



Web ゲートウェイ・ガイド

Version 2024.1
2024-06-03

Web ゲートウェイ・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動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 Web ゲートウェイ：InterSystems IRIS Web アプリケーションと REST API を Web クライアントに提供	1
1.1 Web ゲートウェイが InterSystems Web アプリケーション要求をルーティングする方法	2
1.1.1 InterSystems Web アプリケーション URL の構造	3
1.2 システムの Web ゲートウェイの設定	3
1.3 Web ゲートウェイ接続の管理	4
2 Web サーバを使用した管理ポータルおよび他の組み込みの Web アプリケーションへのアクセス	5
2.1 管理ポータルの URL	5
2.2 新規インストールの場合	6
2.3 アップグレードの場合	6
2.4 Web サーバの自動接続	7
2.4.1 自動構成が可能な場合	7
2.4.2 自動構成動作	8
2.5 Web サーバの手動接続	12
2.5.1 必要なアプリケーション・パスの指定	12
2.5.2 各パスに必要なすべての要求のルーティング	12
2.5.3 ドキュメント・リンクのリダイレクト	13
2.5.4 Windows のみ：InterSystems IRIS サーバ・マネージャの更新	14
2.5.5 Windows のみ：VS Code を有効にするように IIS を構成	14
2.6 2023.2 より前のバージョンからのアップグレード：プライベート Web サーバの無効化および削除	15
3 概要：システムの Web ゲートウェイの設定	17
3.1 Web ゲートウェイ・ファイルのインストール	17
3.1.1 InterSystems IRIS インストールの一部としてのインストール	18
3.1.2 スタンドアロン・コンポーネントとしてのインストール	18
3.1.3 webgateway コンテナの導入	19
3.2 Web ゲートウェイでの Web サーバの機能拡張	19
3.2.1 Web サーバ構成への Web ゲートウェイの追加	19
3.2.2 Web サーバが Web ゲートウェイを介してルーティングする要求の指定	21
3.3 Web ゲートウェイから InterSystems IRIS インスタンスへの要求の送信	23
3.3.1 InterSystems IRIS インスタンスの Web ゲートウェイへの接続	23
3.3.2 インスタンスのアプリケーションとの関連付け	23
3.4 すべての接続の保護	24
3.5 Web ゲートウェイ接続のデコミッション	24
4 スタンドアロン Web ゲートウェイのインストール	27
4.1 手順 1：サポート対象 Web サーバのインストール	27
4.2 手順 2：インストール・キットのダウンロード	28
4.3 手順 3：(UNIX®/Linux/macOS のみ) インストール・キット・ファイルの抽出	28
4.4 手順 4：(UNIX®/Linux/macOS のみ) root としてのログイン	29
4.5 手順 5：インストーラの実行	29
4.5.1 UNIX®/Linux/macOS	29
4.5.2 Windows	30
5 Web ゲートウェイでの Web サーバ構成の拡張	33
5.1 考慮すべきファイル	33

5.1.1 Web ゲートウェイ・ファイル	33
5.1.2 Web サーバ・ファイル	34
5.1.3 静的ファイル	34
5.2 UNIX®/Linux/macOS 向けの Apache	35
5.2.1 Apache が共有オブジェクト・モジュールを管理できることの確認	35
5.2.2 Web サーバ構成への Web ゲートウェイ・モジュールの追加	36
5.3 Windows 向けの Microsoft インターネット・インフォメーション・サービス (IIS)	37
5.3.1 IIS の有効化	37
5.3.2 Web ゲートウェイ・コンポーネントに対する許可の設定	38
5.3.3 ネイティブ・モジュールの登録	38
5.3.4 Web アプリケーション・パスの構成	39
5.3.5 アプリケーション要求のハンドラ・マッピングの設定	40
5.3.6 /bin を含む URLの有効化	41
5.3.7 リモート Web サーバ接続のランチャーの構成	42
5.3.8 SOAP フォルトの詳細を返すための IIS の構成	42
5.3.9 IIS の再起動	42
5.3.10 トラブルシューティング	43
5.4 UNIX®/Linux/macOS 向けの Nginx	44
5.4.1 概要	44
5.4.2 想定	44
5.4.3 インストール	44
5.4.4 CSP のための Nginx Web サーバの構築	45
5.4.5 Nginx での NSD の使用	46
5.4.6 IDE の使用に必要な追加の構成	48
5.4.7 Nginx と NSD の起動および停止	48
5.4.8 非推奨：ユニバーサル・モジュールを使用する Nginx の構築	48
5.5 Windows 向け Nginx	49
5.5.1 概要	49
5.5.2 想定	49
5.5.3 インストール	50
5.5.4 CSP のための Nginx Web サーバの構築	50
5.5.5 NSD を呼び出すように Nginx を構成	52
5.5.6 Nginx と NSD の起動および停止	54
5.5.7 VS Code ユーザの場合：必要な追加構成	54
5.5.8 非推奨：ユニバーサル・モジュールを使用する Nginx の構築	55
6 Web ゲートウェイを介して要求をルーティングする URL パスの選択	57
6.1 Web サーバから	57
6.2 Web ゲートウェイ経由	58
6.3 InterSystems IRIS アプリケーション・サーバへ	58
6.3.1 複数の InterSystems IRIS サーバ上のターゲット・アプリケーション	59
6.3.2 カスタムのインスタンス接頭語を使用した各 InterSystems IRIS サーバのアドレス指定	60
6.3.3 Apache 仮想ホストの構成	61
7 Web ゲートウェイ管理ページの概要	63
7.1 Web ゲートウェイ管理ページへのアクセス	63
7.2 追加クライアント・アドレスからのアクセスの有効化	63
7.3 使用可能なオプション	64
7.4 ローカリゼーション	65
8 InterSystems IRIS インスタンスのサーバ・アクセス・プロファイルの定義	67

8.1 サーバ・アクセス・プロファイルの追加	67
8.1.1 サーバ・アクセス・パラメータ	68
8.1.2 状態なしパラメータ	68
8.1.3 接続セキュリティのパラメータ	69
8.1.4 SSL/TLS パラメータ	69
8.1.5 [オプションパラメータ]	72
8.1.6 [エラーページ]	72
8.2 サーバ・アクセス・プロファイルのコピー	72
8.3 InterSystems IRIS サーバへのアクセスの無効化	72
8.4 サーバ・アクセス・プロファイルの削除	73
9 Web アプリケーション・パスのアプリケーション・アクセス・プロファイルの定義	75
9.1 アプリケーション・アクセス・プロファイルの追加	75
9.1.1 アプリケーション・アクセス・プロファイルの構成パラメータ	76
9.1.2 サーバ・パラメータ	77
9.2 アプリケーション・アクセス・プロファイルのコピー	78
9.3 アプリケーション・パスによるアクセスの無効化	78
9.4 アプリケーション・アクセス・プロファイルの削除	79
10 Web ゲートウェイのシステム全体のパラメータの構成	81
10.1 Web ゲートウェイ・パラメータの構成方法	81
10.2 Web ゲートウェイ (一般設定)	82
10.3 セキュリティ	83
10.4 InterSystems IRIS への接続	84
10.5 ASP リダイレクト	87
10.6 [内部HTTPサーバ]	88
10.7 カスタム・エラー・ページ	88
10.8 イベント・ログ・パラメータ	89
11 InterSystems IRIS への Web ゲートウェイ接続の保護	97
11.1 Web ゲートウェイの接続セキュリティの構成	97
11.2 最小の接続セキュリティ (非推奨)	98
11.3 単純なユーザ名/パスワード認証	98
11.3.1 外部から導入されるパスワード	99
11.3.2 別のコンピュータで暗号化されたパスワード	99
11.3.3 プログラムによるパスワードの取得 (UNIX®/Linux/macOS)	100
11.4 Kerberos ベース認証とデータ保護	100
11.4.1 Kerberos ライブラリ	100
11.4.2 Windows	102
11.4.3 Kerberos のための UNIX® Web ゲートウェイ構成	102
11.5 SSL/TLS ベース認証とデータの保護	103
11.5.1 相互 TLS	103
12 Web ゲートウェイの管理と監視	105
12.1 システム・ステータスの確認	105
12.1.1 InterSystems IRIS への接続	105
12.1.2 InterSystems IRIS サーバ・テーブル	106
12.1.3 アプリケーション・パス・テーブル	106
12.1.4 Web ゲートウェイ・キャッシュ・テーブル	107
12.1.5 手動による接続の切断	107
12.1.6 キャッシュのクリア	107
12.2 サーバ接続のテスト	108
12.3 イベント・ログの表示	108

12.4 HTTP トレース機能の使用法	109
13 Web ゲートウェイによって渡される CGI 環境変数	111
14 Web ゲートウェイにより返される HTTP 応答ヘッダ	113
15 CSP フォームの要求に対する応答の圧縮 (GZIP/ZLIB)	115
15.1 GZIP/ZLIB ライブラリ	116
15.2 GZIP/ZLIB ライブラリの使用法	116
15.3 ページごとの圧縮の指定	117
15.4 アプリケーション・パス内のすべてのページに対する圧縮の指定	117
15.5 監視	118
16 Web アプリケーション向けの HTTP 認証の実装	119
16.1 Apache での標準 HTTP 認証 (mod_auth)	119
16.2 要求の処理と同時に CSP で行われる認証	120
16.3 要求の処理前に CSP で行われる認証	121
17 負荷分散、フェイルオーバー、ミラー構成	123
17.1 複数の Web サーバ間の負荷分散とフェイルオーバー	123
17.2 複数の InterSystems IRIS サーバ・インスタンス間の負荷分散とフェイルオーバー	123
17.3 ミラー構成	124
18 プロセス親和性とステート認識モード (保持モード 1)	127
18.1 ステート認識モードの起動	128
18.2 ステート認識モードの維持とエラーへの応答	129
18.3 ステート認識モードの終了	130
19 InterSystems IRIS の Web ゲートウェイ・レジストリ	131
19.1 構成を再ロードするよう Web ゲートウェイを強制する	132
19.1.1 InterSystems IRIS Web ゲートウェイ・レジストリの使用	132
19.1.2 InterSystems IRIS の外部スクリプトの使用	132
Web ゲートウェイ構成ファイル (CSP.ini) パラメータ・リファレンス	135
[SYSTEM]	136
[<server>]	140
[SYSTEM_INDEX]	143
[APP_PATH:<appPath>]	144
[APP_PATH_INDEX]	147
付録A: NSD の使用法 (Windows)	149
A.1 NSD を使用する場合	149
A.2 NSD モジュールのインストール場所	149
A.3 NSD の操作	150
A.3.1 代替 TCP ポートでの NSD の開始	150
付録B: NSD の使用法 (UNIX®/Linux/macOS)	153
B.1 NSD を使用する場合	153
B.2 NSD モジュールのインストール場所	153
B.3 NSD の操作	153
B.3.1 代替 TCP ポートでの NSD の開始	155
付録C: Apache の代替オプション (UNIX®/Linux/macOS)	159
C.1 インストール場所 (すべての特殊オプション)	159
C.1.1 Apache API モジュールを使用する場合の要件 (推奨オプションおよび代替オプション 1)	160
C.2 代替オプション 1 : NSD を使用した Apache API モジュール (mod_csp24.so)	161

C.2.1 方法 1 : apxs (APache eXtenSion) ツールを使用した、共有オブジェクトとしての CSP モジュールのビルド	161
C.2.2 方法 2 : 共有オブジェクトとしての CSP モジュールの手動ビルド	161
C.2.3 実行時の構成	162
C.3 代替オプション 2 : NSD を使用した CGI モジュール (nph-CSPcgi)	163
C.3.1 CGI および NSD を使用した Web ゲートウェイの運用および管理	165
C.4 代替オプション 3 : NSD を使用した組み込み Apache API モジュール (mod_csp.c)	165
C.4.1 CSP モジュール・ソース・コードを含めた Apache のビルド	165
C.4.2 生成した Apache バイナリの確認	165
C.4.3 実行時の構成	166
C.4.4 Apache API および NSD を使用した Web ゲートウェイの運用および管理	166
付録D: ロック・ダウン Apache インストールへの Web ゲートウェイの追加 (UNIX®/Linux/macOS) ..	167
D.1 Web ゲートウェイ・ファイルのセキュリティ・コンテキストの変更	167
D.2 Web ゲートウェイ・ディレクトリの移動	168
D.2.1 推奨オプション : Apache API モジュール (CSPa24.so)	169
D.2.2 代替オプション 1 : NSD を使用した Apache API モジュール (mod_csp.so)	169
D.2.3 代替オプション 2 : NSD を使用した CGI モジュール (nph-CSPcgi)	169
D.2.4 代替オプション 3 : NSD を使用した組み込み Apache API モジュール (mod_csp.c)	169
付録E: IIS 7 以降の代替オプション (Windows)	171
E.1 ISAPI サービスおよび CGI サービスのインストール	171
E.2 代替オプション 1 : ISAPI モジュールの使用 (CSPms*.dll)	172
E.2.1 ISAPI 拡張機能の有効化	172
E.2.2 InterSystems IRIS ファイル拡張子のマッピング	173
E.2.3 Web ゲートウェイの運用および管理	174
E.3 代替オプション 2 : NSD を使用したネイティブ・モジュールの使用 (CSPcms.dll)	175
E.3.1 ランタイム・ネイティブ・モジュールの登録	175
E.3.2 Web ゲートウェイ管理のための CGI モジュールの有効化	175
E.3.3 InterSystems IRIS ファイル拡張子のマッピング	176
E.3.4 Web ゲートウェイの運用および管理	178
E.4 代替オプション 3 : NSD を使用した ISAPI モジュールの使用 (CSPcms.dll)	178
E.4.1 ランタイム ISAPI 拡張機能の有効化	178
E.4.2 Web ゲートウェイ管理のための CGI モジュールの有効化	179
E.4.3 InterSystems IRIS ファイル拡張子のマッピング	179
E.4.4 Web ゲートウェイの運用および管理	181
E.5 代替オプション 4 : NSD を使用した CGI モジュールの使用 (nph-CSPcgi*.exe)	181
E.5.1 CGI のモジュールの有効化	181
E.5.2 InterSystems IRIS ファイル拡張子のマッピング	182
E.5.3 Web ゲートウェイの運用および管理	183

1

Web ゲートウェイ : InterSystems IRIS Web アプリケーションと REST API を Web クライアントに提供

InterSystems IRIS® [Web アプリケーション](#)は、要求に応じて動的に Web クライアント (通常は Web ブラウザ) にコンテンツを提供するコードで構成されます。これを可能にするのが InterSystems Web ゲートウェイです。InterSystems Web ゲートウェイは、Web サーバと、Web アプリケーション・コードをホストする InterSystems IRIS インスタンス間の接続を仲介するソフトウェア・ユーティリティです。Web ゲートウェイは HTTP、[HTTPS](#)、[WebSocket](#) プロトコルをサポートしており、アプリケーション・トラフィックの[負荷分散](#)や[フェイルオーバー](#)などの機能を提供します。

Web ゲートウェイの機能は以下のとおりです。

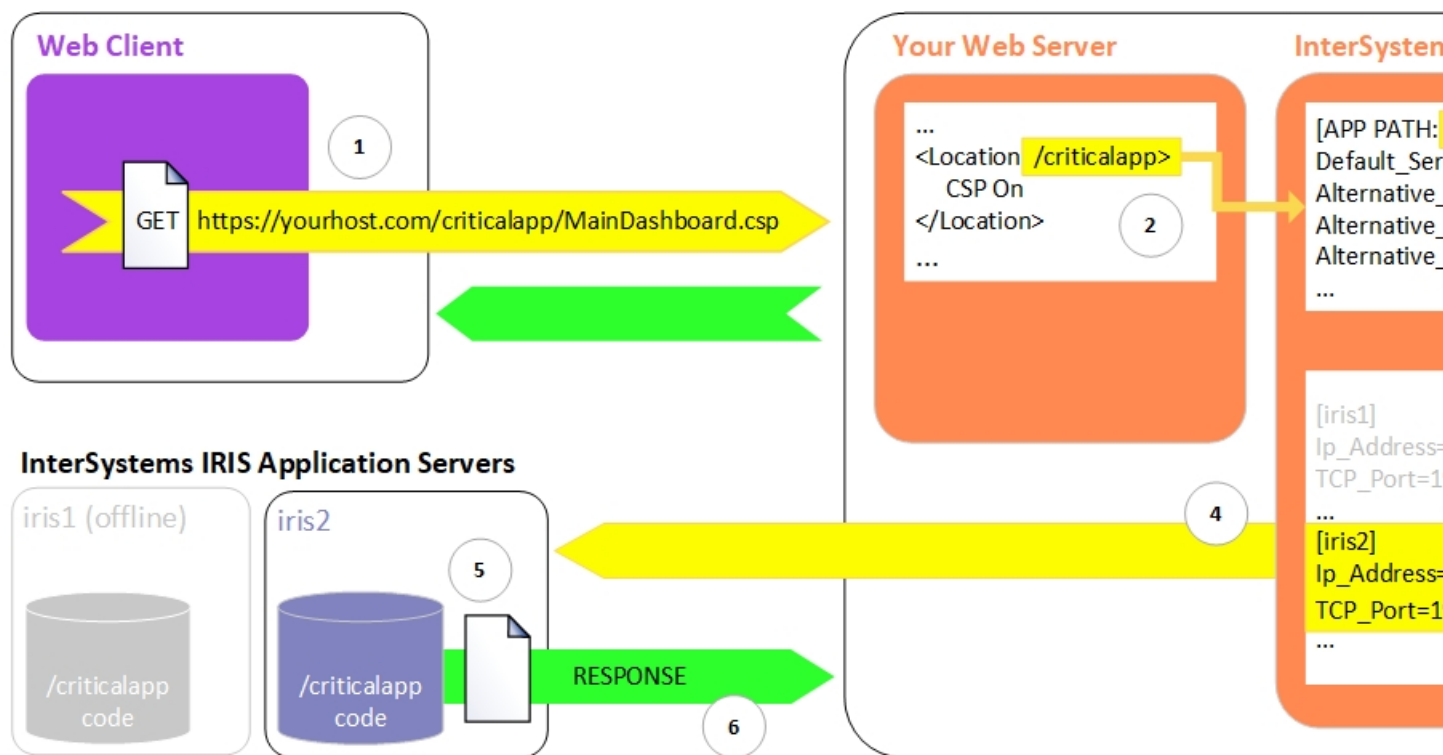
- ・ Web サーバの機能を拡張し、Web サーバが InterSystems Web アプリケーションの要求を認識して処理できるようにします。
- ・ 各 InterSystems IRIS インスタンス内のアプリケーション・サーバ・プロセスと接続するための[サーバ・アクセス・プロファイル](#)を管理します。
- ・ 特定の要求パスと特定の InterSystems IRIS アプリケーション・サーバ間の[アプリケーション・アクセス・プロファイル](#)に基づいて、アプリケーションをホストする InterSystems IRIS インスタンスにアプリケーションの要求をルーティングします。

注釈 受信要求への応答としてアプリケーション・コードを呼び出す InterSystems IRIS インスタンス内のプロセスは、従来の InterSystems Web アプリケーション・フレームワークにちなんで、CSP サーバまたは CSP エンジンと呼ばれることがあります。ただし、このプロセスは CSP アプリケーションだけでなく、すべての InterSystems IRIS Web アプリケーションの要求に応答します。そのため、Web ゲートウェイのドキュメントでは、特にこのプロセスを指す必要がある場合は、“アプリケーション・サーバ”という用語を使用します。その他の場合、ドキュメントでは InterSystems IRIS インスタンス全体を Web ゲートウェイ接続の有効なターゲットとして扱います。

Web アプリケーションは、管理ポータルなどいくつかの重要な InterSystems IRIS システム・ユーティリティへのアクセスを提供します。これらを使用するには、Web ゲートウェイを介して InterSystems IRIS を Web サーバに接続する必要があります。“[Web サーバを使用した管理ポータルおよび他の組み込みの Web アプリケーションへのアクセス](#)”を参照してください。

1.1 Web ゲートウェイが InterSystems Web アプリケーション要求をルーティングする方法

以下の図は、Web アプリケーションの要求がクライアント (Web ブラウザなど) から Web アプリケーションが実行されている InterSystems IRIS インスタンスに送られる様子を示しています。



プロセスは、以下のように要約できます。

- クライアントは特定の URL パスに要求を送信します。この例では、クライアントは `https://yourhost.com/criticalapp/MainDashboard.csp` に HTTP GET 要求を送信します。
- Web サーバは要求を受け取って構成を確認し、その要求に対して Web ゲートウェイを呼び出すべきかどうかを判断します。図では、架空の Apache Web サーバの構成ファイルからの抜粋でこれを示しています。この例では、Web サーバ構成は、Web ゲートウェイを呼び出して `/criticalapp` パス内の要求を処理するよう Web サーバに指示しています。
- Web ゲートウェイは構成を確認して、どの InterSystems IRIS アプリケーション・サーバが要求を受け取るかを決定します。図では、Web ゲートウェイの架空の `CSP.ini` 構成ファイルからの抜粋でこれを示しています。この例では、`/criticalapp` パスのアプリケーション・アクセス・プロファイルは、そのパス内の要求を、既定では InterSystems IRIS インスタンス `iris1` のアプリケーション・サーバ・プロセスに、`iris1` を使用できない場合はインスタンス `iris2` の代替アプリケーション・サーバにルーティングするよう、Web ゲートウェイに指示しています。
- Web ゲートウェイは、要求を保護された TCP 接続を介して該当するアプリケーション・サーバのホストとスーパーサーバ・ポートに転送します。図に示すように、Web ゲートウェイは、これらの接続詳細を各アプリケーション・サーバのサーバ・アクセス・プロファイルに格納します。この例では、`iris1` は現在オフラインであるため、Web ゲートウェイは要求を代替サーバ `iris2` に送信します。

- InterSystems IRIS インスタンスのアプリケーション・サーバ・プロセスは、InterSystems IRIS インスタンス内の Web アプリケーションの設定に基づいて、適切なアプリケーション・コードを呼び出します。アプリケーション・コードは応答を生成します。
- 応答が Web ゲートウェイを介して Web サーバに戻されます。Web サーバは応答をクライアントに転送します。

1.1.1 InterSystems Web アプリケーション URL の構造

Web ゲートウェイでは、以下の形式の URL で InterSystems IRIS Web アプリケーションを提供できます。

[protocol]://[hostname]:[port]/[instancePrefix]/[appPath]/[fileOrQuery]

URL 要素	説明
[protocol]	http または https。TLS を使用するように Web サーバを構成したかどうかによって異なります。TLS の使用を強くお勧めします。
[hostname]	Web サーバが使用可能な IP アドレスまたは DNS 名。Web サーバがクライアントと同じマシン上にある場合、これは通常 localhost です。
[port]	Web サーバが要求を待ち受けるポート番号。Web サーバが HTTP の標準ポート (80) または HTTPS の標準ポート (443) 以外のポートで要求を待ち受ける場合を除き、ポート番号を指定する必要はありません。
[instancePrefix]	<p>InterSystems IRIS アプリケーション・サーバの 1 つを一意に識別する文字列。</p> <p>複数の InterSystems IRIS インスタンスを区別する必要がある場合は、前述の例のように、パスのこの部分を省略した URL で Web アプリケーションを提供するように Web ゲートウェイを構成できます。ただし、1 つの Web サーバを使用して複数の InterSystems IRIS インスタンスを処理し、1 つのインスタンスに固有の Web アプリケーション (インスタンスの管理ポータルなど) にアクセスする必要がある場合は、この識別子をアプリケーション・パスの接頭語として使用して要求をインスタンスにルーティングするように Web サーバと Web ゲートウェイを構成する必要があります。この識別子は、インスタンスの CSPConfigName です。既定では、インスタンスの CSPConfigName はすべて小文字のインスタンス名です。ただし、構成可能です。</p> <p>前述の例のアプリケーションを、iris2 という InterSystems IRIS アプリケーション・サーバで排他的に使用できる場合、Web サーバと Web ゲートウェイの構成内で、次の URL でアプリケーションを提供するように /iris2 パスを構成できます：</p> <p>https://yourhost.com/iris2/criticalapp/MainDashboard.csp</p>
[appPath]	各 InterSystems IRIS アプリケーション・サーバ内のアプリケーションに固有の相対パス。InterSystems IRIS アプリケーション・サーバ内で アプリケーションを定義 する場合、これが アプリケーションの名前 になります。
[fileOrQuery]	オプション。アプリケーションが要求の処理に使用できる下位のパス、ファイル名、クエリ・パラメータの任意の組み合わせ。

1.2 システムの Web ゲートウェイの設定

Web ゲートウェイ接続は複数の方法で設定できます。前述の図で示すように、Web ゲートウェイは、クライアントおよび InterSystems IRIS インスタンスから離れた別個のマシンに Web サーバと共に導入できます。または、Web クライアント、

Web サーバ、Web ゲートウェイをすべて同じマシンに配置することもできます。Web ゲートウェイはオンプレミスでインストールできます。または、[InterSystems Container Registry](#) で入手できる Docker イメージから [webgateway コンテナ](#) を導入することもできます。これらの [webgateway コンテナ・イメージ](#) には、Web ゲートウェイ拡張で事前構成されている Web サーバ (Apache または nginx) が含まれます。

“[システムの Web ゲートウェイの設定](#)” では、ユース・ケースに関係なく、設定プロセス全体の概要を説明します。

1.3 Web ゲートウェイ接続の管理

Web ゲートウェイは、どのように導入されたかに関係なく、Web クライアントと Web アプリケーションをホストする InterSystems IRIS インスタンスとの間の接続を管理する標準的なインタフェースのセットを提供します。

[Web ゲートウェイ管理ページ](#) には、既定のパラメータの構成、[InterSystems IRIS インスタンスへの接続の定義](#)、それらの接続の監視およびテスト、アプリケーションのルーティング動作の定義などを行うためのグラフィカル・ユーザ・インタフェースが用意されています。Web ゲートウェイ構成情報は、[CSP.ini](#) に格納されます。ほとんどの場合、Web ゲートウェイ管理ページまたは [Web ゲートウェイ・レジストリ・メソッド](#) を使用して Web ゲートウェイを構成することをお勧めします。ただし、コンテナ化された導入の場合は、[CSP.ini マージ機能](#) を使用して [CSP.ini](#) ファイルを直接変更することで、導入時に 1 つ以上の [webgateway コンテナ](#) の構成を自動化し、同期することができます。

InterSystems IRIS [Web ゲートウェイ・レジストリ](#) により、InterSystems IRIS インスタンスは Web ゲートウェイへの接続をプログラムによって監視および管理することができます。すべての Web サーバおよび Web ゲートウェイのインストールは、接続時に InterSystems IRIS に登録されます。ゲートウェイ・レジストリを使用することで、InterSystems IRIS コードはインスタンスが接続されている Web ゲートウェイとやり取りし、構成詳細の読み取りおよび書き込みを行い、システム・ステータスを監視し、[Web ゲートウェイのイベント・ログ](#) を監査できます。

2

Web サーバを使用した管理ポータルおよび他の組み込みの Web アプリケーションへのアクセス

InterSystems IRIS® は、[組み込みの Web アプリケーション](#)を介して、管理ポータルなどいくつかの重要なシステム・ユーティリティへのアクセスを提供します。これらの Web ベースのユーティリティを使用するには、[Web ゲートウェイ](#)を介して InterSystems IRIS を Web サーバに接続する必要があります。

[多くの場合](#)、InterSystems IRIS インストーラは、ユーザの Web サーバを使用する組み込みの Web アプリケーションを提供するよう、[新しい](#)または[アップグレードされた](#)インスタンスを[自動的に構成](#)できます。独自のカスタム Web アプリケーションへのアクセスも提供する場合、インストーラの自動手順で基本的な構成が用意されており、これをニーズに合わせて容易に変更することができます。詳細は、“[Web サーバの自動接続](#)”を参照してください。

InterSystems IRIS インスタンスの組み込みの Web アプリケーションの要求をルーティングするように Web サーバを手動で設定する場合は、他の InterSystems IRIS Web アプリケーションの場合と同じ手順に従う必要があります。“[システムの Web ゲートウェイの設定](#)” ページには、この手順の一般的な概要が示されています。このページの“[Web サーバの手動接続](#)” セクションには、これらの組み込みのユーティリティに固有の補足事項が記載されています。

このページの最後には、InterSystems IRIS 2023.2 より前のバージョンに付属していた[プライベート Web サーバ \(PWS\)](#)を無効化および削除する手順が記載されています。

2.1 管理ポータルの URL

管理ポータルは、InterSystems IRIS データ・プラットフォームに組み込まれている最も重要な Web アプリケーション・ユーティリティの 1 つです。管理ポータルは重要であり、どこでも使用されているため、このページでは管理ポータルの例をいくつか取り上げます。

このページで指定したとおりにシステムを構成した場合、インスタンスの管理ポータルのホーム・ページの URL は以下の形式になります。[<baseURL>](#) はインスタンスのベース URL です。

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

2.2 新規インストールの場合

InterSystems IRIS の新規インスタンスをインストールする際に、システムが**自動構成のインストール条件**を満たしている場合、インストーラは、インストール・プロセスの一部として Web サーバを自動的に構成するかどうかを尋ねます。Web サーバを自動的に構成するには、プロンプトが表示されたらそのオプションを選択するだけです。

Windows の場合、スタンドアロン Web ゲートウェイのインストール方法および**カスタム・セットアップ・タイプ**では、Web サーバを自動的に構成するかどうかを尋ねられることはありません。これらの方法では、既定で、通知なしで自動構成が実行されます。**カスタム・インストール**時に自動構成を行わない場合は、**[Webゲートウェイ]→[IIS用 CSP]**のインストール項目をクリアします。

自動手順では、“**自動構成動作**”の説明に従って要求を新しい InterSystems IRIS インスタンスにルーティングするように Web サーバと Web ゲートウェイを構成します。この動作を見直してください。特定の状況下では、一部の構成手順を手動で実行する必要があります。

Web サーバを自動構成せずにインストールを完了し、その後に Web サーバを構成する場合は、インストーラを使用してインストールを変更します。Windows では、必ず **[Webゲートウェイ]→[IIS用 CSP]** インストール項目を含めてください。“**自動構成動作**”に記載されているように、この場合は、インスタンスの Web ゲートウェイのサーバ・アクセス・プロファイル内でインスタンスの CSPSystem 認証情報を手動で設定して、**IIS を再起動**する必要があります。

スタンドアロン Web ゲートウェイのインストールの場合、インストーラの自動構成手順により、Web サーバ構成に Web ゲートウェイ・モジュールを自動的に追加できます。ただし、InterSystems IRIS インスタンスへの接続を構成し、Web アプリケーションの要求を**手動**でルーティングする必要があります。

システムが自動構成のインストール条件を満たしていない場合は、**Web サーバを手動で接続**します。

注釈 インストールが完了したら、Web サーバを再起動して、すべての構成変更が有効になっていることを確認します。

2.3 アップグレードの場合

2023.2 より前のバージョンでは、すべての InterSystems IRIS インストールにプライベート Web サーバ (PWS) が含まれていました。PWS は、インスタンスの**管理ポータル**および他の**組み込みシステムの Web アプリケーション**の要求を処理するように構成されている Apache httpd の最小ビルドです。

InterSystems IRIS を 2023.2 より前のバージョンからアップグレードする際に、システムが**自動構成の要件**を満たしている場合、インストーラから InterSystems IRIS インストール外の Web サーバに接続するようインスタンスを自動的に構成するオプションが提供されます。プロンプトが表示されたらそのオプションを選択するだけです。

自動手順では、“**自動構成動作**”の説明に従って、組み込みのシステム Web アプリケーションの要求をアップグレードした InterSystems IRIS インスタンスにルーティングするように Web サーバと Web ゲートウェイを構成します。この動作を見直してください。特定の状況下では、一部の構成手順を手動で実行する必要があります。

この手順を実行する方法を紹介するビデオを <https://learning.intersystems.com/course/view.php?id=2333> でご覧いただけます。

重要

PWS は、プロダクションでの Web アプリケーションの提供、または保護された環境外での使用には適していません。バージョン 2023.2 以降、インターシステムズは InterSystems IRIS データ・プラットフォーム製品の新規インストールでの PWS のインストールを中止しました (Community Edition および他の評価版ディストリビューションを除く)。ただし、InterSystems IRIS をバージョン 2023.2 にアップグレードしても、PWS ファイルは残されます。これらのファイルは手動で削除できます。2026 年の最初の EM リリースより、InterSystems IRIS インスタンスをアップグレードすると、そのインスタンスの PWS データはすべて削除されます。

インストーラの自動構成手順では、外部 Web サーバを使用して組み込みのアプリケーションを提供するよう InterSystems IRIS インスタンスを設定できますが、PWS および関連付けられた Web ゲートウェイから外部 Web サーバおよび関連付けられた Web ゲートウェイにカスタム構成を移行することはできません。これらのカスタマイズは、**手動**で外部 Web サーバとその Web ゲートウェイに適用する必要があります。例えば、外部 Web サーバを使用してカスタム・アプリケーションを提供する場合は、そのアプリケーションへの要求に対して **Web ゲートウェイを手動で呼び出して**、Web ゲートウェイの **アプリケーション・アクセス・プロファイル**を作成し、それらの要求を該当する InterSystems IRIS インスタンスに送信する必要があります。

注意

InterSystems Caché® または InterSystems Ensemble® のインスタンスを Windows 上の InterSystems IRIS に移行すると、変換プロセスでインスタンスの Apache PWS の Windows サービスは削除されます。インストーラが変換時に IIS を構成できない場合、InterSystems IRIS インスタンスは Web サーバに接続されず、管理ポータルなどの組み込みの Web アプリケーションは利用できません。

Web サーバを自動構成せずにインストールを完了し、その後に Web サーバを構成する場合は、インストーラを使用してインストールを変更します。Windows では、必ず **[Webゲートウェイ]→[IIS用 CSP]** インストール項目を含めてください。**“自動構成動作”**に記載されているように、この場合は、インスタンスの Web ゲートウェイのサーバ・アクセス・プロファイル内でインスタンスの CSPSystem 認証情報を手動で設定して、**IIS を再起動**する必要があります。

自動構成手順後、PWS は無効化されます。ただし、PWS はアンインストールされず、PWS 構成のカスタマイズは保持されます。カスタム構成をすべて移行し、Web サーバが必要なすべての Web アプリケーションの要求を正常にルーティングすることを確認したら、そのインスタンスの **PWS を削除**できます。

Web サーバを自動構成せずにインストールを完了し、その後に Web サーバを構成する場合は、インストーラを使用してインストールを変更します。Windows では、必ず **[Webゲートウェイ]→[IIS用 CSP]** インストール項目を含めてください。**“自動構成動作”**に記載されているように、この場合は、インスタンスの Web ゲートウェイのサーバ・アクセス・プロファイル内でインスタンスの CSPSystem 認証情報を手動で設定して、**IIS を再起動**する必要があります。

自動構成手順後、PWS は無効化されます。ただし、PWS はアンインストールされず、PWS 構成のカスタマイズは保持されます。カスタム構成をすべて移行し、Web サーバが必要なすべての Web アプリケーションの要求を正常にルーティングすることを確認したら、そのインスタンスの **PWS を削除**できます。

注釈 アップグレードおよびその後の Web サーバと Web ゲートウェイへの変更が完了したら、Web サーバを再起動して、すべての構成変更が有効になっていることを確認します。

2.4 Web サーバの自動接続

2.4.1 自動構成が可能な場合

InterSystems IRIS インスタンスのインストール時に Web ゲートウェイを含むセットアップ・タイプを選択する場合は、以下のリストで示す条件下で、Web ゲートウェイを介してインスタンスに要求を提供するように Web サーバを自動的に構成できます。これらの条件が適用される場合は、**Web ゲートウェイをスタンドアロン・コンポーネントとしてインストール**するすべての方法で、Web ゲートウェイを含めて Web サーバを構成することもできます。ただし、各 InterSystems IRIS アプリケーション・サーバへの接続は、**手動**で構成する必要があります。

Web サーバの自動構成が可能な場合を以下に示します。

1. 以下を使用している。
 - ・ UNIX®、Linux、または macOS 上の Apache httpd
 - ・ Microsoft Windows 上の Microsoft インターネット・インフォメーション・サービス (IIS)。
2. インストールを実行する前に、ローカル・ファイル・システム内に Web サーバが既にインストールされている (IIS の場合は、これを有効化している)。UNIX® または Linux システムに Apache をインストールする方法については、次の Developer Community の記事を参照してください：<https://community.intersystems.com/post/how-install-apache-iris-supported-operating-systems>

注釈 macOS システムでは、Homebrew パッケージ・マネージャ (<https://brew.sh/>) を使用して Apache httpd のコピーをインストールしておく必要があります。インストーラは、macOS と共に配布される Apache httpd インストールでの自動構成はサポートしません。他のすべてのプラットフォームについては、インストーラはプラットフォームで提供される Web サーバをサポートします。

3. 既定のインストール場所に Web サーバがインストールされている。
4. Web サーバが実行されている。

2.4.2 自動構成動作

重要 インストーラの自動構成手順では、インストール時にカスタム・ポートを指定した場合でも、Web サーバがリッスンするポートを変更することはできません。インストール時にカスタム・ポートを指定すると、InterSystems IRIS インスタンスの [WebServerPort](#) パラメータのみが変更されます。他の場合はすべて、インストーラは Web サーバが既定の HTTP ポート 80 (macOS の場合は 8080) でリッスンすると想定します。

InterSystems IRIS Web アプリケーションにアクセスするには、Web サーバ構成で指定されたポートを使用する必要があります。

特に明記されていない限り、インストーラの自動構成手順では以下が行われます。

1. Web ゲートウェイ・モジュールが Web サーバ構成に含まれていない場合、インストーラはこれを追加し、Web ゲートウェイ・バイナリを含むディレクトリに対する適切な権限を設定します。
2. Web サーバ構成で、以下の相対 URL パスのハンドラとして Web ゲートウェイを有効にします。
 - ・ `/<instanceName>`

`<instanceName>` は、インストールまたはアップグレードするインスタンスの名前 (すべて小文字) です。

重要 インストーラが Apache Web サーバの自動構成手順を実行するインスタンスの名前が、以前にその Web サーバを使用するように構成されていたインスタンスと同じである場合、構成ファイル (`httpd.conf`) に追加される `ApacheCSP-SECTION-<instanceName>` セクションに手動で加えた編集内容は上書きされ、セクションの元の内容がリストアされ、カスタマイズした内容がすべて削除されます。コメントでこのセクションの開始と終了が示されます。

3. Web サーバ構成で、以下の相対 URL パスのハンドラを有効にします。
 - ・ `/api`
 - ・ `/csp`
 - ・ `/isc`
 - ・ `/oauth2`

・ /ui

重要 インストーラが Apache Web サーバの自動構成手順を実行するたびに、構成ファイル (httpd.conf) に追加される `ApacheCSP-SECTION-<instanceName>` に手動で加えた編集内容は上書きされ、セクションの元の内容がリストアされます。コメントでこのセクションの開始と終了が示されます。Apache 構成をカスタマイズする場合は、ファイルの他の場所に指示文を追加してこれを行う必要があります。

注釈 Windows システムの場合、[VS Code](#) への接続を完全に機能させるには、追加の IIS 構成を実行する必要があります (インスタンスの /api/atelier アプリケーションによって促進されます)。

- UNIX®, Linux, macOS システム上の **カスタム・インストール** の場合、Apache Web サーバ構成にリストされているポート番号を、インストール・ウィザードで指定したポート番号に更新します。
- UNIX®, Linux, macOS システムで、相対パス `/csp/docbook/` 内の要求を InterSystems ドキュメント・サイトの対応するページにリダイレクトする指示文を、Apache Web サーバ構成に追加します。これは、管理ポータルからドキュメントへのリンクに必要になります。

注釈 Windows システムで、要求を `/csp/docbook/` から InterSystems ドキュメント・サイトに **手動** でリダイレクトするように IIS を構成する必要があります。

- SELinux を搭載した RHEL システムで、Web ゲートウェイ構成とログ・ファイルへの適切な読み取り/書き込みアクセス権を構成して、Apache Web サーバからの送信ネットワーク接続を可能にします。
- Web ゲートウェイ構成で、指定したスーパーサーバ・ポートと CSPSystem アカウント認証情報を使用して、インストールまたはアップグレードしている InterSystems IRIS インスタンスのサーバ・アクセス・プロファイルを作成します。このサーバ・アクセス・プロファイルにより、Web ゲートウェイは新規インスタンスに接続できます。サーバ・アクセス・プロファイルの **サーバ名** は、すべて小文字のインスタンス名です。

