



システム管理ガイド

Version 2024.1
2024-06-03

システム管理ガイド

InterSystems IRIS Data Platform Version 2024.1 2024-06-03

Copyright © 2024 InterSystems Corporation

All rights reserved.

InterSystems®, HealthShare Care Community®, HealthShare Unified Care Record®, IntegratedML®, InterSystems Caché®, InterSystems Ensemble®, InterSystems HealthShare®, InterSystems IRIS®, および TrakCare は、InterSystems Corporation の登録商標です。HealthShare® CMS Solution Pack™ HealthShare® Health Connect Cloud™, InterSystems IRIS for Health™, InterSystems Supply Chain Orchestrator™, および InterSystems TotalView™ For Asset Management は、InterSystems Corporation の商標です。TrakCare は、オーストラリアおよび EU における登録商標です。

ここで使われている他の全てのブランドまたは製品名は、各社および各組織の商標または登録商標です。

このドキュメントは、インターシステムズ社(住所: One Memorial Drive, Cambridge, MA 02142)あるいはその子会社が所有する企業秘密および秘密情報を含んでおり、インターシステムズ社の製品を稼働および維持するためにのみ提供される。この発行物のいかなる部分も他の目的のために使用してはならない。また、インターシステムズ社の書面による事前の同意がない限り、本発行物を、いかなる形式、いかなる手段で、その全てまたは一部を、再発行、複製、開示、送付、検索可能なシステムへの保存、あるいは人またはコンピュータ言語への翻訳はしてはならない。

かかるプログラムと関連ドキュメントについて書かれているインターシステムズ社の標準ライセンス契約に記載されている範囲を除き、ここに記載された本ドキュメントとソフトウェアプログラムの複製、使用、廃棄は禁じられている。インターシステムズ社は、ソフトウェアライセンス契約に記載されている事項以外にかかるソフトウェアプログラムに関する説明と保証をするものではない。さらに、かかるソフトウェアに関する、あるいはかかるソフトウェアの使用から起こるいかなる損失、損害に対するインターシステムズ社の責任は、ソフトウェアライセンス契約にある事項に制限される。

前述は、そのコンピュータソフトウェアの使用およびそれによって起こるインターシステムズ社の責任の範囲、制限に関する一般的な概略である。完全な参照情報は、インターシステムズ社の標準ライセンス契約に記載され、そのコピーは要望によって入手することができる。

インターシステムズ社は、本ドキュメントにある誤りに対する責任を放棄する。また、インターシステムズ社は、独自の裁量にて事前通知なしに、本ドキュメントに記載された製品および実行に対する代替と変更を行う権利を有する。

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

1 管理ポータルの使用	1
1.1 管理ポータルの起動	1
1.1.1 管理ポータルのログイン・ページ	1
1.1.2 管理ポータル Web アプリケーションの構造	2
1.2 管理ポータルの概要	2
1.2.1 管理ポータルのホーム・ページ	3
1.2.2 管理ポータルのヘッダ	4
1.2.3 管理ポータルのリボン	5
1.2.4 システムの概要情報	5
1.3 管理ポータルの操作	6
1.3.1 列表示	6
1.3.2 リスト表示	7
1.4 管理ポータルのページ・リファレンス	7
1.4.1 ホーム	7
1.4.2 Health	7
1.4.3 分析	7
1.4.4 相互運用性	7
1.4.5 システム処理	8
1.4.6 システム・エクスプローラ	8
1.4.7 システム管理	9
2 システム情報の構成	11
2.1 メモリと開始設定	11
2.1.1 データベース・キャッシュおよびルーチン・キャッシュへのメモリの割り当て	12
2.1.2 プロセス当たりの最大メモリの設定	13
2.1.3 共有メモリ・ヒープ (gmheap) の構成	13
2.2 IPv6 のサポート	14
3 ネームスペースの構成	15
3.1 ネームスペース名の規則	15
3.2 ネームスペースの作成/変更	15
3.2.1 相互運用対応ネームスペースの作成	16
3.2.2 医療用途の相互運用性向けの Foundation ネームスペースの作成	16
3.3 ネームスペースの名前変更または既定マッピングの変更	17
3.4 ネームスペースの削除	17
3.5 プログラムによるネームスペースの構成	18
3.6 関連項目	18
4 ネームスペースへのマッピングの追加	19
4.1 グローバル・マッピング	19
4.2 ルーチン・マッピング	20
4.3 パッケージ・マッピング	21
4.4 すべてのネームスペースへのマッピング	22
4.5 関連項目	22
5 ローカル・データベースの構成	25
5.1 データベースに関する考慮事項	25
5.1.1 データベースのサイズと増大	25
5.1.2 データベース・キャッシュ	25

5.1.3 データベースの合計に関する制限	26
5.1.4 データベース構成に関する考慮事項	26
5.1.5 ラージ・ブロック・サイズに関する考慮事項	26
5.1.6 データベースの互換性に関する考慮事項	27
5.2 データベース名の規則	28
5.3 マルチボリューム・データベース	28
5.3.1 マルチボリューム・データベースの構成	28
5.3.2 ボリューム・ファイルの移動またはコピー	29
5.3.3 新規ボリュームのエラー動作	29
5.3.4 [Multi-Volume Properties] ページ	30
5.4 ローカル・データベースの表示	30
5.5 ローカル・データベースの作成	31
5.6 ローカル・データベースのプロパティの編集	32
5.6.1 ミラーリングされていないローカル・データベース・プロパティの編集	33
5.6.2 ミラーリングされているローカル・データベース・プロパティの編集	34
5.7 ローカル・データベースの再配置	35
5.8 ローカル・データベースの削除	36
5.9 プログラムによるデータベースの構成	37
5.10 関連項目	37
6 リモート・データベースの構成	39
6.1 リモート・データベースの追加	39
6.2 リモート・データベースの削除	40
6.3 関連項目	40
7 リソースの使用による資源の保護	41
7.1 リソースのタイプ	41
7.2 システム・リソース	42
7.2.1 管理リソース	42
7.2.2 %Development リソース	45
7.2.3 %DocDB_Admin リソース	46
7.2.4 %IAM リソース	46
7.2.5 %System_Callout リソース	46
7.2.6 %System_Attach リソース	47
7.2.7 %Secure_Break リソース	47
7.2.8 %Service_Native リソース	47
7.3 データベース・リソース	47
7.3.1 データベース・リソースの特権	47
7.3.2 共有データベース・リソース	48
7.3.3 既定のデータベース・リソース	48
7.3.4 認識されないリソース名、または無効なリソース名	48
7.3.5 ネームスペース	49
7.3.6 IRISSYS (マネージャ・データベース)	49
7.4 ゲートウェイ・リソース	50
7.5 アプリケーション・リソース	51
7.6 リソースの作成または編集	51
7.6.1 リソースの名前付け規約	52
7.7 管理ポータルによるカスタム・リソースの使用法	52
7.7.1 カスタム・リソースの定義とページへの適用	52
7.7.2 ページからのカスタム・リソースの削除	53
8 特権および許可	55

8.1 特権の機能	55
8.2 パブリック許可	55
8.3 プロセスの特権の確認	56
8.4 組み込みの特権確認機能を持つメソッドの使用	57
8.5 特権の変更が有効になるタイミング	57
9 ロール	59
9.1 ロールについて	59
9.1.1 ロール割り当てについて	60
9.1.2 ロールの最大数	60
9.2 ロール、ユーザ、メンバ、および割り当て	60
9.2.1 複数ロールの割り当て例	61
9.3 ロールの作成	63
9.3.1 命名規則	64
9.4 ロールの管理	65
9.4.1 既存のロールの確認	65
9.4.2 ロールの削除	65
9.4.3 ロールに対する新しい特権の付与	66
9.4.4 ロールの特権の変更	66
9.4.5 ロールからの特権の削除	66
9.4.6 現在のロールに対するユーザまたはロールの割り当て	66
9.4.7 現在のロールからのユーザまたはロールの削除	67
9.4.8 他のロールに対する現在のロールの割り当て	67
9.4.9 他のロールからの現在のロールの削除	68
9.4.10 ロールの SQL 関連オプションの変更	68
9.5 事前定義のロール	71
9.5.1 %All	72
9.5.2 既定のデータベース・リソース・ロール	73
9.6 ログイン・ロールおよび追加ロール	73
9.6.1 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	73
9.7 プログラムで管理するロール	74
10 ユーザ・アカウント	75
10.1 ユーザ・アカウントのプロパティ	75
10.1.1 ユーザ・タイプについて	76
10.2 ユーザ・アカウントの管理	77
10.2.1 ユーザ・アカウントの新規作成	77
10.2.2 既存のユーザ・アカウントの編集	78
10.2.3 ユーザ・プロフィールの表示	82
10.2.4 ユーザ・アカウントの無効化/有効化	82
10.2.5 ユーザ・アカウントの削除	83
10.3 事前定義のユーザ・アカウント	83
10.3.1 さまざまなアカウントに関するメモ	85
10.4 ユーザ・アカウントの検証	86
11 アプリケーション	89
11.1 アプリケーション、およびそのプロパティと特権	89
11.1.1 アプリケーションとそのプロパティ	90
11.1.2 リソースへのアプリケーションの関連付け	91
11.1.3 アプリケーションおよび特権のエスカレーション	91
11.1.4 プログラムによる特権チェック	94
11.2 アプリケーション・タイプ	94

11.2.1 Web アプリケーション	94
11.2.2 特権ルーチン・アプリケーション	96
11.2.3 クライアント・アプリケーション	99
11.2.4 ドキュメント・データベース・アプリケーション	99
11.3 アプリケーションの作成および編集	99
11.3.1 アプリケーションの作成	100
11.3.2 Web アプリケーションの編集 : [一般] タブ	100
11.3.3 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	104
11.3.4 アプリケーションの編集 : [アプリケーションロール] タブ	105
11.3.5 アプリケーションの編集 : [マッチングロール] タブ	105
11.3.6 アプリケーションの編集 : [ルーチン/クラス] タブ	106
11.3.7 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	106
11.4 組み込みアプリケーション	107
12 TLS の構成	109
12.1 TLS 構成の作成または編集	109
12.1.1 証明書に必要な情報	112
12.1.2 有効化された暗号スイートの構文	112
12.1.3 TLS を使用する InterSystems IRIS クライアント・アプリケーションに関するメモ	113
12.2 構成の削除	113
12.3 既存の構成の証明書の更新	113
12.4 予約済みの構成名と必須の構成名	113
12.5 プログラムによる TLS 構成の作成、編集、削除	114
13 スーパーサーバの管理	115
13.1 スーパーサーバの概要	115
13.2 サポートされている要求	115
13.3 仕組み	116
13.3.1 システムの既定のスーパーサーバ	116
13.4 構成方法	116
13.4.1 管理ポータルの使用	116
13.4.2 ^SECURITY の使用	118
13.4.3 CPF と CMF の使用	119
13.4.4 プログラムの使用	119
14 OAuth 2.0 および OpenID Connect に対するサポート	121
14.1 サポートされているシナリオ	121
14.2 OAuth 2.0 および OpenID Connect に対する InterSystems IRIS のサポート	121
14.2.1 クライアントの構成項目	122
14.2.2 サーバの構成項目	122
14.3 InterSystems IRIS でサポートされる標準規格	123
15 タスク・マネージャの電子メール設定の構成	125
15.1 電子メール構成の設定	125
15.2 件名とメッセージのパラメータ	126
16 各国言語サポート (NLS) の構成	127
16.1 管理ポータルの NLS ページの使用法	127
16.1.1 構成済み既定値	127
16.1.2 ロケール定義	128
16.1.3 ロケールのインポート	129
16.2 NLS クラス・パッケージの使用法	129

17	作業キュー・マネージャ・カテゴリの構成	131
17.1	作業キュー・マネージャ・カテゴリの作成	131
17.2	作業キュー・マネージャ・カテゴリの編集または削除	132
18	ローカル・データベースの管理	133
18.1	データベースの一般情報	133
18.2	データベースの空き容量情報	134
18.2.1	管理ポータルを使用した空き容量情報の表示	134
18.2.2	%FREECNT を使用した空き容量情報の表示	135
18.3	データベースの詳細ページ	136
18.4	データベースの圧縮	137
18.5	データベースの領域の削除	138
18.6	データベースのデフラグ	139
18.7	データベースのグローバルの圧縮	140
19	InterSystems IRIS プロセスの管理	143
19.1	使用可能なオプション	143
19.2	プロセス情報の表示	144
19.3	プロセスの詳細の表示	145
19.3.1	プロセスの一時停止または再開	147
19.3.2	プロセスの終了	147
19.3.3	プロセス変数の表示	147
19.4	ターミナルへのメッセージの送信	148
20	サービス	149
20.1	使用可能なサービス	149
20.1.1	個々のサービスに関するメモ	150
20.2	サービスのプロパティ	152
20.3	サービスおよび認証	153
20.4	サービスとそのリソース	154
21	タスク・マネージャの使用	155
21.1	新規タスク	155
21.2	タスク・タイプ	157
21.3	オンデマンドタスク	159
21.4	実行予定タスク	159
21.5	タスクスケジュール	160
21.5.1	タスクの詳細	160
21.5.2	スケジュールされたタスクの履歴	161
21.6	タスク履歴	161
21.7	タスクのインポート	161
21.8	^TASKMGR の使用	161
21.8.1	タスク期限切れ動作の変更	161
21.9	カスタム・タスク	162
22	[バックグラウンド・タスク] ページの使用	165
23	InterSystems IRIS ライセンスの管理	167
23.1	ライセンス管理の概要	167
23.2	ライセンス・サーバとネットワークに関する考慮事項	168
23.3	ライセンス・サーバの構成	168
23.4	ライセンス・サーバへのキーのロード	169
23.5	ライセンス・キーの有効化	169
23.6	ライセンス・キーの更新	170

23.7 ライセンス・トラブルシューティング	171
23.7.1 管理者ターミナル・セッション	171
23.8 オペレーティング・システムのコマンド行からのライセンスのアップグレード	172
23.9 ライセンスの機能範囲と使用の決定	172
23.10 ローカル・ライセンス情報を表示するメソッド	174
23.11 ライセンス・サーバ情報を表示するメソッド	174
23.12 ユーザの認識	175
23.12.1 ライセンス・ログイン	176
23.12.2 ユーザ名ライセンス	177
23.12.3 ライセンス・ログインの特別な考慮事項	177
23.13 アプリケーション・ライセンス	178
23.13.1 アプリケーション・ライセンスのロード	178
23.13.2 アプリケーション・ライセンス API	179
24 Windows での InterSystems IRIS の管理	181
24.1 InterSystems IRIS インスタンスへのアクセスの管理	181
24.1.1 InterSystems サービス	181
24.1.2 インストール・ツリーへのアクセスの制限	182
24.1.3 InterSystems IRIS サービス・アカウントの変更	182
24.2 InterSystems IRIS ランチャー	183
24.3 InterSystems IRIS の開始	184
24.4 InterSystems IRIS の停止	184
24.5 コマンド・プロンプトからのヘルプ情報	185
24.6 コマンド行での InterSystems IRIS への接続	185
25 UNIX®, Linux、および macOS での InterSystems IRIS の管理	187
25.1 UNIX® のユーザ、グループおよび権限	187
25.1.1 データベースおよびデータベース・ディレクトリのアクセス権限	188
25.2 UNIX® での起動	188
25.2.1 デーモン・リソース・ロック	189
25.2.2 iris.ids ファイル	189
25.2.3 起動シーケンス	189
25.3 InterSystems IRIS の管理	190
25.3.1 InterSystems IRIS の開始	190
25.3.2 InterSystems IRIS の実行	191
25.3.3 InterSystems IRIS の停止	192
26 リモート・サーバへの接続	193
26.1 ランチャーのサブメニュー	193
26.2 リモート・サーバ接続の定義	194
26.3 CSP サーバ・インスタンスの指定	195
26.4 Telnet を介した接続	195
27 InterSystems IRIS インスタンスの管理 : iris コマンド	197
27.1 インスタンスの管理オプション	197
27.2 インスタンスへの接続	198
27.3 インスタンスの制御	198
27.4 UNIX®, Linux、および MacOS での iris コマンド	199
27.5 Windows での iris コマンド	202
27.6 iris list、qlist、および all	204
27.7 複数のインスタンスの構成	206
27.7.1 ポート番号の設定	206

付録A: インターシステムズ製品と関係して動作するようにサードパーティ・ソフトウェアを構成する方法	207
付録B: 機能トラッカによる使用量統計の収集	209
B.1 統計を収集する理由	209
B.2 ログの場所	209
B.3 収集される統計データ	210
B.4 機能トラッカを非アクティブ化する方法	210

テーブル一覧

テーブル 7-1: データベースの特権	47
テーブル 7-2: %DB_%DEFAULT の特権	48
テーブル 8-1: 既定のパブリック特権	56
テーブル 9-1: ロールのプロパティ	59
テーブル 9-2: 認証メカニズムとロール割り当てメカニズム	60
テーブル 9-3: 事前定義ロールとその特権	71
テーブル 10-1: ユーザ・アカウントのプロパティ	75
テーブル 10-2: ユーザ・プロファイルのプロパティ	82
テーブル 10-3: 事前定義のユーザ・アカウント	84
テーブル 11-1: 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	93
テーブル 11-2: InterSystems IRIS の組み込み Web アプリケーション	107
テーブル 18-1: ローカル・データベース情報	133
テーブル 18-2: ローカル・データベースの空き容量情報	135
テーブル 18-3: 一般情報	137
テーブル 18-4: データベースサイズ	137
テーブル 19-1: プロセス管理機能	144
テーブル 19-2: プロセス・カラム情報	144
テーブル 19-3: 一般情報	145
テーブル 19-4: クライアント・アプリケーションの詳細	146
テーブル 19-5: 実行詳細	146
テーブル 20-1: 認証メカニズムをサポートしているサービス	153

1

管理ポータルの使用

InterSystems IRIS® データ・プラットフォームでは、Web アプリケーションであるインターシステムズの管理ポータルを使用してシステム管理と管理タスクを実行できます。

1.1 管理ポータルの起動

管理ポータルは、次の方法で起動できます。

- ・ Microsoft Windows プラットフォームで、InterSystems IRIS ランチャー・メニューの **[管理ポータル]** をクリックします。
- ・ ブラウザから、インスタンスの `<baseURL>` を使用して、以下の URL にアクセスします。

`https:<baseURL>/csp/sys/UtilHome.csp`

管理ポータルの詳細は、“[管理ポータルの概要](#)”を参照してください。

システムの認証設定によっては、ポータルのホーム・ページに移動する前にログインする必要があります。ログイン要件の概要は、“[管理ポータルのログイン・ページ](#)”を参照してください。さらに、各機能領域は、特定のリソースにアクセスできる必要があります。“[管理ポータル Web アプリケーションの構造](#)”を参照してください。

重要 Microsoft Windows 2003 ユーザの信頼済みサイトのセキュリティ設定 – 管理ポータルを初めて表示したときに、Web サイトがブロックされているという警告が表示されることがあります。プロンプトが表示されたら、このサイトを信頼済みサイト・ゾーンに追加します。インターシステムズでは、ポータルのプロシージャが正しく機能するために、セッション・クッキーを許可することをお勧めします。

1.1.1 管理ポータルのログイン・ページ

管理ポータル Web アプリケーション (`/csp/sys`) の認証設定によっては、ポータルを使用するためにユーザ名とパスワードを入力する必要があります。これには、以下の 2 つの状況があります。

認証されていないアクセスのみ

ユーザ名もパスワードも不要で、入力もできません – 認証されていない接続のみがポータルで許可されている場合 (最小のセキュリティ・インストールでの既定の設定)、ポータルにアクセスするためにユーザ名とパスワードは不要です。この場合、前述のセクションで説明されている方法を使用してポータルを起動するときには、**[ログイン]** ページが表示されません。例えば、**[ログアウト]** をクリックすることでログイン・ページが表示された場合も、以下のメッセージが表示されます。

Please click [here](#) to log in.

認証されたアクセス

ユーザ名とパスワードが必要です – 管理ポータル Web アプリケーションに対しては認証を必要とするセキュリティ設定を使用していて、システムで未認証である場合は、[ユーザ名]と[パスワード]の入力を求めるログイン・ページが表示されます。これらのフィールドに入力した後、[ログイン]をクリックすると、管理ポータルのホーム・ページが表示されます。

注釈 パスワードを変更するには、自分の名前を[管理ポータルのヘッダ](#)でクリックします。パスワード変更のダイアログ・ボックスが表示されます。

重要 /csp/sys アプリケーション行で [編集] をクリックすると、[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]) で [許可されている認証] 設定を確認できます。

1.1.2 管理ポータル Web アプリケーションの構造

メインのポータル Web アプリケーションは、/csp/sys にあります。管理ポータル内で特権の要求を適用するために、ポータルはさらにいくつかの Web アプリケーションに分かれています。管理ポータル自体は、特権のないユーザの操作を阻止しません。ポータルによって呼び出されるシステム API メソッドがこれを行います。ただし、<PROTECT> エラーを防止するために、ポータルでは、特権のないユーザが制限付きのページにアクセスできないようにしています。

ポータルのユーザには次の 4 つのタイプがあります。それらのタイプは、InterSystems IRIS 内であらかじめ定義されているリソースにおおよそ対応しています。

ユーザ・タイプ	リソース	Web アプリケーション	タスク
セキュリティ・マネージャ	%Admin_Secure	/csp/sys/sec	ユーザ、ロール、およびその他のセキュリティ・タスクのリストを表示および編集します。
マネージャ	%Admin_Manage	/csp/sys/mgr	システム構成を変更し、バックアップ・セットを定義します。
オペレータ	%Admin_Operate	/csp/sys/op	システム状態のページを表示し、バックアップを実行します。
エクスプローラ (パブリック・ユーザ)	%Development	/csp/sys/exp	該当するリソースにアクセスできる場合、ホーム・ページを表示し、クラス、ルーチン、およびグローバルを表示し、SQL ページを使用します。

1.2 管理ポータルの概要

このセクションでは、管理ポータル・ページのいくつかの一般的なレイアウト要素について説明します。以下の項目について説明します。

- ・ [管理ポータルのホーム・ページ](#)
- ・ [管理ポータルのヘッダ](#)
- ・ [管理ポータルのリボン](#)
- ・ [システムの概要情報](#)

注釈 管理ポータル内のどこであっても、メニュー項目の上にカーソルを移動すると、その項目の説明が表示されます。

1.2.1 管理ポータルのホーム・ページ

管理ポータルのホーム・ページには、[ようこそ, <user>] というタイトルが付きます。タイトルの横のリボンには、以下のオプションが含まれます。

- ・ 2 つの [表示] ボタン。メニュー列でのリンクの表示方法を指定できます。
- ・ 検索バー。リボンの右側にあります。単語を指定してから **Enter** キーを押すと、その単語が含まれる全ページのリストが表示されます。そのリストでは、表示するリンク先ページをクリックできるので、サブメニューを使って移動する必要はありません。

The screenshot shows the Management Portal Home Page. The header includes a greeting 'Welcome, Admin', a 'View' toggle with icons for list and grid views, and a search bar. The main content area is divided into three vertical sections:

- Menu Column:** A vertical list of navigation links: Home, Analytics, Interoperability, System Operation, System Explorer, and System Administration.
- Welcome Pane:** A central area with four sections:
 - Favorites:** A link to 'Go to a favorite page'.
 - Did you know?:** A tip stating 'You can jump directly to items by clicking on the Menu at the top of the page.'
 - Recent:** A link to 'Go to a recently viewed page'.
 - Links:** A section titled 'Pages you may be interested in' with links to 'Documentation', 'Support', and 'InterSystems'.
- Message Pane:** A right-hand section containing:
 - SYSTEM INFORMATION:** 'General details on this system' with a link to 'View System Dashboard'.
 - System Up Time:** '0d 0h 08m'.
 - PRODUCTIONS:** 'There are no productions currently running on this system'.

以下のセクションでは、ホーム・ページの各領域について説明します。

- ・ [管理ポータルのメニュー列](#)
- ・ [管理ポータルのようこそペイン](#)
- ・ [管理ポータルのメッセージ・ペイン](#)

1.2.1.1 管理ポータルのメニュー列

ホーム・ページの左端にあるメニュー列は、主要なポータル操作手段です。

管理ポータル・メニューの操作の詳細は、“[管理ポータルの操作](#)” を参照してください。

管理ポータルの特定のページの詳細は、“[管理ポータルのページ・リファレンス](#)” を参照してください。

1.2.1.2 管理ポータルのようにそペイン

ようにそペインはホーム・ページの中央を占め、頻繁にアクセスするページへのショートカットを含みます。ここには以下のフィールドがあります。

- ・ **【お気に入り】** – お気に入りとして指定した管理ポータル・ページがリストされます (“[アクション・ペイン](#)” を参照)。各ページのタイトルをクリックすると、そのページに直接移動できます。
- ・ **【最近アクセスしたページ】** – 最後に InterSystems IRIS を起動した後に表示した最近のページがリストされます。
- ・ **【ご存知ですか ?】** – ヒントを表示します。
- ・ **【リンク】** – アクセスするページにリンクします。

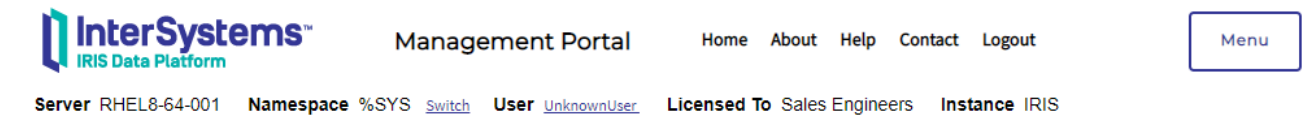
1.2.1.3 管理ポータルのメッセージ・ペイン

ホーム・ページの右端にあるメッセージ・ペインでは、一般的なシステム情報が表示され、システム・ダッシュボードへのリンクが提供されます。ダッシュボードの詳細は、“[管理ポータルを使用した InterSystems IRIS の監視](#)” を参照してください。

インスタンスがミラー・メンバの場合、メッセージ・ペインには、それが属するミラー、そのステータスとメンバ・タイプ、およびミラー・モニタへのリンクも表示されます (“[ミラーの監視](#)” を参照)。

1.2.2 管理ポータルのヘッダ

管理ポータルの各ページの上部にあるヘッダは、ポータルをすばやく操作するのに使用できます。



ヘッダには以下のリンクがあります。

- ・ **【ホーム】** – [管理ポータルのホーム・ページ](#)が表示されます。
- ・ **【概要】** – [システムの概要情報](#)が表示されます。
- ・ **【ヘルプ】** – 表示しているページやトピックのオンライン・ドキュメント (ヘルプ) が表示されます。
- ・ **【コンタクト】** – [インターシステムズのサポート窓口](#)のお問い合わせページが表示されます。
- ・ **【ログアウト】** – ログアウトして、管理ポータルの **【ログイン】** ページが表示されます。
- ・ **【メニュー】** – ユーザが保持するロールに基づいた一般タスクのリストが表示されます。

ヘッダには、以下のような有用な情報も含まれます。

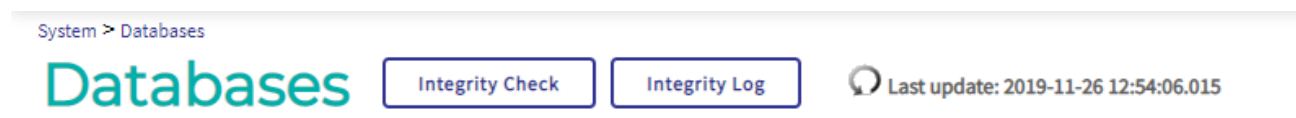
- ・ **【サーバ】** – InterSystems IRIS が実行されているサーバの名前
- ・ **【ネームスペース】** – 現在使用されているネームスペースの名前異なるネームスペースで作業するには、ネームスペース名 (例えば、%SYS) をクリックして、目的のネームスペースを選択します。
- ・ **【ユーザ】** – 管理ポータルにログインしているユーザの名前。ユーザのパスワードを変更するには、その名前をクリックします。
- ・ **【ライセンス先】** – ライセンス・キー情報に表示される顧客名
- ・ **【インスタンス】** – サーバ上で実行されている InterSystems IRIS インスタンスの名前

また、[システムモード] ラベルを表示できます (例えば、[テストシステム])。詳細は、“メモリと開始設定” を参照してください。

管理ポータルのヘッダの左側には、使用している製品の名前が表示されます。

1.2.3 管理ポータルのリボン

リボンはヘッダの直下に配置され、各ページに固有のさまざまなコンテンツを表示します。例として、[データベース] ページ ([システムエクスプローラ]→[データベース]) のリボンを以下に示します。



リボンの一般的なコンテンツは以下のとおりです。

- ・ 表示される管理ポータル・ページのタイトル。
- ・ ページ・タイトルのすぐ上に配置された、現在のページへのパンくずリスト。パスにリストされた各ページは、アクティブなリンクです。以前に表示していたサブメニューやリストに戻るために使用できます。ページ上に保存されていない変更がある場合、そのパンくずリストにはアスタリスクが付加されます (システム > 構成 > メモリと開始設定 – (構成設定)* など)。保存していない変更のある場所から別の場所に移動する前には、確認を求めるプロンプトが必ず表示されます。

注釈 パンくずリストは、パス内のすべてのページをリストするわけではありません。また、パンくずリスト内のページはナビゲーション・メニュー内のページと一致するとは限りません。[ホーム] をクリックして、管理ポータルのホーム・ページに戻り、検索ツールを使用すると、特定のページにいつでも移動できます。検索ツールについては、このセクションの後半で説明します。

- ・ ページ上で操作を実行できるいくつかのボタン。例えば、[データベース] ページには、[整合性チェック] および [整合性ログ] というボタンがあります。
- ・ 前回ページが更新されたときの情報を保持する更新ボタン。

1.2.4 システムの概要情報

管理ポータルのヘッダの [概要] をクリックすると、テーブルに以下の情報が表示されます。

- ・ [バージョン] – プラットフォーム、ビルドの番号および日付を含む、InterSystems IRIS のこのインスタンスの固有のビルド情報
- ・ [構成] – このインスタンスが使用する構成 (.cpf) ファイルの名前と場所
- ・ [データベースキャッシュ(MB)] – データベースに割り当てられている領域
- ・ [ルーチンキャッシュ(MB)] – ルーチンに割り当てられている領域
- ・ [ジャーナルファイル] – 現在のジャーナル・ファイルの名前と場所
- ・ [スーパーサーバポート] – InterSystems IRIS サーバを実行するポート番号
- ・ [ウェブサーバポート] – Web サーバを実行するポート番号。
- ・ [ライセンスサーバアドレス/ポート] – InterSystems IRIS ライセンス・サーバの IP アドレス、およびそのサーバを実行するポート番号。
- ・ [ライセンス先] – ライセンス・キー情報に表示される顧客名
- ・ [クラスタサポート] – このインスタンスがクラスタの一部であるかどうか

- ・ **[ミラーリング]** – このインスタンスがミラーのメンバであるかどうか
- ・ **[システム開始日時]** – InterSystems IRIS のこのインスタンスが最後に起動した日付と時刻
- ・ **[暗号化キー識別子]** – 暗号化がアクティブ化されている場合、暗号化キーの GUID (Global Unique ID)
- ・ **[NLSロケール]** – 各国語言語サポートのロケール 詳細は、“[各国言語サポートのシステム・クラスの使用法](#)” および “[管理ポータルの NLS ページの使用法](#)” を参照してください。
- ・ **[このセッションの優先言語]** – 管理ポータルのローカライズが完了していて、表示可能な言語のドロップダウン・リスト。ドロップダウン・リストから新しい言語を選択すると、表示する言語を変更できます。ブラウザ・セッションの最初の優先言語はブラウザで指定された言語になります。そのブラウザ言語がサポートされていない場合は英語になります。特定のブラウザで優先言語を選択すると、後でブラウザ言語が変更されても、そのブラウザの管理ポータルでは選択した言語が使用されます。

1.3 管理ポータルの操作

管理ポータルには、ページを操作するために選択できる 2 つの異なる表示モードがあります。管理ポータルのリボンで目的の **[表示]** ボタンをクリックすることにより、いつでも表示モードを切り替えることができます。表示モードには以下のものがあります。

- ・ **列表示** – 列にページが表示されます。これが既定モードです。この表示モードでは、お気に入りを選択し、カスタム・リソースを割り当てることができます。
- ・ **リスト表示** – ページ名がドロップダウン・リストに表示されます。

注釈 [表示] を使用して管理ポータルに移動することに加えて、検索ツールにより直接ページに移動できます。このツールはリボンに配置されています。詳細は、“[管理ポータルのリボン](#)” を参照してください。

1.3.1 列表示

列表示では、リンク先ページへのアクセスに使用するサブメニューが表示されます。最後のサブメニューのタイトル/名前をクリックすると、目的のページが表示されます。ボックス内でタイトル/名前ではない場所をクリックすると、[アクション・ペイン](#)が右側に表示されます。

1.3.1.1 アクション・ペイン

アクション・ペインは、メニュー・オプションについて説明し、それに関連する複数のアクションを実行できます。アクション・ペインを表示するには、**列表示**の状態、最後のサブメニューのオプションを囲むボックス内の、オプション自体ではない場所をクリックします。アクション・ペインでは、以下の操作を実行できます。

- ・ “[管理ポータルのごようこそペイン](#)” の説明に従って、目的ページをホーム・ページの **[お気に入り]** リストに追加します。
- ・ ページをロードするために必要なシステム・リソースを表示し、カスタム・リソースを割り当てます。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。

注釈 アクション・ペインでリソース名が一部しか表示されていない場合は、名前の上にポインタを移動すると、全体を見ることができます。

- ・ **[進む]** をクリックして目的のページに移動します。

アクション・ペインは、列表示でのみ使用できます。リスト・ビューまたは検索ツールを使用してページに移動すると、お気に入り追加したり、カスタムのリソースを割り当てたりできません。

1.3.2 リスト表示

リスト表示により、リンク先ページへのアクセスに使用するリストが表示されますが、リンク先ページをお気に入りとして指定したりカスタム・リソースを割り当てることはできません。表示するページを選択する場合、**[進む]** をクリックします。

1.4 管理ポータルのページ・リファレンス

管理ポータルは、以下の機能領域に分けられた多くのページで構成されます。

- ・ [\[ホーム\]](#) – 管理ポータルのホーム・ページ
- ・ [\[Health\]](#) – 医療関連ページ。InterSystems IRIS for Health™ および HealthShare® Health Connect 専用です。
- ・ [\[Analytics\]](#) – InterSystems IRIS® データ・プラットフォーム Business Intelligence のページ
- ・ [\[Interoperability\]](#) – プロダクション・ページ
- ・ [\[システム処理\]](#) – システム処理ページ
- ・ [\[システムエクスプローラ\]](#) – データベース管理ページ
- ・ [\[システム管理\]](#) – システム管理ページ

このドキュメントには、ポータルの各ページに適したドキュメント・セクションへのリンクがあります。

1.4.1 ホーム

ホーム・ページには、管理ポータルを操作するための多くのオプションが含まれます。詳細は、[“管理ポータルのホーム・ページ”](#) を参照してください。

1.4.2 Health

InterSystems IRIS for Health および Health Connect の医療用途に特化した機能が、管理ポータルの [\[Health\]](#) から利用可能なページに表示されます。これらの医療関連ページにアクセスするには、Foundation ネームスペースをインストールする必要があります。詳細は、[“InterSystems IRIS for Health インストール・ガイド”](#) または [“HealthShare Health Connect インストール・ガイド”](#) の [“インストーラ・ウィザードの使用”](#) を参照してください。

1.4.3 分析

Business Intelligence のページは、管理ポータルの [\[分析\]](#) 部分に表示されるカテゴリに分けられます。[“Business Intelligence ユーザ・インタフェースの概要”](#) を参照してください。

1.4.4 相互運用性

プロダクションのページは、管理ポータルの [\[相互運用性\]](#) の部分に表示されるカテゴリに分けられます。詳細は、[“メタニュー項目に関する情報の検索”](#) を参照してください。

1.4.5 システム処理

システム・オペレータのページは、管理ポータルの [システム処理] の部分に表示されるカテゴリに分けられます。以下のテーブルに、各カテゴリと関連するドキュメントを示します。

メニュー項目	情報
システム・ダッシュボード	管理ポータルを使用した InterSystems IRIS の監視
バックアップ	オンライン・バックアップの管理
データベース	ローカル・データベースの管理
プロセス	InterSystems IRIS プロセスの管理
SQL アクティビティ	SQL アクティビティの監視
ロック	ロックの監視
ジャーナル	ジャーナリング
ミラー・モニタ	ミラーの監視
タスク・マネージャ	タスク・マネージャの使用
LDAP 構成	“ポータルでの LDAP 構成の %Operator としての表示”
システム・ログ	ログ・ファイルの監視
システム使用	システム・パフォーマンスの監視
ライセンスの使用	InterSystems IRIS ライセンスの管理
相互運用性の使用	相互運用性の使用
バックグラウンド・タスク	JOB
診断レポート	InterSystems IRIS 診断レポートの使用法

1.4.6 システム・エクスプローラ

データベース管理のページは、管理ポータルの [システムエクスプローラ] の部分に表示されるカテゴリに分けられます。以下のテーブルに、各カテゴリと関連するドキュメントを示します。

メニュー項目	情報
クラス	クラス
SQL	InterSystems SQL 入門
ルーチン	ユーザ定義コード
グローバル	グローバルについての正式な規則
ツール	QA、スケーラビリティ、およびネットワーク負荷をテストする目的で、さまざまなアプリケーションに対して、HTTP ベースのスクリプトの記録、ランダム化、および再生を行うために使用できるクラスを提供します。詳細は、“インターシステムズ・クラス・リファレンス” の %WebStress を参照してください。

1.4.7 システム管理

システム管理者のページは、管理ポータルの[システム管理]の部分に表示されるカテゴリに分けられます。以下のテーブルに、各カテゴリと関連するドキュメントを示します。

メニュー項目	情報
構成	“構成ページ”のテーブル (下記)
セキュリティ	システム管理およびセキュリティ
ライセンス	InterSystems IRIS ライセンスの管理
暗号化	暗号化ガイド

1.4.7.1 構成ページ

構成ページは、[システム管理](#)ページのサブカテゴリです。

メニュー項目	情報
システム構成	“システム構成ページ”のテーブル (下記)
接続性	“接続性ページ”のテーブル (下記)
ミラー設定	ミラーリングの構成
データベースバックアップ	オンライン・バックアップ設定の構成
SQL およびオブジェクトの設定	“SQL およびオブジェクトの設定ページ”のテーブル (下記)
デバイス設定	“デバイス設定ページ”のテーブル (下記)
各国言語の設定	各国言語サポート (NLS) の構成
追加設定	“追加設定ページ”のテーブル (下記)

システム構成ページ

システム構成ページは、[構成](#)ページのサブカテゴリです。

メニュー項目	情報
メモリと開始設定	システム情報の構成
ネームスペース	ネームスペースの構成
ローカルデータベース	データベースの構成
リモートデータベース	リモート・データベースの構成
ジャーナル設定	ジャーナル設定の構成

接続性ページ

接続性ページは、[構成](#)ページのサブカテゴリです。

メニュー項目	情報
ECP 設定	分散キャッシュ・クラスタの導入
SQL ゲートウェイ設定	外部ソースへのゲートウェイ接続の作成
外部言語サーバ	[Gateways]

SQL およびオブジェクトの設定ページ

SQL およびオブジェクトの設定ページは、[構成](#)ページのサブカテゴリです。

メニュー項目	情報
SQL	[SQL]
オブジェクト	%ObjectSelectMode=1 を使用したフィールド名・プロパティのスイズリング
TSQL 互換性	TSQL 設定
ISQL 互換性	ISQL 構成の設定
システム DDL マッピング	システム・データ型
ユーザ DDL マッピング	ユーザ・データ型

デバイス設定ページ

デバイス設定ページは、[構成](#)ページのサブカテゴリです。

メニュー項目	情報
デバイス	Devices (構成パラメータ・ファイル・リファレンス)
デバイス・サブタイプ	DeviceSubTypes
IO 設定	[IO]
Telnet 設定	[Telnet]

追加設定ページ

追加設定ページは、[構成](#)ページのサブカテゴリです。

メニュー項目	情報
互換性	[Miscellaneous]
詳細メモリ	[config] パラメータ
モニタ	SNMP を使用した InterSystems IRIS の監視
ソース・コントロール	“InterSystems IRIS とソース・コントロール・システムの統合”
開始	[Startup]
タスク・マネージャの電子メール設定	“システム管理ガイド” の “タスク・マネージャの電子メール設定の構成”

2

システム情報の構成

InterSystems IRIS® データ・プラットフォームでは、構成パラメータ・ファイル (CPF) と呼ばれるファイルにシステム全体の構成情報を格納します。このファイルは、InterSystems IRIS インスタンスの構成可能なほとんどの設定を含んでいるため、重要なツールです。既定の CPF はすべてのインスタンスで導入され、インストール・ディレクトリに配置されます。CPF は、テキスト・エディタで直接変更することも、管理ポータルまたはターミナルから間接的に変更することもできます。UNIX® および Linux では、構成マージ・ファイルを指定することにより、導入時に CPF をカスタマイズすることもできます。InterSystems IRIS は、最初にインスタンスを起動する前に、このファイルを使用して既定の CPF を更新します。CPF とそのパラメータの詳細は、“[構成パラメータ・ファイルの概要](#)”を参照してください。構成マージ機能の詳細は、“[構成マージを使用した InterSystems IRIS の自動構成](#)”を参照してください。

新たにインストールしたインスタンスでは、いくつかの開始設定を変更する必要があります。その他の設定については再確認してください。ここでは、最初に検討する以下の設定について説明します。

2.1 メモリと開始設定

初めて InterSystems IRIS をインストールした場合は、さまざまな構成設定の中でも特に、メモリ割り当てを確認し、調整する必要があります。InterSystems IRIS インスタンスのメモリの使用方法を決定する際に必要になるアクションは主に次の 3 つです。

- ・ [メモリと開始設定] ページ ([システム管理]→[構成]→[システム構成]→[メモリと開始設定]) で、データベース・キャッシュおよびルーチン・キャッシュにメモリを割り当てます。“[データベース・キャッシュおよびルーチン・キャッシュへのメモリの割り当て](#)”を参照してください。
- ・ InterSystems IRIS プロセスが使用できるメモリの最大量を設定します。“[プロセス当たりの最大メモリの設定](#)”を参照してください。
- ・ 共有メモリ・ヒープのサイズを構成します。これにより、ルーチン・キャッシュとデータベース・キャッシュ以外の目的で InterSystems IRIS が使用できるメモリ・サイズが決まります。“[共有メモリ・ヒープ \(gmheap\) の構成](#)”を参照してください。

重要

データベースおよびルーチンのキャッシュ・メモリの割り当ては、InterSystems IRIS の最初のインストール時に [自動] に設定されます。この設定では、InterSystems IRIS は物理メモリ合計の 25% をデータベース・キャッシュ (グローバル・バッファ) に割り当てます。

この設定は、実稼働環境での使用には適していません。InterSystems IRIS のインスタンスのルーチン・キャッシュとデータベース・キャッシュおよび共有メモリ・ヒープへのメモリ割り当てのガイドラインは、“[InterSystems IRIS 共有メモリの構成](#)”を参照してください。

[メモリと開始設定] ページには、メモリ設定の他に以下の設定が含まれています。

- ・ **システム開始時に自動開始** – Windowsシステムの場合、InterSystems IRIS インスタンスは、既定で、ホスト・システムの開始時に自動的に開始されるように構成されています。このチェック・ボックスのチェックを外すことにより、この動作を変更し、システムの開始時にこのインスタンスが自動的に開始されないようにすることができます。

UNIX® システムの場合、InterSystems IRIS はこのオプションを提供していないので、ホスト・システムが再起動しても InterSystems IRIS は再起動しません。ホスト・システムの再起動に伴って InterSystems IRIS も再起動できるようにするには、使用している環境に合わせて OS レベルのスクリプトを使用します。

注釈 ホスト・システムがダウンしているとき、ホスト・システムと共に IRIS が起動しないようにすることが必要な場合もあります (例えば、[リストA](#)中である場合)。これは、OS レベルでのみ設定でき、手順は構成によって異なります。

- ・ **スーパーサーバポート番号** – スーパーサーバ・ポートは、InterSystems IRIS インスタンスが着信するクライアント要求を受け入れるために使用する TCP ポートです。この設定の変更 (多くのクライアントがこのポートを使用してこのインスタンスに接続するよう構成している可能性があるため、この変更は慎重に行う必要があります) は、インスタンスを再起動するまで適用されません。
- ・ **システムモード** – 管理ポータルへのヘッダに表示されるラベルを入力するか、ドロップダウン・リストからいずれかを選択することができます。

