユーザに関連したコンテンツが InterSystems IRIS サーバから提供されたとき、またはユーザに関連したアクションが InterSystems IRIS サーバで実行されたとき、上記と逆方向で同じプロセスが発生します。

InterSystems IRIS に認証されるユーザにとって、ユーザ名とパスワードは完全な形で渡される必要があります。このことから、このアクセス・モードはプロキシ・モードまたはプロキシ接続とも呼ばれます。InterSystems IRIS を実行しているマシンに情報が到達すれば、ユーザとサーバとの関係はローカル・アクセス・モードの場合と似たものになります。実際、Web アクセス・モードでは、プロセス内認証も使用されます。

1.4 認証の設定の概要

1. 認証メカニズムを選択します。認証の要件とアクセス・モードに基づいて選択できます。
2. 以下の手順に従って、認証を構成します。
   - Kerberos 認証
   - オペレーティング・システム・ベースの認証

- インスタンス認証
- LDAP 認証
- 代行認証

3. オプションで 2 要素認証を実装します。

使用する認証メカニズムを InterSystems IRIS インスタンスごとに 1 つのみとすることと、InterSystems IRIS をインストールする前にインスタンスの認証メカニズムを選択しておくことをお勧めします。インストールを開始すると、選択した認証メカニズムを使用するように InterSystems IRIS を構成する作業を開始できます。この作業では以下の手順が必要です。

- Kerberos 認証の場合は、すべての InterSystems IRIS ユーザが、Kerberos の KDC (Key Distribution Center) または Windows のドメイン・コントローラに記録されていることを確認します。
- オペレーティング・システム・ベースの認証の場合は、すべての InterSystems IRIS ユーザがオペレーティング・システムのリストに記録されていることを確認します。
- すべての認証メカニズムについて、選択した認証メカニズムのみが使用されるように、サポート対象のすべてのサービスを構成します。
- すべての認証メカニズムについて、サポートされていないすべてのサービスを無効にします。
- すべての認証メカニズムについて、選択した認証メカニズムのみが使用されるように、すべてのアプリケーションを構成します。

注釈　選択した認証メカニズムに関係なく、起動するときとシャットダウンするときは、必ずオペレーティング・システムの認証が使用されます。

2

Kerberos 認証

2.1 Kerberos 認証について

2.1.1 Kerberos の背景

最も安全な接続を実現するために、InterSystems IRIS では Kerberos 認証システムをサポートしています。この認証システムは、ユーザの身元を確認するための極めて安全で効果的な手段を提供します。Kerberos は、安全性の低いネットワーク上で認証を実現するために、マサチューセッツ工科大学 (MIT) で開発されたもので、巧妙な攻撃から通信を保護します。この保護システムの最大の特長は、ユーザのパスワードが、暗号化されたものであってもネットワーク上には送出されない点にあります。

Kerberos は、信頼されるサードパーティ・システムと呼ばれるものです。パスワードなどの機密性の高い認証情報はすべて Kerberos サーバに保持され、Kerberos サーバそのものは物理的に安全な場所に設置されます。

Kerberos には以下のような特長もあります。

- 長年にわたる実績 ー Kerberos が初めて開発されたのは 1980 年代後半です。その中心となるアーキテクチャと設計は、数多くのサイトで長年使用されています。また、長年の運用で発見された問題は、継続的な改訂で解決されてきました。
- サポート対象のすべての InterSystems IRIS プラットフォームで使用可能 ーもともと UNIX® 向けに開発された Kerberos は、InterSystems IRIS がサポートするすべての UNIX® 系 OS で使用できます。また、Microsoft 社は Windows 2000 以降の Windows に Kerberos を採用しています(Microsoft .NET フレームワークでは Kerberos を直接サポートしていないので、InterSystems IRIS Managed Provider for .NET でも Kerberos はサポートしていません)。
- 柔軟な設定が可能 ー 異種ネットワークに対応できます。
- 高い拡張性 ー Kerberos プロトコルを使用しているので、鍵配布センター (KDC) との対話処理が最小限で済みます。これにより、大規模なシステム上でこのような対話処理がボトルネックになることを防止できます。
- 高速 ー オープン・ソース製品として、Kerberos は長年にわたって徹底的に検討され、最適化されてきました。

Kerberos 認証の基盤となっているのは、AES 暗号化アルゴリズムです。AES (Advanced Encryption Standard) は、一般公開の下で作成されている、ロイヤルティ不要の対称ブロック暗号化方法で、128 ビット、192 ビット、および 256 ビットのキー・サイズをサポートしています。United States National Institute of Standards and Technology (NIST) により採択されるなど、US Federal Information Processing Standard (FIPS) の一部となっています。

Kerberos の背景は、MIT Kerberos の Web サイトおよび Kerberos に関する入手可能な資料のリストを参照してください。

2.1.2 Kerberos の仕組み

Kerberos モデルには、いくつかのアクターがあります。Kerberos で認証されるさまざまなプログラムと人物を総称して、プリンシパルといいます。Kerberos のシステムは、Kerberos Key Distribution Center (KDC) で管理されます。Windows では、Windows ドメイン・コントローラが KDC の役目を果たしています。KDC は、ユーザがプログラムと対話できるようにユーザにチケットを発行します。これらのプログラムそのものは、サービス・プリンシパルで表現されます。ユーザが認証され、サービス・チケットを受け取ると、そのユーザはプログラムを使用できるようになります。

具体的には、Kerberos 認証は 3 つの独立したトランザクションで構成されます。

1. クライアントは“TGT”(“チケット保証チケット”) と暗号化セッション・キーを受け取ります。
2. クライアントは TGT とセッション・キーを使用して、InterSystems IRIS のサービス・チケットと別の暗号化セッション・キーの両方を取得します。
3. クライアントはサービス・チケットと 2 番目のセッション・キーを使用して、InterSystems IRIS への認証を行い、必要に応じて保護された接続を確立します。

該当する場合に表示される最初のパスワードのプロンプトを除き、この処理はユーザには表示されないようになっています。

2.1.3 InterSystems IRIS による Kerberos の使用法

Kerberos によって環境が適切に保護されるように、Kerberos 認証をサポートしているすべてのインターシステムズのサービスで Kerberos を有効にし、Kerberos 認証をサポートしていないインターシステムズのサービスは無効にする必要があります。この要件に対する例外は、インターシステムズの安全な境界内で動作することを意図したサービス (ECP など) で、これらは Kerberos をサポートしていません。これらのサービスは、外部に対して安全が確保されている環境で使用するように設計されているので、有効化と無効化のみを設定できます。

2.2 Kerberos の構成の概要

Kerberos 認証を使用できるように InterSystems IRIS インスタンスを構成するには、以下の手順に従います。

1. Kerberos サービスとして実行されるように InterSystems IRIS が設定されていることを確認します。

   その手順は、InterSystems IRIS サーバのオペレーティング・システムと環境のタイプによって異なります。詳細は、"Kerberos を使用したセキュリティ環境の準備" を参照してください。
2. **[認証/ウェブセッションオプション]** 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/ウェブセッションオプション]**) で、該当する Kerberos メカニズムを有効にします。
3. InterSystems IRIS との接続に使用するサービスを決定し、それ以外のサービスをすべて無効にします。それぞれの接続ツールで使用されるサービスのリストは、表 "接続ツールおよびそのアクセス・モードとサービス" を参照してください。
4. クライアント・サーバ接続について、サーバで要求する Kerberos 接続のセキュリティ・レベルを指定します。これは、サービスを使用する接続の構成要素となる Kerberos 機能をどのように決定するかということです。詳細は、"接続のセキュリティ・レベルの指定" を参照してください。
5. クライアント・サーバ接続について、クライアント側の設定を実行します。これにより、アプリケーションから、その実行時に必要となる情報にアクセスできるようになります。この情報には以下のものがあります。
   - InterSystems IRIS を表すサービス・プリンシパルの名前
   - 許可された接続のセキュリティ・レベル

この情報の設定では、Windows 優先サーバの構成などの何らかの構成メカニズムが必要になることがあります。詳細は、"クライアントの設定" を参照してください。

6. 認証プロセスでユーザの認証情報を入手する方法を指定します。この方法には、ユーザの Kerberos 証明書キャッシュをチェックする方法と、Kerberos のパスワードを要求するプロンプトをユーザに示す方法があります。詳細は、"ユーザの認証情報の入手" を参照してください。
7. Web 接続を可能最大限に保護するには、以下の接続にセキュア・チャンネルを設定します。
   - Web ブラウザと Web サーバとの間
   - Web ゲートウェイと InterSystems IRIS サーバとの間

重要　Windows では、ドメイン・アカウントを使用してログインする場合、OS ベースの認証と Kerberos 認証は同じものになります。ローカルでログオンする場合、Kerberos は KDC スプーフィング攻撃の標的となるので安全ではなく、お勧めできません。

2.3 Kerberos およびアクセス・モードについて

接続ツールごとに、サービスを使用して InterSystems IRIS との接続が設定されます。特定のアクセス・モードも使用されます。最大限の保護を実現するには、使用している接続ツールに基づいて、必要なサービスを決定します。使用しないサービスがあれば無効にします。

以下は、接続ツールおよびそのアクセス・モードとサービスのリストです。

テーブル 2-1: 接続ツールおよびそのアクセス・モードとサービス

接続ツール	アクセス・モード	サービス
InterSystems IRIS Telnet	クライアント・サーバ	%Service_Telnet
コールイン	ローカル	%Service_CallIn
コンソール	ローカル	%Service_Console
Java	クライアント・サーバ	%Service_Bindings
JDBC	クライアント・サーバ	%Service_Bindings
ODBC	クライアント・サーバ	%Service_Bindings
ターミナル	ローカル	%Service_Terminal
Web テクノロジ	Web	%Service_WebGateway

2.3.1 ローカル

ローカル・サービスの Kerberos 認証では、ユーザと InterSystems IRIS の両方が有効な Kerberos プリンシパルとして設定されます。この場合、使用されているマシンは 1 台のみで、そのマシン上でプロセスが 1 つのみ実行されています。したがって、ポータルにあるこれらのサービスの構成ページを使用すると、Kerberos プロンプト (管理ポータルでは単に Kerberos とラベル表示されています)、または Kerberos 証明書キャッシュのどちらを使用するかを指定できます。

このシナリオでは、ユーザと InterSystems IRIS は同じマシン上で同じプロセスを使用しているので、両者の間に接続は存在しません。両者はプロセスを共有していることから、安全でない媒体を通じて情報が受け渡されることがなく、したがって両者のデータに特別な保護を設ける必要はありません (この状況をプロセス内認証といいます)。

2.3.2 クライアント・サーバ

クライアント・サーバ・アプリケーションでは、Java、JDBC、ODBC、および Telnet からの接続を扱います。Kerberos 認証を使用するクライアント・サーバ・アプリケーションを介して InterSystems IRIS を操作するユーザには、認証情報が必要です。

サーバとクライアントのそれぞれを構成する必要があります。サーバの構成では、受け入れる接続のタイプを指定します。クライアントの構成では、使用する接続のタイプを指定します。また、ユーザの資格情報を入手する方法も指定できます。

クライアント・サーバ接続では、接続のさまざまなセキュリティ・レベルが Kerberos でサポートされていて、これらのセキュリティ・レベルは InterSystems IRIS サーバ・マシン上で構成します。このレベルには、以下のものがあります。

- Kerberos ー ユーザと InterSystems IRIS との間の最初の認証が Kerberos で管理されます。それ以降の通信は保護されません。
- Kerberos パケット整合性 ー ユーザと InterSystems IRIS との間の最初の認証が Kerberos で管理されます。それ以降取り交わされる各メッセージは、ソースとコンテンツの検証を可能にするハッシュを備えています。これにより、各方向で取り交わされる各メッセージが、そこに示されている送信者から本当に送信されたものであることを検証できます。また、送信者から受信者への伝送の過程でメッセージが改ざんされていないことも検証できます。
- Kerberos 暗号化 ー Kerberos によって最初の認証が管理され、すべての通信の整合性が確保されます。また、暗号化も Kerberos で実行されます。これには、ユーザと InterSystems IRIS の間でやり取りされるすべてのメッセージを各方向でエンド・ツー・エンド暗号化する処理も含まれます。

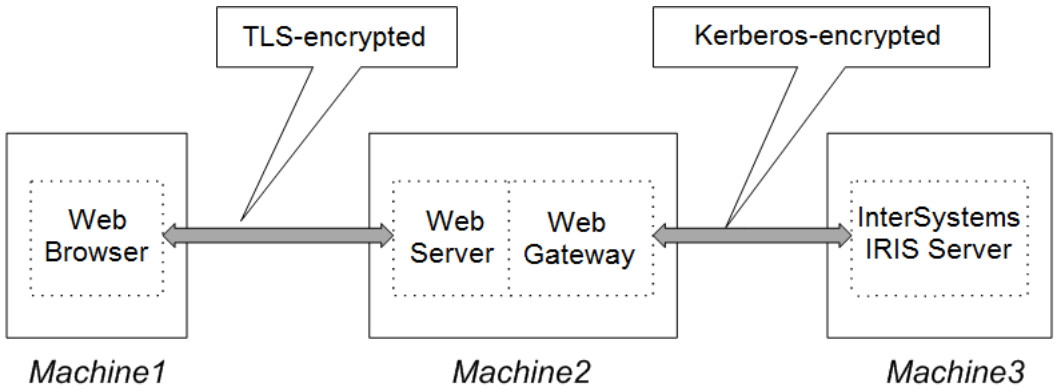
2.3.3 Web

Web アプリケーションの実行では、ユーザと InterSystems IRIS サーバとの間に直接の対話は発生しません。監視からすべての情報を保護するには、ユーザと InterSystems IRIS との間の接続を以下のように暗号化する必要があります。

- TLS を使用してブラウザから Web サーバへの接続を保護するよう Web サーバを構成します。
- Web サーバと Web ゲートウェイは同じ場所に置かれるため、両者間の接続を保護する必要はありません。
- Kerberos 認証と暗号化を使用するように Web ゲートウェイを構成します。このような接続を確立するには、ゲートウェイの Kerberos プリンシパルを使用します。

この場合のアーキテクチャは以下のとおりです。

図 2-1: Kerberos で保護された Web 接続のアーキテクチャ



エンドユーザと InterSystems IRIS との間の通信は、すべて TLS で暗号化されたパイプまたは Kerberos で暗号化されたパイプを通じて行われます。Kerberos で保護された接続の場合、この通信にはエンドユーザの Kerberos 認証が含まれます。

エンドユーザにパスワードを要求するプロンプトを InterSystems IRIS サーバから表示することはできないので、プロンプトを表示する HTML コンテンツをブラウザに送信する API が呼び出されます。送信されたフォームにユーザが所要の情報を入力すると、その情報は Web サーバに返送されます。この情報は、Web サーバから Web ゲートウェイに渡され、さらにそこから InterSystems IRIS そのものを構成する Web サーバに渡されます。Web サーバは、ブラウザを使用しているユーザのためにプロキシとして機能します。この種類の接続がプロキシ接続と呼ばれるのはこのためです。同時に、ローカル・アクセス・モード同様、ユーザに関連するすべての情報はサーバ・マシンに存在します。したがって、Web 接続は、プロセス内認証の 1 つの形式ともいえます。

2.4 接続のセキュリティ・レベルの指定

InterSystems IRIS とのクライアント・サーバ接続では、以下のいずれかのサービスが使用されます。

- **%Service_Bindings** ― Java、JDBC、ODBC
- **%Service_Telnet** ― Telnet

これらのサービスのいずれかを使用している Kerberos 接続では、その接続をサーバが受け入れるためのセキュリティ・レベルを指定する必要があります。目的のサービスでサポートされる接続のセキュリティ・レベルを構成するには、以下の手順に従います。