注釈 以前に Web サーバを構成せずに InterSystems IRIS 2023.2 のインスタンスをインストールし、その後インストーラの変更オプションを実行して、インストーラが自動構成手順を実行できるようにする場合は、**サーバ・アクセス・プロファイル** 内で手動でインスタンスに **CSPSystem アカウント認証情報** を設定する必要があります。その後、**IIS を再起動** します。

- Web ゲートウェイ構成で、次の相対アプリケーション・パスの **アプリケーション・アクセス・プロファイル** を作成します。

・ /<instanceName>

<instanceName> は、インストールまたはアップグレードするインスタンスに指定した名前 (すべて小文字) です。

アプリケーション・アクセス・プロファイルは、新しいサーバ・アクセス・プロファイルを、このアプリケーション・パスに送信されるすべての要求の既定のサーバとして指定します。

- Web ゲートウェイ構成で、次の相対アプリケーション・パスの **アプリケーション・アクセス・プロファイル** を作成します (まだない場合)。

・ / (root)

・ /csp

これらのアプリケーション・アクセス・プロファイルは、新しいサーバ・アクセス・プロファイルを、これらのアプリケーション・パスに送信されるすべての要求の既定のサーバとして指定します。

Web ゲートウェイ構成内にこれらのアプリケーション・アクセス・プロファイルが既に存在する場合、インストーラはそれらを更新して、新しいサーバ・アクセス・プロファイルをこれらのアプリケーション・パスへのすべての要求の既定のサーバとして指定します。

10. InterSystems IRIS インスタンスの [WebServerPort](#) パラメータの値を 80 (macOS では 8080) に設定し、[WebServerURL-Prefix](#) の値をすべて小文字のインスタンス名に設定します。(インストール時にカスタム・ポート番号を指定する場合、インストーラは代わりに WebServerPort パラメータをその番号に設定します。)InterSystems IRIS インスタンスはこれらのパラメータを使用して、[InterSystems スタジオ](#)に接続します。

注釈 Community Edition をインストールする場合は、WebServerPort パラメータを Web サーバのポート番号に合わせて手動で更新する必要があります。

既存の InterSystems IRIS インスタンスを更新または変更していて、そのインスタンスが以前に別のポートを使用して Web アプリケーションを提供していた (または Web アプリケーションをまったく提供していなかった) 場合も、WebServerPort パラメータを手動で更新する必要があります。

11. InterSystems IRIS インスタンスの WebServer パラメータの値を false (0) に設定し、インスタンスの起動時にインスタンスがプライベート Web サーバを起動しないようにします。この変更により、プライベート Web サーバは事実上無効になります。

重要 Community Edition をインストールする場合、インストーラは WebServer パラメータを設定しません。[インスタンスのプライベート Web サーバを手動で無効にする](#)必要があります。

12. Windows システムで、[InterSystems IRIS ランチャー](#)が、ランチャーを使用する Web アプリケーション・ユーティリティ (つまり、管理ポータル、クラス・リファレンス、ドキュメント) の有効な URL を構成するために必要な、関連する [InterSystems IRIS サーバ・マネージャ](#)のパラメータを構成します。これらのパラメータは、InterSystems スタジオに接続するためにも必要です。インストーラは、サーバ・マネージャの[ウェブサーバポート](#)・パラメータを 80 (macOS では 8080) に設定します。

注釈 既存の InterSystems IRIS インスタンスをアップグレードまたは変更していて、そのインスタンスが以前に別のポートを使用して Web アプリケーションを提供していた (または Web アプリケーションをまったく提供していなかった) 場合、サーバ・マネージャの [\[ウェブサーバポート\]](#)・パラメータを Web サーバのポート番号に合わせて手動で更新する必要があります。

自動構成手順では、アプリケーション・パスの前にすべて小文字の[インスタンス名を接頭語として含む URL](#)で、特定の InterSystems IRIS インスタンスの組み込みのアプリケーションにアクセスできるようにします。[インスタンスの指定に使用する接頭語は、簡単に構成できます](#)。

さらに、自動構成手順では、[インスタンス接頭語を省略した URL](#)で、最近構成された InterSystems IRIS インスタンスの組み込みのアプリケーションを利用できるようにします。つまり、1 つのインスタンスのみが構成されている場合、簡略化された URL `<protocol>://<hostname>/csp/sys/UtilHome.csp` でそのインスタンスの管理ポータルを利用できます。

注釈 複数のインスタンスが構成されているシステムで、最近構成したインスタンスをアンインストールする場合、/ (root) と /csp のアプリケーション・アクセス・プロファイルは、アンインストールしたインスタンスのサーバ・アクセス・プロファイルに関連付けられたままになります。これらのパスで要求を処理するには、要求を別の既定アプリケーションに送信するように / (root) および /csp パスのプロファイルを手動で更新する必要があります。その後、そのインスタンスの古いサーバ・アクセス・プロファイルを削除できます。

2.4.2.1 自動構成例

Apache httpd が実行されている Linux システムに IRISserv1 という名前の単一の InterSystems IRIS インスタンスをインストールした場合、自動構成手順により以下の指示文が Apache `httpd.conf` ファイルに追加されます。

```
#### BEGIN-ApacheCSP-SECTION ####
LoadModule csp_module_sa "/opt/webgateway/bin/CSPa24.so"
CSPModulePath "/opt/webgateway/bin/"
CSPConfigPath "/opt/webgateway/bin/"

<Location "/csp/">
    CSP On
```

```

</Location>
<Location "/api/">
    CSP On
</Location>
<Location "/oauth2/">
    CSP On
</Location>
<Location "/isc/">
    CSP On
</Location>
<Location "/ui/">
    CSP On
</Location>

<Directory "/opt/webgateway/bin/">
    AllowOverride None
    Options None
    Require all granted
    <FilesMatch "\.(log|ini|pid|exe)$">
        Require all denied
    </FilesMatch>
</Directory>

Redirect /csp/docbook/ http://docs.intersystems.com/irislatest/csp/docbook/
#### END-ApacheCSP-SECTION #####
#### BEGIN-ApacheCSP-SECTION-IRISSERV1 #####
# Note: IRISSERV1 reinstallation or upgrade may replace this section.
<Location /irisserv1>
    CSP On
</Location>
Redirect /irisserv1/csp/docbook/ http://docs.intersystems.com/irislatest/csp/docbook/
#### END-ApacheCSP-SECTION-IRISSERV1 #####

```

Web ゲートウェイ構成には、IRISSERV1 サーバ・アクセス・プロファイルと、パス / (root)、/csp、/irisserv1 のアプリケーション・アクセス・プロファイルが含まれます。3 つのアプリケーション・アクセス・プロファイルはすべて、IRISSERV1 を既定のアプリケーション・サーバとして指定します。

この構成の結果、IRISSERV1 の管理ポータル ホーム・ページが
<http://localhost/irisserv1/csp/sys/UtilHome.csp> またはより簡易な
<http://localhost/csp/sys/UtilHome.csp> で利用できるようになります。

その後、IRISserv2 という名前の 2 番目の InterSystems IRIS インスタンスを同じシステムにインストールすると、インストールは手動による変更を上書きして、**httpd.conf** ファイルの **ApacheCSP-SECTION** の内容をリストアします。その後、**ApacheCSP-SECTION-IRISSERV1** コード・ブロックの後に以下の指示文を追加します。

```

#### BEGIN-ApacheCSP-SECTION-IRISSERV2 #####
# Note: IRISSERV2 reinstallation or upgrade may replace this section.
<Location /irisserv2>
    CSP On
</Location>
Redirect /irisserv2/csp/docbook/ http://docs.intersystems.com/irislatest/csp/docbook/
#### END-ApacheCSP-SECTION-IRISSERV2 #####

```

自動構成手順により、IRISSERV2 サーバ・アクセス・プロファイルが Web ゲートウェイ構成に追加されます。パス /irisserv2 のアプリケーション・アクセス・プロファイルも追加されます。/irisserv2、/ (root)、/csp のアプリケーション・アクセス・プロファイルは IRISSERV2 を既定のアプリケーション・サーバとして指定します。パス /irisserv1 のアプリケーション・アクセス・プロファイルは変更されません。

これらの変更の結果、IRISSERV1 の管理ポータル ホーム・ページは
<http://localhost/irisserv1/csp/sys/UtilHome.csp> でのみ利用できるようになります。IRISserv2 の管理ポータル ホーム・ページは <http://localhost/irisserv2/csp/sys/UtilHome.csp> または
<http://localhost/csp/sys/UtilHome.csp> で利用できます。

2.5 Web サーバの手動接続

InterSystems IRIS に組み込まれている Web ベースのユーティリティ (管理ポータルなど) の要求をルーティングするように Web サーバを手動で構成する手順は、基本的にカスタム InterSystems IRIS Web アプリケーションを構成する手順と同じです。この手順は、“システムに Web ゲートウェイを設定する” に概説されています。

このセクションでは、これらの組み込みのアプリケーションに固有の補足事項を提供します。

2023.2 より前のバージョンの InterSystems IRIS をアップグレードまたは再構成する場合、プライベート Web サーバを置き換えるように Web サーバを構成した後、[プライベート Web サーバを無効化および削除](#)できます。

2.5.1 必要なアプリケーション・パスの指定

有効化する InterSystems IRIS システム・アプリケーションに対して、以下を行う必要があります。

1. [Web ゲートウェイを呼び出して](#)、対応するアプリケーション・パスとそのアプリケーション・パス内の下位パスに予期されるすべての要求を処理するように、Web サーバを構成します。
2. InterSystems IRIS インスタンスの Web ゲートウェイの[サーバ・アクセス・プロファイル](#)を構成します。
3. Web ゲートウェイの[サーバ・アクセス・プロファイル](#)を構成し、そのアプリケーション・アクセス・プロファイルをそのアプリケーション・パス (とその下位パス) に送信される要求の既定のアプリケーション・サーバとして指定します。

InterSystems IRIS と共にインストールされているシステム Web アプリケーションのリストは、“[組み込みアプリケーション](#)”を参照してください。これらすべての組み込みの Web ユーティリティの要求を処理するように、Web サーバと Web ゲートウェイを構成することを強くお勧めします。これらの Web アプリケーションは、以下のベース・パスを持つエンドポイントを使用します。

- /api
- /csp
- /isc
- /ui
- /oauth2

Web サーバ構成とアプリケーション・アクセス構成はどちらも継承ルールを適用するため、これら 5 つのベース・パスへの要求を構成することが最適なアプローチになります。これが、インストーラの[自動構成手順](#)で採用されるアプローチです。

1 つの Web ゲートウェイから複数の InterSystems IRIS インスタンスを処理する際、インスタンスの組み込みユーティリティに個別にアクセスする場合は、各インスタンスの `CSPConfigName` で始まるパスに送信された要求がそのインスタンスにルーティングされるように、Web サーバと Web ゲートウェイを構成する必要があります。[インスタンスにこのパラメータを構成](#)して、カスタム文字列をこのインスタンス接頭語として機能させることができます。既定では、このパラメータはすべて小文字のインスタンス名です。そのため、各インスタンスにパス `<instanceName>` への要求を構成することが最適なアプローチになります。`<instanceName>` は、すべて小文字のインスタンス名です。これが、インストーラの[自動構成手順](#)で採用されるアプローチです。

2.5.2 各パスの必要なすべての要求のルーティング

Web ゲートウェイを呼び出して、特定のパスに送信されるすべての要求を処理するように、または特定のファイル・タイプ拡張子を持つファイルの要求のみを処理するように、Web サーバを構成できます。

以下のファイル・タイプの要求のみをルーティングすることで、管理ポータルを提供できます。

```
.csp .cls .zen .cxw .jpg .gif .png .svg .css .js
```

ただし、REST API は、ファイルを指定しない（したがって、ファイル・タイプを指定しない）エンドポイントでの要求の受信をサポートする必要があります。このようなアプリケーションの場合、ファイル・タイプに基づいて要求に対して Web ゲートウェイを呼び出すことができないため、Web ゲートウェイをパス・レベルですべての要求のハンドラとして使用できるようにする必要があります。

2.5.3 ドキュメント・リンクのリダイレクト

管理ポータル内のページに関連するドキュメントへのリンクが含まれる場合、それらのリンクはインスタンスのパス `<instancePrefix>/csp/docbook` 内のリソースを参照します。`<instancePrefix>` は、すべて小文字のインスタンス名です。Windows では、InterSystems IRIS ランチャーのドキュメント・リンクもこのパスにルーティングされます。

これらのリンクを機能させるには、Web サーバが要求をこのパスにリダイレクトして、InterSystems ドキュメントの Web サイト上の対応する URL に送信する必要があります。

重要 Windows システムでのすべてのインストールについて、“[Microsoft IIS の場合](#)”の説明に従って、ドキュメント・リンクにリダイレクトするように手動で IIS を構成する必要があります。

ドキュメント・リンクの宛先 URL は、使用しているインターシステムズ製品によって異なります。インスタンスの正しい製品識別子を探すには、使用しているインターシステムズ製品およびバージョンのドキュメント・ページにアクセスして、ブラウザのアドレス・バーを確認します。製品のバージョンを指定する URL (`irislatest`、`healthconnectlatest` など) は、常にその製品の最新リリース・バージョンのドキュメントを提供します。

以下のセクションでは、Apache httpd と Microsoft IIS のドキュメント・リンクをリダイレクトする方法について説明します。お使いのシステムに最適な方法は異なることがあります。利用できるオプションについては、Web サーバのドキュメントを参照してください。

2.5.3.1 Apache の場合

Apache Web サーバの場合、構成する各インスタンスの Web サーバの構成に、インスタンスの `<instancePrefix>/csp/docbook` パスを InterSystems ドキュメント Web サイト上の類似の URL にリダイレクトする `Redirect` 指示文を追加します。(既定では、`<instancePrefix>` は小文字のインスタンス名です。)

例えば、以下の `Redirect` 指示文を (`httpd.conf` ファイルまたはその中に含まれるファイル内の) Web サーバの構成に追加することで、IRISinst2 という名前の InterSystems IRIS インスタンスのドキュメント・リンクをリダイレクトできます。

```
Redirect /irisinst2/csp/docbook/ http://docs.intersystems.com/irislatest/csp/docbook/
```

2.5.3.2 Microsoft IIS の場合

Microsoft IIS Web サーバの場合は、以下の手順を実行します。

1. IIS の [HTTP リダイレクト] が有効になっていることを確認します。
 - a. [Windows の機能の有効化または無効化] を探すか、[コントロール パネル] を開き、[プログラム]→[プログラムと機能]→[Windows の機能の有効化または無効化] を選択して、[Windows の機能] マネージャを開きます。
 - b. まだ選択されていない場合は、[インターネット インフォメーション サービス]→[World Wide Web サービス]→[HTTP 共通機能]→[HTTP リダイレクト] を選択します。
 - c. [OK] をクリックします。
2. インスタンスのインストール・ディレクトリ内に `/csp/csp/docbook` ディレクトリを作成します。

IIS では、Web サーバが静的ファイル进行处理するかどうかに関係なく、各アプリケーション・パスが物理パスに対応している必要があります。インストーラが IIS を自動構成する際、InterSystems IRIS インスタンスのパスは、インスタンスのベース・インストール・ディレクトリではなく、インスタンスの `/csp` ディレクトリにマップされます。その場合は、`<installDir>/csp/csp/docbook` ディレクトリを作成します。`<installDir>` は、インスタンスのインストール・ディレクトリです。アプリケーションは、`/csp/docbook` の要求をこのディレクトリに関連付けます。

3. **[接続]** パネルで `localhost` 接続、**[サイト]**、インスタンスに対応するアプリケーションの順に展開します。そのアプリケーション内で、**[CSP]→[DocBook]** を選択します。
4. **DocBook ホーム・ページ**で、**[HTTP リダイレクト]** を選択 (ダブルクリック) します。
5. **[HTTP リダイレクト]** ページで、以下を行います。
 - ・ **[このリダイレクト先に要求をリダイレクト:]** というフィールドに、オンライン・ドキュメント・ページの URL を入力し、文字列 `$$Q` を追加します (すべての接尾語とクエリ・パラメータを URL 末尾に保持するため)。次に例を示します。

```
https://docs.intersystems.com/irislatest/csp/docbook$$Q
```
 - ・ **[すべての要求を (相対的なリダイレクト先ではなく) 正確なリダイレクト先にリダイレクト]** を選択します。
6. **[アクション]** パネルで、**[適用]** を選択します。
7. IIS を再起動して、変更を適用します。

2.5.4 Windows のみ : InterSystems IRIS サーバ・マネージャの更新

Windows システムでは、InterSystems IRIS ランチャーは InterSystems IRIS サーバ・マネージャ内で指定されたサーバ接続の詳細を使用して、ユーザをインスタンスの Web ベースのユーティリティに誘導します。インスタンスに対して InterSystems IRIS ランチャーを有効にするには、InterSystems IRIS サーバ・マネージャで、新しい Web サーバ構成に合わせてサーバ接続の詳細を変更します。“リモート・サーバ接続の定義”を参照してください。

2.5.5 Windows のみ : VS Code を有効にするように IIS を構成

IIS を使用する Windows システムでは、VS Code が InterSystems IRIS インスタンスに接続できるように IIS を構成する必要があります。必要な IIS 構成については、以降のセクションで説明します。IIS の構成の詳しい手順は、IIS のドキュメントを参照してください (<https://learn.microsoft.com/ja-jp/iis/get-started/introduction-to-iis/iis-web-server-overview>)。

2.5.5.1 WebDAV モジュールの無効化

InterSystems IRIS インスタンスの `/api/atelier` アプリケーションの通信を処理するために、IIS WebDAV モジュール (インストールされている場合) と Web ゲートウェイが両方とも有効になっている場合、IIS WebDAV モジュールは Web ゲートウェイの妨げになります。このアプリケーションは、インスタンスと VS Code 間の接続を提供します。

VS Code を InterSystems IRIS インスタンスで使用するには、関連するパスの WebDAV ハンドラ・マッピングを削除し、WebDAV モジュールを無効にする必要があります。Web サーバ構成に応じて、この関連パスはインスタンス接頭語パス、`<instancePrefix>/api`、または `<instancePrefix>/api/atelier` になります。

これらの構成は、インターネット・インフォメーション・サービス・マネージャを使用するか、`applicationHost.config` ファイルを編集して、関連するパス (`{path}`) の `<location>` 指示文ブロック内に `<remove>` 指示文を含めることで実行できます (次の例を参照)。

```
<location path="[path]">
  <system.webServer>
    <modules>
      <remove name="WebDAVModule" />
    </modules>
    <handlers>
      <remove name="WebDAV" />
    </handlers>
  </system.webServer>
</location>
```

IIS 構成に変更を加えたら、**IIS を再起動**して変更が有効になっていることを確認します。

2.5.5.2 WebSocket 機能の有効化 (デバッグ用)

VS Code のデバッグ・ツールには、InterSystems IRIS インスタンスへの WebSocket 接続が必要です。以下の手順を実行して、IIS の [WebSocket プロトコル] 機能が有効であることを確認します。

1. [Windows の機能の有効化または無効化] を探るか、[コントロール パネル] を開き、[プログラム]→[プログラムと機能]→[Windows の機能の有効化または無効化] を選択して、[Windows の機能] マネージャを開きます。
2. [インターネット インフォメーション サービス]→[World Wide Web サービス]→[アプリケーション開発機能] を選択します。
3. まだ選択されていない場合は、[WebSocket プロトコル] を選択します。
4. [OK] をクリックします。
5. **IIS を再起動**し、すべての変更が有効になっていることを確認します。

2.5.5.3 ダブル・エスケープの許可 (特定のパッケージへのアクセス用)

% (パーセント) 文字で始まり、2 つの 16 進数 (0 ~ 9 の数字または a ~ f の文字) が続く名前のファイルの場合、IIS は既定でこれらの最初の 3 文字をエンコードされた 16 進数文字として解釈します。そのため、VS Code はこのようなファイルを表示または編集できません。

このようなファイルを表示または編集する必要がある場合は、グローバルに、またはインスタンスに対応する特定の IIS アプリケーションの場所に対して、IIS の [要求のフィルタリング] を [ダブル エスケープを許可する] に構成する必要があります。この設定は、IIS マネージャまたはコマンド行インタフェースを使用して (<https://learn.microsoft.com/ja-jp/iis/manage/configuring-security/configure-request-filtering-in-iis> を参照)、または IIS 構成ファイル内の該当する <requestFiltering> 要素に属性 allowDoubleEscaping="true" を設定して (<https://learn.microsoft.com/ja-jp/iis/manage/configuring-security/use-request-filtering> を参照) 変更することができます。

2.6 2023.2 より前のバージョンからのアップグレード：プライベート Web サーバの無効化および削除

InterSystems IRIS 2023.2 からこのバージョンの InterSystems IRIS にアップグレードし、外部 Web サーバを使用するようにインスタンスを再構成した場合、インスタンスのプライベート Web サーバを無効化および (オプションで) 削除することができます。そのためには、以下の手順を実行します。

1. 以下の点を確認してください。
 - ・ 外部 Web サーバと Web ゲートウェイが、必要なすべての Web アプリケーションの要求を正常にルーティングすること。
 - ・ /csp/docbook アプリケーションへの要求がドキュメントの Web サイトにリダイレクトされること。

- ・ Windows の場合：InterSystems IRIS サーバ・マネージャでインスタンスの接続の詳細が、Web サーバおよび Web ゲートウェイの構成と一致していること。（これは、InterSystems IRIS ランチャーを有効にするために必要です。）
2. インスタンスの **WebServer** パラメータを 0 にして、インスタンスの起動時に PWS が起動しないようにします。このパラメータは、管理ポータルで **[システム管理]→[構成]→[追加の設定]→[開始]** に移動して編集するか、**CPF.ini** ファイルで直接編集することができます。（インストーラが外部 Web サーバを使用するようにインスタンスを自動構成した場合、この手順は既に完了しています。）
 3. インスタンスを再起動して PWS を停止します。または、コマンド・プロンプトから以下のコマンドを発行することで、インスタンスを再起動せずに PWS を停止できます。

Unix®/Linux/macOS

```
kill `cat <irisInstallDir>/httpd/logs/httpd.pid`
```

Windows

```
<irisInstallDir>\httpd\bin\httpd -k stop -n <instanceName>httpd
```

<irisInstallDir> はインスタンスのインストール・ディレクトリ、<instanceName> はインスタンスの名前です。

注釈 何らかの理由でインスタンスの PWS を再度有効にする必要がある場合は、**CPF.ini** ファイルで **WebServer** パラメータを 1 に設定してから、インスタンスを再起動します。または、コマンド・プロンプトから以下のコマンドを発行することで、インスタンスを再起動せずに PWS を起動できます。

Unix®/Linux/macOS

```
<irisInstallDir>/httpd/bin/httpd -d <irisInstallDir>/httpd  
-c "Listen <port>"
```

Windows

```
<irisInstallDir>\httpd\bin\httpd -k start -n <instanceName>httpd  
-c "Listen <port>"
```

<irisInstallDir> はインスタンスのインストール・ディレクトリ、<instanceName> はインスタンスの名前、<port> は PWS のポート番号です。

重要 次の手順のとおり PWS を削除し、元に戻すことができなくなる前に、必要なすべての Web アプリケーションの要求をルーティングするように Web サーバと Web ゲートウェイが正常に構成されていることを確認してください。

4. オプション：<irisInstallDir>/httpd/ ディレクトリを削除して、インスタンスの PWS を完全に削除します。<irisInstallDir> は、インスタンスのインストール・ディレクトリです。

3

概要：システムの Web ゲートウェイの設定

InterSystems Web アプリケーションを提供するには、Web クライアント (Web ブラウザなど) からの要求を [InterSystems Web ゲートウェイ](#) を介して InterSystems IRIS® アプリケーション・サーバにルーティングするように Web サーバを構成する必要があります。

注釈 多くの場合、InterSystems IRIS と Web ゲートウェイのどちらのインストーラも、この構成のほとんどを自動的に実行できます。“[Web サーバを使用した管理ポータルおよび他の組み込みの Web アプリケーションへのアクセス](#)” を参照してください。

[プライベート Web サーバ \(PWS\)](#) を使用するインスタンスから InterSystems IRIS インスタンスをアップグレードする場合は、InterSystems IRIS インストーラを使用して Web サーバを簡単に自動構成する方法を紹介するビデオをこちら (<https://learning.intersystems.com/course/view.php?id=2333>) からご覧いただけます。

Web サーバと 1 つ以上の InterSystems IRIS インスタンスの間の接続を手動で設定するプロセスは、以下の手順で構成されます。

1. [Web ゲートウェイ・ファイル](#)をインストールするか、[webgateway コンテナ](#)の Docker イメージを取得します。
2. Web サーバ構成に Web ゲートウェイを追加します。
3. Web サーバ構成内で、Web サーバが Web ゲートウェイを介してルーティングする必要がある要求を指定します。
4. Web ゲートウェイ構成内で、Web アプリケーションを提供する各 InterSystems IRIS アプリケーション・サーバの [サーバ・アクセス・プロファイル](#)を登録します。
5. Web ゲートウェイ構成内で、[アプリケーション・アクセス・プロファイル](#)を登録して、アプリケーション・パスを、そのパスで利用可能なアプリケーションを提供する InterSystems IRIS アプリケーション・サーバに関連付けます。
6. クライアントと Web サーバ間、Web ゲートウェイと InterSystems IRIS インスタンス間、クライアントと Web ゲートウェイ管理ページ間の [接続を保護](#)します。

ここでは、これらの手順について概説します。各手順には、必要に応じて、より具体的な手順へのリンクが用意されています。最後に、[Web ゲートウェイのデコミッション](#)方法の概要を示します。

3.1 Web ゲートウェイ・ファイルのインストール

システム内の各 Web サーバに、Web ゲートウェイのコピーをインストールします。Web ゲートウェイは、ニーズに応じて、[InterSystems IRIS のインストールの一部](#)として、[スタンドアロン・コンポーネント](#)として、または [webgateway Docker コンテナ](#)として導入できます。

多くの場合、InterSystems IRIS インストーラとスタンドアロン Web ゲートウェイ・インストーラは、Web サーバに接続するように Web ゲートウェイを自動的に構成できます。これにより、次に示すように、このページの設定手順を実行する必要性が最小限に抑えられます。

- ・ スタンドアロン Web ゲートウェイの場合、インストーラは Web ゲートウェイを Web サーバ構成に自動的に追加できます。各 InterSystems IRIS インスタンスの組み込みのシステム・アプリケーションを含む、InterSystems Web アプリケーションのハンドラとして Web ゲートウェイを呼び出すように Web サーバを手動で構成する必要があります。
- ・ InterSystems IRIS インスタンスの場合、インストーラは Web ゲートウェイを Web サーバ構成に自動的に追加し、Web ゲートウェイをインスタンスの組み込みのシステム・アプリケーションのハンドラとして呼び出すように Web サーバを自動的に構成できます。システムによっては、“自動構成動作” に示すように、構成を完了するために特定のタスクを手動で実行する必要があります。この構成プロセスが完了したら、カスタム・アプリケーションのニーズに合わせて容易に拡張できます。
- ・ いずれの場合も、接続セキュリティを手動で構成する必要があります。

3.1.1 InterSystems IRIS インストールの一部としてのインストール

InterSystems IRIS は、管理ポータルなどの Web アプリケーションを介して、複数の重要なシステム機能へのアクセスを提供します。このため、インターシステムズのインストーラには、必要な Web ゲートウェイ・ファイルが以下のセットアップ・タイプの一部として含まれています。

- ・ 開発
- ・ サーバ
- ・ Web サーバ (Windows の場合)
- ・ カスタム、Web サーバ・ゲートウェイ・コンポーネントが選択されている場合

3.1.2 スタンドアロン・コンポーネントとしてのインストール

InterSystems IRIS インスタンスをホストするマシンとは別のマシンに Web サーバを導入する場合は、次のいずれかの方法で、Web ゲートウェイをスタンドアロン・コンポーネントとして Web サーバ・マシンにインストールします。

- ・ スタンドアロン Web ゲートウェイ・インストーラを実行します。このインストーラは、WRC のダウンロード・ページ (<https://wrc.intersystems.com/wrc/coDistGen.csp>) で入手できます。[名前] 列に web gateway と入力して、お使いのシステムのキットを見つけます。InterSystems IRIS インストーラと同様に、スタンドアロン Web ゲートウェイ・インストーラは、Windows の場合は実行可能ファイルとして、他のすべてのシステムの場合はスクリプト (GatewayInstall) として実装されます。
- ・ 以下のいずれかのセットアップ・タイプを使用して、InterSystems IRIS インストーラを実行します。
 - Web サーバ (Windows の場合)
 - カスタム、Web サーバ・ゲートウェイ・コンポーネントのみを選択
- ・ InterSystems Container Registry から webgateway コンテナの Docker イメージをプルします。詳細は、“webgateway コンテナの導入” を参照してください。

多くの場合、スタンドアロン・インストーラには、Web ゲートウェイを使用するように Web サーバを自動的に構成するオプションが用意されています。自動構成後、Web ゲートウェイを Web サーバ構成に追加するためにさらなる操作は必要ありません。ただし、システム内の各 InterSystems IRIS インスタンスの組み込みシステム・アプリケーションを含む、InterSystems Web アプリケーションのハンドラとして Web ゲートウェイを呼び出すように、Web サーバを手動で構成する必要があります。また、手動で接続セキュリティを構成することも必要です。

3.1.3 webgateway コンテナの導入

[InterSystems Container Registry](#) は、webgateway コンテナの Docker イメージを提供します。これらのコンテナには、Web ゲートウェイ拡張で事前構成されている Web サーバ (Apache または nginx) が含まれます。

webgateway コンテナを使用して Web ゲートウェイ接続を構成するには、構成が以下のセクションで説明する条件を満たしている必要があります。ただし、コンテナ化された導入の場合、構成方法はこのページで説明する方法とはかなり異なります。webgateway コンテナを使用する場合：

- ・ Web ゲートウェイを Web サーバ構成に追加するのにさらなる操作は必要ありません。
- ・ 導入時の Web サーバ構成を自動化するには、特定のパスに送信された要求に対して Web ゲートウェイを呼び出すように、webgateway コンテナで Web サーバを構成する必要があります。それには、イメージの Dockerfile 内で構成ファイルをプログラムによって変更するか、コンテナ・クラスタの YAML ファイルで構成を記述します。
- ・ 導入時の Web ゲートウェイ構成を自動化するには、[CSP.ini マージ機能](#)を利用して、webgateway コンテナのサーバ・アクセス・プロファイルとアプリケーション・アクセス・プロファイルを、それらの [CSP.ini](#)内で直接定義します。

具体的な手順は、“[InterSystems Web ゲートウェイ・コンテナの使用法](#)”を参照してください。

3.2 Web ゲートウェイでの Web サーバの機能拡張

Web クライアントから静的コンテンツ (プレーン・テキスト、イメージ、JavaScript) を要求された際の Web サーバのロールは簡単で、要求の URL パスに対応するファイル・システムの場所からコンテンツを処理するだけです。Web サーバの構成では、URL パスとアクセス権のあるファイル・システムの場所とのマッピングを定義します。

ただし、Web アプリケーションから動的なコンテンツを提供するには、HTTP 要求を Web アプリケーション・コードの呼び出しとして解釈できるライブラリ・モジュールまたは外部プログラムを使用して、Web サーバの機能を拡張する必要があります。

これが、InterSystems Web ゲートウェイの目的です。Web サーバの機能を拡張して、InterSystems Web アプリケーションの要求を認識して処理します。Web サーバが InterSystems IRIS Web アプリケーションを提供するには、[Web ゲートウェイ拡張](#)を含めるように [Web サーバを構成](#)した後、通常、要求で指定された URL パスに基づいて、[Web ゲートウェイ拡張で処理すべき要求を認識](#)するよう Web サーバを構成する必要があります。

3.2.1 Web サーバ構成への Web ゲートウェイの追加

インターシステムズでサポートされているテクノロジーのページには、このバージョンの製品と互換性のある Web サーバがリストされています。[多くの場合](#)、InterSystems IRIS インストーラは Web ゲートウェイを Web サーバに自動的に追加できます。

Web ゲートウェイを Web サーバに手動で追加する手順は、オペレーティング・システム、使用する Web サーバ、およびユース・ケースによって大きく異なります。最初に考慮すべきことを以下に示します。

- ・ 一般に、インターシステムズでは Web サーバの Web ゲートウェイ拡張をバイナリのペアとして実装します。1 つのバイナリはコア・ランタイム機能を担当し、もう 1 つのバイナリは [Web ゲートウェイ管理ページ](#)の機能を担当します。管理バイナリはファイル名の末尾に `sys` を追加することで、ランタイム・バイナリと区別されます (例：`CSPa24.so` と `CSPa24Sys.so`)。どちらのバイナリも同じディレクトリに置く必要があります。
- ・ Apache (UNIX®/Linux/macOS) および Microsoft IIS (Windows) の場合、Web ゲートウェイを導入するには、Web サーバ独自の API を利用して、動的にロードされる拡張モジュール (Apache の場合は `.so` ファイル、IIS の場合は `.dll` ファイル) を追加するのが最も簡単です。Nginx (UNIX®/Linux/macOS または Windows) は、コンパイル時に

Web ゲートウェイ拡張のソース・コードを Web サーバに組み込むことによる Web ゲートウェイ拡張の追加のみをサポートします。

- ・ Nginx の Web ゲートウェイ拡張では、Web ゲートウェイの機能を実行する外部プロセスを呼び出します。このプロセスは、Network Service Daemon (NSD) と呼ばれます。Web ゲートウェイのオペレーションを Web サーバから切り離す必要がある場合、インターシステムズは Apache および Microsoft IIS でそれぞれ独自の API を使用して NSD Web ゲートウェイを導入するための拡張機能も提供しています。

重要 セキュリティ上の理由により、Web サーバのプロセスから NSD ファイルにアクセスすることはできません。NSD ファイルは、Web サーバのプロセスがアクセスできるディレクトリの外にあるファイル・システムの場所にインストールすることをお勧めします。

- ・ Common Gateway Interface (CGI) 拡張を使用する必要がある場合、インターシステムズは Apache および Microsoft IIS 向けに NSD Web ゲートウェイを構成するための CGI 実行可能ファイルも提供しています。
- ・ 選択する導入方法に関係なく、Web サーバのワーカ・プロセスが以下のディレクトリ (およびそのコンテンツ) にアクセスするための適切な権限を持つように、システムを構成する必要があります。
 - Web ゲートウェイ・バイナリのインストール・ディレクトリ。既定では、これらの場所は UNIX®/Linux/macOS では `/opt/webgateway`、Windows では `C:\¥Inetpub¥CSPGateway` です。
 - Web ゲートウェイの構成ファイル (`CSP.ini`) を含むディレクトリ。
 - Web ゲートウェイ・キャッシュの内容を `.dat` ファイルの形式で記録する、Web ゲートウェイのインストール・ディレクトリ内の `/temp` サブディレクトリ。
 - Web アプリケーションが処理する可能性のある静的ファイルのディレクトリ。CSP アプリケーションに関連付けられている静的ファイルは、`<IRISInstallDir>/csp/` パス内の対応するディレクトリに格納されます。`<IRISInstallDir>` は、アプリケーションをホストする InterSystems IRIS インスタンスのインストール・パスです。`<IRISInstallDir>/csp/broker` パスには、複数の組み込みのシステム・アプリケーションの静的ファイルが含まれます。

以下の表では、オペレーティング・システムと Web サーバに基づいて、Web ゲートウェイを Web サーバ構成に追加するための具体的な手順へのリンクを提供します。お使いの Web サーバに固有の詳細情報は、Web サーバの製品ドキュメントを参照してください。

オペレーティング・システム	Web サーバ	導入手順
UNIX®/Linux/macOS	Apache	<ul style="list-style-type: none"> 推奨 : Web ゲートウェイと連携させるための Apache の構成 SELinux などロック・ダウン・バージョンの Apache の場合 : ロック・ダウン Apache インストールへの Web ゲートウェイの追加 (および、このリストの他の関連ページ) NSD 導入の場合 : Apache の代替オプション および NSD (Network Service Daemon) の使用 (UNIX®/Linux/macOS) 特殊な構成および従来の構成 : Apache の代替オプション その他の考慮事項 : Apache Web サーバに関する考慮事項 (UNIX®/Linux/macOS)
UNIX®/Linux/macOS	nginx	<ul style="list-style-type: none"> Nginx の構築と構成 (UNIX®/Linux/macOS) および NSD (Network Service Daemon) の使用 (UNIX®/Linux/macOS)
Windows	IIS	<ul style="list-style-type: none"> 推奨 : Web ゲートウェイと連携させるための IIS の構成 NSD 導入の場合 : IIS 7 以降の代替オプション および NSD (Network Service Daemon) の使用 (Windows) 特殊な構成および従来の構成 : IIS 7 以降の代替オプション
Windows	nginx	<ul style="list-style-type: none"> Nginx の構築と構成 (Windows) および NSD (Network Service Daemon) の使用 (Windows)

3.2.2 Web サーバが Web ゲートウェイを介してルーティングする要求の指定

InterSystems Web アプリケーションを提供するには、Web サーバは Web ゲートウェイ拡張を呼び出して、そのアプリケーションに指定された URL パスで受け取った HTTP 要求を処理する必要があります。この構成手順は、Web サーバによって異なります。

お使いの Web サーバに関係なく、インターシステムズは Web ゲートウェイが処理すべき要求を指定するための 2 つの方法を提供しています。

1. クライアントが指定されたパスで特定のファイル・タイプを要求したときに、Web ゲートウェイを呼び出します。InterSystems Web アプリケーション要求の多くは、末尾が以下のいずれかのインターシステムズ・ファイル・タイプ拡張子になっています。

```
.csp .cls .zen .cwx
```

最初の 3 つのファイル・タイプは、InterSystems IRIS で実行できるコードの種類を示します。最後のファイル・タイプ (.cwx) は、Web ゲートウェイ管理ページで排他的に使用するために予約されています。Web アプリケーションが応答の一部として静的ファイル (.jpg、.js など) を処理する場合、それらの静的ファイル・タイプの要求に対する応答として Web ゲートウェイを呼び出すように、Web サーバを構成することも必要です。

2. 指定されたパスでの任意の要求に対して、Web ゲートウェイを呼び出します。

ほとんどの REST API はファイルまたはファイル・タイプを指定せずに、エンドポイント・パスを公開します (例えば、InterSystems IRIS パフォーマンス・データを収集するためのエンドポイント : `/api/monitor/metrics`)。このようなアプリケーション

ンを提供するには、アプリケーションの URL パスでのすべての要求に対して、Web ゲートウェイを呼び出す必要があります。

以下の表は、Web ゲートウェイを呼び出す両方の方法を、サポート対象の Web サーバごとに簡単にまとめたものです。お使いの Web サーバの構成手順の詳細は、Web サーバの製品ドキュメントを参照してください。

Web サーバ	パスでのすべての要求のルーティング	特定のファイル・タイプのルーティング
Apache	<p><Location> ブロック内で指示文 <code>CSP On</code> および <code>CSP Off</code> を使用して Apache 構成ファイル (通常は <code>httpd.conf</code>) を編集します。例：</p> <pre><Location [/path]> CSP On </Location></pre> <p><code>[/path]</code> は相対 URL パスです。</p> <p>注釈 この Web サーバ指示文を <code><VirtualHost></code> ブロック内で使用することはできません。</p>	<p><Location> ブロック内で指示文 <code>CSPFileTypes</code> を使用して Apache 構成ファイル (通常は <code>httpd.conf</code>) を編集します。例：</p> <pre><Location [/path]> CSPFileTypes {xxx yyy ...} </Location></pre> <p><code>[/path]</code> は相対 URL パスで、<code>{xxx yyy ...}</code> は空白で区切られたファイル・タイプ拡張子のリストです。</p> <p>注釈 この Web サーバ指示文を <code><VirtualHost></code> ブロック内で使用することはできません。</p>
IIS	<p>Web ゲートウェイ・ネイティブ・モジュールを呼び出す、制限のない * (ワイルドカード) ハンドラ・マッピングを追加します。詳細は、“アプリケーション要求のハンドラ・マッピングの設定” を参照してください。</p>	<p>目的のファイル拡張子の、ファイル制限のあるハンドラ・マッピングを追加します。詳細は、“アプリケーション要求のハンドラ・マッピングの設定” を参照してください。</p>
nginx	<p>指示文 <code>CSP on</code> および <code>CSP off</code> を使用して nginx 構成ファイル (<code>nginx.conf</code>) を編集します。例：</p> <pre>location </path> { CSP on; }</pre> <p><code></path></code> は相対 URL パスです。詳細は、“NSD を呼び出すように Nginx を構成” を参照してください。</p>	<p>指示文 <code>CSPFileTypes</code> を使用して nginx 構成ファイル (<code>nginx.conf</code>) を編集します。例：</p> <pre>location </path> { CSPFileTypes <xxx yyy ...> }</pre> <p><code></path></code> は URL パスで、<code><xxx yyy ...></code> は空白で区切られたファイル・タイプ拡張子のリストです。詳細は、“NSD を呼び出すように Nginx を構成” を参照してください。</p>