[保存] をクリックして、これらの設定に対する変更を保存します。

重要 このページで実行する変更には、InterSystems IRIS の再起動が必要なものと、必要でないものがあります。再起動が必要な設定を変更し、それを含めた複数の変更を保存した場合、それらはすべて (それ自体、再起動が必要でない変更であっても)、InterSystems IRIS を再起動するまで適用されません。再起動が必要な場合は、「[再起動が必要な設定](#)」というメッセージが表示されます。ページを閉じると再起動が必要であることを知らせる警告メッセージは再表示されないため、直ちにインスタンスを再起動することをお勧めします。

2.1.1 データベース・キャッシュおよびルーチン・キャッシュへのメモリの割り当て

データベース・キャッシュは、データをバッファリングする目的で割り当てるシステム・メモリであり、グローバル・バッファ・プールと呼ばれることもあります。ルーチン・キャッシュは、ルーチンをバッファリングする目的で割り当てるシステム・メモリです。

InterSystems IRIS を初めてインストールすると、データベース・キャッシュの構成は [初期] に設定されます。この設定では、全システム物理メモリの 25% がデータベース・キャッシュ (グローバル・バッファ) に割り当てられます。この設定は、実稼働環境での使用には適していません。実稼働環境での使用にシステムをデプロイする前、または実稼働環境での使用のシミュレーションを目的としたテストやベンチマーキングを実行する前に、以下の手順に従って、データベース・キャッシュに適したメモリ割り当てを手動で作成する必要があります。

ルーチン・キャッシュの構成は既定で [自動] に設定されています。この設定では、バッファ・サイズを既定の 8 KB としたデータベース・キャッシュの 10% に等しいメモリ量がルーチン・キャッシュに割り当てられます。ただし、80 MB を最小値、1020 MB を最大値とします。一般的な実稼働環境でのインスタンスでは、データベース・キャッシュのサイズを適切に構成していれば、ルーチン・キャッシュへの割り当ては自動割り当てで十分です。しかし、個々の用途に理想的な割り当ては多くの要因に左右されるので、最適なパフォーマンスを実現するには割り当ての調整が必要になることもあります。

InterSystems IRIS のインスタンスでデータベース・キャッシュやルーチン・キャッシュなどにメモリを割り当てるためのガイドラインは、「[メモリ要件の見積もり](#)」を参照してください。

データベース・キャッシュとルーチン・キャッシュにメモリを割り当てるには以下の手順に従います。

1. 管理ポータルで、[メモリと開始設定] ページ ([システム管理] → [構成] → [システム構成] → [メモリと開始設定]) に移動します。

2. **[データベースキャッシュ(グローバルバッファ)を構成]** で、**[量の指定]** を選択してデータベース・キャッシュへのメモリ割り当て量 (MB) を入力します。

[開始設定] ページ (**[システム管理]**→**[追加設定]**→**[開始]**) の **DBSizesAllowed** 設定を使用してデータベースのブロック・サイズに既定の 8 KB 以外を有効にしている場合は、有効にしているブロック・サイズごとの割り当て量を入力します。ブロック・サイズごとのメモリ範囲がページに表示されます。ブロック・サイズと利用可能なバッファの最大数は、どちらもパフォーマンスに影響します。特定のブロック・サイズで InterSystems IRIS によって作成されるグローバル・バッファの数を判断するには、ブロック・サイズへの割り当て量をブロック・サイズで除算します。アプリケーションに適したブロック・サイズ選択のガイドラインについては、“[ラージ・ブロック・サイズに関する考慮事項](#)”を参照してください。

重要 大規模な ECP システムを構成する場合、ECP 経由で 8 KB ブロックを提供する 8 KB バッファに加え、この構造に最低でも 50 MB の 8 KB バッファを割り当ててください。“[ECP 制御構造用のデータ・サーバ・データベース・キャッシュの増大](#)”を参照してください。

3. **[ルーチンキャッシュを構成]** では、バッファ・サイズを既定値の 8 KB としたデータベース・キャッシュの 10% に等しいメモリ量が割り当てられる既定の設定である**[自動]**をそのまま使用できるほか、**[量の指定]**を選択してルーチン・キャッシュの割り当てを MB 単位で入力することもできます。

データベース・キャッシュとルーチン・キャッシュの指定方法の詳細は、“[構成パラメータ・ファイル・リファレンス](#)”の“[globals](#)”と“[routines](#)”を参照してください。

2.1.2 プロセス当たりの最大メモリの設定

[メモリと開始設定] ページの**[プロセスあたりの最大メモリ (KB)]** 設定は、この InterSystems IRIS インスタンスで実行される新規プロセスに割り当てることのできる最大メモリを指定します。有効値の範囲は、256 KB から 2,147,483,647 KB です。既定値は、**bbsiz** パラメータの初期値です。ほとんどの状況で、これは -1 (これは最大値に解決されます) に設定することをお勧めします。

注釈 この値は、既定値未満に設定していない限り、設定し直す必要はありません。〈STORE〉エラーが報告された場合は、サイズを増やして、エラーが発生したプロセスを再起動します。

シンボル・テーブル割り当ておよびその他メモリ要件 (I/O デバイス・アクセス構造体やバッファなど) で使用するプロセス・プライベート・メモリは、アプリケーションにおける必要性に応じて指定された最大値に達するまで増加します。一度プライベート・メモリがプロセスに割り当てられると、プロセスが終了するまで、その割り当ては解除されません。

プログラムによって設定を変更する方法を含め、この設定の詳細は“[bbsiz](#)”を参照してください。InterSystems IRIS のプロセス・メモリの詳細は、“[インターシステムズ製品のプロセス・メモリ](#)”を参照してください。

2.1.3 共有メモリ・ヒープ (gmheap) の構成

共有メモリ・ヒープは、ルーチン・キャッシュとデータベース・キャッシュ以外の目的で InterSystems IRIS によって使用されるメモリです。この設定は、**[メモリと開始設定]** ページにはありません。**gmheap** の構成は、**[メモリ詳細]** ページ (**[システム管理]**→**[構成]**→**[追加の設定]**→**[メモリ詳細]**) またはプログラムで実行できます。詳細は、“[構成パラメータ・ファイル・リファレンス](#)”の“[gmheap](#)”を参照してください。

gmheap で使用されているメモリと使用できるメモリの詳細を確認するには、**[共有メモリヒープ使用状況]** ページ (**[システムオペレーション]**→**[システム使用]**) に移動し、**[共有メモリヒープ]** リンクをクリックします。詳細は、“[監視ガイド](#)”の“[管理ポータルを使用した InterSystems IRIS の監視](#)”の章で“[共有メモリ・ヒープ使用状況](#)”を参照してください。

アプリケーションで多数の SQL クエリを使用する場合、または並列デジャナリングを有効にすることを計画している場合は、追加メモリを **gmheap** に割り当てる必要があります。詳細は、“[共有メモリの考慮事項](#)”および“[並列デジャナリングのシステム要件](#)”を参照してください。

2.2 IPv6 のサポート

[開始設定] ページ ([システム管理]→[構成]→[追加設定]→[開始]) に移動し、[IPv6] 行で [編集] をクリックすることで、InterSystems IRIS での IPv6 アドレスの使用を有効または無効にできます。[IPv6] を選択すると、このオプションが有効になります。

注釈 このオプションは、この InterSystems IRIS インスタンスが接続されているネットワークが IPv6 アドレスを許可している場合にのみ表示されます。

IPv6 を有効にすると、InterSystems IRIS で、IPv6 アドレス、IPv4 アドレス、および DNS 形式のアドレス (ホスト名、ドメイン修飾子ありおよびなし) が受け入れられ、IPv6 を無効にすると、InterSystems IRIS で受け入れられるアドレスが IPv4 アドレス、および DNS 形式のアドレスのみになります。

ドットで区切られた十進数形式の IPv4 アドレス (例 : 192.29.233.19) が指定されている場合は IPv4 接続が試行され、コロンで区切られた IPv6 アドレス (例 : 2001:fece:ba23:cd1f:dcbl:1010:9234:4085) が指定されている場合は IPv6 接続が試行されます。DNS 名 (例 : mycomputer.myorg.com) が指定されている場合は、実際の IP アドレスに解決されます。初めに IPv4 接続が試行され、IPv4 接続が確立できない場合は IPv6 接続が試行されます。

InterSystems IRIS では DNS、IPv4、および IPv6 形式でインターネット・アドレスを指定できます。例えば、localhost、127.0.0.1、::1 は、ループバック・アドレスのそれぞれの形式の表現です。IPv6 アドレスの詳細は、以下の Internet Engineering Task Force ドキュメントを参照してください。

- ・ [IP Version 6 Addressing Architecture \(RFC 4291\)](#)
- ・ [Application Aspects of IPv6 Transition \(RFC 4038\)](#)
- ・ [Format for Literal IPv6 Addresses in URL's \(RFC 2732\)](#)

IPv6 アドレスも、(現在のプロセスの場合は) %SYSTEM.Process クラスの IPv6Format メソッド、(システム全般の場合は) Config.Startup クラスの IPv6 メソッドを使用して確認し、制御できます。

InterSystems IRIS インスタンスが IPv4 ネットワークを使用している場合、指定された IPv6 アドレスに有効な IPv4 の等価なアドレスがあれば、さまざまなサービスの入力として IPv6 アドレスを使用することもできます。このセクションの冒頭で使ったループバック・アドレスなどがその例です。“RFC 4291”では、さらにいくつかの形式について説明しています。そのため、接続されているネットワークで使用できるように指定されたアドレス形式を正しく変換できれば、各種 InterSystems IRIS サービスはエラーが発生することなく IPv4 または IPv6 アドレスのいずれかを受理できます。以下のすべての形式 (とその他いくつかの形式) は受理されます。

- ・ localhost (DNS)
- ・ 127.0.0.1 (IPv4)
- ・ ::FFFF:127.0.0.1 (IPv4 射影 IPv6 形式)
- ・ 0:0:0:0:0:0:1 (完全な IPv6)
- ・ ::1 (圧縮 IPv6)

これらはすべてループバック・アドレスの有効な表現です。

一般に、従来の InterSystems IRIS サービスに指定されたインターネット・アドレスを求められても、InterSystems IRIS はアドレス形式を変更しません。IPv4 または IPv6 形式で指定されたアドレスは、それぞれ IPv4 または IPv6 として返されます。唯一の例外として、ホスト名として指定され、ドメイン・ネーム・サーバ (DNS) で変換されたアドレスは、DNS が返す形式で返されます。

注釈 InterSystems IRIS では、IPv6 アドレスの指定にワイルドカード文字は使用できず、範囲による指定もできません。

3

ネームスペースの構成

ここでは、InterSystems IRIS® データ・プラットフォームでネームスペースを構成する方法について説明します。背景情報は、“[ネームスペースとデータベース](#)”を参照してください。“[ネームスペースへのマッピングの追加](#)”も参照してください。

管理ポータル の **[ネームスペース]** ページでネームスペースを構成できます。目的のページに移動するには、ホーム・ページで **[システム管理]** を選択してから、**[構成]**、**[システム構成]**、**[ネームスペース]** の順に選択します。

1 つの InterSystems IRIS インスタンス内のネームスペース数の絶対上限は、2048 です。ネームスペース・テーブルのサイズを構成することはできません。

3.1 ネームスペース名の規則

ネームスペース名には以下の規則が適用されます。

- ・ 1 文字から 64 文字の範囲内に収まること。
- ・ 最初の文字はアルファベット文字またはパーセント記号 (%) にすること (それに続く文字には、英数字、ダッシュ、またはアンダースコアが使用可能)。

システム提供ネームスペースの詳細は、“[システム提供ネームスペース](#)”を参照してください。

3.2 ネームスペースの作成/変更

ネームスペースを作成するには、**[新規ネームスペース作成]** をクリックして **[新規ネームスペース]** ページを表示してから、以下の手順を実行します。

1. ネームスペースの **[名前]** を入力します。“[ネームスペース名の規則](#)”を参照してください。
2. **[コピー元]** を使用して既存のネームスペースからコピーして、選択したネームスペースの複製を作成することができます。この場合、後述のステップ 6 で説明されている Web アプリケーション用チェック・ボックス以外のオプションは、すべて使用できなくなります。
3. グローバルの既定のデータベースを **[ローカルデータベース]** または **[リモートデータベース]** のどちらにするか選択します。

4. [グローバルに既存のデータベースを選択] で、このネームスペースの既定のグローバル・マッピングを選択するか、[新規データベース作成] をクリックします。[新規データベース作成] をクリックすると、データベース・ウィザードまたはリモート・データベース・ウィザードのいずれかが起動します。
5. オプションで、ルーチンの既定のデータベースを[ローカルデータベース] または[リモートデータベース] のどちらにするか選択できます。その後で、[ルーチンに既存のデータベースを選択] ドロップダウンを使用して、このネームスペースの既定のルーチン・マッピングのデータベースを選択するか、[新規データベース作成] をクリックします。[新規データベース作成] をクリックすると、データベース・ウィザードまたはリモート・データベース・ウィザードのいずれかが起動します。
6. このネームスペースにアクセスする Web アプリケーションを作成する場合は、[このネームスペースの既定の Web アプリケーションを作成する] チェック・ボックスにチェックを付けます。
7. 必要に応じて、[相互運用プロダクション用にネームスペースを有効化] チェック・ボックスのチェックを外すことができます。詳細は、下記の“相互運用対応ネームスペースの作成”を参照してください。
8. 必要な情報を入力したら、[保存] をクリックし、構成にネームスペースを追加します。

3.2.1 相互運用対応ネームスペースの作成

InterSystems IRIS インスタンスでネームスペースを作成する場合、[ネームスペースを相互運用プロダクション対応にする] チェック・ボックスが[新規ネームスペース] ページの最下部に表示され、自動的にチェックが付きます。相互運用対応でないネームスペースを作成するには、このチェック・ボックスをクリアしてから[保存] をクリックします。

このチェック・ボックスをクリアせずに相互運用対応のネームスペースを作成すると、以下のように、システムにより自動的にこの新規ネームスペース用の追加構成タスクが実行されます。

- ・ このネームスペースの既定グローバル・データベースが既存のデータベースである場合、そのデータベースで一部のクラスのアップグレードとリコンパイルが実行されます。

注意 他のネームスペースでもこのデータベースを使用している場合、この変更が適切でないことがあります。InterSystems IRIS インスタンスで新規ネームスペースを作成する場合は、既存データベースの再使用がこのネームスペースに対して適切であるかどうか注意深く検討してください。

- ・ これにより、グローバル・マッピング、ルーチン・マッピング、およびパッケージ・マッピングが定義され、InterSystems IRIS システムのクラスおよびデータがこのネームスペースで使用可能になります。
- ・ ノードが %SYS グローバルに追加されます。
- ・ InterSystems IRIS で必要となるアプリケーション名 /csp/namespace を使用して、このネームスペースの Web アプリケーションが作成されます。

3.2.2 医療用途の相互運用性向けの Foundation ネームスペースの作成

医療用途の相互運用性向けに Foundation ネームスペースと Foundation プロダクションを作成するには、InterSystems IRIS for Health™ のインストーラ・ウィザードまたは HealthShare® HealthConnect のインストーラ・ウィザードを使用します。[新規ネームスペース] ページは使用しません。インストーラ・ウィザードで Foundation ネームスペースを有効にしておくと、後でルーチン・データベースなどのオプションの変更が必要になった場合でも[ネームスペース] ページを使用して処理できます。

3.3 ネームスペースの名前変更または既定マッピングの変更

以下の手順を実行することで、InterSystems IRIS を再起動せずにネームスペースの名前変更や、ネームスペースがマップされるデータベースの変更を行うことができます。

1. [ネームスペース] ページに移動します ([システム管理]→[構成]→[システム構成]→[ネームスペース])。
2. [ネームスペース] ページで、変更するネームスペースの名前をクリックします。
3. 既存の名前を変更または置換してネームスペースの名前を変更します。

重要 相互運用対応のネームスペースの名前を変更している場合、プロセスを完了するには追加手順が必要となります。

- a. システム・トレイから InterSystems ターミナルを開きます。
- b. 以下のように入力します。

```
do ##class(%Library.EnsembleMgr).EnableNamespace("<NewNamespace>",1)
```

<NewNamespace> は既存のネームスペースの新しい名前です。

- c. [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。
- d. ネームスペースの古い名前に対応するアプリケーションの名前を検索し、[削除] をクリックします。
- e. ネームスペースの新しい名前に対応するアプリケーションの名前をクリックします。
- f. [ネームスペースのデフォルト・アプリケーション] を選択し、[保存] をクリックします。
- g. ターミナルで、次のように入力します。

```
do ##class(%EnsembleMgr).DisableNamespace("<OldName>",1)
```

<OldName> は、名前を変更しているネームスペースの元の名前です。

4. 定義済みのデータベースのリストから [グローバルの既定のデータベース]、[ルーチンの既定のデータベース]、および [一時的な格納場所の既定のデータベース] を選択します。

注釈 [仮ストレージのデフォルトデータベース] ドロップダウン・リストからジャーナル・グローバル以外 (つまり、[ジャーナルグローバル] プロパティを [いいえ] に設定) で構成されたデータベースを選択することは、IRISTEMP の選択と同じではありません。詳細は、"[一時グローバルと IRISTEMP の使用](#)" を参照してください。

5. [保存] をクリックします。

変更時にデータベースに直接アクセスしているユーザは、ネームスペース・マッピングを更新するために、一度 InterSystems IRIS をログアウトして、再度ログインする必要がある場合もあります。

3.4 ネームスペースの削除

ネームスペースとそれに関連付けられたすべてのマッピングを次の手順で削除できます。

1. [ネームスペース] ページに移動して ([システム管理]→[構成]→[システム構成]→[ネームスペース])、削除するネームスペースの行で [削除] をクリックします。
2. 物理パスから Web ゲートウェイ・ページを削除するには、[ネームスペースの削除] ページで該当のチェック・ボックスにチェックを付けます。
3. ネームスペースとそれに関連付けられたマッピングを削除するには、[すぐに実行する] をクリックします。

3.5 プログラムによるネームスペースの構成

InterSystems IRIS には、ネームスペース構成の自動化に役立つ、プログラムによるツールも用意されています。

Config.Namespaces クラスを使用して、ネームスペースを作成および構成できます。詳細は、“インターシステムズ・クラス・リファレンス” を参照してください。

ネームスペースを構成するもう 1 つの方法として、構成マージと組み合わせて [CreateNamespace](#)、[ModifyNamespace](#)、または [DeleteNamespace](#) の操作を含めることもできます。構成マージにより、InterSystems IRIS インスタンスに適用する設定や操作を指定する宣言型マージ・ファイルを適用することで、このインスタンスをカスタマイズできます。この機能の詳細は、“[構成マージを使用した InterSystems IRIS の自動構成](#)” を参照してください。

3.6 関連項目

- ・ [ネームスペースとデータベース](#) (システム定義のネームスペースおよびデータベースに関する情報が含まれています)
- ・ [ネームスペースへのマッピングの追加](#)
- ・ [構成マージを使用した InterSystems IRIS の自動構成](#)

4

ネームスペースへのマッピングの追加

ネームスペースの構成の一部として、同じシステムまたは異なるシステムに存在する、他のデータベースのグローバル、ルーチン、およびクラス・パッケージをマップすることもできます。

これによって、さまざまな場所に存在するデータを、簡単に参照できるようになります。グローバル全体でも一部でもマップできます。これによって、データはディスクを簡単に行き来できます。

注釈 マッピングはアルファベット順に並べ替えられ、添え字が指定されている場合は、名前と添え字を基準にして並べ替えられます。詳細は、“[グローバルについての正式な規則](#)”を参照してください。

重要 マップされたコンテンツの識別子がローカル・コンテンツ（パッケージ、クラス、グローバル、またはルーチン名）と同じ場合、ローカル・コンテンツではなくマップされたコンテンツが表示されます。そのため、マッピングが既存のコンテンツから切り離されないよう、マッピングはできる限り具体的にする必要があります。

コンテンツを命名したり、マッピングを作成する際は、不要な競合を避けるため、必ず“[識別子のルールとガイドライン](#)”に従ってください。

4.1 グローバル・マッピング

グローバルおよびグローバル・サブスクリプト・レベルで、ネームスペースに新規グローバルのマッピングを追加し、そのマッピングをそのネームスペースのグローバルに対する既定のデータベース・マッピングよりも優先させるようにできます。

1. [ネームスペース] ページに移動して ([システム管理]→[構成]→[システム構成]→[ネームスペース])、グローバルをマッピングするネームスペースの行で [グローバルマッピング] をクリックします。
2. [グローバルマッピング] ページで、[新規] をクリックします。
3. グローバルが配置されるデータベースの [グローバル・データベースの位置] を選択します。
4. [グローバル名] を入力します。ABC* のようにグローバル名の一部として * 文字を使用すると、複数のグローバルを指定できます。
5. [マップするグローバルサブスクリプト] を入力します。添え字参照は開き括弧で始まる必要があります。以下はその例です。

```
(1)
( "A" )
(1):(5)
( "A" ):( "Z" )
( "B" , 23 , "m" ):( "E" , 5 )
(BEGIN):( "X" )
( "Y" ):(END)
```

注釈 範囲(例: ("A") : ("Z"))を指定する場合、指定された添え字の範囲は“～から、～以内”ではなく“～から、～未満”になります。つまり、定義された添え字範囲の下限は含まれますが、上限は含まれないことになります。例えば、Name (1) : (10) の場合、Name (1) は含まれますが、Name (10) は含まれません。範囲の上限を除外することによって、添え字の範囲を Name ("a") : ("b") のように指定して処理すると、Name ("aa") および Name ("aaaaa") を含め Name ("b") の直前までを有効範囲とする上限境界を指定できます。

予約語 BEGIN および END を使用して、使用可能な先頭および最後の添え字を示すことができますが、グローバルの添え字は個別にマッピングする必要があるため、添え字のあるグローバルで アスタリスク (*) を使用することはできません。

添え字レベルのマッピング (SLM) の範囲に関する詳細は、“[Setting Up Global Mappings](#)” を参照してください。

6. [詳細] をクリックすると、以下のオプションが表示されます。

- a. [照合] を選択します。照合は新規添え字レベルのマッピング・グローバルのみに適用されます。
- b. [ロックデータベース位置] を選択します。詳細は、“構成パラメータ・ファイル・リファレンス” の “[Map]” セクションの “[Global](#)” を参照してください。

7. [OK] をクリックします。

注釈 新規のマッピング行の最初の列に表示される [>>] は、マッピングを編集可能で開いていることを示します。

8. マッピングを cpf ファイルに保存するには、[変更を保存] をクリックします。

重要

既存のグローバルのデータベースの場所を変更するマッピングを追加することもできますが、これは実際にはグローバルを移動させません。グローバルは元のデータベース内にあるものの、ネームスペースはそのグローバルを新しくマップされたデータベース内で見つけようとするため、結果としてグローバルはアクセスできなくなります。既存のグローバルの新規マッピングを成功させるには、グローバルを手動で再配置する必要があります。これは例えばターミナルや管理ポータルを使用して、グローバルを新規データベース上で作成し、元のデータベースから削除することで実行します。

4.2 ルーチン・マッピング

ルーチン・レベルでネームスペースにマッピングを追加すると、そのマッピングでそのネームスペースのルーチンに対する既定のデータベース・マッピングをオーバーライドできます。

1. [ネームスペース] ページに移動して ([システム管理] → [構成] → [システム構成] → [ネームスペース])、グローバルをマッピングするネームスペースの行で [ルーチンマッピング] をクリックします。
2. [ルーチンマッピング] ページで、[新規] をクリックします。
3. ルーチンが配置される [ルーチン・データベースの場所] のデータベースを選択します。
4. [ルーチン名] を入力します。このルーチンは、マッピングの時点で存在していなくてもかまいません (つまり、これから作成するルーチンの名前を使用できます)。
5. [OK] をクリックします。

注釈 新規のマッピング行の最初の列に表示される [>>] は、マッピングを編集可能で開いていることを示します。

6. マッピングを cpf ファイルに保存するには、[変更を保存] をクリックします。

例えば、前出の[テスト・ネームスペース・マッピング](#)を例にして考えてみましょう。airports データベース (FlightSchedule ネームスペース内) でスケジュール・ルーチン (例えば BOSZZairline) を作成し、TravelAgent ネームスペースのユーザに利用を許可する計画がある場合、[\[ルーチンマッピング\]](#) ページ (TravelAgent ネームスペース行) に移動して、[\[新規ルーチンマッピング\]](#) をクリックします。下図の [\[ルーチン・マッピング\]](#) ダイアログ・ボックスに情報を入力します。

重要 1 つ以上のルーチンをマップするときには、それらのルーチンが必要とするすべてのコードとデータを必ず特定して、そのすべてのコードとデータがすべてのターゲット・ネームスペースで使用できることを確認します。マップされるルーチンは、以下の項目に依存している可能性があります。

- ・ インクルード・ファイル
- ・ その他のルーチン
- ・ クラス
- ・ テーブル
- ・ グローバル

追加のルーチン、パッケージ、およびグローバル・マッピングを必要に応じて使用して、これらの項目がターゲット・ネームスペースで使用できるようにします。

4.3 パッケージ・マッピング

クラス・パッケージ・マッピングを追加すると、特定のデータベースのパッケージにあるすべてのクラス (および、それらのクラスの生成されたすべてのルーチン) を、別のネームスペースから認識できるようになります。

1. [\[ネームスペース\]](#) ページに移動して ([\[システム管理\]](#)→[\[構成\]](#)→[\[システム構成\]](#)→[\[ネームスペース\]](#))、パッケージをマッピングするネームスペースの行で [\[パッケージマッピング\]](#) をクリックします。
2. [\[パッケージマッピング\]](#) ページで、[\[新規\]](#) をクリックします。
3. パッケージが配置される [\[パッケージデータベース位置\]](#) のデータベースを選択します。
4. [\[パッケージ名\]](#) を選択します。このパッケージは、マッピングの際に存在していなくてもかまいません (つまり、作成を計画しているパッケージの名前を使用できます)。新規パッケージ名を指定する手順は以下のとおりです。
 - a. [\[新規パッケージ\]](#) をクリックします。
 - b. [\[新規パッケージ名\]](#) テキスト・ボックスに名前を入力します。
5. [\[OK\]](#) をクリックします。

注釈 新規のマッピング行の最初の列に表示される [\[>>\]](#) は、マッピングを編集可能で開いていることを示します。

6. マッピングを `cpf` ファイルに保存するには、[\[変更を保存\]](#) をクリックします。

パッケージの説明とそのマッピング手順については、["パッケージ・マッピング"](#) を参照してください。

重要

パッケージをマップするときには、そのパッケージ内のクラスが必要とするすべてのコードとデータを特定して、そのすべてのコードとデータがすべてのターゲット・ネームスペースで使用できることを必ず確認します。マップされるクラスは、以下の項目に依存している可能性があります。

- ・ インクルード・ファイル
- ・ ルーチン
- ・ その他のクラス
- ・ テーブル
- ・ グローバル

追加のルーチン、パッケージ、およびグローバル・マッピングを必要に応じて使用して、これらの項目がターゲット・ネームスペースで使用できるようにします。

InterSystems IRIS は、1 つの動作で[すべてのネームスペース](#)にわたって同じパッケージを使用できるようにする機能を備えています。

4.4 すべてのネームスペースへのマッピング

グローバル、ルーチン、およびパッケージは、特定のネームスペースにマッピングされるだけでなく、すべてのネームスペースにマッピングすることもできます。この形式のマッピングを有効にするには、以下を行います。

1. まず、**%ALL** という名前のネームスペースを作成します ("[ネームスペースの作成/変更](#)" を参照)。

注釈 **%ALL** は、データのマッピング時以外は表示されません。つまり、これは現実のネームスペースではなく、すべてのネームスペースにデータをマッピングするための機能です。

2. 次に、**%ALL** ネームスペースで目的のマッピングを行います ("[ネームスペースへのグローバル、ルーチン、およびパッケージ・マッピングの追加](#)" を参照)。

これらの **%ALL** マッピングは、すべてのネームスペースに適用されます。**%ALL** ネームスペースでマップされているリソースに対する、ネームスペース固有のマッピングを作成することはできません。**%ALL** のマッピングが、同じリソースへのすべてのネームスペース固有のマッピングをオーバーライドするためです。

注意

%ALL のマッピングは、**%SYS** を含むすべてのネームスペースに適用されます。インスタンスが使用するルーチンまたはグローバルを上書きするマッピングを作成すると、特定の機能が壊れる可能性があります。

%ALL を使用する場合、マッピングはできる限り限定的にする必要があります。また、"[識別子のルールとガイドライン](#)" に厳格に従ってください。特に、"[回避する必要があるグローバル変数名](#)" にリストされたグローバルに対して **%ALL** のマッピングを作成しないでください。

%ALL ネームスペースに添え字レベルのマッピングを作成する場合、ルート・グローバルのマッピングが **%DEFAULTDB** に自動的に作成されます。**%DEFAULTDB** 変数は、任意のネームスペースの既定のデータベースを表します。

4.5 関連項目

- ・ [ネームスペースとデータベース](#)

- ・ [ネームスペースの構成](#)
- ・ [構成マージを使用した InterSystems IRIS の自動構成](#)

5

ローカル・データベースの構成

ここでは、InterSystems IRIS® データ・プラットフォームでのローカル・データベースの構成について説明します。ほとんどのデータベース構成は、動的に変更できることに注意してください。つまり、システムの実行中に、データベースを作成および削除したり、データベース属性を修正することができます。

5.1 データベースに関する考慮事項

背景情報は、“[ネームスペースとデータベース](#)” を参照してください。ここでは、システム管理者に関連した追加の概念について説明します。

注意 Windows システムでは、IRIS.DAT データベース・ファイルに対してファイル圧縮機能を使用しないでください(Windows エクスプローラでファイルまたはフォルダを右クリックし、[プロパティ]、[詳細]、[内容を圧縮してディスク領域を節約する]の順に選択して、ファイルを圧縮します。圧縮後、Windows エクスプローラでフォルダ名またはファイル名が青で表示されます)。IRIS.DAT ファイルを圧縮した場合、このファイルが属するインスタンスは開始できません。これは誤解を招きやすいエラーです。

以降のセクションでは、ローカル・データベースを作成、構成、および維持する方法について詳しく説明します。補足的な資料として、“[Managing Database Size](#)” では、データベースのサイジングと拡大に関する概要の簡単なビデオを提供しています。

5.1.1 データベースのサイズと増大

InterSystems IRIS データベースには最大サイズを指定できますが、その範囲でサイズは必要に応じて動的に変化します(空き領域が利用可能な場合)。既定の 8KB のブロック・サイズを使用する場合、データベースは 32 テラバイトまで増大する可能性があります。

5.1.2 データベース・キャッシュ

それぞれの InterSystems IRIS システムは、データベース・キャッシュを保持します。データベース・キャッシュとは、ローカルの共有メモリ・バッファで、物理データベースから取得されるデータをキャッシュするために使用します。このキャッシュは、データ・アクセスに必要な入出力処理の量を大幅に削減し、InterSystems IRIS の性能を大幅に向上させます(データベース・キャッシュの割り当てに関する詳細は、“[メモリと開始設定](#)” を参照してください)。

5.1.3 データベースの合計に関する制限

単独の InterSystems IRIS インスタンスに構成可能なデータベース数の絶対的な制限値は 15,998 です (十分な格納領域がある場合)。そのほかにも、以下のような制限があります。

- データベースのディレクトリ情報が 256 KB を超えることはできません。つまり、データベース・ディレクトリ名の平均長が長い場合、インスタンスのデータベースの合計は少なくなります。以下の式で、この関係を示します。

$$\text{maximum_DBs} = 258048 / (\text{avg_DB_path_length} + 3)$$
 例えば、すべてのデータベース・ディレクトリ・パスの形式が `c:\¥InterSystems¥IRIS¥mgr¥DBNNNN¥` の場合、平均長は 33 バイトです。したがって、データベースの最大数は 7,168 (計算式: $258048 / (33 + 3) = 7168$) です。
- 絶対上限の 15,998 に向けてのカウントでは、ミラーリングされたデータベースは 2 回カウントされます。インスタンスのすべてのデータベースがミラーリングされている場合、実際の上限は 7,499 データベースです。これは、InterSystems IRIS はミラーリングされるデータベースに対し、ディレクトリ・パスに 1 つ (`c:\¥InterSystems¥IRIS¥mgr¥DBNNNN¥`) とミラー定義に 1 つ (`:mirror:MIRRORNAME:MirrorDBName`) の、計 2 つのデータベース定義を作成するためです。
- 同時に使用できるデータベースの数は、開いているファイル数に関するオペレーティング・システムの制約によって制限されます (プロセスごとまたはシステム全体)。InterSystems IRIS では、オペレーティング・システムの開いているファイルの割り当てのうち、約半数を自身での使用およびデバイス用に予約しています。

5.1.4 データベース構成に関する考慮事項

データベース構成時に考慮するヒントを以下に示します。

- InterSystems IRIS には、複数の物理データベース (IRIS.DAT) ファイルにおいてデータを分散するシームレス・オプションが用意されています。したがって、必要に応じてグローバル・マッピングや添え字レベル・マッピングでデータを分けたり、複数のデータベースによりアプリケーションを構築できます。
- バックアップ、リストア、整合性チェックなどの管理タスクで利用可能なインフラストラクチャに基づいて、管理可能な値にデータベース・サイズを維持します。
- 必要に応じて、[複数のボリューム](#)を使用するようにデータベースを設定できます。
- ストリーム・グローバル (ストリームを IRIS.DAT データベースのファイル内部に格納する場合) を別のデータベースにグローバル・マッピングして、ストリーム・データベースをラージ・ブロック・サイズ (64 KB) で構成することをお勧めします。
- 負荷によっては、既定の 8 KB データベース・ブロック・サイズよりも大きなブロック・サイズを検討した方が効果的な場合があります。一般的なガイドラインは、以下の ["ラージ・ブロック・サイズに関する考慮事項"](#) を参照してください。

5.1.5 ラージ・ブロック・サイズに関する考慮事項

InterSystems IRIS でサポートされている 8 KB (既定) ブロック・サイズ (常に有効になっています) に加えて、以下のブロック・サイズを使用できます。

- 16 KB (16384)
- 32 KB (32768)
- 64 KB (65536)

ただし、ラージ・ブロック・サイズを使用するデータベースを作成する場合は、システムのパフォーマンスに影響を与える可能性があるので注意してください。ラージ・ブロック・サイズを有効化して使用するには、以下の点を考慮します。

- アプリケーションのワークロードが主に連続的な挿入または連続的な読み取りやクエリによるものである場合、ラージ・ブロック・サイズを使用するとパフォーマンスが向上することがあります。

- ・ アプリケーションのワークロードが主にランダムな挿入またはランダムな読み取りやクエリによるものである場合、ラージ・ブロック・サイズを使用するとパフォーマンスが低下することがあります。ラージ・ブロック・サイズを使用すると、データベース・キャッシュに指定された全体のサイズに対してキャッシュされるブロック数が少なくなるため、データベースへのランダム・アクセス時の影響を軽減するには、データベース・キャッシュに使用できるメモリの全体量をより増やすように考慮する必要があります。
- ・ インデックス・タイプのデータベースの場合、既定のブロック・サイズ (8 KB) で最適なパフォーマンスを確保できます。ラージ・ブロック・サイズでは、パフォーマンスが低下する可能性があります。データ用としてラージ・ブロック・サイズを検討している場合は、インデックス・グローバルを別個の 8 KB ブロック・サイズのデータベースにマッピングするように考慮する必要があります。

サポートされているブロック・サイズ以外のブロック・サイズを使用するデータベースを作成するには、以下の手順を実行します。

1. **[開始設定]** ページ (**[システム管理]**→**[追加の設定]**→**[開始]**) の設定を使用してブロック・サイズを有効化します。“**DBSizesAllowed**” を参照してください。
2. “**メモリと開始設定**” で説明されているように、**[開始設定]** ページ (**[システム管理]**→**[追加設定]**→**[開始]**) で、有効化されたブロック・サイズのデータベース・キャッシュを構成します。
3. InterSystems IRIS を再起動します。
4. “**ローカル・データベースの作成**” の説明に従って、データベースを作成します。

5.1.6 データベースの互換性に関する考慮事項

“**ローカル・データベースの作成**” の手順で説明したように、InterSystems IRIS データベースは、そのデータベースの **IRIS.DAT** ファイルをコピーまたは移動することで、データベースを作成したインスタンス以外のインスタンスにコピーまたは移動できます。また、同一システム上の別のインスタンスで作成したデータベースを一時的にマウントすることもできます。さらに、データベースのバックアップをバックアップ元のインスタンスとは別のインスタンスにリストアすることもできます (“**バックアップとリストア**” を参照)。ただし、データの互換性がなくならないように、次の要件を満たしている必要があります。

- ・ ターゲット (新規) インスタンスは、データベースを作成したインスタンスと文字幅を同一にして (8 ビットまたは Unicode。“**文字幅設定**” を参照)、ロケールも同様に同一 (“**管理ポータル**の **NLS 設定ページの使用法**” を参照) にする必要があります。

この要件に対する例外の 1 つとして、ISO 8859 Latin-1 文字セットに基づくロケールを使用する 8 ビット・インスタンスは、対応する幅の文字ロケールを使用する Unicode インスタンスと互換性があります。例えば、**enu8** ロケールを使用した 8 ビット・インスタンスで作成されたデータベースは、**enuw** ロケールを使用した Unicode インスタンスで使用できます。

- ・ ソース・インスタンスとターゲット・インスタンスがエンディアンが異なるシステム上に存在する場合は、使用前に、データベースをターゲット・インスタンスのエンディアンに変換する必要があります。

マルチバイト・データが最下位メモリ・アドレス (つまり先頭) の最上位バイトと最下位バイトのいずれに格納されるかはプラットフォームによって異なります。最上位バイトから格納される場合は “**ビッグ・エンディアン**”、最下位バイトから格納される場合は “**リトル・エンディアン**” と呼ばれます。

異なるエンディアンのシステムで作成された既存の **IRIS.DAT** を使用してデータベースを定義する場合は、**cvendian** ユーティリティ (“**cvendian を使用したビッグ・エンディアン・システムとリトル・エンディアン・システム間の変換**” を参照) を使用して、使用前にデータベースを変換してください。ソース・システムとは異なるエンディアンのシステムに、データベースのバックアップをリストアする場合は、“**エンディアンに関する注意事項**” を参照してください。

5.2 データベース名の規則

データベース名には以下の規則が適用されます。

- ・ 1 文字から 64 文字の範囲内に収まること
- ・ 最初の文字はアルファベット文字またはアンダースコアにすること (それに続く文字には、英数字、ダッシュ、またはアンダースコアが使用可能)

システム提供データベースの詳細は、“[システム提供データベース](#)” を参照してください。

5.3 マルチボリューム・データベース

既定の構成では、データベースは単一の **IRIS.DAT** ファイルを使用します。指定されたサイズのしきい値に到達したら、自動的に追加のボリューム (**IRIS-0001.VOL**、**IRIS-0002.VOL** など) に拡張するようデータベースを構成できます。データベースを意図的に追加のボリュームに拡張することもできます。追加のボリュームは、**IRIS.DAT** と同じディレクトリや追加の一連のディレクトリに存在している場合があります。

複数のボリュームを使用しても、データベースが最大サイズを超えることはできません。データベース・ボリュームの合計サイズは通常の最大サイズに従う必要があります。代わりに、この機能によるオペレーティング・システムとストレージの管理性のために、データベースを一連のボリューム・ファイルの合計サイズまで (オプションとして複数のファイル・システムにまたがって) 拡張できるようになります。

ベスト・プラクティスは、ボリューム・サイズをオペレーティング・システムの制約内でできるだけ大きく保ち、データベースを含むファイル・システムを拡張することが現実的でない場合にのみ追加のディレクトリに拡張することです。通常、この機能は外部のサイズ制約により必要とされる場合にのみ使用してください。InterSystems IRIS 自体での操作を支援するためには必要ありません。マルチボリュームは、グローバルおよび添え字レベルのマッピングを使用したサイズ制約の管理の、より単純な代替手段となります。

データベース・ボリュームの作成、構成、および編成は、[管理ポータル](#)内で、[DATABASE ユーティリティ](#)を使用するか、**SYS.Database** クラスのメンバを介して管理できます。

5.3.1 マルチボリューム・データベースの構成

各データベースでは、新規ボリュームの自動作成を制御するため、新規ボリュームのしきい値サイズを構成できます。このしきい値がゼロ (既定) の場合、新規ボリュームは作成されません。そうでない場合、**IRIS.DAT** または現在のボリューム・ファイルがしきい値のサイズに到達すると、新規ボリュームが作成されます。

新規ボリュームのしきい値が構成されても、データベースの拡張サイズのパラメータは引き続き通常通り機能します。アプリケーションの要求によってトリガされる自動拡張により、データベースにこのサイズが追加されます。値が 0 の場合、システムでは合理的な拡張サイズが選択されます。新しいボリュームが自動拡張によって作成される場合、ボリュームはこの初期サイズで作成されます。新規ボリュームのしきい値になるよう初期化されることはありません。そのため、新規ボリュームのしきい値を適度に大きなサイズに設定しても、そうでない場合に比べデータベースがより大きな単位で拡張されることはありません。

新規ボリュームは、特定の構成可能な新規ボリューム・ディレクトリで作成されます。既定では、すべてのボリュームは **IRIS.DAT** を含むディレクトリ内で作成され、ここに配置されます。これは最も単純な構成です。ただし、ファイル・システムの制約により必要となる場合は、ボリューム・ファイルを別のディレクトリに分散することができます。

データベースでは、ボリューム・ディレクトリ・リストが保持されます。これは、このデータベースのボリュームで 사용되는すべてのディレクトリのリストです。データベース・ボリュームが既定以外のディレクトリに配置されていない場合、このリストは空になります。追加の各ボリューム・ディレクトリには **iris.dbdir** ファイルが含まれます。このファイルは、ボリューム・ディ

レクトリを使用しているデータベース・ディレクトリを識別し、複数のデータベースが同じボリューム・ディレクトリを使用することを防止する役割を果たします。このファイルは、プライマリ・データベース・ディレクトリを示す 1 行を含むテキスト・ファイルで、管理者はいつでもこれを検査できます。このファイルが削除されることはなく、再起動しても維持されます。また、データベースをマウントするために、想定されるコンテンツと共に存在する必要があります。追加のボリューム・ディレクトリには `iris.lck` ファイルがないことに注意してください。このファイルがあるのはプライマリ・ディレクトリのみです。

新規ボリュームは、同じディレクトリまたは異なるディレクトリ内で、管理者によりいつでも作成可能です。これは、新規ボリュームしきい値が設定されているかどうかは関係ありません。このように、ファイル・サイズ制限が指定されていなくても、必要な場合にはデータベースを別のファイル・システム、またはストレージ・デバイスに拡張することができます。

5.3.2 ボリューム・ファイルの移動またはコピー

ボリューム・ファイルを移動またはコピーするには、データベースをディスマウントするか、InterSystems IRIS を正常にシャットダウンする必要があります。

ボリューム・ディレクトリ・リストが空で、すべてのボリューム・ファイルが `IRIS.DAT` と同じディレクトリ内に作成されている場合、そのディレクトリのコンテンツは、1 つのボリュームの `IRIS.DAT` ファイルの場合と同じように移動またはコピーできます。

ボリューム・ファイルが 1 つ以上のその他のディレクトリに存在している場合、`IRIS.DAT` を含むディレクトリとデータベースで保持されるボリューム・ディレクトリ・リスト間で自由にデータベース・ボリュームを移動できます。次回データベースをマウントすると、これらのディレクトリ内のボリュームが検索され、それらが適切にマウントされます。

ただし、ボリューム・ファイルがボリューム・ディレクトリ・リストまたは `IRIS.DAT` を含むディレクトリ内に存在しない場所に移動される場合は、[DATABASE ユーティリティ](#)のオプション 16 [\[マルチボリューム・データベース・プロパティの管理\]](#)を使用して、データベースが正常にマウントされるようにボリューム・ディレクトリ・リストを更新する必要があります。

これは、`IRIS.DAT` を含むディレクトリの外にあるボリュームを含むデータベースが、異なるディレクトリ構造を持つ別のシステムにコピーされる場合、特に必要です。`DATABASE` を使用してボリューム・ディレクトリ・リストを変更する場合、更新されるリスト内の追加のボリューム・ディレクトリには、`iris.dbdir` ファイルを含めないようにするか、プライマリ・データベース・ディレクトリと既に一致しているものを含める必要があります。`iris.dbdir` ファイルがボリューム・ファイルと共に別のシステムからコピーされ、ターゲット・システムが別のディレクトリ構造を持つ場合、`iris.dbdir` ファイルは、ボリューム・ディレクトリ・リストを変更する前に削除する必要があります。