1. **[認証/Web セッション・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、InterSystems IRIS インスタンス全体に対して有効にする接続セキュリティ・レベルを指定します。以下のレベルにできます。
   - **[Kerberos]** ― 最初の認証のみ
   - **[Kerberos パケット整合性]** ― 最初の認証およびパケットの整合性
   - **[Kerberos 暗号化]** ― 最初の認証、パケットの整合性、およびすべてのメッセージの暗号化

   [認証オプション] ページの詳細は、"認証オプション" を参照してください。
2. **[サービス]** ページ (**[システム管理]**→**[セキュリティ]**→**[サービス]**) の **[名前]** 列でサービス名をクリックすると、そのサービスの **[サービス編集]** ページが表示されます。
3. **[サービス編集]** ページで、Kerberos 接続の一部として要求する接続のセキュリティ・レベルを指定します。レベルを選択したら、**[保存]** をクリックします。

サーバに指定されているセキュリティ・レベルより低いレベルのセキュリティを使用して、クライアントがそのサーバに接続しようとしても、その接続は受け入れられません。サーバに指定されているセキュリティ・レベルより高いレベルのセキュリティを使用してクライアントがそのサーバに接続しようとすると、そのサーバ接続では、サーバに指定されたレベルのセキュリティを使用して認証が試行されます。

2.5 クライアントの設定

クライアント・サーバ・アクセス・モードを使用する場合は、クライアントを構成する必要があります。このプロセスの詳細は、使用している接続テクノロジによって異なります。

2.5.1 Telnet : Kerberos で使用する優先接続サーバの設定

Windows クライアントの場合、Windows 用の InterSystems IRIS Telnet を使用して接続を確立するには、リモート・サーバの一部として格納されている構成情報を使用します。

重要　InterSystems IRIS は、Windows 用に専用の Telnet サーバを備えています。Windows 以外で稼動しているマシンに接続しても InterSystems IRIS Telnet サーバは使用できません。この場合は、オペレーティング・システム付属の Telnet サーバを使用します。サーバ・マシンとの接続が確立されれば、**%Service_Terminal** サービスを使用して InterSystems IRIS を起動できます。

Telnet によるクライアント接続を構成するには、該当のクライアント・マシンに移動します。そのマシンで以下の手順を実行します。

1. InterSystems IRIS ランチャーをクリックし、メニューから **[優先接続サーバ]** を選択します (**[優先接続サーバ]** の選択対象には、現在の優先接続サーバの名前も表示されています)。
2. 表示されたサブメニューから、**[追加/編集]** を選択します。
3. リモート・サーバを新規に作成するには、**[追加]** ボタンをクリックします。既存のサーバを構成するには、接続先の InterSystems IRIS サーバを選択し、**[編集]** ボタンをクリックします。
4. **[接続を追加]** ダイアログが表示されます。このダイアログの **[認証方法]** 領域で [Kerberos] をクリックします。ダイアログが広がり、追加のフィールドがいくつか表示されます。
5. 既存のサーバの値を編集する場合は、このダイアログにある一般的な内容のフィールドで値を変更したり、追加したりする必要はありません。これらの値は、編集するサーバを選択することで自動的に決まります。

   新規サーバを追加する場合に入力するフィールドの詳細は、"リモート・サーバ接続の定義" を参照してください。
6. このダイアログの Kerberos に関するフィールドで、以下のフィールドの値を指定します。
   - 接続セキュリティレベル。Kerberos 認証のみ、Kerberos 認証とパケット整合性、または Kerberos 認証、パケット整合性、および暗号化 を選択できます。
   - サービスプリンシパル名。サービス・プリンシパル名の設定の詳細は、"名前および名前付け規約" を参照してください。
   - Windows マシンへの Telnet 接続を構成する場合は、接続に Windows InterSystems IRIS Telnet サーバを使用することを指定するボックスにチェックを付けます。
7. [OK] をクリックすると、指定した値が保存され、ダイアログが閉じます。

2.5.2 Kerberos で使用する ODBC DSN の設定

InterSystems IRIS では、Windows、UNIX®、または Mac で稼動しているクライアントから DSN (データ・ソース・ノード) への Kerberos で保護された ODBC 接続を、すべてのプラットフォームでサポートしています。クライアントの動作を構成する手順は、以下のようにプラットフォームによって異なります。

- すべてのプラットフォームで、名前と値の組み合わせのセットを受け取る SQLDriverConnect 関数を使用できます。SQLDriverConnect は ODBC API を構成する C 呼び出しで、その説明は Microsoft の Web サイトで入手できます。この名前と値の組み合わせは、Windows 以外のプラットフォームで使用できる初期化ファイルにあるものと同じです。
- Windows 以外のプラットフォームでは、InterSystems ODBC 初期化ファイルを使用して、接続情報を提供する名前と値の対を指定します。このファイルについては、"InterSystems ODBC ドライバの使用法" で概要を説明しています。ODBC 初期化ファイルには、以下の Kerberos 関連の変数があります。

- Authentication Method ー ODBC クライアントを DSN に認証する方法を指定します。0 はインスタンス認証、1 は Kerberos を指定します。
- Security Level ー Kerberos 接続で、接続の保護に使用する機能を指定します。1 は認証のみに Kerberos を使用すること、2 は認証およびクライアントとサーバ間で受け渡されるすべてのパケットの整合性の確保に Kerberos を使用すること、3 は認証、パケットの整合性、およびすべてのメッセージの暗号化に Kerberos を使用することをそれぞれ指定します。
- Service Principal Name ー DSN として機能しているインターシステムズのサービスの名前を指定します。例えば、サービス・プリンシパルは“iris/localhost.domain.com”という名前を持ちます。

これらの変数の名前では、各単語の間にスペースを記述する必要があります。大文字と小文字は区別されません。

- Windows クライアントでは、GUI (ODBC DSN 構成ダイアログ) を使って接続情報を指定できます。**[システム DSN]** タブにオプションが表示されます。この画面には、それぞれのフィールドを説明するヘルプがあります。Windows の [スタート] メニューからの操作でこの画面を表示する方法は、Windows のバージョンによって異なりますが、多くの場合は **[管理ツール]** の中から選択できます。

  重要　64 ビットの Windows の場合、**odbcad32.exe** には 2 つのバージョンがあります。1 つは **C:¥Windows¥System32¥** ディレクトリにあり、もう 1 つは **C:¥Windows¥SysWOW64¥** ディレクトリにあります。64 ビットの Windows を実行している場合は、**C:¥Windows¥SysWOW64¥** にあるものを使用して DSN を構成してください。

2.5.3 Kerberos で使用する Java クライアントまたは JDBC クライアントの設定

InterSystems IRIS には、Java クライアントの構成を支援するユーティリティとして機能する Java クラスが用意されています。クライアントを構成する準備ができた段階で、このクラスを実行します。以下はその方法です。

1. Kerberos で使用するクライアントを構成するには、以下のように Java の Configure コマンドを発行します。

   ```
   java -classpath '$IRIS_INSTALL_DIRECTORY/dev/java/lib/JDK18/*' com.intersystems.jgss.Configure
   ```

   これにより、JDK ディレクトリ内からだけでなく、マシン上の任意の場所から Configure を実行できるようになります。このコマンドの詳細は、Windows のパス・スタイルと JDK11 の使用のどちらに対応しているかなど、サイトによって異なることに留意してください。

   このプログラムは、Java Generic Security Services (JGSS) を使用して以下のアクションを実行します。

   - 必要に応じて、**java.security** ファイルを変更します。
   - **isclogin.conf** ファイルを作成または変更します。

2. 次にこのプログラムでは、**krb5.conf** ファイルを作成して構成することを求めるプロンプトが表示されます。このファイルが存在する場合は、その既存の **krb5.conf** を使用するか、新しいファイルと置き換えるか選択することを求められます。ファイルを置き換える場合は、以下の情報を求められます。

   a. Kerberos レルム ー ドメインの既定値として、ローカル・ドメイン名が小文字で提示されます。

   b. プライマリ KDC ー ローカル・マシン名のみを指定すれば、それに Kerberos レルムの名前が自動的に付加されます。

   c. セカンダリ KDC ー ゼロ個以上の KDC の名前を指定して、それにプライマリ KDC の内容を複製できます。

3. これらの情報を入力した後、コマンドをもう一度実行します(実行するように指示されます)。

4. **krb5.conf** の置き換えを求められたら、既存のファイルをそのまま残すことを選択します。指定した Kerberos レルムにあるプリンシパルのユーザ名とパスワードが要求されるので、指示どおりに入力します。これによって、接続がテストされます。

テストが正常に終了すれば、クライアントの構成は完了です。

2.6 ユーザの認証情報の入手

すべてのアクセス・モードについて、アプリケーションでユーザの認証情報を入手する方法を指定する必要があります。この方法には、既存の認証情報キャッシュから得る方法と、ユーザにユーザ名とパスワードを要求して得る方法があります。

2.6.1 ローカル・アクセス・モード用の認証情報の入手

ローカル・アクセス・モードでは、InterSystems IRIS と同じマシンにユーザの認証情報が存在します。この状況では、InterSystems IRIS に接続するサービスがアプリケーションで使用されています。このサービスには、以下のものがあります。

- **%Service_CallIn**
- **%Service_Console**
- **%Service_Terminal**

資格情報の入手方法を指定するには、以下の手順に従います。

1. **[サービス]** ページ (**[システム管理]** > **[セキュリティ]** > **[サービス]**) で、**[名前]** 列からサービスを選択します。選択したサービスの **[サービス編集]** ページが表示されます。
2. **[サービス編集]** ページで、資格情報の入手方法を指定します。プロンプトを表示してユーザに要求する方法 (**[Kerberos]** チェック・ボックス) または資格情報キャッシュを使用する方法 (**[Kerberos 証明書キャッシュ]** チェック・ボックス) を選択します。両方を選択しないでください。

   [保存] をクリックすると、選択した設定内容が使用されます。

注釈　Kerberos (プロンプト) と Kerberos 証明書キャッシュを使用する方法の両方をサービスに対して有効にすると、資格情報キャッシュによる認証が優先されます。この動作は InterSystems IRIS ではなく Kerberos によって指定されたものです。

ドメイン・コントローラを備えた Windows (Windows で多く使用される構成) では、ログインによって Kerberos 証明書キャッシュが設定されます。UNIX®、Linux、および macOS の既定の状態では、通常、Kerberos 資格情報が存在しません。したがって、Kerberos プロンプトを使用するように InterSystems IRIS を構成します。これらのシステムでは、ユーザは以下のいずれかの方法で資格情報を入手できます。

- ターミナルを起動する前に kinit を実行する。
- ユーザに対してログイン・プロセスで Kerberos 認証が実行される条件で、システムにログインする。

これらの状況では、資格情報キャッシュを使用するように InterSystems IRIS を構成できます。

2.6.2 クライアント・サーバ・アクセス・モード用の認証情報の入手

クライアント・サーバ・アクセス・モードでは、クライアント・アプリケーションをホストしているマシンにユーザの認証情報が存在します。この場合は、クライアントが以下のいずれの接続方法を使用しているかによって、認証情報の入手方法を指定する形式が異なります。

- ODBC および Telnet

- Java および JDBC

2.6.2.1 ODBC および Telnet

これらの接続ツールで使用される基盤の InterSystems IRIS コードでは、エンドユーザが既に資格情報を持っていることが前提となっています。したがって、プロンプトの表示は不要です。

Windows では、ドメインにログオンしているすべてのユーザごとに資格情報キャッシュがあります。

Windows 以外のオペレーティング・システムでは、そのユーザに対してオペレーティング・システムで Kerberos 認証を実行済みである場合、またはユーザが明示的に kinit を実行済みである場合に、そのユーザの認証情報キャッシュが存在します。それ以外の場合は、キャッシュにユーザの資格情報が存在しないので、接続ツールは認証に失敗します。

注釈　オペレーティング・システムによっては、一部の接続ツールが使用できないことがあります。

2.6.2.2 Java および JDBC

Java および JDBC を使用する際には、2 つの異なる Java 実装 (Oracle による実装と IBM による実装) があります。2 つの実装に共通する動作もあれば、互いに異なる動作もあります。

注釈　Java の IBM による実装は、バージョン 8 の場合にのみ利用可能です。これ以降のバージョンの場合、IBM ではオープン・ソース・バージョンがサポートされます。

どちらの実装でも、**java.util.Properties** クラスのインスタンスのプロパティに接続情報が格納されます。以下のプロパティがあります。

- user ー InterSystems IRIS サーバに接続しているユーザの名前。この値は特定の接続動作に対してのみ設定されます。
- password ー そのユーザのパスワード。この値は特定の接続動作に対してのみ設定されます。
- service principal name ー InterSystems IRIS サーバの Kerberos プリンシパル名。この値はすべての接続動作に対して設定されます。
- connection security level ー この接続に対して Kerberos が提供する保護のタイプ。1 は認証のみに Kerberos を使用すること、2 は認証およびクライアントとサーバ間で受け渡されるすべてのパケットの整合性の確保に Kerberos を使用すること、3 は認証、パケットの整合性、およびすべてのメッセージの暗号化に Kerberos を使用することをそれぞれ指定します。この値はすべての接続動作に対して設定されます。

以下の説明では、**java.util.Properties** クラスのインスタンスを connection_properties オブジェクトとして指定しています。各プロパティの値は、次のような connection_properties.put メソッドの呼び出しによって設定されます。

```
String principalName = "MyServer";
connection_properties.put("service principal name",principalName);
```

どちらの実装でも、認証情報に関連する動作は **isclogin.conf** ファイル内の特定のパラメータの値で決まります (このファイルの詳細は、"Kerberos で使用する Java クライアントまたは JDBC クライアントの設定" を参照)。

2 つの Java 実装の動作には、次の 2 つの相違点があります。

- 認証情報に関連する動作を指定するために **isclogin.conf** ファイルに設定されるパラメータの名前が実装間で異なります。
  - IBM では、パラメータ名は useDefaultCcache です。
  - Oracle では、パラメータ名は useTicketCache です。
- 使用できる動作が実装間で異なります。これらは以下のセクションで説明しています。

IBM の実装を使用しているクライアントに対する動作の指定

オプションは以下のとおりです。

- 認証情報のキャッシュを使用するには、useDefaultCcache パラメータの値を True に設定し、user プロパティと password プロパティには値を設定しません。資格情報キャッシュが使用できない場合は、例外が発生します。
- プログラムによる処理で渡されるユーザ名とパスワードを使用するには、useDefaultCcache パラメータの値を False に設定し、user プロパティと password プロパティに該当の値を設定します。
- ユーザにユーザ名とパスワードを要求するには、useDefaultCcache パラメータの値を False に設定し、user プロパティと password プロパティには値を設定しません。これらのプロパティに値を設定しないことにより、InterSystems IRIS に付属するライブラリのクラスを使用して、ユーザ名とパスワードを要求するプロンプトを生成できます。

Oracle の実装を使用しているクライアントに対する動作の指定

オプションは、以下のとおりです。

- プログラムによる処理で渡されるユーザ名とパスワードを排他的に使用するには、useTicketCache パラメータの値を False に設定し、user プロパティと password プロパティに該当の値を設定します。
- ユーザにユーザ名とパスワードを排他的に要求するには、useTicketCache パラメータの値を False に設定し、user プロパティと password プロパティには値を設定しません。これらのプロパティに値を設定しないことにより、InterSystems IRIS に付属するライブラリのクラスを使用して、ユーザ名とパスワードを要求するプロンプトを生成できます。
- 認証情報キャッシュを排他的に使用するには、useTicketCache パラメータの値を True に設定します。余分なアクションが発生しないようにするには、user プロパティと password プロパティに、存在しない値を設定します。これによってプロンプトが表示されなくなり、このプロパティの値に基づいて認証しようとしても必ず失敗するようになります。
- 認証情報キャッシュの使用を試し、それが使用できなかった場合にプログラムによる処理で渡されるユーザ名とパスワードを使用するには、useTicketCache パラメータの値を True に設定し、user プロパティと password プロパティに該当の値を設定します。資格情報キャッシュが存在しない場合は、これらのプロパティの値が使用されます。
- 認証情報キャッシュを試し、それが使用できなかった場合にユーザ名とパスワードを要求するには、useTicketCache パラメータを True に設定し、user プロパティと password プロパティには値を設定しません。資格情報キャッシュが存在しない場合は、InterSystems IRIS に付属するライブラリのクラスを使用して、ユーザ名とパスワードを要求するプロンプトを生成できます。