Web サーバが Web ゲートウェイを呼び出す URL パスを選択する前に、Web サーバからアプリケーションを提供する InterSystems IRIS インスタンスにアプリケーション要求をルーティングするプロセス全体を検討することをお勧めします。これは、システムに複数の InterSystems IRIS アプリケーション・サーバが搭載されている場合、または組織に特定の URL 命名規則がある場合に特にお勧めです。詳細は、“[Web ゲートウェイを介して要求をルーティングするパスの選択](#)” を参照してください。

3.3 Web ゲートウェイから InterSystems IRIS インスタンスへの要求の送信

Web サーバが Web アプリケーションの要求を Web ゲートウェイに渡すと、Web ゲートウェイはそのアプリケーションのコードをホストする InterSystems IRIS インスタンスにその要求をルーティングします。

そのためには、Web ゲートウェイに以下の情報を提供する必要があります。

- ・ 各インスタンスのアプリケーション・サーバ・プロセスと通信するのに必要な接続情報。Web ゲートウェイはこの情報を [サーバ・アクセス・プロファイル](#) に保持します。
- ・ Web アプリケーション・パス (または、一連の Web アプリケーションのベース・パス) と、アプリケーションのコードをホストするインスタンスのサーバ・アクセス・プロファイルとのマッピング。Web ゲートウェイはパスのルーティング動作を [アプリケーション・アクセス・プロファイル](#) に保持します。

注釈 InterSystems IRIS インスタンスのアプリケーション・サーバ・プロセスは、従来の InterSystems Web アプリケーション・フレームワークにちなんで “CSP サーバ” と呼ばれることがあります。

インスタンスのアプリケーション・サーバ・プロセスへの接続を構成および監視する Web ゲートウェイ管理ページでは、これらのプロセスを単に “サーバ” と呼んでいます (Web ゲートウェイの Web サーバへの接続は、Web サーバ構成で決定されます)。

混乱を最小限に抑えるために、Web ゲートウェイのドキュメントでは、具体的にアプリケーション・サーバ・プロセスを指す必要がある場合を除き、InterSystems IRIS インスタンスを Web ゲートウェイ接続の有効なターゲットとして記載しています。

どの導入においても、すべての Web ゲートウェイで、[サーバ・アクセス・プロファイル](#) と [アプリケーション・アクセス・プロファイル](#) を構成するための同じ便利な Web インタフェース、Web ゲートウェイ管理ページが提供されています。Web ゲートウェイ管理ページでは、[グローバルな既定のパラメータの構成](#)、[Web ゲートウェイ接続の監視](#)、[テスト接続](#) などを行うこともできます。

Web ゲートウェイは構成情報を [CSP.ini](#) に保持します。

重要 [CSP.ini](#) ファイルを直接編集しなければならないことがあるコンテナ化された導入を除き、[CSP.ini](#) ファイルへのアクセスを制限して、すべての Web ゲートウェイ構成を Web ゲートウェイ管理ページまたは [Web ゲートウェイ・レジストリ・メソッド](#) を使用して実行することをお勧めします。

3.3.1 InterSystems IRIS インスタンスの Web ゲートウェイへの接続

Web ゲートウェイが InterSystems IRIS インスタンスのアプリケーション・サーバ・プロセスへの接続を確立し、維持できるようにするには、各インスタンスの接続情報 (IP アドレス、スーパーサーバ・ポート番号、接続セキュリティの認証情報など) を提供するサーバ・アクセス・プロファイルを定義する必要があります。“[InterSystems IRIS インスタンスのサーバ・アクセス・プロファイルの定義](#)” を参照してください。

3.3.2 インスタンスのアプリケーションとの関連付け

単一の Web ゲートウェイが、複数の InterSystems IRIS インスタンスに、そのインスタンス固有のアプリケーションの要求をルーティングすることができます (各インスタンスの管理ポータルなど)。さらに、Web ゲートウェイは、単一のアプリケーションの要求を、同じアプリケーション・コードをホストする複数のインスタンスに関連付けることができるため、負荷分散とフェイルオーバーが可能になります。

Web ゲートウェイは、アプリケーション・アクセス・プロファイルに基づいて宛先インスタンスを決定します。アプリケーション・アクセス・プロファイルは、要求で指定された URL パスと、1 つ以上の InterSystems IRIS インスタンス（サーバ・アクセス・プロファイルで示された）との間のリレーションシップを定義します。アプリケーション・アクセス・プロファイルには、Web ゲートウェイがそのパスのトラフィックを処理する方法に関するその他の詳細も含まれます。“[Web アプリケーション・パスのアプリケーション・アクセス・プロファイルの定義](#)”を参照してください。

アプリケーション・アクセス・プロファイルを定義する URL パスを選択する前に、Web サーバからアプリケーションを提供する InterSystems IRIS インスタンスにアプリケーション要求をルーティングするプロセス全体を検討することをお勧めします。これは、システムに複数の InterSystems IRIS アプリケーション・サーバが搭載されている場合、または組織に特定の URL 命名規則がある場合に特にお勧めです。詳細は、“[Web ゲートウェイを介して要求をルーティングするパスの選択](#)”を参照してください。

3.4 すべての接続の保護

要求は、Web クライアントから InterSystems IRIS アプリケーション・サーバに送信される途中で、次の 2 つの TCP 接続を通過します。

1. Web クライアントと Web サーバ間の接続。Web クライアントと Web サーバ間の接続を保護するには、つまり、HTTPS を使用するには、Web サーバに SSL/TLS を構成する必要があります。詳細は、Web サーバのドキュメントを参照してください。
2. Web ゲートウェイと InterSystems IRIS アプリケーション・サーバ間の接続。インターシステムズでは、この接続を保護する複数の方法をサポートしています。詳細は、“[InterSystems IRIS への Web ゲートウェイ接続の保護](#)”を参照してください。

また、[ページにアクセスできる IP アドレスを定義](#)し、[ページにアクセスするための認証を要求](#)することで、Web ゲートウェイ管理ページへのアクセスを保護する必要があります。

3.5 Web ゲートウェイ接続のデコミッション

システムから InterSystems IRIS インスタンスをアンインストールした場合、関連するすべての構成要素を削除して、そのインスタンスへの Web ゲートウェイ接続をデコミッションする必要があります。そのためには、以下の操作を実行します。

1. 不要になったアプリケーション・パスの [Web ゲートウェイ・アプリケーション・アクセス・プロファイル](#)を削除します。例えば、インスタンスの名前が IRISserv1 の場合、/irisserv1 パス（[インスタンス接頭語](#)）のアプリケーション・アクセス・プロファイルと、/irisserv1/csp など下位のパスのアプリケーション・アクセス・プロファイルを削除します。
2. インスタンスが提供していたアプリケーションを、他の InterSystems IRIS インスタンスを使用して引き続き提供する場合、[対応するアプリケーション・アクセス・プロファイルのフィールドを更新](#)して、他のインスタンスのサーバ・アクセス・プロファイルを識別できるようにします（新しい InterSystems IRIS アプリケーション・サーバ用の[サーバ・アクセス・プロファイルの追加](#)が行われることがあります）。
3. アンインストールした InterSystems IRIS インスタンスに対応する [Web ゲートウェイのサーバ・アクセス・プロファイル](#)を削除します。Web ゲートウェイがサーバ・アクセス・プロファイルの削除を許可するには、サーバ・アクセス・プロファイルへのすべての参照をアプリケーション・アクセス・プロファイルから削除する必要があることに注意してください。
4. 不要になったパスの Web サーバ構成指示文を削除します。例えば、インスタンスの名前が IRISserv1 の場合、Apache Web サーバの構成ファイルから <Location /irisserv1> ブロックを削除すると共に、/irisserv1/csp など下位のパスの <Location> ブロックを削除します。インスタンスのインストール時に [Web サーバ自動構成オプション](#)を選択し、Apache Web サーバを使用している場合、[この例](#)に示されているように、コメントでインスタンスに固有の構成指示文の開始と終了が示されます。

Web サーバによる Web ゲートウェイの使用を完全に中止する場合は、Web ゲートウェイに関連するすべての要素を Web サーバ構成から削除します。その後、Web ゲートウェイ・ファイルを削除します。既定では、これらのファイルは `/opt/webgateway/bin` (UNIX®/Linux/macOS) または `C:\inetpub\CSPGateway` (Windows) に格納されています。

4

スタンドアロン Web ゲートウェイのインストール

Web アプリケーションをホストする InterSystems IRIS® インスタンスとは別のマシンに Web サーバが配置されている場合、InterSystems Web ゲートウェイをスタンドアロン・コンポーネントとして Web サーバ・マシンにインストールできます。

このページでは、WRC コンポーネントのダウンロード・ページ (<https://wrc.intersystems.com/wrc/coDistGen.csp>) から入手できるインストール・キットを使用して、スタンドアロン Web ゲートウェイをインストールする手順について説明します。

注釈 このようなシナリオで Web サーバ・マシンにスタンドアロン Web ゲートウェイを導入する方法として、他にも以下のようなものがあります (他の箇所で説明しています)。

- ・ InterSystems IRIS インストーラを使用し、カスタム・セットアップ・タイプ (Windows の場合) または Web サーバ・セットアップ・タイプを選択して、Web ゲートウェイのみをインストールする。
- ・ [webgateway](#) する。

多くの場合、スタンドアロン Web ゲートウェイ・インストーラは、Web ゲートウェイを使用するように Web サーバを自動構成できます。そうでない場合は、[Web ゲートウェイを Web サーバ構成に手動で追加](#)するために必要なファイルがインストール・メディアで提供されます。

スタンドアロン Web ゲートウェイ・インストーラでは、InterSystems IRIS インスタンスの接続情報を指定することもできます。スタンドアロン Web ゲートウェイ・インストーラはこの情報を使用して、このインスタンスの[サーバ・アクセス・プロファイル](#)と、[インスタンス接頭語 URL パスのアプリケーション・アクセス・プロファイル](#)を作成します。インストーラは、Web ゲートウェイを Web サーバに自動接続することに成功すると、インスタンス接頭語 URL パスを含め、インスタンスの Web アプリケーションに関連するすべての URL パスに対して Web ゲートウェイを呼び出すように Web サーバを構成します。

4.1 手順 1 : サポート対象 Web サーバのインストール

Web ゲートウェイをインストールする前にサポート対象 Web サーバがインストールされ、実行されている場合、Web ゲートウェイのインストール・キットで、Web ゲートウェイを使用するようにサポート対象 Web サーバを自動的に構成できます。これは、ほとんどのユース・ケースで推奨されるアプローチです。

自動構成がサポートされる Web サーバは、お使いのシステムによって異なります。

- ・ UNIX®/Linux/macOS の場合 : 既定のインストール場所に Apache httpd Web サーバをインストールします。Apache httpd をインストールするための詳細なガイダンスは、Developer Community の記事、<https://community.intersystems.com/post/how-install-apache-iris-supported-operating-systems> を参照してください。
- ・ Windows の場合 : Microsoft インターネット・インフォメーション・サービス (IIS) を有効にします。手順については、“[IIS の有効化](#)” を参照してください。

重要 オペレーティング・システムに指定された Web サーバ以外の Web サーバをインストールする場合、または Web ゲートウェイのインストールが完了するまで指定された Web サーバをインストールしない場合は、[Web サーバを手動で構成](#)する必要があります。

お使いのシステムによっては、Web サーバのインストールで自動構成を有効にするために、追加の条件を満たさなければならないことがあります。”[自動構成が可能な場合](#)”を参照してください。

4.2 手順 2：インストール・キットのダウンロード

スタンドアロン Web ゲートウェイのインストール・キットは、WRC コンポーネントのダウンロード・ページ (<https://wrc.intersystems.com/wrc/coDistGen.csp>) で配布されています。

[名前] 列に `web gateway` と入力し、[OS] および [Arch] 列を使用して、システムに適した Web ゲートウェイ・インストール・キットを見つけます。

UNIX®、Linux、および macOS システムの場合、インターシステムズでは、Web ゲートウェイ・インストール・キットを、インストール・スクリプト (`GatewayInstall`) を含む圧縮された tarball (`.tar.gz`) として配布しています。Windows システムの場合は、インストーラを実行可能ファイルとして提供しています。

4.3 手順 3：(UNIX®/Linux/macOS のみ) インストール・キット・ファイルの抽出

tarball のコンテンツを解凍して、一時的なファイルシステムの場所に抽出します。元の権限を保持しながらこの操作を行うには、以下のコマンドを発行します (<WebGatewayKit> を、ダウンロードした tarball の名前に置き換えます)。

```
tar zpxvf <WebGatewayKit>.tar.gz
```

ファイルは、同じ名前 (<WebGatewayKit>/) のディレクトリに抽出されます。<WebGatewayKit>/ のコンテンツには以下が含まれます。

- ・ `install/` サブディレクトリ。これには以下が含まれます。
 - インストール・スクリプト `GatewayInstall`。このページの後続の手順で、このスクリプトを使用して Web ゲートウェイをインストールする方法について説明します。
 - `nginx/` サブディレクトリ。このサブディレクトリには、Web ゲートウェイを含む [Nginx の構築を手動で構成](#)するために必要なファイルが含まれます。
- ・ オペレーティング・システムとアーキテクチャに対応する名前を持つサブディレクトリ、/`<platformCode>`。このサブディレクトリには、Web ゲートウェイを使用するように [Apache httpd Web サーバを手動で構成](#)するために必要なファイルが含まれます。

注釈 `GatewayInstall` スクリプトでは、Web ゲートウェイを使用するように既存の Apache httpd Web サーバを自動的に構成できます。このため、手動構成が必要になるのは、[NSD を使用する Web ゲートウェイを導入](#)する場合など、特殊な導入シナリオの場合のみです。

4.4 手順 4 : (UNIX®/Linux/macOS のみ) root としてのログイン

GatewayInstall スクリプトを実行するには、コマンド行セッションを root 特権を持つユーザとして実行する必要があります。(別のユーザ・アカウントから root として su を実行すれば十分です。)

4.5 手順 5 : インストーラの実行

4.5.1 UNIX®/Linux/macOS

GatewayInstall スクリプトを実行して、インストール・プロセスを開始します。それには、<WebGatewayKit>/install/ ディレクトリに移動して、以下のコマンドを発行します。

```
./GatewayInstall
```

GatewayInstall により、一連のインタラクティブ・プロンプトが提供されます。プロンプトに従って、以下の手順を実行します。

1. インストールされている Web サーバのタイプを指定します。
 - ・ Apache httpd がインストールされていて、Web ゲートウェイを使用するよう Apache をインストーラで自動構成したい場合は、[Apache] を選択します。変更された Web サーバ構成には、組み込みの InterSystems IRIS Web アプリケーションに関連付けられている相対 URL パスに送信される要求に対して Web ゲートウェイを呼び出す指示文が含まれます。
 - ・ 他のサポート対象 Web サーバがインストールされている場合、または Web サーバがインストールされていない場合は、[] を選択します。このオプションを選択すると、インストーラはサポート対象 Web サーバを手動で構成するのに必要なファイルを、次の手順で指定するディレクトリにコピーします。また、後続の手順で指定する接続情報を使用して、Web ゲートウェイを構成します。Web サーバの構成は試行されません。

注釈 一部のプラットフォームでは、GatewayInstall がシステム・プラットフォーム名を表す文字列の入力を求めることもあります。この文字列は、インストール・キットのファイル名の末尾に示されます。例えば、インストール・キットが WebGateway-<version>-lnxrh8x64.tar.gz 形式の場合、lnxrh8x64 がプラットフォーム名です。cplatname スクリプトを使用してこの値を取得することもできます。それには、インストール・キットの /install サブディレクトリに移動して、以下のコマンドを発行します。

```
# ./cplatname identify
```

2. Web ゲートウェイ・ファイルのインストール先ディレクトリを指定するか、既定のディレクトリ(通常は、/opt/webgateway/)を受け入れます。ディレクトリがまだ存在しない場合は、ディレクトリを作成するかどうかを尋ねられます。
3. Web ゲートウェイを接続する InterSystems IRIS インスタンスに関する情報を提供します。
 - a. インスタンスが配置されているマシンのホスト名。
 - b. インスタンスが Web ゲートウェイ接続を受け入れるスーパーサーバ・ポート。
 - c. インスタンスの**構成名 0**。(既定では、これは小文字のインスタンス名です。)構成名は、特にそのインスタンスに要求をルーティングするために使用できる接頭語 URL パスとして機能します。
 - d. インスタンスの**セキュリティの設定**。
 - ・ インスタンスが安全でない Web ゲートウェイ接続を許可する場合は、[] を選択します。

- ・ インスタンスが、インスタンスの CSPSystem ユーザ・アカウント認証情報を使用して Web ゲートウェイ接続を認証する必要がある場合は、[] を選択します。
- ・ インスタンスが、インスタンスの CSPSystem ユーザ・アカウント認証情報を使用して Web ゲートウェイ接続を認証する必要があり、インスタンスのセキュリティ設定がロック・ダウンされている場合は、[] を選択します。

[] または [] を選択する場合は、インスタンスの CSPSystem ユーザ・アカウントのパスワードも指定する必要があります。

インストーラはこの情報を使用して、Web ゲートウェイの構成ファイル (CSP.ini) を編集し、インスタンスに接続するためのサーバ・アクセス・プロファイルと、インスタンスのアプリケーション (インスタンスの構成名に基づくものを含む) のアプリケーション・アクセス・プロファイルを作成します。

インストーラは、Web ゲートウェイを使用するように Web サーバを正常に自動構成すると、この情報を使用して Web サーバも構成します。Web サーバ構成には、構成名に基づいて、インスタンス接頭語 URL パスに送信される要求に対して Web ゲートウェイを呼び出す指示文が含まれます。

注釈 インストーラは、これらのプロンプトに対する応答を必要とします。Web ゲートウェイが提供する InterSystems IRIS インスタンスをインストールする前に、Web ゲートウェイをインストールする場合は、既定値を受け入れるか、プレースホルダ値を指定します。インストール後に、不要な構成項目を削除したり、既存のインスタンスに接続するように構成項目を変更することができます。

4. 構成の詳細を確認して、インストールを開始します。

4.5.2 Windows

実行可能ファイルを実行して、スタンドアロン Web ゲートウェイ・インストーラ・ウィザードを起動します。ウィザードにより、一連のインタラクティブ・プロンプトが提供されます。プロンプトが表示されたら、以下の手順を実行します。**[次へ >]** を選択すると、次のプロンプトに進みます。

1. セットアップ・タイプを選択します。

- ・ **[完了]** を選択した場合、すべての Web ゲートウェイ・ファイルが既定の場所にインストールされます。つまり、**カスタム・セットアップ・タイプ** (次に説明します) と同じ効果があり、すべてのインストール機能が含まれ、既定の場所は上書きされません。**IIS が有効化**されている場合、インストーラは Web ゲートウェイを使用するように IIS を構成します。
- ・ **[カスタム]** を選択した場合、ファイルの特定のサブセットをインストールしたり、それらをインストールするディレクトリを選択したりするインストール機能を、含めたり、除外したりすることができます。
 - Web ゲートウェイを CGI 拡張として実装するファイルをインストールするには、**[Web ゲートウェイ CGI モジュール]** を選択します。これらのファイルは、組織で CGI 拡張を排他的に使用する必要がある場合を除き、通常は不要です。
 - Web ゲートウェイを使用するように **IIS Web サーバを構成**するのに必要なファイルをインストールするには、**[IIS 用 Web ゲートウェイ]** を選択します。IIS が有効化されていて、この項目が選択されている場合、インストーラは Web ゲートウェイを使用するように IIS を構成します。構成されると、IIS は Web ゲートウェイを呼び出して、組み込みの InterSystems IRIS Web アプリケーションに関連付けられている相対 URL パスに送信される要求を処理します。

注釈 IIS が有効かどうかにかかわらず、インストーラは常に、**[IIS 用 Web ゲートウェイ]** のファイルを既定の IIS ディレクトリ (C:\inetpub\¥) 内にインストールします。

- Web ゲートウェイ・インストール・ディレクトリ内の組み込みの InterSystems IRIS Web アプリケーションによって使用される静的ファイルをインストールするには、**[Web ゲートウェイ用の静的ファイル]** を選択します。こ

れにより、Web サーバ・マシン上の中央のファイルシステムの場所からこれらの静的ファイル进行处理するように Web サーバを構成できます。(既定では、各 InterSystems IRIS インスタンスは独自のディレクトリから、Web アプリケーションに関連付けられているファイルを提供します。)

- Web ゲートウェイを Apache httpd Web サーバの拡張として実装するファイルをインストールするには、**[Apache HTTPD サーバ用 Web ゲートウェイ・モジュール]** を選択します。

注意 インターシステムズは、Windows システムでの Apache httpd Web サーバの使用のサポートを終了しました。

2. インストーラで、InterSystems IRIS インスタンスに接続するように Web ゲートウェイを構成するには、**[サーバに接続するように Web ゲートウェイを構成]** を選択します。このオプションを選択した場合、インスタンスに関する以下の接続情報を指定できます。

- ・ **アプリケーション名** : インスタンスが提供するアプリケーションの要求をルーティングする URL パスとして使用できるカスタム名。ほとんどの場合、**インスタンスの** を指定することをお勧めします。(既定では、これは小文字のインスタンス名です。)
- ・ **IRIS サーバ・アドレス** : インスタンスが配置されているマシンのホスト名。
- ・ **IRIS サーバ・ポート番号** : インスタンスが Web ゲートウェイ接続を受け入れるスーパーサーバ・ポート。
- ・ **接続パスワード(オプション)** : インスタンスの CSPSystem ユーザ・アカウントのパスワード。(CSPSystem ユーザ・アカウントは、**[通常]** または **[ロック・ダウン]** の **初期セキュリティ設定** でインストールされたインスタンスの Web ゲートウェイを認証するために使用されます。)

インストーラはこの情報を使用して、Web ゲートウェイの構成ファイル (**CSP.ini**) を編集し、インスタンスに接続するための **サーバ・アクセス・プロファイル** と、インスタンスのアプリケーション (指定した **[アプリケーション名]** に基づくものを含む) の **アプリケーション・アクセス・プロファイル** を作成します。

インストーラは、Web ゲートウェイを使用するように Web サーバを正常に自動構成すると、このインスタンス情報を使用して Web サーバも構成します。Web サーバ構成には、構成名に基づいて、インスタンス接頭語 URL パスに送信される要求に対して Web ゲートウェイを呼び出す指示文が含まれます。

3. **[インストール]** を選択してインストールを開始します。

4.5.2.1 Web ゲートウェイ・インストールの変更、修復、削除

システムに既にスタンドアロン Web ゲートウェイがインストールされている場合、スタンドアロン Web ゲートウェイ・インストール実行可能ファイルを使用してそのインストールを **[変更]**、**[修復]**、または **[削除]** できます。

- ・ **[変更]** では、Web ゲートウェイ・ファイルの特定のサブセットを表すインストール機能を含めたり除外したりして、インストールをカスタマイズできます。これらのインストール機能は、新規インストールの **カスタム・セットアップ・タイプ** で利用できる機能と同じです。**[変更]** モードでは、InterSystems IRIS インスタンスの接続の詳細を構成 (または再構成) することもできます。

IIS が有効化されている場合、Web ゲートウェイ・インストールを **変更** して **[IIS 用 Web ゲートウェイ]** 機能を含めると、インストーラは新規インストールと同様に Web ゲートウェイを使用するように IIS を構成します。InterSystems IRIS インスタンスの接続の詳細を指定した場合、インストーラは、新規インストールと同様にそのインスタンスの要求进行处理するように Web ゲートウェイと IIS を構成します。

- ・ **[修復]** では、元のインストール設定を使用して、削除されたまたは破損したファイルをリストアし、スタンドアロン Web ゲートウェイを修復できます。元のインストールに **[IIS 用 Web ゲートウェイ]** 機能は含まれていたが、IIS がまだ有効化されていなかった場合、**[修復]** モードでは再び Web ゲートウェイを使用するように IIS を構成しようとします。元のインストールで InterSystems IRIS インスタンスの接続の詳細は指定されていたが、IIS がまだ有効化されていなかった場合、**[修復]** モードでは再びそのインスタンスの要求进行处理するように Web サーバを構成しようとします。

- ・ **【削除】** では、スタンドアロン Web ゲートウェイをアンインストールできます。

5

Web ゲートウェイでの Web サーバ構成の拡張

InterSystems Web ゲートウェイのファイルをインストールしたら、Web ゲートウェイを拡張として Web サーバ構成に追加する必要があります。追加したら、Web サーバは Web ゲートウェイを呼び出して、InterSystems IRIS® Web アプリケーションへの要求を処理できます。

このページでは、Web ゲートウェイ拡張を各サポート対象 Web サーバの Web サーバ構成に追加するための推奨方法について説明します。インターシステムズが実装している、Web ゲートウェイを介してルーティングする要求を指定する方法についても、Web サーバごとに概説します。(アプリケーション要求をルーティングする方法を検討する際は、“Web ゲートウェイを介して要求をルーティングする URL パスの選択”を参照してください。)

多くの場合、インストーラは Apache (UNIX®/Linux/macOS の場合) または IIS (Windows の場合) に対して、このページの構成手順を自動的に実行できます。その後、作成された構成を、お使いのシステムのニーズに合わせて容易にカスタマイズできます。

Web サーバを構成したら、Web ゲートウェイ管理ページを使用して、InterSystems IRIS インスタンスに接続し、それらにアプリケーション・パスを関連付けます。

5.1 考慮すべきファイル

5.1.1 Web ゲートウェイ・ファイル

Web ゲートウェイ・ファイルは、次のいずれかの場所にインストールされます。

- ・ UNIX®/Linux/macOS 上の `/opt/webgateway/bin`。InterSystems IRIS インストールへの変更に影響されないために、InterSystems IRIS のインストール・ディレクトリではなく、この共通の場所で Web ゲートウェイ・ファイルを使用することをお勧めします。
- ・ Windows 上の `C:\inetpub¥` (インターネット・インフォメーション・サービス (IIS) が構成されている場合)。InterSystems IRIS インストールへの変更に影響されないために、InterSystems IRIS のインストール・ディレクトリではなく、この共通の場所で Web ゲートウェイ・ファイルを使用することをお勧めします。
- ・ 以下の場合は、`<installDir>/csp/bin` または `<installDir>/bin` :
 - `<installDir>` に InterSystems IRIS インスタンスがインストールされている
 - Web サーバが構成されていない。
- ・ 以下の場合は、`<installDir>` または `<installDir>/bin` :
 - `<installDir>` にスタンドアロン Web ゲートウェイがインストールされている

- Web サーバが構成されていない。

一般に、インターシステムズでは Web ゲートウェイをバイナリ・ファイルのペアとして実装します。1 つのファイルはランタイム機能を実装し、もう 1 つのファイルは [Web ゲートウェイ管理ページ](#) を実装します。管理バイナリはファイル名の末尾に `sys` を追加することで、ランタイム・バイナリと区別されます。

5.1.2 Web サーバ・ファイル

Web サーバのインストール場所は、Web サーバやオペレーティング・システムによって異なります。Web サーバがインストールされている場所を見つけるには、Web サーバのドキュメントを参照してください。

- Apache : <https://httpd.apache.org/docs/2.4/>
- IIS : <https://learn.microsoft.com/en-us/iis/get-started/introduction-to-iis/iis-web-server-overview>
- Nginx : <http://nginx.org/en/docs/>

注釈 Web ゲートウェイのドキュメントでは、わかりやすくするために、Web サーバのインストール場所（通常、既定の場所）が例で指定されていることがあります。この場所は、お使いのシステム上の場所と一致しないことがあります。その場合は、システム上のファイルの場所に置き換えてください。

5.1.3 静的ファイル

必要なすべてのファイル・タイプについて、Web ゲートウェイがアプリケーションの相対 URL パスに送信される要求を処理できるように Web サーバが構成されている場合、InterSystems IRIS インスタンスは Web アプリケーション応答の一部として静的ファイルを処理できます。

InterSystems IRIS CSP Web アプリケーションが処理する静的ファイルは、Web アプリケーションに対応する `<installDir>/csp/` ディレクトリに配置されます。`<installDir>` は、InterSystems IRIS インスタンスのインストール・ディレクトリです。

以下に例を示します。

- `<installDir>/csp/broker` には、複数の組み込みのシステム・アプリケーションで共有されるファイルが含まれます。
- `<installDir>/csp/sys` には、管理ポータルで使用するファイルが含まれます。

既定では、InterSystems IRIS CSP Web アプリケーションは、関連するディレクトリから [静的ファイル自体を処理するように構成](#)されています。

ただし、Web アプリケーションは、Web サーバが代わりにこれらの静的ファイルを処理できるようにすることもできます。例えば、1 つの Web サーバで複数のリモート InterSystems IRIS インスタンスを処理しているシステムがある場合、この Web サーバ マシンにとってローカルな共通の場所からこれらのファイルを提供する方が適切な可能性があります。ただし、これにより問題が生じる可能性があることに注意してください。例えば、共通の Web サーバが異なるバージョンの InterSystems IRIS を提供する場合、同じファイルの 2 つの異なるバージョン間で競合が発生する可能性があります。

InterSystems IRIS Web アプリケーションの静的ファイルを処理およびキャッシュするように Web サーバを構成するには、アプリケーションのパス（アプリケーションの静的ファイルの要求の送信先）と、Web サーバが適切なアクセス権を持っており、静的ファイルが保存されているローカル・ファイル・システムの場所との間にマッピングを作成する必要があります。

Web サーバごとのこのオプションのマッピング作成方法については、Web ゲートウェイのドキュメントの各所で説明しています。Web サーバが自ら管理するローカル・ファイル・システムの場所の静的ファイルを処理するように Web サーバを構成する方法の詳細は、お使いの Web サーバのドキュメントを参照してください。

5.2 UNIX®/Linux/macOS 向けの Apache

Apache HTTP Web サーバは Apache グループによって提供されており、<http://www.apache.org> から無料でダウンロードできます。多くのシステムは、Apache を事前にインストールおよび構成して、すぐに使用できる状態で出荷されます。サポート対象 UNIX® または Linux システムに Apache をインストールする方法については、次の Developer Community の記事を参照してください：https://docs.intersystems.com/irislatest/csp/docbook/DocBook.UI.Page.cls?KEY=GCGI_web-server#GCGI_ux_apache。

注釈 macOS では、InterSystems IRIS インストーラは事前インストールされているバージョンの Apache を自動的に構成することはできません。ただし、Homebrew (<https://brew.sh/>) を使用してインストールされた Apache のバージョンを自動的に構成することはできます。

このページでは、インターシステムズが Apache のネイティブ API を使用して実装する動的モジュール `csp_module_sa` を使用して、Apache で Web ゲートウェイを導入する方法を説明します。

`csp_module_sa` モジュールは、2 つの動的共有オブジェクト (`.so`) ファイルを使用して Web ゲートウェイを実装します。

- ・ `CSPa24.so` – ランタイム・バイナリ。
- ・ `CSPa24Sys.so` – Web ゲートウェイ管理バイナリ。

これが Apache の推奨導入オプションで、ほとんどのユース・ケースで有効です。

注釈 このモジュールではステート認識接続 (保持モード 1) を使用しないでください。

セキュリティが制限されたバージョンの Apache (SELinux を搭載した RHEL など) を使用するシステムで Web ゲートウェイを構成する場合は、“[ロック・ダウン Apache インストールへの Web ゲートウェイの追加](#)” で補足手順を参照してください。“[Apache の代替オプション](#)” では、[Network Service Daemon \(NSD\)](#) を使用するオプションを含め、Apache で Web ゲートウェイを導入するその他のオプションについて説明しています。

“[Apache に関する考慮事項](#)” では、Web ゲートウェイに接続されている Apache Web サーバの管理に関するガイダンスを提供しています。詳細なガイダンスは、Apache のドキュメント (<https://httpd.apache.org/docs/>) を参照してください。

5.2.1 Apache が共有オブジェクト・モジュールを管理できることの確認

Apache 独自の API を使用して Apache 構成に Web ゲートウェイを追加する前に、Apache のビルドに共有オブジェクトを管理するための組み込みのモジュール (`mod_so`) が含まれていることを確認します。

Red Hat Linux でこのチェックを実行するには、コマンド行から以下のコマンドを発行します (オペレーティング・システムに対応するコマンドを選択します)。

RHEL

```
httpd -l
```

Ubuntu/SUSE

```
apache2 -l
```

このコマンドは、現在の Apache インストールに含まれているモジュールのリストを示します。`mod_so` がこのリストに含まれていない場合は、Apache のドキュメント (<https://httpd.apache.org/docs/2.4/>) で提供されている手順に従って、このモジュールを含む Apache の新しいビルドをコンパイルします。

5.2.2 Web サーバ構成への Web ゲートウェイ・モジュールの追加

1. Web ゲートウェイ・モジュール・ファイルのファイル・システムの場所を記録しておきます。これらの手順では、最も一般的な場所である `/opt/webgateway/bin` を使用します。
2. Apache Web サーバ構成ファイル (`httpd.conf`) をテキスト・エディタで開きます。このファイルは、Apache のインストール・ディレクトリの `/conf` サブディレクトリ内にあります。最も一般的なインストール場所は次のとおりです。

RHEL

`/etc/httpd/conf`

Ubuntu/SUSE

`/usr/apache2/conf`

3. Web ゲートウェイ・モジュールを追加するファイルの最後に指示文を追加します。これらの指示文では、以下を行います。
 - ・ `LoadModule` 指示文を使用して、モジュールの名前と Web ゲートウェイのランタイム機能用の `.so` ファイルの場所を指定し、モジュールをロードします。
 - ・ `CSPModulePath` 指示文を発行し、Web ゲートウェイのランタイム機能用の `.so` ファイルを含むディレクトリのフル・パスを指定します。
 - ・ `CSPConfigPath` 指示文を発行し、Web ゲートウェイの管理機能用の `.so` ファイルを含むディレクトリのフル・パスを指定します。
 - ・ Web ゲートウェイ・バイナリを含むディレクトリのアクセス設定を構成します。そのためには、以下のようにそのディレクトリに `<Directory>` 指示文ブロックを追加します (この例では、最も一般的なインストール場所を想定しています)。

```
<Directory "/opt/webgateway/bin">
    AllowOverride None
    Options MultiViews FollowSymLinks ExecCGI
    Require all granted
    <FilesMatch "^(log|ini|pid|exe)$">
        Require all denied
    </FilesMatch>
</Directory>
```

4. オプション : Web サーバで Web アプリケーションの静的ファイルを処理する場合は、該当する `/csp` パスを Web サーバからアクセスできるファイル・システムの場所にマップする指示文を追加します。Apache `Alias` 指示文は、そのための方法の 1 つを提供します。

例えば、アプリケーション `/dashboard` はリモートの InterSystems IRIS インスタンスによって提供されるが、アプリケーションの静的ファイルは Web サーバ・マシンのディレクトリ `C:\iris\csp\dashboard` から提供したい場合は、以下の指示文を追加できます。

```
Alias /dashboard/ "C:/iris/csp/dashboard"
```

5. Web ゲートウェイを呼び出して、InterSystems IRIS アプリケーションへの要求を処理する指示文を発行します。このために、以下の指示文が用意されています。
 - ・ `CSPFileTypes`。指定した一連のファイル・タイプの要求に対して Web ゲートウェイを呼び出します。(REST API エンドポイントは通常、ファイルではなくパスであるため、REST アプリケーションの要求を呼び出すには、この指示文では不十分です。)
 - ・ `CSP On` と `CSP Off`。Web ゲートウェイをすべての要求のハンドラとして有効または無効にします。

<Location> ブロック内でこれらの指示文を発行することで、Web サーバを細かく構成し、InterSystems IRIS Web アプリケーションに対応する相対 URL パスに対してのみ Web ゲートウェイを呼び出すことができます。

以下の場所ブロックは、iris3 という InterSystems IRIS インスタンスの管理ポータル宛ての要求に対してのみ Web ゲートウェイを呼び出します。

```
<Location "/iris3/csp/sys/">
  CSP On
</Location>
```

注釈 InterSystems IRIS インスタンスにアクセスせずに、Web ゲートウェイの管理ページにアクセスするには、/csp パスに対して Web ゲートウェイを有効にします。

6. httpd.conf ファイルを保存します。
7. Apache を再起動して、構成の変更を有効にします。

インストーラは、Apache Web サーバを自動構成する際にこの手順を使用します。例については、“[自動構成例](#)”を参照してください。

5.3 Windows 向けの Microsoft インターネット・インフォメーション・サービス (IIS)

Windows の多くの配布では、Microsoft IIS (<https://www.iis.net/>) が事前にインストールされています。ただし、通常は既定で無効になっています。“[IIS の有効化](#)”を参照してください。

このページでは、IIS 独自の API を使用して、Web ゲートウェイをネイティブ・モジュールとして導入する方法について説明します。

Web ゲートウェイ・ネイティブ・モジュールは、2 つのダイナミック・リンク・ライブラリ (.dll) ファイルを使用して Web ゲートウェイを実装します。

- ・ CSPms.dll – ランタイム・バイナリ
- ・ CSPmsSys.dll – Web ゲートウェイ管理バイナリ

これが推奨される導入オプションで、ほとんどのユース・ケースで有効です。

“[IIS 7 以降の代替オプション \(Windows\)](#)”では、ISAPI を使用するオプションや [Network Service Daemon \(NSD\)](#) を使用するオプションを含め、IIS で Web ゲートウェイを導入するその他のオプションについて説明しています。

5.3.1 IIS の有効化

IIS を有効にするには、次の手順に従います。

1. [Windows の機能の有効化または無効化] を探すか、[コントロール パネル] を開き、[プログラム]→[プログラムと機能]→[Windows の機能の有効化または無効化] を選択して、[Windows の機能] マネージャを開きます。
2. [インターネット インフォメーション サービス] を選択します。
3. ドキュメント・リンクのリダイレクトを有効にするには、IIS 内で [HTTP リダイレクト] も有効にする必要があります。有効にするには、[インターネット インフォメーション サービス]→[World Wide Web サービス]→[HTTP 共通機能]→[HTTP リダイレクト] が選択されていることを確認します。

4. サポートされている IDE でデバッグ・セッションを有効にするには、IIS 内で [WebSocket プロトコル] も有効にする必要があります。有効にするには、[インターネット インフォメーション サービス]→[World Wide Web サービス]→[アプリケーション開発機能]→[WebSocket プロトコル] が選択されていることを確認します。
5. [OK] をクリックします。

インストーラで、IIS 経由で Web アプリケーションを提供するように新しいまたはアップグレードされた InterSystems IRIS インスタンスを自動構成する場合は、インストールの前に IIS を有効にする必要があります。ただし、IIS の [HTTP リダイレクト] および [WebSocket プロトコル] 機能はいつでも有効にすることができます。

5.3.2 Web ゲートウェイ・コンポーネントに対する許可の設定

既定では、IIS は Web アプリケーションのユーザに対して、Web ドキュメント・ルート・ディレクトリ (通常は `C:\Inetpub\wwwroot`) 外にあるものへのアクセスを許可しません。以下のユーザ・グループが Web ドキュメント・ルート外のディレクトリにある Web アプリケーション・リソースにアクセスするには、そのディレクトリに対する読み取り、書き込み、実行権限を持っている必要があります。

- ・ `[machine_name]\IIS_IUSRS`。IIS ワーカー・プロセスおよび IIS で制御されるアプリケーション (Web ゲートウェイなど) が実行されるユーザ・グループ。
- ・ `[machine_name]\Users`

Web ゲートウェイ・ディレクトリ (通常は `C:\Inetpub\CSPGateway`) および Web アプリケーションが処理する静的ファイルを含むディレクトリに対するこれらの権限を、手動で設定する必要があります。そのためには、以下の操作を実行します。