同様に、ボリューム・ファイルをより小さなディレクトリ・セットに分割する場合、管理者は `DATABASE` を使用してディレクトリ・セットを更新し、ファイル・システムから未使用のディレクトリを削除できます。

追加のボリュームを使用するかどうかに関係なく、`IRIS.DAT` ファイルおよびデータベースを構成するすべての `IRIS-*.VOL` ファイルは、セットとして扱う必要があります。それらはセットとしてコピーおよびリストアップする必要があります。よくあるミス回避し、ボリューム・セットに問題があるように見えるデータベースをマウントすることを防ぐため、一連の内部参照チェックがマウント時に自動的に実行されます。

ボリューム・ディレクトリ・リストの更新は、[DATABASE を使用して](#)のみ実行できます。管理ポータルには、このプロセスのサポートは含まれません。

5.3.3 新規ボリュームのエラー動作

- ・ 自動拡張の際に新規ボリュームの作成が失敗した場合、アプリケーションで `<FILEFULL>` エラーをトリガするのではなく、拡張が最後のボリュームで続行されます。
- ・ 新規ボリュームのしきい値に到達する前に最後のボリュームの拡張で領域が不足し、別の新規ボリューム・ディレクトリは既に指定されているという特殊なケースにおいては、新規ボリュームは新規ボリューム・ディレクトリで作成されて拡張は続行され、最後のボリュームはしきい値サイズより小さくなります。

- ・ 他のエラーによって、新規ボリュームのしきい値により定義されたパラメータおよび新規ボリューム・ディレクトリ外で、システムがボリュームを生成することはありません。特に、指定された新規ボリューム・ディレクトリ内の拡張が失敗した場合、システムがボリューム・ディレクトリ・リスト内のその他のディレクトリを自動的に試行することはありません。

5.3.4 [Multi-Volume Properties] ページ

この管理ポータルページでは、データベースを構成するボリュームのプロパティを表示および構成できます。このページを表示するには、[ローカルデータベース] ページ ([システム管理]→[構成]→[システム構成]→[ローカルデータベース]) に移動して、目的のデータベースに関連付けられた [Volumes] ボタンをクリックします。

[Multi-Volume Properties] ページには、以下の情報が表示されます。

- ・ **Directory** – データベースと関連付けられた IRIS.DAT ファイルを含むディレクトリ。
- ・ **Number of Volumes** – データベース内のボリュームの数。1 の値は、IRIS.DAT ファイルのみが使用中で、IRIS-*.VOL ファイルは存在しないことを意味します。
- ・ **Total Database Size** – すべてのボリュームを含む、データベースの合計サイズ。
- ・ **New Volume Threshold Size** – 現在のボリューム (または IRIS.DAT) が到達可能な最大サイズ。これに達すると、新規ボリュームの作成がトリガされます。
- ・ **Directory for New Volumes** – 新規ボリュームが作成されるディレクトリ。
- ・ **Database Volumes** – データベースを構成するボリュームのリスト。ボリューム番号ごとまたはこれらを含むディレクトリごとに編成されます。ボリューム 0 は通常 IRIS.DAT です。

5.4 ローカル・データベースの表示

ローカル・データベースを表示するには、[ローカルデータベース] ページ ([システム管理]→[構成]→[システム構成]→[ローカルデータベース]) に移動します。

[ローカルデータベース] ページには、システムのデータベースに関する以下の情報が表示されます。

- ・ **名前** – データベースの名前。
- ・ **ミラー** – データベースがミラーリングされている場合は、ミラーの名前。詳細は [“ミラーにデータベースを追加する”](#) を参照してください。
- ・ **ディレクトリ** – IRIS.DAT ファイルの場所。
- ・ **サイズ (MB)** – MB 単位でのデータベースのサイズ。
- ・ **ステータス** – データベースがマウントされているかマウントされていない (ディスマウントされている) かが示されます。マウントされている場合は、読み取り専用権限か読み取り/書き込み権限かが示されます。詳細は、[“ローカル・データベースの管理”](#) の “ローカル・データベース情報” の表を参照してください。
- ・ **リソース名** – 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。詳細は、[“リソースの使用による資源の保護”](#) を参照してください。
- ・ **暗号化** – データベースが暗号化されているかどうかを示されます。詳細は、[“暗号化ガイド”](#) を参照してください。
- ・ **ジャーナル** – データベースがジャーナルされているかどうかを示されます。詳細は、[“ジャーナリングの概要”](#) を参照してください。
- ・ **削除** – 現在のデータベースを削除します。

- ・ **グローバル** – [グローバル] ページを開きます。このページには現在のデータベースに関連付けられたグローバルが表示されます。
- ・ **Volumes** – [Multi-Volume Properties] ページを開きます。ここでは、データベースを構成するボリュームを表示および構成できます。詳細は、“[Multi-Volume Properties] ページ” を参照してください。

このページを使用すると、以下のことができます。

- ・ [ローカル・データベースの作成](#)
- ・ [ローカル・データベースのプロパティの編集](#)
- ・ [ローカル・データベースの再配置](#)
- ・ [ローカル・データベースの削除](#)

5.5 ローカル・データベースの作成

ローカル・データベースを作成するには以下の手順を実行します。

1. [ローカルデータベース] ページ ([システム管理]→[構成]→[システム構成]→[ローカルデータベース]) に移動します。
2. [新規データベース作成] をクリックし、[データベース] ウィザードを起動します。
3. テキスト・ボックスにデータベース名を入力します。データベース名は、InterSystems IRIS インスタンス内で使用されているものであってはなりません。“[データベース名の規則](#)” も参照してください。
4. 特定のブラウザを使用して、InterSystems IRIS インスタンスに初めてローカル・データベースを作成する場合は、次のどちらかの操作を実行する必要があります。
 - ・ データベース・ディレクトリの名前を入力します。この場合は、その名前の確定後に、このディレクトリが **IRIS.DAT** ファイルを含んで **c:\¥InterSystems¥mgr** 内に作成されます。
 - ・ フォルダ・アイコンをクリックして既存のディレクトリを参照します。この場合は、そのディレクトリ内に **IRIS.DAT** ファイルが作成されます。

これ以降、既定では **IRIS.DAT** ファイルを格納しているディレクトリが、指定したデータベース名と同じ名前で、以前のデータベース・ディレクトリと同じ場所に作成されます。例えば、最初にデータベース **db22** を **c:\¥InterSystems¥mgr** 内のディレクトリに作成した場合、再度 [新規データベース作成] をクリックして [データベースの名前を入力してください] ボックスに **db33** と入力すると、[データベースディレクトリ] テキスト・ボックスには **c:\¥InterSystems¥mgr¥db33** が自動的に入力されます。これを **c:\¥InterSystems¥db33** に変更して **db33** を作成すると、次回はベース・ディレクトリ **c:\¥InterSystems** が入力されます。

注釈 InterSystems IRIS では、データベース・ディレクトリを構成する際、シンボリック・リンクの使用はサポートされていません。

5. [次へ] をクリックしてデータベースの構成を続けます。指定したディレクトリに、既に **IRIS.DAT** ファイルが存在する場合は、これについての警告が表示され、次のいずれかを実行できます。
 - ・ [完了] をクリックして既存のファイルを使用します。この場合、データベースのすべての特性が **IRIS.DAT** ファイルによって決定されます。データベースを他のインスタンスからコピーまたは移動する場合や、同じシステムの他のインスタンスで作成したデータベースを一時的にマウントする場合は、一般的に、この操作を行うことになります。
 - ・ [戻る] をクリックして、他のディレクトリを指定します。その後で、再度 [次へ] をクリックして、次に示す手順で新しいデータベースの特性の指定を続行します。

6. **[初期サイズ]** テキスト・ボックスに、MB 単位でデータベースのサイズを入力します (既定は 1 MB です)。

注釈 データベースのサイズが利用可能なディスク容量の合計より大きくなるようにデータベースを作成したり、編集したりすることはできません。指定したサイズがディスクの空き容量の 90% 以内になる場合は、警告が表示され、操作を確定する必要があります。

7. **[このデータベースのブロックサイズは]** ドロップダウン・リストから希望のデータ・ブロック・サイズを選択します。既定では、新しいデータベースはすべて **[ブロックサイズ]** 8 KB で作成されます。

注意 [“ラージ・ブロック・サイズに関する考慮事項”](#) のガイドラインを理解していない限り、このドロップダウン・リストから 8 KB 以外のブロック・サイズを選択しないでください。

8. マルチボリューム・データベースが必要な場合は、**[New volume threshold size]** の値を 0 より大きい数字に変更してから、TB、GB、MB の単位のラジオ・ボタンを選択します。IRIS.DAT がこのサイズに到達すると、新しいデータベース・ボリュームが作成されます。詳細は、[“マルチボリューム・データベース”](#) を参照してください。
9. グローバル・ジャーナリングがこのデータベースで必要かどうかを **[ジャーナルグローバル]** ドロップダウン・リストから選択します。[“ジャーナリングの概要”](#) を参照してください。

注釈 一時グローバルを格納するようデータベースを構成する場合、**[ジャーナルグローバル]** プロパティを **[いいえ]** に設定することは、IRISTEMP に一時グローバルを格納することとは異なります。詳細は、[“一時グローバルと IRISTEMP の使用”](#) を参照してください。

10. 暗号化が有効にされている場合、**[暗号化データベース]** で **[はい]** を選択することによってこのデータベースを暗号化できます。
11. インスタンスがミラーの一部の場合、**[ミラーリングされるデータベース]** で **[はい]** を選択することによってこのデータベースをミラーに追加できます。ミラーリングされるデータベースの作成の詳細は、[“ミラーにデータベースを追加する”](#) を参照してください。
12. これ以降のパネルでは、**[次へ]** をクリックすることでデータベースの構成を続行できます。また、これ以降の既定値をそのまま受け入れる場合は **[完了]** をクリックします。
13. このデータベースへのアクセスをコントロールするリソースを選択します。
- ・ デフォルトリソースを使用する, %DB_%DEFAULT – %DB_%DEFAULT
 - ・ 既存リソースを使用 – 既存のデータベース・リソースのリストから選択します。
 - ・ 新規リソース作成 – 新しいデータベース・リソースを作成します (新しい名前は既定で %DB_%database name になります)。

14. **[次へ]** をクリックし、データベースの属性のリストを表示します。

15. **[完了]** をクリックし、データベースを追加します。

次は、作成したデータベースの構成と管理を行います。

注釈 偶発的なデータベースの破壊を予防するために、IRIS.DAT などのオペレーティング・システム・ファイルは、それがマウントされたデータベースでなくても、オープンや上書きができませんようになっています。

5.6 ローカル・データベースのプロパティの編集

ローカル・データベースのプロパティを編集するには、以下の手順を実行します。

1. [ローカルデータベース] ページ ([システム管理]→[構成]→[システム構成]→[ローカルデータベース]) に移動します。
2. データベースの名前をクリックします。
データベースがミラーされているかどうかによって、表示される情報が異なります。以下に関するセクションを参照してください。
 - ・ [ミラーリングされていないローカル・データベース](#)
 - ・ [ミラーリングされたローカル・データベース](#)

5.6.1 ミラーリングされていないローカル・データベース・プロパティの編集

ミラーリングされていないデータベースを編集している場合、システムには以下のプロパティが表示されます (これらのフィールドの多くは、“[ローカル・データベースの作成](#)” で説明しています)。

- ・ **名前**
- ・ **ディレクトリ** (この設定は、常に IRIS.DAT データベース・ファイルの場所を反映する必要があります)
- ・ **暗号化** (変更できません)
- ・ **ミラー** — InterSystems IRIS インスタンスがプライマリ・フェイルオーバー・メンバであるミラーにデータベースを追加するには、[ミラー mirror_name に追加] をクリックします。 (このオプションは、インスタンスがミラーのプライマリである場合にのみ使用できます)。“[ミラーへ既存データベースを追加する](#)” を参照してください。
- ・ **ブロックサイズ (Bytes)** (変更できません)
- ・ **サイズ (MB)** — 次のように、3 つのサイズ設定があります。
 - － データベースの現在のサイズを変更するには [現在] を変更します。
 注釈 データベースのサイズが利用可能なディスク容量の合計より大きくなるようにデータベースを作成したり、編集したりすることはできません。指定したサイズがディスクの空き容量の 90% 以内になる場合は、警告が表示され、操作を確定する必要があります。
 - － [拡張] は、必要な場合にデータベースを拡張する量を設定します。既定 (推奨) の設定であるゼロ (0) は、現在のサイズの 12% または 10 MB のいずれか大きい方を示します。現在のサイズの 12% を使用する場合、拡張サイズは 1GB より大きくなりません。
 - － [最大] は、増加可能なデータベース最大サイズをメガバイト単位で指定します。既定の設定であるゼロ (0) は、最大がないことを示します。この設定を変更するには、新しい MB 数値を入力するか、数値の前に + または - を入力します。例えば、+10 または -20 と入力して、特定の量だけ最大サイズを増減します。データベースの最大サイズを削減する場合、警告メッセージが表示されるので、確認して操作を確定する必要があります。なお、IRISAUDIT データベースは、最大サイズを指定せずに構成する必要があります。
- ・ **リソース名** — データベースを関連付けるリソースを選択します。ドロップダウンの隣にあるリソース・アイコンをクリックすると、リソースを作成できる [リソース] ページが表示されます。
- ・ **新規グローバル** — 新しいグローバルの属性を指定します。
- ・ **グローバルジャーナル状態** — ジャーナリングを有効にする場合は選択し、無効にする場合はクリアします。“[ジャーナリングの概要](#)” を参照してください。
- ・ **削除時にグローバル属性を保持する** — グローバルのディレクトリのエントリおよび属性を、削除の際に保存するかどうかが指定します。属性には照合、ジャーナリング状態、拡張ポインタが含まれます。選択すると、グローバルの完全削除時にグローバルのディレクトリのエントリおよび属性を保存します。クリアすると、ディレクトリのエントリおよび属性を削除します。

- ・ **読み取り専用でマウント** – データベースを読み取り専用としてマウントすることを指定する場合に選択します。読み書き可能としてマウントすることを指定する場合はクリアします。
- ・ **開始時にマウントが必要** – InterSystems IRIS を起動する際にデータベースをマウントする必要がある場合は選択します。その場合、データベースがマウントできないと、InterSystems IRIS は起動しません。これにより、クラッシュ後に起動する前に、データベースでジャーナルのリカバリとトランザクションのロールバックを確実に実行することができます (“[ジャーナリング](#)” で説明しているとおり)。クリアすると、最初にデータベースをマウントせずに InterSystems IRIS を起動します。

注釈 必須の InterSystems IRIS データベース (IRISLIB、IRISAUDIT など) に対しては、この設定は既定で選択されており、変更できません。既定値はクリアされますが、USER および ENSLIB データベースと同様、作成したデータベースに対して選択することができます。データベースの状態、および明示的なデータベースのディスマウントやマウントの詳細は、“[ローカル・データベースの管理](#)”にある“ローカル・データベース情報”の表を参照してください。

- ・ **ストリームの場所** – [参照] ボタンをクリックして、データベースに関連付けられたストリームが保存されたディレクトリを選択します。既定のローカル・データベースのストリームの場所は、データベースの [ディレクトリ] 内の **stream** という名前のサブディレクトリで、これは上のフィールドのいずれかです (例 : install-dir¥mgr¥DB1¥stream)。

注釈 既定の場所を使用することをお勧めします。

- ・ **新規ボリュームのしきい値** – (オプション。マルチボリューム・データベースにのみ使用されます。) 数値を入力し、単位 (MB、GB、TB) を選択して、新規データベース・ボリュームを作成するサイズを指定します。

5.6.2 ミラーリングされているローカル・データベース・プロパティの編集

ミラーリングされているデータベースを編集している場合、システムには以下のプロパティが表示されます (これらのフィールドの多くは、“[ローカル・データベースの作成](#)”で説明しています)。

注釈 ミラーリングされるデータベースでは、ジャーナリングは必須なので、[グローバルジャーナル状態] 設定は表示されません。

- ・ **名前**
- ・ **ミラー名** – ミラー内でデータベースを識別するための名前。変更はできません。
- ・ **ディレクトリ** (この設定は、常に IRIS.DAT データベース・ファイルの場所を反映する必要があります)
- ・ **暗号化** (変更できません)
- ・ **ストリームの場所** – [参照] ボタンをクリックして、データベースに関連付けられたストリームが保存されたディレクトリを選択します。既定のローカル・データベースのストリームの場所は、データベースの [ディレクトリ] 内の **stream** という名前のサブディレクトリで、これは上のフィールドのいずれかです (例 : install-dir¥mgr¥DB1¥stream)。

注釈 データベース自体に含まれていない他のデータベース関連データと同様に (“[ミラー構成のガイドライン](#)”を参照)、ミラーリングされたデータベースのファイル・ストリームはミラーリングされません (ファイル・ストリームの詳細は、“[ストリームを使用した作業](#)”を参照してください)。

既定の場所を使用することをお勧めします。

- ・ **リソース名** – データベースに関連付けるリソースを設定します。ドロップダウンの隣にあるリソース・アイコンをクリックすると、リソースを作成できる [リソース] ページが表示されます。
- ・ **ブロックサイズ(Bytes)** (変更できません)
- ・ **照合** – グローバル属性の中で、照合属性だけが変更できます (新しいグローバルのみ)。

- ・ **削除時にグローバル属性を保持する** — グローバルのディレクトリのエントリおよび属性を、削除の際に保存するかどうかが指定します。属性には照合、ジャーナリング状態、拡張ポイントが含まれます。選択すると、グローバルの完全削除時にグローバルのディレクトリのエントリおよび属性を保存します。クリアすると、ディレクトリのエントリおよび属性を削除します。
- ・ **読み取り専用でマウント** — データベースを読み取り専用としてマウントすることを指定する場合に選択します。読み書き可能としてマウントすることを指定する場合はクリアします。
- ・ **開始時にマウントが必要** — InterSystems IRIS を起動する、またはこれがミラー・プライマリになる際にデータベースをマウントする必要がある場合は選択します。その場合、データベースがマウントできないと、InterSystems IRIS は起動しない、またはプライマリになりません。これにより、クラッシュ後に開始する前に、データベースでジャーナルのリカバリとトランザクションのロールバックを実行でき ("**ジャーナリング**" で説明しているとおり)、前のプライマリの開いているトランザクションがフェイルオーバーの一部としてロールバックされるようになります。クリアすると、最初にデータベースをマウントせずに InterSystems IRIS を起動します。**[開始時にマウントが必要]** 設定の詳細は、"**ローカル・データベースの管理**" にある "ローカル・データベース情報" の表を参照してください。
- ・ **ローカル・プロパティ** — 次のように、この領域には 3 つのサイズ設定が含まれています。
 - － データベースの現在のサイズを変更するには、**[サイズ]** を変更します。

注釈 データベースのサイズが利用可能なディスク容量の合計より大きくなるようにデータベースを作成したり、編集したりすることはできません。指定したサイズがディスクの空き容量の 90% 以内になる場合は、警告が表示され、操作を確定する必要があります。

 - － **[拡張]** は、必要な場合 (および空き領域が利用可能な場合) にデータベースを拡張する量を設定します。既定 (推奨) の設定であるゼロ (0) は、現在のサイズの 12% または 10 MB のいずれか大きい方を示します。
 - － **[最大]** は、増加可能なデータベース最大サイズをメガバイト単位で指定します。既定の設定であるゼロ (0) は、最大がないことを示します。この設定を変更するには、新しい MB 数値を入力するか、数値の前に + または - を入力します。例えば、+10 または -20 と入力して、特定の量だけ最大サイズを増減します。データベースの最大サイズを削減する場合、警告メッセージが表示されるので、確認して操作を確定する必要があります。なお、IRISAUDIT データベースは、最大サイズを指定せずに構成する必要があります。

この領域にも、**[他のシステム]** 用に **[現在]**、**[拡張]**、および **[最大]** のサイズ設定があります。他のシステムとは、現在のインスタンスがフェイルオーバー・メンバの場合は、もう一方のフェイルオーバー・メンバを指します。現在のインスタンスが非同期メンバの場合は、その非同期メンバが情報を取得できる最初のフェイルオーバー・メンバを指します。バックアップおよび非同期ミラー・メンバ上のミラーリングされたデータベースのプロパティとプライマリ上のデータベースのプロパティの同期方法に関する重要な情報については、"**ミラーリングされたデータベースの考慮事項**" を参照してください。

- ・ **新規ボリュームのしきい値** — (オプション。マルチボリューム・データベースにのみ使用されます。) 数値を入力し、単位 (MB、GB、TB) を選択して、新規データベース・ボリュームを作成するサイズを指定します。

5.7 ローカル・データベースの再配置

ローカル・データベースの IRIS.DAT ファイルを別のディレクトリに移動するには、以下を実行します。

1. 現在のデータベース・ディレクトリをメモしておきます。この情報は、**[ローカルデータベース]** ページ (**[システム管理]** → **[構成]** → **[システム構成]** → **[ローカルデータベース]**) で確認できます。
2. iris stop コマンドを使用するなどして、インスタンスのクリーン・シャットダウンを実行します。

3. IRIS.DAT ファイルと stream ディレクトリを、現在のデータベース・ディレクトリから目的の場所にコピーします。これがマルチボリューム・データベースである場合、IRIS.DAT とすべてのボリューム・ファイルをセットとして扱う必要があります。詳細は、“[ボリューム・ファイルの移動またはコピー](#)”を参照してください。

重要 データベース・ディレクトリに iris.lck ファイルがある場合、そのデータベースを移動しないでください。これは、このデータベースがまだ使用されていることを意味し、これを移動しようとすると、予期できない問題が発生する可能性があります。支援が必要な場合は[インターシステムズのサポート窓口](#)にお問い合わせください。

注釈 InterSystems IRIS では、データベース・ディレクトリを構成する際のシンボリック・リンクの使用はサポートされていません。新規データベースの場所には、完全に解決された物理パスを選択することが推奨されます。

4. テキスト・エディタで、インスタンスの iris.cpf ファイルを開きます。このファイルは通常、installation directory にあります。
5. [Databases] セクションの下でデータベースの名前を探します。古いディレクトリ・パスを新しいパスに置換し、iris.cpf を保存します。
6. 古いデータベース・ディレクトリを削除します。
7. InterSystems IRIS インスタンスを開始し、[ローカルデータベース] ページを表示して、ディレクトリが新しい場所に設定されていることを確認します。
8. データベースがミラーリングされている場合、`MIRROR を使用して、ミラーリングされているデータベースの元の場所を削除します。具体的には、オプション [ミラー管理]→[ミラー・データベースを削除] を使用します。

その後、ミラーを[有効化およびキャッチアップ](#)します。

重要 ローカル・データベース・ディレクトリを再配置したら、リモートでこのデータベースにアクセスするすべてのシステム (ECP アプリケーション・サーバなど) を更新することも必要です。“[リモート・データベースの構成](#)”の説明に従い、リモート・システムごとにリモート・データベースのディレクトリを新しい場所に更新します。

5.8 ローカル・データベースの削除

ローカル・データベースを削除するには、該当する行の [削除] リンクをクリックします。[データベース削除] ページには、削除中のデータベースの情報が表示され、以下のことができます。

- ・ 削除するデータベースにマップされているネームスペースを選択します。ネームスペースがマップされている場合は、データベースを削除できません。一覧にあるすべてのネームスペースを選択しないと、データベースを削除できません。
- 他のデータベースにマップされているネームスペースも削除できません。この場合、[ネームスペース] ページに移動するリンクが表示され、関連するネームスペースの[データベース・マッピングを変更](#)できます。別のデータベースへのすべてのマッピングを削除した後で、そのデータベースは削除する必要があるデータベースのリストから削除されます。
- ・ データベースの IRIS.DAT ファイルを削除することを選択できます (ある場合のみ)。
 - － 他のデータベースは、この IRIS.DAT ファイルを使用しません。
 - － 削除するデータベースにマップされているすべてのネームスペースをマークしました。

これらの条件に合致しない場合でもデータベースを現在の構成から削除できますが、**IRIS.DAT** ファイルは削除できません。

- ・ **[データベースを今削除する]** をクリックして、情報を確認した後にデータベースを削除することを確認します。

IRIS.DAT ファイルを削除できない場合や削除しないことを選択した場合でも、データベースは InterSystems IRIS パラメータ・ファイルの [データベース・セクション](#) から、そして管理ポータルに表示されるローカル・データベースのリストから削除されます。

5.9 プログラムによるデータベースの構成

InterSystems IRIS には、データベース構成の自動化に役立つ、プログラムによるツールも含まれています。

Config.Databases クラスを使用して、データベースを作成および構成できます。詳細は、“[インターシステムズ・クラス・リファレンス](#)” を参照してください。

`^DATABASE` コマンド行ユーティリティを使用してデータベースを構成することもできます。

データベースを構成するもう 1 つの方法として、構成マージと組み合わせて [CreateDatabase](#)、[ModifyDatabase](#)、または [DeleteDatabase](#) の操作を含めることもできます。構成マージにより、InterSystems IRIS インスタンスに適用する設定や操作を指定する宣言型マージ・ファイルを適用することで、このインスタンスをカスタマイズできます。この機能の詳細は、“[構成マージを使用した InterSystems IRIS の自動構成](#)” を参照してください。

5.10 関連項目

- ・ [ネームスペースとデータベース](#)
- ・ [リモート・データベースの構成](#)
- ・ [^DATABASE](#)
- ・ [構成マージを使用した InterSystems IRIS の自動構成](#)
- ・ [cvendian を使用したビッグ・エンディアン・システムとリトル・エンディアン・システム間の変換](#)
- ・ [ローカル・データベースの管理](#)

6

リモート・データベースの構成

ここでは、リモート・データベースを追加および削除する方法を説明します。リモート・データベースは、別のサーバ・システム上に物理的に配置されているデータベースであり、ローカル・サーバ・システムに物理的に配置されているローカル・データベースとは異なります。

6.1 リモート・データベースの追加

データベースのホストがサーバ上で分散キャッシュ・データ・サーバとして構成されている場合は、ローカル・サーバ上にリモート・データベースを定義できます。データ・サーバを追加する手順は、“[アプリケーション・サーバの構成](#)”を参照してください。

リモート・データベースを追加するには、以下の手順に従います。

1. [リモートデータベース] ページ ([システム管理]→[構成]→[システム構成]→[リモートデータベース]) に移動し、[リモートデータベースを作成] をクリックしてウィザードを起動します。
2. [リモートサーバ] ドロップダウンからデータベースをホストするデータ・サーバを選択します。
3. [リモートディレクトリ] ラジオ・ボタンから、リモート・データベース・ディレクトリを指定する方法を選択します。
 - ・ [リストからデータベースを選択] では、リモート・サーバ上のデータベース・ディレクトリのドロップダウン・リストから選択できます。現在リモート・データ・サーバにアクセスできない場合、ドロップダウン・リストは空となります。
 - ・ [データベース仕様を入力する] では、データベース・ディレクトリを直接入力できますが、ポータルはその入力を検証しません。
4. データベース名 (ローカル・サーバ上の名前。リモート・データ・サーバ上の名前と同一にする必要はありません) を入力します。“[データベース名の規則](#)”を参照してください。これで、リモート・データベースを定義できました。
5. オプションで、データベースに関連付けられたストリームが保存されるディレクトリを選択できます。既定の場合、リモート・データベースのストリームの場所は、InterSystems IRIS Temp ディレクトリ (install-dir¥mgr¥Temp) です。

注釈 既定の場所を使用することをお勧めします。

6. リモート・データベースを構成するには、[保存] をクリックします。

リモート・データベースの [編集] リンクをクリックして、リモート・データベース・フィールドをいつでも変更できます。

6.2 リモート・データベースの削除

リモート・データベースを削除するには、該当する行の **削除** をクリックします。[データベース削除] ページには、削除中のデータベースの情報が表示され、以下のことができます。

- ・ 削除するデータベースにマップされているネームスペースを選択します。ネームスペースがマップされている場合は、データベースを削除できません。一覧にあるすべてのネームスペースを選択しないと、データベースを削除できません。

他のデータベースにマップされているネームスペースも削除できません。この場合、[ネームスペース] ページに移動するリンクが表示され、関連するネームスペースの [データベース・マッピングを変更](#) できます。別のデータベースへのすべてのマッピングを削除した後で、そのデータベースは削除する必要があるデータベースのリストから削除されます。

- ・ **データベースを今削除する** をクリックして、情報を確認した後にデータベースを削除することを確認します。

このアクションは、ローカル・インスタンスのリモート・データベース構成からデータベースを削除するだけです。ホストの実際のデータベースとローカル構成には影響がありません。

6.3 関連項目

- ・ [アプリケーション・サーバの構成](#)
- ・ [ローカル・データベースの構成](#)

リソースの使用による資源の保護

- ・ 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。

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。

7.1 リソースのタイプ

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 - 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 – InterSystems のさまざまな接続テクノロジーを使用して 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が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・リソースとして

は %Service_Callerin, %Service_Consumer, %Service_Children, %Service_Parentin, %Service_EITn, %Service_OPIt, %Service_OneC, %Service_OneP, %Service_In, %Service_Out, %Service_S, %Service_Fut, %Service_Pria,
および %Service_WebGateway があります。

- アプリケーション・リソース—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。

7.2 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

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。

7.2.1 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_OAuth2_Client
- %Admin_OAuth2_Registration
- %Admin_OAuth2_Server

- ・ [%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。

7.2.1.1 %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、および **%SYSTEM.java.SQL** クラスと **%SYSTEM.python.SQL** クラスでのメソッドの使用も対象となります。

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。

7.2.1.2 %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)。

7.2.1.3 %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。

7.2.1.4 %Admin_OAuth2_Client

このリソースは、InterSystems IRIS を OAuth2 クライアントとして使用する場合の構成設定を制御します。これにより、管理ポータル の [システム] → [セキュリティ管理] → [OAuth 2.0 クライアント] ページにアクセスできます。このページから、クライアントとして使用する InterSystems IRIS に OAuth2 承認サーバを設定できます。承認サーバを設定した後は、クライアント構成を作成できます。詳細は、“[OAuth 2.0 クライアントとしての InterSystems IRIS Web アプリケーションの使用法](#)” を参照してください。

7.2.1.5 %Admin_OAuth2_Registration

このリソースは、InterSystems IRIS を OAuth2 承認サーバとして使用する場合のクライアント構成の作成、読み取り、更新、および削除を制御します。管理ポータル のクライアント構成のページには、以下の URL でインスタンスの `<baseUrl>` を使用してアクセスできます。

```
https://<baseUrl>/csp/sys/sec/%25CSP.UI.Portal.OAuth2.Server.ClientList.zen
```

これにより、管理ポータル の [システム] → [セキュリティ管理] → [OAuth 2.0 の管理 – (セキュリティ設定)] ページにもアクセスできます。このページから、特定のユーザのトークンを取り消すことができます。

7.2.1.6 %Admin_OAuth2_Server

このリソースは、InterSystems IRIS を OAuth2 承認サーバとして使用する場合の構成設定を制御します。これにより、管理ポータル の [システム] → [セキュリティ管理] → [OAuth 2.0 認可サーバ構成 – (セキュリティ設定)] ページにアクセスできます。詳細は、“[OAuth 2.0 承認サーバとしての InterSystems IRIS の使用法](#)” を参照してください。

7.2.1.7 %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。

7.2.1.8 %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。

7.2.1.9 %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。

7.2.1.10 %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。

7.2.1.11 %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。

7.2.2 %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 of 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マ的にアクセスする 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

7.2.3 %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方法](#)”を参照してください。

7.2.4 %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。

7.2.5 %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。

7.2.6 %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。

7.2.7 %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ム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。

7.2.8 %Service_Native リソース

`%Service_Native` リソースは非推奨です。Python、Java、.NET、Node.js のどれを通じてユーザが Native API 呼び出しを発行できるかの制御については、4 つの `%Native_*` リソースを参照してください。4 つの Native API リソースは、“[%Service_Bindings](#)”で確認できます。

7.3 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。

7.3.1 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 7-1: 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。

7.3.2 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。

7.3.3 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 7-2: **%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。

7.3.4 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するとエラーで失敗し、明示的にマウントしようすると、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）。

7.3.5 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）。

7.3.6 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割を持っており、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、EXECUTE 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。

7.3.6.1 特別な機能

IRISSYS データベースにあるコードで利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。

7.4 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。

7.5 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。

7.6 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 – 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 – 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 –
 - Read – 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 - 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 - 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。

7.6.1 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。

7.7 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、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。

7.7.1 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。

7.7.2 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。

8

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の区切った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ーである必要があります。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。

8.1 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` – 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` – 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` – 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。

8.2 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 8-1: 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

8.3 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。以下に例を示します。

```
$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。以下に例を示します。

```
$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。

8.4 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。

8.5 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。

ル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の認証時に認識されます。

9

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。

9.1 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 9-1: 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。

9.1.1 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 9-2: 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 <ul style="list-style-type: none"> 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 <ul style="list-style-type: none"> 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。

9.1.2 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。

9.2 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 に割り当てるとは、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、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。

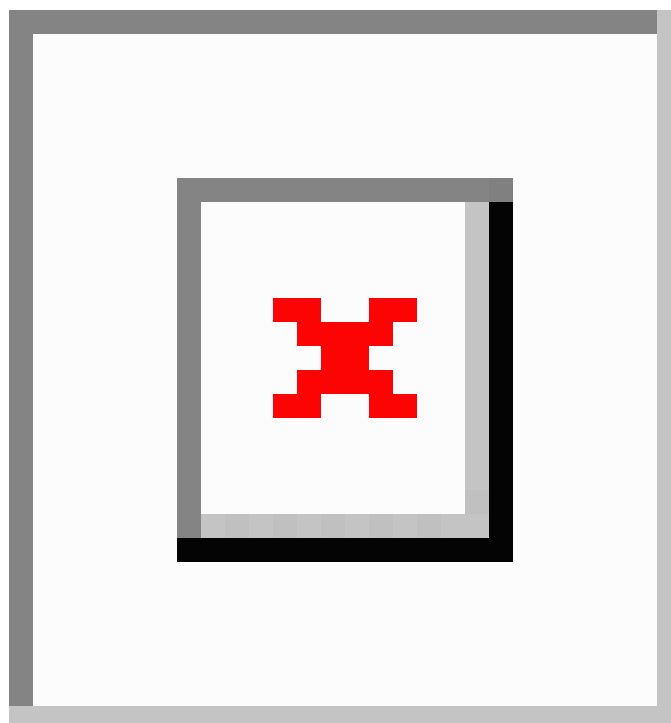
ロールのメンバは、[ロール編集] ページの [メンバ] タブに表示されます。このタブでは、ロールに新しいメンバを割り当てることもできます。ロールを他のロールに割り当てると、その割り当て先ロールは [ロール編集] ページの [割り当て先] タブに表示されます。このタブでは、ロールの割り当て先ロールを追加することもできます。

9.2.1 複数ロールの割り当て例

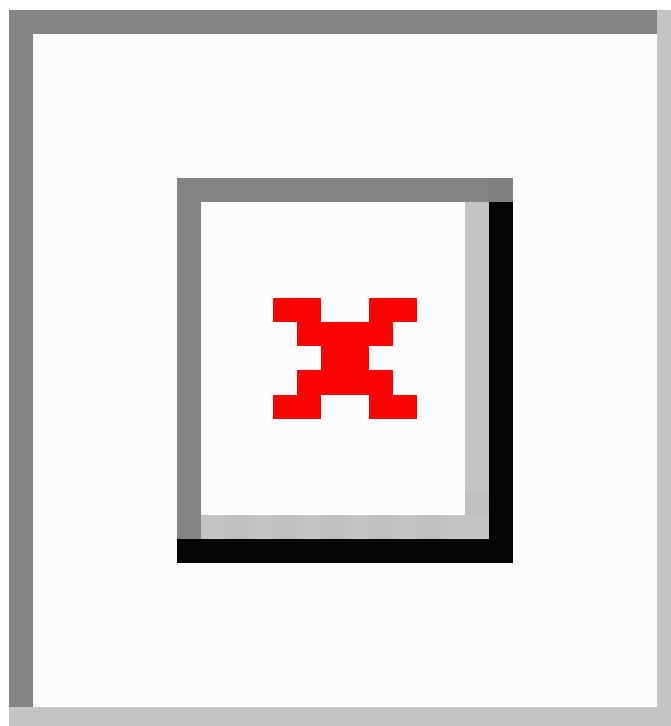
このセクションでは、InterSystems IRIS におけるユーザとロールとの関係の例について説明します。

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。

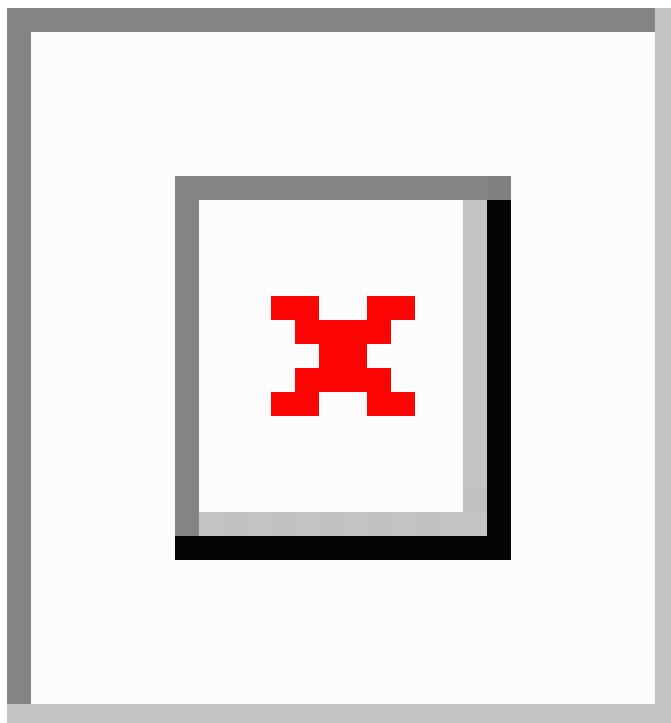
最初の作成時、Lee はどのロールのメンバでもありません。これは以下のように Lee のプロフィールに反映されます。



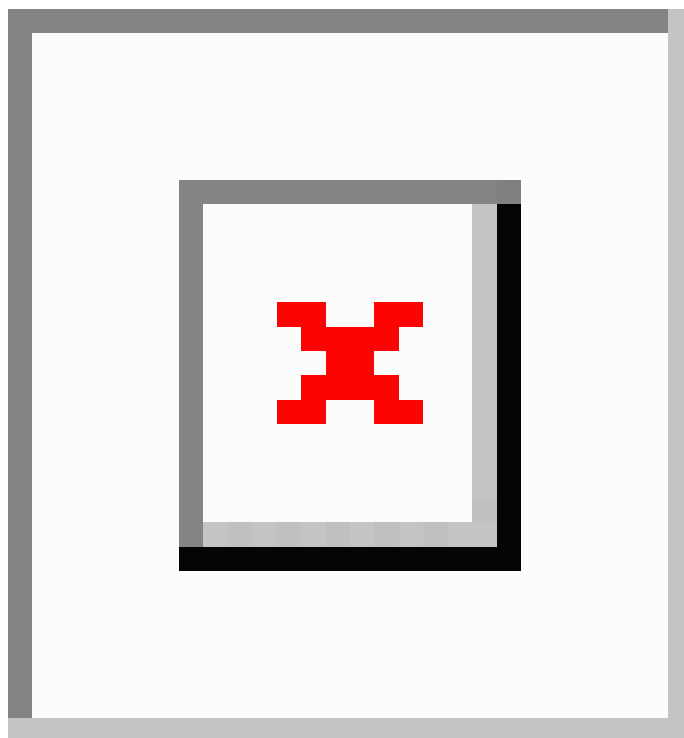
Lee がロール **FirstRole** に割り当てられると、Lee のプロフィールが変更されます。



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 のプロフィールが変更されます。



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。



9.3 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。

9.3.1 命名規則

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。

9.4 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。

9.4.1 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](#)

9.4.2 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。

9.4.3 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す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す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)。

9.4.4 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。

9.4.5 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。

9.4.6 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 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。

9.4.7 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 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。

9.4.8 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 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。

9.4.9 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。

9.4.10 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

9.4.10.1 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。

9.4.10.2 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。

9.4.10.3 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。

9.4.10.4 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。

9.5 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[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[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[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 9-3: 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		

リソース	%Developer	%Manager	%Operator	%SQL	%SecureBreak
%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			
%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。

9.5.1 %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。

9.5.2 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。

9.6 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。

9.6.1 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。

9.7 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。

10

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。

10.1 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 10-1: 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。 <code>%Admin_Secure:Use</code> 特権を持つユーザ、または <code>%All</code> 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、 タイプ 、 タイプ 、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。

10.1.1 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 — 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 — [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 – 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 – [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 – 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。

10.2 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] – このユーザ・アカウントの一意の識別子。
- ・ [フルネーム] – このアカウントの表示可能な名前。
- ・ [有効] – 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) – 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) – 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] – [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](#)

10.2.1 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。

10.2.2 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。

10.2.2.1 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)。

10.2.2.2 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** – 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** – 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。

10.2.3 ユーザ・プロフィールの表示

ユーザ・プロフィールには、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・プロフィールを表示する手順は以下のとおりです。

1. 管理ポータル 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 [プロフィール] をクリックします。ユーザ・プロフィールが表示されます。

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 [プロフィール] をクリックします。

ユーザ・プロフィールの一部として、以下のプロパティが表示されます。

テーブル 10-2: ユーザ・プロフィールのプロパティ

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。

10.2.4 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 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。

10.2.5 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 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。

10.3 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 10-3: 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してこのアカウントを無効に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。このユーザ・アカウントは HealthShare で内部的に使用されるため、無効にしないでください。 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してこのアカウントを無効に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。このアカウントは HealthShare で内部的に使用されるため、無効にしないでください。
HS_Services	%HS_ServiceRole	このアカウントは医療製品で内部用に使用されます。既定でこのアカウントは、InterSystems IRIS for Health™ および HealthShare® Health Connect に対して無効になっているため、これらの製品で FHIR® ベースの IHE プロファイルを使用する前に有効にする必要があります。その他の HealthShare ソリューションに対しては、このアカウントは既定で有効になっており、無効にしないでください。

“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 の新規インストール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。InterSystems IRIS には、**%All** ロールを持つ有効なアカウントが 1 つ以上必要なため、**%All** ロールを持つ一意のアカウントを作成したら、これらの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。

10.3.1 さまざまなアカウントに関するメモ

10.3.1.1 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、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。

10.3.1.2 _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。

10.4 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戻り値が決まります。
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>
```


11

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](#)

11.1 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](#)

11.1.1 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は制御され、[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で、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。

11.1.2 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となりますが、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となります。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。

11.1.3 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・ロールに関連付けます。
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かかわらず依存します。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かかわらず依存します。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。

11.1.3.1 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ことも制御できます。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 11-1: 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を確認され、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、ユーザが学籍系の業務ロールのメンバであるかどうか確認されます。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。

11.1.4 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。

11.2 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](#)

11.2.1 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。

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を 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](#)

“Web アプリケーションへのカスタム・ログインの保護” も参照してください。

11.2.1.1 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 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."
}
}
```

11.2.2 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：\[マッピングロール\] タブ](#)

- ・ [アプリケーションの編集：\[ルーチン/クラス\] タブ](#)

11.2.2.1 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ピングロール（“[アプリケーションの編集：\[マッピングロール\] 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 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。

11.2.2.2 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_DBL

Calling the AddRoles method...