2.6.3 Web アクセス・モード用の認証情報の入手

Kerberos を使用する Web ベース接続では、ユーザ名とパスワードを要求するプロンプトが必ず表示されます。このプロンプトを通じて認証されれば、ユーザ名とパスワードがメモリに置かれ、不要になった時点で破棄されます。

2.7 Web 接続で使用するセキュア・チャンネルの設定

Web 接続に最大限の保護を実現するには、2 つの重要な通信経路であるブラウザと Web サーバの間および Web ゲートウェイと InterSystems IRIS の間でセキュア・チャンネルを使用することをお勧めします。これにより、Kerberos のユーザ名とパスワードなどのあらゆる情報が、複数のポイント間で伝送されるときに保護されます。以下の各通信チャンネルを保護するには、以下の手順に従います。

- Web ブラウザと Web サーバの間
- Web ゲートウェイと InterSystems IRIS の間

2.7.1 Web ブラウザと Web サーバ間の接続の保護

Web ブラウザと Web サーバ間の接続を保護するための一般的な手段は、TLS (Transport Layer Security) を使用することです。InterSystems IRIS には、この保護を実現するためのこのテクノロジの実装は用意されていませんが、この機能を提供する製品がサードパーティから数多く提供されています。

2.7.2 Web ゲートウェイと InterSystems IRIS 間での Kerberos で保護された接続の設定

Web ゲートウェイと InterSystems IRIS サーバの間に暗号化されたセキュア・チャンネルを設定するには、このゲートウェイを表す Kerberos プリンシパルが必要です。このプリンシパルによって InterSystems IRIS への暗号化接続が確立され、すべての情報はこの接続を通じて伝送されます。こうすることで、エンドユーザを InterSystems IRIS に認証でき、このプロセスの間でデータ盗用が防止されます。

注釈　TLS で保護された Web ゲートウェイと InterSystems IRIS サーバの間の接続設定に関する詳細は、"TLS を使用して InterSystems IRIS に接続するための Web ゲートウェイの構成" を参照してください。

以下はその方法です。

1. ゲートウェイを表す Kerberos プリンシパルの名前を決定または選択します。

   Windows の場合は、ゲートウェイ・ホストのネットワーク・サービス・セッションを表すプリンシパル名が、この Kerberos プリンシパル名になります (つまり、ゲートウェイをホストしているマシンの名前に "$" を付加した machine_name$ という形式で、例えば Athens$ となります)。Windows 以外のプラットフォームの場合は、ゲートウェイの構成画面でユーザ名として入力した任意の有効なプリンシパル名が、この Kerberos プリンシパル名になります。このプリンシパル名で、キー・テーブル・ファイルにある適切なキーが特定されます。

2. ゲートウェイの Kerberos プリンシパルと同じ名前を持つユーザを InterSystems IRIS に作成します。これを行うには、"ユーザ・アカウントの作成" の手順に従います。

3. 必要な任意のリソースに対する使用、読み取り、または書き込みの許可 (特権ともいいます) をこのユーザに与えます。これらの特権をロールに関連付け、次にユーザをこのロールに関連付けることで、ユーザに特権を与えることができます。

4. **%Service_WebGateway** サービスを構成します。これを行うには、"サービスのプロパティ" で説明しているフィールドに入力します。

5. ゲートウェイからサーバにアクセスできるようにゲートウェイを構成します。以下はその方法です。

   a. 管理ポータルのホーム・ページで、**[ウェブゲートウェイ管理]** ページ (**[システム管理]** > **[構成]** > **[ウェブゲートウェイ管理]**) に移動します。

   b. [ウェブゲートウェイ管理] ページでは、左側に選択項目のセットが表示されています。**[構成]** の下にある **[サーバ接続]** をクリックします。**[サーバ接続]** ページが表示されます。

   c. **[サーバ接続]** ページでは、新規の構成の追加、または既存の構成の編集が可能です。新規の構成を追加するには、**[サーバ追加]** ボタンをクリックします。既存の構成を編集するには、左側のリストから目的の構成を選択し、**[サーバ編集]** ラジオ・ボタンを選択して **[実行]** をクリックします。サーバ接続パラメータを編集または構成するためのページが表示されます。ヘルプ画面に説明がある一般的なパラメータに加え、このページでは、ゲートウェイのセキュリティに関するパラメータを指定できます。Kerberos 接続の場合は、以下のパラメータがあります。

      - **[接続セキュリティレベル]** – この接続に対して Kerberos で提供する保護の種類を選択します。(前の手順で Web サービスに対して指定したセキュリティのタイプ以上のセキュリティ・レベルを選択する必要があります。)
      - **[ユーザ名]** – ゲートウェイを表す Kerberos プリンシパルの名前。(このプロセスの最初の手順で使用したプリンシパル名と同じ名前にする必要があります。)

- **[パスワード]** – これには値を指定しないでください (このフィールドは、インスタンス認証で使用するためにゲートウェイを構成するときに使用します)。
- **[プロダクト]** – InterSystems IRIS。
- **[サービスプリンシパル名]** – InterSystems IRIS サーバを表すプリンシパルの名前。これには通常、“iris/machine.domain” という形式で、標準の Kerberos プリンシパル名を指定します。`iris` は InterSystems IRIS のサービスであることを示す固定文字列、machine はマシン名、domain は “intersystems.com” のようなドメイン名です。
- **[キーテーブル]** – Windows で InterSystems IRIS のインスタンスに接続する場合は、このフィールドを空白のままにしておきます。Windows 以外のオペレーティング・システムの場合は、Web ゲートウェイに属する永続キーを収めたキータブ・ファイルの名前をフル・パスで指定します。

これらの値をすべて入力した後、**[設定を保存]** ボタンをクリックすると、入力した値が保存されます。

これで、Web サービスを構成できるようになります。これは、Web アプリケーションのサポートに必要な基盤となるインフラストラクチャを、CSP サービスで提供できるということです。

保護された Web アプリケーションを作成する場合、アプリケーション開発者は以下の処理を行う必要があります。

1. 認証方法を選択する。
2. アプリケーションのロールを構成する。
3. 必要に応じて、ブラウザと Web サーバ間の接続に TLS が使用されていることを確認する。

3

オペレーティング・システム・ベースの認証

3.1 OS ベースの認証について

InterSystems IRIS では、オペレーティング・システム・ベース (OS ベース) の認証がサポートされます。オペレーティング・システム認証では、オペレーティング・システムのユーザ ID を使用して、InterSystems IRIS のユーザを識別します。オペレーティング・システム認証が有効になっている場合、ユーザは、オペレーティング・システムのプロトコルに従って、オペレーティング・システムに対する認証を実行します。例えば、UNIX® でネイティブの機能に相当するものは、従来からあるログイン・プロンプトです。このプロンプトでは、パスワードのハッシュ値が、/etc/passwd ファイルに格納されている値とオペレーティング・システム・レベルで比較されます。ユーザが初めて InterSystems IRIS に接続しようとすると、プロセスのオペレーティング・システム・レベルのユーザ識別情報が取得されます。このユーザ識別情報が InterSystems IRIS のユーザ名と一致すれば、そのユーザは認証されます。

この機能は、サーバ側で実行されるプロセスにのみ適用されます。例えば、ターミナル・ベースのアプリケーション (ターミナルによる接続など) やオペレーティング・システムによって起動するバッチ・プロセスなどです。 他のマシンから InterSystems IRIS に接続するアプリケーションでは、この機能は使用できません。例えば、あるマシン上のスタジオが、別のマシン上の InterSystems IRIS サーバに接続する場合です。

この機能は、通常、UNIX® システムで利用されていますが、Windows コンソールでも使用できます。

OS ベースの認証は、ローカル・プロセスに対してのみ使用できます。具体的には、以下のとおりです。

- コールイン (**%Service_Callin**)
- コンソール (**%Service_Console**)
- ターミナル (**%Service_Terminal**)

3.2 OS ベースの認証の構成

このタイプの認証の使用を設定するには、以下の手順に従います。

1. **[認証/Web セッション・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]**) ページで、**[オペレーティングシステム認証を許可]** を選択します。
2. **[サービス]** ページ (**[システム管理]** > **[セキュリティ]** > **[サービス]**) で、**[名前]** 列からサービスを選択します。選択したサービスの **[サービス編集]** ページが表示されます。
3. **[サービス編集]** ページで、**[オペレーティングシステム]** チェック・ボックスにチェックを付けてオペレーティング・システム・ベースの認証を選択します。

[保存] をクリックすると、選択した設定内容が使用されます。

このタイプの認証では、これ以上のアクションは不要です。

注釈　Windows では、ドメイン・アカウントを使用してログインする場合、OS ベースの認証と Kerberos 認証は同じものになります。

3.3 %Service_Console に関するメモ

コンソール (**%Service_Console**) は Windows ベースのサービスであり、Windows ドメインのログインでは Kerberos を使用することが普通なので、コンソールの OS ベース認証でローカル・ログインに対する認証が得られます。

3.4 %Service_Callin に関するメモ

コールイン (**%Service_Callin**) では、OS レベルのプロンプトからのみ OS ベース認証を使用できます。プログラム処理でコールインを使用する場合は、OS ベース認証がサポートされていません。この場合は、認証されていないアクセスのみが可能です。

4

インスタンス認証

4.1 インスタンス認証について

InterSystems IRIS 自体で、インスタンス認証と呼ばれるログイン・メカニズムを提供できます (管理ポータルでは、**パスワード認証**と呼びます)。具体的には、ユーザ・アカウントごとにパスワードの値が InterSystems IRIS に保持されており、ログインするユーザが入力した値とその値が比較されます。従来の OS ベースの認証同様、InterSystems IRIS でも、パスワードをハッシュ化した値が格納されています。ユーザがログインすると、入力されたパスワードがハッシュ化され、これら 2 つのハッシュ値が比較されます。システム管理者は、パスワードの最小長などの一定のパスワード基準を設定することで、ユーザが選択するパスワードに所定の堅牢性を確保できます。この基準は、"パスワードの強固さとパスワードのポリシー" で説明されています。

InterSystems IRIS は、パスワードの回復不可能な暗号化ハッシュのみを格納します。ハッシュは、Public Key Cryptography Standard #5 v2.1 "Password-Based Cryptography Standard" の定義に従って、HMAC-SHA-512 擬似ランダム関数と PBKDF2 アルゴリズムを使用して計算されます。 現在の実装では 10,000 回の反復、64 ビットのソルトを使用して、64 バイトのハッシュ値が生成されます。別のアルゴリズムを指定するか、反復回数を増やすには、それぞれ Security.System.PasswordHashAlgorithm メソッドと Security.System.PasswordHashWorkFactor メソッドを使用します。これらのハッシュ値から元のパスワードを回復するための既知のテクニックはありません。

インスタンス認証による認証で使用できるサービスは、以下のとおりです。

- **%Service_Binding**
- **%Service_CallIn**
- **%Service_ComPort**
- **%Service_Console**
- **%Service_Telnet**
- **%Service_Terminal**
- **%Service_WebGateway**

4.2 インスタンス認証の構成の概要

インスタンス認証を使用するサービスは、以下のように構成する必要があります。

1. **[認証/ウェブセッションオプション]** 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/ウェブセッションオプション]**) で、**[パスワード認証を許可]** を選択して、インスタンス認証を使用した認証を有効にします。
2. 特定のサービスについては、**[サービス]** ページ (**[システム管理]→[セキュリティ]→[サービス]**) に移動し、**[名前]** 列で [%Service_Bindings] などの目的のサービスを選択します。そのサービスの **[サービス編集]** ページが表示されます。
3. このページでインスタンス認証を選択します。インスタンス認証は、認証タイプのリストで単に **[パスワード]** と表示されています。
4. **[保存]** をクリックすると、選択した設定内容が保存されます。
5. この基本的な手順に加え、一部のサービスではさらに構成が必要です。これは、以下のセクションで説明します。
   - Web
   - ODBC
   - Telnet

4.3 Web

Web アクセスでは、必要に応じて、インスタンス認証を通じ、Web ゲートウェイでゲートウェイ自身を InterSystems IRIS サーバに認証するように要求できます。この構成を実行するには、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、**[ウェブゲートウェイ管理]** ページ (**[システム管理] > [構成] > [ウェブゲートウェイ管理]**) に移動します。
2. [ウェブゲートウェイ管理] ページでは、左側に選択項目のセットが表示されています。**[構成]** の下にある **[サーバ接続]** をクリックします。**[サーバ接続]** ページが表示されます。
3. **[サーバ接続]** ページでは、新規の構成の追加、または既存の構成の編集が可能です。新規の構成を追加するには、**[サーバ追加]** ボタンをクリックします。既存の構成を編集するには、左側のリストから目的の構成を選択し、**[サーバ編集]** ラジオ・ボタンを選択して **[実行]** をクリックします。サーバ接続パラメータを編集または構成するためのページが表示されます。ヘルプ画面に説明がある一般的なパラメータに加え、このページでは、ゲートウェイのセキュリティに関するパラメータを指定できます。インスタンス認証接続の場合は、以下のパラメータがあります。
   - **[接続セキュリティレベル]** – **[パスワード]** をドロップダウン・リストから選択してインスタンス認証を使用します。
   - **[ユーザ名]** – ゲートウェイ・サービスが実行されるユーザ名 (インストール・プロセスでは、この目的のために CSPSystem ユーザが作成されます)。このユーザ (CSPSystem またはその他のユーザ) には、有効期限を設定する必要はありません。つまり、Expiration Date プロパティの値は 0 になります。
   - **[パスワード]** – 上で入力したユーザ・アカウントに関連付けられたパスワード。
   - **[プロダクト]** – InterSystems IRIS。
   - **[サービスプリンシパル名]** – これには値を指定しないでください(このフィールドは、Kerberos で使用するためにゲートウェイを構成するときに使用します)。
   - **[キーテーブル]** – これには値を指定しないでください(このフィールドは、Kerberos で使用するためにゲートウェイを構成するときに使用します)。

   これらの値をすべて入力した後、**[設定を保存]** ボタンをクリックすると、入力した値が保存されます。

ゲートウェイに対する認証要件と、そのゲートウェイを使用するアプリケーションに対する認証要件との間に直接の関係はない点に注意する必要があります。例えば、Kerberos 認証を使用するようにゲートウェイを構成していても、Web アプリケーションに対する認証メカニズムとしてインスタンス認証を要求できます。ゲートウェイに対して認証方法をまったく構

成していなくても同様です。実際、ゲートウェイそのものに対して特定の認証メカニズムを選択しても、Web アプリケーションに対して技術的な要件は発生しません。逆に Web アプリケーションに対する認証メカニズムを選択しても、ゲートウェイに対する要件は発生しません。同時に、CSP では他の場合に比べ、ペア化が発生する可能性が高くなります。Web アプリケーションで Kerberos 認証を使用している場合、ゲートウェイに対して他の形式の認証を使用すると、暗号化されていないチャンネルを通じて Kerberos 認証の情報が流れることになります。その結果、Kerberos 認証の効果が低下する可能性があります。

インスタンス認証を使用している Web アプリケーションでは、エンドユーザのユーザ名とパスワードはブラウザから Web サーバに渡され、次にその Web サーバと同じ場所にある Web ゲートウェイに渡されます。このゲートウェイは InterSystems IRIS サーバとの独自の接続を備えているので、ユーザ名とパスワードはその InterSystems IRIS サーバに渡されます。ゲートウェイと InterSystems IRIS サーバとの接続を確立するために、このゲートウェイでは CSPSystem アカウントが使用されます。このアカウントは、InterSystems IRIS の事前定義アカウントの 1 つです。