1. Windows エクスプローラーで、ディレクトリの親ディレクトリに移動します。例えば、`C:\Inetpub\CSPGateway` ディレクトリを構成する場合は、`C:\Inetpub` に移動します。
2. ディレクトリ名を右クリックし、[プロパティ] を選択します。
3. [セキュリティ] タブを選択します。
4. [編集] を選択します。
5. [追加] を選択します。
6. [選択するオブジェクト名を入力してください] テキスト・ボックスに以下のように入力します。
`[machine_name]\IIS_IUSRS`
7. [名前の確認]、[OK] の順に選択します。
8. [グループまたはユーザー名] ウィンドウで `[machine_name]\IIS_IUSRS` を選択してから、以下の操作を実行します。
9. [許可] ウィンドウで [読み取りと実行] および [書き込み] 許可を割り当てます。
10. [適用]、[OK] の順に選択します。
11. `[machine_name]\Users` ユーザ・グループについて、上記の処理を繰り返します。

Web ゲートウェイ構成ファイル (`CSP.ini`) または Web ゲートウェイ・ログ・ファイル (`CSP.log`) がファイル・システム内の他の場所に配置されている場合は、`IIS_IUSRS` グループがこれらのファイルに対する完全な読み取りおよび書き込み権限を持っていることも確認する必要があります。

5.3.3 ネイティブ・モジュールの登録

Web ゲートウェイ・ネイティブ・モジュールを使用する前に、それらを IIS に登録する必要があります。そのためには、以下の操作を実行します。

1. [インターネット インフォメーション サービス (IIS) マネージャ] を開きます。
2. [接続] パネルから localhost 接続を選択します。
3. 接続のホーム・ページの [機能ビュー] で、[モジュール] 機能構成項目を開きます (ダブルクリック)。
4. [操作] パネルから [ネイティブ モジュールの構成...] 操作を選択します。
5. [登録] を選択し、[ネイティブ モジュールの登録] ダイアログに以下を入力します。
名前 : CSPms (または類似した名前)
パス : C:\Inetpub\CSPGateway\CSPms.dll
6. [OK] をクリックします。
7. [接続] パネルで localhost 接続のコンテンツを展開して、[サイト]→[Default Web Site] を選択します。
8. Web ゲートウェイ・モジュールが有効になっていることを確認します。
 - a. [Default Web Site ホーム] ページの [機能ビュー] で、[モジュール] 機能構成項目を開きます (ダブルクリック)。
 - b. [操作] パネルから [ネイティブ モジュールの構成...] 操作を選択します。
 - c. [ネイティブ モジュールの構成] ダイアログで、リストに [CSPms] がある場合はそれを選択し、[OK] を選択します。

5.3.4 Web アプリケーション・パスの構成

提供する InterSystems IRIS Web アプリケーションごとに、その相対パス (/csp または /irisinstance2) を IIS アプリケーションとして構成します。

IIS アプリケーションを構成すると、Web サーバが物理パスにアクセスするための適切な権限を持っている場合、Web サーバが静的ファイル进行处理するために必要なパス・マッピングが作成されます。

構成する各アプリケーション・パスに対して、以下の手順を実行します。

1. [インターネット インフォメーション サービス (IIS) マネージャ] を開きます。
2. [接続] パネルで localhost 接続のコンテンツを展開して、[サイト]→[Default Web Site] を選択します。
3. [操作] パネルから [アプリケーションの表示] を選択します。
4. [操作] パネルから [アプリケーションの追加...] を選択します。
5. [アプリケーションの追加] ダイアログに以下の情報を入力します。
 - ・ **エイリアス** : アプリケーションの相対ベース URL パス。例えば、インスタンス IRISInstance2 でホストされるアプリケーションの場合は /irisinstance2、アプリケーション /csp/sys とその兄弟アプリケーションの場合は csp です。
 - ・ **物理パス** : アプリケーションに関連付けられている静的ファイルを含むディレクトリ。IIS で、現在構成中のアプリケーション・パスの下位パスを持つアプリケーションの、異なるディレクトリからの静的ファイル进行处理の場合は、子ディレクトリの IIS 仮想ディレクトリを定義します。
6. [OK] をクリックします。
7. IIS の構成が完了したら、IIS を再起動して構成の変更を有効にします。

注釈 InterSystems IRIS インスタンスにアクセスせずに、Web ゲートウェイの管理ページにアクセスするには、/csp パスに対して Web ゲートウェイを有効にします。

代替オプションに基づく Web ゲートウェイ・ソリューションを使用している場合は、/csp アプリケーションの下に /bin というアプリケーションをセットアップします。Web ゲートウェイ・バイナリを保持している物理ディレクトリにこれをマップします。ほとんどの場合、これはアプリケーション・パス /csp/bin と物理パス C:\inetpub\CSPGateway 間のマッピングです。

5.3.4.1 仮想ディレクトリの定義

IIS の仮想ディレクトリを使用すると、アプリケーションに指定されたファイル・システムの場所の外にあるファイル・システムからの静的コンテンツを処理できます。例えば、domain.com/images でイメージをホストする静的 Web サイトの管理者は、Web サイトのほとんどのコンテンツを C:\inetpub\wwwroot に格納できますが、仮想ディレクトリ images をその物理パスにマッピングすることで、イメージを C:\siteimg に格納できます。

1 つの親パス・ディレクトリ (/csp など) 内に、InterSystems IRIS アプリケーションごとに異なる静的ファイルのストレージ場所を保持する場合は、IIS の仮想ディレクトリを使用できます。

IIS の仮想ディレクトリを構成するには、次の手順に従います。

1. [インターネット インフォメーション サービス (IIS) マネージャ] 内の [接続] パネルで、下位の仮想ディレクトリを構成する IIS アプリケーションを見つけて選択します。
2. [操作] パネルから [仮想ディレクトリの表示] を選択します。
3. [操作] パネルから [仮想ディレクトリの追加] を選択します。
4. [仮想ディレクトリの追加] ダイアログに以下の情報を入力します。
 - ・ **エイリアス**：アプリケーションの URL パスの下位部分 (例：パス /csp/sys の sys)。
 - ・ **物理パス**：このパスの静的ファイルの代替の場所。
5. [OK] をクリックします。
6. IIS の構成が完了したら、IIS を再起動して構成の変更を有効にします。

5.3.5 アプリケーション要求のハンドラ・マッピングの設定

構成した IIS アプリケーションごとに、ハンドラ・マッピングを作成して、アプリケーション・パスに送信される一部 (またはすべて) の要求のハンドラとして Web ゲートウェイを呼び出すように IIS に指示します。

アプリケーション・パスに送信されるすべての要求のハンドラとして Web ゲートウェイを呼び出すには、以下の手順に従って要求パス * (ワイルドカード) のハンドラ・マッピングを作成します。

特定のファイル・タイプに対してのみ Web ゲートウェイを呼び出すには、ファイル・タイプごとに、以下の手順に従って *.xxx (xxx はファイル拡張子) を要求パスとして指定します。インターシステムズ固有のファイル・タイプを処理するには、*.csp、*.cls、*.zen、および *.cxw の要求パスのハンドラ・マッピングを作成します。アプリケーションで処理する静的ファイル・タイプのハンドラ・マッピングも作成する必要があります。(REST API エンドポイントは通常、ファイルではなくパスであるため、REST アプリケーションの要求を呼び出すには、このファイル・タイプ固有のアプローチでは不十分です。)

以下の手順を実行して、IIS アプリケーションのハンドラ・マッピングを作成します。

1. [インターネット インフォメーション サービス (IIS) マネージャ] を開きます。
2. [接続] パネルで localhost 接続のコンテンツを展開し、[サイト]→[既定の Web サイト] 内のリストから [アプリケーション] の名前を選択します。
3. 接続のホーム・ページの [機能ビュー] で、[ハンドラー マッピング] 機能構成項目を開きます (ダブルクリック)。
4. [操作] パネルで、[モジュール マップの追加..] を選択します。

注釈 Web ゲートウェイをアプリケーション・パスのすべての要求のハンドラとして設定するために、[ワイルドカードスクリプトマップの追加...] 操作を使用しないでください。これを試みると、エラーが発生します。代わりに、ワイルドカード (*) 文字のモジュール・マッピングを追加します。

5. [モジュール マップの追加] ダイアログで以下を指定します。
 - ・ **要求パス** – Web ゲートウェイが処理する一連の要求を指定する式を指定します。(有効な式の例は、このセクションの冒頭を参照してください。)
 - ・ **モジュール** – ドロップダウン・メニューから、Web ゲートウェイ・モジュールに指定した名前を選択します (CSPms)。
 - ・ **名前** – このマッピングにわかりやすい名前を指定します (WebGateway_* など)。
6. [要求の制限] を選択し、[要求のマッピング先が次の場合のみハンドラーを呼び出す] の横にあるボックスにチェックが付けられていない (選択されていない) ことを確認します。この操作により、[ハンドラー マッピング] テーブルに示すように、[パスの種類] の値が [指定なし] に設定されます。
7. [OK] を選択して [モジュール マップの追加] ダイアログに戻り、もう一度、[OK] を選択します。
8. 上記のプロセスを繰り返して、IIS アプリケーションに必要なすべてのハンドラ・マッピングを追加します。
9. IIS の構成が完了したら、[IIS を再起動](#)して構成の変更を有効にします。

5.3.6 /bin を含む URLの有効化

既定では、IIS はディレクトリ /bin を含む要求パスからはリソースを提供しません。[Web ゲートウェイ管理ページ](#)を提供するには、このフィルタを無効にする必要があります。

インストーラが Web ゲートウェイを自動構成した場合、この手順は自動的に実行されています。Web ゲートウェイを手動でインストールする場合は、この構成を手動で実行する必要があります。

具体的には、以下の手順によって、IIS で Web ゲートウェイ管理ページを提供できるようにします。

1. テキスト・エディタで、applicationHost.config ファイルを開きます。このファイルは通常、**C:\Windows\System32\Inetsrv\Config** ディレクトリにあります。お使いのシステムで、このファイルの場所を見つけるには、以下を実行します。
 - a. [接続] パネルで、localhost 接続またはその接続内の任意のサイト項目のホーム・ページから、[構成エディター] を選択 (ダブルクリック) します。
 - b. [操作] パネルから [構成の検索...] を選択します。
 - c. [階層表示] で、[ApplicationHost.config] (またはその子項目) を選択します。この操作によって、[構成検索] ダイアログに ApplicationHost.config ファイルの場所が表示されます。
2. <configuration> ブロックの最後に (ただし、ブロック内) に、以下の <Location> タグを追加します。

```
<location path="localhost/csp/bin/Systems/Module.cxx">
  <system.webServer>
    <security>
      <requestFiltering>
        <hiddenSegments>
          <remove segment="bin" />
        </hiddenSegments>
      </requestFiltering>
    </security>
  </system.webServer>
</location>
```

3. ファイルを保存します。
4. IIS の構成が完了したら、[IIS を再起動](#)して構成の変更を有効にします。

5.3.7 リモート Web サーバ接続のランチャーの構成

InterSystems IRIS ランチャーが InterSystems IRIS インスタンスの組み込みの Web の有効な URL を構成できるようにするには、InterSystems IRIS サーバ・マネージャでインスタンスの接続の詳細を指定する必要があります。これらの接続の詳細は、Web サーバ構成と一致する必要があります。“システム管理ガイド”の“[リモート・サーバ接続の定義](#)”を参照してください。

5.3.8 SOAP フォルトの詳細を返すための IIS の構成

エラーが発生している InterSystems IRIS Web サービスは、関連する SOAP フォルトの詳細を含めずに HTTP 500 エラーを返すことがあります。既定では、IIS は広範なエラー情報をローカル・クライアントに対してのみ返します。しかし、構成ファイル **web.config** 内の **<httpErrors>** 要素でこの動作を変更できます。変更するには、以下のセクションを追加して、詳細なエラー情報をすべてのクライアントに配信するように IIS に指示します。

```
<configuration>
  <system.webServer>
    <httpErrors errorMode="Detailed" />
  </system.webServer>
</configuration>
```

ホスト環境に関する機密情報がクライアントに公開されることがあるため、この方法を使用する場合は注意が必要です。**errorMode="Detailed"** の使用に関するセキュリティ上の問題を回避する代替の方法は、**existingResponse="PassThrough"** 指示文を使用することです。

```
<configuration>
  <system.webServer>
    <httpErrors existingResponse="PassThrough" />
  </system.webServer>
</configuration>
```

構成の変更後に、IIS を再起動します。

これらの変更は、IIS **web.config** ファイルに対して手動で行うことができます。または、より適切で、エラーの発生が少ない方法を選択するには、IIS マネージャに組み込まれている**構成エディタ**を使用します。

1. IIS マネージャの左側の**[接続]**パネルから、Web サービスに対応するパスを選択します。例：**[既定の Web サイト]**の後に **csp** を選択。
2. 中央のパネル下部の**[管理]**という見出しの下で、**[構成エディタ]**をダブルクリックします。
3. **[セクション]**のラベルが付いた上部の**[構成エディタ]**ドロップダウンで、**system.webServer**を展開し、**[httpErrors]**をクリックします。
4. **[existingResponse]**の横にある値をクリックし、ドロップダウンを使用してオプションを表示します。**[PassThrough]**を選択します。
5. 右側の**[アクション]**ペインで、**[適用]**をクリックします。
6. IIS の構成が完了したら、**IIS を再起動**して構成の変更を有効にします。

IIS でのエラー処理の詳細は、以下を参照してください。

<https://docs.microsoft.com/ja-jp/iis/configuration/system.webServer/httpErrors/>

5.3.9 IIS の再起動

このセクションでは、各種コントロール・パネル経由で IIS を再起動したときに何が起こるかについて説明します。

アクティブな IIS インストールに対する構成変更の大半はリアルタイムで行うことができます。しかし、[インターネット インフォメーション サービス (IIS) マネージャ] コントロール・パネルには停止、開始、および再起動オプションが用意されています。これらのオプションは Web サーバ構成の更新には便利です。しかし、Web ゲートウェイ・インストールは再初期化されません。

Web ゲートウェイ・インストールに変更を加えた場合は、以下のいずれかの方法で IIS を再起動します。

- ・ Windows サービス・マネージャを開いて、World Wide Web Publishing サービスを再起動します。
- ・ コマンド行から以下のコマンドを実行して、IIS を停止します。

```
sc stop W3SVC
```

Web サーバが停止したら、以下のコマンドを実行してこの Web サーバを再起動します。

```
sc start W3SVC
```

5.3.10 トラブルシューティング

このセクションでは、IIS で使用するためにサードパーティ・モジュール (ネイティブ・モジュールおよび ISAPI モジュールの両方) を構成しているときによく発生する問題について説明します。

最もよく発生する問題は、再構成後、IIS への要求が失敗し、以下のエラーが表示されるというものです。

Service Unavailable

HTTP Error 503. The service is unavailable.

これは通常、既定のアプリケーション・プールが終了していることを示します。

1. [インターネット インフォメーション サービス (IIS) マネージャ] ウィンドウを開きます。
2. 左側のパネルで、最上位レベルを展開します。[アプリケーション プール] セクションが表示されます。

[MACHINE_NAME] ([machine_name]¥[user_name])

アプリケーション・プール

3. 既定のアプリケーション・プール (DefaultAppPool)、またはサーバで使用が構成されているアプリケーション・プールのステータスが [開始] になっていることを確認します。
4. 必要に応じて、右側のパネルのオプションを使用して、アプリケーション・プールを再開します。
5. それでも問題が解決されない場合は、メインの Windows イベント・ログ ([アプリケーション] セクション内) を見て、ヒントを探してください。特に、次のエラー・メッセージを確認してください。

Failed to find the RegisterModule entrypoint in the module DLL

C:¥Inetpub¥CSPGateway¥CSPms.dll.The data is the error.

例えば、これは、使用しているバージョンの Web ゲートウェイ DLL が、このネイティブ・モジュール・インタフェースを実装していないことを示します。インターシステムズから最新の DLL を入手するか、または従来の ISAPI インタフェースで動作するように Web ゲートウェイを構成します。

すべてのソフトウェアと同様、一時的な問題が再起動により解決されることはよくあります。IIS を完全に再起動するには、メインの Windows サービス・コントロール・パネルから World Wide Web Publishing サービスを再起動します。

ファイル拡張子をマップするのにワイルドカードスクリプトマップの追加ユーティリティを使用しないでください。使用すると、

IsapiModule CgiModule

代わりに、モジュールマップの追

加を使用し、ワイルドカードを指定してファイル拡張子をマップしてください。

/bin を含む URL が機能しない場合には、[/bin を含む URLの有効化の手動手順](#)を参照してください。

5.4 UNIX®/Linux/macOS 向けの Nginx

5.4.1 概要

ここでは、UNIX®、Linux、または macOS 上の InterSystems Web ゲートウェイで使用するための、Nginx Web サーバの構築および構成方法について説明します。

Nginx はオープン・ソース製品で、ソース・コードは <http://nginx.org/> から無料でダウンロードできます。

Linux 用の一部のビルド済みキットを入手できますが、通常、これは最新の Nginx ビルドよりもいくつか前のバージョンをビルドしたものです。ただし、拡張子を Nginx コアにコンパイルする必要があるため、CSP のサポートを含めるには、Web サーバをソース・コードからローカルにビルドすることが必要です。

このページの手順の後、[Web ゲートウェイ管理ページ](#)を使用して、さらに Web ゲートウェイを構成できます。

5.4.2 想定

ここでは、以下のように想定しています。

- ・ CSP/Web ゲートウェイの Web サーバ・コンポーネントは `/opt/webgateway/bin/` にインストールされている
- ・ InterSystems IRIS はローカルにインストールされている場合は、`/opt/iris/` に配置されている
- ・ Web サーバは `/opt/nginx/` の下にインストールされている

システムのレイアウトが異なる場合は、必要に応じて、構成指示文を修正してください。

5.4.3 インストール

以下の Web ゲートウェイのコンポーネントおよび CSP 静的ファイルをインストールします。

1. Web ゲートウェイ Network Service Daemon (NSD)

- ・ `CSPnsd`

このバイナリの既定の場所は `/opt/webgateway/bin/` です。

2. ハイパーイベントのコンポーネント

`CSPBroker.js`

`CSPxmlhttp.js`

これらのファイルの既定の場所は `/opt/iris/csp/broker` です。

これらのファイルが静的コンポーネントとして Web サーバによって直接処理される場合は、`/opt/nginx/html/csp/broker` にコピーしてください。

3. 管理ポータルで使用するその他の静的リソース

管理ポータルでは、(イメージ・ファイルなど)いくつかの静的な Web リソースが必要になります。これらのファイルの既定の場所は `/opt/iris/csp/sys` です。

これらのファイルが静的コンポーネントとして Web サーバによって直接処理される場合は、`/opt/nginx/html/csp/sys` にコピーしてください。

5.4.4 CSP のための Nginx Web サーバの構築

Web ゲートウェイ機能のほとんどは、NSD (CSPnsd) によって提供されます。CSP アクセスのために、小さなコンパイル済みモジュール `ngx_http_csp_module.c` を介して NSD と通信するように、Nginx を構築および構成することができます。便宜のため、すべての Web ゲートウェイ・インストールにはこのソース・ファイルが含まれています。

ここで説明する構築手順は、UNIX® システム下で Nginx を構築するための公式ドキュメントに基づいています。

<http://nginx.org/en/docs/configure.html>

Nginx のドキュメントには、以下のサードパーティのアドオンが必要であると記載されています。

- ・ PCRE
<http://www.pcre.org/>
- ・ OpenSSL (SSL/TLS の場合)
<https://www.openssl.org/>
- ・ Zlib
<http://zlib.net/>

重要 使用する OpenSSL ツールキットは、ご使用の InterSystems IRIS のバージョンに適合している必要があります。“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”を参照してください。

ただし、最終インストールで、上記のコンポーネントによって提供される機能を必要としない場合は、これらのコンポーネントを使用しなくても十分に機能するサーバを作成することは可能です。

上記のオプションのモジュールすべてを含む、Nginx を構築するための一般的な構成スクリプトは、以下のとおりです。

```
./configure --prefix=/opt/nginx --with-http_ssl_module
```

これにより、既定の Nginx ビルドが `/opt/nginx` 下にインストールされます。

構築プロセスを以下のように変更して、オプションのモジュールを除外することもできます。

- ・ OpenSSL - SSL/TLS 機能を削除します。
指示文の削除: `--with-http_ssl_module`
- ・ Zlib - GZIP 機能を削除します。
指示文の追加: `--without-http_gzip_module`
- ・ PCRE - HTTP 書き換え機能を削除します。
指示文の追加: `--without-http_rewrite_module`

5.4.4.1 CSP のための Nginx の構築手順

1. ソース・ディストリビューションを任意の場所に解凍します。以下に例を示します。

```
/opt/
```

解凍後、`/opt/` を指定すると、ソース・コード・ディストリビューションは以下に配置されます。

```
/opt/nginx-n.n.n/
```

2. CSP 拡張子に以下のディレクトリを作成します。

```
/opt/nginx-n.n.n/csp/
```

3. モジュールのソース・コード (`ngx_http_csp_module.c`) を、上記で作成したディレクトリにコピーします。
4. 同じディレクトリ内に、`config` という名前の構成ファイルを作成します。このファイルには、以下の行が含まれる必要があります。

```
ngx_addon_name=ngx_http_csp_module
HTTP_MODULES="$HTTP_MODULES ngx_http_csp_module"
NGX_ADDON_SRCS="$NGX_ADDON_SRCS $ngx_addon_dir/ngx_http_csp_module.c"
CORE_LIBS="$CORE_LIBS -ldl"
```

5. `/opt/nginx-n.n.n/` で作業し、Nginx 構築環境を構成します。

```
./configure --prefix=/opt/nginx
             --with-http_ssl_module
             --add-module=/opt/nginx-n.n.n/csp
```

または、OpenSSL、ZLIB、および PCRE により提供されるオプションの機能を使用しない場合は、以下のようになります。

```
./configure --prefix=/opt/nginx
             --without-http_rewrite_module
             --without-http_gzip_module
             --add-module=/opt/nginx-n.n.n/csp
```

最後の行に、CSP モジュールをインクルードする指示が含まれることに注意してください。

6. Nginx のコンパイル:

```
make
```

7. Nginx のインストール:

```
make install
```

成功した場合、以下のディレクトリに完全なサーバ・インストールが表示されます。

```
/opt/nginx/
```

5.4.5 Nginx での NSD の使用

InterSystems IRIS アプリケーションが処理する必要のある要求を認識し、それらの要求を NSD に渡すように、Web サーバを構成する必要があります。

そのためには、`/opt/nginx/conf` にある Web サーバ構成ファイル (`nginx.conf`) を編集します。

ここでは、CSP 拡張モジュールが Web サーバの構成のために提供するサーバ構成指示文について説明します。location ブロックのコンテキスト内でこれらの指示文を発行すると、その指示文は指定されたパスのトラフィックに適用されます。

CSPNSD_pass hostname:portNum;

(必須。)NSD が待ち受けるアドレス (hostname と port) を指定します。

特定のパスの NSD アドレスを指定しない場合、NSD は既定で `127.0.0.1:7038` のアドレスで待ち受けます。

CSP on; および **CSP off;**

すべての要求に対し、Web ゲートウェイを介した CSP サーバへのルーティングを有効または無効にします。

特定のパスに適用する CSP 指示文を発行しない場合、そのパスに送信された要求は `CSP off` によりルーティングされず、Web サーバは Web ゲートウェイを介してそのパスに送信される要求をルーティングしません。

CSPFileTypes filetype1[filetype2...];

特定のファイル・タイプ (filetype1、filetype2 など) の要求を Web ゲートウェイを介して CSP サーバにルーティングできるようにします。

例えば、`.csp` または `.cls` ファイルを要求している場合 (そのような要求に限り)、Web ゲートウェイがその要求を `/demo/app` パスに送信するようにするには、以下の指示文ブロックを発行します。

```
location /demo/app {
    CSPFileTypes csp cls;
}
```

指示文 `CSPFileTypes *` を発行することにより、すべてのファイル・タイプの要求のルーティングが有効になります。

この指示文は、ファイルを要求しない URL の要求はルーティングしません。そのため、この指示文は、エンドポイントでパスを指定することが多いほとんどの API には不適切です。

CSPNSD_response_headers_maxsize size;

HTTP 応答のヘッダの最大サイズを指定します。応答ヘッダがこのサイズを超えると、Web クライアントはエラーを受け取ります。

既定では、CSP 拡張モジュールは指示文 `CSPNSD_response_headers_maxsize 8k` を適用します。

CSPNSD_connect_timeout time;

Web クライアントからの要求受信時の、NSD への接続タイムアウトを指定します。

既定では、CSP 拡張モジュールは指示文 `CSPNSD_connect_timeout 300s` を適用します。

CSPNSD_send_timeout time;

単一の送信操作要求 (POST、PUT など) のタイムアウトを指定します。タイムアウトは連続する送信操作間でのみ適用されます。単一の送信が開始されてから完了までの間に適用されるものではありません。

既定では、CSP 拡張モジュールは指示文 `CSP_send_timeout 300s` を適用します。

CSPNSD_read_timeout time;

単一の読み取り操作 (GET など) の応答が返る際のタイムアウトを指定します。タイムアウトは連続する読み取り操作間でのみ適用されます。単一の読み取りが開始されてから完了までの間に適用されるものではありません。

既定では、CSP 拡張モジュールは指示文 `CSP_read_timeout 300s` を適用します。

5.4.5.1 例：特定のパスでのすべてのトラフィックに対する CSP ルーティングを有効にする

適切な `server` 構成ブロック内に以下のセクションを配置し、`/csp` パスに送信されるすべてのトラフィックを Web ゲートウェイにルーティングします。

```
location /csp {
    CSP On;
    CSPNSD_pass localhost:7038;
}
```

5.4.5.2 例 : InterSystems IRIS のファイル・タイプに対する要求を Web ゲートウェイにルーティングする

適切な **server** 構成ブロック内に以下のセクションを配置し、**/csp** パスに送信される InterSystems IRIS のファイル・タイプ (**.csp**、**.cls**、**.zen**、および **.cxw**)、および管理ポータル・インタフェースの静的要素を処理するのに必要なファイル・タイプに対する要求の CSP ルーティングを有効にします。

```
location /csp {
    CSPFileTypes csp cls zen cxw .jpg .gif .png .svg .css .js;
    CSPNSD_pass localhost:7038;
}
```

5.4.6 IDE の使用に必要な追加の構成

InterSystems ObjectScript 拡張機能を含む VS Code (または **/api/atelier** Web アプリケーションを使用する IDE) を ObjectScript IDE として使用するには、Nginx 構成に以下の追加の変更を行う必要があります。

1. **/api/atelier** Web アプリケーションは、アンダースコアを含む HTTP ヘッダを使用します。既定では、Nginx はこれらのヘッダを通知なしで削除します。これらを含めるように Nginx サーバを構成する必要があります。
https://www.nginx.com/resources/wiki/start/topics/tutorials/config_pitfalls/#missing-disappearing-http-headers を参照してください。
2. デバッガなどの機能には、WebSocket 接続が必要です。既定では、Nginx は WebSocket をサポートするように構成されていません。WebSocket の有効化に関する情報は、Nginx のドキュメント (<http://nginx.org/en/docs/http/websocket.html>) を参照してください。

5.4.7 Nginx と NSD の起動および停止

Nginx を起動する方法は、以下のとおりです。

```
/opt/nginx/sbin/nginx
```

Nginx を停止する方法は、以下のとおりです。

```
/opt/nginx/sbin/nginx -s stop
```

NSD の操作方法については、“**NSD の操作**” を参照してください。

5.4.8 非推奨 : ユニバーサル・モジュールを使用する Nginx の構築

重要 Nginx でのユニバーサル・モジュールの使用は、安定性に問題があるため推奨されていません。NSD を使用して Nginx に接続する Web ゲートウェイの導入では、**WebSockets** を含め、すべての機能を完全にサポートしています。

現在、Nginx でユニバーサル・モジュールを使用している場合は、Web ゲートウェイを最新バージョンにアップグレードし、**NSD と連携するように Nginx サーバを再構築することをお勧めします**。**サーバ構成ファイルを編集する際は**、サーバ構成から **CSPModulePath** 指示文を必ず削除してください。

以下の手順は、既存のインストールのリファレンスとしてのみ機能します。

動的にリンクされたユニバーサル・モジュール **CSPx.so** (実行時) と **CSPxSys.so** (Web ゲートウェイ・システム管理) と連携するように Nginx を構築できます。ユニバーサル・モジュールと連携するように Nginx を構築および構成する手順は、以下のように NSD ベースの導入とは異なります。

- ・ 手順 3 で、モジュールのソース・コードとして、**ngx_http_csp_module.c** ではなく **ngx_http_csp_module_sa.c**、**cspapi.h**、および **ngx_http_csp_common.h** を指定のディレクトリにコピーします。

- ・ 手順 4 で、CSP の構成ファイル (`/opt/nginx-n.n.n/csp/config`) は、以下のようになります。

```
ngx_addon_name=ngx_http_csp_module_sa
HTTP_MODULES="$HTTP_MODULES ngx_http_csp_module_sa"
NGX_ADDON_SRCS="$NGX_ADDON_SRCS $ngx_addon_dir/ngx_http_csp_module_sa.c"
```

- ・ `CSPModulePath` 指示文を `http` 構成ブロックから追加して、ユニバーサル・ゲートウェイ・モジュールへのパスを指定します。

```
CSPModulePath /opt/webgateway/bin;
```

- ・ 以下の指示文はサポートされていません。

- `CSPNSD_pass`
- `CSPNSD_response_headers_maxsize`
- `CSPNSD_connect_timeout`
- `CSPNSD_send_timeout`
- `CSPNSD_read_timeout`

- ・ 以下の指示文はサポートされています。

- `CSP`
- `CSPFileTypes`

5.5 Windows 向け Nginx

5.5.1 概要

ここでは、Windows 上の InterSystems [Web ゲートウェイ](#)で使用するための、Nginx Web サーバの構築および構成方法について説明します。

Nginx はオープン・ソース製品です。ソース・コードは <http://nginx.org/> から無料でダウンロードできます。

Windows 用の一部のビルド済みキットを入手できますが、通常、これは最新の Nginx ビルドよりもいくつか前のバージョンをビルドしたものです。ただし、拡張子を Nginx コアにコンパイルする必要があるため、CSP のサポートを含めるには、Web サーバをソース・コードからローカルにビルドする必要があります。

このページの手順の後、[Web ゲートウェイ管理ページ](#)を使用して、さらに Web ゲートウェイを構成できます。

5.5.2 想定

ここでは、以下のよう想定しています。

- ・ CSP/Web ゲートウェイ・コンポーネントは `install-dir¥csp¥` にインストールされている。
- ・ Web サーバは `C:¥nginx¥` にインストールされている。

システムのレイアウトが異なる場合は、必要に応じて、構成指示文を修正してください。

5.5.3 インストール

以下の Web ゲートウェイのコンポーネントおよび CSP 静的ファイルをインストールします。

1. Web ゲートウェイ Network Service Daemon (NSD)

- ・ **CSPnsd.exe** (メイン・バイナリ)
- ・ **CSPnsdSv.exe** (Windows サービス)

これらのファイルの既定の場所は `install-dir¥bin` です。

構成ファイルと**ログ・ファイル**は、このディレクトリに書き込まれます。

2. ハイパーイベントのコンポーネント

- ・ **CSPBroker.js**
- ・ **CSPxmlhttp.js**

これらのファイルの既定の場所は `install-dir¥csp¥broker` です。

これらのファイルが静的コンポーネントとして Web サーバによって直接処理される場合は、`C:¥nginx¥html¥csp¥broker` にコピーしてください。

3. 管理ポータルで使用するその他の静的リソース

管理ポータルでは、(イメージ・ファイルなど)いくつかの静的な Web リソースが必要になります。これらのファイルの既定の場所は `install-dir¥csp¥sys` です。

これらのファイルが Web サーバによって直接処理される場合は、`C:¥nginx¥html¥csp¥sys` にコピーしてください。

5.5.4 CSP のための Nginx Web サーバの構築

Web ゲートウェイ機能のほとんどは、NSD (**CSPnsd[Sv].exe**) によって提供されます。CSP アクセスのために、小さなコンパイル済みモジュール `ngx_http_csp_module.c` を介して NSD と通信するように、Nginx を構築および構成することができます。便宜のため、すべての Web ゲートウェイ・インストールにはこのソース・ファイルが含まれています。

Nginx 構築の前提条件：

- ・ Microsoft Visual Studio (バージョン 10 以上)：<http://www.microsoft.com>
- ・ MSYS2 (MinGW より)：<https://www.msys2.org/>
- ・ Perl (ActivePerl を推奨)：<https://www.activestate.com/products/perl/>
- ・ Mercurial ソース・コントロール・クライアント：<https://www.mercurial-scm.org/>

ここで説明する構築手順は、Windows 下で Nginx を構築するための公式ドキュメントに基づいています。

http://nginx.org/en/docs/howto_build_on_win32.html

Nginx のドキュメントには、以下のサードパーティのアドオンが必要であると記載されています。

- ・ PCRE：<http://www.pcre.org/>
- ・ OpenSSL (SSL/TLS の場合)：<https://www.openssl.org/>
- ・ Zlib：<http://zlib.net/>

重要 使用する OpenSSL ツールキットは、ご使用の InterSystems IRIS のバージョンに適合している必要があります。“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”を参照してください。

ただし、最終インストールで、上記のコンポーネントによって提供される機能を必要としない場合は、これらのコンポーネントを使用しなくても十分に機能するサーバを作成することは可能です。

上記のオプションのモジュールすべてを含む、Nginx を構築するための既定の構成スクリプトは、以下のとおりです。

```
auto/configure \
--with-cc=cl \
--with-debug \
--prefix= \
--conf-path=conf/nginx.conf \
--pid-path=logs/nginx.pid \
--http-log-path=logs/access.log \
--error-log-path=logs/error.log \
--sbin-path=nginx.exe \
--http-client-body-temp-path=temp/client_body_temp \
--http-proxy-temp-path=temp/proxy_temp \
--http-fastcgi-temp-path=temp/fastcgi_temp \
--http-scgi-temp-path=temp/scgi_temp \
--http-uwsgi-temp-path=temp/uwsgi_temp \
--with-cc-opt=-D_FFD_SETSIZE=1024 \
--with-pcre=objs/lib/pcre-8.44 \
--with-zlib=objs/lib/zlib-1.2.12 \
--with-openssl=objs/lib/openssl-1.1.1k \
--with-openssl-opt=no-asm \
--with-http_ssl_module
```

構築プロセスを以下のように変更して、オプションのモジュールを除外することもできます。

- ・ OpenSSL – SSL/TLS 機能を削除します。
指示文の削除:--with-http_ssl_module
- ・ Zlib – GZIP 機能を削除します。
指示文の追加:--without-http_gzip_module
- ・ PCRE – HTTP 書き換え機能を削除します。
指示文の追加:--without-http_rewrite_module

5.5.4.1 CSP のための Nginx の構築手順

1. MSYS2 シェルで作業を開始し、Nginx のドキュメントで推奨される作業ディレクトリ構造を作成します。

```
/opt/
```

2. /opt で作業して、以下のコマンドを使用して Nginx ソース・コードをチェックアウトします。

```
hg clone http://hg.nginx.org/nginx
```

これにより、Nginx ソース・コードが /opt/nginx/ の下に配置されます。

3. CSP 拡張子に以下のディレクトリを作成します。

```
mkdir /opt/nginx/objs/lib/csp/
```

4. モジュールのソース・コード (ngx_http_csp_module.c) を、前の手順で作成したディレクトリにコピーします。
5. 同じディレクトリ内に、**config** という名前の構成ファイルを作成します。このファイルには、以下の行が含まれる必要があります。

```
ngx_addon_name=ngx_http_csp_module
HTTP_MODULES="$HTTP_MODULES ngx_http_csp_module"
NGX_ADDON_SRCS="$NGX_ADDON_SRCS $ngx_addon_dir/ngx_http_csp_module.c"
```

6. /opt/nginx/ で作業し、Nginx 構築環境を構成します。

```
auto/configure --with-cc=cl --bulddir=objs --prefix=
--conf-path=conf/nginx.conf --pid-path=logs/nginx.pid
--http-log-path=logs/access.log --error-log-path=logs/error.log
--sbin-path=nginx.exe
--http-client-body-temp-path=temp/client_body_temp
--http-proxy-temp-path=temp/proxy_temp
--http-fastcgi-temp-path=temp/fastcgi_temp
--with-cc-opt=-D_FSIZE=1024 --without-http_rewrite_module
--without-http_gzip_module
--with-select_module --with-ipv6
--add-module=objs/lib/csp
```

最後の行に、CSP モジュールをインクルードする指示が含まれることに注意してください。

7. Nginx をコンパイルします。これは、現在の MSYS2 シェルまたは Visual Studio 開発者コマンド・プロンプトで実行できます。

MSYS2 シェルを使用するには、目的の Visual Studio 構築環境に対応する **vcvarsall.bat** スクリプトを見つけて、Nginx をコンパイルします。

```
cd /c/path/to/vcvarsall
vcvarsall.bat
cd -
nmake -f objs/Makefile
```

vcvarsall.bat を探す場所がわからない場合は、Visual Studio 開発者コマンド・プロンプトを開くと、構築環境が設定されます。まず、MSYS2 パスを現在の MSYS2 シェル内の同等の Windows パスに変換します。

```
cygpath -m $(pwd)
```

次に、目的の構築環境のために Visual Studio コマンド・プロンプトを開き、その Windows パスに移動します。Nginx をコンパイルします。

```
nmake -f objs/Makefile
```

成功した場合、/opt/nginx/objs/ にサーバ (nginx.exe) が表示されます。

8. Nginx のインストール：これを実行する最も簡単な方法は、まず Windows 用の Nginx の事前構築済みバージョンをダウンロードおよびインストールして、ディレクトリ構造 (通常 **C:\nginx**) を取得し、次にそのインストール内の **nginx.exe** ファイルを、ローカルで作成したものと置き換えることです。

動作可能な Nginx インストールの通常のディレクトリ構造は、以下のとおりです。

```
Directory of C:\nginx

03/07/2017  09:09    <DIR>          .
03/07/2017  09:09    <DIR>          ..
26/06/2017  10:14    <DIR>          conf
26/06/2017  10:14    <DIR>          contrib
10/05/2018  12:53    <DIR>          csp
26/06/2017  10:14    <DIR>          docs
26/06/2017  10:14    <DIR>          html
10/05/2018  15:57    <DIR>          logs
04/07/2017  15:52       715,264  nginx.exe
26/06/2017  10:17    <DIR>          scgi_temp
26/06/2017  10:17    <DIR>          temp
26/06/2017  10:17    <DIR>          uwsgi_temp
```

このディレクトリの nginx.exe のコピーを、構築手順で作成したバージョンで置き換えます。

5.5.5 NSD を呼び出すように Nginx を構成

インターシステムズのファイル・システムの要求を認識し、これらの要求 (および InterSystems IRIS アプリケーションが処理するその他の静的ファイルの要求) を処理のため NSD に渡すよう、Web サーバを構成する必要があります。

そのためには、**C:\nginx\conf**にある Web サーバ構成ファイル (**nginx.conf**) を編集します。

ここでは、CSP 拡張モジュールが Web サーバの構成のために提供するサーバ構成指示文について説明します。
location ブロックのコンテキスト内でこれらの指示文を発行すると、その指示文は指定されたパスのトラフィックに適用されます。

CSPNSD_pass hostname:portNum;

(必須。)NSD が待ち受けるアドレス (hostname と port) を指定します。

特定のパスの NSD アドレスを指定しない場合、NSD は既定で 127.0.0.1:7038 のアドレスで待ち受けます。

CSP on; および CSP off;

すべての要求に対し、Web ゲートウェイを介した CSP サーバへのルーティングを有効または無効にします。

特定のパスに適用する CSP 指示文を発行しない場合、そのパスに送信された要求は **CSP off** によりルーティングされず、Web サーバは Web ゲートウェイを介してそのパスに送信される要求をルーティングしません。

CSPFileTypes filetype1[filetype2...];

特定のファイル・タイプ (filetype1、filetype2 など) の要求を Web ゲートウェイを介して CSP サーバにルーティングできるようにします。

例えば、**.csp** または **.cls** ファイルを要求している場合 (そのような要求に限り)、Web ゲートウェイがその要求を **/demo/app** パスに送信するようにするには、以下の指示文ブロックを発行します。

```
location /demo/app {
    CSPFileTypes csp cls;
}
```