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

USER>
```

11.2.3 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

11.2.4 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

11.3 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](#)

11.3.1 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](#)

11.3.2 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。

11.3.2.1 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、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]、[WSGI]、または [CSP/ZEN] の有効化

Web アプリケーションのタイプ：

- ・ [REST] – 選択したサードパーティ・フロントエンド・テクノロジーを使用する [REST アプリケーション](#)。
- ・ [WSGI] – [PEP-3333](#) 標準に準拠し、Python フロント・エンドを使用する [WSGI アプリケーション](#)。
- ・ [CSP/ZEN] – [InterSystems CSP フロント・エンド](#)または従来の InterSystems Zen フロント・エンドを使用するアプリケーション。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、そのテクノロジー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 アプリケーションの場合) 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。

空のパスをリダイレクト

(REST アプリケーションの場合) アプリケーションが空のパスを / に送るかどうか。例えば、アプリケーション /csp/appname の場合、/csp/appname に対する要求は、/csp/appname/ にリダイレクトされます。

JWT 認証を使用

(REST アプリケーションの場合) アプリケーションが JSON Web トークン (JWT) 認証をサポートするかどうか。

JWTアクセストークンタイムアウト

(REST アプリケーションの場合) JWT が失効するまでの秒数。

JWTリフレッシュトークンタイムアウト

(REST アプリケーションの場合) JWT のリフレッシュ・トークンが失効するまでの秒数。

アプリケーション名

(WSGI アプリケーションの場合) 呼び出し可能な Python アプリケーションを含むファイルの名前。

呼び出し可能名

(WSGI アプリケーションの場合) 既定値 (application) でない場合、呼び出し可能なアプリケーションの名前。

WSGI アプリケーション・ディレクトリ

(WSGI アプリケーションの場合) 呼び出し可能な Python アプリケーションを含むディレクトリ。

11.3.2.2 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。

11.3.2.3 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 を使用するか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 アプリケーションに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 のいくつかのドキュメントで扱われています。

11.3.3 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 (ル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。

11.3.4 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。

11.3.5 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。

11.3.6 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。

11.3.7 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。

11.4 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 11-2: 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	InterSystems VS Code 拡張機能によって使用される REST API (%Api.Atelier ディスパッチ・クラス)。	%Development	いいえ
/api/deepsee	InterSystems IRIS Code Business Intelligence によって使用される 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
/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/interop-editors	ルール・エディタ REST API (%Api.InteropEditors ディスパッチ・クラス)。		いいえ
/api/mgmt	API 管理 REST API (%Api.Mgmt 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

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
/csp/sys/sec	ポータル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
/oauth2	InterSystems IRIS が OAuth 2.0 承認サーバとして構成されている場合、そのサーバによって使用。	%Admin_Secure	いいえ
/ui/interop/rule-editor	ルール・エディタのユーザ・インタフェース。		いいえ

12

TLS の構成

InterSystems IRIS® データ・プラットフォームでは複数の構成をサポートできます。それぞれの構成が、TLS 関連値の名前付きセットを指定します。既存の構成はすべて起動時に有効になります。管理ポータルで新しい構成を作成すると、その構成は保存時に有効になります。TLS 構成を管理するページは、[SSL/TLS 構成] ページ ([システム管理]→[セキュリティ]→[SSL/TLS 構成]) です。

12.1 TLS 構成の作成または編集

TLS 構成を作成または編集するページは、[SSL/TLS 構成] ページ ([システム管理]→[セキュリティ]→[SSL/TLS 構成]) です。新しい構成を作成するには、[新規構成の作成] をクリックして [新規 SSL/TLS 構成] ページを表示します。既存の構成を編集するには、その構成の名前の右側にある [編集] をクリックします([ミラーのための構成を作成] をクリックすることで、ミラー・メンバの新しい構成セットも作成できます。ミラーリングおよび TLS 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)。

TLS 構成を作成または編集する場合、以下のフィールドを使用できます。

- ・ **[構成名]** – 構成を識別するための文字列。構成名には、すべての英数字、および “|” 文字以外の句読点を使用できます。
- ・ **[説明]** – 任意のテキスト。
- ・ **[有効]** – キーの有効化の際にこの構成を利用可能とするかどうかの指定。
- ・ **[タイプ]** – この構成の使用目的。[クライアント] または [サーバ] を選択します。既定値は [クライアント] です。クライアントはプロトコルの使用を開始し、サーバは最初の要求に応答します(InterSystems IRIS スーパーサーバではサーバ構成が使用されます。TLS クライアントではクライアント構成が使用されます)。このフィールドに選択する値は、以下によって決まります。
 - 次のフィールドが、[サーバ証明書認証] フィールドと [クライアント証明書認証] フィールドのどちらであるか。クライアント用の構成の場合、次のフィールドは [サーバ証明書認証] になります。このフィールドでは、クライアントの接続先サーバの証明書に求められる可能性のある認証を指定します。サーバ用の構成の場合、次のフィールドは [クライアント証明書認証] になります。このフィールドでは、サーバへの接続を試行するクライアントの証明書に求められる可能性のある認証を指定します。
 - [信頼された証明書機関の証明書を含むファイル] フィールドの動作。
- ・ **[サーバ証明書の検証] または [クライアント証明書の検証]** – 構成で接続相手の証明書の検証が必要かどうかを指定します。

クライアント用の構成では、[サーバ証明書認証] を指定する必要があるため、以下の使用可能な値がサポートされています。

- [なし] - どのような状況でも続行します。
- [必要] - 証明書の認証が成功する場合にのみ続行します。

サーバ用の構成では、**[クライアント証明書認証]** を指定する必要がある、以下の使用可能な値がサポートされています。

- [なし] - サーバ側でクライアント証明書を要求せず、また必要としないことを示します。
- [要求] - 証明書を提供する（または提供しない）クライアントを許可します。クライアントが証明書を提供しない場合、認証は続行します。クライアントが証明書を提供して認証に失敗すると、認証が失敗します。
- [必要] - クライアントが証明書を提供する必要があることを示します。認証は証明書の認証によって決まります。

・ **[信頼された認証機関の証明書を含むファイル]** - この構成が信頼する 1 つまたは複数の認証機関 (CA) の X.509 証明書 (PEM 形式) が含まれているファイルのパスと名前。構成では、信頼された CA の証明書を使用して、接続相手の証明書を検証します。一般に、プロダクション・システムは、公的に利用可能な証明書を持つ商用認証機関からの証明書を使用します。

このフィールドについては、以下の点に注意してください。

- ファイルのパスは、絶対パスとして指定することも、`<install-dir>/mgr/` ディレクトリが基準になる相対パスとして指定することもできます。
- Windows と macOS では、ローカル・オペレーティング・システムが提供する、信頼された CA 証明書を構成で使用するよう指定できます。そのためには、文字列 `%OSCertificateStore` をこのフィールドの値として指定します。

Windows では、InterSystems IRIS は Microsoft ルート証明書プログラムと互換性があり、このプログラムが Windows Update を使用して追加の証明書をオンデマンドでフェッチします。証明書の更新を構成する方法の詳細は、Microsoft の Web サイトで ["Configure Trusted Roots and Disallowed Certificates"](#) を参照してください。

- **[クライアント証明書認証]** の値が [なし] のサーバ構成の場合、このフィールドは使用できません (相手認証が存在しないため)。
- Windows の証明書のエクスポート・ウィザードからエクスポートした証明書は、既定の DER でエンコードしたバイナリ X.509 ではなく、PEM でエンコードした X.509 形式とする必要があります。
- **ミラーリング**では、独自の証明書を検証するために十分な情報も構成に含まれている必要があります。

これらの証明書を使用する方法は、["必須証明書チェーンの確立"](#) を参照してください。このような証明書のファイル名と、証明書チェーンの確認方法については、OpenSSL のドキュメントで [verify](#) コマンドを参照してください。

- ・ **[このクライアントの認証情報]** または **[このサーバの認証情報]** - ローカル構成の X.509 証明書および秘密鍵がファイルとして必要な場合に、これらを格納したファイル名。
 - **[このクライアントの証明書を含むファイル]** または **[このサーバの証明書を含むファイル]** - この構成独自の X.509 証明書の場所。この値は、絶対パスと相対パスのいずれかとして、PEM エンコードする必要があります。証明書チェーンも含めることができます。これを認証に使用する方法については、["必須証明書チェーンの確立"](#) を参照してください (Windows の証明書のエクスポート ウィザードからエクスポートした証明書は、既定の DER でエンコードしたバイナリ X.509 ではなく、PEM でエンコードした X.509 形式とする必要があることに注意してください)。
 - **[関連づけられた秘密鍵を含むファイル]** - 構成の秘密鍵ファイルを格納する場所。絶対パスまたは相対パスで指定します。
 - **[秘密鍵タイプ]** - 秘密鍵の生成に使用するアルゴリズム。有効なオプションは、**[DSA]** (Digital Signature Algorithm)、およびアルゴリズム開発者の名前から名付けられた **[RSA]** (Rivest, Shamir, Adleman) です。
 - **[秘密鍵パスワード]** - 構成の秘密鍵を暗号化および解読するための任意のパスワード。

注釈 秘密鍵がパスワードで保護されていて、ここに値を入力しない場合、InterSystems IRIS は、秘密鍵および証明書の公開鍵が相互に対応することを確認できません。この結果、対応しない鍵が鍵のペアとして保存される可能性が生じます。

- **[秘密鍵パスワード(確認)]**— 入力したパスワードが目的の文字列であることを確認するために、パスワードを再度入力します。

・ **[暗号方式設定] :**

- **[最小プロトコル・バージョン]**— この構成でサポートされる TLS バージョンのうち、最も古いバージョン。これは、インスタンスでサポートできるすべてのバージョンが表示されるドロップダウン・メニューで、既定値は [TLS v1.2] です。以下の注を参照してください。
- **[最大プロトコルバージョン]**— この構成でサポートされる TLS バージョンのうち、最新のバージョン。これは、インスタンスでサポートできるすべてのバージョンが表示されるドロップダウン・メニューで、既定値は [TLS v1.3] です。以下の注を参照してください。
- **[有効な暗号リスト (TLSv1.2以下)]**— クライアントとサーバ間の通信の保護に使用する一連の暗号 (TLS v1.2 以前のバージョンを使用している場合)。このトピックの詳細は、[“サポートされる暗号構文”](#) を参照してください。
- **[有効な暗号スイート (TLSv1.3)]**— クライアントとサーバ間の通信の保護に使用する一連の暗号 (TLS v1.3 を使用している場合)。このトピックの詳細は、[“サポートされる暗号構文”](#) を参照してください。
- **[DiffieHellmanビット数]**— (サーバ専用) Diffie Hellman 暗号で使用する鍵のサイズ (ビット数)。鍵の最小サイズはオペレーティング・システムによって異なります。**[自動]** オプションでは、ローカルのオペレーティング・システムで必要最小限のサイズ以上の鍵サイズが指定されます。Open Web Application Security Project (OWASP) では、[最小鍵サイズとして 2048 ビット](#) を推奨しています。

注釈 “[自身の InterSystems IRIS のインスタンスでサポートされている TLS のバージョン](#)” で説明されているように、利用可能な TLS のバージョンは、使用している基礎の OpenSSL ライブラリのバージョンによって異なります。InterSystems IRIS は、使用されている OpenSSL ライブラリを確認し、関連する利用可能な TLS のバージョンのみを提示しようと試みます。ただし、InterSystems IRIS が特定できない方法でシステムが構成されていることがあります。例えば、Ubuntu 20.04 に付属する OpenSSL の構成は既定値から変更されていて、TLS 1.0 と TLS 1.1 が禁止されています。このような場合に備えて、InterSystems IRIS のエラー・メッセージとログは、OpenSSL の構成と InterSystems IRIS 構成との間に競合があるかどうかを判断するのに役立つように設計されています。予期しない動作が発生した場合は、[インターシステムズのサポート窓口 \(WRC\)](#) またはオペレーティング・システムのベンダまでお問い合わせください。

・ **[OCSP settings]:**

- **[OCSPステープリング]**— 構成で OCSP ステープリングがサポートされているかどうかを示します。

クライアントで OCSP ステープリングを有効にすると、そのクライアントは OCSP ステープリングを要求します。サーバがステープリングされた OCSP 応答を提供しないか、応答が検証に失敗した場合、ハンドシェイクは失敗します。サーバで OCSP ステープリングが有効になっていると、そのサーバはクライアントからの要求を受信したときに、ステープリングされた OCSP 応答を提供します。

InterSystems IRIS では、TLS 構成を保存する際に OCSP 応答ファイルが存在しない場合はそれが作成されます。失効した OCSP 応答は、TLS 構成を保存する際またはサーバが要求を受信した際に、自動的に更新されます。また、CertCheck システム・センサ (%SYS.Monitor.SystemSensors::CertCheck()) によって、応答は定期的に更新されます。IRIS バックグラウンド・ジョブおよび TLS サーバ接続を開始するすべてのジョブに、OCSP 応答ファイルに対する読み取り/書き込み権限が必要です。InterSystems IRIS の実効グループに読み取り/書き込みアクセス権を付与することで、これらの要件を満たすことができます。

OCSP ステープリングを有効にするかどうかにかかわらず、サーバ側 TLS ソケットを作成できるユーザはすべて、有効なあらゆる TLS 構成でこれを行うことができます。

注釈 構成がクライアントであるのかサーバであるのか、および目的の機能により、必須フィールドは異なります。TLS 構成によっては必要ではないフィールドもあります。

構成の作成プロセスまたは編集プロセスを完了するには、このページの上部に表示される以下のボタンを使用します。

- ・ **[保存]** – 構成を保存および有効にして、ダイアログを閉じます。既存の構成の変更や作成する構成を保存します。
- ・ **[キャンセル]** – 既存の構成の変更や作成する構成を保存せずにダイアログを閉じます。
- ・ **[テスト]** – 有効な構成情報であるかどうかを確認します。構成のロールがクライアントの場合、このボタンを選択するとサーバ (URL でなく、ホスト名) やポート番号のプロンプトも表示されます。InterSystems IRIS ではそのサーバとのテスト接続を確立しようとします(サーバ構成の作成時、このボタンは使用できません)。

注釈 構成にエラーがなくても、一部の TLS サーバには **[テスト]** ボタンで正常に接続できないことがあります。これは、接続テストでは TLS ハンドシェイクの後に HTTP 要求が実行されるためです。サーバがハンドシェイクの前に StartTLS メッセージを予期している場合 (LDAP、SMTP、FTPS、または別のプロトコルでの使用時など)、実際にはサーバとの TLS 接続に成功していても、テストは失敗します。

12.1.1 証明書に必要な情報

クライアントがサーバを認証する場合、クライアントには、サーバ独自の証明書から、サーバの信頼された CA 証明書まで、これらの間にあるものすべてを含む完全な証明書チェーンが必要です。

サーバ TLS 構成を設定するときに、サーバの信頼された CA 証明書がルート証明書ではない場合、問題があります。認証を適切に機能させるために、クライアントでは、サーバの個人証明書から信頼された自己署名 CA 証明書への証明書チェーンを構成するすべての証明書へのアクセスを必要とします。このチェーンはサーバの証明書ファイル (ハンドシェイク時に送信されたもの) とクライアントの信頼された CA 証明書ファイルの組み合わせから得られます。信頼された自己署名ルート CA 証明書はクライアントの CA 証明書ファイルに入っている必要があります。また、サーバの個人証明書はサーバの証明書ファイルの先頭エントリでなければなりません。その他の証明書は、これらの 2 つの場所に分けることができます。クライアントがサーバに対して認証を実行するときには、同じ制約が逆に適用されます。

証明書の形式に関しては、Windows の証明書のエクスポート・ウィザードからエクスポートした証明書は、既定の DER でエンコードしたバイナリ X.509 ではなく、PEM でエンコードした X.509 形式とする必要があります。ファイルの拡張子に関係なく、すべての証明書は PEM でエンコードする必要があります。

12.1.2 有効化された暗号スイートの構文

構成で許可されるのは、有効化された暗号スイートを使用する接続のみです。有効化された暗号スイートを指定するには、以下のいずれかを実行できます。

- ・ 個々の暗号スイートのリストを提供する
- ・ OpenSSL の構文を使用して、有効化/無効化する暗号スイートを指定する

暗号スイート名のリスト、および有効な暗号スイートを指定するための構文は、どちらも [openssl.org の ciphers\(1\) のマニュアル・ページ](#) で説明されています。この構文を使用すれば、構成に対してさまざまな機能やアルゴリズムを使用する場合の必要事項や禁止事項のガイドラインを指定できます。

InterSystems IRIS 構成で暗号スイートの既定値は、`ALL:!aNULL:!eNULL:!EXP:!SSLv2` です。これは、コロンで区切られた文で以下のグループに分けられます。

- ・ `ALL` – `eNULL` 暗号以外のすべての暗号スイートを含めます。
- ・ `!aNULL` – 認証を提供しない暗号を除外します。
- ・ `!eNULL` – 暗号化を提供しない暗号を除外します。

- ・ **!EXP** – 輸出承認済みアルゴリズム (40 ビットおよび 56 ビット) を除外します。
- ・ **!SSLv2** – SSL v2.0 暗号スイートを除外します。

詳細は、OpenSSL のドキュメントで [ciphers](#) コマンドを参照してください。

12.1.3 TLS を使用する InterSystems IRIS クライアント・アプリケーションに関するメモ

一部の動作については、InterSystems IRIS スーパーサーバと対話するクライアント・アプリケーションのサポートに InterSystems IRIS インスタンスを使用できます。

TLS を使用して InterSystems IRIS スーパーサーバと対話するクライアント・アプリケーションを使用する場合は、構成について、次の点に特に注意してください。

- ・ **[構成名]** – クライアントの名前には制限はありませんが、接続を構成するためにはこの情報は必須です。
- ・ **[タイプ]** – インスタンスは TLS クライアントと共にサービスを提供するので、[タイプ] には **[クライアント]** を指定する必要があります。
- ・ **[暗号スイート]** – 指定された暗号スイートは、サーバにより要求または指定されたものと一致する必要があります。

また、“[必須証明書チェーンの確立](#)” で説明しているように、クライアントとサーバは互いの証明書チェーンを検証できるように構成することも必要です。

12.2 構成の削除

TLS 構成を削除するページは、**[SSL/TLS 構成]** ページ (**[システム管理]**→**[セキュリティ]**→**[SSL/TLS 構成]**) です。構成を削除するには、構成名の右側にある **[削除]** をクリックします。ポータルからアクションの確認が求められます。

12.3 既存の構成の証明書の更新

InterSystems IRIS では、TLS 構成に関連付けられている証明書を更新する手段を提供しています。新しい証明書を取得した後に、TLS 構成でこれを更新できます。新しいネットワーク接続の確立時に、その変更が適用されます。多くの接続が切断されてはすぐに再確立されるため、変更はすぐに適用できます。ただし、一部の接続は長時間にわたって開かれたままです。例えば、ミラーリングや一部の相互運用インタフェース接続がこれに当たります。更新した証明書が確実に適用されるように、こういった接続を切断して再確立することをお勧めします。

ミラーリング環境には、接続の再確立後に追加の手順が必要です。詳細は、“[X.509 DN 更新の承認](#)” を参照してください。

12.4 予約済みの構成名と必須の構成名

InterSystems IRIS では、特定の機能で使用するために、いくつかの TLS 構成名が予約されています。このような機能を使用するときには、この予約構成名を使用する必要があります。予約構成名は以下のとおりです。

- ・ **%MirrorClient** – TLS クライアントとして動作する場合のミラー・メンバ用。ミラーリングおよび TLS 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。

- ・ %MirrorServer – TLS サーバとして動作する場合のミラー・メンバ用。ミラーリングおよび TLS 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。
- ・ %TELNET/SSL – TLS で保護された接続を受け入れる場合の Windows Telnet サーバ用。ミラーリングおよび Telnet の詳細は、“[TLS に対する InterSystems IRIS Telnet サーバの構成](#)”を参照してください。

重要 TLS が正しく機能するようにするには、ここに表示されているとおりに、大文字と小文字を正確に区別して各構成名を使用する必要があります。

12.5 プログラムによる TLS 構成の作成、編集、削除

TLS 構成をプログラムで管理するには以下を使用します。

- ・ **Security.SSLConfigs** クラス
- ・ 適用可能な[構成マージ・アクション](#)
 - CreateSSLConfigs
 - ModifySSLConfigs
 - DeleteSSLConfigs

13

スーパーサーバの管理

13.1 スーパーサーバの概要

InterSystems IRIS® スーパーサーバは、%SYS.SERVER ルーチンを実行するプロセスです。これは、以下を行います。

1. スーパーサーバ・ポートでリッスンする
2. 特定のタイプの受信 TCP 要求を認識する
3. それらの要求を処理して、元の送信者に応答する

InterSystems IRIS は、インスタンスごとに複数のスーパーサーバをサポートします。スーパーサーバごとに異なる組み合わせの接続要求を受け入れ、互いに独立して TLS を使用することができます。

13.2 サポートされている要求

スーパーサーバでは、すぐに使用できるさまざまな要求をサポートしています。これには、以下のものがあります。

- ・ JDBC クライアントと ODBC クライアント
- ・ CSP/REST
- ・ DataCheck
- ・ ECP (システムの既定のスーパーサーバのみ)
- ・ ミラーリング (システムの既定のスーパーサーバのみ)
- ・ シャーディング (システムの既定のスーパーサーバのみ)
- ・ SNMP (Windows のみ)

また、スーパーサーバでは、従来の接続も一部サポートしています。これらの接続は、既定で無効になっています。これには、以下のものがあります。

- ・ CacheDirect
- ・ シャドウイング
- ・ WebLink
- ・ NodeJS

13.3 仕組み

スーパーサーバは、事前定義されたポートでリッスンするプロセスです。InterSystems IRIS の最初にインストールされたインスタンスの場合、システムの既定のスーパーサーバ・ポート番号は 1972 です。追加のスーパーサーバを作成する場合は、作成時にポート番号を設定します。

各スーパーサーバが、事前定義されたポートで受信する TCP 要求を認識し（“サポートされている要求”を参照）、InterSystems IRIS との接続を適切に処理します。スーパーサーバは、クライアントや他の InterSystems IRIS インスタンスなどの外部接続と、スーパーサーバに関連付けられている InterSystems IRIS インスタンスとの間の中間層として機能します。作成するスーパーサーバごとに異なる要求を処理するように構成できます。例えば、システムの既定のスーパーサーバを、システム接続と CSP/REST 接続のみを受け入れるように設定します。次に、2 つ目のスーパーサーバを、クライアントとその他の接続のいくつかの組み合わせを処理するように構成し、3 つ目のスーパーサーバを、別の一連の接続を処理するように構成できます。

13.3.1 システムの既定のスーパーサーバ

InterSystems IRIS には、システムの既定のスーパーサーバとして 1 つのスーパーサーバがあらかじめインストールされています。このスーパーサーバを、クライアント接続、システム接続（ECP、ミラーリング、シャードイングなど）、その他の接続、および管理ポータルへのアクセスを処理するように構成できます。このスーパーサーバを変更する方法の詳細は、“管理ポータルを使用したシステムの既定のスーパーサーバの変更” および “CPF と CMF の使用” を参照してください。

13.4 構成方法

スーパーサーバの構成は、管理ポータル、`SECURITY ルーチン、または構成マージ・ファイルを使用して行うか、Security.Servers クラスを使用してプログラムによって行えます。

13.4.1 管理ポータルの使用

スーパーサーバ構成の定義はすべて、[システム]→[セキュリティ管理]→[スーパーサーバ] で確認できます。

13.4.1.1 管理ポータルを使用したスーパーサーバの作成

新しいスーパーサーバを作成するには、[スーパーサーバの新規作成] をクリックします。これにより、以下のフィールドが含まれる構成設定のフォームが表示されます。

- ・ **[説明]** – オプション。このスーパーサーバを説明するテキスト。
- ・ **[ポート]** – 必須。システム上の有効な未使用ポートの番号。スーパーサーバはこのポートでリッスンします。
- ・ **[バインドアドレス]** – オプション。スーパーサーバがリッスンするネットワーク・インタフェースを表す IP アドレスまたは DNS ホスト名。例えば、値 127.0.0.1 を指定すると、スーパーサーバは localhost にバインドされます。このバインド・アドレスが指定されたスーパーサーバは、ローカル接続のみを受け入れます。
- ・ **[有効]** – これを選択した場合、このスーパーサーバはアクティブに設定され、リッスンします。
- ・ **[Client Connections]** – TLS をサポートする接続。これには、以下のものがあります。
 - **[クライアント有効]** – JDBC および ODBC クライアント接続を許可します。ODBC または JDBC の詳細は、それぞれ “はじめに：インターシステムズ・データベースへの ODBC 接続” または “JDBC の基本” を参照してください。

- [CSP/REST有効] – Web API 接続を許可します。CSP または REST の詳細は、それぞれ [“CSP ベースの Web アプリケーションの概要”](#) または [“REST サービスの作成の概要”](#) を参照してください。

重要 管理ポータルへのアクセスを維持するには、システムの既定のスーパーサーバに対して [CSP/REST有効] を選択しておく必要があります。

- [DataCheck 有効] – データ整合性確認のための接続を許可します。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。

InterSystems IRIS スーパーサーバは、従来のクライアント接続もサポートします。[レガシー設定] をクリックすると、その他の接続オプションが表示されます。これには、以下のものがあります。

- [CacheDirect 有効] – 詳細は、[“Introduction to Caché Direct”](#) を参照してください。
- [シャドウ有効] – 詳細は、[“シャドウイング”](#) を参照してください。

クライアント接続では、通信の暗号化のために TLS がサポートされます。スーパーサーバに対する TLS の構成オプションは、以下のとおりです。

- [SSL/TLS support level] – TLS を使用するための接続の要件レベル。以下の 3 つのオプションがあります。
 - ・ [無効] – クライアント通信で TLS を使用しません。
 - ・ [有効] – クライアント通信で TLS を使用できます。
 - ・ [必須] – クライアント通信で TLS を使用する必要があります。
- [SSL/TLS構成] – [SSL/TLS support level] が [有効] または [必須] の場合は必須。詳細は、[“TLS の構成”](#) を参照してください。
- ・ [System Connections] – システムの既定のスーパーサーバのみがシステム接続に対応します。これには、以下のものがあります。
 - [ECP 有効] – 詳細は、[“分散キャッシュによるユーザ数に応じた水平方向の拡張”](#) を参照してください。
 - [ミラー有効] – 詳細は、[“ミラーリングの概要”](#) を参照してください。
 - [シャード有効] – 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。
- ・ [Other Connections] – TLS をサポートしない接続。これには、以下のものがあります。
 - [SNMP 有効] – Windows システムでのみ、Simple Network Management Protocol (SNMP) 接続を許可します。詳細は、[“SNMP を使用した InterSystems IRIS の監視”](#) を参照してください。

InterSystems IRIS スーパーサーバは、他の従来の接続もサポートします。これらの接続を構成するには、[レガシー設定] をクリックします。オプションは以下のとおりです。

- [Enable WebLink] – 詳細は、[“Caché WebLink Developer Guide”](#) を参照してください。
- [NodeJS 有効] – 詳細は、[“Using Node.js with Caché”](#) を参照してください。

13.4.1.2 管理ポータルを使用したスーパーサーバの編集

スーパーサーバ構成を編集するには、スーパーサーバ定義リストで、変更するスーパーサーバのポート番号をクリックします。ポート番号、バインド・アドレス、およびシステムの既定のスーパーサーバの選択は、このページからは変更できません。

13.4.1.3 管理ポータルを使用したシステムの既定のスーパーサーバの変更

管理ポータルで、システムの既定のスーパーサーバのポート番号を変更できます。**[スーパーサーバポート番号]**は、**[システム]→[構成]→[メモリと開始設定]→(構成設定)**にあります。ポート番号がスーパーサーバ構成に関連付けられていない場合は、現在のシステムの既定のスーパーサーバのポート番号が、新しく定義されたポート番号に変更されます。ポート番号がスーパーサーバ構成に関連付けられている場合、このスーパーサーバ構成が、新しいシステムの既定のスーパーサーバになります。変更は、システムの再起動時に適用されます。

重要 管理ポータルへのアクセスを維持するには、システムの既定のスーパーサーバに対して **[CSP/REST有効]** を選択しておく必要があります。

13.4.2 ^SECURITY の使用

^SECURITY ルーチンには、以下のスーパーサーバ管理タスクを実行できるようにするメニュー駆動型のインタフェースが含まれます。

- ・ スーパーサーバの作成。
- ・ スーパーサーバの編集。
- ・ スーパーサーバの一覧表示。
- ・ スーパーサーバの削除。
- ・ スーパーサーバのエクスポート。
- ・ スーパーサーバのインポート。

^SECURITY の詳細は、“[コマンド行セキュリティ管理ユーティリティ](#)” を参照してください。

13.4.2.1 ^SECURITY を使用したスーパーサーバの作成

管理ポータルと同様に、^SECURITY を使用してスーパーサーバを作成できます。フィールドは、管理ポータルの場合と同じです。^SECURITY を使用してスーパーサーバを作成するには、以下の手順を実行します。

1. ターミナルを起動し、該当するインスタンスに対して十分な特権を持つユーザとしてログインします。
2. ターミナル・プロンプトで、**%SYS** ネームスペースに切り替えます。

```
>zn "%SYS"
```

3. ^SECURITY を実行します。

```
%SYS>do ^SECURITY
```

4. ^SECURITY で、オプション **[15]スーパーサーバの設定** を選択します。これにより、スーパーサーバのさまざまな管理オプションを使用できるようになります。スーパーサーバを作成するには、オプション **[1]スーパーサーバの作成** を選択します。
5. プロンプトに従って、作成するスーパーサーバ構成の詳細を入力します。フィールドは、管理ポータルの場合と同じです。詳細は、“[管理ポータルの使用](#)” を参照してください。

13.4.2.2 ^SECURITY を使用したスーパーサーバの編集

管理ポータルと同様に、^SECURITY を使用してスーパーサーバを編集できます。このためには、以下の手順を実行します。

1. ターミナルを起動し、該当するインスタンスに対して十分な特権を持つユーザとしてログインします。

- ターミナル・プロンプトで、%SYS ネームスペースに切り替えます。

```
>zn "%SYS"
```

- ^SECURITY を実行します。

```
%SYS>do ^SECURITY
```

- ^SECURITY で、オプション [15) スーパーサーバの設定] を選択します。これにより、スーパーサーバのさまざまな管理オプションを使用できるようになります。スーパーサーバを編集するには、オプション [2) スーパーサーバの編集] を選択します。
- 編集するスーパーサーバを特定するには、まずそのスーパーサーバに関連付けられているポート番号を入力します。バインド・アドレスの入力を求められたら、そのスーパーサーバに関連付けられているバインド・アドレスを入力します。バインド・アドレスが空の場合は、そのままプロンプトに沿って続行します。スーパーサーバを適切に特定して編集するには、これら両方の情報が必要です。

重要 既存のスーパーサーバ構成を編集するには、ポート番号とバインド・アドレスがその構成と完全に一致している必要があります。

13.4.3 CPF と CMF の使用

システムの既定のスーパーサーバのポート構成は、DefaultPort フィールドの [Startup] セクションにある構成パラメータ・ファイル (CPF) で定義されます。CPF で、システムの既定のスーパーサーバの既定のポートを変更できます。また、DefaultPortBindAddress フィールドで、既定のバインド・アドレスも変更できます。ポート番号がスーパーサーバ構成に関連付けられていない場合は、現在のシステムの既定のスーパーサーバのポート番号が、新しく定義されたポート番号に変更されます。ポート番号がスーパーサーバ構成に関連付けられている場合、このスーパーサーバ構成が、新しいシステムの既定のスーパーサーバになります。ポートとバインド・アドレスは、インスタンスの起動時に初期化されます。

構成マージ・ファイル (CMF) を使用してスーパーサーバを構成することもできます。CMF では、スーパーサーバに対し、Security.Servers クラスで定義されているプロパティを使用した、CreateServer、DeleteServer、および ModifyServer の 3 つのアクションをサポートしています。

13.4.4 プログラムの使用

Security.Servers クラスを使用して、プログラムによってスーパーサーバを構成できます。このクラスの使用の詳細は、クラス・リファレンスを参照してください。

14

OAuth 2.0 および OpenID Connect に対するサポート

このページでは、[OAuth 2.0](#) および [OpenID Connect](#) に対する InterSystems IRIS® のサポートについて説明します。

14.1 サポートされているシナリオ

OAuth 2.0 および OpenID connect に対する InterSystems IRIS のサポートにより、以下のいずれかまたはすべてを行うことができます。

- ・ クライアントとして InterSystems IRIS Web アプリケーションを使用する
- ・ リソース・サーバとして InterSystems IRIS Web アプリケーションを使用する
- ・ 承認サーバとして InterSystems IRIS インスタンスを使用する

例えば、サードパーティ・テクノロジーを採用した承認サーバのクライアントとして、InterSystems IRIS Web アプリケーションを使用することができます。また、InterSystems IRIS を基盤とする承認サーバとサードパーティのクライアントを使用できます。1 台以上のリソース・サーバを InterSystems IRIS または別のテクノロジーで実装することもできます。

いずれの場合でも、承認サーバが最も複雑な要素であり、通常は最初に作成します。クライアントはそのあと作成します。一般にクライアントを作成するときは、サポートするスコープなど、承認サーバの機能と要件に関する知識を必要とします。

14.2 OAuth 2.0 および OpenID Connect に対する InterSystems IRIS のサポート

OAuth 2.0 および OpenID Connect に対する InterSystems IRIS のサポートは、以下の要素で構成されます。

- ・ 管理ポータル構成ページ。
クライアント (またはリソース・サーバ) を構成する場合は、[システム管理]→[セキュリティ]→[OAuth 2.0]→[クライアント構成] のオプションを使用します。

承認サーバを構成する場合は、[システム管理]→[セキュリティ]→[OAuth 2.0]→[サーバ構成] のオプションを使用します。

- ・ **%SYS.OAuth2** パッケージのクラス。これらのクラスはクライアント API です。InterSystems IRIS Web アプリケーションを OAuth 2.0 クライアントとして定義すると、そのクライアントはこれらのクラスのメソッドを使用します。
- ・ **%OAuth2** パッケージのクラス。OAuth 2.0 承認サーバとして InterSystems IRIS インスタンスを使用する場合は、**%OAuth2.Server** パッケージの 1 つ以上のクラスからサブクラスを作成することでサーバをカスタマイズします。**%OAuth2** の他のクラスは、コードが呼び出すユーティリティ・メソッドを提供します。
- ・ **OAuth2** パッケージのクラス (IRISSYS データベース内)。このクラスには、InterSystems IRIS 内部で使用する永続クラスが含まれます。そのほとんどは無視できます。ただし、構成項目をプログラムで作成する場合は、このパッケージに含まれるクラスのサブセットを使用することになります。

以下のサブセクションに構成項目の概要を示します。

14.2.1 クライアントの構成項目

OAuth 2.0 クライアントとして機能している InterSystems IRIS インスタンス内で、サーバ記述 (承認サーバを記述) とクライアント構成 (クライアントを構成) という 2 つの結合された構成項目を指定クライアント・アプリケーションに対して定義する必要があります。特定のインスタンスでは、任意の数のサーバ記述を定義できます。下図が示すように、それぞれのサーバ記述は複数のクライアント構成があります。この図は、これらの構成項目に格納されている情報の一部も示しています。



このアーキテクチャにより、同じ承認サーバを使用する複数のクライアント構成を定義することが可能になり、承認サーバの詳細を繰り返し定義する必要がなくなるため構成を簡素化することができます。

“[OAuth 2.0 クライアントとしての InterSystems IRIS Web アプリケーションの使用法](#)” に記載されているように、これらの項目は管理ポータルで作成できます。また、“[プログラムによる構成項目の作成方法](#)” に記載されているように、プログラムで構成項目を作成することもできます。

14.2.2 サーバの構成項目

OAuth 2.0 承認サーバとして機能している InterSystems IRIS インスタンス内で、サーバ構成 (承認サーバを構成) と複数のクライアント記述を定義する必要があります。下図は、これらの構成項目に格納されている情報の一部を示しています。



特定の InterSystems IRIS インスタンスは、1 つのサーバ構成と複数のクライアント記述を持つことができます。クライアント・アプリケーションごとに 1 つのクライアント記述が必要になります。クライアント記述は、承認サーバのエンドポイントを使用する各リソース・サーバでもそれぞれ 1 つ必要になります。承認サーバのエンドポイントを使用しないリソース・サーバでは、クライアント記述を作成する必要はありません。

“[OAuth 2.0 承認サーバとしての InterSystems IRIS の使用法](#)” に記載されているように、これらの項目は管理ポータルで作成できます。また、“[プログラムによる構成項目の作成方法](#)” に記載されているように、プログラムで構成項目を作成することもできます。

14.3 InterSystems IRIS でサポートされる標準規格

このセクションでは、OAuth 2.0 および OpenID Connect に対して InterSystems IRIS がサポートしている標準規格を示します。

- ・ The OAuth 2.0 Authorization Framework (RFC 6749) – <https://datatracker.ietf.org/doc/rfc6749> を参照
- ・ The OAuth 2.0 Authorization Framework: Bearer Token Usage (RFC 6750) – <https://datatracker.ietf.org/doc/rfc6750> を参照
- ・ OAuth 2.0 Token Revocation (RFC 7009) – <https://datatracker.ietf.org/doc/rfc7009> を参照
- ・ JSON Web Token (JWT) (RFC 7519) – <https://datatracker.ietf.org/doc/rfc7519> を参照
- ・ OAuth 2.0 Token Introspection (RFC 7662) – <https://datatracker.ietf.org/doc/rfc7662> を参照
- ・ OpenID Connect Core 1.0 – http://openid.net/specs/openid-connect-core-1_0.html を参照
- ・ OAuth 2.0 フォーム送信レスポンス・モード – http://openid.net/specs/oauth-v2-form-post-response-mode-1_0.html を参照
- ・ JSON Web Key (JWK) (RFC 7517) – <https://datatracker.ietf.org/doc/rfc7517> を参照
- ・ OpenID Connect Discovery 1.0 – https://openid.net/specs/openid-connect-discovery-1_0.html を参照
- ・ OpenID Connect ダイナミック・クライアント登録 – http://openid.net/specs/openid-connect-registration-1_0-19.html を参照
- ・ OAuth 2.0 クライアント認証および承認付与用 JSON Web トークン (JWT) プロファイル (RFC 7523) – <https://tools.ietf.org/html/rfc7523> を参照してください。
- ・ Proof Key for Code Exchange (RFC 7636) – <https://tools.ietf.org/html/rfc7636> を参照してください。

15

タスク・マネージャの電子メール設定の構成

“[タスク・マネージャの使用](#)”で説明しているとおり、タスクが完了したら電子メール通知を送信するようにタスク・マネージャを設定できます。[\[タスクマネージャ email 設定\]](#) ページ ([\[システム管理\]](#)→[\[構成\]](#)→[\[追加の設定\]](#)→[\[タスクマネージャ Eメール\]](#)) で、このページで説明する通知設定を構成できます。

15.1 電子メール構成の設定

SMTP サーバとポート

送信 SMTP (Simple Mail Transfer Protocol) メール・サーバのアドレスとポート。

SSL 構成

SSL/TLS を使用して電子メールを暗号化する場合に使用する SSL 構成。インスタンスに SSL 構成がない場合、または新たな SSL 構成を作成する場合は、“[TLS 構成の作成または編集](#)”を参照してください。SSL 構成を選択しないと、SSL/TLS は使用されません。

SMTP 認証ユーザとパスワード

SMTP サーバに対する SMTP 認証でのみ必要です。詳細は、“RFC 2554”を参照してください。エントリを入力しないと、SMTP のユーザ名とパスワードは NULL に設定されます。

送信者

SMTP サーバに対する SMTP 認証でのみ必要です。詳細は、“RFC 2554”を参照してください。

返信先

受信者が返信に使用する電子メール・アドレス。

成功時の件名

正常に実行されたタスクのメッセージの、フォーマットされた件名行。後述の“[件名とメッセージのパラメータ](#)”を参照してください。

成功時のメッセージ

タスクが正常に実行された後に送信される、フォーマットされたメッセージ。

失敗時の件名

実行に失敗したタスクのメッセージの、フォーマットされた件名行。

失敗時のメッセージ

タスクの実行に失敗した後に送信される、フォーマットされたメッセージ。

注釈 また、%SYS.Task.Config クラスにより、プログラムで電子メールの設定を構成できます。

15.2 件名とメッセージのパラメータ

件名およびメッセージのテキスト・ボックスの情報は、Web ページの下部および以下のテーブルに示すタスク・パラメータを使用してフォーマットします。Web ページには、例も用意されています。

タスク・パラメータ	説明
ID	タスクの ID
DESCRIPTION	タスクの説明
NAME	タスク名
LASTSTARTED	タスクが最後に開始された時刻
LASTFINISHED	タスクが最後に終了した時刻
SCHEDULED	予定されている最後の開始時刻
CURRENTDATE	電子メールが送信された日付
CURRENTTIME	電子メールが送信された時刻
STATUS	タスクの返り値 : <ul style="list-style-type: none"> ・ タスクが成功した場合、" " が返されます。 ・ タスクによりエラーがトラップされた場合、%Status の値のテキストが返されます。 ・ タスクによりトラップされないエラーが発生した場合、\$ZERROR が返されます。
TASKCLASS	このタスクで使用されるタスク・クラス。例えば、データベース整合性チェック・タスクの場合は %SYS.Task.IntegrityCheck。
ERROR	タスクが正常に実行されなかった場合のエラー・コード
SUCCESS	タスクが正常に実行された場合の完了メッセージ

16

各国言語サポート (NLS) の構成

各国語ロケールはすべてのテキスト・データが InterSystems IRIS® データ・プラットフォームによりエンコードされる文字セットを定義します。文字セットは 16 ビット Unicode UCS-16 です。

各ロケールには、テキスト表示、データ照合 (“照合” を参照)、大文字および小文字間の変換、パターン照合などに InterSystems IRIS が使用する多くの文字テーブルがあります。各ロケールはこれらの目的のそれぞれに使用されるテーブルを定義するだけでなく、日付、時間、数字形式など、その他の詳細も定義します。

各 InterSystems IRIS インスタンスは、単一の現在のロケールを使用します。これはインスタンスのインストール時に決定されますが、いつでも変更可能です。現在のロケールを変更すると、InterSystems IRIS により使用されるロケール・テーブルの一部またはすべてが変更されます。

新しいロケールのインストールによって、データが変換されることはありませんが、データの表現方法は変わります。

新しいロケールのインストールの目的は、主にアップグレード・オプションや、インストールの選択を修正する手段であり、頻繁な実行はお勧めできません。データ変換が必要になる場合があり、グローバル・サブスクリプトには特別な注意を払う必要があることを忘れないようにしてください。

InterSystems IRIS により提供されたシステム・ロケールを変更することはできません。これらは、インスタンスのアップグレード時に上書きされます。

16.1 管理ポータルでの NLS ページの使用法

[各国言語の設定] ページ ([システム管理]→[構成]→[各国言語の設定]) では、既存のロケールとテーブルの参照だけでなく、カスタム・ロケールの作成もできます。管理ポータルを使用して、新しい現在のロケールのインストール、メモリへの新規テーブルの読み込みなどが可能です。[システム管理]→[構成]→[各国言語の設定] を選択すると、右側の列で次のオプションが利用可能になります。

- ・ [構成済み既定値](#)
- ・ [ロケール定義](#)
- ・ [ロケールのインポート](#)

16.1.1 構成済み既定値

[構成済み既定値] ページ ([システム管理]→[構成]→[各国言語の設定]→[構成済み既定値]) では、InterSystems IRIS 内で各目的のために既定で現在使用されているロケール・テーブルが表示されます。ObjectScript コードを記述する場合、またはいくつかのユーティリティを使用する場合、特定の目的のために特定のテーブルを指定することができます。テーブルが指定されない場合、既定のテーブルが使用されます。

各テーブル名は、設定がインストール時に現在のロケールから継承されたのか、または NLS クラス・パッケージ (“[各国言語サポートのシステム・クラスの用法](#)” で説明) を使用して指定されたのかを示すために色分けされます。

構成既定値は、ロケールのプロパティではなく、インスタンスのプロパティです。したがって、インスタンスのアップグレード時には既定の選択が保持されます。

16.1.2 ロケール定義

[**ロケール定義**] ページから ([**システム管理**]→[**構成**]→[**各国言語の設定**]→[**ロケール定義**]、[**ロケール選択**] ドロップダウンでロケールを選択し、いくつかのアクションを実行できます。このページが最初に表示される時、ドロップダウンは常に現在のロケールに設定されます。

- ・ **[[現在のロケール] の日付/時間/数字形式を使用する]** ドロップダウンを使用して、現在のロケールによって指定されている日付、時間、および数字形式を使用するかどうかを示します。これは常に現在のロケールに適用されます。**[ロケール選択]** ドロップダウンで選択したがインストールされていないロケールではないので注意してください。
- ・ 選択したロケールの詳細を表示するには、**[プロパティ]** をクリックします。次のページには、カテゴリにグループ分けされたロケールのプロパティが表示されます。追加したロケールの場合、フィールドを編集して **[保存]** をクリックすることでこれらの変更を保存できます。InterSystems IRIS により提供されたシステム・ロケールの編集はできません。プロパティは次のとおりです。

- **基本のプロパティ**

- **日付、時刻、および数値の形式**

- **内部テーブル** – 内部テーブルを編集するときには次の 2 つのオプションがあります。

- ・ **テーブルの編集** – 項目をダブル・クリックすることにより、リスト・ボックスからテーブルを選択または削除できます。または、項目を選択してから **[>]** または **[<]** ボタンをクリックすることにより、該当するリストからその項目を移動させることもできます。

1 つ以上のエントリが必要なテーブルには、アスタリスク (*) が付いています。その他のテーブルは空のままにしておけます。

- ・ **デフォルト編集** – **[内部テーブル]** カテゴリの **[テーブル編集]** 機能に入力する値から既定の値を選択できます。

- **入力/出力テーブル** – このカテゴリの編集を選択すると、テーブルの編集、追加、および削除を実行できます。

- ・ テーブルを編集するには、最初のリストからテーブルをクリックします。下のボックスにテーブル名が表示されます。値を変更して、**[保存]** をクリックします。
 - ・ テーブルを削除するには、最初のリストからテーブルをクリックします。下のボックスにテーブル名が表示されます。**[削除]** をクリックします。削除に対して **[キャンセル]** または **[OK]** を選択できる確認ボックスが表示されます。
 - ・ テーブルを追加するには、**[追加]** をクリックします。下のボックスの **[テーブル]** フィールドが有効になり、**[削除]** オプションが無効になります。テーブル名を入力して、**[出力先]** フィールドと **[入力元]** フィールドに入力します。

更新をすべて完了したら、**[保存]** をクリックします。正常に保存されると、更新したリストが表示されます。正常に保存されなかった場合は、該当するエラー・メッセージが表示されます。

- **入出力既定値**

- **文字列**

- ・ 高度な操作をするには、次のボタンをクリックします。

- **検証** - 選択されたロケールを検証します。ロケールを検証できなかった場合は、エラー・メッセージを表示します。これはカスタム・ロケールを作成する場合に役立ちます。
- **コピー** - 選択されたロケールのコピーを作成します。これにより、コピーをカスタマイズできます。コピーの名前には、`y` で始まり、8 または `w` で終わる 4 文字を使用する必要があります。既定の説明は、`Copy of %locale` です。ここで `%locale` は、選択したロケール名です。コピーが作成されると、**[ロケール選択]** ドロップダウンに追加されます。
- **エクスポート** - ロケールを `.xml` ファイルにエクスポートします。例えば、作成したカスタム・ロケールをエクスポートし、**[ロケールのインポート]** ページを使用して別のインスタンスにインポートします。既定の名前は、`loc_%locale.xml` です。ここで `%locale` は、選択したロケールです。さらに、エクスポート・ファイルのパスを組み込むことができます。パスを指定しない場合、既定の場所は `install-dir¥mgr` です。
- **インストール** - 選択されたロケールをインスタンスの現在のロケールとしてインストールします。最初の検証が実行され、失敗した場合、エラー・メッセージが表示されます。それ以外はインストールを続行できます。
- **テーブル読み込み** - 選択したロケール（現在のロケールまたは別のロケール）からディスクのメモリにテーブルを読み込むことができます。テーブル・タイプを選択します。次に、タイプ選択後に生成されたリストからテーブル名を選択します。**[OK]** をクリックすると、テーブルが読み込まれます。**[キャンセル]** をクリックすると、ダイアログ・ボックスが閉じられ、**[ロケール定義]** ページに戻ります。
- **削除** - ロケールを削除します。削除できるのは、カスタム・ロケールのみです。システム・ロケールが選択された場合、ボタンは無効化されます。カスタム・ロケールでも現在のロケールは削除できません。続行するにはロケールの削除を確定する必要があります。

16.1.3 ロケールのインポート

[ロケールのインポート] ページで (**[システム管理]**→**[構成]**→**[各国言語の設定]**→**[ロケールまたはテーブルのインポート]**)、ロケールまたはテーブルをインポートできます。例えば、(前のセクションで説明したように) 別のインスタンスからエクスポートされたカスタム・ロケールをインポートできます。

1. **[インポートタイプ]** を選択します。**[ロケール]** が既定値です。
2. ファイル名を入力して、**[インポート]** をクリックします。有効なファイル拡張子は、`.xml` と `.goq` のみです。
3. インポートしたロケール数、テーブル数、サブテーブル数を示すメッセージが表示されます。

16.2 NLS クラス・パッケージの使用法

“**各国言語サポートのシステム・クラス**” では、**%SYS.NLS** クラス・パッケージと **Config.NLS** クラス・パッケージの使用法を詳しく説明しています。

“**%SYS.NLS クラス**” のセクションには、次のクラスの使用法の詳細が示されています。

- ・ **%SYS.NLS.Device** - 現在のデバイスのプロパティ
- ・ **%SYS.NLS.Format** - 日付、時刻、および数値の形式
- ・ **%SYS.NLS.Locale** - 現在のロケールの基本のプロパティ (読み取り専用)
- ・ **%SYS.NLS.Table** - システムとプロセスのテーブル (入出力および内部)

“**Config.NLS クラス**” のセクションには、次のクラスの使用法の詳細が示されています。

- ・ **Config.NLS.Locales**

- ・ `Config.NLS.SubTables`
- ・ `Config.NLS.Tables`

これらのクラスそれぞれの詳細は、“インターシステムズ・クラス・リファレンス” を参照してください。

17

作業キュー・マネージャ・カテゴリの構成

作業キュー・マネージャは、プログラムによって複数の同時プロセスに作業を分散し、パフォーマンスを向上させることができるようにします。詳細は、“[作業キュー・マネージャの使用](#)”を参照してください。

作業キュー・マネージャを使用してワーカ・ジョブのセットをコードで初期化する際に、そのコードでワーカ・ジョブを供給するカテゴリを指定できます。カテゴリは、ワーカの独立したプールであり、それぞれにワーカの最大数が設定されます。

システムは、ユーザが削除することのできない **Default** および **SQL** の 2 つのカテゴリを提供しています。既定では、これらのカテゴリのワーカの最大数は **【動的】** で、システムで利用可能なコアの 2 倍に相当します。

【ワークキューマネージャのカテゴリ】 ページでは、カテゴリを作成、変更、および削除できます（以降のセクションを参照）。あらゆる変更が **構成パラメータ・ファイル** `iris.cpf` に反映されます。

17.1 作業キュー・マネージャ・カテゴリの作成

【Work Queue Manager Categories】 ページでは、新しいカテゴリを作成できます。

カテゴリ名は一意である必要があり、大文字と小文字を区別します。また、カテゴリ名は、文字、数字、ダッシュ、アンダースコア、ピリオドのみを使用して、64 文字以内とします。

新しいカテゴリを作成する手順は以下のとおりです。

1. **【システム管理】**→**【構成】**→**【システム構成】**→**【WQM Categories】** の順に移動します。
2. **【Create Category】** をクリックします。
3. **【カテゴリの名前】** フィールドに、カテゴリの名前を入力します。
4. このカテゴリでリクエストを処理するジョブのプールに保持するアクティブ・ワーカ・ジョブの最大数を、**【最大アクティブワーカ】** フィールドで選択します。InterSystems IRIS は、アイドル状態のジョブを検出し、アクティブ・ジョブの最大数をこの上限値近くで保持できるように新しいジョブを自動的に開始します。既定では、システム・リソースに基づき、16 を最大ワーカ数として、この上限値が動的に調整されます。
5. このカテゴリのワーク・グループを作成して、そのワーカ・ジョブ数を指定していない場合は、ワーク・グループに既定で割り当てるワーカ・ジョブの数を **【デフォルトワーカ】** フィールドで選択します。既定では、システム・リソースに基づき、8 を最大ワーカ数として、この値が動的に調整されます。
6. **【最大ワーカ】** フィールドで、このカテゴリでリクエストを処理するジョブのプールに保持するワーカ・ジョブの最大数を選択します。このカテゴリのワーク・グループでリクエストされているジョブの数がこの上限値を超えている場合は、この最大数のワーカのみが用意されます。既定では、システム・リソースに基づき、16 を最大ワーカ数として、この値が動的に調整されます。

17.2 作業キュー・マネージャ・カテゴリの編集または削除

[Work Queue Manager Categories] ページから、次の手順で、既存のカテゴリの最大ワーカ数を変更し、ユーザ指定のカテゴリを削除できます。

1. [システム管理]→[構成]→[システム構成]→[WQM Categories] の順に移動します。
2. 該当するカテゴリの [編集] または [削除] をクリックします。

18

ローカル・データベースの管理

管理ポータルの [データベース] ページ ([システム処理]→[データベース]) では、ローカル・データベースの確認と保守ができます。このページには、次の情報が表示されます。

- ・ [データベースの一般情報](#) – すべてのデータベースの概要情報。
- ・ [データベースの空き容量情報](#) – すべてのデータベースの空き容量情報。
- ・ [データベースの詳細ページ](#) – 各データベースに固有の情報。このページから、空き容量を増やすためのメンテナンスを実行することもできます。

ページ上部のボタンを使用して、整合性チェックを実行することもできます。詳細は、“[構造的な整合性の検証](#)” を参照してください。

このセクションで説明しているほとんどの情報および操作は、`DATABASE` コマンド行ユーティリティにも存在し、これを使用して実行することもできます。

18.1 データベースの一般情報

[データベース] ページ ([システム処理]→[データベース]) には、すべてのローカル・データベースのリストが含まれています。各ローカル・データベースについて以下の情報が表示されます。

テーブル 18-1: ローカル・データベース情報

列名	定義
名前	データベース名。この名前をクリックすると詳細が表示されます。“ データベースの詳細ページ ” を参照してください。
ディレクトリ	データベースが存在するシステム・ディレクトリ。
最大サイズ (GB)	割り当てられた許容されるデータベースの最大サイズ。ギガバイト単位。
サイズ (MB)	現在割り当てられているデータベースのサイズ。メガバイト単位。 注釈 このフィールドは、データベースの論理サイズを測定します。その結果、ここで報告されるサイズが、データベースの物理サイズより小さくなる場合があります (特に IRISTEMP データベースの場合)。

列名	定義
状態	<p>データベースの状態。マウントされているかマウントされていない（ディスマウントされている）かと、設定されているアクセス権を示します。</p> <ul style="list-style-type: none"> マウントされているデータベースとは、[開始時にマウントが必要] が選択されており（“データベース・プロパティの編集” を参照）、InterSystems IRIS® データ・プラットフォームを起動したりミラー・プライマリにするには、マウントされる必要があるデータベースです。この場合、InterSystems IRIS が起動されるたびにマウントされ、アクセス可能となります。または、以前にマウント解除され、アクセスしたり明示的にマウントした際に動的にマウントされたデータベースです。この場合、明示的にディスマウントしたり InterSystems IRIS の再起動や停止が行われるまで、マウントされたままになります。 マウントされていないデータベースとは、[開始時にマウントが必要] が選択されておらず、したがって、InterSystems IRIS を起動したりミラー・プライマリにするためにマウントされる必要がなく、アクセスされたり明示的にマウントされていないデータベースです。アクセスしたり明示的にマウントすると動的にマウントされ、明示的にディスマウントしたり InterSystems IRIS の再起動や停止が行われるまで、マウントされたままになります。 ディスマウントされたデータベースとは、明示的にディスマウントされたデータベースのことです。明示的にマウントしたり InterSystems IRIS の再起動や停止が行われるまで、アクセスできません（つまり、アクセスしても、ディスマウントされたデータベースは動的にマウントされません）。永久にデータベースをディスマウントするには、構成から削除する必要があります。詳細は、このドキュメントの “InterSystems IRIS の構成” の章の “ローカル・データベース” を参照してください。
暗号化	データベースが暗号化されているかどうかを表示します。
ジャーナル	データベースのグローバルがジャーナルされているかどうかを [] または [] で示します。

また、表示するデータベースの数の制御に使用できるフィルタ・バーをページは含んでいます。例えば、システム・データベースのみをリスト表示するには、[フィルタ:] テキスト・ボックスに IRIS* を、ページごとにデータベースを 5 つだけリスト表示するには、[ページサイズ] テキスト・ボックスに 5 を、表示する行数を 3 行に制限するには、[最大行数:] テキスト・ボックスに 3 をそれぞれ入力します ([結果] フィールドに数字付きで表示される [+] 記号は、指定された基準に合うデータベースがその他にあることを示していますが、表示はされません)。

18.2 データベースの空き容量情報

データベースの空き容量（空のブロック）の管理は、データベース・メンテナンスの重要な側面です。空き容量の情報を確認するには、管理ポータルの [\[空き容量\] ビュー](#) を表示するか、`%FREECNT` ユーティリティを使用します。通常運用時のデータベースのサイズと空き容量属性は常に変化しているため、ポータルやユーティリティで報告される特定の時点での数値は概算に過ぎないことに注意してください。

データベースが必要以上の空き容量を持つようにする場合、“[データベースの詳細ページ](#)” での説明のとおり、データベースを圧縮し、切り捨てることができます。

18.2.1 管理ポータルを使用した空き容量情報の表示

各ローカル・データベース上の空き容量に関する情報を示す、空き容量情報を表示するには、[データベース] ページ ([システム処理] → [データベース]) に移動し、[空き容量ビュー] ラジオ・ボタンをクリックします。以下のテーブルは、表示される情報の説明です。

テーブル 18-2: ローカル・データベースの空き容量情報

列名	定義
名前	データベース名。この名前をクリックすると詳細が表示されます。“ データベースの詳細ページ ”を参照してください。
ディレクトリ	データベースのプライマリ・ボリュームが存在するシステム・ディレクトリ。
最大サイズ	最大割り当てサイズ (ギガバイト単位)。データベースは、このサイズまで拡大可能です。データベース作成時の既定値は無制限です。
サイズ	現在割り当てられているデータベースのサイズ。メガバイト単位。 注釈 このフィールドは、データベースの論理サイズを測定します。その結果、ここで報告されるサイズが、データベースの物理サイズより小さくなる場合があります (特に IRISTEMP データベースの場合)。
拡張サイズ	データベースを拡張できるサイズ (メガバイト単位)。データベースの作成時の既定値はゼロ (0) に設定することを推奨します。これは、システムの既定値 (現在のサイズの 12% または 10 MB のどちらか大きい方) を使用することを意味します。この設定では、拡張サイズは 1GB より大きくなりません。
使用可能	データベースで利用できる空き領域の容量 (メガバイト単位)。
空き (%)	データベースで利用できる空き領域の割合。
ディスク空き容量	ボリュームの空き領域の容量。
状態	ディレクトリの状態。データベースがマウントされているかどうかと、アクセス許可を示します。

管理ポータルからの空き容量管理の実行の詳細は、“[データベースの詳細ページ](#)”を参照してください。

18.2.2 %FREECNT を使用した空き容量情報の表示

InterSystems IRIS には、データベースで使用可能な空き容量を表示する %FREECNT ユーティリティが用意されています。このユーティリティは、ターミナルで `do ^%FREECNT` と入力して実行します。

%SYS ネームスペースで ^%FREECNT を使用している場合は、プロンプトにアスタリスク (*) を入力してすべてのデータベースの空き容量を表示するか、1 つのデータベース・ディレクトリ名を入力するかを選択できます。次に例を示します。