既定では、これらのトランザクションはどれも暗号化されていません。TLS を使用すれば、ブラウザから Web サーバに送信されるメッセージを暗号化できます。Kerberos を使用すると、"Web 接続で使用するセキュア・チャンネルの設定" で説明したように、ゲートウェイから InterSystems IRIS サーバに送信されるメッセージを暗号化できます。Kerberos を使用していない場合は、ゲートウェイのホスト・マシンと InterSystems IRIS サーバのホスト・マシンとの間の接続を物理的に保護することをお勧めします。例えば、両方のマシンをロックされた同じ領域に置いて、両者間で直接的な物理接続を確立することにより、接続を保護できます。

4.4 ODBC

InterSystems IRIS では、サポート対象のすべてのプラットフォーム間で ODBC 接続に対するインスタンス認証がサポートされています。この場合は、クライアント側の構成が必要です。クライアントの動作を構成する手順は、以下のようにプラットフォームによって異なります。

- Windows 以外のプラットフォームでは、InterSystems ODBC 初期化ファイルを使用して、接続情報を提供する名前と値の対を指定します。このファイルについては、"InterSystems ODBC ドライバの使用法" で概要を説明しています。このファイルには、インスタンス認証に関連する以下の変数が含まれています。
  - Authentication Method ― ODBC クライアントを DSN に認証する方法を指定します。0 はインスタンス認証、1 は Kerberos を指定します。
  - UID ― DSN への接続に使用する既定のユーザ・アカウントの名前を指定します。実行時には、アプリケーションの動作に応じて、エンドユーザはこの値を別のユーザ・アカウントに置き換えることができます。
  - Password ― 既定のユーザ・アカウントに関連付けられたパスワードを指定します。エンドユーザによる UID 値の変更が許可されている場合は、新しく指定されたユーザのパスワードがアプリケーションで受け入れられます。
- Windows クライアントに対する接続情報は、GUI を使用した指定、またはプログラムによる指定のいずれかが可能です。
  - GUI では、ODBC DSN を構成するダイアログを使用します。**[システム DSN]** タブにオプションが表示されます。この画面には、それぞれのフィールドを説明するヘルプがあります。Windows の [スタート] メニューからの操作でこの画面を表示する方法は、Windows のバージョンによって異なりますが、多くの場合は **[コントロール パネル]** の **[管理ツール]** で **[データ・ソース (ODBC)]** の画面に表示されます。
  - プログラムで指定する場合は、名前と値の組み合わせのセットを受け取る SQLDriverConnect 関数を使用できます。SQLDriverConnect は、ODBC API の一部である C 呼び出しです。この名前と値の組み合わせは、パスワードが PWD キーワードで識別される点を除けば、Windows 以外のプラットフォームで使用できる初期化ファイルにあるものと同じです。

4.5 Telnet

クライアントで Windows 用の InterSystems IRIS Telnet を使用して接続を確立するには、InterSystems IRIS リモート・サーバの一部として格納されている構成情報を使用します。リモート・サーバを構成するには、クライアント・マシンに移動します。そのマシンで以下の手順を実行します。

1. InterSystems IRIS ランチャーをクリックし、メニューから **[優先接続サーバ]** を選択します (**[優先接続サーバ]** の選択対象には、現在の優先接続サーバの名前も表示されています)。
2. 表示されたサブメニューから、**[追加/編集]** を選択します。
3. リモート・サーバを新規に作成するには、**[追加]** ボタンをクリックします。既存のサーバを構成するには、接続先の InterSystems IRIS サーバを選択し、**[編集]** ボタンをクリックします。
4. **[接続を追加]** ダイアログが表示されます。このダイアログの **[認証方法]** 領域で、インスタンス認証を指定する **[パスワード]** をクリックします。
5. 既存のサーバの値を編集する場合は、このダイアログにある一般的な内容のフィールドで値を変更したり、追加したりする必要はありません。これらの値は、編集するサーバを選択することで自動的に決まります。

   新規サーバを追加する場合に入力するフィールドの詳細は、"リモート・サーバ接続の定義" を参照してください。
6. [OK] をクリックすると、指定した値が保存され、ダイアログが閉じます。

重要　Windows 以外で稼動しているマシンに Telnet を使用して接続しても InterSystems IRIS Telnet サーバは使用できません。この場合は、オペレーティング・システム付属の Telnet サーバを使用します。サーバ・マシンとの接続が確立されれば、**%Service_Terminal** サービスを使用して InterSystems IRIS に接続できます。

5

代行認証

5.1 代行認証について

5.1.1 代行認証の背景

InterSystems IRIS では、代行認証がサポートされます。代行認証を使用すると、企業の既存の認証システムなど、カスタムのメカニズムを実装して、インターシステムズのセキュリティに含まれる認証アクティビティやロール管理アクティビティを置き換えることができます。アプリケーション開発者は、代行認証コードのコンテンツを完全に制御します。代行認証は、InterSystems IRIS のインスタンスの %SYS ネームスペースで ZAUTHENTICATE ルーチンが検出された場合に行われます。このようなルーチンが存在する場合、InterSystems IRIS はこれを使用して、新規コードと既存のコードのいずれかを呼び出してユーザを認証します。InterSystems IRIS には **ZAUTHENTICATE.mac** ルーチンがあり、これは ZAUTHENTICATE ルーチンを作成するためのテンプレートとして機能します。

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。

5.1.2 代行認証の動作

ユーザがログインを試行して、InterSystems IRIS が代行認証を呼び出す場合、イベントのシーケンスは以下のようになります。

1. サービスまたはアプリケーションが代行認証を使用する場合、ログイン試行によって ZAUTHENTICATE ルーチンが自動的に呼び出されます。このルーチンの認証コードは、任意のユーザ定義の ObjectScript、クラス・メソッド、または $ZF コールアウト・コードにできます。
2. 次に行う手順は、認証が成功するかどうか、および今回が ZAUTHENTICATE を使用した最初のログインであるかどうかによって異なります。
   - ZAUTHENTICATE が成功し、今回がこのメカニズムを使用した初めてのユーザ認証である場合、そのユーザはタイプを“代行ユーザ”として InterSystems IRIS ユーザのリストに追加されます。ZAUTHENTICATE によってロールまたは他の特性が設定されると、そのロールまたは特性がそのユーザのプロパティの一部になります。
   - ZAUTHENTICATE が成功し、今回が最初のログインでない場合、ZAUTHENTICATE によってそのユーザのプロパティが更新されます。
   - ZAUTHENTICATE が失敗すると、アクセス拒否のエラーが表示されます。ユーザはシステムにアクセスできません。この原因を調べる手順は次のとおりです。
     a. ユーザ・プロファイルの **[ログイン失敗の理由]** フィールドの内容をチェックします。

b. さらに、監査ログで %System/%Login/LoginFailure イベントをチェックして、詳細な情報を確認します。監査や LoginFailure イベントが有効になっていない場合は、これらを両方とも有効にしてログイン失敗の状況を再現する必要がある場合があります。

3. インスタンスと関連サービスの 2 要素認証が有効な場合は、ユーザの **PhoneNumber** および **PhoneProvider** プロパティが設定されていることが確認されます。これらのプロパティが設定されている場合は、2 要素認証が実行されます。これらのプロパティが設定されていない場合は、2 要素認証に進むことはできず、ユーザは認証されません。
4. **[ユーザ]** ページ (**[システム管理]** > **[セキュリティ]** > **[ユーザ]**) で、ユーザ・リストの [タイプ] 列に代行ユーザとしてユーザが表示されます。管理ポータルに表示されるユーザのプロパティは読み取り専用です。InterSystems IRIS から編集することはできません (これらの情報はすべて InterSystems IRIS の外部から取得されるため)。

   注釈　InterSystems IRIS パスワード・ユーザが、同時に代行ユーザになることはできません。

5.2 代行認証の構成の概要

代行認証を使用するには、以下の手順を実行します。

1. ZAUTHENTICATE ルーチンで、ユーザ定義の認証コードを作成します。これには、2 要素認証を使用できます。このルーチンは、ロールや他のユーザ・プロパティの指定など、ユーザ・アカウントの基本的な設定も実行することができます。

   HealthShare Health Connect を使用している場合は、このドキュメントの説明に従って ZAUTHENTICATE カスタム・ルーチンを作成します。

   HealthShare Unified Care Record を使用している場合、Unified Care Record には独自のバージョンのルーチンが付属しているため、カスタム・バージョンの ZAUTHENTICATE を作成して代行認証を実装することはできません。代わりに、クラス **HS.Local.ZAUTHENTICATE** のメソッドをカスタマイズする必要があります。詳細は、ブック "Authenticating Users in Unified Care Record" の "Customizing Authentication via Local ZAUTHENTICATE" を参照してください。
2. [認証オプション] ページで、InterSystems IRIS インスタンスに対して代行認証を有効にします。
3. 必要に応じて、関連するサービス、アプリケーション、またはその両方に対して代行認証を有効にします。
4. オプションとして、InterSystems IRIS インスタンスおよび必要に応じて Web アプリケーションとクライアント・サーバ・アプリケーションで、2 要素認証を有効化します。

例えば、インスタンスの管理ポータルで代行認証を使用するには、以下の手順を実行します。

1. ZAUTHENTICATE で、ユーザ定義の認証コードを作成します。
2. InterSystems IRIS インスタンス全体の代行認証を有効にします。
3. /csp/sys* アプリケーションのセットの代行認証を有効にします。

5.3 代行 (ユーザ定義) 認証コードの作成

ここでは、ユーザ専用の ZAUTHENTICATE ルーチン作成のさまざまな側面について説明します。

- 認証コードの基礎

- シグニチャ
- 認証コード
- ロールと他のユーザ特性の値の設定
- 返り値とエラー・メッセージ

5.3.1 認証コードの基礎

インターシステムズが提供するサンプル・ルーチン ZAUTHENTICATE.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 でルーチンを開いて、その内容をコピーする方が簡単です。

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

1. ZAUTHENTICATE.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 ZAUTHENTICATE.mac ルーチンにコピーして保存します。
2. ZAUTHENTICATE.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、カスタム認証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 ZAUTHENTICATE の認証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。

5.3.2 シグニチャ

ZAUTHENTICATE のシグニチャは以下のとおりです。

ObjectScript

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

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

各項目の内容は次のとおりです。

- 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表す文字列 (**%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表す文字列。これは、スタジオや ODBC など、**%Service_Bindings** サービスで使用します。
- Username ー ルーチンのコードによって検証されるユーザが入力したアカウントの名前を表す文字列。
- Password ー 検証されるユーザが入力したパスワードを表す文字列。
- 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定義する返り値の配列。

InterSystems IRIS が ZAUTHENTICATE を呼び出すと、これらの引数の値がルーチンに提供されます。

注釈　インターシステムズ製品の古いバージョンでは、ZAUTHENTICATE は引数を 4 つ取ります。下位互換性を保つために、引き続き引数が 4 つのバージョンを使用できます。コードを古いバージョンから新しいバージョンに更新する場合、新しい引数は 2 番目と 5 番目になることに注意してください。すなわち、Namespace と Credentials です。

5.3.3 認証コード

認証コードのコンテンツはアプリケーションに固有です。認証が成功すると、ルーチンから $$$OK マクロが返されます。失敗した場合はエラー・コードが返されます。返り値の詳細は、"返り値とエラー・メッセージ" を参照してください。

注意　InterSystems IRIS では ZAUTHENTICATE の認証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。

5.3.3.1 GetCredentials エントリ・ポイント

ZAUTHENTICATE は GetCredentials エントリ・ポイントを含んでいます。エントリ・ポイントは代行認証がサービスに対して有効となるたびに、ユーザ名およびパスワードの入力が要求される前に呼び出されます。ユーザからユーザ名およびパスワードを得る代わりに、(アプリケーション開発者が作成した) 関数のコードがユーザ名およびパスワードを指定します。その後、返されたユーザ名およびパスワードは、ユーザが入力したかのように通常の方法にて認証されます。このメカニズムを使用することで、ユーザ名およびパスワードをエントリ・ポイント内さらには認証コード内で、プロセスの $roles に対して提供することができます。

このエントリ・ポイントより返されるユーザ名とパスワードは、アプリケーション開発者が選択する任意のメカニズムにより取得できます。これらは、グローバルから取得することも、外部の DLL または LDAP 呼び出しから取得することも、または単純にルーチン内で設定することもできます。さらに、アプリケーション開発者は、ターミナル接続やログイン・ページなどでユーザ名およびパスワードの入力を要求するコードを提供することもできます。

GetCredentials エントリ・ポイントの呼び出しがある場合、戻り値およびその他の要素によって次の動作が決まります。

- コードが Username および Password の値を設定し、成功ステータス ($$$OK) も返した場合は、以下のように動作します。
  - さらにユーザ名とパスワードの入力が求められることはありません。
  - 認証プロセスが続行されます。

  重要　アクセス・ポイントから $$$OK が返された場合、コードは Username および Password の値を設定する必要があります。それ以外の場合、ユーザはシステムへのアクセスを拒否され、エラーが監査ログに書き込まれます。
- エントリ・ポイントからエラー・ステータス $SYSTEM.Status.Error($$$GetCredentialsFailed) が返された場合は、通常のユーザ名とパスワードの入力が求められます。
- エントリ・ポイントからその他のエラー・ステータスが返された場合は、以下のように動作します。
  - ユーザはシステムへのアクセスを拒否されます。
  - エラーが監査ログに記録されます。

GetCredentials エントリ・ポイントの以下の例では、コードは各種サービスに対してさまざまな動作を実行します。

- **%Service_Console** では、ユーザに情報を要求せずに、プロセスのユーザ名とパスワードをそれぞれ _SYSTEM および SYS に設定します。
- **%Service_Bindings** では、ユーザにユーザ名とパスワードの入力を強制します。

- Web アプリケーションでは、使用中のアプリケーションが /csp/ サンプル・アプリケーションであるかどうかをチェックします。そうである場合は、ユーザ名とパスワードを AdminUser および Test に設定します。他の Web アプリケーションである場合は、アクセスはすべて拒否されます。
- その他のどのサービスの場合も、アクセスは拒否されます。

最後に、Error エントリ・ポイントは必要に応じてクリーンアップを行います。

このコードは以下のとおりです。

ObjectScript

```
GetCredentials(ServiceName,Namespace,Username,Password,Credentials) Public {

  // For console sessions, authenticate as _SYSTEM.
  If ServiceName="%Service_Console" {
    Set Username="_SYSTEM"
    Set Password="SYS"
    Quit $SYSTEM.Status.OK()
  }

  // For a web application, authenticate as AdminUser.
  If $isobject($get(%request)) {
    If %request.Application="/csp/samples/" {
      Set Username="AdminUser"
      Set Password="Test"
      Quit $System.Status.OK()
    }
  }

  // For bindings connections, use regular prompting.
  If ServiceName="%Service_Bindings" {
    Quit $SYSTEM.Status.Error($$$GetCredentialsFailed)
  }

  // For all other connections, deny access.
  Quit $SYSTEM.Status.Error($$$AccessDenied)
}
```

詳細は、ZAUTHENTICATE.mac の、このエントリ・ポイントのコメントを参照してください。

5.3.3.2 SendTwoFactorToken エントリ・ポイント

ZAUTHENTICATE は SendTwoFactorToken エントリ・ポイントを含んでいます。このエントリ・ポイントは 2 要素認証と共に使用します。これが定義されており、InterSystems IRIS インスタンスで 2 要素認証が有効な場合、インスタンスがユーザの携帯電話に送信するメッセージおよびトークンの形式の既定のシステム設定をオーバーライドできます。これによって、同じ InterSystems IRIS インスタンス上でもアプリケーションごとに異なるメッセージを使用できます。