指示文 **CSPFileTypes *** を発行することにより、すべてのファイル・タイプの要求のルーティングが有効になります。ただし、REST API は、ファイルを指定しない (したがって、ファイル・タイプを指定しない) エンドポイントでの要求の受信をサポートする必要があります。このようなアプリケーションの場合、ファイル・タイプに基づいて要求に対して Web ゲートウェイを呼び出すことができないため、Web ゲートウェイをパスのすべての要求のハンドラとして使用できるようにする必要があります (**CSP on;** を使用)。

CSPNSD_response_headers_maxsize size;

HTTP 応答のヘッダの最大サイズを指定します。応答ヘッダがこのサイズを超えると、Web クライアントはエラーを受け取ります。

既定では、CSP 拡張モジュールは指示文 **CSPNSD_response_headers_maxsize 8k** を適用します。

CSPNSD_connect_timeout time;

Web クライアントからの要求受信時の、NSD への接続タイムアウトを指定します。

既定では、CSP 拡張モジュールは指示文 **CSPNSD_connect_timeout 300s** を適用します。

CSPNSD_send_timeout time;

単一の送信操作要求 (POST、PUT など) のタイムアウトを指定します。タイムアウトは連続する送信操作間でのみ適用されます。単一の送信が開始されてから完了までの間に適用されるものではありません。

既定では、CSP 拡張モジュールは指示文 **CSP_send_timeout 300s** を適用します。

CSPNSD_read_timeout time;

単一の読み取り操作 (GET など) の応答が返る際のタイムアウトを指定します。タイムアウトは連続する読み取り操作間でのみ適用されます。単一の読み取りが開始されてから完了までの間に適用されるものではありません。

既定では、CSP 拡張モジュールは指示文 `CSP_read_timeout 300s` を適用します。

5.5.5.1 例：特定のパスでのすべてのトラフィックに対する CSP ルーティングを有効にする

適切な **server** 構成ブロック内に以下のセクションを配置し、`/csp` パスに送信されるすべてのトラフィックを Web ゲートウェイにルーティングします。

```
location /csp {
    CSP On;
    CSPNSD_pass localhost:7038;
}
```

5.5.5.2 例：InterSystems IRIS のファイル・タイプに対する要求を Web ゲートウェイにルーティングする

適切な **server** 構成ブロック内に以下のセクションを配置し、`/csp` パスに送信される InterSystems IRIS のファイル・タイプ (`.csp`、`.cls`、`.zen`、および `.cxw`)、および管理ポータル・インタフェースが使用する静的ファイルのファイル・タイプに対する要求の CSP ルーティングを有効にします。

```
location /csp {
    CSPFileTypes csp cls zen cxw jpg gif png svg css js;
    CSPNSD_pass localhost:7038;
}
```

5.5.6 Nginx と NSD の起動および停止

Nginx を起動する方法は、以下のとおりです。

```
C:\nginx\nginx
```

Nginx を停止する方法は、以下のとおりです。

```
C:\nginx\nginx -s stop
```

NSD の操作方法については、“[NSD の操作](#)” を参照してください。

5.5.7 VS Code ユーザの場合：必要な追加構成

InterSystems ObjectScript 拡張機能を含む VS Code を、Nginx Web サーバを使用する InterSystems IRIS インスタンスに接続する場合は、Nginx を以下のように構成する必要があります。

1. サポートされている ObjectScript IDE は、`/api/atelier` Web アプリケーションを使用して InterSystems IRIS に接続します。この Web アプリケーションは、アンダースコアがある HTTP ヘッダを含む要求を送信します。既定では、Nginx はこれらのヘッダを通知なしで削除します。Nginx がアンダースコアを含む HTTP ヘッダを保持できるようにしてください。https://www.nginx.com/resources/wiki/start/topics/tutorials/config_pitfalls/#missing-disappearing-http-headers を参照してください。
2. デバッガなどの機能には、WebSocket 接続が必要です。既定では、Nginx は WebSocket をサポートするように構成されていません。Nginx が WebSocket をサポートできるようにしてください。<http://nginx.org/en/docs/http/websocket.html> を参照してください。

5.5.8 非推奨：ユニバーサル・モジュールを使用する Nginx の構築

重要

Nginx でのユニバーサル・モジュールの使用は、安定性に問題があるため推奨されていません。NSD を使用して Nginx に接続する Web ゲートウェイの導入では、[WebSockets](#) を含め、すべての機能を完全にサポートしています。

現在、Nginx でユニバーサル・モジュールを使用している場合は、Web ゲートウェイを最新バージョンにアップグレードし、[NSD と連携するように Nginx サーバを再構築することをお勧めします](#)。[サーバ構成ファイルを編集する](#)際は、サーバ構成から `CSPModulePath` 指示文を必ず削除してください。

以下の手順は、既存のインストールのリファレンスとしてのみ機能します。

NSD の代わりに動的にリンクされたユニバーサル・モジュール `CSPx.dll` (実行時) と `CSPxSys.dll` (Web ゲートウェイ・システム管理) と連携するように Nginx を構築できます。ユニバーサル・モジュールと連携するように Nginx を構築および構成する手順は、以下のように NSD ベースの導入とは異なります。

- ・ 手順 4 で、モジュールのソース・コードとして、`ngx_http_csp_module.c` ではなく `ngx_http_csp_module_sa.c` および `ngx_http_csp_common.h` を指定のディレクトリにコピーします。

- ・ 手順 5 で、CSP の構成ファイル (`/opt/nginx/objs/lib/csp/config`) は、以下のようになります。

```
ngx_addon_name=ngx_http_csp_module_sa
HTTP_MODULES="$HTTP_MODULES ngx_http_csp_module_sa"
NGX_ADDON_SRCS="$NGX_ADDON_SRCS $ngx_addon_dir/ngx_http_csp_module_sa.c"
```

- ・ `CSPModulePath` 指示文を `http` 構成ブロックに追加して、ユニバーサル・ゲートウェイ・モジュールへのパスを指定します。

```
CSPModulePath install-dir/bin;
```

- ・ Windows の場合、スレッド・スタック・サイズを 2MB まで増やす必要があります。以下の指示文を Nginx 構成ファイルの先頭 (`http` セクションの前) に追加します。

```
thread_stack_size 2000000;
```

- ・ 以下の指示文はサポートされていません。

- `CSPNSD_pass`
- `CSPNSD_response_headers_maxsize`
- `CSPNSD_connect_timeout`
- `CSPNSD_send_timeout`
- `CSPNSD_read_timeout`

- ・ 以下の指示文はサポートされています。

- `CSP`
- `CSPFileTypes`

6

Web ゲートウェイを介して要求をルーティングする URL パスの選択

InterSystems Web ゲートウェイ・モジュールで Web サーバ構成を拡張したら、InterSystems Web アプリケーション用の Web ゲートウェイ接続を設定するプロセスの次のステップは、[要求の URL](#) の相対パス（つまり、プロトコル、ホスト名、ポート番号などの要素を省略したパス）に基づいて、処理するために Web サーバが Web ゲートウェイに渡す要求を指定することです。

ただし、構成するために選択するパスは、[ルーティング・プロセス](#)の他のステージでも重要になります。

- ・ Web ゲートウェイは要求の相対パス（プロトコル、ホスト名、ポート番号などの [URL の構成要素](#)を省略したパス）を使用し、ユーザが定義した[アプリケーション・アクセス・プロファイル](#)に基づいて、要求を受信する InterSystems IRIS アプリケーション・サーバを決定します。
- ・ InterSystems IRIS アプリケーション・サーバは、要求の相対パスといずれかの Web アプリケーションの名前を照合して、特定の要求に対して呼び出す Web アプリケーションを決定します。
- ・ 複数のインスタンスがあるシステム上で、特定の InterSystems IRIS インスタンスをターゲットにするには、要求をルーティングするために選択する相対パスに、[各インスタンスを一意に識別する接頭語](#)（ただし、アプリケーション・サーバが要求のパスをいずれかの Web アプリケーションと照合するのを妨げないもの）を含める必要があります。

このページでは、要求の URL で指定された相対パスに基づいて、各ステージで要求がルーティングされる方法を決定するための関連条件について詳しく説明します。Web アプリケーションをホストするための URL パスを決定する際には、これらの条件を考慮してください。

このページでは、以下の状況での要求のルーティングに関する情報も提供します。

- ・ 単一の Web サーバで複数の InterSystems IRIS インスタンスをホストする場合。
- ・ 単一の Web サーバで複数の InterSystems IRIS インスタンスをホストする場合（各インスタンスのアドレス指定のために使用されるインスタンス接頭語をカスタマイズする必要がある）。
- ・ Apache 仮想ホストを使用し、異なるホスト名を使用してアプリケーション要求をルーティングする場合。

6.1 Web サーバから

一般に、Web サーバは URL パスに送信される要求の処理方法を決定する際に、継承のルールを適用します。そうでない場合は、特定の要求パスに一致する最も具体的なパスの構成指示文を適用します。

例えば、Web クライアントが相対 URL パス `/Accounts/Invoices` に要求を送信する場合、Apache Web サーバは `<Location>` ブロックの構成を以下の順序で確認し、見つかった最も適切なブロック内の指示文を適用します。

1. `<Location /Accounts/Invoices>`
2. `<Location /Accounts>`
3. `<Location />`

ルーティング動作の詳細は、Web サーバのドキュメントを参照してください。特定の URL パスの要求に対して Web ゲートウェイを呼び出すように Web サーバを構成する方法は、使用している Web サーバによっても異なります。概要は、[設定の概要ページのこのセクション](#)、詳細は、“[Web ゲートウェイでの Web サーバ構成の拡張](#)” の該当セクションを参照してください。

6.2 Web ゲートウェイ経由

Web ゲートウェイも、アプリケーション・パスに継承のルールを適用します。これは特定の要求パスに一致する最も具体的なパスのアプリケーション・アクセス・プロファイルを使用します。

前のセクションの要求例 (`/Accounts/Invoices`) を考えた場合、Web ゲートウェイは、パスに有効なアプリケーション・アクセス・プロファイルが定義されているかどうかを次の順序で確認し、見つかった最も具体的なパスのプロファイルを適用します。

- ・ `/Accounts/Invoices`
- ・ `/Accounts`
- ・ `/ (root)`

Web ゲートウェイは、`/csp` および `/ (root)` パスのアプリケーション・アクセス・プロファイルを自動的に定義します。

注釈 新規またはアップグレードしたインスタンスについて、Web アプリケーションを提供するよう [Web サーバを自動構成](#) することをインストーラに許可するたびに、インストーラは最近インストールまたはアップグレードしたインスタンスに要求をルーティングするように、既存の `/csp` および `/ (root)` アプリケーション・アクセス・プロファイルを上書きします。不便を避けるために、`/csp` パス内のアプリケーション (`/csp/foo/bar` など) を手動で構成する際に、`/csp` の下位のパス (`/csp/foo` や `/csp/foo/bar` など) の [アプリケーション・アクセス・プロファイルを定義](#) します。

アプリケーション・パスでは大文字と小文字が区別されます。あいまいさが生じるため、アプリケーション・パスにドット (ピリオド) を含めることはできません。パス `/csp/samples/menu.csp/csp/foo/bar/baz.cls` を例として考えると、Web ゲートウェイでは、これを `/csp/samples/menu.csp/csp/foo/bar/baz.cls` に対する要求とも、`/csp/samples/menu.csp` に対する REST 要求 (この場合、`PATH_INFO` は `/csp/foo/bar/baz.cls`) とも解釈できます。

6.3 InterSystems IRIS アプリケーション・サーバへ

InterSystems IRIS アプリケーション・サーバは、Web ゲートウェイから要求を受け取ると、要求の相対 URL パスをその InterSystems IRIS インスタンスで構成された Web アプリケーションの名前と照合しようとします。次に、Web アプリケーションの適切なコードを呼び出します。Web サーバや Web ゲートウェイと同様に、アプリケーション・サーバも継承のルールを適用します。`/Accounts/Invoices` に対する要求を考えた場合、アプリケーション・サーバはまず `/Accounts/Invoices` アプリケーションのアプリケーション・コードを呼び出そうとします。そのようなアプリケーションが定義されていない場合は、`/Accounts` アプリケーションのアプリケーション・コードを呼び出そうとします。

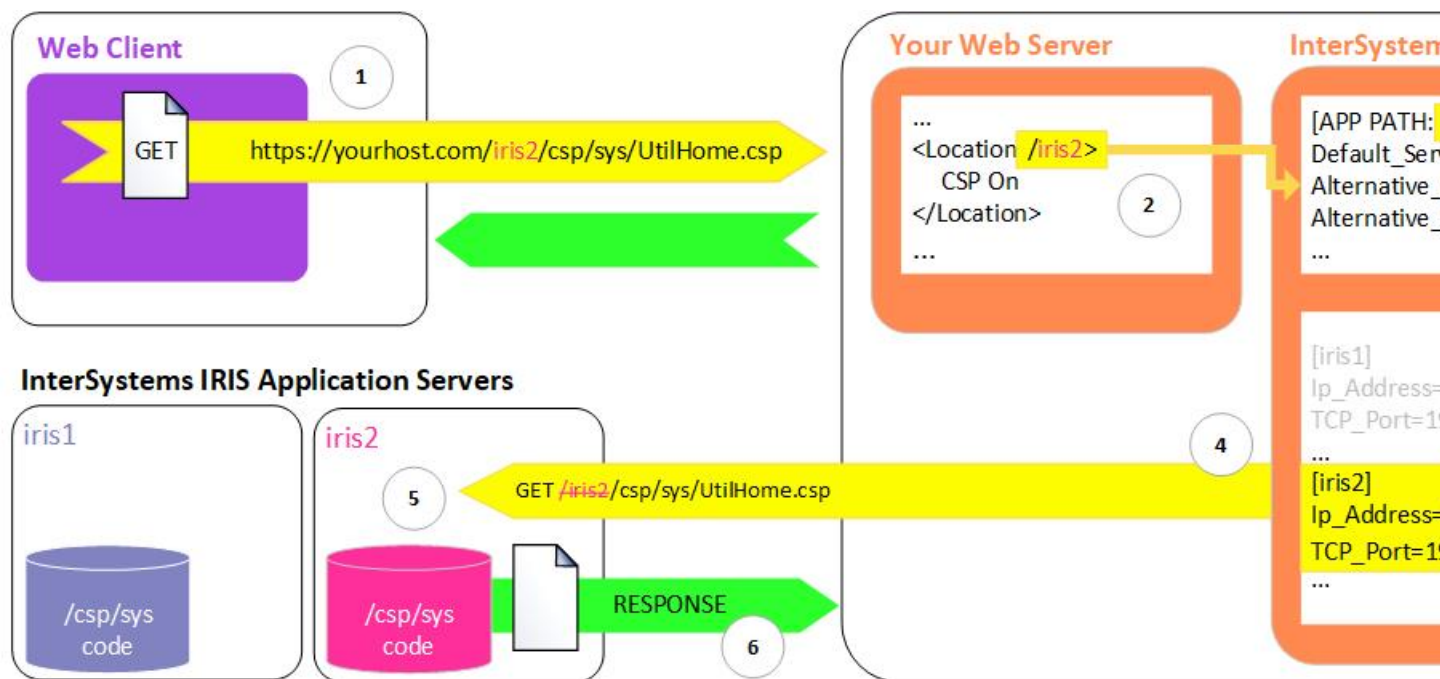
単一の Web サーバで複数の InterSystems IRIS インスタンスをホストし、特定のアプリケーション要求のターゲットを特定のインスタンスにすることもできます。前の例で考えると、組織には Payable インスタンス（ベンダーへの支払を追跡するため）と Receivable インスタンス（顧客からの支払を追跡するため）があると思われます。どちらのインスタンスも同じ `/Accounts/Invoices` アプリケーションのコードをホストしていますが、データが異なります。単一の Web サーバで Payable と Receivable の両方から `/Accounts/Invoices` データを利用できるようにするには、要求 URL は、ターゲット・アプリケーション・サーバが要求とアプリケーションを照合する機能を妨げずに、ターゲット・インスタンスを識別する必要があります。このセクションの残りの部分では、これを実現する方法について説明します。

6.3.1 複数の InterSystems IRIS サーバ上のターゲット・アプリケーション

InterSystems IRIS Web アプリケーションの中には、単一の InterSystems IRIS インスタンスに固有の機能を提供するものもあります。例えば、どのインスタンスにも、管理ポータル Web アプリケーションを提供するコードが含まれます。しかし、管理ポータルは特定の 1 つのインスタンスを管理するためのインタフェースを提供します。

複数の InterSystems IRIS インスタンスがあるシステムでは、インスタンスの `CSPConfigName` パラメータをアプリケーション・パスの接頭語として含むパスの **アプリケーション・アクセス・プロファイル** を構成することで、特定のインスタンスに宛てた要求をルーティングすることができます。既定では、このパラメータはすべて小文字のインスタンス名です。InterSystems IRIS アプリケーション・サーバは、要求に応じて呼び出すアプリケーション・コードを決定する前に、要求に関連するアプリケーション・パスの先頭から `CSPConfigName` の値を破棄します。これにより、Web サーバと Web ゲートウェイは、`CSPConfigName` を接頭語として使用して、インスタンスのアプリケーションの動作をカスタマイズすることなく、そのインスタンスを一意にターゲット指定することができます。

以下の図に示されているシステムを考えてみます。



1. Web クライアントは、iris2 インスタンスの管理ポータルのホーム・ページのアプリケーション・パスを要求します。アプリケーション・パスには、ターゲット・インスタンス (iris2) の既定の `CSPConfigName` が接頭語として付いています。
2. 架空の Apache Web サーバの構成指示文は、`/iris2` パス内のすべての相対 URL パスに送信された要求に対して Web ゲートウェイを呼び出します。これには、クライアントの現在の要求が含まれます。
3. Web ゲートウェイ構成内で、アプリケーション・アクセス・プロファイルは `/iris2` パス内の要求を `iris2` というサーバ・アクセス・プロファイルに関連付けます。

- Web ゲートウェイは、サーバ・アクセス・プロファイル iris2 で指定された IP アドレスとポートの InterSystems IRIS アプリケーション・サーバに要求を送信します。
- iris2 インスタンスの InterSystems IRIS アプリケーション・サーバは、要求の相対パスの最初の部分を破棄します。インスタンスの CSPConfigName の既定値（インスタンス名）に一致するためです。パスの残りの部分（/csp/sys/UtilHome.csp）に基づいて、/csp/sys アプリケーションの適切なアプリケーション・コードを呼び出します。
- /csp/sys アプリケーション・コードは Web ゲートウェイに応答を返し、Web ゲートウェイは Web サーバを介して要求をクライアントに返します。

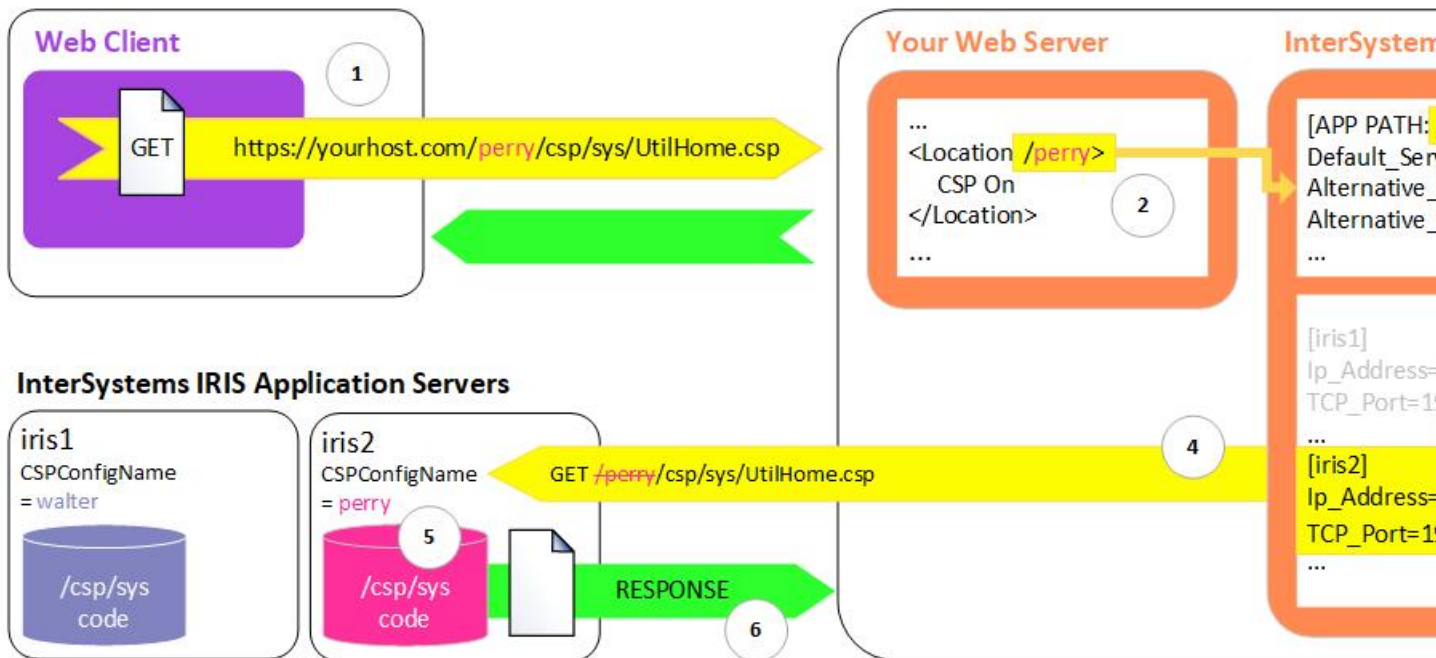
InterSystems IRIS インストーラは、インストールまたはアップグレード時に Web サーバに接続するようにインスタンスを自動構成する際、インスタンスの既定の CSPConfigName に対応する Web サーバ構成指示文と Web ゲートウェイのアプリケーション・アクセス・プロファイルが自動的に追加されます。インスタンスの Web ゲートウェイを手動で構成する場合は、Web サーバを構成し、このアプリケーション・アクセス・プロファイルを手動で追加する必要があります。（まだインスタンスのサーバ・アクセス・プロファイルを追加していない場合は、最初にこれを追加する必要があります。）

Windows システムでは、Web 接続の詳細が変更された InterSystems IRIS インスタンスに対して InterSystems IRIS ランチャーを有効にするには、追加の構成が必要です。“システム管理ガイド”の“リモート・サーバ接続の定義”を参照してください。

6.3.2 カスタムのインスタンス接頭語を使用した各 InterSystems IRIS サーバのアドレス指定

InterSystems IRIS アプリケーション・サーバのインスタンス名を Web アプリケーション URL に表示したくない場合は、このセクションで説明する手順に従い、指定された名前とアプリケーション・アクセス要件は、自分の名前とアプリケーション・アクセス要件に置き換えてください。

例えば、iris1 および iris2 という 2 つのインスタンスに管理ポータルを提供したいが、以下の図に示すように、iris1 の代わりに walter という名前を使用し、iris2 の代わりに perry という名前を使用するとします。



これらの代替名を使用して、各インスタンスに管理ポータルを提供するには、以下の手順を実行します。

- 以下の条件が両方とも満たされていることを確認します。

- a. Web サーバが、`/walter/csp/sys` および `/perry/csp/sys` (または親ディレクトリ) に送信された要求に対して [Web ゲートウェイ](#)を呼び出している。
 - b. Web ゲートウェイ構成に、適宜名前が付けられた `iris1` と `iris2` の両方のサーバ・アクセス・プロファイルが含まれている。
2. `iris1` インスタンスの [ObjectScript シェル](#)で、以下のコマンドを実行します。

```
d $System.CSP.SetConfig("CSPConfigName","walter")
```
 3. `iris2` インスタンスの [ObjectScript シェル](#)で、以下を実行します。

```
d $System.CSP.SetConfig("CSPConfigName","perry")
```
 4. [Web ゲートウェイ管理ページ](#)で、[\[デフォルトサーバ\]](#) が `iris1` であるアプリケーション・パス `/walter/csp/sys` (または `/walter/csp`、または `/walter`) の [アプリケーション・アクセス・プロファイル](#)を追加します。
 5. [\[デフォルトサーバ\]](#) が `iris2` であるアプリケーション・パス `/perry/csp/sys` (または `/perry/csp`、または `/walter`) の [アプリケーション・アクセス・プロファイル](#)を追加します。

`CSPConfigName` は、CSP 構成名のコンマ区切りリストも受け入れます。これにより、単一ではなく複数の構成名を持つことができます。以下はその例です。

```
d $System.CSP.SetConfig("CSPConfigName","perry,perry1,perry2,perry3")
```

他の CSP グローバル・パラメータを確認するには、`%SYS>d $system.CSP.DisplayConfig()` と入力します。

Windows システムでは、変更された Web 接続の詳細を使用して InterSystems IRIS インスタンスに対して InterSystems IRIS ランチャーを有効にするには、追加の構成が必要です。“システム管理ガイド”の[“リモート・サーバ接続の定義”](#)を参照してください。

6.3.3 Apache 仮想ホストの構成

前のセクションで説明したインスタンス接頭語メカニズムの代替手段として、Web ゲートウェイでは仮想ホスト名の使用をサポートし、複数の InterSystems IRIS インスタンスでアプリケーションを提供しています。Apache 仮想ホストは、単一の Web サーバを使用して、異なるホスト名で異なるアプリケーションを透過的に提供する手段を提供します。仮想ホストの詳細は、Apache のドキュメント (<https://httpd.apache.org/docs/2.4/vhosts/>) を参照してください。

例えば、単一の Apache Web サーバで 2 つの異なる Web サイト (`www.virtualhost1.com` と `www.virtualhost2.com`) の InterSystems IRIS アプリケーションをホストするように構成できます。

Web サーバのグローバル構成で、仮想ホスト名への要求に対して Web ゲートウェイを呼び出すように Web サーバを構成した場合、Web ゲートウェイは仮想ホスト名の要求を特定の InterSystems IRIS インスタンスにルーティングできます。2 つのスラッシュ (`/`) で始まるアプリケーション・パスのアプリケーション・アクセス・プロファイルを作成するだけです。

以下の構成手順により、Web ゲートウェイは `www.virtualhost1.com/csp/sys/UtilHome.csp` の要求を `IRISserv1` というインスタンスの管理ポータル ホーム・ページに送信し、`www.virtualhost2.com/csp/sys/UtilHome.csp` の要求を `IRISserv2` というインスタンスの管理ポータル ホーム・ページに送信できるようになります。

1. [Web ゲートウェイ管理ページ](#)のメイン・メニューに移動します。
2. `irisserv1` と `irisserv2` の [サーバ・アクセス・プロファイル](#)を作成します。
3. `//virtualhost1.com/csp/sys/` と `//virtualhost2.com/csp/sys/` の [アプリケーション・アクセス・プロファイル](#)を作成します。次のように構成します。
 - ・ パス `//virtualhost1.com/csp/sys/` の [サーバ 0](#) をサーバ・アクセス・プロファイル `irisserv1` に設定します。

- ・ パス `//virtualhost2.com/csp/sys/` のサーバ 0 をサーバ・アクセス・プロファイル `irisserv2` に設定します。

注釈 Web ゲートウェイはアプリケーション・アクセス・プロファイルでの仮想ホスト名の使用をサポートしていますが、`<VirtualHost>` 指示文ブロック内での Web ゲートウェイを呼び出すための Apache 構成指示文 (`CSPFileTypes` と `CSP On/Off`) の発行はサポートされておらず、これを行うとエラーが発生します。つまり、目的の仮想ホストのみに対して Web ゲートウェイを有効にすることはできません。Web サーバのグローバル構成内で Web ゲートウェイを有効にする必要があります。

Windows システムでは、変更された Web 接続の詳細を使用して InterSystems IRIS インスタンスに対して InterSystems IRIS ランチャーを有効にするには、追加の構成が必要です。“システム管理ガイド”の“[リモート・サーバ接続の定義](#)”を参照してください。

7

Web ゲートウェイ管理ページの概要

Web ゲートウェイには、Web ゲートウェイを構成および監視するために使用できる一連の管理ページが用意されています。ここでは、これらのページにアクセスする方法やこれらをローカライズする方法を説明し、これらのページのオプションの概要を示します。

7.1 Web ゲートウェイ管理ページへのアクセス

既定では、Web ゲートウェイのホスト・コンピュータのローカル側にあるクライアントのみが Web ゲートウェイ管理ページにアクセスできるため、これらのページへのアクセスに使用するブラウザは、Web サーバおよび Web ゲートウェイと同じマシン上で実行されている必要があります。

この場合、Web ゲートウェイ管理ページの URL は以下のような形式になります。

```
http://localhost/csp/bin/Systems/Module.cxxw
```

非標準ポートでリッスンするように Web サーバを構成した場合は、ホスト名の後に `:<portNum>` を追加します。<portNum> はポート番号です (例 : `localhost:11555`)。

アクセスしやすいように、この URL をブックマークします。

注釈 Apache Web サーバの場合、URL パスとファイル名は大文字と小文字を区別します。

Web ゲートウェイ管理ページにアクセスしようとする、CSP システム・ユーザの認証情報を求められます。構成ファイルでユーザ名を探します。パスワードは、InterSystems IRIS のインストール時に入力したものです。パスワードを忘れた場合は、“セキュリティ”を参照してください。

Tip ヒン これらのページには、Web ゲートウェイに接続されている任意の InterSystems IRIS インスタンスの管理ポータルからもアクセスできます。[システム管理]→[構成]→[ウェブゲートウェイ管理]に移動します。同じ考慮事項が、クライアント・アドレスに対しても適用されます。

7.2 追加クライアント・アドレスからのアクセスの有効化

承認された管理者のリストにクライアントを追加するには、構成ファイルの SYSTEM セクションにある `System_Manager` パラメータにクライアントの IP アドレスを追加します。このパラメータには、Web ゲートウェイ管理ページへのアクセスを許可するクライアントの IP アドレスをコンマ (,) またはプラス記号で区切って指定します。以下の指示文は、既定のローカル・アクセスに加え、3 つのリモート・クライアントにアクセスを許可します。

```
[SYSTEM]
System_Manager=190.8.7.6, 190.8.7.5, 190.8.7.4
```

使用できるローカルなブラウザがない新しいゲートウェイ・インストールの場合は、手動で構成ファイルを編集し、System_Manager パラメータを追加します。このパラメータは、Web ゲートウェイ管理ページの **[デフォルトパラメータ]** セクションにある **[システム管理マシン]** の設定と同じです。このパラメータのエントリには、ワイルドカードと数値域を指定できます。

注釈 Web ゲートウェイ管理ページをロードしようとして、ブラウザでページをロードできず、**[この機能を使用する権限がありません]** というエラーが表示される場合は、System_Manager の設定により IP アドレスへのアクセスがブロックされることが原因であると思われます。

以下の例は、IP アドレスの最後の部分が 4 ～ 6 の数値となることを示しています。

```
[SYSTEM]
System_Manager=190.8.7.4-6
```

上記の例は、以下の記述をより簡単にしたものです。

```
[SYSTEM]
System_Manager=190.8.7.6, 190.8.7.5, 190.8.7.4
```

以下の例のように、ワイルドカードも使用できます。

```
[SYSTEM]
System_Manager=190.8.7.*
```

以下の指示文は、すべてのクライアントにアクセス権を付与します。

```
[SYSTEM]
System_Manager=*.*.*.*
```

ただし、このような指示文を運用システムで使用することはお勧めしません。クライアント IP アドレスはスプーフィングが可能なため、このアプローチでは強力なセキュリティを提供できません。

クライアントと Web サーバ/ゲートウェイ間でプロキシを使用すると、事実上すべてのクライアントの IP アドレスがプロキシの IP アドレスに変換されます。この場合、Gateway Systems Manager としてプロキシの IP アドレスを指定して、Systems Manager がプロキシ経由で接続するすべての Web ユーザにアクセス権を付与するようにするか、指定した Systems Manager がプロキシ層を完全にバイパスするように設定する (推奨の設定) 必要があります。

IP 方式は防御の第一段階としては有効ですが、Web ゲートウェイ管理ページへのアクセスを制御する手段としてはこれだけでは不十分です。明らかに、インターネット経由で利用する CSP インストール環境には適していません。プロダクション・システムでは、ホスト Web サーバの構成を使用して、Web ゲートウェイのシステム管理モジュールへのアクセスを制御することをお勧めします。

7.3 使用可能なオプション

以下のテーブルに、Web ゲートウェイ管理のメイン・メニュー・ページで利用可能なオプションを示します。

メニュー項目	アクション
Web ゲートウェイについて	Web ゲートウェイについての情報が表示されます。表示される情報には、InterSystems IRIS ディストリビューションのバージョン、Web ゲートウェイのビルド番号、ロードされる OpenSSL のバージョン、ホスト Web サーバのバージョン、アクティブ・インタフェース、Web ゲートウェイ構成ファイル (CSP.ini) の名前と場所、およびイベント・ログ (CSP.log) の名前と場所が含まれています。Web ゲートウェイのビルド番号は、2 つの数値要素から構成されます。最初の数値は、InterSystems IRIS のバージョンを示しています。2 番目の数値は、Web ゲートウェイの内部ビルド番号です。
[システムステータス]	アクティブなサーバ接続のステータスを表示します。また、 接続を閉じる ことや、 Web ゲートウェイ・キャッシュをクリア することもできます。
[サーバ接続のテスト]	ステートレス・セッションを開いて、InterSystems IRIS サーバへの接続をテストします。
[イベントログを表示]	Web ゲートウェイのイベント・ログの情報を参照したり、その内容を消去したりすることができます。このログは、Web サーバ・ホスト上で保持されるファイルです。
[HTTP トレースを表示]	Web ゲートウェイで処理された HTTP 要求および応答のインタラクティブ・ビューを表示します。
[既定のパラメータ]	特定の Web サーバで Web ゲートウェイを構成できます。また、エラーやその他の状態に対する CSP の応答をカスタマイズできます。
[サーバ・アクセス]	特定の InterSystems IRIS サーバへの Web ゲートウェイのアクセスを構成します。
[アプリケーション・アクセス]	アプリケーションのパスに従って、アプリケーションへのアクセスを構成します。ここでいうパスとは、アプリケーションの URL に含まれているパスのことです。
管理ポータルに戻る	InterSystems IRIS 管理ポータル・ページに戻ります。

これらのページには、[ヘルプ] ボタンが含まれます。

7.4 ローカリゼーション

Web ゲートウェイ管理ページのローカリゼーションは、CSPres.xml ファイルの内容に基づきます (このファイルがインストールされている場合)。ローカリゼーション・ファイルが存在しない場合、Web ゲートウェイ管理ページは、埋め込みの英語テキストを使用します。ブラウザの言語設定はこのメカニズムに影響しません。

代替言語は、Web ゲートウェイのホーム・ディレクトリにある CSPres.xml という名前の適切なテキスト・リソース・ファイルをインストールすることによってサポートできます。Web ゲートウェイが起動または再起動されると、CSPres.xml にあるテキスト・リソースがロードされ、管理フォームが選択した言語で表示されます。

CSPres.xml ファイルを作成するには、InterSystems IRIS bin ディレクトリの該当する CSPres_xx.xml ファイルの名前を CSPres.xml に変更します。

例えば、スペイン語に変換するには、以下のようになります。

1. CSPres_es.xml を CSPres.xml に変更します。
2. Web サーバを再起動します。言語テキストは指定された Web サーバの CSP モジュールに影響するため。再起動する必要があります。

英語に戻すには、以下のようになります。

1. CSPres.xml を CSPres_es.xml に戻します。
2. Web サーバを再起動します。

8

InterSystems IRIS インスタンスのサーバ・アクセス・プロファイルの定義

このページでは、InterSystems Web ゲートウェイのサーバ・アクセス・プロファイルを構成する方法について説明します。サーバ・アクセス・プロファイルにより、[InterSystems Web ゲートウェイ](#)は Web アプリケーションをホストする InterSystems IRIS® アプリケーション・サーバとの接続を確立し、保持することができます。

Web ゲートウェイからアクセスする各 InterSystems IRIS インスタンスを、サーバ・アクセス・プロファイルで定義する必要があります。指定していないオプションのパラメータやカスタム・システム・フォームは、Web ゲートウェイの[既定の設定](#)から自動的に継承されます。InterSystems IRIS アプリケーション・サーバのサーバ・アクセス・プロファイルを定義したら、[アプリケーション・アクセス・プロファイル](#)を定義して、アプリケーション・パスと対応するアプリケーション・サーバを関連付けることができます。

これらおよび[その他の構成タスク](#)には、Web ゲートウェイ[管理ページ](#)または [Web ゲートウェイ・レジストリ・メソッド](#)を使用します。Web ゲートウェイはこの構成情報を [CSP.ini](#)に保持します。

重要 CSP.ini ファイルを直接編集しなければならないことがあるコンテナ化された導入を除き、CSP.ini ファイルへのアクセスを制限して、すべての Web ゲートウェイ構成を Web ゲートウェイ管理ページを使用して実行することをお勧めします。

8.1 サーバ・アクセス・プロファイルの追加

Web ゲートウェイが InterSystems IRIS アプリケーション・サーバに接続できるようにするには、Web ゲートウェイ構成内で InterSystems IRIS サーバを特定するためのサーバ・アクセス・プロファイルを定義します。そのためには、以下の操作を実行します。

1. Web ゲートウェイ[管理ページ](#)のメイン・メニューから、**[サーバ接続]**を選択します。
2. **[サーバ追加]**を選択します。2 番目の構成画面が表示されます。多くのパラメータ・フィールドには既定の設定が入力されています。
3. **[サーバ名]** テキスト・ボックスに、一意のわかりやすいサーバの名前を入力します。この論理名は、CSP 構成ファイルでサーバ構成を識別するために使用されます。
4. このサーバ・アクセス・プロファイルのシステム・パラメータ (以下を参照) を入力します。
5. **[設定を保存]**を選択します。

8.1.1 サーバ・アクセス・パラメータ

サーバ構成の基本的なパラメータは以下のとおりです。

サーバ構成のパラメータ	機能
[サーバ名]	CSP 構成ファイルでこのサーバ・アクセス・プロファイルを識別するための論理名。
[サービス状態]	Web ゲートウェイ構成内で、このサーバを有効または無効にできます（既定値は [有効] です）。
[IP アドレス]	接続先の InterSystems IRIS サーバの DNS ホスト名または IP アドレス（物理アドレスまたは仮想アドレス）。
[スーパーサーバの TCP ポート]	InterSystems IRIS サーバが着信接続を待ち受ける TCP ポート番号。これは InterSystems IRIS スーパーサーバの TCP ポート番号です。既定値は 1972 ですが、同じシステムに複数のインスタンスが導入される場合は異なる値になることがあります。
[構成はミラーを認識する]	<p>ミラー・プライマリをミラーリングされるデータベースにアクセスするためのサーバとして構成します。フェイルオーバーまたは災害復旧では、接続がリダイレクトされません。既定では、選択されていません。</p> <p>注意：ミラー VIP を構成している場合、ミラー認識 Web ゲートウェイを構成しないでください。構成すると、Web ゲートウェイは VIP を無視します。その代わりに、他のクライアントと同様に、単に Web ゲートウェイを VIP に接続するように構成します。一般的に、ミラー認識 Web ゲートウェイの使用は特殊な状況下でのみ適切な選択といえます。</p> <p>構成するには、フェイルオーバー・メンバのいずれかの IP アドレスを入力します。Web ゲートウェイは、このフェイルオーバー・メンバから、ミラーにあるフェイルオーバー・メンバおよび災害復旧 (DR) 非同期メンバのリストを取得し、このリストに基づいて現在のプライマリに接続します（構成されている場合でも、VIP に接続しません）。プライマリが見つかるまで、CSP 接続は失敗します。</p> <p>接続が確立されると、ミラーがフェイルオーバーする場合、Web ゲートウェイは接続を新しいプライマリに変更します。プライマリがフェイルオーバー・メンバの中になければ、Web ゲートウェイはリスト内の DR 非同期の中から見つけようとします。その場合、災害復旧の状況で DR 非同期がプライマリに昇格されると、ゲートウェイは接続を再確立できます。</p> <p>詳細は、“高可用性ガイド” の “ミラーリング” にある “フェイルオーバーまたは災害復旧後のアプリケーション接続のリダイレクト” を参照してください。</p>

8.1.2 状態なしパラメータ

ステートレス接続に関連するパラメータは以下のとおりです。

状態なしパラメータ	機能
[サーバ接続最小数]	Web ゲートウェイは、プロセスとの親和性を提供します。つまり、可能であれば、前回の要求を処理したときと同じ InterSystems IRIS プロセスとのセッションに必ず再接続しようとします。このパラメータは、多くのクライアント間で接続を共有する前に、Web ゲートウェイが InterSystems IRIS サーバに対して行う接続の最小数を指定します。この数が多いほど、プロセスとの親和性の効果が高くなります。既定値は 3 です。