```
%SYS>do ^%FREECNT
```

```
Database directory to show free space for (*=All)? *
```

```
Databases Selected
```

```
-----
c:\MyIris\mgr\
c:\MyIris\mgr\irisaudit\
c:\MyIris\mgr\irislib\
c:\MyIris\mgr\irislocaldata\
c:\MyIris\mgr\iristemp\
c:\MyIris\mgr\user\
Device:
Right margin: 80 =>
```

```
Database Free Space
Feb 15 2012 7:25 PM
```

Database	Max Size	Size	Available	%Free	Disk Free
c:\MyIris\mgr\	Unlimited	191MB	19MB	9.94	60.79GB
c:\MyIris\mgr\irisaudit\	Unlimited	1MB	0.43MB	43	60.79GB
c:\MyIris\mgr\irislib\	Unlimited	319MB	27MB	8.46	60.79GB
c:\MyIris\mgr\irislocaldata\	Unlimited	1MB	0.55MB	55	60.79GB
c:\MyIris\mgr\iristemp\	Unlimited	4MB	1.5MB	37.5	60.79GB
c:\MyIris\mgr\user\	Unlimited	1MB	0.43MB	43	60.79GB

%SYS 以外のネームスペースでは、このユーティリティは、そのネームスペース内のデータベースの空き容量を示します。次に例を示します。

```
USER>Do ^%FREECNT
```

```
Databases Selected
-----
c:\MyIris\mgr\user\
Device:
Right margin: 80 =>
```

```

Database Free Space
Feb 15 2012 7:28 PM
Database      Max Size Size      Available %Free  Disk Free
c:\MyIris\mgr\user\ Unlimited 1MB      0.52MB    52    42.72GB
```

注釈 <- フラグ (%Free 列内) は、指定されたデータベースの空き容量の割合が 5% 未満に低下したことを示します。ファイル・システムでデータベースの拡張を処理できるだけの十分な領域を確保する必要があります。

空き容量情報を表示し、その他のデータベース・タスクを実施するもう一つの方法が、`DATABASE` ユーティリティです。例えば、[データベースのグローバルを圧縮](#)したり、データベースを再作成でき、これによりデータベースの名前やサイズを変更することなく既存のデータベース内のデータをクリアできます。ユーティリティの情報については、"[DATABASE](#)"を参照してください。

注釈 InterSystems IRIS によって使用されるデータ構造は自身で均衡を保つように構成されており、長時間使用し続けてもパフォーマンスが低下することはありません。パフォーマンスを回復するために、データベースを終了してデータやインデックスの再構築や圧縮を行う必要はありません。

18.3 データベースの詳細ページ

[データベースの詳細] ページには、データベースの詳細情報が表示されます。管理ポータルからこの情報にアクセスする手順は以下のとおりです。

1. [データベース] ページ ([システム処理]→[データベース]) を表示します。
2. 詳細を確認するデータベースの名前をクリックします。

[データベースの詳細] ページの上部に並んだボタンにより、以下のようなさまざまなデータベース操作を行うことができます。

- ・ **マウント / ディスマウント** - データベースをマウントまたはディスマウントします。明示的に変更したり InterSystems IRIS の再起動や停止が行われるまで、新しい状態は有効です。永久にデータベースをディスマウントするには、構成から削除する必要があります。
- ・ **圧縮** - データベース全体に分散した空き領域をデータベースの末尾に移動します。
- ・ **削除** - データベースの末尾の空き領域を基盤となるファイル・システムに返します。
- ・ **デフラグ** - グローバル・ブロックを再配置して、特定のグローバルを表すブロックが連続するようにします。

このページには、選択したデータベースに固有の情報も含まれています。これらは [\[一般情報\]](#) テーブルと [\[データベースサイズ\]](#) テーブルにまとめられています。

テーブル 18-3: 一般情報

フィールド	定義
ディレクトリ	このデータベースが存在するディレクトリ名。
リソース名	このデータベースに割り当てられたリソース名。
マウント	このデータベースがマウントされているかどうかを示します。
読み取り専用	このデータベースが読み取り専用としてマウントされているかどうかを示します。
読み取り専用の理由	このデータベースが読み取り専用としてマウントされている理由。
暗号化	このデータベースが暗号化されているかどうかを示します。
暗号化キーID	暗号化キーIDを示します。

テーブル 18-4: データベースサイズ

フィールド	定義
ブロックサイズ	このデータベースのブロック・サイズ (バイト)。
ブロック	現在のこのデータベース内のブロック数。
最大サイズ	このデータベースで許容される最大サイズ (MB)。
サイズ	このデータベースの現在のサイズ (MB)。
拡張サイズ	このデータベースが拡張される量 (MB)。
利用可能空き容量	このデータベース内の利用可能スペース。
% 空き容量	このデータベース内の空き容量の割合。
ディスク空き容量	このデータベースを含むディスク上の空き容量。
最終拡張時刻	このデータベースが最後に拡張された時刻。
フル	データベースが満杯であることを示します。

18.4 データベースの圧縮

データベースの圧縮では、グローバル・ブロックを再配置することで、データベース全体に分散された空き領域を、そのデータベースの末尾に移動します。その後、データベースの[領域の削除](#)を実行することで、空き領域を基盤となるファイル・システムに返すことができます。(グローバルも圧縮できます。“[データベースのグローバルの圧縮](#)”を参照してください。)

データベースを圧縮するときには、そのデータベースの末尾に配置する利用可能な空き領域の容量を指定し、最低でも指定した空き容量を末尾に配置するために、グローバル・ブロックを再配置する操作を行います。(この操作では、追加の空き領域を作成することはできません。そのため、利用可能容量の合計よりも多くの容量を末尾に配置することはできません。)

例えば、サイズが 50 MB のデータベースがあり、そのうちの 15 MB が空き領域になっていて、その空き領域のうちの 5 MB が既にデータベースの末尾に配置されているとします。このデータベースを圧縮して、5 MB より大きく 15 MB より小さい容量を指定すると、データベースの末尾の空き領域が指定した容量と等しくなるまで、グローバル・ブロックがデータベースの末尾から先頭に移動されます。この容量に 15 MB を指定すると、移動可能なすべてのグローバル・ブロックがデータベースの先頭に移動されます。

データベース領域を圧縮するには、以下の手順を実行します。

1. [データベース] ページ ([システム処理]→[データベース]) に移動します。
2. 圧縮するデータベースの名前をクリックします。これにより [\[データベースの詳細\]](#) ページに移動します。
3. ページ上部のリボン上の **[圧縮]** をクリックします。これにより、**[データベース圧縮]** ダイアログ・ボックスが表示されます。このページには、データベースの名前と場所、現在のサイズ、利用可能な空き容量の合計、および現在ファイルの末尾にある空き領域の量が表示されます。
4. **[ファイルの末尾でのターゲット空き容量 (MB)]** テキスト・ボックスに、ファイルの末尾に必要な空き容量を指定します。提示された範囲内の値を入力する必要があります。容量を入力したら、**[OK]** をクリックします。すべての空き容量が既に末尾にある場合や、空き容量がない場合、プロンプトは表示されず、**[OK]** ボタンも無効になります。

注釈 いくつかの理由から、この操作で移動される空き容量は、指定した容量よりも大きくなる可能性があります。逆に、報告される数値は概算であるため、表示された空き容量のすべてが、実際に移動できるわけではない可能性もあります。

5. タスクが完了すると、**[バックグラウンドタスク情報]** ダイアログ・ボックスが表示されます。オプションで、**[バックグラウンドタスク]** ページ ([システム処理]→[バックグラウンドタスク]) にタスクを表示することもできます。
6. **[閉じる]** をクリックすると、**[データベースの詳細]** ページが再表示されます。このページは、新しいデータベース情報を示すよう、自動的に更新されます。

注釈 データベースの圧縮操作は、通常のデータベース・アクティビティと並行して実行するように設計されています。ただし、この操作は、ある程度のシステム・リソースを消費するため、システムが極度に高負荷の状態にあると完了しないことがあります。このような理由から、この操作やその他のデータベース再編成操作 (グローバルの [圧縮](#) や [デフラグ](#) などを含む) は、高負荷時を避けて実行し、このような操作をシステム上で同時に複数実行しないようにすることをお勧めします。

18.5 データベースの領域の削除

データベースの未使用領域を削除すると、空き領域がデータベースの末尾から基盤となるファイル・システムに返されます。多くの場合、データベースの領域は [圧縮](#) 後に削除されます。圧縮すると、空き領域がデータベースの末尾に移動します。

データベースの未使用領域を削除するときには、そのデータベースのターゲット・サイズを指定します。データベースの末尾に十分な空き領域がある場合は、この操作により、ターゲット・サイズに合わせてデータベースを縮小する分の領域が削除されます。十分な空き領域がない場合は、削除可能なすべての領域が削除されます。(末尾に配置されるデータベースの空き領域の容量を調べるには、データベースを圧縮します。現在の合計空き容量と末尾の空き容量を表示するために、この操作を完了する必要はありません。)

注釈 この機能は、raw ボリュームを持つデータベースには使用できません。

データベース領域を削除するには、以下の手順を実行します。

1. [データベース] ページ ([システム処理]→[データベース]) に移動します。
2. 削除するデータベースの名前をクリックします。これにより [\[データベースの詳細\]](#) ページに移動します。
3. ページ上部のリボン上の **[削除]** をクリックします。選択したデータベースの名前、場所、およびメガバイト (MB) 単位の現在のサイズを示す **[データベース領域の削除]** ダイアログ・ボックスが表示されます。
4. **[ターゲットファイルサイズ (MB)]** に入力します。これは、現在のサイズより小さくする必要があります。その後、**[OK]** をクリックします。0 を入力すると、利用可能な空き領域をファイルの末尾からすべて削除します。

注釈 切り捨てを目的とした空き容量は、データの格納を目的とした空き容量とは異なります。特定の制御構造は、それらがデータ・ストレージに使用されていない場合は、切り捨て時に削除できます。これにより、報告される空き容量で違いが生じる可能性があります。

18.6 データベースのデフラグ

データベースのデフラグにより、データベース内のグローバル・ブロックが再編成され、特定のグローバルのデータを含むすべてのブロックが連続するようになります。この操作では、1 つのグローバルからのビッグ・ストリング・ブロックやポインタ・ブロックを順序どおりには配置しませんが、それらを連続する領域に配置します。このプロセスの一環として、**[データベースをデフラグ]** オプションでは、**[Compact globals in a database]** オプションと同じ方法ですべてのグローバルを圧縮します。ただし、目標密度は 70% に設定されています。(この目標密度がデータベースの現在のグローバル・ブロック密度よりも低い場合、データベースのサイズが増加することはありません。)

注釈 **IRISTEMP** データベースはデフラグできません。

一般に、デフラグを定期的に行う必要はありません。ただし、ワークロードによっては、特にデータベースの大部分を連続的に読み込むような場合は、グローバル・ブロックを連続的に編成することで良好な結果が得られます。

デフラグ処理には、データベースの末尾に特定の量の空き領域が必要になります。そのため、以下に示すような状況が考えられます。

- この操作の実行に必要な十分な空き領域がデータベースの末尾にある場合は、グローバルのデフラグ以外の変更がデータベースに加えられることなく操作が完了します。
- データベースに十分な空き領域がない場合は、データベースが必要に応じて拡張されます。デフラグの完了後に、データベースの**領域の削除**を実行すると、追加された空き領域を削除できます。
- データベース内に十分な空き容量はないが、かなりの空き容量を末尾に移動できる場合、これが通知されます。この場合、**[データベースをデフラグ]** オプションを選択する前にデータベースを**圧縮**すると、デフラグ操作を完了するために必要な拡張量を減らすことができます。

データベースのグローバルをデフラグするには、以下の手順を実行します。

- [データベース]** ページ (**[システム処理]**→**[データベース]**) に移動します。
- デフラグするデータベースの名前をクリックします。これにより **[データベースの詳細]** ページに移動します。
- ページ上部のリボン上の **[デフラグ]** をクリックします。選択したデータベースの名前、場所、およびメガバイト (MB) 単位の現在のサイズと、デフラグする必要のある空き容量を表すメッセージを示す **[データベースのデフラグ]** ダイアログ・ボックスが表示されます。
- [OK]** をクリックします。
- タスクが完了すると、**[バックグラウンドタスク情報]** ダイアログ・ボックスが表示されます。オプションで、**[バックグラウンドタスク]** ページ (**[システム処理]**→**[バックグラウンドタスク]**) にタスクを表示することもできます。
- [閉じる]** をクリックすると、**[データベースの詳細]** ページが再表示されます。このページは、新しいデータベース情報を示すよう、自動的に更新されます。

重要 デフラグ操作では、データベース内のすべてのデータが一時的に再配置されます。これは、この操作の実行前のデータベースのグローバル断片化の程度には関係ありません。後続の操作の実行では、ほぼ同量のリソースを消費しますが、それ以上の効果は得られません。

注釈 デフラグ操作は、通常のデータベース・アクティビティと並行して実行するように設計されています。ただし、この操作は、ある程度のシステム・リソースを消費するため、システムが極度に高負荷の状態にあると完了しないことがあります。このような理由から、この操作やその他のデータベース再編成操作([データベースの圧縮](#)や[グローバルの圧縮](#)などを含む)は、高負荷時を避けて実行し、このような操作をシステム上で同時に複数実行しないようにすることをお勧めします。

グローバルのデフラグでは、圧縮中にデータベースのサイズが一時的に増大することがあります。これによってデータベースが構成最大サイズに到達した場合 (このガイドの“InterSystems IRIS の構成”の章で[ローカル・データベース](#)を参照)、またはストレージ・ボリュームで利用可能な領域が不足するために拡張が不可能でない場合、操作はキャンセルされます。

18.7 データベースのグローバルの圧縮

データベース容量を管理するもう 1 つの方法は、`^DATABASE` ルーチンを使用してデータベースのグローバルを圧縮する方法です。グローバルを圧縮することにより、グローバル・データはより少数のブロックに統合され、データベースの空き容量が増加します。

グローバルの作成時および構成時に、InterSystems IRIS は、通常、グローバル・ブロックの容量の約 70% が満たされるようにデータを割り当てます。(完全に照合順で増加したグローバルは、90% 近くまで割り当てられることがあります。)一般に、InterSystems IRIS による自動的なグローバル・ブロック密度の管理で問題はありません。ただし、連続しないデータ削除のパターンによっては、平均のグローバル・ブロック密度が大幅に小さくなる場合があります。

注釈 データベース内の 1 つのグローバルに対するグローバル・ブロックについて、現在の密度をグローバル単位で確認するには、整合性チェックを実行して([構造的な整合性の検証](#)を参照)、各グローバルの `Data Level` 出力を調べます。

グローバルを圧縮するときには、目標のグローバル・ブロック密度 (既定では 90%) と、この密度に可能な限り近づくようにするためのデータの統合操作 (3 つのブロックに分散されたグローバル・データを 2 つのブロックに再編成するなど) を指定します。通常 (例外もありますが)、グローバルを圧縮すると、データベース内で利用可能な空き容量が大幅に増加します。(現在のデータベースのグローバル・ブロック密度よりも低い目標密度を指定しても、データベースのサイズが増加することはありません。)

データベースのグローバルを圧縮するには、以下の手順を実行します。

1. ターミナルを開いて、`%SYS` ネームスペースに変更します。
2. `do ^DATABASE` と入力して、メニューから [7) Compact globals in a database] を選択します。
3. 操作を実行するデータベースのディレクトリを指定します。複数のデータベースを指定するには、[Database directories to compact?] プロンプトで ? を入力して、番号のリストを入力します。
4. すべてのグローバルの圧縮を指示します。または、圧縮する個別のグローバルのリストを入力します。
5. 平均グローバル・ブロック密度の目標値を指定して、残りのプロンプトに応答し、確定します。

注釈 グローバルの圧縮操作は、通常のデータベース・アクティビティと並行して実行するように設計されています。ただし、この操作は、ある程度のシステム・リソースを消費するため、システムが極度に高負荷の状態にあると完了しないことがあります。このような理由から、この操作やその他のデータベース再編成操作（データベースの[圧縮](#)やグローバルの[デフラグ](#)などを含む）は、高負荷時を避けて実行し、このような操作をシステム上で同時に複数実行しないようにすることをお勧めします。

グローバル圧縮では、圧縮中にデータベースのサイズが一時的に増大することがあります。これによってデータベースが構成最大サイズに到達した場合（このガイドの“InterSystems IRIS の構成”の章で“[ローカル・データベース](#)”を参照）、またはストレージ・ボリュームで利用可能な領域が不足するために拡張が不可能でない場合、操作はキャンセルされます。

19

InterSystems IRIS プロセスの管理

InterSystems IRIS システムは多くのプロセスを実行します。アプリケーション・コードと InterSystems IRIS システム・コードは、プロセス内で実行されます。InterSystems IRIS プロセスは、以下の 3 つのカテゴリに分けられています。

- ・ ユーザ・プロセス: ユーザが InterSystems IRIS に接続したときに作成される
- ・ バックグラウンド・プロセス: ユーザが ObjectScript Job コマンドを発行したときに作成される、または管理ポータルやユーティリティ (“[\[バックグラウンド・タスク\] ページの使用](#)” を参照) で作成される
- ・ InterSystems IRIS システム・プロセス

このページでは、ユーザ・プロセスとバックグラウンド・プロセスの両方の意味でプロセスという用語を使用します。

19.1 使用可能なオプション

プロセスは以下のように管理ポータルを使用して管理および制御できます。

テーブル 19-1: プロセス管理機能

機能	ポータルから機能にアクセスする方法
プロセス情報の表示	[プロセス] ページ ([システム処理]→[プロセス]) が表示されます。
プロセスの詳細の表示	[プロセス] ページを表示し、選択したプロセスの右側の列で、[詳細] をクリックして、[プロセス詳細] ページを表示します。
プロセスの一時停止または再開	[プロセス] ページを表示し、選択したプロセスの右側の列で、[詳細] をクリックして、[プロセス詳細] ページを表示します。次に、必要に応じて操作バーの [一時停止] または [再開] をクリックします。
プロセスの終了	[プロセス] ページを表示し、選択したプロセスの右側の列で、[詳細] をクリックして、[プロセス詳細] ページを表示します。次に、必要に応じて操作バーの [終了] または [〈RESJOB〉 エラーで終了] をクリックします。
プロセス変数の表示	[プロセス] ページを表示し、選択したプロセスの右側の列で、[詳細] をクリックして、[プロセス詳細] ページを表示します。次に、[変数] タブをクリックしてプロセス変数を表示します。
ターミナルへのメッセージの送信	[プロセス] ページが表示されるので、[ブロードキャスト] ボタンをクリックして [ブロードキャスト] ダイアログを開きます。

19.2 プロセス情報の表示

システム上のアクティブな全プロセス、および各プロセスの基本情報を表示するには、[プロセス] ページ ([システム処理]→[プロセス]) に移動します。ページには、列内に各プロセスとその統計のテーブルが表示されます。

以下の表は、ディスプレイで利用できるプロセス情報を説明しています。

テーブル 19-2: プロセス・カラム情報

列名	定義
ジョブ番号	プロセス・テーブルのインデックス。
プロセスID	オペレーティング・システム・プロセス識別番号 (PID)。*
合計CPU時間 ミリ秒	プロセスの実行で使用されている、システムとユーザの CPU 時間の合計値 (ミリ秒)。
ユーザ	プロセスを所有しているユーザの名前
デバイス	プロセスが使用している現在のデバイス以下のように指定できます。 <ul style="list-style-type: none"> TCP IP_address:Port_number — インスタンスからのアウトバウンド接続用。 TCP Port_number — スーパーサーバ用。 //./nul — NULL デバイス。これはすべての出力を破棄します。これを読み取ろうとしてもデータはありません。
ネームスペース	プロセスが実行している場所のネームスペース
ルーチン	プロセスが現在実行しているルーチンの名前

列名	定義
コマンド	実行したコマンドの数。
グローバル	InterSystems IRIS の開始以降、更新を含めて、実行されたグローバル参照の数（データベースの読み取りと書き込み）。
状態	プロセス状態。各状態の説明については、“%SYS.ProcessQuery” クラスの “State” プロパティを参照してください。
クライアント名	プロセスに接続した、またはプロセスへの接続を開始したクライアント・システムの名前
クライアント EXE	プロセスを呼び出した実行可能プログラムの名前
クライアント IP	プロセスを開始したシステムの IP アドレス
O/S Username	オペレーティング・システムによりプロセスに割り当てられるユーザ名。
詳細	このプロセスを管理する権限を持っている場合に表示されるボタン。“ プロセスの詳細の表示 ” を参照してください。

* ユーザがプログラマ・モードで InterSystems IRIS を使用する場合、アスタリスク (*) はプロセス id に表示されます。プラス記号またはマイナス記号は、コールイン・プロセスの隣に表示されます。

- ・ + プロセスは InterSystems IRIS にあります。
- ・ - プロセスは InterSystems IRIS にありません。

コールイン API は、InterSystems IRIS 機能です。これを使用すると、C プログラム内から ObjectScript のコマンドや式を実行したり、評価できるようになります。

19.3 プロセスの詳細の表示

[プロセス詳細] ページに、プロセスの詳細情報が表示されます。管理ポータルからこの情報にアクセスする手順は以下のとおりです。

1. [プロセス] ページ ([システム処理]→[プロセス]) が表示されます。
2. 該当するプロセスの行で [詳細] をクリックします(このオプションは、ユーザにプロセスを管理する権限がある場合にのみ表示されます)。
3. 必要に応じて、[SQL table & statement info] チェック・ボックスにチェックを付けます。このオプションを選択すると、SQL 関連情報が追加表示されます。

このページには、選択したプロセスに固有の情報も含まれています。これらは [\[一般情報\]](#) テーブル、[\[クライアントアプリケーションの詳細\]](#) テーブル、および [\[実行詳細\]](#) テーブルにまとめられています。

テーブル 19-3: 一般情報

フィールド	定義
Process ID	このプロセスのプロセス ID (PID) 番号。
ユーザ名	このプロセスに現在ログインしているユーザの名前。
ログインロール	プロセスのログイン・ロール。
エスカレーションしたロール	プロセスに付与される追加ロール。これらのロールにログイン・ロールを加えたものが、プロセスに付与されるすべてのロール・セットとなります。

フィールド	定義
OS ユーザ名	オペレーティング・システムによりプロセスに割り当てられるユーザ名。
ネームスペース	プロセスを実行中のネームスペース。
プロセスの優先度	このプロセスの優先度レベル。
グローバル参照	このプロセスのグローバル参照数。
プライベート・グローバル参照	このプロセスのプライベート・グローバル参照数。
実行されたコマンド	このプロセスで実行したコマンドの数。
メモリ制限	このプロセスで使用できるように割り当てられたメモリ量 (KB)。
メモリ・ピーク	このプロセスで使用されるメモリ量のピーク値 (KB)。
使用メモリ	このプロセスで現在使用しているメモリ量 (KB)。
合計CPU時間	プロセスの実行で使用されている、システムとユーザの CPU 時間の合計値 (ミリ秒)。
プライベートグローバルブロック	このプロセスで使用されるプライベート・グローバル・データ・ブロックの数。
現在のデバイス	このプロセスで現在使用している I/O デバイス名。
開かれたデバイス	このプロセスで現在開かれているデバイスのリスト。
ロック	このプロセスのロック情報。詳細 (モード、カウント、完全参照) を表示するには、詳細ボックス上部のリンクをクリックします。

テーブル 19-4: クライアント・アプリケーションの詳細

フィールド	定義
クライアント名	このプロセスに接続した、またはプロセスへの接続を開始したクライアントのノード名 (存在する場合)。
EXE 名	このプロセスに接続された実行可能クライアント・アプリケーションのクライアント名 (存在する場合)。
クライアント IP アドレス	このプロセスに接続された実行可能クライアント・アプリケーションのクライアントの IP アドレス (存在する場合)。
情報	ユーザ定義の情報 (存在する場合)。

テーブル 19-5: 実行詳細

フィールド	定義
プロセス状態	このプロセスの現在の実行状態。
トランザクション中	このプロセスが現在トランザクションで処理されているかどうかを示します。
最終グローバル参照	このプロセスで参照された最後のグローバル。
最後に行ったSQLテーブル参照	このプロセスで最後に参照された SQL テーブル (ある場合) (このデータを表示するには、[SQL table & statement info] チェック・ボックスにチェックを付けます)。
ルーチン	このプロセスが現在実行しているルーチンの名前。
ソースの場所	このプロセスで最後にレポートされたソースの場所 (ルーチン名とオフセット)。

フィールド	定義
ソース行	最後にレポートされたこのプロセスでの実行ソース・コードの行数（取得可能な場合）。

プロセスの停止

このページから、プロセスを停止または再開することもできます。プロセスは、以下のいずれかの方法で停止できます。

- ・ [プロセスの一時停止](#)。後でプロセスを再開することを前提としています。
- ・ [プロセスの終了](#)。プロセスを完全にキャンセルします。

19.3.1 プロセスの一時停止または再開

プロセスの実行内容が不明で確認を要する場合や、より重要なプロセスを実行するために CPU サイクルが必要な場合にプロセスの一時停止が必要になることがあります。管理ポータルからこのオプションにアクセスする手順は以下のとおりです。

1. **[プロセス]** ページ (**[システム処理]**→**[プロセス]**) が表示されます。
2. 該当するプロセスの行で **[詳細]** をクリックしますこのオプションは、ユーザにプロセスを管理する権限がある場合にのみ表示されます。
3. オプション・バーで **[一時停止]** をクリックします。

同じページの **[再開]** をクリックすれば、いつでも一時停止のプロセスを再開できます。

19.3.2 プロセスの終了

プロセスが応答しなくなったり、他のプロセスやユーザに影響を及ぼす場合は、終了できます。管理ポータルからこのオプションにアクセスする手順は以下のとおりです。

1. **[プロセス]** ページ (**[システム処理]**→**[プロセス]**) が表示されます。
2. 該当するプロセスの行で **[詳細]** をクリックします(このオプションは、ユーザにプロセスを管理する権限がある場合にのみ表示されます)。ポータルに、選択したプロセスの **[プロセスの詳細]** ページが表示されます。
3. オプション・バーで **[終了]** をクリックします。

必要に応じてプロセスの終了時の状態をログに記録するには、**[RES JOB エラーで終了]** チェック・ボックスにチェックを付けます。

注釈 このオプションは既定で有効となります。

4. 選択したプロセスを終了するために **[はい]** をクリックします。終了したプロセスは、二度と再開できません。

19.3.3 プロセス変数の表示

[プロセス変数] ページには、選択したプロセスで使用されているすべての変数が、グローバルの名前と値と共に表示されます。管理ポータルからこの情報にアクセスする手順は以下のとおりです。

1. **[プロセス]** ページ (**[システム処理]**→**[プロセス]**) が表示されます。
2. 該当するプロセスの行で **[詳細]** をクリックします(このオプションは、ユーザにプロセスを管理する権限がある場合にのみ表示されます)。ポータルに、選択したプロセスの **[プロセスの詳細]** ページが表示されます。
3. オプション・バーで **[変数]** をクリックします。

[SQL table & statement info] チェック・ボックスにチェックを付けた場合、このテーブルにはクエリ・キャッシュ (ある場合) を直接表示できる列が含まれます。

19.4 ターミナルへのメッセージの送信

選択したプロセス、あるいはすべてのプロセスに関連付けられているターミナルにメッセージを送信できます。このユーティリティは、例えばユーザにシステムからのサイン・オフを依頼する場合などに便利です。しかし、メッセージがその時点で印刷中のレポートの中央に表示されることがあるため、使用には注意を要します。

このユーティリティは、メッセージを送信する際に各ターミナルの制御を一時的に占有します。メッセージの受信後、ターミナルは継続してプロセスを実行します。メッセージはターミナル画面に表示されます。したがって画面の表示が中断されることがありますが、ユーザの入力には影響しません。メッセージは、実行中の ObjectScript ユーティリティ・ウィンドウには表示されません。

選択したプロセスに関連するターミナルにメッセージを同報通信するには、管理ポータルで以下のように実行します。

1. **[プロセス]** ページ (**[システム処理]**→**[プロセス]**) が表示されます。
2. **[ブロードキャスト]** (操作バー上) をクリックし、**[ブロードキャスト]** ウィンドウを開きます。
3. テキスト・ボックスに同報通信するメッセージを入力します
(メッセージを受信するアクティブなプロセスがない場合は、それを示すダイアログ・ボックスが表示されます。メッセージ・テキスト・ボックスやプロセスのリストは表示されません。**[閉じる]** をクリックします)。
4. ブロードキャスト・メッセージを受け取るプロセス (PID) のチェック・ボックスにチェックを付けます。必要に応じて、**[全て選択]** または **[すべて解除]** を使用すると便利です。
5. **[ブロードキャスト]** をクリックします。
6. 完成したメッセージが表示されたら、**[閉じる]** をクリックします。

20

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。

20.1 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 – 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 – 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** – Native API、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** – 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** – コールイン・インタフェース (リソース・ベース)
- ・ **%Service_ComPort** – システムに接続する COMM ポート (リソース・ベース)
- ・ **%Service_Console** – 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** – DataCheck ユーティリティ (基本)
- ・ **%Service_DocDB** – 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** – 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。

20.1.1 個々のサービスに関するメモ

20.1.1.1 %Service_Bindings

%Service_Bindings サービスの場合、アクセスを管理するリソースのカテゴリは、Native API、オブジェクト、および SQL の 3 つです。Native API カテゴリには、ユーザが認証後に Native API を使用してデータにアクセスできるかどうかを制御する 4 つのリソースがあります。それらのリソースは、**%Native_ClassExecution** (ClassMethodValue()) などのクラス実行関数へのアクセスを制御)、**%Native_Concurrency** (Lock()) などの並行処理関数へのアクセスを制御)、**%Native_GlobalAccess** (Kill()) などのグローバル・アクセス関数へのアクセスを制御)、**%Native_Transaction** (TCommit()) などのトランザクション関数へのアクセスを制御) です。**%Developer** ロールおよび **%Manager** ロールには、**%Native_ClassExecution** および **%Native_GlobalAccess** が含まれ、既定で Use 権限が割り当てられます。

注釈 InterSystems IRIS では、Native API を使用してルーチン実行にアクセスすることは禁止されています。Native API を介してルーチンを使用する必要がある場合は、代わりにクラス実行を利用することをお勧めします。

また、InterSystems IRIS には、**%Service_Object** リソースおよび **%Service_SQL** リソースが用意されています。これらのリソースは、ユーザがデータにオブジェクトとしてアクセスできるかどうか、および SQL を使用してアクセスできるかどうかをそれぞれ制御します (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。

20.1.1.2 %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。

20.1.1.3 %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。

20.1.1.4 %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。

20.1.1.5 %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。

20.1.1.6 %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。

20.1.1.7 %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。

20.1.1.8 %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.cwx` 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、管理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の、**%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。

20.2 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 – 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 – 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。

20.3 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 20-1: 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 – Kerberos キャッシュ
- ・ KRB ログイン – Kerberos ログイン
- ・ Del – 代行認証
- ・ LDAP – LDAP 認証
- ・ OS – オペレーティング・システム・ベースの認証
- ・ IA – インスタンス認証
- ・ Un – 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。

20.4 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。

21

タスク・マネージャの使用

タスク・マネージャにアクセスするには、[システムオペレーション]→[タスク・マネージャ]に移動します。

ターミナルを使用してタスク・マネージャを操作することもできます（“[TASKMGR の使用](#)”を参照）。タスクを管理するその他の方法については、“タスク”を参照してください。

注釈 タスク・マネージャは 60 秒ごとにポーリングして、実行するタスクがあるかどうかを確認します。**[すぐに実行する]**をクリックしてタスクをスケジュールすると、新しくスケジュールされたタスクが実際に実行されるまで最大で 60 秒の遅延が発生する場合があります。

21.1 新規タスク

[新規タスク] オプションを選択すると、**[タスクスケジューラウィザード]** が起動します。このツールでは、実行する新規タスクをスケジュールできます。

注釈 **[タスクスケジューラウィザード]** を使用し、**[タスクスケジュール]** ページから既存のタスクを選択して編集することもできます。

1. **[タスクスケジューラウィザード]** の最初のページには、新規タスクに関する以下の詳細情報が示されます。

- ・ **タスク名**
- ・ **説明**
- ・ **[このタスクを実行するネームスペース]**—タスクを実行する定義済みのネームスペースのリストから選択します。
- ・ **[タスクタイプ]**—リストされたタスクから選択します。使用可能なタスクの詳細は、以下の“**タスク・タイプ**”を参照してください。

注釈 選択したタスクのタイプによっては、追加情報の指定に使用するフォームが表示される場合があります。例えば、IntegrityCheck をスケジュールしているときは、**[ディレクトリ]**、**[ファイル名]**、**[KeepDays]**（ファイルを保持する日数）の入力を求めるフォームが表示されます。

2. **[タスクの優先度]**—**[優先度通常]**、**[優先度低]**、または**[優先度高]**から選択します。優先度の詳細は、“**優先度**”を参照してください。
3. **[このユーザでタスクを実行]**—定義済みのユーザのリストから選択します。現在ログイン中のユーザとは別のユーザを選択するには、**%Admin_Secure:Use** 特権を持つ必要があります。

注釈 選択したユーザが無効な場合、タスクはそのユーザが有効になるまで一時停止され、手動で再開されます。これは組み込みのシステム・タスクには適用されません。組み込みのタスクは選択したユーザが無効でも実行されます。

各タスクは、選択された InterSystems IRIS® データ・プラットフォーム・ユーザ名に基づくライセンス・ユーザ ID でライセンス数を 1 つ消費します。詳細は、“[ライセンス・ログインの特別な考慮事項](#)” を参照してください。

4. **[タスク実行時に出力ファイルを開く]** および **[出力ファイル]** – タスクによって出力が生成された場合は、指定したディレクトリにログ・ファイルが書き込まれます。
5. **[エラー時にタスクを一時停止]** – エラー後にタスクを再スケジュールして実行を継続するか、タスクを一時停止するかを指定します。既定は **[いいえ]** です。エラー後にタスクを再スケジュールして実行します。
6. **[システム再起動後にタスクを再スケジュール]** – システムの再起動時（タスクの実行を予定していたときにシステムがダウンした場合）に、タスクを再スケジュールするかどうかを以下のとおり指定します。
 - ・ **[いいえ]** はシステムの再起動時にタスクを実行するように指定します。
 - ・ **[はい]** はシステムの再起動後、タスクを次の論理時刻に実行するようにスケジュールの再設定を指定します。
7. **[完了通知の送信先メールアドレス]** – タスク・マネージャの電子メール設定を構成済みの場合（“[タスク・マネージャの電子メール設定の構成](#)” を参照）、タスクが正常に終了した通知の送信先とする電子メール・アドレスのコンマ区切りリストを入力します。
8. **[エラー通知の送信先メールアドレス]** – タスク・マネージャの電子メール設定を構成済みの場合、タスクが失敗した通知の送信先とする電子メール・アドレスのコンマ区切りリストを入力します。
9. **[ミラーにおけるタスクの実行方法]** – このインスタンスがミラー・メンバの場合、タスクを実行できるメンバのタイプをします。
 - ・ プライマリ・フェイルオーバー・メンバのみ
 - ・ バックアップ・フェイルオーバー・メンバおよび非同期メンバのみ（プライマリを除くすべて）
 - ・ すべてのミラー・メンバ（プライマリ、バックアップ、および非同期）

重要 ミラー・メンバで実行するタスクでは、このオプションをミラー・メンバのすべてのタスクに設定する必要があります。ミラー・メンバにインスタンスを追加することによって、ユーザ定義タスクでこのオプションが自動的に更新されることはありません。このため、以下のいずれかを行う必要があります。

- ・ インスタンスがミラーではない場合でもタスクの作成時にこの設定を定義し、インスタンスがミラーに追加される場合にそのタスクを実行できるようにします。
- ・ インスタンスがミラーに追加される場合は、必ずすべてのユーザ定義タスクを確認し、**[ミラーにおけるタスクの実行方法]** を設定します。

注釈 バックアップがプライマリとしての引き継ぎ中など、フェイルオーバー・メンバの状態が移行中の場合は、タスク・マネージャはプライマリまたはバックアップとして状態が確立されるまでそのメンバでタスクを実行しません。

すべての必須フィールドに値を入力したら、**[次へ]** をクリックします。

10. **[タスクスケジューラウィザード]** の 2 ページ目で、新規タスクをいつ実行するかを指定します。オプションは、以下のとおりです。
 - ・ **[日次]** – 日次間隔（1 日おき、2 日おきなど）。
 - ・ **[週次]** – 週次間隔（特定の曜日に）。

- ・ [月次] – 月次間隔 (月の特定の日に)。

注釈 選択した日が存在しない月の場合は、その月でその日に最も近い日が使用されます。

- ・ [月次 (曜日指定)] – 月次間隔 (指定された曜日に) (第 1 月曜日または第 3 水曜日など)。
- ・ [他のタスク完了後] – 指定したタスクの実行後。
- ・ [オンデマンド] – 手動実行時のみ。

[日次]、[週次]、[月次]、および [月次 (曜日指定)] フィールドでは、1 日の中でタスクを実行する時刻を指定できます。

- ・ [開始日] – スケジュールされたタスクを実行する最初の日。
- ・ [終了日] – スケジュールされたタスクを実行する最後の日。
- ・ [タスク実行詳細] – タスクが実行されるとき動作を、次のように指定します。
 - [この時刻に一度実行] – 指定された時刻に 1 回タスクを実行します。
 - [次の間隔で実行 ____] – 指定した時間範囲の間、指定した間隔でタスクを繰り返し実行します。

注釈 何らかの理由でスケジュールされた時刻に実行し損ねた場合、既定では、すべての新規タスクが期限切れになります。この動作を変更するには、以下の [「タスク期限切れ動作の変更」](#) を参照してください。

すべての必須フィールドに値を入力したら、[完了] をクリックしてタスクをスケジュールします。

21.2 タスク・タイプ

このセクションでは、タスクの作成または編集時に選択できる、事前定義のタスク・タイプについて説明します。一部のタスクは、特定のネームスペースからのみ使用できます。

InterSystems IRIS には、既定で実行されるさまざまなタスクが含まれています。以下の表に、これらの既定値を示します。これらは、[\[タスクスケジュール\]](#) ページで編集できます。

注釈 %SYS.Task.Definition API を使用すると自分のタスク・タイプを定義できます。詳細は、「インターシステムズ・クラス・リファレンス」のクラス・ドキュメントを参照してください。

タスク・タイプ	対応する既定のタスク	説明
CheckLogging	[ログギングアクティビティチェック] – 夜間に実行されます。2 日後にアラートを送信します。	ログギング・メカニズム (^ISCSOAP など) が意図せずに実行されたままになっていないことを確認します。DaysForAlert で指定した日数後にアラートを作成し、DaysForShutoff で指定した日数後にログを無効にします。
CumuIncrDBList	なし	定義済みリストのデータベースの累積バックアップを実行します。
DiagnosticReport	[診断レポート] – オンデマンドで実行されます。	診断レポートを収集し、WRC へ配信します。

タスク・タイプ	対応する既定のタスク	説明
FullAllDatabases	なし	すべてのデータベースのフル・バックアップを実行します。
FullDBList	なし	定義済みリストのデータベースのフル・バックアップを実行します。
IncrementalDBList	なし	定義済みリストのデータベースのインクリメンタル・バックアップを実行します。
IntegrityCheck	[整合性チェック] – 既定では無効になっています。	ネームスペースのデータベースの整合性チェックを実行します。整合性チェック・タスクは、%SYS ネームスペースが選択されている場合にのみ表示されます。
InventoryScan	[インベントリスキャン] – オンデマンドで実行されます。	インスタンスのインベントリ・スキャンをコンパイルします。
PurgeAudit	[監査データベースのパージ] – [ジャーナル切り替え] タスク後に実行されます。	ジャーナル切り替えの指定時間後に監査データベースを削除します。
PurgeBackupLog	[バックアップログの削除] – 夜間に実行されます。	データベース・バックアップから指定時間が経過した後にバックアップ・ログを削除します。
PurgeErrorsAndLogs	[エラーとログファイルを削除] – 夜間に実行されます。	<p>iris.cpf ファイルの設定に基づいて、エラー・グローバルと messages.log が削除されます (“構成パラメータ・ファイル・リファレンス” の “[Startup]” セクションで “ErrorPurge” と “MaxConsoleLogSize” のパラメータを参照してください)。</p> <p>注釈 システム・モニタの実行時には、メッセージ・ログのサイズの監視および制限を行います。</p>
PurgeJournal	[ジャーナルファイル削除] – 夜間に実行されます。	削除条件と一致するジャーナル・ファイルを削除します。
PurgeTaskHistory	[タスク削除] – 夜間に実行されます。	タスク・マネージャの履歴ファイルを削除します。

タスク・タイプ	対応する既定のタスク	説明
RunLegacyTask	なし	<p>レガシー・タスクを実行します (独自のプログラム・タスクなど)。テキスト・ボックスに入力する行 (do ^MyCode など) は、ターミナルで実行可能である必要があります。</p> <p>注釈 RunLegacyTask コードでローカル変数の Status を使用しないでください。InterSystems IRIS はこの変数を使用します。この変数が RunLegacyTask コードによって変更されている場合、タスクはエラー・ステータスで終了する可能性があります。</p> <p>エラーのため完了できないレガシー・タスクには、[エラーのために一時停止しています] というマークが付けられます。このマークが付いたタスクは、エラーが修正されてユーザがタスクを再開するまで実行スケジュールに入れられません。</p>
SecurityScan	[セキュリティスキャン] – 夜間に実行されます。	失効したユーザ・アカウントを無効化し、ユーザ・パスワードと Web セッション・トークンの失効を設定します。
SwitchJournal	[ジャーナル切り替え] – 夜間に実行されます。	新しいジャーナル・ファイルに切り替えます。

21.3 オンデマンドタスク

[オンデマンドタスク] ページ ([システム処理]→[タスク・マネージャ]→[オンデマンドタスク]) には、必要に応じて実行するようにスケジュールされたタスクが一覧表示されます。このリストには、タスク名、説明、およびタスクをこのページから [実行] するオプションが含まれています。列見出しをクリックすることにより、テーブルの情報を並べ替えることができます。[実行] をクリックすると、[タスクの実行ウィザード] ページにタスク名と ID、およびタスクを実行する日付と時刻が表示されます。情報を確認し、タスクをスケジュールするには、[すぐに実行する] をクリックします。

21.4 実行予定タスク

[実行予定タスク] ページ ([システムオペレーション]→[タスクマネージャ]→[実行予定タスク]) には、特定の間隔内に実行される予定のタスクが一覧表示されます。間隔を選択するには、タスク・リストの左にある [実行予定:] 検索ペイン内

のオプションをクリックします。[日付] オプションを選択した場合、日付は yyyy-mm-dd 形式で入力するか、またはカレンダー・アイコンをクリックしてカレンダーから日付を選択することができます。

列見出しをクリックすることにより、タスク・リストの情報を並べ替えることができます。各タスクのスケジュールは、[一時停止] または [再開] の各オプションを使用して、一時停止または再開できます。

- ・ [一時停止] – タスクを一時停止します。[タスク実行が一時停止された場合にタスクを再スケジュールしますか?] ドロップダウン・リストを使用して、以下のいずれかを指定します。
 - [いいえ]。タスクの再開の際、タスク・マネージャはそのタスクの見つからないインスタンスを実行してから通常のスケジュールに戻るを試みます。
 - [はい]。タスクの再開の際、タスク・マネージャは見つからないインスタンスの実行を試みることなく通常のスケジュールに戻ります。
- ・ [再開] – 一時停止されたタスクを再開できます。

21.5 タスクスケジュール

[タスクスケジュール] ページ ([システム処理]→[タスク・マネージャ]→[タスクスケジュール]) には、予定されているすべてのタスクが一覧表示されます。列見出しをクリックすることにより、テーブルの情報を並べ替えることができます。各オプションを使用して、[詳細] や [履歴] を表示したり、スケジュールされたタスクを [実行] できます。

- ・ [タスク名] – タスクに関する[詳細](#)を表示して、操作を実行できます。
- ・ [履歴] – タスクの[履歴](#)を表示します。
- ・ [実行] – タスクを実行できます。[タスクの実行] ウィザードには、タスク名と ID、およびタスクを実行する日付と時刻が表示されます。情報を確認し、タスクをスケジュールするには、[すぐに実行する] をクリックします。

21.5.1 タスクの詳細

予定されているタスクの詳細を表示し、操作のいずれかを実行するには、[タスク名] 列のタスク名をクリックします。[タスク詳細] ページには、選択したタスクの情報と実行詳細が表示されます。以下の適切なボタンをクリックして、タスクで以下の操作のいずれかを実行できます。

- ・ [編集] – [タスクスケジューラウィザード] を使用してタスク定義とスケジュールを変更します。
- ・ [履歴] – タスクの[履歴](#)を表示します。
- ・ [再開]/[一時停止] – [実行予定タスク] ページと同様に、タスクを一時停止したり、一時停止されているタスクを再開します。
- ・ [削除] – タスクを永久に削除します。

注釈 システム・タスクは削除できません。

- ・ [エクスポート] – タスクをファイルにエクスポートします。その後、別の InterSystems IRIS インスタンスなどにより、そのファイルを[インポート](#)できます。
- ・ [実行] – タスクをスケジュールします。[実行] をクリックすると、[タスクの実行ウィザード] ページにタスク名と ID、およびタスクを実行する日付と時刻が表示されます。情報を確認し、タスクをスケジュールするには、[すぐに実行する] をクリックします。

注釈 ここで説明したアクションには、タスクの実行中には使用できないものもあります。

21.5.2 スケジュールされたタスクの履歴

スケジュールされた個別のタスクの履歴情報を表示するには、項目の行にある **[履歴]** リンクをクリックします。**[タスク履歴]** ページには、選択したタスクの詳細履歴が表示されます。**[結果]** 列は、最後にタスクが実行されたときの結果を **[成功]** またはエラー・メッセージで示します。列見出しをクリックすることにより、テーブルの情報を並べ替えることができます。

ページ上部の **[詳細]** リンクには、選択したタスクに対応する **[タスク詳細]** ページが表示されます。

21.6 タスク履歴

[タスク履歴] ページ (**[システム処理]**→**[タスク・マネージャ]**→**[タスク履歴]**) には、タスク・マネージャによって実行されたすべてのタスクの履歴が表示されます。

列見出しをクリックすることにより、テーブルの情報を並べ替えることができます。システム・タスクをフィルタで除外するには、テーブルの上部にある **[ユーザ定義のタスクのみ表示]** チェック・ボックスにチェックを付けます。

21.7 タスクのインポート

[タスクのインポート] ページ (**[システム処理]**→**[タスク・マネージャ]**→**[タスクのインポート]**) により、タスクのインポートと実行ができます。そのためには、以前にエクスポートしたタスク・ファイルを参照してから、**[すぐに実行する]** をクリックします。タスクをファイルにエクスポートする方法の詳細は、このセクションの **"タスクの詳細"** を参照してください。

注釈 同じバージョンの InterSystems IRIS を実行しているタスク間でのみタスクをインポートでき、またエクスポートできます。

21.8 ^TASKMGR の使用

^TASKMGR ルーチンでは、ターミナルを使用してタスク・マネージャを構成できます。特に明記されていない限り、^TASKMGR と管理ポータルには、タスクを構成するための同じオプションが含まれています。どちらを使用するかは、好みの問題です。

^TASKMGR を使用するには、以下の手順を実行します。

1. ターミナルを開きます。
2. `set $namespace = "%SYS"` と入力して、%SYS ネームスペースに変更します。
3. `do ^TASKMGR` と入力します。

プログラムでタスクをスケジュールまたは編集する方法の詳細は、**"インターシステムズ・クラス・リファレンス"** の **%SYS.Task** クラスのドキュメントを参照してください。

21.8.1 タスク期限切れ動作の変更

既定では、すべてのタスクは、スケジュールされた時刻に実行し損ねた場合、期限切れになるように構成されます。これは、スケジュールされた時間内に InterSystems IRIS がダウンした場合や、前のタスクの実行が次のスケジュール時間ま

で延長された場合など、さまざまな理由によって発生します。タスクは、期限切れになると、次のスケジュール時間まで実行されません。

^TASKMGR ルーチンには、スケジュールされた実行がスキップされないように、この動作を変更するオプションがあります (管理ポータルにはありません)。そのためには、以下の操作を実行します。

1. ターミナルを開きます。
2. `set $namespace = "%SYS"` と入力して、%SYS ネームスペースに変更します。
3. `do ^TASKMGR` と入力します。
4. オプション 2 を選択します。
5. 編集するタスク番号を入力します。
6. Task Expires? に到達するまで、Enter キーを押して各オプションの既定値を保持します。
7. No と入力します。

または、タスクを実行し損ねた後、タスクが直ちに期限切れにならないように、猶予期間を指定することもできます。代わりに、タスク・マネージャが、実行し損ねたタスクをできる限り速やかに実行します。特定のタスクの有効期限が切れる、この猶予期間を調整するには、以下の操作を実行します。

1. ターミナルを開きます。
2. `set $namespace = "%SYS"` と入力して、%SYS ネームスペースに変更します。
3. `do ^TASKMGR` と入力します。
4. オプション 2 を選択します。
5. 編集するタスク番号を入力します。
6. Task Expires? に到達するまで、Enter キーを押して各オプションの既定値を保持します。
7. Yes と入力します。
8. Expires in how many days? というプロンプトが表示されたら、タスクの期限が切れるまでの日数を入力します。
9. Expires in how many hours? というプロンプトが表示されたら、タスクの期限が切れるまでの時間数を入力します。
10. Expires in how many minutes? というプロンプトが表示されたら、タスクの期限が切れるまでの分数を入力します。

21.9 カスタム・タスク

カスタム・タスクを定義するには、%SYS.Task.Definition のサブクラスを定義します。このサブクラスに、必要な処理を実行して %Status を返す OnTask() メソッドを実装します。

このタスクを実行するための特定の特権を必要とすることができます。そのためには、このクラスに RESOURCE クラス・パラメータを指定します。値は、resource:permission の形式で指定します。以下に例を示します。

Class Member