このエントリ・ポイントの詳細と使用例は、サンプル ZAUTHENTICATE.mac でこのエントリ・ポイントを参照してください。

5.3.4 ロールと他のユーザ特性の値の設定

最初の認証が成功すると、ZAUTHENTICATE は認証されたユーザのロールと他の特性を確立できます。以降のログインでは、ユーザ・レコードのこれらの要素を更新できます。

これを行うために、ZAUTHENTICATE のコードにより Properties 配列の値が設定されます (Properties は、ZAUTHENTICATE への参照によって渡されます)。通常、設定する値のソースは、ZAUTHENTICATE が使用できるユーザ情報のリポジトリです。

5.3.4.1 ユーザのプロパティ

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
- Properties("Password") ー ユーザのパスワード
- Properties("Username") ー ユーザのユーザ名
- Properties("PhoneNumber") ー ユーザの携帯電話番号で、2 要素認証で使用
- Properties("PhoneProvider") ー ユーザの携帯電話のサービス・プロバイダで、2 要素認証で使用

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　プロパティの配列の各要素の値によって、それぞれの要素に関連する、認証対象のユーザのプロパティ値が設定されます。これらのプロパティのサブセットのみを使用したり、認証後にプロパティ値を操作したりすることはできません。

Comment

ZAUTHENTICATE で Properties("Comment") 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 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

ZAUTHENTICATE で Properties("FullName") 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 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

ZAUTHENTICATE 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) 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 (Properties("Namespace") に値が指定されないため)、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

ZAUTHENTICATE 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。

Roles

ZAUTHENTICATE 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・アカウントの Roles 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。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

ZAUTHENTICATE 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") に値を指定しない場合、ログインはプログラマ・モードでターミナル・セッションを開始します。

Username

ZAUTHENTICATE で Username プロパティを返す場合、Username の値は関数での処理の後にセキュリティ・データベースに書き込まれます。このため、ユーザがプロンプトで入力した値が変更される可能性があります。ZAUTHENTICATE で Username プロパティを返さない場合、プロパティの値は入力時にセキュリティ・データベースに書き込まれます。

ZAUTHENTICATE 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。

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。

PhoneNumber と PhoneProvider

これらは 2 要素認証と関連付けられたプロパティです。

ZAUTHENTICATE が Properties("PhoneNumber") と Properties("PhoneProvider") の値を設定すると、これらはユーザの携帯電話番号および携帯電話サービス・プロバイダとしてユーザの InterSystems IRIS データベースに書き込まれます。これらが呼び出しルーチンに渡されない場合、InterSystems IRIS データベースに書き込まれた電話番号とサービス・プロバイダは NULL 文字列です。したがって、代行認証で 2 要素認証を使用するには、これらの両方を提供する必要があります。

5.3.4.2 ユーザ情報のリポジトリ

ZAUTHENTICATE 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、ZAUTHENTICATE 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 ZAUTHENTICATE 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 ZAUTHENTICATE によってリセットされます。

5.3.5 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 $$$OK。ユーザ名とパスワードの組み合わせが正常に認証されたことを示します。
- 失敗 ー $SYSTEM.Status.Error($$$ERRORMESSAGE)。認証が失敗したことを示します。

ZAUTHENTICATE 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 Servicename 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

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。以下はその例です。

```
$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。

5.4 代行認証の設定

認証 (および、オプションで承認タスク) を実行する ZAUTHENTICATE 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. インスタンス全体の代行認証を有効にします。[**認証/Web セッション・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、[**代行認証を許可**] を選択し、[**保存**] をクリックします。

   インスタンスで代行認証を有効にすると、関連サービスの [**サービス編集**] ページと、関連アプリケーションの [**ウェブ・アプリケーション編集**] ページに [**代行**] チェック・ボックスが表示されます。

2. 必要に応じて、サービスとアプリケーションの代行認証を有効にします。

以下のサービスは認証代行をサポートしています。

- **%Service_Bindings**
- **%Service_CallIn**
- **%Service_ComPort**
- **%Service_Console**
- **%Service_Login**
- **%Service_Terminal**
- **%Service_Telnet**
- **%Service_WebGateway**

これらのサービスは、アクセス・モードによっていくつかのカテゴリに分類されます。

- ローカル・アクセス ー

  %Service_CallIn、**%Service_ComPort**、**%Service_Console**、**%Service_Login**、**%Service_Terminal**、**%Service_Telnet**

  ローカル接続で代行認証を使用するには、サービスの代行認証を有効にします。

- クライアント・サーバ・アクセス ー

  %Service_Bindings

  クライアント・サーバ接続で代行認証を使用するには、サービスの代行認証を有効にします。

- Web アクセス ー

  %Service_WebGateway

Web ベースの接続で代行認証を使用するには、Web アプリケーションの代行認証を有効にします。また、サービス **%Service_WebGateway** を有効にすることで、Web ゲートウェイでこれを有効にすることもできます。

5.5 代行認証成功後の注意事項

ユーザが認証された後の重要な項目は以下のとおりです。

- システムの状態
- パスワードの変更

5.5.1 システムの状態

代行認証を使用して最初に認証されるユーザは、“代行ユーザ” というタイプで **[ユーザ]** ページ (**[システム管理]** > **[セキュリティ]** > **[ユーザ]**) のユーザ・テーブルに表示されます。システム管理者が管理ポータル (またはその他の InterSystems IRIS のネイティブ機能) を使用して明示的にユーザを作成した場合、そのユーザのタイプは “InterSystems IRIS パスワード・ユーザ” になります。ユーザが代行認証を使用してログインを試行し、認証が正常に行われる場合でも、そのユーザが既に (代行ユーザではなく) InterSystems IRIS ユーザとして存在することを InterSystems IRIS が検出すると、ログインは失敗します。

注釈　シャード・クラスタでシャード操作を実行するには、代行ユーザがシャード・クラスタの各ノードで内部シャード接続以外の方法によって事前に認証されている必要があります。シャードの詳細は、″シャーディングによるデータ量に応じた InterSystems IRIS の水平方向の拡張″ を参照してください。

5.5.2 パスワードの変更

ZAUTHENTICATE ルーチンには、エントリ・ポイント ChangePassword も含まれており、そこにはユーザのパスワードを変更するコードが含まれています。このエントリ・ポイントのシグニチャは以下のとおりです。

ObjectScript