状態なしパラメータ	機能
[サーバ接続最大数]	Web ゲートウェイが InterSystems IRIS サーバに対して行うことができる、接続の絶対最大数です。同時使用数がこの数を超えると、Web ゲートウェイは要求のキューイングを開始します。InterSystems IRIS 接続が要求を処理できるようになるまで、または Queued Request Timeout を超えるまで、要求はキューに維持されます。これは既定では指定されていません。ハード最大値のみが Web ゲートウェイの最大接続数として示され、この既定値が 1024 です。
[セッションあたりの最大接続数]	このパラメータは、個々のセッションが同時に使用可能な InterSystems IRIS への最大接続数を表します。既定値は 3 です。

8.1.3 接続セキュリティのパラメータ

接続セキュリティ設定は、Web ゲートウェイが InterSystems IRIS アプリケーション・サーバにアクセスするために必要です。これらのパラメータの詳細は、“[InterSystems IRIS への Web ゲートウェイ接続の保護](#)”を参照してください。接続セキュリティに関連するパラメータは以下のとおりです。

接続セキュリティのパラメータ	機能
接続セキュリティ・レベル	InterSystems IRIS サーバへの接続に必要なセキュリティのレベルです。以下のいずれかのオプションを選択します。 <ul style="list-style-type: none"> ・ [パスワード] ・ [Kerberos] ・ [Kerberos/パケット整合性] ・ [Kerberos暗号化] ・ SSL/TLS
[ユーザ名]	Web ゲートウェイから InterSystems IRIS サーバへの接続に必要なユーザ名です。
[パスワード]	Web ゲートウェイから InterSystems IRIS サーバへの接続に必要なパスワードです。 また、UNIX®/Linux/macOS システムでは、このフィールドで、 パスワードをプログラムによって取得するオペレーティング・システム・コマンド を中括弧({})で囲んで指定することもできます。
[パスワード (確認)]	新しいパスワードを作成する場合、そのパスワードを再入力して確認します。
[製品]	接続先の製品 (InterSystems IRIS)。
[サービス・プリンシパル名]	サービス・プリンシパル名です。[生成] ボタンは、ターゲットの InterSystems IRIS サーバに関して既定の名前を作成するために用意されています。
[キーテーブル]	キー・テーブル・ファイルのフル・パスです。

8.1.4 SSL/TLS パラメータ

以下のパラメータは、[SSL/TLS](#) を使用して Web ゲートウェイと InterSystems IRIS 間の接続をセキュリティ保護するインストール環境にのみ関連します。

SSL/TLS パラメータ	機能
---------------	----

SSL/TLS パラメータ	機能
[Minimum SSL/TLS Protocol Version]	<p>使用する SSL/TLS プロトコルの最小バージョンです。以下のオプションが用意されています。</p> <ul style="list-style-type: none"> ・ TLSv1.0 ・ TLSv1.1 ・ TLSv1.2 ・ TLSv1.3 (サポートされているプラットフォームで) <p>TLSv1.3 がサポートされているプラットフォームでは、既定値は TLSv1.2 です。それ以外の場合、既定値は TLSv1.1 です。</p>
[Maximum SSL/TLS Protocol Version]	<p>使用する SSL/TLS プロトコルの最大バージョンです。以下のオプションが用意されています。</p> <ul style="list-style-type: none"> ・ TLSv1.0 ・ TLSv1.1 ・ TLSv1.2 ・ TLSv1.3 (サポートされているプラットフォームで) <p>TLSv1.3 がサポートされているプラットフォームでは、既定値は TLSv1.3 です。それ以外の場合、既定値は TLSv1.2 です。</p>
[SSL/TLS キー・タイプ]	<p>SSL/TLS キー・ファイルのタイプです (その生成に使用するアルゴリズムに基づきます)。以下のオプションが用意されています。</p> <ul style="list-style-type: none"> ・ DSA – Digital Signature Algorithm ・ RSA – Rivest, Shamir, Adelman (アルゴリズム開発者の名前) <p>既定は [DSA] です。</p>
[相手証明書認証の要求]	<p>チェックを付けると、このインストールに対して相手証明書認証が必要になります。</p>
[SSL/TLS Cipher Suites (TLSv1.2 and below)]	<p>TLSv1.2 以下用の暗号スイート。既定値は ALL:!aNULL:!eNULL:!EXP:!SSLv2 です。</p>
[SSL/TLS Cipher Suites (TLSv1.3)]	<p>TLSv1.3 用の暗号スイート。既定値は、 TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256 です。TLSv1.3 がサポートされているプラットフォームでのみ使用できます。</p>
[SSL/TLS 認証ファイル]	<p>Web ゲートウェイの SSL/TLS 証明書ファイルへのフル・パスです。証明書ファイルでサポートされているファイル形式は、InterSystems IRIS TLS 構成でサポートされている形式と同じです。</p> <p>C:\InterSystems\certificates\clcert.pem</p>
[SSL/TLS 秘密鍵ファイル]	<p>Web ゲートウェイの SSL/TLS 証明書に関連付けられた秘密鍵へのフル・パスです。証明書ファイルでサポートされているファイル形式は、InterSystems IRIS TLS 構成でサポートされている形式と同じです。</p> <p>C:\InterSystems\certificates\clikey.pem</p>

SSL/TLS パラメータ	機能
[SSL/TLS CA 認証ファイル]	Web ゲートウェイの証明書用の認証機関 (CA) の証明書へのフル・パスです。証明書ファイルでサポートされているファイル形式は、InterSystems IRIS TLS 構成 でサポートされている形式と同じです。 C:\InterSystems\certificates\cacert.pem
[SSL/TLS 秘密鍵パスワード]	SSL/TLS 秘密鍵のパスワードです。 また、UNIX®/Linux/macOS システムでは、このフィールドで、 パスワードをプログラムによって取得 するオペレーティング・システム・コマンドを中括弧({})で囲んで指定することもできます。

8.1.5 [オプションパラメータ]

オプション・パラメータについては、“[既定のパラメータの構成](#)”を参照してください。これらのパラメータのいずれかが空白の場合、その値は Web ゲートウェイのグローバル構成から継承されます (“[InterSystems IRIS への接続](#)”を参照)。

8.1.6 [エラーページ]

[エラーページ] のパラメータを使用すると、Web ゲートウェイの応答をカスタマイズできます。指定しない場合、パラメータはグローバル構成から継承されます。各パラメータの詳細は、“[カスタム・エラー・ページ](#)”を参照してください。

8.2 サーバ・アクセス・プロファイルのコピー

新しいサーバ・アクセス・プロファイルは、既存のサーバ・アクセス・プロファイルをコピーすることで、簡単に構成できます。コピーした場合、両方の構成エントリはサーバ名を除いて同一になります。2 番目の構成を編集して、変更できます (IP アドレスの変更など)。

Tip この機能は、構成を適切に調整する場合にも有用です。サーバの一時的な構成を別に作成することで、元の構成を失うことなくパラメータの変更をテストできます。

既存のサーバ・アクセス・プロファイルをコピーするには、以下の手順を実行します。

1. Web ゲートウェイ[管理ページ](#)のメイン・メニューから、[サーバ接続] を選択します。
2. [サーバ接続] 画面で、既存のサーバ名を選択します。
3. [サーバコピー] オプションを選択します。
4. [実行] を選択します。2 番目の構成画面が表示されます。
5. [サーバ名] テキスト・ボックスに、新しいサーバに付ける一意のわかりやすい名前を入力します。
6. [設定を保存] を選択します。

8.3 InterSystems IRIS サーバへのアクセスの無効化

この機能を使用すると、ユーザは、構成した InterSystems IRIS サーバにこのゲートウェイのインストール環境を介してアクセスできなくなります。

サーバへのアクセスを無効にするには、以下の手順を実行します。

1. Web ゲートウェイ[管理ページ](#)のメイン・メニューから、[サーバ接続] を選択します。
2. [サーバ接続] 画面で、既存のサーバ名を選択します。
3. [サーバ編集] オプションを選択します。
4. [実行] を選択します。サーバの構成画面が表示されます。
5. [サービス状態] パラメータで、[無効] を選択します。
6. [設定を保存] を選択します。

アクセスを再度有効にするには、上記の手順を繰り返して、手順 5 で [有効] を選択します。

8.4 サーバ・アクセス・プロファイルの削除

サーバ・アクセス・プロファイルを削除するには、以下の手順を実行します。

1. Web ゲートウェイ[管理ページ](#)のメイン・メニューから、[サーバ接続] を選択します。
2. [サーバ接続] 画面で、サーバ名を選択します。
3. [サーバ削除] オプションを選択します。
4. [実行] を選択します。
5. [はい: 削除] を選択して確認します。

9

Web アプリケーション・パスのアプリケーション・アクセス・プロファイルの定義

ここでは、InterSystems [Web ゲートウェイ](#)のアプリケーション・アクセス・プロファイルを構成する方法について説明します。アプリケーション・アクセス・プロファイルによって、Web ゲートウェイが特定の Web アプリケーション・パスの要求をルーティングする場所が決定されます。アプリケーション・アクセス・プロファイルでは、1 つのターゲット・アプリケーション・サーバの[サーバ・アクセス・プロファイル](#)を指定することも、一連のサーバ・アクセス・プロファイルを負荷分散またはフェイルオーバー・スキーマの一部として指定することもできます。

[Web アプリケーション](#)の要求をルーティングするには、要求の URL パスを含むパスのアプリケーション・アクセス・プロファイルを構成する必要があります。特定の要求については、指定された Web アプリケーションに複数の方法でその要求をルーティングすることもできます。詳細は、[こちらのページ](#)を参照してください。

これらおよび[その他の構成タスク](#)には、Web ゲートウェイ[管理ページ](#)または [Web ゲートウェイ・レジストリ・メソッド](#)を使用します。Web ゲートウェイは構成情報を [CSP.ini](#)に保持します。

重要 CSP.ini ファイルを直接編集しなければならないことがあるコンテナ化された導入を除き、CSP.ini ファイルへのアクセスを制限して、すべての Web ゲートウェイ構成を Web ゲートウェイ管理ページを使用して実行することをお勧めします。

9.1 アプリケーション・アクセス・プロファイルの追加

Web ゲートウェイが特定のアプリケーション・パスの要求をルーティングすべき場所と方法を指定するには、Web ゲートウェイ構成内でそのパスのアプリケーション・アクセス・プロファイルを定義します。そのためには、以下の操作を実行します。

1. Web ゲートウェイ[管理ページ](#)のメイン・メニューで、[\[アプリケーションアクセス\]](#)を選択します。
2. [\[アプリケーション追加\]](#)を選択します。多くのパラメータには既定の設定が入力されています。
3. [\[アプリケーションパス\]](#)テキスト・ボックスに、アプリケーションの一意のパスを入力します。このパスは、アプリケーションの URL に表示されるパスです。
4. このアプリケーションで使用されるもう 1 つの構成パスとサーバ・パラメータ (下の表を参照) を入力します。
5. 入力を終わったら、[\[設定を保存\]](#)を選択します。アプリケーション構成に加える変更は、そのアプリケーション・パスに対して新しいユーザ・セッションが作成されると有効になります。既存のユーザへの影響はありません。

9.1.1 アプリケーション・アクセス・プロファイルの構成パラメータ

基本パラメータは以下のとおりです。

パラメータ	機能
[サービス状態]	アプリケーションのパスを使用して、アプリケーションへのアクセスを有効および無効にします（既定は [有効]）。
[ウェブサーバ物理パス]	Web サーバ上の対応するディレクトリのパスです。この設定は、構成した各パスを、Web サーバ構成の仮想ディレクトリとして設定する必要がある Microsoft IIS システムでは特に重要です。IIS 内で定義される各仮想ディレクトリに、物理パスを関連付ける必要があります。IIS 用のこの追加構成手順の目的は、InterSystems IRIS によって使用されるパスが、実行許可で定義できるようにすることです。既定では、実行許可が拒否されるようになっています。
[追加の CGI 環境変数]	要求が行われるたびに InterSystems IRIS 環境に返される、追加の CGI 環境変数のコンマ区切りリストです。共通で使われる CGI 環境変数は、要求が行われるたびに自動的に送信されます。ワイルドカード文字 (*) を入力すると、要求があるたびに、 Web サーバから提供されるすべての環境変数 が Web ゲートウェイから InterSystems IRIS サーバに送信されるようになります。
[このクラスで処理する]	指定したクラスでこのパス内のファイルを処理します。これにより、専用の要求ハンドラを CSP に構築できます。
[GZIP 圧縮]	このパスで返されるすべての CSP ページの GZIP 圧縮を有効または無効にします（既定は [無効]）。
[GZIP 最小ファイル・サイズ]	GZIP 圧縮が呼び出される最小応答サイズ。単位はバイトです。既定は 500 バイトです。
[GZIP 除外ファイルの種類]	<p>GZIP 圧縮の除外対象とするファイルの種類のリストです。除外対象とするファイルは、MIME タイプ (image/jpeg など) または一般的な拡張子 (jpeg など) を使用してリストできます。</p> <p>既定では、これらの一般的な（標準で圧縮されている）イメージ・ファイルは除外されません。</p> <p>GZIP Exclude File Types: jpeg gif ico png gz zip mp3 mp4 tiff</p> <p>追加の種類または拡張子をスペースで区切ります。</p>

パラメータ	機能
[応答サイズの通知]	<p>このパラメータでは、それぞれの応答に含まれるデータの量をクライアントに通知するときに Web ゲートウェイで使用するメソッドの制御を構成できます。</p> <p>HTTP キープアライブ接続が使用されている場合、通常、Web クライアントには応答サイズを通知する何らかのフォームが必要になります。このような状況では、HTTP v1.1 が使用されている場合、Web ゲートウェイは既定でチャンク転送エンコーディングを使用します。これよりも前の HTTP プロトコルが使用されている場合は、InterSystems IRIS からの応答データはバッファされ、代わりに Content-Length ヘッダが生成されます。また、全体の応答が 1 つの出力バッファに収まる場合には、チャンク転送を使用する代わりに Content-Length ヘッダが生成されます。</p> <p>どちらのメソッドを使用するかを具体的に Web ゲートウェイに指示する方が望ましい場合もあります。例えば、HTTP v1.1 が使用されているにもかかわらず、媒体（プロキシなど）がチャンク転送を適切にサポートできない場合です。また、すべての Web クライアントで、いずれのフォームのサイズ通知も送信しない（応答のターミネータとして 接続の切断 イベントが使用される場合など）ようにできる必要がありますが、適切な方法としては、すべての応答に何らかのフォームのサイズ通知を付随させることをお勧めします。実際に、一部のクライアントにはこの方法が必要になります。</p> <p>以下のオプションが用意されています。</p> <ul style="list-style-type: none"> ・ チャンク転送エンコーディングと Content-Length (既定) ・ チャンク転送エンコーディング ・ Content-Length <p>このパラメータには、キープアライブを使用するかどうかにかかわらず、常にすべての要求に対してサイズ通知を生成するよう Web ゲートウェイに指示するためのチェック・ボックスが用意されています。</p> <p>チャンク応答とは異なり、Content-Length ヘッダを指定する HTTP 応答のサイズには 500 キロバイトの制限があります。この制限を超えると、CSP のログに警告メッセージが表示されます。</p> <pre>WARNING: Unable to generate a 'Content-Length' header directive for this oversize response (Current size: size; Maximum buffer size allowed: 500000)</pre>
[キープアライブ]	<p>このパスの HTTP [キープアライブ] 接続を有効または無効にします。既定値は [] です。この場合、[キープアライブ] ステータスは、各要求の HTTP 応答ヘッダにより決定されます。</p>
[非解析ヘッダ]	<p>このパスの [非解析ヘッダ] プロトコルを有効または無効にします。既定値は [有効] です。この場合、HTTP 応答ヘッダはクライアントに直接返送されます。このプロパティが無効化されている場合、応答ヘッダはホスト Web サーバに返送されます。これにより、ヘッダを解析し、指定されている出力フィルタがあれば、それを呼び出す機会が Web サーバに与えられます(例：Apache グループの mod_deflate 機能)。Apache Web サーバでキープ・アライブが有効になっている場合、非解析ヘッダの設定に関係なく、応答ヘッダは Apache に返送されます。</p>

9.1.2 サーバ・パラメータ

アプリケーションに対して使用する InterSystems IRIS サーバと、それらの使用目的のリストを定義できます。

パラメータ	機能
代替サーバの使用目的	<p>リストされている最初のサーバ [サーバ 0] は、既定の InterSystems IRIS サーバです。これが最初に使用されます。リストされている他のサーバは、チェックが付いているオプションに応じて、負荷分散またはフェイルオーバーに使用できます。</p> <ul style="list-style-type: none"> ・ フェイルオーバー：最初のサーバに障害が発生した（使用不能になった）場合に、代替を使用します。 ・ 負荷分散とフェイルオーバー：最初のサーバで障害が発生した場合、フェイルオーバーまたは負荷分散のいずれかとして構成されているサーバを使用します。
サーバの数：	<p>サーバのリスト。構成画面には、最初は 3 つのサーバ・スロットのみが表示されていますが、追加のスロットが表示され、任意の数の代替サーバを定義できます。各サーバに [有効] または [無効] としてチェックを付けることができます。既定値は常に [有効] です。詳細は、複数の InterSystems IRIS サーバ・インスタンス間の負荷分散とフェイルオーバー を参照してください。</p>

9.2 アプリケーション・アクセス・プロファイルのコピー

新しいアプリケーション・パスを簡単に構成するには、既存のパスのアプリケーション・アクセス・プロファイルをコピーして編集します。

Tip ヒン この機能は、構成を適切に調整する場合にも有用です。アプリケーション・パスの一時的なアプリケーション・アクセス・プロファイルを別に作成することにより、元の構成を失うことなくパラメータの変更をテストできます。

既存のアプリケーション・アクセス・プロファイルをコピーするには、以下の手順を実行します。

1. Web ゲートウェイ[管理ページ](#)のメイン・メニューから、[アプリケーションアクセス] を選択します。
2. [アプリケーションアクセス] の画面で、既存のアプリケーション・パスを選択します。
3. [アプリケーションコピー] を選択します。
4. [実行] を選択します。
5. [アプリケーションパス] テキスト・ボックスに、新しい一意のアプリケーション・パスを入力します。
6. [設定を保存] を選択します。新しいアプリケーション構成は、新しいユーザ・セッションが新しいアプリケーション・パスに対して作成されると有効になります。既存のユーザへの影響はありません。

9.3 アプリケーション・パスによるアクセスの無効化

この機能を使用すると、ユーザは構成したアプリケーションにこの Web ゲートウェイを介してアクセスできなくなります。

アプリケーション・パスを介したアクセスを無効にするには、以下の手順を実行します。

1. Web ゲートウェイ[管理ページ](#)のメイン・メニューから、[アプリケーションアクセス] を選択します。
2. [アプリケーションアクセス] の画面で、アプリケーション・パスを選択します。
3. [アプリケーション編集] を選択します。
4. [実行] を選択します。アプリケーション・パスの構成画面が表示されます。

5. [サービス状態] パラメータで、[無効] を選択します。
6. [設定を保存] を選択します。

アクセスを再度有効にするには、上記の手順を繰り返して、手順 5 で [有効] を選択します。

9.4 アプリケーション・アクセス・プロファイルの削除

既存のアプリケーション・アクセス・プロファイルを削除するには、以下の手順を実行します。

1. Web ゲートウェイ管理ページのメイン・メニューから、[アプリケーションアクセス] を選択します。
2. [アプリケーションアクセス] の画面で、アプリケーション・パスを選択します。
3. [アプリケーション削除] オプションを選択します。
4. [実行] を選択します。
5. Web サーバを再起動して、すべてのアプリケーションを強制的に再起動します。

10

Web ゲートウェイのシステム全体のパラメータの構成

特定の InterSystems IRIS® インスタンスにアクセスするように InterSystems IRIS Web ゲートウェイを構成した場合、インスタンスのサーバ・アクセス・プロファイル内で指定していないオプション・パラメータやカスタムのシステム・フォームは、Web ゲートウェイに対してグローバルに（システム全体で）指定された一連の既定パラメータから自動的に継承されます。例えば、インスタンスのサーバ・アクセス・プロファイル内で[サーバ応答タイムアウト]パラメータを設定していない場合、そのインスタンスはグローバルな[サーバ応答タイムアウト]の設定を継承します。

ここでは、Web ゲートウェイにグローバルな（システム全体の）構成パラメータを構成する方法について説明します。サーバ・アクセス・プロファイル（InterSystems IRIS インスタンス向け）とアプリケーション・アクセス・プロファイル（それらのインスタンスで利用できる Web アプリケーション向け）を構成する方法については他の記事で説明しています。

10.1 Web ゲートウェイ・パラメータの構成方法

Web ゲートウェイ管理ページ内の[デフォルトパラメータ]ページでは、Web ゲートウェイのすべてのグローバルな構成パラメータを変更できます。（このオプションを使用するには、システム管理者であることが必要です。）一般に、このインタフェースを使用して、Web ゲートウェイのグローバルなパラメータを構成することをお勧めします。このページの残りの部分では、ユーザがこの方法を使用していると想定しています。

一方、Web ゲートウェイ・レジストリの %CSP.Mgr.GatewayMgr クラスは、メソッド SetDefaultParms, を提供します。このメソッドでは、接続されている任意の InterSystems IRIS インスタンスから、プログラムによって Web ゲートウェイのグローバルなパラメータを構成できます。

Web ゲートウェイは、これらのシステム全体の既定パラメータを CSP.ini ファイルに保持します。“CSP.ini”では、以下で説明するパラメータに類似した名前が記載されています。類似パラメータ名を使用し、Web ゲートウェイ・レジストリと CSP.ini ファイルを使用して Web ゲートウェイを構成します。

重要

CSP.ini ファイルを直接編集しなければならないことがあるコンテナ化された導入を除き、CSP.ini ファイルへのアクセスを制限して、すべての Web ゲートウェイ構成を Web ゲートウェイ管理ページまたは Web ゲートウェイ・レジストリ・メソッドを使用して実行することをお勧めします。

Web ゲートウェイ・レジストリを使用して、または CSP.ini ファイルを編集して Web ゲートウェイのグローバルなパラメータを変更する場合、変更を有効にするには、構成を再ロードするように Web ゲートウェイを強制する必要があります。

10.2 Web ゲートウェイ (一般設定)

Web ゲートウェイの [デフォルトパラメータ] 管理ページのこのセクションには、Web ゲートウェイの一般設定を構成するための以下のパラメータが含まれます。

[インスタンスホスト名]

これは、Web ゲートウェイのこのインスタンスに対するネットワーク・ホスト名です。この Web ゲートウェイは既定値を生成し、その値はテキスト・ボックスの下に表示されます。このパラメータ値は、システム変数 CSPIHN として、要求データと共に InterSystems IRIS へ送信されます。アプリケーションはこの値を使用して、ネットワーク経由で Web ゲートウェイにより提供される管理サービスにアクセスできます。

このパラメータの形式は `server_name:port` です。

[最大接続数]

このゲートウェイ・インスタンスで作成可能な InterSystems IRIS への最大接続数。既定値は 1024 に設定されます。アプリケーションが使用する接続数が多いほど、この値を増加させることでアプリケーションの応答性を高めることができますが、サーバ・リソースをより使用する結果になる可能性もあります。

[最大接続数] パラメータへの変更は、Web ゲートウェイ (またはホスト Web サーバ) の再起動後にのみ有効になります。

[最大キャッシュ・サイズ]

CSP の応答データをキャッシュするために確保される共有メモリの最大量。

キャッシュ・サイズは、バイトの接尾語がない数字に、キロバイトの場合は **K**、メガバイトの場合は **M** を付加して指定されます。

このパラメータの既定値は 256K です。必要に応じてこの値を増やしたり減らしたりできます。

[最大キャッシュ・サイズ] パラメータへの変更は、ゲートウェイ (またはホスト Web サーバ) の再起動後に有効になります。

[Web サーバ ID クッキー]

Web サーバ ID クッキー (CSPWSEVERID) を抑制します。以下に設定できます。

- ・ 有効 (既定)
- ・ 無効

[Web サーバ ID クッキー] は、ロード・バランサを有効にして、Web アプリケーションのパッシブなクッキーの親和性を実装するために使用されます。ただし、このクッキーの自動生成を抑制する方が望ましい場合もあります。例えば、Web 要求を処理するために他のサーバに透過的に渡すプロキシ・アプリケーションなどがあります。

Web サーバ ID クッキーは、静的と見なされるリソース (イメージおよび JS ファイルなど) を返す場合は送信されません。このコンテキストでは、静的ファイルには、Web サーバ ID クッキーが付随しない InterSystems IRIS により生成されたすべての応答が含まれます。この規則の例外は、アプリケーションがセッション・クッキーを使用 するように構成されている場合です。この場合、Web サーバ ID クッキーはすべての応答に含められます (以前と同じ)。

10.3 セキュリティ

Web ゲートウェイの[デフォルトパラメータ]管理ページのこのセクションでユーザ名とパスワードが定義されている場合、すべてのシステム管理者は Web ゲートウェイ管理ページにアクセスする際に、このユーザ名とパスワードを入力する必要があります。

パスワードを忘れた場合は、以下の手順を使用して新しいパスワードを設定します。

1. **CSP.ini**を編集し、ファイルの **SYSTEM** セクションで **Username** と **Password** の新しい値を指定します。パスワードの値は平文で指定できます。
2. Web サーバを再起動します。Web サーバは構成を再ロードしてファイルを更新し、平文のパスワードではなくパスワード・ハッシュを保持します。

これでユーザは、更新されたユーザ名とパスワードで Web サーバにログインできます。

```
[SYSTEM]
Username=cm
Password=1Bx4tt88mttAWaf7isJg3Urqc2zE
```

管理ページでは、以下の Web ゲートウェイのセキュリティ・パラメータを構成できます。

[このフォームにアクセス]

Web ゲートウェイ管理ページのオプションへのアクセスを有効または無効にすることができます。既定は[有効]です。アクセスが[無効]である場合、Web ゲートウェイ管理ページを使用して、アクセスを再度有効にすることはできません。アクセスを再度有効化するには、**構成ファイル**を手動で編集します。このファイルの **SYSTEM** セクションで、**SM_Forms** パラメータを **Enabled** に設定します。

```
[SYSTEM]
SM_Forms=Enabled
```

[ユーザ名]

Web ゲートウェイ管理ページへのアクセスに必要なユーザ名です。

[パスワード]

Web ゲートウェイ管理ページへのアクセスに必要なパスワードです。

[パスワード(再入力)]

パスワードを修正する場合、ここで新しいパスワードを確認入力します。

[セッション・タイムアウト]

アクティブなシステム管理セッションがログオン状態になっているアイドル時間(秒)。この時間が経過すると、管理セッションの有効期限が切れ、管理者は Web ゲートウェイ管理ページから自動的にログアウトします。

[システム管理マシン]

これらのシステム管理オプションにアクセスできるクライアント・マシンの IP アドレスのリストを定義します。システム管理者のアクセス権を持つクライアントは、CSP システムへのアクセスの追加や削除、構成ファイル内の設定の変更、およびアクティブなセッションを閉じる操作が可能です。各アドレスは、コンマまたはプラス記号で区切ります。以下の例では、2 つのクライアントにシステム管理者のアクセス権があります。

```
127.0.0.1, 45.123.231.12
```

このフィールドを定義しない場合、Web ゲートウェイと同じマシンで動作しているクライアント (Web サーバのホスト) のみが CSP を構成できます。詳細は、“[追加クライアント・アドレスからのアクセスの有効化](#)”を参照してください。

このフィールドには、[ユーザ名とパスワードをオーバーライド]というチェック・ボックスも用意されています。チェック・ボックスにチェックを付けると、リストされているクライアント・マシンから管理フォームにアクセスするときにユーザ名やパスワードを入力しなくても済むようになります。

カスタム・ログイン・フォーム

Web ゲートウェイ管理ページへのアクセスを制御するカスタム・ログイン・フォームを定義します。このパラメータは、物理ファイルまたはフォームを処理するホスト Web サーバを有効にするリンクへのフル・パスにすることができます。

例：

```
C:\Inetpub\wwwroot\login.html
/login.html
```

物理ファイル名が指定されている場合、Web ゲートウェイはフォームを取得してクライアントに送信します。そうでない場合、「HTTP リダイレクト」応答ヘッダを送信して、クライアントがホスト Web サーバから直接フォームを要求できるようにします。カスタム・フォームでゲートウェイ管理者がログインするには、HTTP POST 要求を実装する必要があります。

必須フォーム・フィールドは以下のとおりです。

```
<FORM METHOD=POST ACTION="/csp/bin/Systems/Module.cxx">
<INPUT TYPE=HIDDEN NAME="CSPSYS" VALUE="17">
<INPUT TYPE=HIDDEN NAME="CSPSYSsmSection" VALUE="SYSTEM">
<INPUT TYPE=TEXT NAME="CSPUNM" SIZE='20' VALUE="">
<INPUT TYPE=PASSWORD NAME="CSPPWD" SIZE='20' VALUE="">
<INPUT TYPE=SUBMIT NAME="CSPSYSbok" VALUE="Login">
```

ここで、CSPUNM はユーザ名で、CSPPWD はパスワードです。ログイン (送信) ボタン (上記で Login として表示) に割り当てられるテキストは変更可能です。

単純ですが完全な例を以下に示します。

```
<html>
<head>
<title>Web Gateway Management</title>
</head>
<h2>Web Gateway Management</h2>
<FORM METHOD=POST ACTION="/csp/bin/Systems/Module.cxx">
<INPUT TYPE=HIDDEN NAME="CSPSYS" VALUE="17">
<INPUT TYPE=HIDDEN NAME="CSPSYSsmSection" VALUE="SYSTEM">
<BR>
Username:
<INPUT TYPE=TEXT NAME="CSPUNM" SIZE='20' VALUE="">
<BR>
Password:
<INPUT TYPE=PASSWORD NAME="CSPPWD" SIZE='20' VALUE="">
<BR>
<INPUT TYPE=SUBMIT NAME="CSPSYSbok" VALUE="Login">
</form>
</html>
```

10.4 InterSystems IRIS への接続

このセクションでは、InterSystems IRIS への接続の維持に関連するパラメータについて説明します。

[サーバ応答タイムアウト]

ターゲットの InterSystems IRIS サーバが Web サーバからの要求に応答できるまでの最大許容時間 (秒)。タイムアウトとは、アクティビティのない時間のことです。例えば、HTML データの行を毎秒 10 時間送信していれば、タイムアウトは発生しません。このフィールドに入力できる最小値は 5 秒です。

ここで設定した値がシステムの既定値です。継承値が指定されていない場合、値は [デフォルトパラメータ] ページから取られます。ただし、サーバ固有の構成またはアプリケーション自体で別の値を設定できます。

Apache サーバがある場合は、Apache `httpd.conf` ファイルの `Timeout` を使用してこの値を設定できますので注意してください。これら 2 つのうち値の低い方が最初にトリガされます。

[キューイングされたリクエストのタイムアウト]

要求が、該当する InterSystems IRIS システムへの使用可能な接続をキュー内で待機できる最大時間 (秒)。入力できる最小値は 5 秒です。継承値が指定されていない場合、値は [デフォルトパラメータ] ページから取られます。

[無使用タイムアウト]

このパラメータが該当するのはステートレス接続のみです。このパラメータは、ステートレス接続が開いたままのアイドル状態から閉じるまでの最大時間 (秒) を示します。このタイムアウトを超えると、セッションは自動的に閉じます。この機能を使用すると、InterSystems IRIS サーバにステートレス・セッションが蓄積されません。特に、負荷の増加に対応するために多数の接続が開かれた高アクティビティ期間の後でも、ステートレス・セッションが蓄積されません。この値を指定しない場合は、手動で閉じるまでステートレス接続は開いたままです。継承値が指定されていない場合、値は [デフォルトパラメータ] ページから取られます。

構成されたタイムアウトを過ぎても、最大数分間はプロセスが残る場合があります。設計上、Web ゲートウェイは接続タイムアウトを定期的にチェックしています。タイムアウトになってもすぐに通知されるわけではありません。このチェックのタイミングによって、プロセスは最大 420 秒残存します。

[すべての接続にタイムアウトを適用]

[無使用タイムアウト] オプションを (最小の接続プール構成を含む) すべての接続に適用します。このオプションにチェックを付けていない場合、Web ゲートウェイは [無使用タイムアウト] を最小の接続プール ([サーバ接続最小数] パラメータにて定義) に適用しません。このオプションにチェックを付けている場合、Web ゲートウェイはタイムアウトをプールの接続すべてに適用します。このオプションは、CSP の使用率がきわめて低いインストール環境で使用するものであり、その結果として、すべての CSP プロセスでタイムアウトよりもこのオプションが優先します。継承値が指定されていない場合、値は [デフォルトパラメータ] ページから取られます。

[イベントログレベル]

Web ゲートウェイ・イベント・ログに書き込まれる情報を制御します。詳細は、“イベント・ログ・パラメータ”を参照してください。

[イベントログファイル]

Web ゲートウェイ・イベント・ログの場所とファイル名を指定します。指定しない場合、ログは Web ゲートウェイのインストール環境をホストするディレクトリに書き込まれます。例えば、以下のようになります。

代替の場所を指定するには：

`/opt/logfiles/cspgateway/`

代替の場所およびファイル名を指定するには：

`/opt/logfiles/cspgateway/event_log_01012006.log`

[すべてのログファイルを残す]

[イベントログ切り替えサイズ] が空白の場合 (既定)、Web ゲートウェイ・イベント・ログは管理者が手動でクリアするまで増大します。ファイルの容量が [イベントログ切り替えサイズ] で指定されている場合、InterSystems IRIS はこのログ・ファイルを **filename.old** (filename は元の完全なファイル名) という名前のファイルにコピーします。その後のログのローテーションによって、**filename.old** は、ログの現在の内容に上書きされます。すべてのログ・ファイルを残すには、[すべてのログファイルを残す] にチェックを付けます。コピーの実行時に、各ログに日付と時刻を含む名前が付けられます。

[イベントログ切り替えサイズ]

このパラメータには、ログのローテーションが開始されるサイズを定義します。既定値は空白です。これは、Web ゲートウェイが 1 つのログ・ファイルを管理し、管理者がそのファイルを手動で消去するまでファイルのサイズが増えることを意味します。

ローテーションが必要な場合、サイズはバイトの接尾語がない数字に、キロバイトの場合は **K**、メガバイトの場合は **M** を付加して指定されます。

指定可能な最小サイズは 100K です。この値は、管理者が管理スイートでこれよりも低い値を設定しようとするとき自動的に設定されます。

ローテーションされたログ・ファイルのコピーは、保持される場合、ローテーションの日付と時刻に応じて以下のように名前が付けられます。

CSP_YYYYMMDD_hhmm.log

YYYY は年、MM は月、DD は日付、hh は時間、mm はその時間を経過した分です。以下に例を示します。

CSP_20090109_1830.log (Log rotated at 18:30 on 9th January 2009)

1 分間に複数のログ・ファイル・ローテーションが行われる場合は、重複を防ぐために、ファイル名にシリアル番号が付加されます。以下に例を示します。

03/12/2015 17:02	106,660 CSP_20151203_1702.log
03/12/2015 17:02	124,752 CSP_20151203_1702.log.0001
03/12/2015 17:02	124,752 CSP_20151203_1702.log.0002

保持されないローテーションされたログ・ファイルには、**filename.old** (filename は元の完全なファイル名) という名前が付けられます。

この機能を使用するには、Web ゲートウェイ・バイナリ (つまり、メインのログ・ファイルが格納されている場所) をホストするディレクトリに対する Web ゲートウェイの作成/書き込みアクセス権が必要です。Web ゲートウェイが正常にローテーションを実行できない場合は現行のログ・ファイルに対して書き込みが続けられます。

このフィールドには、[すべてのログファイルを残す] というチェック・ボックスも用意されています。このチェック・ボックスにチェックを付けると、前述した名前付け方式に従ってすべてのログ・ファイルを保存するよう Web ゲートウェイに指示されます。

[ログするリクエスト最大サイズ]

V9 (または V9b などのバリエーション) の **イベント・ログ・レベル** を指定することで、HTTP 要求のログを有効にしている場合、このパラメータは、ログ内に含める HTTP 要求の量を指定します。この最大サイズを超える要求は切り捨てられます。

このパラメータの既定値は 256K で、最小値は 40K です。このフィールドを空のままにすると、既定値 (256K) に設定されます。最低値は強制されます。この最低値より小さい値を割り当てようとするとき、40K に設定されます。

[SSL/TLS ライブラリ・パス]

OpenSSL ライブラリへのパスを指定します。UNIX® の場合、これらのファイルは `libssl.so` および `libcrypto.so`、Windows の場合、これらのファイルは `libcrypto-1_1-x64.dll` および `libssl-1_1-x64.dll` です。既定では、Web ゲートウェイは、そのホーム・ディレクトリでローカルにこれらのライブラリを参照します。詳細は、“[Kerberos ライブラリ](#)”の“SSL/TLS を使用する場合のライブラリ・パスのオーバーライド”を参照してください。

[保持モード除外ファイルの種類]

静的ファイルがステート認識アプリケーションで非同期に処理されるようにします。ステートレス・アプリケーションでは、静的ファイル (`csp`、`cls`、`csr`、および `zen` 以外のファイル) はメイン・セッションに対して非同期に処理されます。つまり、これらのファイルの要求はセッション・ロックをバイパスして、アプリケーションのメイン処理ストリームの外部で同時に処理できます。

このパラメータにより、この方式がステート認識アプリケーションにも拡張されます。ステート認識アプリケーションは、従来のセッション・ロックだけでなく、Web ゲートウェイ内の接続ロックの対象にもなります。接続ロックは、ユーザ/セッションに対するすべての要求が、同じ InterSystems IRIS プロセスに確実にルーティングされるようにします。InterSystems IRIS から提供される静的コンポーネントに依存するアプリケーションの場合、これによって過剰な要求がキューイングされ、結果としてブラウザの動作が不安定になる (停止など) 場合があります。

このパラメータを使用して、ファイルの種類 (拡張子ごと) のスペースで区切られたリストを定義し、非同期の処理を可能にして、Web ゲートウェイおよび InterSystems IRIS において接続/セッション・ロックから除外されるようにします。リストに `*` (アスタリスクとハイフン) の接頭語が付けられている場合、以下のリストで定義されたものを除くすべてのファイルが非同期で処理されます。

例

```
Preserve Mode Exclude File Types=gif jpg jpeg
```

ステート認識セッションに対して GIF、JPG、および JPEG の種類のファイルを非同期で処理します。

```
Preserve Mode Exclude File Types=- csp cls csr zen
```

ステート認識セッションに対して、CSP、CLS、CSR、および ZEN の種類を除くすべてのファイルを非同期で処理します。なお、これはステートレス・アプリケーションに対して [CSP エンジン](#) で適用されるルールです。

このメカニズムは、[ログ・レベル v4](#) を使用して監視できます。要求に対して起動されると、以下に示すようなレコードがログに追加されます。

```
>>> Time: Fri Oct 04 14:56:40 2017 ...GET /csp/samples/zenutils.js
State-Aware Session (preserve == 1)
Process this request concurrently in the pool of stateless connections (File Type=js)
```

10.5 ASP リダイレクト

[ウェブドキュメントルート]

Web サーバのドキュメント・ルート・ディレクトリの完全な物理パスです。例えば、Microsoft IIS Web サーバの場合、このパスは通常 `c:\inetpub\wwwroot` となります。このパラメータは、CSP 内でこの機能を使用して、Microsoft ASP エンジンを通じて CSP 出力を送信し、最後のページを表示する場合にのみ必要です。

[ASP一時ディレクトリ]

Web ゲートウェイが Microsoft ASP のコンテンツを一時的に格納できるディレクトリの完全な物理パスです。このパラメータは、CSP 内でこの機能を使用して、Microsoft ASP エンジンを通じて CSP 出力を送信し、最後のページを表示する場合にのみ必要です。

10.6 [内部HTTPサーバ]

このセクションは NSD にのみ関連します。このセクションには、以下のパラメータが含まれます。

[サービス状態]

HTTP サーバは、[有効] または [無効] のいずれかに設定できます。以下のどちらかを選択します。

- ・ 有効
- ・ 無効

既定は [] です。

NSD が未処理の HTTP 要求に応答できるようにする場合を除き、セキュリティの面ではこの機能を無効にすることをお勧めします。

[NSDドキュメントルート]