```
Parameter RESOURCE As String = "%Admin_Manage:USE";
```

追加情報は、クラス・リファレンスを参照してください。

このクラスは、コンパイルすると、このページの説明に従ってタスク・マネージャのスケジュールに追加できます。そこでは、ユーザ・タスクとして (システム・タスクではなく) リストされます。

22

[バックグラウンド・タスク] ページの使用

バックグラウンド・タスクとは、そのタスクを作成したプロセスから独立して、バックグラウンドで実行する非同期のジョブ・プロセスのことです。バックグラウンド・タスクは、ユーザが ObjectScript [JOB コマンド](#)を発行したときに作成されるか、完了までユーザが待機する必要のないジョブを管理ポータルまたはユーティリティで実行することで作成されます。例えば、“[データベースの領域の削除](#)”の手順に従い、管理ポータルを使用してデータベースの領域を削除すると、バックグラウンド・タスクが開始されます。

[バックグラウンド・タスク] ページ ([システムオペレーション]→[バックグラウンド・タスク]) には、過去のバックグラウンド・タスクとアクティブなバックグラウンド・タスクが表示されます。過去のバックグラウンド・タスクについてのログは、いつでも削除できます。

バックグラウンド・タスクがアクティブのときは、[プロセス] ページ ([システムオペレーション]→[プロセス]) でバックグラウンドのプロセスを確認することもできます (“[InterSystems IRIS プロセスの管理](#)”を参照)。

23

InterSystems IRIS ライセンスの管理

ここでは、InterSystems IRIS® データ・プラットフォームのライセンス・システムの概要を示し、ライセンスを管理および監視する方法を説明します。

- 重要** ライセンス付与された InterSystems IRIS ソフトウェアの使用法は、“使用条件”によって定められています。ただし、その実装に適用される条件は緩和される場合があります。お客様が記述されたライセンス関連のコードが、これら使用条件に適合することを必ずご確認ください。
- ライセンス・サーバを使用する場合、考慮すべき重要な[ネットワークの検討事項](#)があります。

23.1 ライセンス管理の概要

各 InterSystems IRIS インスタンスには、そのライセンスの機能範囲および現在の使用状況の独立したローカル・ビューが保持されており、各インスタンスには、ライセンス・キーへのアクセス権が必要です(評価版インストールを除く)。これらのライセンスを管理する方法は、使用しているインスタンスの数に応じて、いくつか用意されています。

- 各インスタンスにローカル・ライセンス・キー・ファイルをインストールしてこれをアクティブにできます。
- 一元的な場所に格納されているキー・ファイルを管理するようライセンス・サーバを構成できます。そうすれば、このライセンス・サーバから他のインスタンスにキー・ファイルを配布できます。この場合、各インスタンスをキー・ファイルの **LicenseID** で構成する必要があります。これにより、各インスタンスは起動時にキーのコピーを取得できます。

ライセンス・サーバによって、すべてのインスタンスでローカルに保持されているライセンス使用のビューが調整されます。ライセンス・サーバは InterSystems IRIS プロセスではないので、InterSystems IRIS インスタンスが終了しても影響はありません。1 つのライセンス・サーバで、複数のインスタンスを処理できます。したがって、ホスト上で実行される InterSystems IRIS インスタンスの数にかかわらず、各ホストに必要なライセンス・サーバは 1 つのみです。ただし、各 InterSystems IRIS インスタンスのローカルには、認証ライセンス・キー・ファイルのコピーが必要です。また、ライセンス・サーバは、実行中のインスタンスによって起動されるので、InterSystems IRIS インスタンスが常に稼働しているシステムで実行するように構成する必要があります。

ライセンス・サーバは、InterSystems IRIS インスタンスが実行されているコンピュータであればどのコンピュータ上でも実行できます。監視プロセス(ライセンス・モニタ)がライセンス・サーバに更新メッセージを送信し、複数のインスタンスがライセンスを共有している場合は、ライセンス・サーバによってライセンス割り当てが調整されます。

- 冗長性を確保するために、複数のライセンス・サーバ(通常は異なるホスト上)を構成できます。ライセンス・ソフトウェアによって、ライセンス・サーバの 1 つがアクティブ・サーバとして選択されます。その他のサーバは、アクティブ・サーバに障害が発生したときに処理を引き継ぎます。ライセンス・サーバの構成時に、ライセンス・サーバをホストするために使用するマシンを決定します。実行するホストの台数は任意ですが、4 台以上では多すぎます。

マルチサーバ・ライセンスは、同一マシンまたは異なるマシン上で協力関係にあるインスタンス間で共有できます。共有は、マルチサーバ・キーでのみ許可されます。マルチサーバ・ライセンスを使用するには、InterSystems IRIS ライセンス・サーバを 1 つ以上構成し、キーで認証される InterSystems IRIS ライセンス・ユニットを割り当てる必要があります。キーを共有するインスタンスはすべて、同じ 1 つのまたは 1 組みのライセンス・サーバを使用するように構成する必要があります。

各インスタンスに、適切な `iris.key` ファイルの独自コピーがあり、同じキーで認証されるすべてのインスタンスが同じライセンス・サーバを使用している限り、ライセンス・キーが異なり異なるプラットフォーム上で実行される複数のインスタンスで同じライセンス・サーバを使用してライセンスを調整できます。ただし、複数のライセンス・キーのライセンス数が合計されることはありません。異なるライセンス・キーを使用する InterSystems IRIS インスタンスでは、ライセンス数の共有は行われません。また、異なるライセンス・キーを使用して 2 つのインスタンスにログインしたユーザは、それぞれのキーに由来するそれぞれのライセンスを使用します。

23.2 ライセンス・サーバとネットワークに関する考慮事項

ライセンス・サーバを使用する場合、以下の点に留意してください。

- 各ライセンス・モニタは、TCP ポート経由でライセンス・サーバにクエリを実行します (`%SYSTEM.License` を使用)。したがって、ライセンス・サーバへの TCP トラフィックが許可されていることを確認する必要があります。

重要 リモート・ライセンス・サーバがファイアウォールで保護されている場合、UDP トラフィックのためにライセンス・サーバ・ポートが開かれている必要があります。

- ライセンス・サーバの組み込みの冗長性の一環として、IRIS インスタンス上のライセンス・モニタは、その IRIS インスタンスがアクティブなライセンス・サーバに接続されていることを定期的に確認します。現在接続されているものと異なる IP アドレスが検出された場合、ライセンス・モニタはライセンス・サーバが変更されたと見なして、再起動モードに入り、現在のライセンス使用状況をすべて再送信するため、警告メッセージが生成されることになります。この問題が生じないように、ライセンス・サーバの場所を安定した一貫性のある IP アドレスに指定してください。

23.3 ライセンス・サーバの構成

ライセンス・サーバは、管理ポータル内の **[ライセンスサーバ]** ページ (**[システム管理]** → **[ライセンス]** → **[ライセンスサーバ]**) で追加または削除できます。このページには、このインスタンスに対して構成されたライセンス・サーバのリストが表示されます。このインスタンスに複数のライセンス・サーバが構成されている場合、アクティブなライセンス・サーバの行に影が付きます。

注釈 `$System.License.ShowServer()` メソッドを使用してどのライセンス・サーバがアクティブであるかを判断することもできます。

リストされたライセンス・サーバの名前をクリックしてその情報を更新するか、**[削除]** をクリックしてこれを削除します。新しいライセンス・サーバを追加するには、以下の手順を実行します。

- [ライセンス・サーバを作成]** をクリックし、ライセンス・サーバを構成します。画面右端に **[新しいライセンス・サーバを作成]** ボックスが表示されます。
- [名前]** ボックスに、ライセンス・サーバの名前を入力します。ライセンス・サーバ名によって構成内のライセンス・サーバが特定されるため、構成内で一意にする必要があります。
- [ホスト名/IPアドレス]** ボックスに、ライセンス・サーバをホストする IP アドレスを入力します。ドット十進法形式 (192.29.233.19) や、アルファベット形式 (mycomputer.myorg.com) で IP アドレスを入力できます。IPv6 を有効

にした場合は、コロン区切りの形式で IPv6 アドレスを入力できます
(2001:fece:ba23:cd1f:dcb1:1010:9234:4085)。

4. **[ポート]** ボックスに、ライセンス・サーバが使用するポート番号を入力します。ライセンス・サーバのポート番号は、1024 と 65535 の間の数値にする必要があります。既定のポート番号は 4002 です。冗長ライセンス・サーバが異なるホストで実行されている場合、一意のポート番号を使用する必要はありませんが、その IP アドレスで既に使用されているポート番号ではない番号を使用する必要があります。
5. (オプション) **[KeyDirectory]** ボックスに、ライセンス・キーが格納される一元的なディレクトリを入力します。詳細は、以下の [「ライセンス・サーバへのキーのロード」](#) を参照してください。
6. **[保存]** をクリックすると、ライセンス・サーバが作成されます。

ライセンス・サーバを追加または削除したら、InterSystems IRIS インスタンスを再起動する必要があります。

注釈 別個のインスタンスがすべて同じライセンス・サーバ・アドレスおよびポートを構成している場合、これらはすべて同じライセンス・サーバを使用します。このような場合、各インスタンス上の既定の **[LOCAL]** ライセンス・サーバ (127.0.0.1) を削除する必要があります。各インスタンスに同じキーがロードされると、これらのインスタンスはそのキーを共有します。インスタンスごとに異なるキーがロードされると、ライセンス・サーバではインスタンス・セットのそれぞれがキー別に処理されます。

23.4 ライセンス・サーバへのキーのロード

共有ライセンスの機能範囲の管理に加えて、ライセンス・サーバは一元的なディレクトリに格納されているキーを InterSystems IRIS インスタンスに配布することもできます。

KeyDirectory プロパティは、**Config.LicenseServers** の一部として定義されます。これを入力すると、ライセンス・サーバを起動するインスタンスは起動時にそのインスタンスで見つかった有効な ***.key** ファイルをすべて読み取り、ライセンス・サーバに送信します。各キー・ファイルには、一意の **LicenseID** プロパティが必要です。インスタンスは、正常にロードされたファイルの数とエラーをログに記録します。**%SYS** ネームスペースから **ReloadKeys`%SYS.LICENSE** を呼び出すことで、ディレクトリからキー・ファイルを手動で再ロードし、任意のライセンスを更新できます。

注釈 ライセンス・サーバのキー・テーブル内の既存のキーと同じ **LicenseID** を持つ新しいキーをロードすると、既存のキーには **replaced** というマークが付けられます。インスタンスからその **LicenseID** への要求では、最も新しくロードされたキーが取得されます。既存の **\$System.License.DumpKeys()** メソッドを使用して、ライセンス・サーバ内のキーの現在の状態を確認できます。

23.5 ライセンス・キーの有効化

InterSystems IRIS では、ライセンス・キーを使用して、登録されたサイトで適切な操作を行い、使用可能な容量を定義し、InterSystems IRIS 機能へのアクセスを制御できるようにします (評価版インストールには、ライセンス・キーは必要ありません)。ライセンス・キーは、通常、**iris.key** という名前のライセンス・キー・ファイルの形式で提供されます。

InterSystems IRIS をインストールした後に、以下の手順を使用して、ライセンス・キーを有効にします。同じ手順を使用して、インストールされているすべてのインスタンスで新しいライセンス・キーをいつでも有効にできます (キーのアップグレード)。管理ポータルからアクセスできる場所にあるライセンス・キーを有効にできます。有効化の一環として、ライセンス・キーは、インスタンスの **install-dir/mgr** ディレクトリにコピーされます。ライセンス・キーに、事前に名前が付いていない場合は、**iris.key**、という名前になります。

注釈 Windows では、インストール中にライセンス・キーを選択することもできます (“[Microsoft Windows への InterSystems IRIS のインストール](#)” を参照)。この場合、ライセンスは自動的に有効になり、ライセンス・キーは `iris.key` という名前でインスタンスの `install-dir/mgr` ディレクトリにコピーされます。ここで説明する有効化の手順は必要ありません。

また、このセクションでは、[ライセンス・トラブルシューティング](#) およびすべてのライセンス・ユニットが使用中の場合に [オペレーティング・システムのコマンド行からライセンスをアップグレード](#) する方法についても説明しています。

ライセンス・キーを有効にするには、以下の手順を使用します。

1. **[ライセンスキー]** ページ (**[システム管理]**→**[ライセンス]**→**[ライセンスキー]**) に移動します。現在のアクティブなライセンス・キーの情報が表示されます。アクティブなライセンスがない場合は、**[顧客名: ライセンス不明または読み取り不可能]** などの表記で示されます。このページにある **[印刷]** ボタンを使用すると、表示されている情報を簡単に印刷できます。
2. **[ライセンスキー有効化]** をクリックし、有効にするライセンス・キー・ファイルを参照します。ファイルを選択すると、ファイルに関する情報が表示され、目的の容量が提供されることや有効期限が適切であることなど、正しいライセンス・キーかどうかを有効にする前に検証できます。キーが有効でない場合は、エラー・メッセージにそのことが表示されます。現在アクティブなライセンスがある場合は、現在のライセンスと選択されているライセンスの情報が並んで表示されます。ライセンス・キーを適用にするために有効にした後にインスタンスを再起動する必要がある場合は、そのことが通知され、その理由が表示されます。このダイアログにある **[印刷]** ボタンを使用すると、現在アクティブなライセンスと選択した新しいライセンス・キー両方の情報を簡単に印刷できます。
3. **[アクティベート]** をクリックして、新しいライセンス・キーを有効にします。新しいキーは `iris.key` という名前で、インスタンスの `install-dir/mgr` ディレクトリにコピーされ、前のライセンス・キー (ある場合) を上書きします。確認ダイアログに、必要に応じてインスタンスを再起動することが表示され、新しいライセンスが有効にする機能が現在のライセンスよりも少ない場合は、そのことが警告されます。

Config.Startup の **LicenseID** プロパティを使用することで、ライセンス・サーバからのライセンス・キーを要求するようインスタンスを構成できます。インスタンスの起動時に、`iris.key` ファイルが存在せず、**LicenseID** が定義されている場合、インスタンスはライセンス・サーバからのライセンス・キーを要求して、これを有効にします。

注釈 同じ **LicenseID** が、ライセンス・キー・ファイルに存在する必要があり、ライセンスのダウンロードが必要なインスタンスでも定義されている必要があります。

通常は、インスタンスを再起動する必要はありませんが、ライセンス・キーをアップグレードするときには制約があります。ライセンス・タイプをパワー・ユニットから他に変更する場合、新規のキーは自動ではアクティベートされません。ただし、このような変更は極めて稀です。

ライセンスのアップグレードでは、共有メモリ・ヒープ (gmheap) 領域から消費できるメモリ量に対する制約もあります。gmheap 領域がない場合、ライセンス・テーブルのエントリ数は拡張できません。gmheap 領域にライセンス・アップグレードに使用できる十分な空間がない場合は、メッセージはメッセージ・ログに書き込まれます。**[メモリ詳細設定]** ページ (**[システム管理]**→**[構成]**→**[メモリ詳細設定]**) で、[\[gmheap\]](#) 設定のサイズを大きくすることができます。

新しいライセンス・キーが既存のライセンス・キーよりも、少なくとも 1000 個の 64 KB ページ分多くの gmheap 領域を消費する場合は、新しいライセンス・キーを完全にアクティブ化するために、InterSystems IRIS インスタンスを再起動する必要があります。各ページが少なくとも 227 個のライセンスを表すため、この状況が発生することは、まずありません。

23.6 ライセンス・キーの更新

ライセンス・キーを更新するには、KeyDirectory 内のキー・ファイルを置き換えて、`ReloadKeys ^%SYS.LICENSE` を実行します。各インスタンスのライセンス・モニタ (`LMFMON`) が、構成されている **LicenseID** に対して異なるキーがあるかどうかを 30 分おきに確認し、ある場合はアップグレードを実行しようとします。

注釈 実動インスタンスではほとんどのアップグレードが成功しますが、状況によっては、インスタンスの再起動が必要な場合があります。このような場合、ライセンス・モニタはエラーをログに記録し、次の日までそのキーのアップグレードを再度試行することはありません（繰り返されるエラーがログに記録されないようにするため）。インスタンスの再起動によって、起動時に新しいキーがロードされます。

23.7 ライセンス・トラブルシューティング

ライセンスを入力して InterSystems IRIS を再起動した後、1 人のユーザしかログインできない場合は、管理ポータルを使用して調査してください。[ライセンス使用量] ページ ([システムオペレーション]→[ライセンス使用量]) の [プロセス毎の使用] を選択すると、実行中のプロセスの数が表示されます。ライセンス情報の表示には、ポータルの [ライセンスキー] ページ ([システム管理]→[ライセンス]→[ライセンスキー]) を使用することもできます。“[ライセンス・キーの有効化](#)”を参照してください。キーが無効の場合、[CustomerName] フィールドにその説明が表示されます。

また、ライセンス・エラー・メッセージは、メッセージ・ログおよびシステム・モニタ・ログで確認することもできます。これらのログは、それぞれポータルの [メッセージログ] ページ ([システムオペレーション]→[システムログ]→[メッセージログ])、および [システム・モニタのログ] ページ ([システムオペレーション]→[システムログ]→[システム・モニタのログ]) で表示できます (“[ログ・ファイルの監視](#)”を参照)。システム・モニタは、ライセンスの有効期限の警告とアラートをこれらのログに書き込み、ヘルス・モニタは、ライセンス取得のアラートと警告を書き込みます。ライセンス制限が超過されると、ライセンス・モジュールごとのメッセージ・ログに警告が書き込まれます。アプリケーション・モニタでは、ライセンス・メトリックベースのアラートを構成して、電子メール通知を送信したり、通知メソッドを呼び出したりできます。詳細は、“[システム・モニタの使用](#)”を参照してください。

ライセンスの有効期限が近付くと、メッセージ・ログに警告が表示されます。ライセンスは、有効期限が切れる日付の終了時刻まで有効です。例えば、ライセンスの有効期限が 2022 年 11 月 30 日である場合、そのライセンスは 2022 年 11 月 30 日の終了時刻まで有効ですが、2022 年 12 月 1 日には機能しなくなります。

\$System.License.Help では、ライセンス問題のトラブルシューティングに使用できる方法のリストが表示されます。

ObjectScript

```
Do $System.License.Help()
```

このドキュメントでは、これらのメソッドの多くについて説明しています。

23.7.1 管理者ターミナル・セッション

いくつかの問題によって、ターミナル・セッションを取得できないことがあります。これは、InterSystems IRIS を適切に起動できず、シングル・ユーザ・モードになった場合、または単に使用可能なライセンスがない場合に発生することがあります。このような場合、管理者ターミナル・セッションを作成する必要があることがあります。管理者ターミナル・セッションでは、特別なライセンスを使用して問題を解決できます。

管理者セッションを開くためのコマンドは、[Windows](#) と [UNIX®](#)、[Linux](#)、[MacOS](#) では異なります。

23.7.1.1 Windows での管理者セッション

コマンド・プロンプトを使用して、install-dir¥bin に移動します。次に、以下のコマンドを管理者として実行します。

```
irisdb -s<install-dir>\mgr -B
```

これにより、InterSystems IRIS インストール環境の bin ディレクトリ (install-dir¥bin) から InterSystems IRIS の実行可能ファイルが実行され、install-dir¥mgr のパス名が示されます (-s 引数の使用により)。また、緊急のログイン 1 件を除き、すべてのログインが禁止されます (-B 引数の使用により)。

例として、既定のディレクトリにある MyIRIS という名前のインスタンスでは、コマンドは次のようになります。

```
c:\InterSystems\MyIRIS\bin>irisdb -sc:\InterSystems\MyIRIS\mgr -B
```

23.7.1.2 UNIX®, Linux、および MacOS での管理者セッション

コマンド・プロンプトを使用して、install-dir/bin ディレクトリに移動します。次に、以下のコマンドを実行します。

```
iris terminal <instance-name> -B
```

例として、既定のディレクトリにインストールされている MyIRIS という名前のインスタンスでは、コマンドは次のようになります。

```
User:/InterSystems/MyIRIS/bin$ iris terminal MyIRIS -B
```

23.8 オペレーティング・システムのコマンド行からのライセンスのアップグレード

%SYSTEM.License.Upgrade() メソッドにより、install-dir\mgr ディレクトリにコピーされた新しいライセンス・キーがアクティブ化されます。すべてのライセンス・ユニットがユーザによって消費され、ターミナル・ウィンドウを開くことができない場合、以下のようにして、このメソッドをコマンド行から実行して容量の大きい新しいライセンス・キーをアクティブ化できます。

```
iris terminal <instancename> -U %SYS '##Class(%SYSTEM.License).Upgrade()'
```

iris コマンドの詳細は、“[InterSystems IRIS インスタンスの接続](#)”を参照してください。

23.9 ライセンスの機能範囲と使用の決定

現在使用されているライセンス数と使用者はどのようにして確認できるのでしょうか。%SYSTEM.License クラスが、InterSystems IRIS ライセンス・アプリケーション・プログラミング・インタフェース (API) に対するインタフェースを提供します。これの多数のメソッドや関連クエリを使用してライセンスの機能範囲および現在の使用状況を問い合わせることができます。

%Library.%ResultSet クラスの RunQuery メソッドを使用して、多数のライセンス・クエリを実行できます。例を以下に示します。

ObjectScript

```
do ##class(%ResultSet).RunQuery("%SYSTEM.License", "Summary")
```

また、以下の表に示すように、管理ポータルの [ライセンス使用量] ページ ([システムオペレーション]→[ライセンス使用量]) で、これらのクエリの出力を表示できます。

[ライセンス使用量] ページのリンク	ライセンス・クエリ
要約	Summary() – \$System.License.ShowSummary によって示されるライセンスの使用状況の要約を返します。
プロセスごとの使用	ProcessList() – \$System.License.DumpLocalPID によって示されるオペレーティング・システム・プロセス ID (PID) 別にライセンスの使用状況を返します。
ユーザごとの使用	UserList() – ユーザ ID 別にライセンスの使用状況を返します。
配布ライセンス使用状況	AllKeyConnectionList() – ユーザ別に並べ替えて配布ライセンスの現在の使用状況を返します。(ライセンス・サーバが接続されていない場合、これは無効になります。)

%SYSTEM.License クラスから以下のクラス・メソッドを使用して情報を表示したり、ライセンス・データベースをファイルにダンプしたりすることもできます。

\$System.License.CKEY は、キーを表示します。このサブルーチンは、互換性のために保持されている `CKEY` プログラムによって呼び出されます。

ObjectScript

```
Do $System.License.CKEY()
```

\$System.License.ShowCounts は、ローカル・システムの共有メモリに記録されているライセンスの使用状況の要約を示します。

ObjectScript

```
Do $System.License.ShowCounts()
```

\$System.License.ShowServer は、アクティブなライセンス・サーバのアドレスとポートを示します。

ObjectScript

```
Do $System.License.ShowServer()
```

REST ベースのアプリケーションを開発した場合、ライセンスは使用によって消費されます。この事態が発生しないようにするには、確立可能な Web ゲートウェイ接続の数を構成します。管理ポータル の [ウェブゲートウェイ管理] セクションで、以下の操作を実行します。

1. **[サーバ接続]** に移動します。
2. **[状態なしパラメータ]** を選択します。
3. サーバ側ログインに許容されたライセンスよりも 2 または 3 少ないライセンス数を **[最大]** に設定します。

注釈 アプリケーションのサーバ側のニーズによっては、この設定の調整が必要になることがあります。

これを実行することで、利用可能なすべての接続がビジーになり、新規の要求は拒否されずにキューに入れられます。ライセンス数の超過による拒否は現れなくなります。ボリュームの増加に伴って、クライアントの応答時間は遅くなります。そのような状況は、追加ライセンスの購入が必要なことを示しています。

以下のセクションでは、ライセンス情報を表示する他のいくつかのメソッドについて説明します。

- ・ [ローカル・ライセンス情報を表示するメソッド](#)
- ・ [ライセンス・サーバ情報を表示するメソッド](#)

23.10 ローカル・ライセンス情報を表示するメソッド

以下のサブルーチンは、ローカルのインスタンス共有メモリに格納されているライセンス・テーブルのコンテンツをダンプします。通常は、クライアントを識別します。

\$System.License.DumpLocalAll は、現在のディレクトリの **all.dmp** ファイルにすべてのローカル・ライセンス・テーブル・エントリをダンプします。

ObjectScript

```
Do $System.License.DumpLocalAll()
```

all.dmp ファイルのコンテンツの例を以下に示します。

```
License Capacity = 5, Current use = 2, Units Remaining = 3
```

```
0) User ID = 127.0.0.1, Connections = 2, CSP Count = 0, Time active = 90
1) User ID = 192.9.202.81, Connections = 1, CSP Count = 0, Time active = 49
2) free
3) free
4) free
```

\$System.License.DumpLocalInUse は、現在のディレクトリの **inuse.dmp** ファイルに、使用中のすべてのローカル・ライセンス・テーブル・エントリをダンプします。

ObjectScript

```
Do $System.License.DumpLocalInUse()
```

inuse.dmp ファイルのコンテンツの例を以下に示します。

```
License Capacity = 5, Current use = 2, Units Remaining = 3
```

\$System.License.DumpLocalPID は、現在のディレクトリの **piduse.dmp** ファイルに、ローカル・ライセンス・テーブルの使用状況をプロセス ID 別にダンプします。

ObjectScript