```
ChangePassword(Username,NewPassword,OldPassword,Status) Public {}
```

各項目の内容は次のとおりです。

- Username は、パスワードを変更するユーザを指定する文字列です。
- NewPassword は、ユーザのパスワードの新しい値を指定する文字列です。
- OldPassword は、ユーザのパスワードの古い値を指定する文字列です。
- Status (参照によって渡される) は、InterSystems IRIS の状態値を受信します。受信した値では、パスワード変更が成功したことを示すか、ルーチンの失敗の原因となったエラーを指定します。

5.6 代行認証または他のメカニズムでの LDAP の使用

カスタム認証システムの一部として (つまり、InterSystems IRIS の代行認証機能で) LDAP を使用することもできます。そのためには、ZAUTHENTICATE ルーチンのカスタム認証コードの一部として **%SYS.LDAP** クラスの呼び出しを使用します。

インターシステムズでは、これらの呼び出し方法を示すサンプル・ルーチン LDAP.mac を提供しています。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)。

さらに、LDAP に対して認証する必要がある場合、または別のメカニズムによって認証情報を収集した後にインスタンス認証を使用する必要がある場合は、それらの認証情報で $SYSTEM.Security.Login を呼び出してユーザを認証します。

6

2 要素認証

使用中の認証メカニズムの他に、InterSystems IRIS は 2 要素認証の使用をサポートしています。つまり、インターシステムズの認証では、エンドユーザが 2 つの別々のエレメントつまり“要素”を持つことを要求できます。エンドユーザの観点では、最初の要素はユーザが知っているもの (パスワードなど)、2 番目の要素はユーザが持っているもの (スマートフォンなど) です。InterSystems IRIS は、以下の 2 つのメカニズムのいずれかを使用してエンドユーザの 2 要素認証を実行します。

- SMS テキスト認証 ー InterSystems IRIS はセキュリティ・コードをエンドユーザの電話に SMS で送ります。エンドユーザは、指示に従ってそのコードを入力します。
- タイムベース・ワンタイム・パスワード (TOTP) ー エンドユーザは最初に InterSystems IRIS から秘密鍵を受け取ります。この鍵は、InterSystems IRIS とエンドユーザのアプリケーション (携帯電話のアプリケーションなど) または物理的な認証デバイスとの間の共有秘密です。両者が、この鍵およびその他の情報を使用して、検証コードとして作用し、InterSystems IRIS の指示に従ってエンドユーザが入力する TOTP を生成します。TOTP は 60 秒後に期限切れになり、エンドユーザはこれを 1 回しか使用できません。これがタイムベースおよびワンタイムと呼ばれている理由です。

このセクションでは、以下のトピックについて説明します。

- 2 要素認証の設定の概要
- サーバの 2 要素認証の構成
- サービスに対する 2 要素認証の有効化または無効化
- 2 要素認証向けの Web アプリケーションの構成
- 2 要素認証向けのエンドユーザの構成
- 2 要素認証向けのバインディング・クライアントの構成

6.1 2 要素認証の設定の概要

2 要素認証の設定の主な手順は以下のとおりです。

1. インスタンス全体に対する 2 要素認証の有効化および構成を行います。インスタンスを構成して SMS テキスト認証、TOTP 認証、またはその両方を使用できます。TOTP 認証の詳細は、″2 要素の TOTP の概要″ を参照してください。
2. SMS テキスト認証の場合、必要に応じて携帯電話サービス・プロバイダの構成を行います。これには以下が含まれます。

- ある携帯電話サービス・プロバイダが必要であるのに、既定のプロバイダのリストに含まれていない場合、このサービス・プロバイダを追加する。
- 既存のプロバイダに対し、必要に応じて構成情報を変更する (既定または追加)。

3. 以下のサービスを適切に構成します。
   - **%Service_Bindings** ― サービスに対して 2 要素認証を有効化し、次の手順に進みます。
   - **%Service_Console** および **%Service_Terminal** ― 単にサービスに対して 2 要素認証を有効化します。必要な手順はこれですべてです。
   - **%Service_WebGateway** ― **%Service_WebGateway** の 2 要素認証を一元的に有効にする方法はありません。次の手順に進みます。

   各サービスに対していずれかまたは両方の認証のタイプを有効にできます。サービスの詳細は、"サービス" を参照してください。
4. クライアント・サーバ・アプリケーションおよび Web アプリケーションを適切に構成します。
   a. クライアント・サーバ・アプリケーション (**%Service_Bindings** を使用するアプリケーション) の場合、2 要素認証をサポートするクライアント・アプリケーションに適切な呼び出しを追加します。これは、使用中のクライアント側のコンポーネント (Java、JDBC、.NET など) に応じて異なるプログラミング・タスクです。

      重要　2 要素認証は、人のエンドユーザからの応答をリアルタイムで受け取るように設計されています。1 つのセッションが実際には複数の連続セッションから構成されているとエンドユーザが見なしている場合、2 番目の要素を繰り返し要求することは、予期しない困難なユーザ・エクスペリエンスをもたらすことになる場合があります。クライアント・サーバ・アプリケーションでは、基礎プロトコルによって、クライアントが接続の確立、切断、再確立を繰り返し強いられることがよくあります。この種のアプリケーションでは、このような作業のために、2 要素認証の使用は不適切になります。

   b. Web アプリケーション (**%Service_WebGateway** を使用するアプリケーション) については、2 要素認証をサポートするように各アプリケーションを構成します。

   注釈　Windows では **%Service_Console** サービスを使用し、他のオペレーティング・システムでは **%Service_Terminal** サービスを使用する InterSystems IRIS ターミナルの場合、サーバ側の設定の他に必要な構成はありません。InterSystems IRIS は、これらのシステムにおいてプロンプトを制御するため、2 要素認証プロンプトを使用して (認証メカニズムに関係なく) 標準プロンプトに従い、エンドユーザ入力をこれに応じて処理するだけです。
5. 代行認証を使用する場合は、必要に応じて ZAUTHENTICATE.mac ルーチンを変更します。詳細は "代行認証の使用法" を参照してください。
6. 各エンドユーザを構成し、SMS テキスト認証または TOTP 認証を有効にします。エンドユーザは、両方のメカニズムを使用するように構成できますが、同時に両方のメカニズムを有効にすることはできません。

6.1.1 2 要素の TOTP の概要

タイムベース・ワンタイム・パスワード (TOTP) 認証を使用した 2 要素認証は以下のように動作します。

1. TOTP を生成する認証デバイスまたはアプリケーションを選択し、そのデバイスまたはアプリケーションを提供するか、ユーザがそのデバイスまたはアプリケーションを所持していることを確認します。
2. 2 要素の TOTP 認証向けにエンドユーザを構成すると、システムは秘密鍵を生成します。この鍵は、Base 32 でエンコードされてランダム化されたビット文字列として表示されます。InterSystems IRIS とエンドユーザはこの秘密鍵を

共有します(これが共有秘密と呼ばれる理由です)。InterSystems IRIS とエンドユーザの両方の認証デバイスまたはアプリケーションはこの秘密鍵を使用して、検証コードとして作用する TOTP 自体を生成します。**[検証コード]** フィールドまたはプロンプトにエンドユーザが入力する TOTP は 6 桁の文字列で、定期的 (既定では 30 秒毎) に新しいものが生成されます。

3. ログイン時、エンドユーザが InterSystems IRIS にパスワードを入力した後、InterSystems IRIS は追加で TOTP の入力を求めます。エンドユーザが TOTP を入力すると、ログイン・プロセスが完了します。

エンドユーザは、以下のいくつかの方法で InterSystems IRIS から秘密鍵を取得できます。

- 2 要素の TOTP 認証をサポートするようにエンドユーザのアカウントを構成するとき、エンドユーザの **[ユーザ編集]** ページにエンドユーザの秘密鍵が発行者の名前およびエンドユーザのアカウント名と共に表示されます。上記の情報をすべて含む QR コードも表示されます (QR コードは下の写真のようなマシンが読むことができるコードです)。ここでエンドユーザは、コードをスキャンするか情報を手入力することによって、情報を認証デバイスまたはアプリケーションに入力できます。
- Web アプリケーションまたはターミナル・セッションにログインするとき (**%Service_Console** または **%Service_Terminal** を使用) エンドユーザに秘密鍵を表示することを選択する場合、**[ユーザ編集]** ページの **[次回のログイン時にタイムベース・ワンタイム・パスワードの QR コードを表示する]** フィールドを選択することによってこの動作を有効にできます。これを行うと、ターミナル・セッションはエンドユーザの発行者、アカウント、および秘密鍵を表示します。Web アプリケーションは、エンドユーザの発行者、アカウント、および秘密鍵を QR コードと共に表示します。ここで、エンドユーザはコードをスキャンするか、情報を手入力できます。

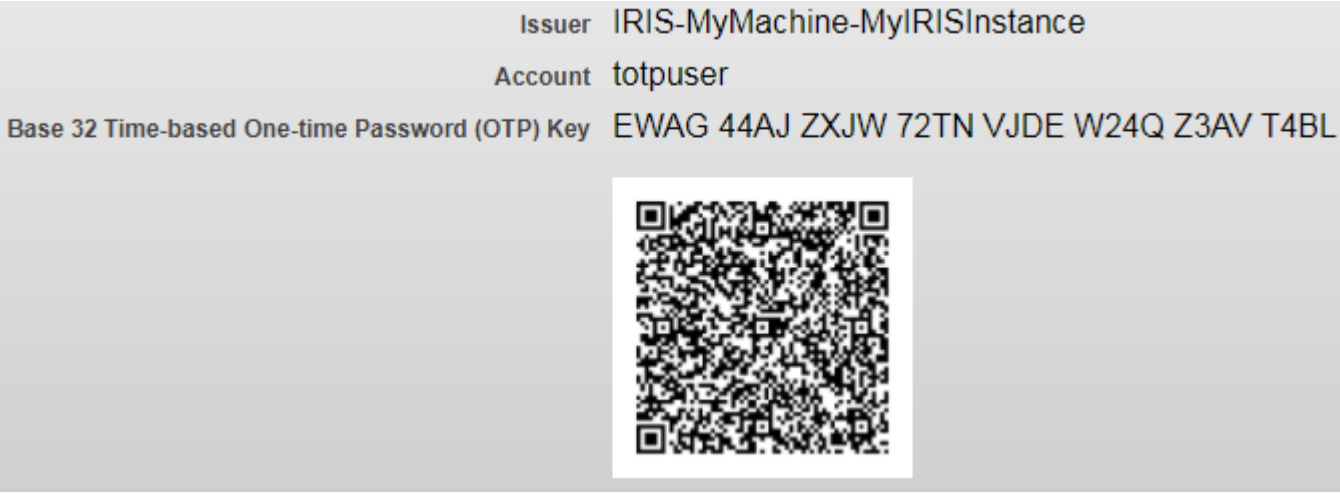
  重要　　このオプションはお勧めしません。詳細は、以下の注意事項を参照してください。

注意　　以下は、2 要素の TOTP 認証を使用するときの重要なセキュリティ上の問題です。

- 安全でない環境では、秘密鍵や QR コードを転送しないでください。安全なネットワークでも帯域外転送をお勧めします(秘密鍵は InterSystems IRIS または InterSystems IRIS アプリケーションにログインする手段をエンドユーザに提供します。ユーザまたはエンドユーザが秘密鍵の安全性を確保できない場合、攻撃者がアクセスできるようになる可能性があり、秘密鍵がセキュリティの役に立たなくなります)。
- 組織に対して 2 要素の TOTP 認証を構成するとき、各エンドユーザに秘密鍵を直接渡すか、電話で伝えるか、または管理者が立ち会っている状況でエンドユーザに QR コードをスキャンさせることを強くお勧めします。こうすることによって、秘密鍵を取得する個人を認証する機会がもたらされます。

  ネットワークを介して秘密鍵を渡すと、漏えいの可能性が高くなります。これには、Web アプリケーション、コンソール、またはターミナルに最初にログインするときにエンドユーザに秘密鍵を表示することが含まれます。また、Web アプリケーションに最初にログインするときにエンドユーザに QR コードを表示することも含まれます。

図 6-1: TOTP 発行者、アカウント、鍵、および QR コード



注釈　2 要素の TOTP 認証を使用していて、QR コードを生成する場合、InterSystems IRIS サーバで Java 1.7 以降を実行している必要があります。Java なしでも InterSystems IRIS で 2 要素の TOTP 認証を使用できますが、認証デバイスまたはアプリケーションでエンドユーザが発行者、アカウント、および鍵の値を手入力する必要があります。

6.2 サーバの 2 要素認証の構成

InterSystems IRIS サーバの 2 要素認証を構成する手順は以下のとおりです。

1. インスタンス全体に対する 2 要素認証の有効化および構成を行います。インスタンスを構成して SMS テキスト認証、TOTP 認証、またはその両方を使用できます。
2. SMS テキスト認証の場合、必要に応じて携帯電話サービス・プロバイダの構成を行います。これには以下が含まれます。
   - ある携帯電話サービス・プロバイダが必要であるのに、既定のプロバイダのリストに含まれていない場合、このサービス・プロバイダを追加する。
   - 既存のプロバイダに対し、必要に応じて構成情報を変更する (既定または追加)。

6.2.1 インスタンスに対する 2 要素認証設定の有効化および構成

InterSystems IRIS インスタンス (サーバ) 向けに 2 要素認証を設定すると、以下の 1 つまたは両方を有効にできます。

- **[2 要素のタイムベース・ワンタイム・パスワード認証]** (TOTP 認証)
- **[2 要素の SMS テキスト認証]**

いずれかの形式の 2 要素認証を有効にするための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、**[認証/Web セッション・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]**) に移動します。
2. 2 要素の TOTP 認証を有効にするには、**[認証/Web セッション・オプション]** ページで、**[2 要素のタイムベース・ワンタイム・パスワード認証を許可]** チェック・ボックスにチェックを付けます。**[2 要素のタイムベース・ワンタイム・パスワードの発行者]** フィールドが表示されます。ここにこの InterSystems IRIS インスタンスを識別する文字列を入力します。

3. 2 要素の SMS テキスト認証を有効にするには、**[認証/Web セッション・オプション]** ページで、**[2 要素の SMS テキスト認証を許可]** チェック・ボックスにチェックを付けます。以下の各フィールドが表示されます。
   - **[2要素タイムアウト (秒)]** – 1 回限りのセキュリティ・トークンを入力する際のタイムアウト秒数 (オプション)。
   - **[SMTPサーバのDNS名]** – SMS テキスト・メッセージの送信にこの InterSystems IRIS インスタンスが使用している SMTP (Simple Mail Transfer Protocol) サーバの DNS (Domain Name Service) 名 (**smtp.example.com** など) (必須)。
   - **[From (アドレス)]** – メッセージの“From”フィールドに表示されるアドレス (必須)。
   - **[SMTPユーザ名]** – (SMTP サーバが要求する場合) SMTP 認証のユーザ名 (オプション)。
   - **[SMTPパスワード]** および **[SMTPパスワード (確認)]** – SMTP 認証に対して入力および確認されるパスワード (オプション)。
4. **[保存]** をクリックします。
5. インスタンスが SMS テキスト認証をサポートしている場合、必要に応じて携帯電話サービス・プロバイダを構成します。これらの手順について、以下のセクションで説明します。

インスタンス自体に対してこのプロセスを完了した後、インスタンスのサービス、Web アプリケーション、クライアント・サーバ・アプリケーションなど、その他の構成を実行する必要があります。また、インスタンスのユーザを構成する必要があります。″2 要素認証の設定の概要″ にこれに関する一般的な指示が提示されています。

6.2.2 携帯電話サービス・プロバイダの構成

携帯電話サービス・プロバイダの構成に関連する項目は以下のとおりです。

- 携帯電話サービス・プロバイダの作成または編集
- 携帯電話サービス・プロバイダの削除
- 事前定義された携帯電話サービス・プロバイダ

6.2.2.1 携帯電話サービス・プロバイダの作成または編集

携帯電話サービス・プロバイダを作成または編集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、**[携帯電話サービスプロバイダ]** ページ (**[システム管理]** > **[セキュリティ]** > **[携帯電話]**) に移動します。
   - プロバイダを新規作成するには、**[新規プロバイダ]** をクリックします。
   - 既存のプロバイダを編集するには、プロバイダのテーブルにあるプロバイダの行で **[編集]** をクリックします。

   選択した携帯電話サービス・プロバイダの **[電話プロバイダ編集]** ページが表示されます。
2. **[電話プロバイダ編集]** ページで、以下のフィールドそれぞれの値を入力または変更します。
   - **[サービスプロバイダ]** – 携帯電話サービス・プロバイダの名前 (通常は会社名)。
   - **[SMSゲートウェイ]** – SMS (short message service) メッセージを送信するために携帯電話サービス・プロバイダが使用するサーバのアドレス。

6.2.2.2 携帯電話サービス・プロバイダの削除

携帯電話サービス・プロバイダ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、[**携帯電話サービスプロバイダ**] ページ ([**システム管理**] > [**セキュリティ**] > [**携帯電話**]) に移動します。
2. [**携帯電話サービスプロバイダ**] ページにあるプロバイダの行で [**削除**] をクリックします。
3. 削除の確認を求めるプロンプトが表示されたら、[OK] をクリックします。

6.2.2.3 事前定義された携帯電話サービス・プロバイダ

InterSystems IRIS には、事前定義された携帯電話サービス・プロバイダのリストが付属しており、それぞれのプロバイダには SMS (short message service) ゲートウェイが事前に設定されています。事前定義されている携帯電話サービス・プロバイダは以下のとおりです。

- AT&T Wireless — **txt.att.net**
- Alltel — **message.alltel.com**
- Cellular One — **mobile.celloneusa.com**
- Nextel — **messaging.nextel.com**
- Sprint PCS — **messaging.sprintpcs.com**
- T-Mobile — **tmomail.net**
- Verizon — **vtext.com**

6.3 サービスに対する 2 要素認証の有効化または無効化

重要　**%Service_WebGateway** に対して、2 要素認証を一元的に有効または無効にすることができる場所はありません。"2 要素認証向けの Web アプリケーションの構成" の説明に従って、各アプリケーションに対して 2 要素認証を有効化または無効化します。

%Service_Bindings、**%Service_Console**、および **%Service _Terminal** に対して 2 要素認証を有効化または無効化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、[**サービス**] ページ ([**システム管理**] > [**セキュリティ**] > [**サービス**]) に移動します。
2. [**サービス**] ページで、いずれかの形式の 2 要素認証を有効にするサービスの名前をクリックします。選択したサービスの [**サービス編集**] ページが表示されます。
3. そのサービスの [**サービス編集**] ページで、[**2 要素の SMS**] チェック・ボックス、[**2 要素のタイムベース・ワンタイム・パスワード**] チェック・ボックス、またはその両方にチェックを付けるか、またはチェックを外します。これらの各チェック・ボックスは、そのインスタンスに対して 2 要素認証が有効になっている場合にのみ表示されます。
4. [**保存**] をクリックします。

6.4 2 要素認証向けの Web アプリケーションの構成

インスタンスに対して 2 要素認証を有効にしたら、これを使用するすべての Web アプリケーションでもこれを有効にする必要があります。アプリケーションでこれを有効に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、[**ウェブ・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**] > [**ウェブ・アプリケーション**]) に移動します。

2. [**Web アプリケーション**] ページで、2 要素認証を有効にするアプリケーションの名前をクリックし、そのアプリケーションの [**編集**] ページを表示します。
3. [**編集**] ページの [**セキュリティ設定**] セクションで、[**2 要素の SMS**] チェック・ボックス、[**2 要素のタイムベース・ワンタイム・パスワード**] チェック・ボックス、またはその両方にチェックを付けるか、またはチェックを外します。これらの各チェック・ボックスは、そのインスタンスに対して 2 要素認証が有効になっている場合にのみ表示されます。

注釈　Web アプリケーションでは 2 要素認証と Web サービスの両方を同時にサポートすることはできません。

6.5 2 要素認証向けのエンドユーザの構成

2 要素認証向けに 1 回限りのセキュリティ・トークンを受信するようにエンドユーザを構成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. 既存のユーザの場合は、編集するユーザの名前をクリックします。新規ユーザの場合は、[**新規ユーザ作成**] をクリックしてユーザの作成を始めます (新規ユーザの作成の詳細は、"新規ユーザの作成" を参照)。これらのアクションのどちらを実行しても、そのエンドユーザの [**編集**] ページが表示されます。
3. [**ユーザ編集**] ページで、[**SMS テキスト有効**] または [**タイムベース・ワンタイム・パスワード有効**] を適切に選択します。
4. [**SMS テキスト**] を選択した場合、以下のフィールドをすべて入力する必要があります。
   - [**携帯電話サービスプロバイダ**] – ユーザに携帯電話サービスを提供する会社。リストからプロバイダを選択します。プロバイダがリストに表示されていない場合は、[**新規プロバイダ**] をクリックして、InterSystems IRIS インスタンスに新規プロバイダを追加します ([**新しいプロバイダーを作成**] をクリックすると、[**新しい携帯電話プロバイダーを作成**] ウィンドウが表示されます。このウィンドウには、[**サービスプロバイダ**] フィールドおよび [**SMSゲートウェイ**] フィールドがあります。これらの目的は、"携帯電話サービス・プロバイダの作成または編集" で説明した目的と同じです)。
   - [**携帯電話番号**] – ユーザの携帯電話番号。これは 2 つ目の要素で、1 回限りのセキュリティ・トークンを含むテキスト・メッセージをユーザが受信する電話番号です。
5. [**タイムベース・ワンタイム・パスワード有効**] を選択した場合、ページに以下のフィールドおよび情報が表示されます。
   - [**次回のログイン時にタイムベース・ワンタイム・パスワードの QR コードを表示する**] – ユーザが次回ログインしたときに QR コードを表示するかどうか。選択されている場合、InterSystems IRIS は、次回のログイン時に QR コードを表示し、これをスキャンして認証デバイスまたはアプリケーションに取り込むようユーザに求めます。さらに、トークンを表示して提供し、認証プロセスの完了を促します。既定では、このオプションは選択されていません。このオプションは使用しないことをお勧めします。
   - [**新しいタイムベース・ワンタイム・パスワードの鍵を生成する**] – エンドユーザの新しい共有秘密と新しい QR コードの両方を作成して表示します。

     重要　ユーザに新しいタイムベース・ワンタイム・パスワードの鍵を生成すると、ユーザの認証アプリケーションの現在の鍵は機能しなくなります。ログインする前に、ユーザは、QR コードをスキャンするか、または手入力して新しい鍵を認証アプリケーションに入力する必要があります(これは既存のセッションには影響しません)。
   - [**発行者**] – インスタンスの 2 要素の TOTP 認証を構成するときに設定した InterSystems IRIS インスタンスの識別子。
   - [**アカウント**] – InterSystems IRIS アカウントの識別子。これはアカウントのユーザ名です。

- [Base32 のタイムベース・ワンタイム・パスワード (OTP) の鍵] – エンドユーザが認証デバイスまたはアプリケーションに入力する秘密鍵。
- [QR コード] – 発行者、アカウント、および秘密鍵の値が格納されているスキャン可能なコード。

6. [保存] をクリックして、このユーザのこれらの値を保存します。

サービスが 2 要素認証を使用していて、エンドユーザが 2 要素認証を有効にしている場合、認証には以下が必要です。

- SMS テキスト認証では、テキスト・メッセージを受信できる携帯電話。
- TOTP 認証では、検証コードを生成できるアプリケーションまたは認証デバイス。

これらがないと、エンドユーザは認証できません。

- SMS テキスト認証では、エンドユーザは携帯電話を所持していて、その電話でテキスト・メッセージを受信できる必要があります。これは、1 回限りのセキュリティ・トークンを含むテキスト・メッセージを SMS テキストとしてユーザが受信する電話番号です。
- TOTP 認証では、ユーザは、QR コードをスキャンできるか、または TOTP (検証コードとして作用) の生成に必要な秘密鍵などの情報を受け入れることができる、認証デバイスまたはアプリケーションを所持している必要があります。

6.6 2 要素認証向けのバインディング・クライアントの構成

クライアント・サーバ接続は **%Service_Bindings** を使用します。この接続で 2 要素認証を使用するために必要なコードは、プログラミング言語によって異なります (コンソール、ターミナル、および Web アプリケーションではクライアント側の構成は必要ありません)。サポートされている言語には以下のものがあります。

- Java および JDBC
- .NET
- ODBC

クライアント側のコードで実行される処理は以下の 3 つです。

1. InterSystems IRIS サーバへの接続を確立した後、2 要素認証がサーバで有効になっているかどうかを確認します。通常、この確認にはクライアントの接続オブジェクトのメソッドが使用されます。
2. ユーザから 1 回限りのセキュリティ・トークンを取得します。この処理では一般的に、InterSystems IRIS と特別な関係のないユーザ・インタフェース・コードが使用されます。
3. 1 回限りのセキュリティ・トークンを InterSystems IRIS サーバに提供します。この処理でも一般的に、接続オブジェクトのメソッドが使用されます。

注釈　ユーザが **%Service_Bindings** を使用してログインすると、InterSystems IRIS はスキャンする QR コードを表示しません。ユーザは、認証デバイスまたはアプリケーションを事前に設定しておく必要があります。

重要　**%Service_Bindings** を使用して InterSystems IRIS サーバに接続するスタジオは、2 要素認証をサポートしていません。

6.6.1 Java および JDBC

Java では、2 要素認証のサポートは、以下のように IRISConnection クラスのメソッドを 2 つ使用します。

. `public boolean isTwoFactorEnabled() throws Exception`

  このメソッドは、サーバで 2 要素認証が有効になっているかどうか確認します。このメソッドはブーリアン値を返します。`true` は、2 要素認証が有効であることを意味します。

. `public void sendTwoFactorToken(String token) throws Exception`

  このメソッドは、1 回限りのセキュリティ・トークンをサーバに提供します。このメソッドは 1 つの引数 token をとります。これは、ユーザが受信した 1 回限りのセキュリティ・トークンです。

以下の例では、conn という接続のインスタンスを使用します。

1. この例では、このインスタンスのメソッドを使用して、2 要素認証が有効になっているかどうか確認します。
2. この例では、サーバにトークンを提供しようとし、これに失敗するとエラー処理が行われます。

重要　2 要素認証がサーバで有効になっていて、クライアント・コードが 2 要素認証呼び出しを実装していない場合、サーバはクライアントとの接続を切断します。