NSD 自体をスタンドアロンの Web サーバとして使用する場合、このパラメータは Web ドキュメント・ルートの完全な物理パスを定義します。以下はその例です。

```
/opt/webgateway/home/
```

このサーバを使用して [Web アプリケーション](#) を提供する場合は、ブローカ・コンポーネントを以下にインストールする必要があります。

```
/opt/webgateway/home/broker/
```

CSP サンプルのサポートに使用する静的ファイルは以下のとおりです。

```
/opt/webgateway/home/samples/
```

管理ポータルをサポートに使用する静的ファイルは以下のとおりです。

```
/opt/webgateway/home/sys/
```

10.7 カスタム・エラー・ページ

グローバルな構成画面の [エラーページ] セクションでは、Web ゲートウェイのエラー・メッセージとシステム応答をカスタマイズできます。グローバルに設定できるほか、InterSystems IRIS サーバごとに設定することもできます。既定の CSP 応答をカスタマイズするには、以下の手順を実行します。

1. Web ゲートウェイ [管理ページ](#) のメイン・メニューから、[デフォルトパラメータ] を選択します。
2. [エラーページ] セクションで、対応するゲートウェイのページを置き換える CSP ページの名前を入力します。CSP ページの完全な物理パスを入力するか、Web ゲートウェイのパスを基準とする相対パスを入力します。
3. [設定を保存] を選択します。

次の Web ゲートウェイ・システム応答をカスタマイズできます。

[サーバエラー]

Web ゲートウェイに内部エラーが発生した場合に表示されるページ。例えば、InterSystems IRIS サーバとの通信に問題があるとエラーが発生します。特定のエラーは [Web ゲートウェイ・イベント・ログ](#) に必ず記録されます。

[サーバビジー]

使用可能な CSP 接続がすべて使用中の場合に表示されるページ。

[サーバが利用可能ではありません]

InterSystems IRIS サーバ（またはアプリケーション）が、構成内で故意に無効にされている場合に表示されるページ。

[サーバタイムアウト]

要求がタイムアウトしたときに表示されるページ。

[接続が閉じられました]

ステート認識セッションからログアウトした場合に表示されるページ。

10.8 イベント・ログ・パラメータ

[イベントログレベル] フィールドは、Web ゲートウェイが [Web ゲートウェイ・イベント・ログ](#) に書き込む情報を指定します。ログ・オプションは文字列として定義され、各文字がログ・コマンドを表します。ここでログ・レベルについて設定した値がシステム（つまり、すべての InterSystems IRIS サーバ）の規定値となります。特に明記されていない限り、個々の InterSystems IRIS サーバに異なる値を設定できます。

CSP Web ゲートウェイ管理ページのメニューから、ログを参照または消去できます。以下に示すログ・パラメータは、主にトラブルシューティングに使用します。

ログ・オプション	機能
E	すべてのエラーを記録します。接続の失敗を監視できます。
V	Verbose : Web ゲートウェイと InterSystems IRIS システム間の基本的な接続ダイアログを記録します。このオプションを使用して、Web ゲートウェイと InterSystems IRIS サーバ間の通信に存在する重要なポイントを記録します。このコマンドには 7 つのレベル（1 ～ 7）があります。これらのレベルごとに、詳細情報が記録されます。レベルは累積されます。例えば、レベル V3 には、V1 および V2 に指定したすべてのログ情報が含まれます。
EV	基本的なイベント・ログを有効にするには EV と入力します。ログのレベルが高くなるほど、ログ・ファイルには大量のデータが生成されます。したがって、レベルの高いログは問題を診断する場合にのみ使用してください。プロダクション・システムの場合、ログのレベルを EV より低く設定することをお勧めします。
V1	V と同じです。

ログ・オプション	機能
V2	<p>上記のレベルに指定した情報のほかに、以下の情報を記録します。</p> <ul style="list-style-type: none"> Web ゲートウェイと InterSystems IRIS 間の基本的な接続管理に関する情報（接続ごとの開始ポイントおよび終了ポイント）。 ブラウザから受信した送信の中断。 （InterSystems IRIS からの応答がない、またはその他のエラーが原因で接続が回復できないために）InterSystems IRIS への接続が強制的に切断された場合。 ステート認識（保持モード 1）セッションでのアクセス違反（無効なセッション ID など）。
V3	<p>上記のレベルに指定した情報のほかに、InterSystems IRIS ヘッダおよび HTTP ヘッダの情報を記録します。</p> <p>注:このログ・レベルが個々のサーバに対して指定されている場合、要求ヘッダはログに記録されませんが、応答ヘッダやその他のデータは記録されます。</p>
V4	<p>上記のレベルに指定した情報のほかに、ステート認識セッションのシリアル化に関する情報を記録します。</p> <p>注:このログ・レベルが個々のサーバに対して指定されている場合、要求ヘッダはログに記録されませんが、応答ヘッダやその他のデータは記録されます。</p>
V5	<p>上記のレベルに指定した情報のほかに、WebSocket プロトコルを介して InterSystems IRIS との間で送受信されたデータ・バッファの内容を記録します。データ・フレーミング（該当する場合）もすべて記録されます。最後に、作成された WebSocket の性質に関する詳細情報も初期接続時に記録されます。以下に例を示します。</p> <ul style="list-style-type: none"> WebSocket 接続 InterSystems IRIS で受け入れられた WebSocket 接続 : WSClassProtocolVersion=2; SharedConnection=0; NoDataFraming=2; BinaryData=1; <p>注:このログ・レベルが個々のサーバに対して指定されている場合、要求ヘッダはログに記録されませんが、応答ヘッダやその他のデータは記録されます。</p>
V6	<p>上記のレベルに指定した情報のほかに、以下の情報を記録します。</p> <ul style="list-style-type: none"> InterSystems IRIS に送信するデータ・ブロックのヘッダ。 Web サーバからの要求データ（マルチパートの添付を除く）。 InterSystems IRIS から受信したデータ・ブロックのヘッダ。 <p>注:このログ・レベルが個々のサーバに対して指定されている場合、要求ヘッダはログに記録されませんが、応答ヘッダやその他のデータは記録されます。</p>
V7	<p>上記のレベルに指定した情報のほかに、InterSystems IRIS から返された内容全体を記録します。</p> <p>注:このログ・レベルが個々のサーバに対して指定されている場合、要求ヘッダはログに記録されませんが、応答ヘッダやその他のデータは記録されます。</p>

ログ・オプション	機能
V9	<p>着信 HTTP 要求データを記録します。すべての HTTP 要求の本文すべてが記録されます。このログ指示文は、さらに拡張し、改善することができます。</p> <ul style="list-style-type: none"> ・ v9r : HTTP 要求すべてのログに加えて、HTTP 応答すべてを記録します。 ・ v9a : Web ゲートウェイのホーム・ディレクトリで http.log に HTTP 要求をすべて記録します。 ・ v9b : セッションごとに HTTP 要求をすべて記録します。ログ・ファイルは、http[session_id].log の形式で、Web ゲートウェイのホーム・ディレクトリに作成されます。ここで、session_id は 10 バイトのセッション ID です。 ・ v9m : Web ゲートウェイのホーム・ディレクトリにマルチパート・ポストをすべてログします。未処理の着信 HTTP 要求は、個々のコンポーネントと共に、暗号化された形式と解読された形式の両方で記録されます。 <p>注: V9、V9r、V9a、および V9b の形式は、個々のサーバに対して指定された場合、影響はありません。これらのログ形式は、既定のレベルでのみ有効にできます。</p>
s	<p>セッション : セッション・トークンの管理に関する以下の情報を記録します。</p> <ul style="list-style-type: none"> ・ 新しいセッション ID が割り当てられるポイント。 ・ 既存のセッションについて : セッション・トークンが cookie と形式/URL 変数 CSPCHD のどちらから抽出されたものかを示します。 ・ すべての要求について : InterSystems IRIS に送信される最終セッション ID。 <p>注: このログ・オプションは、個々のサーバに対して指定された場合、影響はありません。このオプションは、既定のレベルでのみ有効にできます。</p>

ログ・オプション	機能
c	<p>接続 : Kerberos ライブラリを使用して作成された接続に関する情報を記録します (IRISCONNECT)。</p> <p>小文字の <code>c</code> のログ・レベルを含めて、呼び出されたすべての IRISCONNECT 関数の完全な監査を、指定された入力パラメータおよび返された結果と共に記録するよう、Web ゲートウェイに指示します。簡潔にするため、InterSystems IRIS との間の入力および出力バッファの内容はこのレベルでは記録されません。IRISCONNECT 関数の呼び出しに加えて、入力および出力バッファの内容を記録するには、大文字の <code>C</code> のログ・レベルを設定します。</p> <p>Web ゲートウェイで提供されるログ機能のほかに、内部プロセスを記録する詳細なトレースを生成するよう、IRISCONNECT ライブラリに指示することもできます。さらに IRISCONNECT トレースの生成も要求するには、必要なトレースのタイプを示す数字を <code>c</code> ディレクティブに追加します。</p> <p>例えば、標準のゲートウェイ・ログ・エントリにログ・レベル <code>c3</code> を追加すると、レベル 3 の IRISCONNECT トレースが生成されます。有効な IRISCONNECT トレース・レベルは 1 ～ 6 で、以下のように定義されます。</p> <ul style="list-style-type: none"> ・ 6 – エラー ・ 5 – 警告 ・ 4 – 情報メッセージ ・ 3 – 出力データ ・ 2 – 入力データ ・ 1 – 通常イベント <p>Web ゲートウェイのログ・レベルとは異なり、IRISCONNECT トレースでは、ログ・レベルが高いほど詳細ではなくなります。このため、ログ・レベル 1 が最も詳細なトレース・ファイルを提供します。Web ゲートウェイは、Web ゲートウェイのホーム・ディレクトリにある <code>irisconnect.log</code> というファイルにトレースを保持するよう、IRISCONNECT ライブラリに指示します。このファイルのセキュリティ上の考慮事項および権限は、Web ゲートウェイ・イベント・ログ と同じです。</p> <p>注: IRISCONNECT トレースはプロセスごとにしか有効にできないため、サーバに対して完全に分離することはできません。いったん構成したら、新しい SSL 接続が試行されるまで、トレース・ログ生成はトリガされません。</p>

ログ・オプション	機能
t	<p>転送：Web ゲートウェイが送受信した未処理のデータ・バッファを記録します。このオプションの形式は <code>t[x][y]</code> です。</p> <p>値 <code>x</code> は、Web ゲートウェイと InterSystems IRIS の間で転送されたデータ・バッファを記録するよう Web ゲートウェイに指示し、値 <code>y</code> は、ホスト Web サーバ経由で Web ゲートウェイとクライアントとの間で転送されたデータ・バッファを記録するよう Web ゲートウェイに指示します。</p> <p><code>x</code> および <code>y</code> に利用可能な値は、以下のとおりです。</p> <ul style="list-style-type: none"> ・ 0：転送データを記録しません。 ・ 1：要求データのみを記録します。 ・ 2：応答データのみを記録します。 ・ 3：要求データと応答データを記録します。 <p>小文字 <code>t</code> を使用した場合、Web ゲートウェイは各バッファの転送データの最初の 256 バイトのみを記録します。大文字 <code>T</code> を使用した場合、Web ゲートウェイはデータ・バッファ全体を記録します。出力できない文字はすべて、エスケープされた形式で記録されます。</p> <p>注：このログ・レベルが個々のサーバに対して指定されている場合、<code>y</code> オプションにより、クライアントに送信された応答バッファは記録されますが、クライアントから受信する要求バッファは記録されません。</p>
p[n]	<p>パフォーマンス：CSP インストールのパフォーマンスを評価するための情報を取得するようにゲートウェイに指示します。</p> <p><code>n</code> に指定した秒数（サービス時間の合計）を下回ると、要求に対するデータは記録されません。例えば、指示文 <code>p</code> ではすべての要求のデータが記録されますが、<code>p2</code> ではサービス時間が 2 秒を超える要求のデータが記録されます。</p> <p>記録される情報は以下のとおりです。</p> <ul style="list-style-type: none"> ・ 要求にサービスを提供した総時間：要求にサービスを提供するために費やされた時間の合計（要求が Web ゲートウェイに到着してから、応答データの最後のバイトが Web ゲートウェイ環境を離れるまでの時間）。 ・ InterSystems IRIS への〔新規〕接続を取得：要求が Web ゲートウェイに到着してから、要求にサービスを提供するために予約されていた InterSystems IRIS に接続するまでにかかった時間。記録されたメッセージは、（再使用されている既存の接続に対して）この時間中に新しい接続が作成されたかどうかを示します。 ・ InterSystems IRIS に要求を送信：要求データの先頭バイトを Web サーバから読み取り、最終バイトを InterSystems IRIS に送信するまでにかかった時間。 ・ InterSystems IRIS で要求を処理：要求データの最終バイトを InterSystems IRIS に送信してから、応答データの先頭バイトが Web ゲートウェイで受信されるまでにかかった時間。 ・ InterSystems IRIS から応答を受信：応答データの先頭バイトを InterSystems IRIS から受信してから、最終バイトを Web サーバに送信するまでにかかった時間。

ログ・オプション	機能
p[n]([v])	<p>パフォーマンス・モニタの結果に基づいて、詳細ログを条件付きで有効にする機能を提供します。処理に一定時間以上かかる要求について、詳細情報を記録する必要がある場合に便利です。</p> <p>n は、パフォーマンス・データが記録されるオプションの処理時間下限値 (秒単位) で、v は必要な詳細ログ・レベルです。</p> <p>このメカニズムは、詳細イベント・ログと HTTP ログ設定に適用されます。エラー情報を記録する要求の場合は、パフォーマンス・モニタによって記録されるかどうかに関係なく、常にすべての要求に e が適用されます。</p> <p>以下に例を示します。</p> <p>ep5 (v9)</p> <p>このオプションでは、すべての要求について、要求を処理する際に発生したすべてのエラーを記録します (e)。さらに、HTTP 要求メッセージを記録しますが (v9)、処理時間が 5 秒を超える要求のみを記録します (p5)。</p> <p>ゲートウェイ・イベント・ログは、パフォーマンスに与える影響を最小限に抑え、使用するシステム・リソースに関して専有するフットプリントが小さくなるように設計されています。このため、以下の制限が適用されます。</p> <ul style="list-style-type: none"> ・ 個々の設定ごとに、1 つの詳細ログ・レベルのみを指定できます。すなわち、パフォーマンス・モニタによって記録される要求にレベル v9 を指定し、他のすべての要求にレベル v2 を指定することはできません。例えば、v2p5 (v9) を指定した場合、条件付きで適用された v9 レベルのみが使用されます。 ・ Web ゲートウェイ構成では、イベント・ログ・レベルをグローバルにも、サーバごとにも指定できます。詳細ログが有効な場合、一部のレコードはターゲットの InterSystems IRIS サーバがそのように指定される前に書き込み済みであるため、最善の結果を得るには、[デフォルトパラメータ] で条件付きログをグローバル・レベルで指定することをお勧めします。

ログ・オプション	機能
pp[n]	<p>詳細の時間計測情報を以下に示します。</p> <ul style="list-style-type: none"> ・ 要求の前処理：対象 InterSystems IRIS サーバの特定にかかった時間。Web サーバからの初期受け渡しおよびサーバを特定するための基本的な要求処理を含みます。 ・ InterSystems IRIS への [新規] 接続を取得：接続を適切な InterSystems IRIS サーバに割り当てるまでにかかった時間。（既存の接続の再使用ではなく）新しい接続が作成されたかどうかを示します。 ・ 要求をフォーマット：InterSystems IRIS に送信する要求メッセージの解析およびフォーマットにかかった時間。 ・ InterSystems IRIS に要求を送信：要求データの先頭バイトを Web サーバから読み取り、最終バイトを InterSystems IRIS に送信するまでにかかった時間。 ・ InterSystems IRIS で要求を処理：要求データの最終バイトを InterSystems IRIS に送信してから、応答データの先頭バイトが Web ゲートウェイで受信されるまでにかかった時間。 ・ 応答の後処理 (b)：Content-Length ヘッダが必要な場合、Web サーバを介して応答データをクライアントに送信して戻すまでにかかった時間が報告されます。 ・ 応答の後処理 (c)：応答を送信してから、InterSystems IRIS からの応答フッタのデータを Web ゲートウェイが読み取る準備が完了するまでにかかった時間。フッタのデータは、Web ゲートウェイと InterSystems IRIS の間の内部通信プロトコルの一部であり、制御情報（セッションの保持設定を変更する命令など）を含みます。 ・ InterSystems IRIS からフッタを受信：InterSystems IRIS から応答フッタのデータを受信するまでにかかった時間。 ・ フッタの後処理：フッタのデータを処理し、受信した命令に応答するまでにかかった時間。 ・ InterSystems IRIS への接続を解放：アクティブな接続を InterSystems IRIS に解放するまでにかかった時間。 ・ クリーンアップ：要求にサービスを提供する際に使用したリソースを解放し、ホスト Web サーバに制御を返すまでにかかった時間。

ログ・オプション	機能
W (または w)	<p>Windows では、クラッシュが発生するとメモリ・ダンプが生成されます。このオプションは大文字と小文字を区別しません。</p> <p>AIX では、<code>gencore</code> ユーティリティを使用してコア・ファイルが生成されます。このオプションは大文字と小文字を区別しません。</p> <p>Linux または MacOS では、このオプションは大文字と小文字を区別します。<code>w</code> を指定すると、<code>gcore</code> を使用して標準のコア・ダンプが生成されます。<code>W</code> を指定すると、<code>gcore -a</code> を実行してすべてのメモリ・マッピング (共有メモリを含む) がコア・ファイルにダンプされます。</p> <p>Unix システムでは、次の前提条件が満たされる必要があります。</p> <ul style="list-style-type: none"> • <code>gcore</code> (Linux または MacOS) または <code>gencore</code> (AIX) がマシン上に存在し、PATH 環境変数により使用可能であること。Linux および MacOS システムでは、<code>gcore</code> のバージョンが <code>-a</code> コマンド行オプションをサポートしている必要があります。 • Web サーバのワーカ・プロセスに、Web ゲートウェイ・モジュールが配置されているディレクトリへの書き込み権限があること。既定のインストールの場合、このディレクトリは <code>/opt/webgateway/bin</code> です。 • Root 以外のプロセスに、同じユーザ ID で実行される別のプロセスのコア・ダンプを生成する権限があること。MacOS では、システム整合性保護が無効である必要があります。 <p>Linux では、(RHEL および Ubuntu システムのように) Yama セキュリティ・モジュールが存在する場合、コマンド <code>echo 0 sudo tee /proc/sys/kernel/yama/ptrace_scope</code> を実行して、次の再起動まで必要な権限を付与します。この権限を永続的に付与するには、ファイル <code>/etc/sysctl.d/10-ptrace.conf</code> を作成または編集します。“<code>kernel.yama.ptrace_scope</code>” で始まる行がある場合、これを “<code>kernel.yama.ptrace_scope = 0</code>” に変更します。そのような行が存在しない場合は、“<code>kernel.yama.ptrace_scope = 0</code>” を追加してから、<code>sysctl -p</code> を実行します。</p> <p>注釈 セキュリティ上の理由により、このような権限は一時的にのみ付与することをお勧めします。</p>

11

InterSystems IRIS への Web ゲートウェイ接続の保護

ここでは、Web ゲートウェイから InterSystems IRIS® への接続を保護するオプションについて説明します。CSP 認証の詳細は、“[認証ガイド](#)”を参照してください。InterSystems IRIS への Web ゲートウェイ接続は、以下のセキュリティ・レベルに応じて保護できます。

1. [最小の接続セキュリティ](#) (非推奨)
2. [ユーザ名ベースおよびパスワードベースの単純な認証](#)
3. [Kerberos ベース認証とデータの保護](#)
4. [SSL/TLS ベース認証とデータの保護](#) (相互 TLS 認証を含む)

ここで適用されるセキュリティは、InterSystems IRIS サーバに対して Web ゲートウェイのホストを認証する目的のみで 사용되는ことに注意してください。Web ゲートウェイと [InterSystems IRIS アプリケーション・サーバ](#) (%cspServer) 間に承認されていない接続が生成されないように保護します。ただし、Web アプリケーションの個々のユーザは識別されません。ユーザのログイン機能が Web アプリケーション自体に用意されている場合に限り、Web アプリケーションのユーザを確実に識別できます。例えば、管理ポータルにログオンしているシステム管理者の場合、管理ポータルのログイン・フォームに入力したユーザ名とパスワードによってのみ識別できます。

Web のステートレスな特性も、念頭に置く必要があります。InterSystems IRIS への Web ゲートウェイ接続と、Web アプリケーションの個々のユーザとの間には、固定された関係が存在しません。多くのユーザが同じ接続を共有します。

接続時に InterSystems IRIS に対して Web ゲートウェイを認証することは重要です。攻撃者が Web ゲートウェイになりすますことができると、その攻撃者は、技術的な手段またはソーシャル・エンジニアリング、およびその両方を使用して自分の制御下に置いたシステムでトラフィックをリダイレクトし、データを自由に読み取り、変更できるようになります。これは、Web アプリケーションに対する個人ユーザの認証とは異なります。Web ゲートウェイの InterSystems IRIS のユーザ名およびパスワード、Windows ネットワークの資格情報、および UNIX® Kerberos キー・テーブルのいずれも、通常のユーザが使用しないようにする必要があります。

11.1 Web ゲートウェイの接続セキュリティの構成

Web ゲートウェイの接続セキュリティを構成するには、常に [Web ゲートウェイ管理ページ](#)を使用します。関連するオプションは、[\[構成\]](#)→[\[サーバ接続\]](#)→[\[接続セキュリティ\]](#) セクションにあり、ここでは次のような設定が用意されています。

- ・ [\[接続セキュリティレベル\]](#): 選択肢は以下のとおりです。
 - [\[パスワード\]](#)

- [Kerberos]
- [Kerberosパケット整合性]
- [Kerberos暗号化]
- SSL/TLS
- ・ [ユーザ名]
- ・ [パスワード]
- ・ [製品]
- ・ [サービス・プリンシパル名]
- ・ [キーテーブル]

11.2 最小の接続セキュリティ (非推奨)

最小の接続セキュリティを使用するには、**[接続セキュリティレベル]** を **[パスワード]** に設定し、**[ユーザ名]** フィールドおよび **[パスワード]** フィールドを空欄のままにします。

このモードでは、Web ゲートウェイと InterSystems IRIS 間の接続に対して最小レベルのセキュリティが適用されます。

この処理モードの場合、Web ゲートウェイ・サービス (%Service_WebGateway) と、そのサービスが機能するユーザ名 (CSPSystem など) が、任意の形式の認証を要求していないかどうかを確認します。

11.3 単純なユーザ名/パスワード認証

ユーザ名/パスワード認証は、Web ゲートウェイと InterSystems IRIS 間に適用できる最も簡単な認証形式です。インスタンスのインストール・プロセスでは、インスタンスの Web ゲートウェイ・アクセスを認証する CSPSystem ユーザが作成されます。このユーザ (CSPSystem またはその他のユーザ) には、有効期限を設定する必要はありません。つまり、Expiration Date プロパティの値は 0 になります。

どのような場合でも、Web ゲートウェイに使用する既定のユーザ名とパスワードは以下のようになります。

```
Username: CSPSystem  
Password: SYS
```

Web ゲートウェイ管理ページでインスタンスに単純なユーザ名/パスワード認証を構成するには、**[接続セキュリティレベル]** を [] に設定し、**[ユーザ名]** と **[パスワード]** に値を指定して、サーバ・アクセス・プロファイルを変更します。

InterSystems IRIS アプリケーション・サーバでは、Web ゲートウェイへのアクセス権を与えるユーザ・アカウントの認証情報を必要に応じて変更できます。ただし、Web ゲートウェイの対応するサーバ・アクセス・プロファイルも、これらの新しい認証情報で更新する必要があります。そうしないと、Web ゲートウェイはそのアプリケーション・サーバへの新しい接続を確立できません。パスワードは { で始まり、} で終わることはできません。Web ゲートウェイはこれらの中括弧内の文字列を **パスワード取得コマンド** として解釈しようとするためです。

パスワードは、InterSystems IRIS で認証するプレーン・テキストとしてネットワーク上に送信する必要があるため、セキュリティの面で脆弱な形式の認証であることに注意してください。簡単に実行可能なネットワーク・スニффイングによって、これらのパスワードが盗まれる可能性があります。この構成オプションで使用するパスワードは、以下のガイドラインに従って、Web ゲートウェイの **構成ファイル** で保持する必要があります。

UNIX®、Linux、および macOS では、CSP.ini ファイル内のパスワードは base64 ハッシュとして格納されます。Windows では、パスワードは、Microsoft のデータ保護 API (DPAPI) で提供される機能を使用して Web ゲートウェイ構成ファイルに暗号化されます。このパスワードの暗号化は、Web ゲートウェイ管理の [デフォルトパラメータ] ページで処理されます。

注釈 通常の Windows のユーザ・アカウントには Administrators グループのメンバシップが付与されていることがあるため、Windows ではこのパスワード暗号化が使用されています。ただし、プロダクション・システムではこの方法をお勧めしません。パスワードを暗号化すると、Windows のすべてのインストールに、より高いレベルの保護が提供されます。

11.3.1 外部から導入されるパスワード

Web ゲートウェイ管理ページのコンテキスト以外でパスワードを扱うが必要になる場合もあります。例えば、Web ゲートウェイ構成がカスタム構成スクリプトによって設定されている場合などです。この場合、パスワードをプレーン・テキストとして保存しておく、Web ゲートウェイを初めて起動したときにこのパスワードが暗号化されます。このメソッドを使用する場合、ゲートウェイ管理フォームへのパスワードを '1' または 'PBKDF2' で始めることはできず、IRIS へのパスワードを ']]]' で始めることはできません。これらの文字で始まるパスワードが必要な場合は、代わりに OS のコマンド・プロンプトで CSPpwd ユーティリティを使用します。このユーティリティは、Web ゲートウェイ構成ファイルに保持されているパスワードをエンコードします。一般的なフォームは次のとおりです。

CSPpwd <path to the CSPx.so|dll library> <context> <clear text password>

説明:

- ・ context = 0: Web ゲートウェイ管理ページへのパスワード
- ・ context = 2: Cache/IRIS サーバへのパスワード

エンコードされたパスワードが標準出力に書き込まれます。

例 (Windows):

```
CSPpwd C:\Inetpub\CSPGateway\CSPx.dll 0 MyGatewayManagementPassword
CSPpwd C:\Inetpub\CSPGateway\CSPx.dll 2 MyIRIServerPassword
```

例 (UNIX®):

```
CSPpwd /opt/webgateway/bin/CSPx.so 0 MyGatewayManagementPassword
CSPpwd /opt/webgateway/bin/CSPx.so 2 MyIRIServerPassword
```

11.3.2 別のコンピュータで暗号化されたパスワード

Windows では、Web サーバはユーザ・ストアではなくマシン・ストアを使用します。そのため、DPAPI パスワード暗号化はマシン固有です。他のコンピュータで暗号化された Web ゲートウェイ・パスワードを解読することはできません。つまり、クラスタ環境内のマシンが CSP.ini ファイル内の情報を共有することはできません。

この問題に対して考えられるソリューションを以下に示します。

- ・ クラスタ外のマシンを Web サーバとして使用する。
- ・ フェイルオーバーするたびに、Web ゲートウェイで同じパスワードを再設定する。
- ・ そのクラスタに属していないディスクに Web ゲートウェイ構成ファイルのコピーをそれぞれ独自に持つように、クラスタに属する各コンピュータを構成する。InterSystems IRIS が、Web ゲートウェイの DLL をホストするディレクトリにファイルを保持する。個々のコンピュータそれぞれにパスワードを保存し、暗号化したうえで、ノードをクラスタに導入する。

例えば、各マシンの Disk C がクラスタに属さず、InterSystems IRIS が Disk S にインストールされている場合は、次のようになります。

CLUNODE-1 : CLUNODE-1 で暗号化したパスワード XXX を記述した CSP.ini のコピー

CLUNODE-2 : CLUNODE-2 で暗号化したパスワード XXX を記述した CSP.ini のコピー

- Web ゲートウェイを起動してパスワードを追加する前に、以下の指示文を構成ファイルに手動で追加することによって、パスワードの暗号化を無効にする。

```
[SYSTEM]
DPAPI=Disabled
```

11.3.3 プログラムによるパスワードの取得 (UNIX®/Linux/macOS)

UNIX®/Linux/macOS システムでは、パスワードそのものをプレーン・テキストで指定するのではなく、安全なストレージ・ソリューション (ボールド・アプリケーションやマウントされたシークレット・ファイルなど) からパスワードを取得するオペレーティング・システム・コマンドを実行するよう Web ゲートウェイに指示できます。それには、[Web ゲートウェイ管理ページ](#)を使用して、インスタンスの[サーバ・アクセス・プロファイル](#)の [パスワード] フィールドで、中括弧 ({}) で囲んでコマンドを指定します。インスタンスに対応する [CSP.ini](#) ファイルのセクション内で、対応する CSP.ini パラメータ ([password](#)) を直接設定することもできます。例えば、[CSP.ini](#) ファイルには以下の行が含まれる場合があります。

```
Password={sh /tmp/PWretrieve.sh}
```

文字列は、プレーン・テキストのパスワードのように、[CSP.ini](#) ファイル内に base64 ハッシュ値として格納されます。

Web ゲートウェイは、[Web ゲートウェイ管理ページ](#)を使用してサーバ・アクセス・プロファイルを保存するとき、または [CSP.ini](#) ファイルの [SYSTEM] セクションで [RELOAD=1](#) が検出されたときにコマンドを実行します。コマンドの出力は、インスタンスのパスワードとしてメモリに格納されます。指定するコマンドは、追加アクションなしでパスワードを取得できるものである必要があります。安全なストレージ・ソリューションのコマンド行インタフェースをシステムで完全に構成し、必要な認証情報が環境変数または構成ファイルで利用できる必要があります。

注釈 この方法を使用して、SSL/TLS 秘密鍵パスワードを[サーバ・アクセス・プロファイル内](#)または [CSP.ini ファイル内](#)でプログラムによって取得することもできます。

11.4 Kerberos ベース認証とデータ保護

Kerberos ベース認証とデータ保護を使用するために、[接続セキュリティレベル] パラメータによって 3 種類の認証レベル (およびデータ保護) が提供されています。

1. Kerberos : 接続に対する初期の認証のみを提供します。
2. Kerberos パケット整合性 : 初期の認証を提供し、データ・パケットの整合性を保証します。
3. Kerberos 暗号化 : 最高レベルのセキュリティです。初期の認証、データ・パケットの整合性の保証、および送信されるすべてのメッセージの暗号化を行います。

11.4.1 Kerberos ライブラリ

Kerberos ベースのいずれかのモードを使用する場合、Web ゲートウェイは次の InterSystems Kerberos クライアント・ライブラリをロードする必要があります。

- Windows DLL : [irisconnect.dll](#)
- UNIX® 共有オブジェクト : [irisconnect.so](#)

オペレーティング・システムの PATH 環境変数に指定した場所、または Web ゲートウェイのインストール場所を基準にした以下のいずれかの場所に、適切なライブラリをインストールします。

- ・ . (Web ゲートウェイのローカル側)
- ・ ./bin
- ・ ../bin
- ・ ../../bin

Web ゲートウェイは、ライブラリが初めて必要になったときにライブラリのロードを試みます。成功すると、[Web ゲートウェイ・イベント・ログ](#)に次のステータス・メッセージが書き込まれます。

```
Web Gateway Initialization The IRISCONNECT library is loaded - Version: 5.3.0.175.0.
```

(このライブラリは、Web ゲートウェイと InterSystems IRIS の間のオプションの Kerberos ベースのセキュリティで使用されます。)

Web ゲートウェイが IRISCONNECT ライブラリを見つけられない場合や、リンクできない場合は、失敗を示す適切な説明やエラー・メッセージが [Web ゲートウェイ・イベント・ログ](#)に書き込まれます。

Web ゲートウェイと InterSystems IRIS 間の通信が Kerberos で保護されている場合は、Web ゲートウェイが Kerberos クライアントになります。

Kerberos を使用するように Web ゲートウェイを構成する手順は、[Windows](#) のセクションに記載されています。

11.4.1.1 SSL/TLS を使用する場合のライブラリ・パスのオーバーライド

既定の Web ゲートウェイでは、そのホーム・ディレクトリ (つまり、Web ゲートウェイ・バイナリを保持するディレクトリ) に依存セキュリティ・ライブラリ (共有オブジェクト) がインストールされることを想定しています。

Web ゲートウェイと InterSystems IRIS 間で SSL/TLS 接続を使用する場合、これらのライブラリとして IRISCONNECT ライブラリおよび SSL/TLS ライブラリ (UNIX® の場合: libssl.so と libcrypto.so、Windows の場合: libcrypto-1_1-x64.dll と libssl-1_1-x64.dll) があります。

Web ゲートウェイおよび IRISCONNECT ライブラリ (Web サーバの処理領域にロードされます) が SSL/TLS ライブラリのコピーをロードする場合、ホスト Web サーバによって以前にロードされた同じライブラリの異なるバージョン間で競合が発生します。SSL/TLS ライブラリの 1 つのコピーだけが Web サーバの処理領域にロードされるようにするには、Web ゲートウェイが IRISCONNECT ライブラリに対して、ホスト Web サーバで使用されているものと同じ場所から SSL/TLS ライブラリをロードするように指示する必要があります。

Web ゲートウェイ管理の **[デフォルトパラメータ]** ページでは、OpenSSL ライブラリの代替セットを使用するためのパラメータ **[SSL/TLS ライブラリ・パス]** が提供されます。例えば、以下のとおりです。

```
SSL/TLS Library Path = /usr/bin/
```

重要

OpenSSL の使用を許可しない Apache インストールを作成することも、OpenSSL を無効にするように Apache を構成することも可能です。この状況では、**[SSL/TLS ライブラリ・パス]** が別の場所に設定されない限り、Web ゲートウェイは付属していたライブラリをロードします。

ライブラリ・バージョンが Apache と Web ゲートウェイで一致しない場合、Web ゲートウェイと InterSystems IRIS インスタンス間の TLS 接続は失敗する可能性があります。接続を試みたときに TLS エラーが発生したり、OpenSSL 関数を呼び出すときに Web ゲートウェイが SIGSEGV でクラッシュする可能性があります。

11.4.2 Windows

Windows の場合、Kerberos キー・テーブルは実装されません。したがって、ホスト・サービスが指定のアカウントで開始したときに取得されるネットワーク資格情報、またはホスト・サービスをシステム・ログオン・セッションで (LOCAL SYSTEM として) 実行するときに Trusted Computing Base (TCB) から取得されるネットワーク資格情報が認証で使用されます。

Windows のドメイン・アカウントは、パスワードから派生した永続キーを使用して、ローカル・マシン用の Kerberos Ticket Granting Ticket (TGT) およびサービス・チケットを取得します。ローカル・マシンにも、ドメイン・コントローラの Key Distribution Centre (KDC) コンポーネントとの間で共有される永続 Kerberos キーが必要です。このキーを使用して、InterSystems IRIS などの別の Kerberos プリンシパルに対して認証を行うための TGT やサービス・チケットを取得できます。

実際に、Windows ベースの Web サーバ内で実行する Web ゲートウェイは、ネットワーク・サービス・ログオン・セッションまたはシステム・ログオン・セッションによって動作しています。使用するアカウントには、する権利を
割り当てる必要があります。

組み込みネットワーク・サービス・ログオン・セッションは、マシンの資格情報にアクセスできます。このセッションは、その他のマシンに対して認証を行うためにネットワーク資格情報を必要とするサービス用に設計されています。ただし、ネットワーク・サービス・ログオン・セッションは常に存在するわけではありません。InterSystems IRIS に対して Web ゲートウェイを認証する目的でシステム・ログオン・セッションを使用することもできます。

IIS インストール (特に ISAPI 拡張) の場合、ネットワーク・サービス・ログオン・セッションを使用して両方のデータベース (ローカルおよびリモート) とリモート・コンピュータにアクセスするのが望ましい方法です。

11.4.2.1 Kerberos のための Windows Web ゲートウェイ構成

- ・ **[サービス・プリンシパル名]** を、Web ゲートウェイの接続先であるターゲットの InterSystems IRIS サーバ名に設定します。
- ・ **[ユーザ名]**、**[パスワード]**、および **[キーテーブル]** の各フィールドは、空欄にしておきます。
- ・ クライアント・プリンシパル名 (クライアント・ユーザ名) は、Web ゲートウェイ・ホストの名前です。これは、Web ゲートウェイ・ホストのネットワーク・サービス・セッションを表す、次の Kerberos 名です。<computer_name>\$
- ・ このプリンシパルに InterSystems IRIS サーバで必要な特権を割り当て、Web ゲートウェイのサービスが動作するようにします。

11.4.3 Kerberos のための UNIX® Web ゲートウェイ構成

これらのオペレーティング・システムは、Kerberos キー・テーブルをサポートします。

11.4.3.1 Kerberos のための UNIX® Web ゲートウェイ構成

概念的には、これらのシステムの Web ゲートウェイ構成の方が簡単です。

- ・ **[サービス・プリンシパル名]** を、Web ゲートウェイの接続先であるターゲットの InterSystems IRIS サーバ名に設定します。
- ・ **[キーテーブル]** フィールドに、キー・テーブル・ファイル名 (フル・パス指定) を入力します。
- ・ **[ユーザ名]** フィールドを、キー・テーブル・ファイルの適切なキー名に設定します。
- ・ **[パスワード]** フィールドは空欄にします。
- ・ クライアント・プリンシパル名 (クライアント・ユーザ名) は、Web ゲートウェイ・ホストの名前です。これは Kerberos キー・テーブルでキーを識別するために使用する名前です。このプリンシパルに InterSystems IRIS サーバで必要な特権を割り当て、Web ゲートウェイのサービスが動作するようにします。

11.5 SSL/TLS ベース認証とデータの保護

SSL/TLS プロトコルを使用して、Web ゲートウェイと InterSystems IRIS 間の通信を保護することもできます。

このモードでは、ホストに構成された SSL/TLS 転送により InterSystems IRIS への接続が保護されます。**[SSL/TLS 構成]** フィールドを、ターゲット・サーバに適切な値に設定する必要があります。**[サービスプリンシパル名]** および **[キーテーブル]** フィールドは無関係なので、空白にしておく必要があります。

InterSystems IRIS システムの SSL/TLS クライアント構成の作成に関する詳細は、[“TLS を使用して InterSystems IRIS に接続するための Web ゲートウェイの構成”](#) を参照してください。

11.5.1 相互 TLS

相互 TLS は、パスワード認証の代替手段として Web ゲートウェイと InterSystems IRIS の間で適用できる認証形式です。詳細は、[“相互 TLS \(mTLS\)”](#) を参照してください。

12

Web ゲートウェイの管理と監視

ここでは、Web ゲートウェイ管理ページから、InterSystems Web ゲートウェイを管理および監視する方法について説明します。

12.1 システム・ステータスの確認

[システムステータス] オプションには、すべてのアクティブな接続のステータスが表示されます。この機能を使用するには、システム管理者である必要があります。以下の各表では、列見出しをクリックするとその列を基準にして並べ替えることができます。

12.1.1 InterSystems IRIS への接続

最初のステータス・テーブル (InterSystems IRIS への接続) は、InterSystems IRIS® への接続に関する情報を表示します。

アイテム	機能
[接続番号]	Web ゲートウェイが接続に割り当てる番号。可能な接続数は InterSystems IRIS ライセンスで決まります。
[ゲートウェイPID]	接続の Web ゲートウェイ (またはホスト Web サーバ) プロセス ID
[サーバ名]	接続先 InterSystems IRIS システムの名前。ミラー・メンバは、ミラーメンバ名が付加された現在の構成名を示します。
InterSystems IRIS PID	InterSystems IRIS サーバのプロセス ID。
[状態]	以下のように、InterSystems IRIS システムとの間で情報の送受信が行われているかどうかを示します。Free - 情報は送信されておらず、接続は次の要求を処理する準備をしています。In Use - 情報は接続を介して送信中です。Private - 接続はステート認識 (保持モード 1) であり、一般的な用途には解放されていません。Server - 接続は InterSystems IRIS サーバによって使用されています。
[アイドル時間]/[タイムアウト]	その接続に適用されるタイムアウトに対して、接続がアイドル状態になっている時間を示します。タイムアウトは、状態なしプール内の接続の場合は '無使用タイムアウト' で、'プライベート' としてマークされた接続の場合は 'アプリケーション・タイムアウト' です (ステート認識)。
[アクティビティ]	この接続で処理したトランザクション (ヒット) 数。