```
Do $System.License.DumpLocalPID()
```

piduse.dmp ファイルのコンテンツの例を以下に示します。

PID	Process LID	Type	Con	MaxCon	CSPCon	LU	Active	Grace
592	System		0	0	0	0	0	0
2816	System		0	0	0	0	0	0
688	System		0	0	0	0	0	0

23.11 ライセンス・サーバ情報を表示するメソッド

以下のサブルーチンは、ライセンス・サーバによって保持されているライセンス・テーブルのコンテンツをダンプします。出力ファイルは、アクティブなライセンス・サーバが実行されているホスト上の指定ディレクトリにあります。

\$System.License.ShowSummary は、ライセンス・サーバでのライセンス情報の要約を表示します。Distributed license use セクションには、ライセンス・サーバによって現在サポートされているすべての InterSystems IRIS インスタンスのライセンス使用状況がまとめて表示されます。Local license use セクションには、プログラムが実行されている 1 つの InterSystems IRIS インスタンスに対するライセンス使用状況のビューが表示されます。

ObjectScript

```
Do $System.License.ShowSummary()
```

\$System.License.DumpServer は、ライセンス・サーバを実行しているホスト上のファイル **dumpserver.txt** に、このルーチンを実行したサーバに関連するライセンス・サーバ・データベース情報をダンプします。

ObjectScript

```
Do $System.License.DumpServer()
```

\$System.License.DumpServers は、ライセンス・サーバを実行しているホスト上のファイル **dumpservers.txt** に、既知のすべてのサーバに関するライセンス・サーバ・データベース情報をダンプします。

ObjectScript

```
Do $System.License.DumpServers()
```

\$System.License.DumpKey は、ライセンス・サーバを実行しているホスト上のファイル **dumpkey.txt** に、このインスタンスによって使用されるキーとそれを共有するインスタンスをダンプします。

ObjectScript

```
Do $System.License.DumpKey()
```

\$System.License.DumpKeys は、ライセンス・サーバを実行しているホスト上のファイル **dumpkeys.txt** に、すべてのキーを、それらを使用するインスタンスとクライアントを示して、ダンプします。

ObjectScript

```
Do $System.License.DumpKeys()
```

注釈 ローカル・ライセンス・メソッドによって表示された情報は、ライセンス・サーバ・メソッドによって表示された情報よりも最新であることに注意してください。これは、ライセンス・サーバは定期的に更新されるだけですが、ローカル・データはリアルタイムで更新されるためです。

ログインはローカルで制御されるため、ライセンスの数が上限値を一時的に上回ることが可能ですが、ライセンス・サーバで制限が強制されます。各インスタンスは、インスタンス共有メモリ内で維持されるローカル・ライセンス・テーブルに基づいて、ログインを許可または拒否します。各インスタンスからは、ローカル・ライセンス・テーブルに加えられた変更を説明する定期的な更新がライセンス・サーバに送信されます。すべてのインスタンスのライセンスの使用の合計が制限を超えると、ライセンス・サーバは各インスタンスからの更新メッセージに対して否定応答を送信します。

この否定応答を受け取ると、これ以上のライセンス・ユニットは使用できないため、各インスタンスでは新しいログインが拒否されます。開始しようとしている InterSystems IRIS プロセスのライセンス・ユーザ ID が、現在のいずれのプロセスのライセンス・ユーザ ID にも一致しない場合、そのログインは新しいものであると見なされます。この状態は、すべてのインスタンスによる使用の合計が許可された上限を下回るまで存続し、その時点で、ライセンス・サーバはインスタンスの更新に対して肯定応答の送信を開始します。これにより、インスタンスのそれぞれで、新しいログインが許可されます。

23.12 ユーザの認識

InterSystems IRIS ライセンス・システムは、個々のユーザを識別し、ユーザごとに 1 つのライセンス・ユニットを割り当てます。1 人のユーザは、1 つのライセンス・ユーザ ID (IP アドレス、ユーザ名、Web セッション ID、あるいはユーザの接続方法による他の識別条件) によって識別されます。

1 人のユーザによって、または 1 人のユーザに対して開始された複数のプロセスは、ユーザごとの最大プロセス数に達するまで 1 つのライセンス・ユニットを共有します。プロセス数がこの最大値を超えた場合、移行が発生し、InterSystems IRIS はそのユーザ ID に対して 1 プロセスごとに 1 ライセンス・ユニットを割り当て始めます。1 つのユーザ ID に関連付けられているプロセスの数が最大数を越える場合、システムは複数のユーザが何らかの中間媒体 (ファイアウォール・システムなど) を通じて InterSystems IRIS にアクセスしていると仮定し、追加のライセンス・ユニットを要求します。(Job コマンドで開始したプロセスは、そのコマンドを呼び出したユーザ ID の下でカウントされます。)

そのユーザ ID のプロセス数が最大値を下回るようになった場合でも、InterSystems IRIS は、そのユーザ ID に対してプロセスごとに 1 ライセンス・ユニットの割り当てを継続します。そのユーザ ID による接続がすべて閉じられ、そのユーザ ID のプロセスがなくなった場合にのみ、ライセンス割り当ては、そのユーザ ID に対して 1 ユニットになるようにリセットされます。

アプリケーションが名前ユーザを識別できる場合、クライアント IP アドレスに基づく既定のユーザ ID、Web セッション ID、および接続から派生した他のユーザ ID の使用に関連する問題が解決されます。

例えば、ファイアウォールやターミナル・サーバ・ソフトウェアを使用すると、InterSystems IRIS は接続中のユーザの区別ができず、そのために最大接続移行ルールに依存することになります。同じクライアントから混合接続を使用する場合も、自動 ID 作成を使用して適切にユーザをカウントできなくなります。

ユーザ名をライセンス識別子として使用すると、これらの問題は発生しません。組織が新たにアクセス要件や監査要件を導入するにつれて、ユーザを正確に識別する必要性は高まります。ユーザ認識を使用してライセンス規約を制御することは、時代の流れから考えても当然の結果と言えるでしょう。

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。

- ・ [ライセンス・ログイン](#)
- ・ [ユーザ名ライセンス](#)
- ・ [ライセンス・ログインの特殊なケース](#)

23.12.1 ライセンス・ログイン

ライセンス・ログインには、自動ログインと明示ログインの 2 つのモードがあります。既定は自動ログインです。ライセンス・システムは、クライアントの IP アドレスを識別して、そのアドレスをライセンス・ユーザ ID として使用します。クライアントが IP を使用してサーバに直接接続する場合、これは正確に機能します。しかし、ファイアウォールがクライアントとサーバの間にある場合は、すべてのクライアントが同じ IP アドレスを持つように見えるため、うまく機能しません。また、ターミナル・サーバが telnet プロトコルを使用している場合、InterSystems IRIS にはすべてのターミナル・サーバ・ポートが 1 つの IP アドレスとして認識されるため、自動ログインではユーザが区別できません。すべての接続が同じアドレスからのものであるため、すべての接続が同じユーザ ID を持ちます。ユーザがファイアウォールを経由して接続したり、ターミナル・サーバから telnet 転送を使用する場合は、明示ログインを使用してください。

ネットワーク転送に IP を使用していない場合は、IP アドレスをライセンス・ユーザ ID として使用できません。この場合、ライセンス・システムは、その他のさまざまなソースをユーザ ID として使用します。UNIX®/Linux システムで at デーモンによって開始されたバッチ・プロセスには、また別の問題があります。このようなプロセスは、ユーザと関連していないのでライセンス・ユニットを共有しません。このようなプロセスでは、プロセス ID がライセンス識別子として使用されます。

明示ログインが選択されているときは、InterSystems IRIS ではユーザ ID の自動検出は試行されません。アプリケーションは \$System.License.Login(UserIdentifier) メソッドを明示的に呼び出して、ライセンス・ユーザ ID を提供し、ライセンスを取得する必要があります。

明示ログインは、\$System.License.DeferUserIdentification([0 or 1]) 関数を呼び出すことで使用可能になります。この呼び出しは、インスタンスの起動時に %ZSTART ルーチンの SYSTEM エントリ・ポイントから実行できます。引数値が 1 の場合、ライセンスの取得はログイン時に延期され、明示ログインを実行できます。引数値が 0 の場合、ライセンスの取得はプロセス開始時に自動で行われます。

ログインが延期されたとき、ユーザは即時にライセンス・ログイン・メソッドを呼び出す必要があります。ライセンス・ログインを実行していないプロセスは、最初の 4000 個の ObjectScript コマンドで停止し、その後 1000 個の ObjectScript コマンドごとに停止します。

自動ログインが処理できない場合は、明示ログインを使用します。自動ログインに構成されている場合でも、いつでも `$System.License.Login(UserIdentifier)` を呼び出して、明示的にユーザのライセンスを識別できることを覚えておいてください。

23.12.2 ユーザ名ライセンス

`$USERNAME` の値を使用してライセンス付与するユーザを識別できます。これにより、IP アドレスのみを使用して個々のユーザを確実に識別できるわけではない状況でも、より正確なカウントが可能です。

`%SYSTEM.License` クラスの `$SYSTEM.License.UserNameLicensing()` メソッドを使用してライセンス・ユーザ ID を指定する方法を変更できます。既定で、InterSystems IRIS ではクライアント IP アドレスを使用して、ライセンス追跡サブシステムに対してユーザを識別します。最小ではない初期セキュリティ設定で InterSystems IRIS をインストールした場合は、各プロセスにはユーザ ID (`$USERNAME`) があります。`$SYSTEM.License.UserNameLicensing()` システム・メソッドを呼び出して、InterSystems IRIS ライセンス・サブシステムで、ライセンス・ユーザ識別子として `$USERNAME` を使用するようになります。

`$SYSTEM.License.UserNameLicensing()` メソッドはシステムの状態を変更します。このメソッドを `SYSTEM^%ZSTART` から呼び出して、インスタンスの起動時にユーザ名ライセンスを有効にすることができます。このメソッドには、以下の関数があります。

- ・ `$SYSTEM.License.UserNameLicensing(1)` – `$USERNAME` ベースのライセンスを有効にし、前の状態を返します。
- ・ `$SYSTEM.License.UserNameLicensing(0)` – `$USERNAME` ベースのライセンスを無効にし、前の状態を返します。
- ・ `$SYSTEM.License.UserNameLicensing()` – 現在の状態を返します。特殊なログイン規則を使用するライセンス・タイプの引数で呼び出された場合は、エラーを返す可能性があります。

以下の例は、ユーザ名ライセンスが現在有効であるか無効であるかを示します。

ObjectScript

```
Write " Username Licensing",!
Write " 1-enabled, 0-disabled",!
Write $SYSTEM.License.UserNameLicensing(),!
```

以下の例は、ユーザ名ライセンスを有効にし、その後無効にします。

ObjectScript

```
Set RC=$SYSTEM.License.UserNameLicensing(1)
Write RC,!
Set RC=$SYSTEM.License.UserNameLicensing(0)
Write RC
```

詳細は、“ObjectScript リファレンス”の“`$USERNAME`”特殊変数を参照してください。

23.12.3 ライセンス・ログインの特別な考慮事項

ライセンス・ログインに関して、以下に示す特別な考慮事項に注意してください。

- ・ ログインにおいて、CSP 接続は特殊です。インターシステムズは、CSP アプリケーションでは `%CSP.Session` と同様のメソッド `%CSP.Session.Login` を使用して、ユーザのライセンスを識別することを強くお勧めします。そうしない場合、ライセンス・ユーザ ID として Web セッション ID が使用されますが、各セッションで 1 つのライセンス・ユニットを消費するので、多くの場合は不適切な設定となります。例えば、ユーザは複数のブラウザ・ウィンドウを同時に開くことができます。また、ユーザは複数の経路を使用して接続できます。この場合、`%CSP.Session` メソッド

%CSP.Session.Login(username, password) を使用して、セッションに対する明示的なライセンス・ログインを実行できます。

注釈 CSP セッションが終了 (ログアウトまたはタイムアウト) し、ユーザがアクセスしているのが 1 ページのみの場合、CSP はすぐにライセンスを解放しません。代わりに CSP はライセンスを予約し、そのユーザに最大 10 分の猶予期間を与えます。

- 匿名の SOAP 要求および REST 要求 (つまり、インスタンス認証が不要の要求) は、最短 10 秒で 1 つのライセンス・ユニットを消費しますが、ユーザを識別する SOAP 要求または REST 要求は、“ユーザ要求”と見なされるため、ライセンスが必要です。
- InterSystems IRIS は、バックグラウンド・プロセスを区別したり、これらのプロセスを個別にカウントすることはありません。ユーザ・プロセスが他のプロセスを開始する場合、その子プロセスはユーザの最大許容プロセス数全体に対し 1 つとしてカウントされます。
- [新しいタスク] ページ ([システムオペレーション]→[タスクマネージャ]→[新しいタスク])** (“タスク・マネージャの使用”を参照) を使用して作成した各タスクは、**[タスクを実行するユーザ]** セレクタで指定した InterSystems IRIS のユーザ名と、ホスト IP アドレスに変換されるループバック IP アドレス (127.0.0.1) に基づいたライセンス・ユーザ ID で 1 つのライセンス・ユニットを消費します。これにより、特定のユーザとして別々のホスト上で実行されるタスクが、そのユーザの最大許容プロセス数 (“ユーザの識別”を参照) に対して、確実に合計されるようになります。
- ユーザの起動ルーチン (%ZSTART および ^ZMIRROR) によって始まるプロセスも特別なケースです。このルーチンを実行中のプロセスには、親プロセスがありません。したがってログインは、このルーチンが呼び出される前に、そのユーザ ID User Startup に対して実行されます。このルーチンから Job コマンドによって開始されたプロセスは、このユーザ ID を持ちます。必要な場合は、ルーチンから \$System.License.Login(UserId) を呼び出して、ユーザ ID を変更できます。これは、このルーチンで開始するバックグラウンド・プロセスが maxconn より 1 つ少なく、消費するライセンスが 1 つで済むことを意味します。ライセンス条件に従って、これらのプロセスに個別のライセンスが必要な場合 (例えば、個別のライセンスが必要な実験装置の一部を起動している場合など)、適切なユーザ ID に対するライセンスを取得するために \$System.License.Login(UserId) を呼び出す必要があります。

23.13 アプリケーション・ライセンス

インターシステムズ・アプリケーション・パートナーは、アプリケーション・ライセンスを使用して、独自のライセンスの用途に合わせて InterSystems IRIS のライセンス容量を活用できます。InterSystems IRIS は、独自のアプリケーション・ライセンスと同じ方法でアプリケーション・ライセンスを管理します。使用カウントを管理して必要に応じてユーザ・ライセンスを取得および返却します。プロセスまたは Web セッションによって消費されるアプリケーション・ライセンスは、プロセスが終了したり、停止したり、プロセス・テーブルから削除された場合や Web セッションがタイムアウトしたり削除された場合に、プロセスまたはセッションによって消費される InterSystems IRIS ライセンスと共に自動的に解放されます。

アプリケーション・ライセンスは、標準の .ini 形式のファイルまたはファイルのセクションで、アプリケーションを識別するセクション・ヘッダといくつかの keyword=value ペアを含みます。このペアは、ライセンス内で一意であり、ライセンス付与された機能を表します。適切にフォーマットされたアプリケーション・ライセンスは、[実行時にアプリケーションによって InterSystems IRIS にロード](#)できます。

[アプリケーション・ライセンス API](#) には、アプリケーションがユーザやプログラムの代わりにライセンスを消費および返却したり、使用中のライセンス数や使用可能なライセンス数などのアプリケーションと機能のライセンスに関する情報を取得したりできるメソッドやクエリが含まれます。

23.13.1 アプリケーション・ライセンスのロード

アプリケーションの実行時に \$SYSTEM.License.LoadAppLicenseFile メソッドを使用すると、どのようなアプリケーション・ライセンスでも InterSystems IRIS インスタンスでロードしてアクティブ化できます。このメソッドの詳細は、[%SYSTEM.License](#)

のクラス・リファレンスに記載されています (“[アプリケーション・ライセンス API](#)” を参照)。この方法でロードされたアプリケーション・ライセンスは、アクティブな InterSystems IRIS ライセンスには関連付けられませんが、InterSystems IRIS インスタンスによって個別に追跡されます。

各アプリケーション・ライセンスは、[AppName] で始まるセクションに含まれます。アプリケーション名 (AppName) には、ピリオド(.)は使用できません。ライセンスの残りは、ライセンス付与された機能を表す、繰り返されない一連の keyword=value ペアで構成されます。必要なフォーマットの詳細は、\$SYSTEM.License.LoadAppLicenseFile メソッドのドキュメントを参照してください。

以下のアプリケーション・ライセンスのサンプルでは、keyword=value のペアを使用して複数のアプリケーション機能に対してライセンス付与されるユーザの数を制限し、すべてのユーザに対して Extended Lab Reports 機能を有効にします。

```
[BestLabApplication]
Lab Users=50
Lab Administrators=2
Lab Devices=5
Extended Lab Reports=Enabled
```

アプリケーション・ライセンスには、改ざんを防ぐ InterSystems IRIS の保護は適用されませんが、カスタムのアプリケーション・コードによって保護できます。例えば、チェックサムをキーワード・セクションに埋め込み、有効化する前にアプリケーションによって検証できます。

23.13.2 アプリケーション・ライセンス API

%SYSTEM.License クラスは、アプリケーション・ライセンスを管理するための以下のメソッドを提供します。

LoadAppLicenseFile()

ファイルからアプリケーション・ライセンスをロードします。

GetAppLicense()

現在ロードされているアプリケーション・ライセンスのコンテンツを keyword=value のペアの文字列として返します。

GetAppLicenseMaxUsed()

ライセンスのロード以降、アプリケーション・ライセンス・キーワードに関連付けられている、消費されるライセンスの最大数を取得します。

GetAppNames()

現在インターシステムズ、インターシステムズ・アプリケーション・パートナー、あるいはその両方によってライセンス付与されているアプリケーションの名前を返します。

IsAppLicensed()

アプリケーションと機能のどちらがライセンス付与されているのかを示します。

ReturnApplicationLicense()

現在のプロセスのライセンス・ユーザ ID または CSP の場合はセッション・ライセンス ID に関連付けられているライセンス・リソースを含むアプリケーション・ライセンスおよび機能のライセンスを解放します。

SetConnectionLimit()

このインスタンスでユーザごとに許容される最大接続数を設定します。これにより、システム管理者は、1 人のユーザが使用可能なライセンス数すべてを消費しないようにすることができます。

ApplicationServerLogin()

現在ロードされているアプリケーション・ライセンスを使用しているインスタンスの数が、キーで指定した制限を超過しているかどうかを示します。

GetAppLicenseValue()

アプリケーション・ライセンス・キーワードに関連付けられている値を取得します。

GetAppLicenseMinimum()

アプリケーション・ライセンス機能キーワードに関連付けられている使用可能なライセンスの最低数を取得します。キーワードに関連付けられている値は数値です。

GetAppLicenseFree()

アプリケーション・ライセンス機能キーワードに関連付けられている使用可能なライセンスの最大数を取得します。キーワードに関連付けられている値は数値です。

GetProcessAppsTaken()

プロセス PID に対して取得されたアプリケーション・ライセンス・タイプの \$List (存在しない場合は "") を取得します。

ApplicationUserList()

ユーザ ID によってライセンスとアプリケーション・ライセンスを返します。

24

Windows での InterSystems IRIS の管理

Microsoft Windows プラットフォームでの InterSystems IRIS® データ・プラットフォーム・インスタンスの管理は簡単です。大部分のタスクは、管理ポータルと InterSystems IRIS ランチャー (キューブとも呼ばれる) を使用して実行できます。コマンド・プロンプトから InterSystems IRIS インスタンスを制御することもできます。

このトピックでは、install-dir を使用して InterSystems IRIS のインストール・ディレクトリを表しています。既定のディレクトリは、“インストール・ガイド” の “[既定のインストール・ディレクトリ](#)” のセクションを参照してください。

注意 InterSystems IRIS **IRIS.DAT** データベース・ファイルに対して Windows のファイル圧縮機能を使用しないでください。次に、Windows エクスプローラでファイルまたはフォルダを右クリックし、**[プロパティ]**、**[詳細]**、**[内容を圧縮してディスク領域を節約する]** の順に選択して、ファイルを圧縮します。圧縮後、Windows エクスプローラでフォルダ名またはファイル名が青で表示されます。**IRIS.DAT** ファイルを圧縮した場合、このファイルが属するインスタンスは開始できません。これは誤解を招きやすいエラーです。

24.1 InterSystems IRIS インスタンスへのアクセスの管理

このセクションでは、InterSystems IRIS へのアクセスの管理に関する以下のトピックについて説明します。

- [InterSystems サービス](#)
- [インストール・ツリーへのアクセスの制限](#)
- [InterSystems IRIS サービス・アカウントの変更](#)

24.1.1 InterSystems サービス

すべての InterSystems IRIS ジョブおよびプロセスは、InterSystems サービス InterSystems IRIS Controller for <instance-name> から実行されます。InterSystems サービスが有する権限は、関連する Windows ユーザ・アカウントによって決定されます。これがローカルの SYSTEM アカウントの場合、InterSystems IRIS は、Windows システム上のすべてのファイルと権限への広範なアクセスが可能です。より安全で限定的な環境を維持するには、必要な特権とアクセス権のみを持つ InterSystems サービスの Windows アカウントを選択する必要があります。詳細は、“[Windows ユーザ・アカウント](#)” を参照してください。

通常のインストールやロック・ダウンのインストールでは、InterSystems IRIS は、このインスタンスへのアクセス権を付与する 2 つのローカル・ユーザ・グループを作成します。既定のローカルな SYSTEM アカウント以外に InterSystems サービスの Windows ユーザ・アカウントを指定する場合、InterSystems IRIS は、その Windows ユーザ・アカウントを各グループに追加します。これらのグループは、以下のとおりです。

- **IRISServices。** InterSystems IRIS インスタンスを開始、停止、および制御する権限を付与します。

- ・ **IRIS_Instance_instancename**。インストール・ツリー（つまり、InterSystems IRIS がインストールされているディレクトリおよびそのすべてのサブディレクトリ）へのアクセス権限を付与します。

注釈 これらのグループとその特権に関して、次の点に留意してください。

- ・ **IRISServices** グループには、その作成時に、**Replace a process level token** 特権と **Adjust memory quotas for a process** 特権が付与されます。これらの特権を削除しないでください。
- ・ **IRISServices** および **IRIS_Instance_instancename** グループは、特定のアクションを実行するために InterSystems IRIS が必要とするすべての権限を付与するとは限りません。インストール・ツリーの外部にあるすべてのインスタンス、ジャーナル、およびログ・ファイルに必要なアクセス権を InterSystems IRIS に確実に付与するには、これらのファイルおよびそれらの格納ディレクトリへのフル・アクセス権限を **IRIS_Instance_instancename** グループに付与します。必要に応じてこのグループに追加の権限を付与することもできます。

一般に、InterSystems IRIS Controller for <instance-name> の Windows アカウントはインストール時に選択します。“[Windows ユーザ・アカウント](#)”を参照してください。インストール後にサービス・アカウントを変更するには、以下の“[InterSystems IRIS サービス・アカウントの変更](#)”を参照してください。

24.1.2 インストール・ツリーへのアクセスの制限

既定では、認証されたすべての Windows ユーザがインストール・ツリーにアクセスできますが、これが望ましくない場合もあります。認証されたユーザの Windows アクセス制御エントリ (ACE) を削除するには、次の手順を実行します。

1. Windows 10 などの一部の Windows プラットフォームでは、継承された権限を最初に削除することが必要な場合があります。この操作には、以下のコマンドを使用します。

```
icacls <install-dir> /inheritance:d
```

2. 次に以下のように、認証されたユーザの ACE を削除します。

```
icacls <install-dir> /remove "NT AUTHORITY\Authenticated Users"
```

3. 最後に、以下のコマンドを使用して、“認証されたユーザ”への参照が存在しないことを確認します。

```
icacls <install-dir>
```

このコマンドは、インストール・ディレクトリに対するアクセス権をリストします。

これで、インストール・ツリーにアクセスできるのは、管理者ユーザか、**IRIS_Instance_instancename** グループのユーザだけになります。

重要 これを実行しない場合、ホスト Windows システムにログインできるユーザであれば誰でも容易にファイルや設定を変更したり、InterSystems IRIS インスタンス全体を無効にしたりすることができます。

場合によっては、InterSystems IRIS Controller for <instance-name> サービスで使用するアカウントに加え、別の Windows アカウントにインストール・ツリーへのアクセス権を付与することができます。これには、例えば自動化タスクを実行するアカウント、あるいは Windows サーバに直接ログインし、(ローカルのターミナル・セッションを使用するか、またはカスタム・コールイン実行可能プログラムを呼び出して) InterSystems IRIS にアクセスするアカウントが含まれます。そのようなアカウントに必要なアクセス権限を付与するには、そのアカウントを **IRIS_Instance_instancename** グループに追加します。

24.1.3 InterSystems IRIS サービス・アカウントの変更

コマンド行に以下を入力し、InterSystems サービス InterSystems IRIS Controller for <instance-name> で使用する Windows ユーザ・アカウントを変更します。


```
<install-dir>%bin%IRISinstall.exe setserviceusername <instance-name> <username> <password>
```

このコマンドは、Windows ユーザ・アカウントを、指定したアカウントに変更します。また、必要に応じて **IRISServices** グループと **IRIS_Instance** instancename グループを作成し、そのユーザをこれらのグループに追加します。このコマンドを実行して InterSystems IRIS インスタンスを再起動すると、インスタンスは新たに指定された Windows ユーザ・アカウントで実行されます。

24.2 InterSystems IRIS ランチャー

Microsoft Windows プラットフォームでの InterSystems IRIS の主要なインタフェースは、InterSystems IRIS ランチャーです。InterSystems IRIS ランチャーからすべての InterSystems IRIS 構成および管理ツールを起動できます。また、ショートカットやコマンド行から各ランチャー・コマンドを起動することもできます。

同様に、Windows のプログラム・メニューから多くの InterSystems IRIS ツールを開始できます。この場合は、**[IRIS]** フォルダをポイントして、該当する InterSystems IRIS インスタンス名の **[InterSystems IRIS の開始]** をポイントします。

Windows ベースのシステムで InterSystems IRIS を起動すると、タスクバーのシステム・トレイに InterSystems IRIS ランチャー  が表示されます。

InterSystems IRIS ランチャーをクリックすると、ObjectScript ユーティリティとプログラミング環境を使用するためのコマンドが並んだメニューが表示されます。

以下の表に、InterSystems IRIS ランチャー・メニューから使用できるコマンドを示します。

InterSystems IRIS ランチャー・コマンド	概要
はじめに	チュートリアル、リリース・ノート、ドキュメント、およびその他の関連情報へのリンクが表示されます。
InterSystems IRIS の開始	メニュー項目の後の角括弧内に指定されている既定のインスタンス（例えば [ii2081]）を開始します。InterSystems IRIS サーバが既に開始されている場合は、このオプションは淡色で表示され、選択できません。 注釈 インスタンスが自動的に起動されないようにする方法の詳細は、“ メモリと開始設定 ”を参照してください。
InterSystems の停止	ローカルの InterSystems IRIS インスタンスをシャットダウンまたは再起動します。InterSystems IRIS サーバが停止している場合は、このオプションは淡色で表示され、選択できません。
スタジオ	InterSystems IRIS クラス定義およびルーチンの作成、編集、削除、およびコンパイルを行います。詳細は、“ スタジオの使用法 ”を参照してください。
ターミナル	InterSystems IRIS のコマンド行インタープリタを呼び出します。詳細は、“ ObjectScript シェルの使用法 ”を参照してください。
管理ポータル	一般的なシステム管理タスクを構成します。データベースとネームスペースを作成し、すべての InterSystems IRIS 構成設定を調整します。クラス、グローバル、ルーチン、および機能を表示し、それぞれを管理します。テーブルとビューを表示し、クエリと SQL 管理関数を構成します。“ 管理ポータルの使用 ”を参照してください。
ドキュメント	InterSystems IRIS オンライン・ドキュメントを表示します。

InterSystems IRIS ランチャー・コマンド	概要
優先接続サーバ [server name]	リモート・サーバのリストを表示し、サブメニューの [追加/編集] コマンドでサーバ接続を維持管理します。優先接続サーバは括弧で囲まれており、サーバ・リストでは横にチェック・マークが付いています。“ リモート・サーバ接続の定義 ”を参照してください。
Caché バージョン情報	InterSystems IRIS のバージョンおよびビルドの情報を表示します。
終了	[InterSystems IRIS ランチャー] アイコンをシステム・トレイから削除します。これによって InterSystems IRIS が停止することはありません。インスタンスが再起動されるとランチャーが再度表示されます。

24.3 InterSystems IRIS の開始

InterSystems IRIS を開始するには、システム・レベルでスタートアップ・プロシーダを実行します。このプロシーダは、既定の構成ファイル、または指定した構成ファイルのいずれかを使用して実行します。

注釈 InterSystems IRIS の開始で問題が発生した場合は、“[ログ・ファイルの監視](#)”の説明に従って、`messages.log` ファイルを表示してください。

Windows プラットフォームで InterSystems IRIS を開始するには、InterSystems IRIS ランチャーの [InterSystems IRIS の開始] を選択します。これにより、指定された構成ファイルを使用して InterSystems IRIS インスタンスが起動します。InterSystems IRIS が起動していないときは、[InterSystems IRIS ランチャー] アイコンが淡色表示になっています。

InterSystems IRIS ランチャーがシステム・トレイにない場合は、Windows のプログラム・メニューから [IRIS] フォルダを選択して、対象のインスタンスの [InterSystems IRIS の開始] を選択します。ランチャーをシステム・トレイに表示するには、`install-dir/bin` ディレクトリに移動し、`iristray.exe` ファイルをダブルクリックします。

あるいは、[スタート] メニューで [ファイル名を指定して実行] をクリックし、[名前] フィールドで `install-dir/bin` ディレクトリから上記コマンドを入力することもできます。例えば、`MyIris\bin` ディレクトリから `MyIris` というインスタンスを起動する場合、以下のコマンドを入力します。

```
c:\MyIris\bin\iris start iris
```

InterSystems IRIS を起動するこれらのメソッドは、`iris start` コマンドを呼び出します。`iris` コマンドのオプションと情報の詳細は、“[コマンド・プロンプトでの InterSystems IRIS の制御](#)”を参照してください。

24.4 InterSystems IRIS の停止

通常は、InterSystems IRIS システムを実行したままの状態にしておきます。しかし、オペレーティング・システムに再起動が必要であれば、システムをシャットダウンする前に InterSystems IRIS を停止します。バックアップやデータベースの修復ユーティリティなどのメンテナンス・タスクの場合は、InterSystems IRIS を停止させる必要はありません。

InterSystems IRIS ランチャー・メニューから [InterSystems の停止] をクリックして、ローカル InterSystems IRIS インスタンスの終了や再起動を行います。このオプションの既定では、既定の停止ルーチンを使用して、直ちに InterSystems IRIS を終了（または再起動）します。遅延シャットダウンのタイマ設定、ユーザ定義のシャットダウン・ルーチンの実行、サーバ上のユーザへの警告メッセージ発行、およびフェイルオーバーなしのシャットダウンといったオプションも用意しています。Windows のプログラム・メニューからこの同じプロセスを実行できます。[IRIS] フォルダを選択して、対象のイ

インスタンス名の **[InterSystems IRIS の停止]** を選択します。カウントダウンが 0 になって終了のプロシージャが開始されると、終了処理の中止はできません。

重要 正常に終了するためにも、**[InterSystems IRIS の停止]** を実行して InterSystems IRIS を終了することをお勧めします。

InterSystems IRIS を停止するこれらのメソッドは、`iris stop` コマンドを呼び出します。`iris` コマンドのオプションと情報の詳細は、“[InterSystems IRIS インスタンスの制御](#)” を参照してください。

InterSystems の停止コマンドの削除

[InterSystems の停止] コマンドを誤って実行してしまうことを防ぐために、対応する InterSystems IRIS インスタンスの `install-dir/bin` ディレクトリから `irisstop.exe` ファイルを削除することにより、InterSystems IRIS ランチャーからこのコマンドを削除できます。**[スタート]** メニューの該当する InterSystems IRIS インスタンスから **[InterSystems の停止]** ショートカットを削除することもできます。**[プログラム]** から対象の InterSystems IRIS インスタンス名をポイントして、**[InterSystems の停止]** を右クリックし、**[削除]** をクリックします。

24.5 コマンド・プロンプトからのヘルプ情報

`install-dir\bin` ディレクトリで `iris.exe` プログラムを実行することによって、Windows のコマンド・プロンプトから InterSystems IRIS インスタンスを制御することができます。`iris` コマンドの詳細は、“[InterSystems IRIS インスタンスの制御](#)” を参照してください。

このコマンドの最新のヘルプ情報を表示するには、`install-dir\bin` ディレクトリから **`iris help`** を呼び出します。以下に例を示します。

```
C:\MyIris\bin>iris help
```

`iris help` コマンドによって、ターミナルに最新のヘルプ情報が表示されます。`>outputfilename` を **`iris help`** コマンドに追加することによって、ヘルプ情報を `install-dir\bin` ディレクトリ内のファイルに保存することもできます。次に、例を示します。

```
C:\MyIris\bin>iris help>helpinformation
```

24.6 コマンド行での InterSystems IRIS への接続

`iris terminal` コマンドを使用して、コマンド行で InterSystems IRIS インスタンスにログインできます。`iris terminal` の詳細は、“[InterSystems IRIS インスタンスの接続](#)” を参照してください。

25

UNIX®、Linux、および macOS での InterSystems IRIS の管理

ここでは、UNIX®、Linux、および macOS に固有の管理手順について説明します。

25.1 UNIX® のユーザ、グループおよび権限

UNIX® プラットフォームでの InterSystems IRIS® データ・プラットフォームのインストールはすべて、以下のユーザおよびグループを有しています。

- ・ root – InterSystems IRIS は root によってインストールされ、InterSystems IRIS システム・デーモンによる処理の一部は root として実行する必要があります。
- ・ インスタンスの所有者 – このユーザがほとんどのインストール・ファイルが所有し、インスタンスを完全にコントロールします。最小の初期セキュリティ設定でインストールした場合は、root が既定の所有者となります。それ以外の場合、インストール中に所有者の指定を求められます。
- ・ InterSystems IRIS スーパーサーバの実効ユーザとそのジョブ – 受信要求処理のためにスーパーサーバにより生成された InterSystems IRIS プロセスはすべて、このユーザとして実行されます。ジョブ・サーバ・プロセスによりホストされるジョブのほかにタスク・マネージャのジョブおよびユーザ定義の起動ルーチン (^%ZSTART など) もこのユーザで実行されます。既定では、このユーザが irisusr となりますが、[カスタム・インストール](#)中にユーザを変更することができます。
- ・ InterSystems IRIS プロセスの実効グループ – すべての InterSystems IRIS プロセスは、自動的にこのグループとして実行され、InterSystems IRIS 内部においての通常のユーザが InterSystems IRIS データベースやジャーナル・ファイルにアクセスできますが、それまではこれらのファイルへのアクセスが認められない場合があります。これらおよびその他の InterSystems IRIS ファイルに対してファイルのアクセス権限を設定することにより、このグループが適正なアクセスを得られます。安全なシステムでは、InterSystems IRIS スーパーサーバとそのジョブの実効ユーザのみがこのグループのメンバーとなります。既定では、このグループに含まれるのは irisusr ですが、[カスタム・インストール](#)中にグループを変更することができます。
- ・ インスタンスの開始と停止を許可されたグループ – このグループ、root、およびインスタンスの所有者は、InterSystems IRIS を開始および停止することができます。

すべてのジャーナルおよびジャーナル・ディレクトリは、InterSystems IRIS プロセスの実効グループのグループに設定されたグループ所有権を備え、そのグループに完全アクセス権限（ジャーナルには rw、ジャーナル・ディレクトリには rwx）

を与える必要があります。ジャーナルおよびジャーナル・ディレクトリを所有しているユーザは、それらが作成された方法によって異なる場合があります。

InterSystems IRIS 内で作成されたジャーナルおよびジャーナル・ディレクトリは、適切なアクセス権限で作成されます。ただし、(スクリプトまたは管理者のアクションを通じて) 外部的にジャーナルおよびジャーナル・ディレクトリの移動、コピー、作成を行う場合は、適正なアクセス権限が確実に維持されるようにする必要があります。アクセス権限を適切に設定できないと、不測かつ深刻なエラーを誘発するおそれがあります。

以下の例では、InterSystems IRIS プロセスの実効グループが `irisusr` であり、インスタンスの所有者が `irisowner` ですが、ファイルは作成された状況により、異なるユーザ所有権を備えていることを想定しています。例えば以下のようになります。

```
journal directory    irisowner    irisusr    drwxrwxr-x
20170801.001        irisowner    irisusr    -rw-rw----
```

注釈 これらの設定は部分的に、InterSystems IRIS インストールの `install-dir/bin` ディレクトリ内の実行可能ファイルに対する一連のアクセス権限として維持されます。関連するプロパティには、所有権、グループ、モード、`set-UID`、`set-GID` ビットがあります。オペレーティング・システム・レベルでの管理タスク実行時に、これらのアクセス権限を変更しないよう注意してください。

[InterSystems IRIS filecheck ユーティリティ](#)を使用して、UNIX インストールの権限、所有者、およびグループが必要な設定に従っているかどうかを確認できます。

25.1.1 データベースおよびデータベース・ディレクトリのアクセス権限

すべてのデータベースおよびデータベース・ディレクトリは、InterSystems IRIS プロセスの実効グループのグループに設定されたグループ所有権を備え、そのグループに完全アクセス権限 (データベースには `rw`、データベース・ディレクトリには `rwX`) を与える必要があります。データベースおよびデータベース・ディレクトリを所有しているユーザは、それらが作成された方法によって異なる場合があります。

InterSystems IRIS 内で作成されたデータベースおよびデータベース・ディレクトリは、適切なアクセス権限で作成されます。しかし、(スクリプトまたは管理者のアクションを通じて) 外部的にデータベースおよびデータベース・ディレクトリの移動、コピー、作成を行う場合、適正なアクセス権限を確実に維持する必要があります。アクセス権限を適切に設定できないと、不測かつ深刻なエラーを誘発するおそれがあります。

以下の例では、InterSystems IRIS プロセスの実効グループが `irisusr` であり、インスタンスの所有者が `irisowner` ですが、ファイルは作成された状況により、異なるユーザ所有権を備えていることを想定しています。

```
dataset directory    irisowner    irisusr    drwxrwxr-x
IRIS.DAT             irisowner    irisusr    -rw-rw----
```

25.2 UNIX® での起動

InterSystems IRIS インスタンスがプロセスの開始、停止、および新規プロセスの作成の制御に使用するリソースは、以下のリソースです。

- `install-dir¥mgr` ディレクトリの `iris.ids` ファイル。
- 共有メモリ。

25.2.1 デーモン・リソース・ロック

InterSystems IRIS では、推奨されるファイル・ロックを使用して、異なるマシン上で同一のインスタンスが同時に複数で起動されないようにします。推奨されるファイル・ロックでは、1 つのロック・ファイル (ここでは、`install-dir/mgr` ディレクトリのファイル `ilock`) を複数リソースの排他ロックに使用できます。制御プロセス、ライト・デーモン、およびジャーナル・デーモンのそれぞれによって、ロック・ファイル内でロックされるセクションが異なります。`ilock` ファイルの当該のセクションが既にロックされている場合は、起動が強制終了します。別のデーモンが保持しているロックは、デーモン・リソース・ロックと呼ばれます。

ファイル・ロックはプロセスが終了するまで、そのプロセスによって保持されます。このように、保持されているロックが存在している場合は、いずれかのノードでいずれかのデーモン・プロセスが実行中であることが示されます。ただし、インスタンスの実行が正常であるかどうか、問題なく実行されているかどうかは示されません。

25.2.2 iris.ids ファイル

`iris.ids` ファイルには、InterSystems IRIS が開始されたノードの名前が格納されます。`iris.ids` ファイルの存在によって、ObjectScript ユーティリティ、およびユーザ作成のスクリプトは、インスタンスが起動され実行中であることを認識できます。たいていの場合、起動時にはこのファイルが無視されます。ただし、`iris.ids` の読み取り時にエラーが発生した場合は、InterSystems IRIS が起動されなくなります。これまでのバージョンの InterSystems IRIS では、共有メモリ識別子も `iris.ids` ファイルに格納されていましたが、今回のリリースからは格納されません。

25.2.3 起動シーケンス

起動シーケンスの理解を深めるため、インスタンスがノード A とノード B という 2 つの異なるノード (マシン) から起動できると考えてみましょう。`iris.ids` ファイルは、(共有ファイルに対する) デーモン・リソース・ロックの場合と同様に両方のノードから認識できます。しかし、共有メモリ自体は、作成されたノード (つまり、InterSystems IRIS を起動したノード) でのみ認識できます。

25.2.3.1 手順 1. インスタンスの状態のチェック

起動ルーチンによって `irisdb -cV` が実行され、インスタンスの状態が検出されます。最初、インスタンスの共有メモリへのアタッチが試行されます。

- ・ インスタンスの共有メモリがない場合は、デーモン・リソース・ロックが調べられます。
 - 保持されているデーモン・リソース・ロックが存在しない場合、インスタンスは停止しているとレポートされます。
 - デーモン・リソース・ロックが保持されている場合、`iris.ids` ファイルで指定されたノード上でインスタンスが実行中であるとレポートされます。`iris.ids` ファイルが存在しない場合は、デーモンを実行中の場所に関する情報を取得できません。

アクション：ユーザは `iris stop`、または `iris force` を実行し、該当するノード上で実行中のインスタンスを中止する必要があります。これによってデーモンが停止し、`iris.ids` ファイルが削除されます。
- ・ アタッチが成功した場合、システムは起動され、実行中であると見なされます。この状態がユーザにレポートされます。起動は中止されます。
- ・ 共有メモリがまだアタッチされているため、起動を完了できないことを示すエラーが表示されている場合は、メモリが解放されるまで数分間待機します。エラーが続く場合は、`iris.shid` ファイルに関連するエラーが生じている可能性があります。このファイルは、macOS 以外のすべての UNIX® プラットフォームで InterSystems IRIS の共有メモリ ID を追跡します。このファイルの修正にサポートが必要な場合は [インターシステムズのサポート窓口](#) にお問い合わせください。

- ・ オプションの `filecheck` パラメータを使用して起動が開始された場合は、[InterSystems IRIS filecheck ユーティリティ](#)によって、InterSystems IRIS のシステム・ファイルに関連付けられている権限、所有者、およびグループがシステム要件を満たしていることが確認されます。

25.2.3.2 手順 2. InterSystems IRIS の起動

InterSystems IRIS の起動プロセスが実行されます。チェックが繰り返し実行され、起動リソースに対して別の起動処理が完了していないことが保証されます。

- ・ デーモン・リソース・ロックが保持されている場合は、このインスタンスのいずれかのノードで 1 つ以上のデーモンが実行されていることを示します。この場合 InterSystems IRIS はこれをレポートして、エラーによって終了します。起動は中止されます。

`iris.ids` ファイルが存在しないと、デーモンが実行されているノードが不明になります。

アクション：いずれかのノード上で起動が行われていると見なされます。インスタンスが開始されたノードを判定するには、`iris.ids` ファイルを調べます。

InterSystems IRIS の起動が続行されます。

25.3 InterSystems IRIS の管理

`sysmgr` グループ内のユーザ ID を持つユーザは、シェルから `iris` を実行できます ("[InterSystems IRIS インスタンスの制御](#)" を参照)。このコマンドは、`install-dir/bin` ディレクトリの InterSystems IRIS の実行可能ファイルやスクリプトを呼び出します。以下のセクションでは、InterSystems IRIS インスタンスでこれらの管理タスクを実行する方法について説明します。

- ・ [InterSystems IRIS の開始](#)
- ・ [InterSystems IRIS の実行](#)
- ・ [InterSystems IRIS の停止](#)

重要 インストールの所有者は、該当インスタンスの開始および停止、システム管理の実行、該当インスタンスの診断プログラムの実行などが可能な、すべての権限を保持します。

インスタンスの所有者であるユーザ ID のみが、すべての診断操作を実行できます。また、そうする必要があります。これで、作成されたすべてのファイルやリソースは、`root` ユーザではなくインスタンスの所有者のものとなります (`root` ユーザ以外はこれらのリソースにアクセスできなくなります)。このため、`root` ユーザの所有ではないインスタンスに対して、`root` ユーザが何らかの管理 (インスタンスの開始や停止など) をできないようにすることをお勧めします。`root` ユーザが管理できるインスタンスは、その `root` ユーザが所有するもののみとする必要があります。

25.3.1 InterSystems IRIS の開始

InterSystems IRIS を開始するには、システム・レベルでスタートアップ・プロシージャを実行します。このプロシージャは、既定の構成ファイル、または指定した構成ファイルのいずれかを起動します。

注釈 InterSystems IRIS の開始で問題が発生した場合は、"[ログ・ファイルの監視](#)" の説明に従って、`messages.log` ファイルを表示してください。

コンソール・マシンでない場合は、Telnet を実行して、InterSystems IRIS がインストールされているターゲットのマシンに接続します。UNIX® で InterSystems IRIS システムを開始する前に、以下の条件のうち 1 つを満たしている必要があります。

- ・ スーパーユーザであること
- ・ root ユーザとしてサイン・インしていること (root 以外のアカウントでログインしているときでも、su (スーパーユーザ) コマンドで root に変更できます)。
- ・ UNIX® グループ ID が、システムの停止と開始の特権を持つよう InterSystems IRIS のインストールで指定したグループに一致していること。

インストール時にこれらの特権を指定する方法の詳細は、“[UNIX® および Linux への InterSystems IRIS のインストール](#)” を参照してください。

iris コマンドを使用して InterSystems IRIS を開始します。