```
// Given a connection called "conn"
if (conn.isTwoFactorEnabled()) {
  // Prompt the user for the two-factor authentication token.
  // Store the token in the "token" variable.
  try {
    conn.sendTwoFactorToken(token);
  }
  catch (Exception ex) {
    // Process the error from a invalid authentication token here.
  }
}
```

6.6.2 .NET

.NET の場合、InterSystems IRIS は、Managed Provider および ADO.NET との 2 要素認証を使用した接続をサポートします。2 要素認証のサポートは、以下のように tcp_conn クラスのメソッドを 2 つ使用します。

. `bool IRISConnection.isTwoFactorEnabledOpen()`

  このメソッドは、InterSystems IRIS サーバへの接続を開き、2 要素認証がこの接続で有効になっているかどうかを確認します。このメソッドはブーリアン値を返します。`true` は、2 要素認証が有効であることを意味します。

. `void IRISConnection.sendTwoFactorToken(token)`

  このメソッドは、1 回限りのセキュリティ・トークンをサーバに提供します。返り値はありません。このメソッドは 1 つの引数 token をとります。これは、ユーザが受信した 1 回限りのセキュリティ・トークンです。トークンの問題 (有効でないなど) と接続の問題のいずれかがある場合、このメソッドは例外をスローします。

重要　クライアント・アプリケーションは、IRISConnection.Open を呼び出す代わりに isTwoFactorEnabledOpen を呼び出します。isTwoFactorEnabledOpen メソッドは、続けて sendTwoFactorToken を呼び出す必要があります。

また、2 要素認証がサーバで有効になっていて、クライアント・コードが 2 要素認証呼び出しを実装していない場合、サーバはクライアントとの接続を切断します。

以下の例では、conn という接続のインスタンスを使用します。

1. この例では、このインスタンスのメソッドを使用して、2 要素認証が有効になっているかどうか確認します。
2. この例では、サーバにトークンを提供しようとし、これに失敗するとエラー処理が行われます。

```
// Given a connection called "conn"
try {
  if (conn.isTwoFactorEnabledOpen()) {
    // Prompt the user for the two-factor authentication token.
    // Store the token in the "token" variable.
    conn.sendTwoFactorToken(token);
  }
}
catch (Exception ex) {
  // Process exception
}
```

6.6.3 ODBC

ODBC では、2 要素認証のサポートは、ODBC の標準的な関数呼び出しを 2 つ使用します (″Microsoft ODBC API リファレンス″ を参照)。

. `SQLRETURN rc = SQLGetConnectAttr(conn, 1002, &attr, sizeof(attr), &stringLengthPtr);`

   Microsoft ODBC API の一部である SQLGetConnectAttr 関数は、指定された接続属性の現在値を返します。InterSystems ODBC クライアントは、この関数を使用して、サーバが 2 要素認証をサポートしているかどうかを判定します。最初の引数の値は、クライアントからサーバへの接続のハンドルです。2 つ目の引数の値は、`1002` で、これは 2 要素認証がサポートされているかどうかを指定する ODBC 属性です。これ以降の引数の値は、属性 1002 の値を含む文字列および関係のある変数のサイズ用です。

. `SQLRETURN rc = SQLSetConnectAttr(conn, 1002, securityToken, SQL_NTS);`

   同じく Microsoft ODBC API の一部である SQLSetConnectAttr 関数は、指定された接続属性の値を設定します。InterSystems ODBC クライアントは、この関数を使用して、2 要素認証トークンの値をサーバに送信します。4 つの引数の値は、それぞれ以下のとおりです。

   - クライアントからサーバへの接続。
   - `1002`。2 要素認証がサポートされているかどうかを指定する ODBC 属性です。
   - 1 回限りのセキュリティ・トークンの値。
   - `SQLNTS`。1 回限りのセキュリティ・トークンが文字列に格納されていることを示します。

重要　2 要素認証がサーバで有効になっていて、クライアント・コードが 2 要素認証呼び出しを実装していない場合、サーバはクライアントとの接続を切断します。

以下の例では、conn という接続のインスタンスを使用します。

1. この例では、SQLGetConnectAttr を使用して、2 要素認証が有効になっているかどうか確認します。

2. この例では、SQLSetConnectAttr 呼び出しによってサーバにトークンを提供しようとし、これに失敗するとエラー処理が行われます。SQLSetConnectAttr が失敗すると、サーバは接続を切断します。そのため、再び認証を行うには、その前に接続を再確立する必要があります。

```
// Given a connection called "conn"
SQLINTEGER stringLengthPtr;
SQLINTEGER attr;
SQLRETURN rc = SQLGetConnectAttr(conn, 1002, &attr, sizeof(attr), &stringLengthPtr);
if attr {
  // Prompt the user for the two-factor authentication token.
  wstring token;
  SQLRETURN rc = SQLSetConnectAttr(conn, 1002, token, SQL_NTS);
    if !rc {
      // Process the error from a invalid authentication token.
    }
}
```

7
サービス

InterSystems IRIS® インスタンスにユーザ、アプリケーション、さらには他の InterSystems IRIS インスタンスから接続するには、さまざまな経路があります。これらの経路はインターシステムズのサービスで管理され、これらのサービスは InterSystems IRIS への接続の監視役として機能します。インターシステムズのサービスは、ユーザおよびコンピュータといったエンティティから InterSystems IRIS に接続するための主要な手段なので、セキュリティ管理においてこれらのサービスの管理が不可欠な要素となります。

7.1 使用可能なサービス

[サービス] ページ (**[システム管理]** > **[セキュリティ]** > **[サービス]**) には、InterSystems IRIS が提供するサービスのリストが表示されます。

サービスは、以下の 2 つのグループに分けて考えることができます。

- リソース・ベース・サービス ー ユーザが InterSystems IRIS にアクセスできるようにするサービスです。この種類のサービスでは、InterSystems セキュリティで認証および承認のインフラストラクチャが必要です。したがって、これらのサービスはリソースに関連付けられ、さまざまな認証メカニズムを利用します。
- 基本サービス ー InterSystems IRIS サーバと InterSystems IRIS アプリケーション間の接続を提供するサービスです。これらのサービスには関連付けられたリソースがありません。したがって、これらのサービスが提供するセキュリティ機能は、有効にするか無効にするかのみを設定できる基本的なものにとどまります。これらを有効または無効にすることで、すべての形式のアクセスを制御できます。

以下のリストは、使用可能なサービスとその制御対象、および各サービスの種類をまとめたものです。

- **%Service_Bindings** ー SQL またはオブジェクト、スタジオの使用 (リソース・ベース)
- **%Service_CacheDirect** ー 他のインターシステムズ社製品に接続するための独自のメカニズム (リソース・ベース)
- **%Service_CallIn** ー コールイン・インタフェース (リソース・ベース)
- **%Service_ComPort** ー システムに接続する COMM ポート (リソース・ベース)
- **%Service_Console** ー Windows コンソールからのローカル・ターミナル (macOS、UNIX®、および Linux で使用する **%Service_Terminal** に相当) (リソース・ベース)
- **%Service_DataCheck** ー DataCheck ユーティリティ (基本)
- **%Service_DocDB** ー ドキュメント・データベース・アプリケーション (リソース・ベース)
- **%Service_ECP** ー エンタープライズ・キャッシュ・プロトコル (ECP) (基本)

- **%Service_Login** ― $SYSTEM.Security.Login の使用 (リソース・ベース)
- **%Service_Mirror** ― InterSystems IRIS のデータベース・ミラーリング (基本)
- **%Service_Monitor** ― SNMP およびリモート・モニタ・コマンド (基本)
- **%Service_Shadow** ― シャドウの宛先からこのインスタンスへのアクセス (既存の構成のみで使用) (基本)
- **%Service_Sharding** ― このインスタンスにシャード・サーバとしてアクセス (基本)
- **%Service_Telnet** ― Windows サーバおよびリモート Windows ターミナル・セッション (リソース・ベース)
- **%Service_Terminal** ― macOS、UNIX®、および Linux の各コンソールからのターミナル (Windows で使用する **%Service_Console** に相当) (リソース・ベース)
- **%Service_WebGateway** ― Web アプリケーション・ページ (リソース・ベース)
- **%Service_Weblink** ― レガシ・サービスとして使用できる WebLink (基本)

サービスのテーブルには、各サービス・プロパティの列があります。

7.1.1 個々のサービスに関するメモ

7.1.1.1 %Service_Bindings

%Service_Bindings サービスには、アクセスを管理するリソースとして **%Service_Native** リソース、**%Service_Object** リソース、**%Service_SQL** リソースの 3 つがあります。ユーザが認証されると、データに対して Native SDK を使用したアクセス、オブジェクトとしてのアクセス、または SQL を使用したアクセスのいずれがそのユーザに可能かを、これらのリソースで制御します。(ユーザがデータに対するテーブル・レベルの SQL 特権を持っている場合は、接続している間、認証されたユーザに **%Service_SQL:Use** 特権が自動的に与えられます)。

また、このサービスはスタジオへのアクセスも制御します。スタジオとセキュリティの詳細は、"セキュリティ" を参照してください。

7.1.1.2 %Service_Console および %Service_Terminal

これら 2 つのサービスは、どちらも InterSystems IRIS に対するコンソール形式またはターミナル形式のアクセスを提供します。これらのサービスにより、Windows システムと非 Windows システムの両方で類似の機能が実現します。Windows に対しては **%Service_Console** で、UNIX®、Linux、および Mac に対しては **%Service_Terminal** でこの機能が提供されます。

注意　ターミナルまたはコンソールへのアクセスは、インターシステムズのセキュリティの中でも機密性に最も深くかかわる機能の 1 つです。攻撃者がこれらのいずれかの方法で InterSystems IRIS にアクセスできると、機密性の高いデータが読み取られたり、破壊されたりする恐れがあります。

7.1.1.3 %Service_DataCheck

このサービスは、DataCheck ユーティリティの使用を管理します。このユーティリティは、2 つのシステムにあるデータの状態を比較するメカニズムを提供します。詳細は、"複数のシステムでのデータ整合性" を参照してください。また、セキュリティの問題については、特に "DataCheck サービスの有効化" を参照してください。

7.1.1.4 %Service_ECP

リソースは ECP の使用を制御しません。ユーザがサービスを有効化または無効化します (この理由から、ECP は "基本サービス" と呼ばれています)。そのため、分散キャッシュ・クラスタなど、ECP 構成のすべてのインスタンスが、InterSystems IRIS の安全な境界の内部に存在する必要があります。

ECP ベースの構成内で特権が機能する仕組みの詳細は、"分散キャッシュ・クラスタのセキュリティ" を参照してください。

7.1.1.5 %Service_Login

このサービスは、%SYSTEM.Security クラスの Login メソッドを明示的に呼び出す機能を制御します。このメソッドの呼び出し形式は以下のとおりです。

ObjectScript

```
Set Success = $SYSTEM.Security.Login(username, password)
```

ここで、username にはログインしているユーザ、password にはこのユーザのパスワードが入ります。

7.1.1.6 %Service_Mirror

このサービスは、InterSystems IRIS のデータベース・ミラーリングの使用を規制します。一般的なミラーリングの詳細は、"ミラーリング" を参照してください。ミラーリングの (TLS を使用した) セキュリティの詳細は、"ミラーリングで TLS を使用するための InterSystems IRIS の構成" を参照してください。

7.1.1.7 %Service_Sharding

このサービスは、シャード・データ・サーバとしての InterSystems IRIS インスタンスの使用を規制します。詳細は、"シャーディングによるデータ量に応じた水平方向の拡張" を参照してください。

7.1.1.8 %Service_WebGateway

このサービスは、Web ページを提供する接続を管理します。具体的には、Web サーバ上で実行されている Web ゲートウェイのプロセスと InterSystems IRIS サーバとの接続を管理します。このサービスを Web アプリケーションで直接操作することはありません。代わりに、関連する Web アプリケーション定義に認証メカニズムが構成されています。

以下の状況では、Web ゲートウェイを経由してサーバにアクセスすることはできません。

1. サービスに対して有効になっている認証メカニズムがある
2. 有効になっている認証メカニズムのいずれに対しても、有効な認証情報が Web ゲートウェイにない
3. サービスに対して非認証アクセスが無効になっている

したがって、このサービスによる非認証アクセスを無効にする場合 (すなわち、[認証なし] の認証メカニズムが無効になっている場合)、InterSystems IRIS サーバへの認証を受けるために必要な情報が Web ゲートウェイに必ず存在しているようにする必要があります。例えば、この情報とは、インスタンス認証 (パスワード) アクセスでは、有効なユーザ名とパスワードのペアであり、Kerberos アクセスでは、有効なサービス・プリンシパル名とキー・テーブルの位置です。Web ゲートウェイの認証情報を指定するには、その Web 管理インタフェースを使用します。標準インストールでは、このインタフェースの URL は http://localhost:52773/csp/bin/systems/module.cxw です。ここで、localhost は、IPv4 では 127.0.0.1、IPv6 では ::1 です。

%Service_WebGateway では、Web サーバ上で実行されている Web ゲートウェイのプロセスと InterSystems IRIS のインスタンス間のバックグラウンド認証のみが制御されます。その結果、Web アプリケーション向けの認証メカニズムは、関連の Web アプリケーション定義で構成され、管理されます。%Service_WebGateway 自体の許可された認証方法を編集することで構成と管理がされるわけではありません。

%Service_WebGateway は、ポータルとそのサブアプリケーションの使用を規制するものなので、**%Service_WebGateway** を無効にしてもシステム・アプリケーションは無効になりません。したがって、ポータルには常にアクセスできます。システム・アプリケーションの詳細は、"組み込みアプリケーション" を参照してください。

重要　不注意な操作でシステム管理ポータルからロックアウトされた場合は、緊急アクセス・モードを使用してシステム管理ポータルにアクセスし、問題を解決できます。詳細は、"緊急アクセス" を参照してください。

7.2 サービスのプロパティ

サービスにはそれぞれ、その動作を制御する一連のプロパティがあります。これには以下のものがあります。

- **[サービス名]** ― サービスの識別子を指定します。
- **[説明]** ― 必要に応じてサービスについての説明を提供します。
- **[サービス有効]** ― サービスを有効にするか無効にするかを制御します。サービスが有効になっていれば、認証および承認されたユーザは InterSystems IRIS に接続できます。サービスが無効の場合、InterSystems IRIS への接続は許可されません。

  システムが起動したときの各サービスは、InterSystems IRIS が終了したときと同じ状態 (有効または無効) になります。サービスの有効化や無効化は単なるセキュリティ設定ではありません。この設定によって、特定の機能が InterSystems IRIS で提供されるかどうかが決まります。例えば、特定のデーモン・プロセスを起動するかどうか、メモリ構造を割り当てるかどうかなどが決まります。
- **[許可された認証方法]** ― サービスに接続する際に利用できる認証メカニズムを指定します (2 要素認証メカニズムのいずれかなど)。複数の認証メカニズムが選択されている場合、接続しようとするユーザやクライアントは、それらのどの認証メカニズムでも使用できます。使用可能なメカニズムは、**[認証/Web セッション・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。複数の認証メカニズムをサポートしているサービスでは、InterSystems IRIS に設定されているカスケード認証の規則に基づいて、これらの認証メカニズムが使用されます。

  2 要素認証メカニズムのいずれかが有効になっている場合、それにはチェック・ボックスがあります。表示されている場合には以下の 2 つです。

  - **[2 要素のタイムベース・ワンタイム・パスワード]** ― InterSystems IRIS ユーザの携帯電話または認証デバイスが 2 つ目の認証「要素」として機能します。InterSystems IRIS と電話またはデバイスが秘密鍵を共有します。この鍵はタイムベース・ワンタイム・パスワード (TOTP) の生成に使用されます。ユーザは認証プロセスの一部として、このパスワードを指示に従って入力する必要があります。
  - **[2 要素の SMS]** ― InterSystems IRIS ユーザの携帯電話が 2 つ目の認証 “要素” として使用されます。InterSystems IRIS は携帯電話に 8 桁のセキュリティ・トークンを送信し、ユーザは認証プロセスの一部として、このセキュリティ・トークンを指示に従って入力する必要があります。

  詳細は、”2 要素認証” を参照してください。

  注釈　インスタンスに対して 2 要素認証が有効な場合、このチェック・ボックスはそのインスタンスのすべてのサービスの **[サービス編集]** ページに表示されます。ただし、2 要素認証は、**%Service_Bindings**、**%Service_Console**、および **%Service_Terminal** で (そのインスタンスに対して有効になっている場合) にのみ使用できます。
- **[許可済みの接続元]** ― このサービスに接続できる接続元の IP アドレスまたはマシン名のリストを指定します。IP アドレスやマシン名が関連付けられていないサービスには、どのマシンからでも接続できます。多層構成では、この機能が極めて便利です。例えば、Web ゲートウェイ・サービスではこの機能を使用して、InterSystems IRIS に接続できる Web サーバを制限できます。分散キャッシュ・クラスタ・データ・サーバの [許可済みの接続元] 機能には、他の機能もあります。”分散キャッシュ・クラスタのセキュリティ” を参照してください。

リソース・ベースのサービスでは、このサービスをパブリックに指定できます。認証されているユーザはすべて、パブリック・サービスを使用できますが、非パブリック・サービスを使用できるのは、そのサービスのリソースに対する Use 許可を持っているユーザのみです。この値は、サービスのメイン・ページ (**[システム管理]** > **[セキュリティ]** > **[サービス]**) に表示され、サービスのリソースについての **[リソースの編集]** ページで設定できます。可能な値は以下のとおりです。

- N/A ― このサービスに関連付けられたリソースはありません。このサービスには有効化または無効化のみを設定できます。

- いいえ ー このサービスのリソースに対して Use 許可があるロールを持つユーザであれば、アクセスが可能です。許可を持っているかどうかは、ユーザが認証された後に確認されます。
- はい ー あらゆるユーザがアクセスできます。

注釈　　サービスに対する変更を有効化するには、サービスを再起動する必要があります。

7.3 サービスおよび認証

基本サービスでは、インターシステムズのセキュリティの認証をサポートしていません。サービスを有効にするか無効にするかのみを設定できます。これらのサービスのいずれかを有効にすると、すべての接続がそのサービスで受け入れられます。これらのサービスでは、そのサービスを使用するすべてのインスタンスまたはマシンが安全な境界の内部に存在し、有効なユーザのみがアクセス可能であることが前提となっています。これらのサービスには、**%Service_ECP**、**%Service_Monitor**、**%Service_Shadow**、および **%Service_Weblink** があります。

リソース・ベース・サービスで認証メカニズムを有効にするには、まず、**[認証/Web セッション・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、InterSystems IRIS インスタンスに対してこのメカニズムを有効にする必要があります。リソース・ベース・サービスでは、以下のテーブルにある認証メカニズムをサポートしています。複数の認証メカニズムが有効なサービスの場合、InterSystems IRIS ではカスケード認証をサポートしています。

テーブル 7-1: 認証メカニズムをサポートしているサービス

サービス名	KRB キャッシュ	KRB ログイン	Del	LDAP	OS	IA	Un
%Service_Bindings	N	Y	Y	Y	N	Y	Y
%Service_CallIn	N	N	Y	Y	Y	N	Y
%Service_ComPort	N	N	Y	Y	N	Y	Y
%Service_Console	Y	Y	Y	Y	Y	Y	Y
%Service_Login	N	N	Y	Y	Y	Y	Y
%Service_Telnet	N	Y	Y	Y	N	Y	Y
%Service_Terminal	Y	Y	Y	Y	Y	Y	Y
%Service_WebGateway	N	Y	Y	Y	N	Y	Y

キー:

- KRB キャッシュ ー Kerberos キャッシュ
- KRB ログイン ー Kerberos ログイン
- Del ー 代行認証
- LDAP ー LDAP 認証
- OS ー オペレーティング・システム・ベースの認証
- IA ー インスタンス認証
- Un ー 非認証のアクセス

各リソース・ベース・サービスで、有効な認証メカニズムが複数存在する場合、InterSystems IRIS によるユーザの認証では、有効な認証メカニズムのうち最も厳しい基準のものが最初に適用され、基準が緩くなる順番で順次適用されていきます。非認証のアクセスが有効であれば、最後には非認証のアクセスが許可されます。このプロセスは、"カスケード認証" で説明されています。

7.4 サービスとそのリソース

リソース・ベース・サービスでは、サービスそのもののプロパティで InterSystems IRIS へのアクセスが制御されます。同時に、サービスのリソースのプロパティで、そのサービスへのアクセスとそのサービスの動作が制御されます。**%Service_Bindings** を除くすべてのリソース・ベース・サービスでは、サービスに関連付けられたリソースにはそのサービスと同じ名前が付けられています。したがって、**%Service_WebGateway** リソースは、**%Service_WebGateway** サービスに対するアクセスを管理します(**%Service_SQL** リソースおよび **%Service_Object** リソースは **%Service_Bindings** に対するアクセスを管理します)。

リソースそのものの関連プロパティは 2 つのみです。1 つはリソースがパブリックかどうかを指定するプロパティで、もう 1 つはリソースがパブリックである場合に許可の種類を指定するプロパティです。サービス・リソースの場合、関連する許可は Use のみです。リソースがパブリックであれば、そのサービスに対してすべてのユーザが Use 許可を持ちます。リソースの詳細は、"リソース" を参照してください。

他のリソースに対する特権がないと、サービスの特権から得られるものはほとんどありません。

8

認証に関する高度なトピック

この章では、認証の高度な使用法について説明します。

8.1 システム変数および認証

認証の後は、以下の 2 つの変数に値が設定されます。

- $USERNAME にはユーザ名が設定されます。
- $ROLES には、このユーザが保持するロールのコンマ区切りのリストが格納されます。

$ROLES 変数を使用すると、ロールをプログラムで管理できます。

8.2 複数の認証メカニズムの使用

複数の認証メカニズムの使用が推奨される状況として、現状よりも高い厳格さを備えた認証メカニズムに移行する場合があります。例えば、認証を使用していないインスタンスで Kerberos への移行を図る場合は、以下のシナリオが考えられます。

1. 移行期間については、認証されていないアクセスと Kerberos 認証によるアクセスの両方を使用できるように、サポート対象のすべてのサービスを構成します。これにより、ユーザはどちらのメカニズムを使用しても接続できます。
2. 適切であれば、認証に Kerberos を使用するクライアント・ソフトウェアを新規にインストールします。
3. InterSystems IRIS ユーザのリストと Kerberos データベースにあるユーザのリストが同じ内容になった時点で、すべてのサービスに対し、認証されていないアクセスを無効にします。

複数の認証メカニズムを使用する場合は、通常、以下のセクションで説明するカスケード認証を組み合わせます。

8.3 カスケード認証

InterSystems IRIS では、多数の認証メカニズムをサポートしていますが、Kerberos と共に他のパスワード・ベースの認証メカニズムを使用しないことをお勧めします。また、一部の特定の状況では、インスタンスでの複数の認証メカニズムの使用が推奨されます。

サービスが複数の認証メカニズムをサポートしている場合、InterSystems IRIS では カスケード認証 によってユーザ・アクセスが管理されます。カスケード認証では、指定されているメカニズムが以下の順番で適用されてユーザの認証が行われます。

- Kerberos キャッシュ (整合性チェックや暗号化を伴う Kerberos と伴わない Kerberos のどちらも含みます)。
- OS ベース
- LDAP (LDAP 資格情報キャッシュを 2 番目にチェック)
- 代行
- インスタンス認証
- 認証なし

注釈　Kerberos プロンプトの使用が指定されているサービスで認証が失敗すると、カスケード認証は適用されません。Kerberos プロンプトと Kerberos キャッシュの両方が指定されているサービスでは、Kerberos キャッシュのみが使用されます。

例えば、以下のメカニズムによる認証をサポートしているサービスを考えます。

1. Kerberos キャッシュ
2. OS ベース
3. 認証なし

ユーザが InterSystems IRIS に接続しようとすると、そのユーザが Kerberos のチケット保証チケット (TGT) を所有するかどうかがチェックされます。このチケットがあれば、InterSystems IRIS のサービス・チケットを取得しようとする処理が実行されます。この処理が成功すると、ユーザは接続します。最初の TGT が存在しない場合、または InterSystems サービスを取得できない場合、認証が失敗し、カスケードにある次の順番のメカニズムに移ります。

InterSystems IRIS ユーザ・リストにユーザの OS ベース識別情報が含まれていれば、ユーザは接続します。ユーザの OS ベース識別情報が InterSystems IRIS ユーザ・リストにない場合は認証が失敗し、カスケードにある次の順番のメカニズムに移ります。

認証されていないアクセスが、カスケード認証にある最後の選択肢であれば、このレベルまで移ってきたユーザは全員 InterSystems IRIS にアクセスできます。

注釈　インスタンスがカスケード認証をサポートしている場合、ユーザが 2 番目またはそれ以降の認証メカニズムで認証されるということは、その認証に成功するまでにいずれかのメカニズムでログインが失敗しているということになります。%System/%Login/LoginFailure 監査イベントが有効な場合、そのようなログイン失敗の情報はインスタンスの監査ログに記録されます。

8.4 UnknownUser アカウントとの接続の確立

インスタンス認証と認証なしモードの両方が有効になっている場合、ユーザは **[ユーザ名]** プロンプトと **[パスワード]** プロンプトで **Enter** キーを押すだけで、認証なしモードでサービスに接続できます。この場合は、UnknownUser アカウントが使用されます。インスタンス認証のみが有効になっている場合は、**[ユーザ名]** プロンプトと **[パスワード]** プロンプトで **Enter** キーを押しただけでは、サービスへのアクセスが拒否されます。InterSystems IRIS では、この処理が、UnknownUser アカウントでログインしようとしたユーザが、正しくないパスワードを指定したものとして扱われるためです。

8.5 プログラムによるログイン

状況によっては、アプリケーションの実行が始まった後でユーザがログインすることが必要な場合があります。このような状況の例として、認証されていないユーザには一部の機能のみを提供し、保護された機能を提供する場合には、ユーザにログインを要求するアプリケーションがあります。

$SYSTEM.Security クラスの Login メソッドを使用すると、アプリケーションから InterSystems IRIS ログイン機能を呼び出すことができます。これには、以下の構文を使用します。

ObjectScript

```
set success = $SYSTEM.Security.Login(username,password)
```

以下はその説明です。

- success はブーリアン値で、1 は成功、0 は失敗を示します。
- username は、ログインするアカウントの名前を保持する文字列です。
- password は、username アカウントのパスワードを保持する文字列です。

有効なユーザ名とパスワードが指定され、該当のユーザ・アカウントが有効で期限切れになっていなければ、ユーザはログインできます。それに応じて $USERNAME と $ROLES が更新され、この関数からは 1 が返されます。 それ以外の場合、$USERNAME と $ROLES は変更されず、この関数からは 0 が返されます。

$SYSTEM.Security.Login. を実行した結果として、特権のチェックは行われません。その結果、プロセスがそれまで保持していた特権が失われる可能性があります。

$SYSTEM.Security.Login. には、引数が 1 つのみの以下のような形式もあります。

ObjectScript

```
set success = $SYSTEM.Security.Login(username)
```

この形式の動作は、パスワードがチェックされない点を除けば、引数が 2 つの形式とまったく同じになります。引数を 1 つ使用した形式の $SYSTEM.Security.Login が便利なのは、アプリケーションで独自の認証が実行されることから、InterSystems IRIS ユーザの識別方法をそれに応じて設定する必要がある場合です。 また、あるプロセスが特定のユーザのために実行されても、そのプロセスを開始するのはそのユーザではない場合にも、この形式を使用できます。

注釈　　引数を 1 つ使用した形式の Login メソッドは、制限付きのシステム機能です。

8.6 JOB コマンド、および新しいユーザ識別の確立

JOB コマンドを使用してプロセスを作成すると、その基となったプロセスからセキュリティの特性 ($USERNAME の値と $ROLES の値) が継承されます。親プロセスに保持されている、User や Added などのすべてのロールが継承されます。

しかし、新しく作成するプロセスの $USERNAME と $ROLES の値を、その親とは異なる値にすることが必要な場合があります。例えば、特定のユーザ向けに特定の時間に特定のタスクを開始する目的で、タスク・マネージャが作成されることがあります。タスク・マネージャ自体は大きな特権を持っていることが普通ですが、該当のタスクはタスク・マネージャの特権ではなく、そのタスクの対象であるユーザの特権で実行する必要があります。

以下の擬似コードは、この処理方法を示しています。

```
WHILE ConditionToTest {
    IF SomeThingToStart {
        DO Start(Routine, User)
    }
}

Start(Routine, User) {
    NEW $ROLES    // Preserve $USERNAME and $ROLES

    // Try to change username and roles
    IF $SYSTEM.Security.Login(User) {
        JOB ...
        QUIT $TEST
    }
    QUIT 0       // Login call failed
}
```

8.7 認証と管理ポータル

管理ポータルは、複数の個別の Web アプリケーションで構成されています。ポータルのメイン・ページは /csp/sys のアプリケーションと関連付けられており、他のページはさまざまな /csp/sys/* のアプリケーションと関連付けられています (/csp/sys/sec のアプリケーションと関連付けられている、セキュリティ関連のコンテンツなど)。これらのアプリケーションすべてに使用される認証メカニズムの共通セットがないと、ユーザがあるポータル・ページから別のページに移動するときに、ログイン・プロンプトが表示されたり、突然特権レベルの移行が発生したりすることがあります。

例えば、/csp/sys のアプリケーションが排他的にインスタンス認証を使用し、関連する他のポータル・アプリケーションが排他的に認証なしのアクセスを使用している場合、ユーザがあるポータル・ページから別のページに移動すると、認証なしのアクセスから認証が必要な状態に移行します。また、/csp/sys のアプリケーションはインスタンス認証のみをサポートし、他のアプリケーションは認証なしのアクセスのみをサポートしている場合、UnknownUser に特別な特権がないと、ユーザがポータルのメイン・ページからその他のページに移動するときに、何らかのアクションを実行するのに十分な特権がないという可能性もあります。

Web アプリケーションの認証メカニズムを確認および構成するには、ポータルの [**ウェブ・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**]→[**ウェブ・アプリケーション**]) からアプリケーションを選択し、表示されたアプリケーションについて、必要に応じて、[**許可された認証方法**] で選択します (通常は、/csp/sys と /csp/sys/* で認証メカニズムの共通セットを共有するようにします)。