```
iris start <instname>
```

instname は、開始する InterSystems IRIS インスタンスの名前です。その他のオプションと情報については、“[InterSystems IRIS インスタンスの制御](#)” を参照してください。

sysmgr グループ内のユーザ ID を持つユーザは、シェルから iris start を実行できます。このコマンドにより、現在のノードまたは別のノードでインスタンスが実行されていないことの確認、複数の補助ライト・デーモン (AWD) を含む基本的な InterSystems IRIS デーモンや共有メモリの作成、その他のデーモン (ECP デーモンなど) を作成するスタートアップ・ルーチン (STU) の実行の後に、ユーザのログインが許可されます。

25.3.2 InterSystems IRIS の実行

ユーザはそのユーザ ID およびグループ ID (この例では anyuser:anygroup) に関係なく、シェルから iris terminal を実行できます (“[InterSystems IRIS インスタンスの接続](#)” を参照)。これによって install-dir/bin ディレクトリで irisuxsession が実行されます。

InterSystems IRIS を anyuser:irisusr として実行すると、Kerberos ネゴシエーションを含む標準スタートアップ・ロジックが実行され、\$USERNAME と一連のログイン・ロールが識別されます。多くの場合、この \$USERNAME 値は、iris terminal を呼び出した実際のユーザと関連付けられています。したがって、どのユーザも InterSystems IRIS を実行できますが、InterSystems IRIS でのユーザのアクティビティは、そのユーザに割り当てられたセキュリティ・ロールによって定義され、制限されます。

注意 install-dir/bin ディレクトリから InterSystems IRIS の実行可能ファイルを直接呼び出す方法で、InterSystems IRIS を開始しないでください。

InterSystems IRIS 実行可能ファイルは、それ自体は setgid 実行可能ファイルではありません。InterSystems IRIS を開始するユーザの代わりにグループを適切に設定するのは、iris terminal ラップの役割です。

iris terminal に加え、/usr/bin ディレクトリの irisdb 実行可能ファイルを使用して InterSystems IRIS を実行することもできます。このコマンドは、iris terminal <default-instance> を呼び出し、権限を設定して、既定の InterSystems IRIS インスタンスのターミナルを開きます。irisdb 実行可能ファイルを作成するには、iris default <instname> 関数を実行することにより、既定のインスタンスを指定します。

注釈 macOS 10.11 以降では、iris と iris terminal は /usr/bin からのリンクを使用して、/usr/local/bin に格納されます。

25.3.3 InterSystems IRIS の停止

通常は、InterSystems IRIS システムを実行したままの状態にしておきます。しかし、オペレーティング・システムに再起動が必要であれば、システムをシャットダウンする前に InterSystems IRIS を停止します。バックアップやデータベースの修復ユーティリティなど InterSystems IRIS のメンテナンス・タスクの場合は、InterSystems IRIS を停止させる必要はありません。

UNIX® で InterSystems IRIS を停止するには、InterSystems IRIS の開始時と同じ手順を行います。以下の条件のうち 1 つを満たしている必要があります。

- ・ スーパーユーザであること
- ・ ルート・ユーザとしてサイン・インしていること (root 以外のアカウントでログインしているときでも、su (スーパーユーザ) コマンドで root に変更できます)。
- ・ UNIX® グループ ID が、システムの停止と開始の特権を持つよう InterSystems IRIS のインストールで指定したグループに一致していること。

コマンド行から InterSystems IRIS を停止するには以下の作業を行います。

1. iris stop コマンドを使用します。

```
iris stop <instname>
```

instname は、停止する InterSystems IRIS インスタンスの名前です (その他の iris オプションと情報については、“[InterSystems IRIS インスタンスの制御](#)” を参照してください)。

注意 iris force コマンドで InterSystems IRIS を停止することができますが、データが失われることがあるので、これを行うときは注意が必要です。

2. このプロシージャは InterSystems IRIS SHUTDOWN ユーティリティを呼び出し、状態レポートを表示します。レポートで起動中のプロセスをチェックし、次のステップが必要かどうかを決定します。
3. 必要な場合は、システム上のすべてのユーザにメッセージを送信します。

```
Do you want to broadcast a message to anyone? No=> Yes
Send a message to other terminals. Message => Please sign off
Terminal => /dev/tty/06
Terminal =>
Message =>
```

4. Message プロンプトで **Enter** キーを押すまで、メッセージの送信後も他のメッセージを送信できます。
5. 他のシステムの状態を参照するかどうかを尋ねられた場合、参照するには「**Yes**」と入力し、参照する必要がない場合は **Enter** キーを押します。
6. 「**Yes**」と応答すると、システムの状態が再度表示され、動作中のターミナルを確認できます。
7. 停止したいときは「**Yes**」と入力します。「**No**」と応答すると、シャットダウン・プロシージャを中止し、InterSystems IRIS は起動されたままになります。

注釈 UNIX® プラットフォームでは、停止、再起動、または強制終了された InterSystems IRIS インスタンスは、すべてのプロセスが共有メモリからアタッチ解除されるまで最大 30 秒待機します。30 秒後にこのインスタンスは閉じます。インスタンスが閉じた後でも共有メモリにアタッチされているプロセスがまだある場合、このインスタンスの再起動は失敗します。

26

リモート・サーバへの接続

ここでは、InterSystems IRIS® データ・プラットフォームのランチャーを使用してリモート・サーバにアクセスする方法、および InterSystems IRIS サーバ・マネージャを使用してサーバ接続を定義する方法を説明します。

ランチャーとサーバ・マネージャは、Windows でのみ提供されます。

26.1 ランチャーのサブメニュー

リモートの InterSystems IRIS サーバに簡単にアクセスできるように、[InterSystems IRIS ランチャー](#)には、リモート・インスタンスをホストしているマシンに対する OS レベルの権限を必要としない以下のオプションが用意されています。

- ・ **[リモート システム アクセス]** サブメニュー。InterSystems IRIS サーバ・マネージャで定義されているあらゆるインスタンスに、ターミナルおよび管理ポータルへのアクセスを提供します。

このサブメニューは、スタジオ、ドキュメント、およびクラス・リファレンスへのアクセスも提供しますが、ここでは具体的には説明しません。

[リモート システム アクセス] は、[Telnet](#) 経由の接続に使用できるオプションも提供します。

- ・ **[優先接続サーバ]** サブメニュー。InterSystems IRIS サーバ・マネージャで定義されているインスタンスのリストを表示し、そのリストの管理に使用できるオプションを提供します。

リモート・インスタンスに対して **[リモート システム アクセス]** サブメニューのユーティリティを使用するには、以下のことを行います。

1. 優先接続サーバ・リストにサーバを追加するため、[リモート・サーバ接続を定義](#)します。
2. InterSystems IRIS ランチャーをクリックし、**[リモート システム アクセス]** を選択します。
3. ユーティリティ (管理ポータルなど) を選択します。
4. サーバ名を選択します。

管理ポータル、ドキュメント、またはクラス・リファレンスに接続する際に、後で迅速にアクセスできるように URL をブックマークに登録できます。

26.2 リモート・サーバ接続の定義

リモート・サーバ上の InterSystems IRIS ランチャー・ユーティリティまたは他の InterSystems IRIS アプリケーションを使用するには、そのサーバが InterSystems IRIS サーバ・マネージャの接続リストに含まれている必要があります。この事前に設定したリモート・サーバのリストを使用すると、すぐに接続が可能です。リモート・サーバは、一意のサーバの IP アドレスや TCP ポートを使用して定義され、これがサーバにおける InterSystems IRIS のインスタンスとなります。

重要 システム上で InterSystems IRIS ランチャー・ユーティリティを使用するには、InterSystems IRIS スーパーサーバがリモート・マシン上で動作し、そのポートがファイアウォール上で開いている必要があります。
セキュリティ上の理由のため、ユーザ名とパスワードはリモート接続情報に保存されません。

リモート・サーバを定義するには、以下のことを行います。

1. InterSystems IRIS ランチャー・メニューから、**[優先接続サーバ]**を選択し、**[追加/編集]**をクリックして、InterSystems IRIS サーバ・マネージャを開きます(このステップを Windows システムで実行するには、管理者権限が必要です)。
2. **[追加]**をクリックして、**[接続を編集]**ダイアログ・ボックスを開きます。
3. 各フィールドに入力し**[OK]**をクリックします。フィールドの詳細は以下のテーブルで説明しています。指定がない限り、すべてのフィールドに入力する必要があります。

- ・ **[サーバ名]** – サーバを識別する名前。InterSystems IRIS ランチャーに選択肢として表示されます。
- ・ **[IPアドレス]** – リモート・サーバの IP アドレス、ホスト名 (DNS サーバを使用の場合)、または FQDN。InterSystems IRIS は、リモート・サーバに適合する名前を参照します。
- ・ **[ポート]** – スーパーサーバのポート番号。既定のポート番号は 1972 です。
- ・ **[Telnetポート]** – Telnet 接続のポート番号。既定のポート番号は 23 です。
- ・ **[ウェブサーバIPアドレス]** – オプション。この InterSystems IRIS インスタンスの管理に使用する Web サーバの IP アドレス。指定しない場合、既定の IP アドレスになります。

管理する InterSystems IRIS インスタンスとは別のマシン上に存在する Web サーバを使用する場合は、このフィールドにその Web サーバ・マシンの IP アドレスを入力します。指定しない場合、既定の IP アドレスになります。

- ・ **[ウェブサーバポート]** – Web サーバのポート番号 (必要に応じて)。
- ・ **[CSPサーバインスタンス]** – オプション。1 つの Web サーバを、複数の InterSystems IRIS インスタンスに接続するように構成する場合に、接続する InterSystems IRIS インスタンス名。詳細は、“[CSP サーバ・インスタンスの指定](#)”を参照してください。

注釈 **[CSPサーバインスタンス]** は、要求が適切にルーティングされるように、すべて小文字にする必要があります。

- ・ **[HTTPS]** – [InterSystems IRIS ランチャー](#)からのリンクが HTTPS を使用するかどうか。
- ・ **[認証方法]** – **[Kerberos]** または **[パスワード]** を選択します。

これらのフィールドの使用法の詳細は、“セキュリティ管理ガイド”の“認証”の章にある“[Kerberos 認証](#)”セクションを参照してください。

InterSystems IRIS でのパスワードベースの認証については、“セキュリティ管理ガイド”の“認証”の章にある“[インスタンス認証](#)”セクションを参照してください。

- ・ **[接続セキュリティレベル]** – Kerberos¹ の場合のみ。**[Kerberos]**、**[Kerberos とパケット整合性]**、または **[Kerberos と暗号化]** のいずれかを選択します。

- ・ **[Windows InterSystems IRIS Telnet サーバ]** – Kerberos の場合のみ。Windows サーバへの接続を定義する場合は、このチェック・ボックスにチェックを付けます。
- ・ **[サービスプリンシパル名]** – Kerberos の場合のみ。このフィールドには、推奨のサービス・プリンシパル名形式 “cache/<FQHN>” (FQHN は完全修飾されたホスト名) で、入力した IP アドレスがあらかじめ入力されています。
- ・ **[コメント]** – オプション。リモート・サーバの説明。

26.3 CSP サーバ・インスタンスの指定

複数の InterSystems IRIS インスタンスに接続するように 1 つの Web サーバを構成する場合は、接続する InterSystems IRIS インスタンス名を **[CSPサーバインスタンス]** フィールドに入力します。この名前は、すべて小文字で入力する必要があります。

例えば、IIS Web サーバをインストール済みの Windows マシンに InterSystems IRIS の 2 つのインスタンス (**iris1** と **iris2**) をインストールすると、既定の設定では、InterSystems IRIS の一部としてインストールされた固有プライベート Apache Web サーバで、それぞれのインスタンスを管理します。

ただし、各インスタンスのサーバ定義を作成するときに **[ウェブサーバポート]** (IIS の場合、既定の設定では 80) を変更し、**[CSPサーバインスタンス]** フィールドに **iris1** と **iris2** を指定することによって、両方のインスタンスをパブリック IIS Web サーバから管理することもできます。これによって、パブリック Web サーバ上に対応する InterSystems IRIS インスタンスを指す仮想ディレクトリ **/iris1** と **/iris2** が自動的に作成されます。

[CSPサーバインスタンス] フィールドに値を入力し、このサーバのランチャー上で管理ポータルを選択すると、URI は次のような形式になります。

```
http://<web srvr addr>:<web srvr port>/<csp srvr inst>/csp/sys/UtilHome.csp
```

これは、URI の **/csp/sys/UtilHome.csp** 部分の前にインスタンス名が置かれた形式で、例にある 2 つのインスタンスに対しては次の URI が生成されます。

```
http://localhost:80/iris1/csp/sys/UtilHome.csp
http://localhost:80/iris2/csp/sys/UtilHome.csp
```

リモート Web サーバが含まれる、より複雑な構成の場合は、“[Web ゲートウェイ構成ガイド](#)” の該当するプラットフォームのセクションを参照してください。

26.4 Telnet を介した接続

Telnet セッションから InterSystems IRIS のリモート・インスタンスへの接続も可能です。

1. InterSystems IRIS ランチャーをクリックし、**[リモート システム アクセス]** を選択します。
2. **[InterSystems IRIS Telnet]** をクリックしてリモート・サーバに接続し、ユーザ名とパスワードを入力して InterSystems IRIS システムにログインします。

PC で実行しているターミナルや Telnet の実行が可能なワークステーション・クライアントから、サポートされているプラットフォームの InterSystems IRIS インスタンスにリモートでログインできます。このクライアントはユーティリティのみを備えていれば十分であり、InterSystems IRIS サーバ・インスタンスはなくてもかまいません。一般的に、クライアント・マシン上の InterSystems IRIS のバージョンは、それを管理する InterSystems IRIS のバージョン以上である必要があります。

注釈 macOS プラットフォームでは、Windows のコマンド・プロンプトから SSH を使用して自分の macOS マシンに接続してから、以下のコマンドで接続します。

```
sudo /sbin/service telnet start
```

27

InterSystems IRIS インスタンスの管理 : iris コマンド

InterSystems IRIS® データ・プラットフォームの複数のインスタンスを 1 つのホスト・システムにインストールして実行できます。各インスタンスは、独立した一意の InterSystems IRIS 環境です。

27.1 インスタンスの管理オプション

指定されたシステムにインストールされたいくつかの InterSystems IRIS インスタンスのうちの 1 つに接続し、これを管理する方法は数多くあります。最も一般的な 2 つのメソッドを以下に示します。

- Windows ランチャー

Windows システム上にインストールされた各 InterSystems IRIS インスタンスは、システム・トレイに独自のランチャーを備え、以下の操作が可能となります。

- 管理ポータル、InterSystems ターミナル、およびスタジオ開発者クライアントを開くことで、インスタンスに接続します。
- インスタンスを開始、停止、および再起動します。
- ユーザおよび開発者のドキュメントを開きます。

ランチャーから、複数のリモート InterSystems IRIS インスタンスを管理することもできます。これらの管理には、リモート・バックアップの実行、構成設定の編集、リモートのオブジェクトおよびルーチンの作成やコンパイルなどが含まれますが、これらに限られるわけではありません。詳細は、“[リモート・サーバへの接続](#)”を参照してください。

- iris コマンド

オペレーティング・システムのコマンド行で iris コマンドを実行すると、InterSystems IRIS インスタンスへの管理アクセスが可能となり、以下の操作を行うことができます。

- InterSystems ターミナルを使用してインスタンスに接続します。
- インスタンスを開始、停止、および再起動します。
- インスタンスに関する情報、およびシステムにインストールされている他のインスタンスに関する情報を表示します。

iris コマンドの詳細は、“[InterSystems IRIS インスタンスの接続](#)” および “[InterSystems IRIS インスタンスの制御](#)”を参照してください。

リモート・サーバで iris コマンドを使用するには、Telnet または SSH クライアントを使用します。コンテナ化されたインスタンスでこれを使用するには、コンテナ内でこれを使用するか、docker exec コマンドを使用して、コンテナ外からこれを実行します。

27.2 インスタンスへの接続

ここでは、シェルの一つ、具体的には [ObjectScript シェル](#) を使用して、インスタンスに接続し、そのインスタンスにアクセスできるようにする方法を説明します。このシェルは、InterSystems IRIS インスタンスの任意のネームスペースで使用できます。ObjectScript シェルはターミナルと呼ばれることもありますが、ターミナルは実際にはこのシェルを提供する Windows アプリケーションです。実行中のインスタンスに対してこのシェルを開くには、コマンド `iris terminal instname` を使用します。instname は、インストール時にインスタンスに付けた名前です。コンテナ化されたインスタンスは、一般に **IRIS** という名前が付与されます。

注釈 インターシステムズのオンライン学習 Web サイトの [“Using The Terminal”](#) (ビデオ) を参照してください。

[事前定義のユーザ・アカウント](#)のいずれかとインストール時に指定したパスワードを使用するか、作成したアカウントを使用してログインします。表示されるプロンプトは、次のようなログイン・ネームスペースを示します。

```
# iris terminal IRIS
Node: intersystems2588, Instance: IRIS27
Username: admin
Password: *****
USER>
```

ターミナルを終了し、ウィンドウを閉じるには、コマンド `halt` を入力します。

docker exec コマンドを使用して、コンテナ化されたインスタンスのターミナルを開く場合 ([“InterSystems ターミナルを使用した操作”](#) の説明に従って)、自動的に irisowner としてログインされるため、認証は不要です。

Windows システムでは、InterSystems IRIS インスタンスの `install-dir¥bin` ディレクトリからこのコマンドを実行するか、例えば `c:¥InterSystems¥IRIS27¥bin¥iris terminal IRIS4` のようにコマンドにフル・パスを含める必要があります。指定されたインスタンスのバイナリを実行して、そのインスタンスまたは別のインスタンスに接続できます。どちらの場合もインスタンス名が必要です。

27.3 インスタンスの制御

iris コマンドは、terminal を超える多くの関数をサポートし、`iris function instname arguments` の形式で呼び出されます。ここで instname はインストール時に選択したインスタンス名、arguments は関数によって異なります。

重要 iris help コマンドによってコマンドのすべての関数と引数が表示されます。**IRISHelp.html** ファイルは `install-dir¥Help` ディレクトリにあります。iris コマンドの一部の関数は、このドキュメントに記載されていませんが、help ディスプレイには表示されます。

iris コマンドは、プラットフォームによって動作が異なります。これを次の表で説明します。

- [UNIX®, Linux、および MacOS での iris コマンド](#)
- [Windows での iris コマンド](#)

27.4 UNIX®, Linux、および MacOS での iris コマンド

注釈 iris コマンドは、メッセージ・ボックスにエラー情報を表示することがよくあります。iris コマンドへの最後の引数として `quietly` を追加することで、このメッセージ・ボックスを抑制できます。この引数は、ダイアログを最小限にして非対話式でコマンドを実行します。この引数は、コマンドの確認を行うことなくインスタンスをシャットダウンする場合などにも有用です。

iris all

インストールされているすべてのインスタンスについて、1 行につき 1 インスタンスで、[以下の説明](#)のとおり要約情報をリストします。

注釈 解析やレポートのために完全な情報が必要な場合は、`iris list` を使用します。

iris allw

それぞれのインスタンスに対して[同じ情報](#)を `iris all` としてリストしますが、長いフィールド値はラップしません。行の長さは 80 文字以上になる場合があります。

iris console instname [arguments]

[ObjectScript シェル](#)のコマンド・ウィンドウを開きます。

引数：`iris terminal` と同じ引数。

iris force の instname

インスタンスを強制終了します。

iris help [arguments]

iris コマンドの最新情報を表示します。

引数：`start`、`stop`、`force` — 起動、シャットダウン、および強制終了の各機能に関する固有のヘルプを表示します。

iris list [arguments]

インストールされている InterSystems IRIS インスタンスに関する情報を、[以下の説明](#)のとおりに表示します。

引数：`instname` — オプションで、InterSystems IRIS インスタンスの名前を指定し、そのインスタンスに関する情報のみを表示します。例えば、`iris list MyIRIS` は、MyIRIS インスタンスに関する情報のみを表示します。

iris mdx の instname

`##class(%DeepSee.Shell).%Go()` を実行することにより、ターミナルから DeepSee シェルに直接アクセスできます。

iris merge instname [arguments]

構成マージ・ファイルをインスタンスに適用し、その CPF を更新します (“[構成マージを使用した InterSystems IRIS の自動構成](#)” を参照)。

引数：[`merge-file`]、[`target-CPF`] — 必要に応じて、適用するマージ・ファイルの場所、ターゲットの CPF (インスタンスのアクティブな CPF) の場所、またはその両方を指定できます。例えば、`iris merge MyIRIS /tmp/merge.cpf /net/home/MyIRIS` は、マージ・ファイル `/tmp/merge.cpf` をインスタンス **MyIRIS** に適

用します。このインスタンスのアクティブな CPF は `/net/home/MyIRIS` にあります。マージ・ファイルまたはターゲット CPF を指定していない場合、環境変数が存在すればそれが使用されます。詳細は、“[構成マージを使用して既存のインスタンスを再構成する方法](#)”を参照してください。

iris python の instname

`##class(%SYS.Python).Shell()` を実行することにより、ターミナルから Python シェルに直接アクセスできます。

注釈 このコマンドが機能するためには、“[組み込み Python の使用法](#)”の説明に従って Python をインストールする必要があります。

iris qall

それぞれのインスタンスに対して[同じ情報](#)を `iris all` としてリストしますが、長い行は 78 文字に終了チルダ (^) 1 つを加えた長さに切り捨てられます。

iris qlist [arguments]

`iris list` と同様ですが、追加情報があります。各インスタンス ([以下で説明](#)) は、1 行でキャレット (^) で区切って出力されます。

引数 : `instname` — オプションで、InterSystems IRIS インスタンスの名前を指定し、そのインスタンスに関する情報のみを表示します。例えば、`iris qlist MyIRIS` は、MyIRIS インスタンスに関する情報のみを表示します。

iris rename instname newname

インスタンスの名前を変更します。

iris restart instname [arguments]

インスタンスを再起動します。`iris stop instname` と同等の機能です。restart

引数 : `nofailover` — オプションのこの引数を指定すると、ミラー・フェイルオーバーがトリガされません。

iris sql の instname

`##class(%SQL.Shell).%Go()` を実行することにより、ターミナルから SQL シェルに直接アクセスできます。

iris start instname [arguments]

インスタンスを開始します。

注釈 緊急モードでの起動を求めるメッセージが表示されることがあります。その場合は、“[緊急アクセス](#)”を参照してください。

引数 :

- ・ `full CPF path` — 既定で InterSystems IRIS は、`<install-dir>/mgr` ディレクトリにある `iris.cpf` ファイルから特定の設定を読み取ります。代わりに使用する別の `.cpf` ファイルへのフル・パスを指定することもできます。
- ・ `nostu` — `^STU` を実行せずに、指定したインスタンスを開始します。

iris stat instname

`irisstat` ユーティリティと同じシステムの統計を取得します (“[irisstat ユーティリティを使用した InterSystems IRIS の監視](#)”を参照)。

iris stop instname [arguments]

インスタンスをシャットダウンします。

引数：

- ・ restart — インスタンスのシャットダウン後にそのインスタンスを開始します。
- ・ nofailover — このオプションの引数を指定すると、ミラー・フェイルオーバーがトリガされません。
- ・ quietly — 最後の引数として、確認を必要とせずにインスタンスをシャットダウンします (他の iris コマンドを非対話式で実行する場合にも使用できます)。

iris stopnoshut instname [arguments]

INTNOSHUT^SHUTDOWN を実行して、ユーザのシャットダウン・ルーチンを実行せずに、指定したインスタンスをシャットダウンします。

注釈 インスタンス所有者と **irisusr** のみが、ターミナルにログインすることなく INTNOSHUT^SHUTDOWN を実行できます。

引数： nofailover — オプションのこの引数を指定すると、ミラー・フェイルオーバーがトリガされません。

iris terminal instname [arguments]

インスタンスの **ObjectScript シェル**を開きます。

引数：

- ・ -B — システム管理者の緊急ログインを有効にします (“[管理者ターミナル・セッション](#)”を参照)。
- ・ -b partition_size — プロセスの最大パーティション・サイズ (KB) を指定します。
- ・ “[label[+offset]]^routine” — ユーザ・モードで実行する ObjectScript プログラムの名前を指定します。指定の形式のほかに、文字列または数値リテラルで構成するパラメータ、および省略 (void) パラメータを以下のように渡すことができます。
 - "routine([parameter-list])"
 - "[label]^routine([parameter-list])"
 - "##CLASS(package.class).method([parameter-list])"

ここでは例えば、parameter-list を "string literal", ,--000123.45600E+07 の形式で指定し、省略パラメータは \$Data(parameter)=0 としてターゲットへ渡します。

注釈 空白やシェル・メタ文字は、オペレーティング・システムに依存する形式で引用する必要があります。

- ・ -U namespace — ログイン・ネームスペースを指定します。

注釈 **[開始ネームスペース]** が指定されているユーザ・アカウントで InterSystems IRIS を開始した場合は、-U 引数に効果はありません (“[ユーザ・アカウントのプロパティ](#)”を参照)。

27.5 Windows での iris コマンド

Windows では、install-dir\bin ディレクトリから iris コマンドを実行する必要があります (または、コマンドにフル・パスを含めます)。

iris all

インストールされているすべてのインスタンスについて、1 行につき 1 インスタンスで、[以下の説明](#)のとおり要約情報をリストします。

注釈 解析やレポートのために完全な情報が必要な場合は、iris list を使用します。

iris allw

それぞれのインスタンスに対して[同じ情報](#)を iris all としてリストしますが、長いフィールド値はラップしません。行の長さは 80 文字以上になる場合があります。

iris console instname [arguments]

[ObjectScript シェル](#)のコマンド・ウィンドウを開きます。

引数 : iris terminal と同じ引数。

iris force の instname

インスタンスを強制終了します。

iris help

iris コマンドの最新情報を表示します。

iris list [arguments]

インストールされている InterSystems IRIS インスタンスに関する情報を、[以下の説明](#)のとおりに表示します。

引数 : instname - オプションで、InterSystems IRIS インスタンスの名前を指定し、そのインスタンスに関する情報のみを表示します。例えば、iris list MyIRIS は、MyIRIS インスタンスに関する情報のみを表示します。

iris merge instname [arguments]

構成マージ・ファイルをインスタンスに適用し、その CPF を更新します ("[構成マージを使用した InterSystems IRIS の自動構成](#)" を参照)。

引数 : [merge-file]、[target-CPF] - 必要に応じて、適用するマージ・ファイルの場所、ターゲットの CPF (インスタンスのアクティブな CPF) の場所、またはその両方を指定できます。マージ・ファイルまたはターゲット CPF を指定していない場合、環境変数が存在すればそれが使用されます。詳細は、"[構成マージを使用して既存のインスタンスを再構成する方法](#)" を参照してください。

iris qlist [arguments]

iris list と同様ですが、追加情報があります。各インスタンス ([以下で説明](#)) は、1 行でキャレット (^) で区切って出力されます。

引数 : instname - オプションで、InterSystems IRIS インスタンスの名前を指定し、そのインスタンスに関する情報のみを表示します。例えば、iris qlist MyIRIS は、MyIRIS インスタンスに関する情報のみを表示します。

iris restart instname [arguments]

インスタンスのシャットダウン後にそのインスタンスを開始します。

引数：/nofailover — オプションのこの引数を指定すると、ミラー・フェイルオーバーがトリガされません。

iris run instname [arguments]

[\\$Principal](#) の入出力デバイスを使用せず、[プログラマ・モード](#)で InterSystems IRIS を実行します。

引数：iris terminal と同じ引数。

iris runw instname routine [arguments]

[\\$Principal](#) の入出力デバイスを使用せず、[アプリケーション・モード](#)で指定の InterSystems IRIS ルーチンを実行します。パッチ・スクリプトから実行した場合、コマンドは InterSystems IRIS プロセスが終了するのを待ってから、そのプロセスからの終了コードを返します。

引数：namespace — 指定されたネームスペースでルーチンを実行します。

注釈 **[開始ネームスペース]** が指定されているユーザ・アカウントで InterSystems IRIS を開始した場合は、namespace 引数に効果はありません ("[ユーザ・アカウントのプロパティ](#)" を参照)。

iris start instname [arguments]

インスタンスを開始します。

注釈 “緊急モード” での起動を求めるメッセージが表示されることがあります。その場合は、“暗号化ガイド” の “[緊急事態への対処](#)” を参照してください。

引数：full CPF path — 既定で InterSystems IRIS は、<install-dir>/mgr ディレクトリにある **iris.cpf** ファイルから特定の設定を読み取ります。代わりに使用する別の **.cpf** ファイルへのフル・パスを指定することもできます。

iris startnostu instname

^STU を実行せずに、指定したインスタンスを開始します。

iris stop instname [arguments]

インスタンスをシャットダウンします。

引数：/nofailover — オプションのこの引数を指定すると、ミラー・フェイルオーバーがトリガされません。

iris stopnoshut instname [arguments]

INTNOSHUT^SHUTDOWN を実行して、ユーザのシャットダウン・ルーチンを実行せずに、指定したインスタンスをシャットダウンします。

注釈 インスタンス所有者と **irisusr** のみが、ターミナルにログインすることなく INTNOSHUT^SHUTDOWN を実行できます。

引数：/nofailover — オプションのこの引数を指定すると、ミラー・フェイルオーバーがトリガされません。

iris stopstart instname [arguments]

インスタンスのシャットダウン後にそのインスタンスを開始します。

引数：/nofailover — オプションのこの引数を指定すると、ミラー・フェイルオーバーがトリガされません。

iris terminal instname [arguments]

インスタンスの InterSystems ターミナル (正式には [ObjectScript シェル](#)) を開きます。

引数 :

- ・ routine — [\\$Principal](#) のターミナルで、指定した InterSystems IRIS ルーチンを [アプリケーション・モード](#) で実行します
- ・ "[label[+offset]]^routine" — ユーザ・モードで実行する ObjectScript プログラムの名前を指定します。指定の形式のほかに、文字列または数値リテラルで構成するパラメータ、および省略 (void) パラメータを以下のように渡すことができます。

```
- "routine([parameter-list])"
- "[label]^routine([parameter-list])"
- "##CLASS(package.class).method([parameter-list])"
```

ここでは例えば、parameter-list を "string literal", , -+000123.45600E+07 の形式で指定し、省略パラメータは \$Data(parameter)=0 としてターゲットへ渡します。

注釈 空白やシェル・メタ文字は、オペレーティング・システムに依存する形式で引用する必要があります。

- ・ namespace — routine と共に使用され、指示されたネームスペースでルーチンを実行します。

注釈 **[開始ネームスペース]** が指定されているユーザ・アカウントで InterSystems IRIS を開始した場合は、namespace に効果はありません ("[ユーザ・アカウントのプロパティ](#)" を参照)。

27.6 iris list、qlist、および all

ここでは、一部の iris 関数に関する追加情報について説明します。

iris all

1 つ以上の InterSystems IRIS インスタンスに関する以下の情報をリストします。

- ・ インスタンスの状態 (以下の状態)
 - <blank> (状態利用不可、ログイン無効)
 - dn (ダウンまたはクラッシュ)
 - up (実行中)
 - st (開始または停止)
- ・ インスタンス名
- ・ InterSystems IRIS のバージョン
- ・ スーパーサーバのポート番号
- ・ インストール・ディレクトリ

iris list

1 つ以上の InterSystems IRIS インスタンスに関する以下の情報をリストします。

- ・ インスタンス名 (およびインストール・タイプ)
- ・ インストール・ディレクトリ
- ・ InterSystems IRIS のバージョン
- ・ InterSystems IRIS パラメータ (.cpf) ファイルのパス名
- ・ スーパーサーバおよび Web サーバのポート番号
- ・ インスタンスの状態 (以下の状態)
 - 実行
 - ダウン
 - 開始または停止
 - 開始または停止が不完全、ログイン無効
- ・ 実行中の場合、インスタンスのシステム・ヘルス状態 (“システム・モニタのヘルス状態” を参照) (Windows の場合は含まれません)
- ・ ミラー・メンバのタイプとステータス (ミラー・メンバの場合。%SYSTEM.Mirror.GetMemberType() および %SYSTEM.Mirror.GetMemberStatus() を参照)

iris qlist

1 つ以上の InterSystems IRIS インスタンスについて、以下の情報を 1 行にキャレット (^) で区切って出力します。

1. インスタンス名 (およびインストール・タイプ)
2. インストール・ディレクトリ
3. InterSystems IRIS のバージョン
4. インスタンスの状態
5. インストール・ディレクトリを基準にした、現在の構成パラメータ・ファイルのパス名。Windows システムでは、代わりにフル・パスが示されます。
6. スーパーサーバのポート番号
7. Web サーバのポート番号
8. JDBC ゲートウェイのポート番号
9. インスタンスのシステム・ヘルス状態 (実行中の場合。Windows の場合は常に空白)
10. 製品名
11. ミラー・メンバのタイプ (ミラー・メンバの場合)
12. ミラー・ステータス (ミラー・メンバの場合)
13. データ・ディレクトリ (該当する場合)

27.7 複数のインスタンスの構成

1 台のマシンに、InterSystems IRIS の複数のインスタンスをインストールし、同時に実行できます。単一のインストールとして InterSystems IRIS をインストールする場合は、各インスタンスに一意の名前、一意のインストール・ディレクトリ、およびスーパーサーバ、Web サーバ、および Telnet 用の一意のポート番号を指定します。

複数のインスタンスに関する特別な考慮事項は以下のとおりです。

- 複数のインスタンスをインストールする場合、1 つのシステムにつき 1 つのコンポーネントしかインストールできません。例えば、通常 Web サーバはシステムに 1 つに制限されるため、InterSystems IRIS をインストールすると、Web ゲートウェイが最新のインストールに対して構成されます。レジストリに格納されている InterSystems IRIS クライアント・コンポーネントについても、同様の問題が発生します。InterSystems IRIS では、その ODBC ドライバが、それぞれ 1 つの名前を使用してレジストリに格納されます。現在、これらのコンポーネントは、インストールされている最新のインスタンスを参照するように、最新のインストールによって更新されるようになっています。

インターシステムズは、InterSystems IRIS インスタンス全体で共有できるように、共通のコンポーネントを共通のディレクトリに移動する作業を進めています。

- 複数のインスタンスで、同じマルチサーバ・キーを共有できます。しかし、その場合は同じ 1 つのあるいは 1 組みのライセンス・サーバを使用する必要があります。1 つ以上のライセンス・サーバの下で InterSystems IRIS インスタンスを実行するそれぞれのシステムのローカルには、インスタンスごとに認証ライセンス・キー・ファイルのコピーがインストールされている必要があります。
- 複数のインスタンスはネットワーク接続できます。
- データベースは、同時に複数のインスタンスからは使用できないように保護されています (各インスタンスは個別にデータベースが必要です。他のインスタンスのデータベースにアクセスしたり修正することはできません)。
- 複数のインスタンスが構成されているときに、特定のインスタンスに要求をルーティングする場合は、追加の考慮事項があります。この詳細は、“[複数の InterSystems IRIS サーバ上のターゲット・アプリケーション](#)” を参照してください。
- 各インスタンスには一意のポート番号が必要です。[ポート番号の設定方法](#)の詳細は、次のセクションを参照してください。

27.7.1 ポート番号の設定

標準では、単一の InterSystems IRIS インスタンスのスーパーサーバ・ポート番号は既定で 1972 です。1 台のマシンに InterSystems IRIS のインスタンスを複数設定する場合、それぞれに一意のスーパーサーバ・ポート番号が必要です。自動的にポート番号を設定するようにした場合、後続のインスタンスにはインストール中に番号が割り当てられます。また、インストール中に手動でポート番号を入力することもできます。

InterSystems IRIS インストールの際にスーパーサーバ・ポート番号が割り当てられるため、たいいてい場合はこれらのポート番号を変更する必要はありません。スーパーサーバ・ポートの値は、インストール後に、管理ポータルの **[メモリと開始設定]** ページ (**[システム管理]**→**[構成]**→**[システム構成]**→**[メモリと開始設定]**) から変更できます。

各インスタンスに一意の Telnet ポート番号を割り当てる必要はありません。Telnet ポートの値は、インストール後に、管理ポータル **[開始設定]** ページ (**[システム管理]**→**[構成]**→**[追加の設定]**→**[開始]**) および **[Telnet設定]** ページ (**[システム管理]**→**[構成]**→**[デバイス設定]**→**[Telnet設定]**) から変更できます。

A

インターシステムズ製品と関係して動作するようにサードパーティ・ソフトウェアを構成する方法

インターシステムズ以外のプロバイダの製品も存在する環境でインターシステムズ製品を使用することは少なくありません。このような環境では、インターシステムズ製品とそれ以外のツールの相互作用によって好ましくない影響が発生することがあります。信頼性に優れた最適な導入構成に関するインターシステムズの指針では、サードパーティのツールからの干渉を受けることなく、インターシステムズ製品を導入できることを推定しています。例えば、インターシステムズは、セキュリティ、システム監視、またはウイルス・スキャン用のソフトウェアがインターシステムズ製品のインストール、パフォーマンス、および機能に影響を及ぼす可能性があることを確認しています。これは、ウイルス・スキャンなどのツールに特に当てはまることです。このようなツールは、インターシステムズ製品の一部であるファイル（または製品で使用するファイル）に直接作用します。

インターシステムズでは、ビジネス上やコンプライアンス上の要件をはじめとするさまざまな要件にお客様が取り組んでいることを理解しています。このような要件の中には、所定の環境でどのソフトウェアを実行し、それをどのように構成するかといった意思決定に影響するものがあります。

ウイルス・スキャンについての考慮事項：

1. ウイルスチェック済みのソフトウェアを提供するために、インターシステムズの製品はセキュリティで保護されたクリーンな環境からお客様に届けられ、検証用のチェックサムも備えています。
2. インターシステムズ製品をインストールするときや実行するときに、ウイルス・スキャンによって誤検出が報告されることがあります。インターシステムズでは、ウイルス・スキャンをはじめとするサードパーティ製セキュリティ製品による検出結果を管理できません。ドキュメントの詳細やガイダンスは、使用されている各製品のベンダーにお問い合わせください。
3. ウイルス・スキャンによって、インターシステムズのプロセス上でウイルス・パターンが誤検出され、その対応としてファイルの隔離や削除などの措置が取られることが考えられます。このような措置が深刻な結果につながる場合があります。特に、インスタンス実行中の影響は甚大です。データベースやジャーナルなど、短時間で頻繁に変更されるファイルでは、このようなウイルス・パターンが検出される可能性が高くなります。
4. データベースなどの大規模なファイルに対するほぼ定常的なスキャンやきわめて高い頻度でのスキャンは、インターシステムズ製品の総合的なパフォーマンスに少なからぬ影響が発生する原因になりやすいといえます。
5. 独自のリスク評価に従って、以下のファイルとディレクトリのスキャンを調整することが必要な場合もあります。
 - ・ WIJ ファイルおよび WIJ ファイルを収めたディレクトリ。このディレクトリは構成設定に応じて異なります。“[ライト・イメージ・ジャーナリングとリカバリ](#)”のページを参照してください。
 - ・ すべてのデータベース・ファイル(.DAT) およびデータベース・ファイルを収めたディレクトリ。これらは構成設定に応じて異なります。“[データベースの構成](#)”のページを参照してください。

- ジャーナル・ファイルが格納または処理される任意のディレクトリ。このディレクトリは構成設定に応じて異なります。“[ジャーナリング](#)”のページを参照してください。
- ・ InterSystems IRIS® データ・プラットフォームが機能するために頻繁に要求される、その他すべてのファイルまたはディレクトリ。これらも構成設定に応じて異なります。この例として、代替ジャーナル・ディレクトリ (“[ジャーナリング](#)”のページを参照)、ビジネス・サービスやプロダクションで使用しているあらゆるディレクトリなどがあります。

ウイルス・スキャンからファイルを長期間または永久的に除外すると、“感染”のリスクが高くなることが考えられます。

警告

マルウェア・スキャンから項目を除外すると、保護されたデバイスおよびアプリケーションに脆弱性を招き入れる可能性があります。クライアントは、マルウェア対策の構成に関わるすべての責任を負います。

B

機能トラッカによる使用量統計の収集

顧客のニーズに合わせて製品を改善するというインターシステムズの持続的取り組みの一環として、インターシステムズは InterSystems IRIS® データ・プラットフォームに機能トラッカというソフトウェア・ユーティリティを組み込んでいます。これは、ソフトウェア・モジュールの使用量に関する統計を収集するユーティリティです。このトピックでは機能トラッカについて説明します。

B.1 統計を収集する理由

機能トラッカによって収集される統計は、特定の InterSystems IRIS インスタンスにソフトウェア・モジュールが存在し、使用されているかどうかを記録します。機能トラッカは、毎週この情報を https 経由でインターシステムズに送信します。これらの統計データは、開発およびサポート計画の参考にされます。

収集される情報にはアプリケーション・データは一切含まれていません。

B.2 ログの場所

InterSystems IRIS はデータ送信を正常に完了すると、<installdir>\mgr ディレクトリ内の **FeatureTracker.log** ファイルを更新します。このファイルには、送信されたデータの JSON 形式のコピーが含まれています。各エントリの形式は、以下のとおりです。

```
"FT.<feature>":<value>,
```

InterSystems IRIS は、データを送信できない場合、送信を数回試行します。それでも送信できない場合、InterSystems IRIS は試行を停止し、次の定期予定時間に再試行します。

InterSystems IRIS は、試行の成功および失敗のいずれの場合もメッセージ・ログ・ファイル (**messages.log**) を更新します。送信成功を示すエントリの形式は、以下のとおりです。

```
mm/dd/yy-hh:mm:ss 0 %SYS.Task.FeatureTracker transferred data to ats.intersystems.com
```

送信失敗を示すエントリの形式は、以下のとおりです。

```
mm/dd/yy-hh:mm:ss 1 %SYS.Task.FeatureTracker failed to transfer data
```

送信されたデータには、エンコード済みライセンス・キーとホスト名、およびインスタンス GUID が含まれています。

B.3 収集される統計データ

機能トラッカは以下のソフトウェア機能が有効化されているかどうかをログに記録します。

- ・ BitTrakCare
- ・ BI ユーザ (ランタイム) および BI 開発
- ・ C-タイプ・ライセンス
- ・ ExtremeNoUserLimit
- ・ Healthshare、Healthshare Foundation
- ・ IPNeutral
- ・ Web ライセンス

機能トラッカは以下の情報についてもログに記録します。

- ・ データベース (DB) 暗号化:これが有効であるかどうか、およびスナップショット時に利用可能なマウント済み暗号化データベースの数
- ・ ミラーリング:このインスタンスがミラーに接続されているかどうか、フェイルオーバー・メンバの数、非同期メンバの数とタイプ、および各非同期メンバがメンバとなっているミラー・セットの数
- ・ システム:承認、InterSystems IRIS バージョン、オペレーティング・システム、ホスト名、インスタンス名、インスタンス ID、および注文 ID

B.4 機能トラッカを非アクティブ化する方法

機能トラッカは、既定で有効化されています。この機能を非アクティブ化すると、インターシステムズへのデータの送信が停止します (ただし、データは引き続き収集します)。機能トラッカを非アクティブ化するには、以下に示すようにタスク・マネージャを使用します。

1. 管理ポータルで、[システム操作]→[タスクマネージャ]→[タスクスケジュール] を選択します。
2. 機能トラッカの行を見つけて、[機能トラッカ] を選択します。
3. タスク詳細のページが表示されたら、[一時停止] を選択します。
4. [?] という質問に適切に回答します。
5. [このアクションをすぐに実行する] を選択して、タスクを停止します。

InterSystems IRIS をアップグレードしても、機能トラッカの状態は維持されます。アップグレード前にタスクがスケジュールされていた場合、そのタスクはスケジュールされたままになります。また、タスクが一時停止されていた場合、そのタスクは一時停止されたままになります。