アイテム	機能
[中断]	ステータスが '使用中' の接続では、[中断] ボタンは、対応する InterSystems IRIS プロセスを中断し、次の Web 要求を受け入れ可能な状態に戻そうとします。
[閉じる]	使用可能な場合、これを選択すると接続を強制的に閉じることができます。“ 手動による接続の切断 ”を参照してください。

12.1.2 InterSystems IRIS サーバ・テーブル

2 番目のステータス・テーブル (InterSystems IRIS サーバ) は、InterSystems IRIS サーバに関する情報を表示します。

アイテム	機能
[サーバ番号]	Web ゲートウェイがサーバに割り当てる番号。
[サーバ名]	接続先 InterSystems IRIS システムの名前。
[ミラー・メンバ]	ミラー認識構成の場合、ミラー・メンバの名前。
[ミラー・ステータス]	ミラー認識構成の場合、ミラー構成の名前とサーバのミラー・ステータス。メンバ・タイプと、フェイルオーバーまたは非同期のどちらであるかが表示され、プライマリには 'プライマリ' というラベルが付けられます。
[総接続]	InterSystems IRIS システムへの接続数。
[使用中の接続]	現在使用中の (Web 要求を実際に処理している) 接続数。
[プライベート接続]	ステート認識セッション (保持モード 1) として現在使用中の接続数。
[総アクティビティ]	InterSystems IRIS システムが処理したトランザクション (ヒット) 数。
[キューイングされた要求]	InterSystems IRIS システムへの接続が空くのを待機して、キューに保持されている Web 要求の数。キューイングされた要求は、高いパフォーマンスを維持するために InterSystems IRIS ライセンスを増加する必要があるかどうかを判断する目安になります。
[閉じる]	この InterSystems IRIS サーバ上のすべての接続を閉じます。“ 手動による接続の切断 ”を参照してください。

12.1.3 アプリケーション・パス・テーブル

3 番目のステータス・テーブルはアプリケーション・パスの情報を表示します。

アイテム	機能
[パス番号]	Web ゲートウェイがアプリケーション・パスに割り当てる番号。
[パス]	アプリケーション・パス。
[サーバ番号]	Web ゲートウェイが InterSystems IRIS サーバに割り当てる番号。
[サーバ名]	接続先 InterSystems IRIS システムの名前。
[アクティビティ]	最後のゲートウェイ以降にこのサーバによって処理されたこのパスの要求数。
[状態]	このサーバの状態。Disabled、Enabled または Offline のいずれかです。このセットの現在のマスタ・サーバ (またはプライマリ・サーバ) もこの列で示されます。
[アクション]	サーバが Offline とマークされている場合、この列には管理者が再度 Online/Enabled とマークできるボタンが含まれます。

12.1.4 Web ゲートウェイ・キャッシュ・テーブル

4 番目のテーブルに、Web ゲートウェイの応答キャッシュに保持されるフォームを示します。

アイテム	機能
[キャッシュされたフォーム]	キャッシュされたフォームの名前 (パスを含む)。
[キャッシュされたデータ (バイト単位)]	ゲートウェイに保持されているキャッシュされたフォーム・データの量 (バイト単位)。
[使用中のキャッシュ・ブロック]	使用中のキャッシュ・メモリ・ブロック数の合計
[キャッシュ・ファイル]	(Web サーバ・ホスト上の) 永続ストレージを使用してファイルをキャッシュする場合、物理ファイルの名前。
[キャッシュ・フォーム・アクティビティ]	このフォームがキャッシュに要求された回数の合計数。
[クリア]	このフォームをキャッシュからクリアします。“ キャッシュのクリア ”を参照してください。

12.1.5 手動による接続の切断

CSP 接続がまだアクティブなときに InterSystems IRIS システムが停止した場合、以下のいずれかの状況になるまで、CSP は引き続きシステムに接続しようとしています。

- ・ システムへの再接続が成功する。
- ・ CSP が停止する。
- ・ 手動で接続を閉じる。

長時間にわたる InterSystems IRIS システムのダウンタイムを予定している場合は、接続を閉じることができます。**[システムステータス]** ページの **[閉じる]** ボタンを使用して、セッションを手動で閉じることができます。

InterSystems IRIS システムの停止中に接続を閉じることができます。

12.1.6 キャッシュのクリア

Web アプリケーションの開発プロセスなどの特定の状況では、Web ゲートウェイ・キャッシュをクリアすることが必要になる場合があります。これを行うには、以下を実行します。

1. 管理ポータルで **[システム管理]**→**[構成]**→**[ウェブゲートウェイ管理]** に移動して、**[システムステータス]** を選択します。
2. **[システムステータス]** ページには、いくつかのテーブルがあります。キャッシュをクリアするには、**[キャッシュされたフォーム]** テーブルで、**[合計]** 行 (下の行) の **[クリア]** 列 (右端列) にあるボタンを選択します。**[システムステータス]** ページに **[キャッシュされたフォーム]** テーブルが表示されない場合、現在キャッシュされたコンテンツはありません。これは、キャッシュが最近クリアされ、それ以降何もキャッシュされていないためです。

このアクションでは、Web ゲートウェイのキャッシュされたコンテンツをすべてクリアし、コンテンツが新しくキャッシュされるまで、**[キャッシュされたフォーム]** テーブルをページから削除します。

12.2 サーバ接続のテスト

[サーバ接続のテスト] オプションは、InterSystems IRIS システムへの Web ゲートウェイの接続をテストする際に役立ちます。この機能を使用するには、システム管理者である必要があります。

CSP 接続をテストするには、以下の手順を実行します。

1. Web ゲートウェイ管理ページから、[サーバ接続のテスト] を選択します。
2. 表示されたリストで、目的の InterSystems IRIS システムを選択します。
3. [接続] を選択します。

選択した内容とサーバ接続の状態に応じて、以下のいずれかの結果が表示されます。

結果	意味
CSP テスト・フォーム	Web ゲートウェイは正常に動作しており、InterSystems IRIS に接続できます。このフォームは、ターゲットの InterSystems IRIS サーバ によって返される基本パラメータ（バージョンおよびプロセス ID）を示します。
サーバアベイラビリティエラー	InterSystems IRIS にアクセスできない場合、このエラーが発生します。追加のエラー・メッセージがない場合は、InterSystems IRIS システムが稼動しているかどうかを確認します。また、 Web ゲートウェイ・イベント・ログ で特定の接続エラー・メッセージを確認します。

どのような場合でも、エラー状況が返された場合は、[Web ゲートウェイ・イベント・ログ](#)で追加および具体的なエラー情報を確認します。必要に応じ、[ログ・レベル](#)を上げて、追加の診断情報を取得することを検討してください。

12.3 イベント・ログの表示

Web ゲートウェイ管理ページのメイン・メニューから [\[イベントログを参照\]](#) オプションを使用して、[Web ゲートウェイ・イベント・ログ](#)の内容を確認します。[\[イベントログレベル\]](#) パラメータは、どのような情報がログに記録されるかを決定します。

Web ゲートウェイは、既定では `CSP.log` という名前のログ・ファイルにイベント・ログを格納します。便宜のために、Web ゲートウェイの [\[Web ゲートウェイについて\]](#) 管理ページには、このファイルの場所が示されています。アクティブなログ・ファイルが [\[イベントログ切り替えサイズ\]](#) で指定した容量に達すると、このファイルは `filename.old` (`filename` は元の完全なファイル名) にコピーされます。[\[イベントログ切り替えサイズ\]](#) が空白の場合（既定）、このファイルは手動でクリアされるまで増大します。すべてのログを、日付と時刻を含む名前を付けて保存するには、[\[デフォルトパラメータ\]](#) ページの [\[すべてのログ・ファイルを残す\]](#) にチェックを付けます。各ログ・エントリには、そのログ・エントリが作成された日付、時刻、および追加情報を記録したヘッダ・レコードが表示されています。

ログ・エントリは、InterSystems IRIS インスタンスが [構造化ログ](#)機能で使用しているものと同じ、機械で判読可能な形式に従っています。つまり、Web ゲートウェイ・イベント・ログの分析には、InterSystems IRIS インスタンスのログの分析に使用しているものと同じサードパーティのツールを使用できます。効率性を高めるため、Web ゲートウェイ・イベント・ログでは、`when`、`level`、`event`、`pid`、`text` といった、他の構造化ログで使用されているものと同じフィールド名を使用しています。[\[イベントログを表示\]](#) ページでは、`text` および `details` フィールドは、フィールド名なしで表示されています。ただし、`CSP.log` ファイルで提供されるログ・エントリでは、名前と値のペア (NVP) 形式に完全に準拠しています。

以下に、`CSP.log` に表示される Web ゲートウェイ・イベント・ログのエントリの例を示します。`CSP.log` では各エントリが 1 行で記述されます。ただし、この例では、読みやすいように、複数の行に分割されています（\ 文字でマークされています）。


```
local-time="Thu Jul 21 11:39:20 2022" ¥  
wg-build="RT 2202.1825 (win32/apapi:sv=2.4.52/apr=1.7.0/apu=1.6.1/mpm=WinNT)" ¥  
wg-log-level=0 when="2022-07-21 15:39:20.831" level=WARNING event=WebGateway.SessionOpen ¥  
pid=17216 thread-id=2072 text="Warning" ¥  
details="A Connection between the Web Gateway and InterSystems IRIS has been found to be ¥  
closed (possibly as a result of an intermediary, such as a firewall, timing-out the TCP session)"
```

イベント・ログから現在のすべてのエントリを消去するには、**[ログをクリア]** を選択します。

ログは日付/時刻の昇順(既定)または降順で表示できます。フォームの右上隅にあるリンクを選択すると表示順が逆になります。このリンクには、2 つのモードを切り替える役割があります。

最後に、ほとんどのブラウザでは、1MB ほどを超えるログ・データを 1 つのフォームで表示することはできません。このため、返されるログ・データの容量が 1MB に達すると Web ゲートウェイは表示を終了し、次ページのデータを表示するようユーザに指示します(フォームの左下隅にある**[次]**リンクを参照)。また、フォームの右下隅にある**[トップ]**ハイパーリンクを使用すると、一連のフォームのうち最初のフォームにすばやく戻ることができます。

12.4 HTTP トレース機能の使用法

HTTP トレース機能には**[HTTP トレースを表示]** オプションからアクセスします。

トレース・ウィンドウは 2 つの主なフレームから構成されています。左側のフレームには、Web ゲートウェイで処理された HTTP 要求のリストがあり、時刻と (Web ゲートウェイから割り当てられた) 一意の要求 ID が示されます。それぞれの要求を選択すると、要求および応答データが右側のフレームに表示されます。リンクにより、要求メッセージと応答メッセージの表示を簡単に切り替えることができます。

注釈 Web ゲートウェイが報告する HTTP 要求ヘッダは再構成されているという点に注意してください。これは、ホスト Web サーバが常に、要求ヘッダを解析する責任を負っているためです。Web ゲートウェイは Web サーバが提供する CGI 環境変数から完全なヘッダを再構成します。ただし、要求が NSD コンポーネントを介して直接(つまり、実際には Web サーバをバイパスして)渡される場合、記録された要求ヘッダは、クライアントからディスパッチされた場合とバイト単位で同じになります。

13

Web ゲートウェイによって渡される CGI 環境変数

CGI 環境変数は、クライアントの HTTP 要求ヘッダ、および Web サーバが動作している環境の両方から派生します。[Web ゲートウェイ](#)は、要求が発生するたびに InterSystems IRIS® に共通の環境変数を転送します。アプリケーションで追加の環境変数が必要な場合は、Web ゲートウェイ構成の **[アプリケーションアクセス]** セクションにある **[追加の CGI 環境変数]** 設定で明示的に要求する必要があります。InterSystems IRIS 管理ポータルで **[システム管理]**→**[構成]**→**[ウェブゲートウェイ管理]** に移動して、**[アプリケーションアクセス]** を選択します。

以下のテーブルは、転送される環境変数、および各変数の簡単な説明を示しています。他のドキュメントは、標準の Web テキスト・ブックから入手できます。

環境変数	値
AUTH_PASSWORD	クライアントの認証ダイアログに入力された値です。この値は、基本認証が使用される場合にのみ利用できます。
AUTH_TYPE	保護されているスクリプトにユーザがアクセスしようとする際に、サーバがそのユーザの検証に使用する認証方法です。
CONTENT_TYPE	HTTP POST や PUT などの情報が添付されている要求の場合、これがデータのコンテンツ・タイプとなります。
GATEWAY_INTERFACE	このサーバが適合する CGI 仕様のリビジョンです。形式は CGI/revision です。
HTTP_ACCEPT	受け入れられる形式 (MIME タイプ) のリストを含む受け入れ要求ヘッダの値です。image/gif、image/x-xbitmap、image/jpeg、image/pjpeg、application/vnd.ms-excel などがあります。HTTP_ACCEPT 変数のフィールドの各値は、コンマ (,) で区切って並べます。
HTTP_ACCEPT_CHARSET	クライアントが受け入れる、文字エンコードのコンマ区切りリストです。
HTTP_ACCEPT_LANGUAGE	コンテンツの表示に使用する言語 (en-us など) を記述する文字列です。
HTTP_AUTHORIZATION	クライアントから送信される、BASE64 でエンコードされたユーザ名、パスワード、スキーマ、およびレルムが含まれます。
HTTP_COOKIE	クライアントの Cookie のコンテンツを保持します。
HTTP_REFERER	HTML の <A> タグを使用して現在のページを要求の参照先とするページの URL を含む文字列を保持します。この URL はユーザがブラウザのアドレス・バーに入力したもので、既定のドキュメント名は含まれません。ページがリダイレクトされる場合、HTTP_REFERER は空です。

環境変数	値
HTTP_SOAPACTION	SOAPAction HTTP 要求ヘッダ・フィールドを使用して、SOAP HTTP 要求の目的を示すことができます。値は目的を特定する URI です。SOAP はこの URI の形式や具体性、および解決可能性について、何も制限していません。HTTP クライアントは、SOAP HTTP 要求の発行時にこのヘッダ・フィールドを使用する必要があります。
HTTP_USER_AGENT	クライアントが要求の送信に使用しているブラウザです。一般的な形式は software/version library/version です。
HTTPS	On または Off に設定します (数値ではなく、単語で設定します)。セキュア・サーバを通して (SSL/TLS を使用して) スクリプトが呼び出される場合は、on に設定します。
PATH_TRANSLATED	仮想から物理へのマッピングがパスに適用された、PATH_INFO の変換後のバージョンです。
REMOTE_ADDR	要求を行っているリモート・ホストの IP アドレスです。
REMOTE_HOST	要求を行っているホスト名です。サーバにこの情報がない場合、REMOTE_ADDR が設定され、このパラメータは設定されません。
REMOTE_IDENT	HTTP サーバが RFC 931 による識別をサポートしている場合、この変数は、サーバから取得したリモート・ユーザ名に設定されます。
REMOTE_USER	クライアントによって送信された認証ヘッダから取得されるユーザの名前です。
REQUEST_METHOD	要求の生成に使用したメソッドです。HTTP の場合は、GET、HEAD、POST などです。
SERVER_NAME	サーバのホスト名、DNS エイリアス、または自己参照 URL に示される IP アドレスです。
SERVER_PORT	要求の送信先のポート番号です。例 : 80
SERVER_PORT_SECURE	0 または 1 に設定されます。要求が Web サーバの安全なポートで処理される場合は 1 に設定され、それ以外の場合は 0 に設定されます。
SERVER_PROTOCOL	その要求の送信で使用された情報プロトコルの名前とリビジョンです。形式は protocol/revision です。
SERVER_SOFTWARE	要求に応答する Web サーバ・ソフトウェアの名前とバージョンです。形式は name/version です。

14

Web ゲートウェイにより返される HTTP 応答ヘッダ

InterSystems IRIS® 内の [Web アプリケーション](#) (REST ベースのアプリケーションを含む) は、通常、完全な HTTP 応答ヘッダを構築する役割があることが前提となっています。パフォーマンス上の理由で、[Web ゲートウェイ](#)は既定で、Web サーバを経由して以下のコンテンツと共に応答ヘッダをクライアントに直接ストリーム転送しています。この処理モードを、非解析ヘッダ (NPH) 方式といいます。Web ゲートウェイは、ホスト Web サーバに用意された専用の API 機能を使用して応答ヘッダを渡すことにより、ホスト Web サーバが応答ヘッダを制御できないようにしています。応答ヘッダの指示文の読み取りおよび解釈を行う必要があるのは、Web サーバではなくクライアントであることが前提です。

ただし、CSP エンジンで生成したヘッダの指示文に示される Web サーバベースの追加機能呼び出すために Web サーバが応答ヘッダを解釈する必要がある場合、この前提は成り立ちません。例えば、応答を追加処理するために、出力フィルタを呼び出す場合です (圧縮や暗号化のユーティリティなど)。このような出力フィルタは、通常、非解析ヘッダ・モードの処理に従って返された CSP コンテンツに対しては機能しません。

クライアントに応答ヘッダを直接ストリーミングするのではなく、ホスト Web サーバ経由で応答ヘッダを明示的に渡すように、Web ゲートウェイに指示する機能が存在します。

この機能を使用するには、CSP ヘッダ指示文 `CSP-nph: false` を設定します。

この指示文は、`OnPreHTTP()` メソッドで設定する必要があります。以下に例を示します。

```
<script language=objectscript method=OnPreHTTP arguments=""
returntype=%Boolean>
Do %response.SetHeader("CSP-nph", "false")
Quit 1 </script>
```

`false` に設定すると (Web ゲートウェイの既定の設定は `true`)、`CSP-nph` 指示文によって、[CSP エンジン](#)から返された応答ヘッダによって、応答の種類がホスト Web サーバに適切に通知されるようになります。その結果、必要に応じて以降の処理が可能になります。これが解析ヘッダ・モードです。

Web ゲートウェイが解析ヘッダ・モードで動作している場合、ホスト Web サーバが応答ヘッダを解釈し、場合によっては独自のヘッダ指示文を追加します。少なくとも、`Server` ヘッダを応答に追加します。次に例を示します。

`Server: Apache/2.0.48 (Win32)`

または以下ようになります。

`Server: Microsoft-IIS/5.1`

この機能は、Web サーバ API に対して直接機能する Web ゲートウェイ実装を使用する場合にのみ適用されます。つまり、CGI 以外はすべて適用されます。

Web ゲートウェイの CGI モジュールを使用するときにこの機能が必要な場合、NPH バージョンではない CSP CGI モジュールを使用するように Web サーバを構成する必要があります。例えば、`nph-CSPcgi` ではなく `CSPcgi` を使用します。CGI モジュール名で使用する接頭辞 `nph-` は、モジュールによって返される応答ヘッダの読み取りおよび解釈が不要であること（つまり、非解析ヘッダ・モードで動作していること）を Web サーバに伝える標準的な手段です。

これらのモジュールの解析バージョンと非解析バージョンとの本質的な違いは、HTTP 応答ステータス行の作成方法にあります。これはヘッダ・ブロックの先頭行です

解析ヘッダの場合、HTTP ステータス行の形式は以下のようになります。

`Status: <status_code>`

例:

`Status: 200 OK`

非解析ヘッダの場合、HTTP ステータス行の形式は以下のようになります。

`HTTP/1.1<status_code>`

例:

`HTTP/1.1 200 OK`

Web ゲートウェイで提供される CGI モジュールは、これらの相違を内部で自動的に処理します。[CSP エンジン](#) は常に、標準の HTTP ヘッダ・ブロック (2) を返します。

["アプリケーション・パスの追加"](#) の [Non-Parsed Headers] パラメータも参照してください。

15

CSP フォームの要求に対する応答の圧縮 (GZIP/ZLIB)

CSP エンジンで生成された応答をクライアントに配信する前に圧縮すると、クライアントに応答を転送する際に必要なネットワーク帯域幅が大幅に減少するので、効率的です。クライアントの観点からは、アプリケーションのパフォーマンスが向上します。比較的速度が遅い通信ネットワークでクライアントがモバイル・デバイスを使用してアプリケーションにアクセスする場合は、特に効果的です。もちろん、データを実際に圧縮するために Web サーバ・ホストで必要な CPU 時間が発生しますが、利点の対価としてのわずかな代償にすぎません。

特に、圧縮した応答データが役立つのは、大容量の応答データを生成する CSP ページの場合です。

Web サーバ環境で GZIP を実装するには、2 つの方法があります。

- ・ ここで説明する GZIP ライブラリへの Web ゲートウェイ固有のインタフェースを使用します。
- ・ GZIP 出力フィルタをホスト Web サーバのアド・オンとして使用します。

ほとんどの Web サーバは、データを圧縮するアド・オン機能を備えています。Windows/IIS には、gzip フィルタ (ISAPI フィルタとして実装) があります。Apache グループでは、アド・オン・モジュールとして実装される圧縮フィルタ `mod_deflate.c` (紛らわしい名前ですが、deflate 圧縮ではなく、gzip 圧縮を実装します) が用意されています。また、`mod_gzip.c` という Apache 用のサードパーティ・モジュールもあります。サードパーティの GZIP 製品は数多くあり、ほとんどの Web サーバにアド・オンとして利用できます。

Web ゲートウェイに直接圧縮ソリューションを実装する利点は以下のとおりです。

- ・ 設定と構成が容易です。
- ・ CSP ファイルの圧縮を制御する際に高い柔軟性が得られます。
- ・ 大容量のバッファでデータを圧縮関数に送ると、圧縮のパフォーマンスが高まります。Web ゲートウェイは、InterSystems IRIS からの応答の内容をかなり大きなチャンクで受信します。したがって、圧縮のパフォーマンスおよび効率が高まります。

チャンク転送エンコーディングが Web ゲートウェイ・レベルで有効化されており、Apache の `mod_deflate` 出力フィルタが同じリソースに対して有効化されていると、ブラウザでは応答の内容が表示できない場合があることがわかっています。

Web ゲートウェイは、データの圧縮の実装に、無償で利用できる GZIP (または `zlib`) ライブラリを使用します。使用する圧縮アルゴリズムについては、RFC (Request for Comments) の 1950 ~ 1952 に説明があります。

15.1 GZIP/ZLIB ライブラリ

GZIP/ZLIB ライブラリは、Jean-loup Gailly および Mark Adler (Copyright (C) 1995-2009) によって開発されました。Windows の InterSystems IRIS ディストリビューションには、このライブラリの事前構築済みバージョンが付属します。UNIX システムでは、Webゲートウェイは OS が提供する ZLIB のビルドを使用します。

Web ゲートウェイは、初めて応答の圧縮が要求されたときに、ZLIB ライブラリに動的にリンクします。以降は、Web ゲートウェイが停止するまで ZLIB ライブラリはロードされた状態になります。

Web ゲートウェイが必要に応じて ZLIB ライブラリをロードでき、すべての必要な機能を特定できれば、初期化メッセージが以下の形式でイベント・ログに書き込まれます (x.x.xx は、システム上のライブラリのバージョンを表します)。

```
Web Gateway Initialization
The ZLIB library is loaded - Version x.x.xx.
(This library is used for the optional GZIP compression facility)
```

Web ゲートウェイが ZLIB ライブラリを見つけられないか、リンクできない場合、以前と同様に動作し、ページは圧縮されずに返されます。この場合は、失敗を示すメッセージがイベント・ログに書き込まれます。

15.2 GZIP/ZLIB ライブラリの使用法

Web ゲートウェイは、ZLIB ライブラリを使用して応答データを圧縮する場合に 2 つの処理モード (1 および 2) を実装します。

1. このモードでは、Web ゲートウェイは InterSystems IRIS から受信したすべてのデータを圧縮機能にストリーム転送します。すべてのデータが処理されると、圧縮データが Web ゲートウェイに返されます。この時点でデータはクライアントに転送されます。

このモードでは、待ち時間はわずかに長くなりますが、可能な限り最適な圧縮が行われます。もちろん、形式が大きくなるほど待ち時間は長くなります。

2. このモードでは、Web ゲートウェイは InterSystems IRIS から受信したすべてのデータを圧縮機能にストリーム転送します。呼び出しを行うたびに、可能な範囲で最大限に圧縮されたデータが作成され、Web ゲートウェイに渡されてクライアントに転送されます。

このモードでは、圧縮のレベルはわずかに低くなりますが、待ち時間は最小限に抑えられます。もちろん、形式が大きくなるほど圧縮レベルは低くなります。一般的に、モード 2 は応答に含まれるデータの量を事前に把握できない場合の Web アプリケーションに適しています。

Web ゲートウェイが InterSystems IRIS から返されたデータ・ストリームを正常に圧縮できる場合 (そして圧縮できる場合のみ)、Web ゲートウェイでは HTTP 応答ヘッダを修正して、適切な Content-Encoding 指示文を追加します。以下に例を示します。

```
HTTP/1.1 200 OK
Content-Type: text/html; charset=ISO-8859-1
Set-Cookie: CSPSESSIONID=000000000002119qMwh3003228403243; path=/csp/samples/;
Cache-Control: no-cache
Connection: Close
Date: <date and time>
Expires: <date and ttime>
Pragma: no-cache
Content-Encoding: gzip
```

Web ゲートウェイは応答データを圧縮する前に、常に Accept-Encoding HTTP 要求ヘッダの値 (HTTP.ACCEPT_ENCODING CGI 環境変数) を確認します。Web ゲートウェイは、クライアントが圧縮したコンテンツを処理できることを示した場合のみ応答を圧縮します。

次に例を示します。

```
Accept-Encoding: gzip, deflate
```

CSP 応答の圧縮を指定するにはいくつかの方法があります。これらは以下のセクションで説明しています。

15.3 ページごとの圧縮の指定

Web アプリケーション内の `%response` オブジェクトには、`GzipOutput` というプロパティがあります。このプロパティを `True` (または必要なモード) に設定すると、Web ゲートウェイは応答の圧縮を試みます。

```
<script language=objectscript method=OnPreHTTP arguments=""
    returntype=%Boolean>
    Set %response.GzipOutput = 2
    Quit 1
</script>
```

HTTP 応答ヘッダに `CSP-gzip` 指示文を追加することで、ページごとに圧縮を指定することもできます。もちろん、`OnPreHTTP` メソッドで行う必要があります。次に例を示します。

```
<script language=objectscript method=OnPreHTTP arguments=""
    returntype=%Boolean>
    Do %response.SetHeader("CSP-gzip", "2")
    Quit 1
</script>
```

`CSP-gzip` ヘッダ指示文は、必要な圧縮モード (1 または 2) に設定します。

15.4 アプリケーション・パス内のすべてのページに対する圧縮の指定

アプリケーション・パスごとに圧縮を指定できます。同時にこれは、Web サーバの出力フィルタ (`mod_deflate` など) を使用する場合に圧縮が必要であることを指定する最も一般的な方法です。

Web ゲートウェイの [アプリケーションアクセス] セクションで以下の構成パラメータを使用します。

アイテム	機能
[GZIP圧縮]	[有効] の場合、そのパスのすべての CSP 出力が圧縮されます。既定値は [有効] です。
[GZIP 最小ファイル・サイズ]	圧縮がアクティブ化される最小応答サイズ (単位はバイト) を制御します。空のままにした場合、GZIP が有効になっているすべての応答が圧縮されます。
[GZIP 除外ファイルの種類]	GZIP 圧縮の除外対象とするファイルの種類のリストです。ファイルは、MIME タイプ (<code>image/jpeg</code> など) または一般的な拡張子 (<code>jpeg</code> など) を使用してリストできます。 既定では、これらの一般的な (標準で圧縮されている) イメージ・ファイルは除外されます。 GZIP Exclude File Types: <code>jpeg gif ico png gz zip mp3 mp4 tiff</code> 追加の種類または拡張子をスペースで区切ります。

15.5 監視

[ログ・レベル](#) V3 では、圧縮が成功したすべての応答に対する圧縮のレベルを記録するように Web ゲートウェイに指示します。圧縮したデータのサイズと圧縮していない元のデータ・ストリームのサイズが記録されます。

次に例を示します。

```
GZIP Compression for /csp/samples/inspector.csp  
GZIP Mode=1; Uncompressed Content Size=19042; Compressed Content Size=2499 (13 percent)
```

16

Web アプリケーション向けの HTTP 認証の実装

Apache モジュール (`mod_csp*.so/dll` および `CSPa*[Sys].so/dll`) により、HTTP 認証を InterSystems IRIS® で制御できます。

Web 要求の HTTP 認証は、通常、Web サーバとクライアント (ブラウザ) の間で行われます。このため、CGI プログラムや Web サーバの API をベースにした要求ハンドラなど、Web サーバによりホストされるカスタム要求ハンドラに HTTP 認証を実装することはできません。もちろん、このような拡張機能で 401 Authorization Required 応答ヘッダを発行することができます。これに対して、ブラウザには HTTP ログイン・ダイアログが表示されます。ただし、それ以降の要求では、Web サーバはユーザ・ログインの詳細を傍受し、独自の組み込み機能を使用してユーザ認証を試みます。Web サーバが独自の方法でユーザを認証するまで、少なくとも、最初のインスタンスでは、ユーザ名とパスワードが要求処理拡張機能に渡されることはありません。

このスキーマは、サードパーティ開発技術 (CSP など) を使用して、その技術の範囲内でローカルに (プログラムによって) HTTP 認証を実行しようとしているユーザにとっては問題となります。

ここで説明した機能により、これらの技術的問題は解決されているため、ユーザは InterSystems IRIS 環境で Apache のホストする [Web アプリケーション](#) に対して HTTP 認証を実行できるようになります。Apache ユーザは、これ以降のセクションで説明する 3 つのアプローチから選択できます。

16.1 Apache での標準 HTTP 認証 (mod_auth)

この方法は、Apache により (mod_auth モジュール経由で) 提供される標準メカニズムで、Web ゲートウェイは使用しません。ここでは、完全を期すために説明しています。

Apache ベースの認証を使用する CSP サンプルを保護するために必要な基本パラメータの例は、以下の構成ブロック (`httpd.conf`) のとおりです。

```
<Location "/csp/samples/">
    AuthType Basic
    AuthName "CSP samples"
    AuthUserFile conf/csp.pwd
    require valid-user
</Location>
```

以下はその説明です。

AuthType は要求された認証タイプ (通常は Basic) です。

AuthName はレルムです。

AuthUserFile はユーザ名とそれに関連付けられたパスワードを (暗号化された形で) 保持するファイル (Web サーバのルートとの相対パスで表されたもの) です。このファイルは、Apache httpasswd ユーティリティで作成され、管理されます。

require パラメータは、保護されたリソース (この場合は CSP サンプル) にアクセスが許可されたユーザのリストです。valid-user 引数は、ユーザを (AuthUserFile で宣言されたとおり) ユーザ名/パスワード・ファイルで定義する必要があることを示します。

Apache は、ユーザ 'groups' にグループ化されるユーザに対応しています。詳細は、AuthGroupFile 指示文を参照してください。

https://httpd.apache.org/docs/2.4/mod/mod_authz_groupfile.html#authgroupfile

16.2 要求の処理と同時に CSP で行われる認証

これは、Web アプリケーションでの HTTP 認証の実装に推奨される (また、最もパフォーマンスのよい) メソッドです。

CSP ベースの認証を使用する CSP サンプルを保護するために必要な基本パラメータは、以下の Apache 構成ブロック (httpd.conf) のとおりです。

```
<Location "/csp/samples/">
    AuthType Basic
    AuthName "CSP samples"
    require valid-user
    AuthCSPEnable On
    AuthBasicAuthoritative Off
</Location>
```

パラメータ AuthType、AuthName、および require は認証のトリガに使用される標準 Apache パラメータです。

また、追加の AuthCSPEnable パラメータは、(mod_auth で) Apache により行われる認証チェックを迂回し、本来の Web 要求と共にユーザ名とパスワードを認証のために InterSystems IRIS に渡すように CSP モジュールに指示します。Web アプリケーションは、以下の CGI 環境変数を使用して、このユーザをチェックする必要があります。

- ・ AUTH_TYPE : これは Basic です。
- ・ REMOTE_USER : ユーザ名。
- ・ AUTH_PASSWORD : ユーザのパスワード (プレーン・テキスト)。

これらのパラメータに保持された値に基づいて、ユーザが正常に認証されると、アプリケーションは要求の処理を続行します (つまり、要求された CSP リソースを返します)。認証されなかった場合、最低でも次のような HTTP 401 Authorization Required 応答が返されます。

```
HTTP/1.1 401 Authorization Required
WWW-Authenticate: Basic realm="CSP samples"
Content-Type: text/html
Connection: close
<html>
<head><title>401 Authorization Required</title>
</head><body> <h1>Authorization Required</h1>
<p> The server could not verify that you are authorized
to access the application. Check your username and password.
</p>
<hr>
</body>
</html>
```

このメッセージが受信されると、ユーザが既定の回数 (通常は 3 回) を超えてログインを試行していない限り、ブラウザにはログイン・ダイアログが再表示されます。超えてしまった場合は、ヘッダに続いてこのメッセージが表示されます。

この認証方法は、ログイン・ページを変更することで実装できます。要求が到着したときに、アプリケーションを実行するために必要な権限をユーザが保持していなかった場合、ログイン・ページが呼び出され、要求から認証情報 (AUTH_TYPE、REMOTE_USER、AUTH_PASSWORD など) を抽出する処理が行われます。これらのパラメータが正当と判断された場合、ログイン・スクリプトは、本来要求されたアプリケーション・ページに制御をリダイレクトします。InterSystems IRIS セキュリティのコントロール・レイヤが展開されている場合、パブリック・ページすべてに対してこの認証プロシーダを繰り返す必要はありません。

16.3 要求の処理前に CSP で行われる認証

これは、InterSystems IRIS で HTTP 認証を実装するための代替方法です。これは主に、Web アプリケーションでの要求処理時には認証を実行することが難しいか、実行に時間がかかる場合に行われます。

この方法では、ユーザ認証のために、専用の認証クラスが呼び出されます。[Web ゲートウェイ](#)は、InterSystems IRIS に元の要求を送信する前に、このチェックを実行します。認証クラスによるユーザの詳細チェックが正常に終了すると、これ以上、Web アプリケーションでチェックを実行する必要はありません。

もちろん、この方法には、Web 要求 1 回に対して、認証に 1 回、CSP リソースに対する実際の要求の処理に 1 回の合計 2 回 (InterSystems IRIS への) 要求を処理するというオーバーヘッドがあります。

この認証方法を実装するために必要な基本パラメータは、以下の Apache 構成ブロック (`httpd.conf`) のとおりです。

```
<Location "/csp/samples/">
    AuthType Basic
    AuthName "CSP samples"
    require valid-user
    AuthCSPEnable On
    AuthCSPClass /csp/samples/%CSP.HTTPAuthentication.cls
    AuthBasicAuthoritative Off
</Location>
```

パラメータ `AuthType`、`AuthName`、`require`、および `AuthCSPEnable` はメソッド (2) と同じです。

また、`AuthCSPClass` パラメータはユーザ認証を実行するクラスを定義します。このクラスは、適切な CGI 環境変数を使って `%CSP.Page` を拡張し、ユーザ・ログインの詳細をチェックして、操作に成功した場合は 200 OK 応答ヘッダ、失敗した場合は 401 Authorization Required 応答ヘッダを返す必要があります。

`%Users` ファイルに保存されているレコードに対するユーザ・ログインの詳細のチェックが行われる単純な認証クラスは以下のとおりです。

```
Class %CSP.HTTPAuthentication Extends %CSP.Page
{
    ClassMethod OnPreHTTP() As %Boolean
    {
        Set %response.ContentType = "text/html"
        Set %session.Preserve = 0
        Quit 1
    }
    ClassMethod OnPage() As %Status
    {
        Set crlf=$Char(13,10)
        Set type=%request.GetCgiEnv("AUTH_TYPE", "")
        Set user=%request.GetCgiEnv("REMOTE_USER", "")
        Set pwd=%request.GetCgiEnv("AUTH_PASSWORD", "")
        Set httpauth=%request.GetCgiEnv("HTTP_AUTHORIZATION", "")
        If httpauth="" {
            Set type=$Piece(httpauth," ",1)
            Set user=$system.Encryption.Base64Decode($Piece(httpauth," ",2))
            Set pwd=$Piece(user,":",2)
            Set user=$Piece(user,":",1)
        }
        Set auth=0
        If $ZConvert(type,"L")="basic" Set auth=1
        If auth=0,user=""",$Get(^%Users(user))=pwd Set auth=1
        If auth=1 {
            Write "HTTP/1.1 200 OK"_crlf
            Write "Content-Type: text/html"_crlf
        }
    }
}
```

```
        Write "Content-Length: 0"_crlf
        Write "Connection: close"_crlf_crlf
    }
    Else {
        Write "HTTP/1.1 401 Authorization Required"_crlf
        Write "WWW-Authenticate: Basic realm=""CSP samples""_crlf
        Write "Content-Type: text/html"_crlf
        Write "Content-Length: 0"_crlf
        Write "Connection: close"_crlf_crlf
    }
    Quit $$$OK
}
ClassMethod OnHTTPHeader(ByRef OutputBody As %Boolean) As %Status
{
    Quit $$$OK
}
}
```

メソッド (1) および (3) では、Apache `ErrorDocument` 指示文を使用して、ログインの失敗に対するカスタム・エラー・ページを指定できます。以下はその例です。

```
ErrorDocument /error/my_authentication_error.html
```

もちろん、メソッド (2) では、エラー・メッセージの内容は Web アプリケーションによって制御されます。

17

負荷分散、フェイルオーバー、ミラー構成

17.1 複数の Web サーバ間の負荷分散とフェイルオーバー

ほとんどの環境では、Web サーバ層で負荷を分散して高可用性を実現するために、複数の Web サーバが使用されます。一般に、参加する Web サーバに対してユーザ接続を行うには、ロード・バランサが必要です。最適なパフォーマンスと復元力を得るには、ハードウェア・ベースのソリューションの使用をお勧めします。Cisco ACE 4710 や F5 BigIP LTM アプライアンスなどの負荷分散システムは、Web サーバ群の前面に配置されます。この構成では、分散キャッシュ・クラスタのように、複数の InterSystems IRIS サーバ・インスタンスも存在している場合、各 Web サーバ (および Web ゲートウェイ・インスタンスも含めて) を特定の InterSystems IRIS® サーバ・インスタンスに接続するように構成してください。

ソフトウェア・ベースの負荷分散およびフェイルオーバー・システムは、ハードウェア・ベースのソリューションほど堅牢ではありませんが、非常に低コストで導入できます。ソフトウェア・ベースのソリューションの例としては、HAProxy や Apache Group の `mod_proxy_balancer` があります。詳細は、HAProxy のサイト www.haproxy.org を参照してください。

重要 [Web アプリケーション](#)に対して必ず永続 (“スティッキー”) セッションを有効にしておいてください。セッションが存続している間は、それぞれのユーザ・セッションを同一のバックエンド InterSystems IRIS サーバで維持する必要があります。ただし、当然のことながら、フェイルオーバー・イベントが発生した場合はこの限りではありません。

上記のアプローチが最も推奨される方法ですが、Web ゲートウェイには複数の InterSystems IRIS サーバ (すなわち、複数の InterSystems IRIS インスタンスの CSP サーバ) 間で負荷分散やフェイルオーバーを実装するための基本の (ソフトウェアベースの) システムが用意されています。この機能については、以下のセクションで説明します。

17.2 複数の InterSystems IRIS サーバ・インスタンス間の負荷分散とフェイルオーバー

複数 (同等) の InterSystems IRIS サーバ・インスタンスが含まれる構成 (分散キャッシュ・クラスタなど) では、[Web ゲートウェイ](#)は、それらの InterSystems IRIS インスタンス間で負荷分散とフェイルオーバーを実装するための基本 (ソフトウェアベース) の機能を Web アプリケーションに提供します。しかし、第一に推奨されるのは、前述のような外部ソリューションです。

Web ゲートウェイによって提供されるフェイルオーバー・メカニズムでは、フェイルオーバー・クラスタリングや InterSystems IRIS ミラーリングなどの一般的な高可用性構成において、複数の InterSystems IRIS データベース・サーバ間にフェイルオーバーを実装する必要はありません。それらのテクノロジーは仮想 IP ベースのフェイルオーバーを提供しており、その IP アドレスに接続するように Web ゲートウェイを構成できます。

このセクションの残りの部分では、Web ゲートウェイによって提供される負荷分散およびフェイルオーバー機能について説明します。

Web ゲートウェイの負荷分散およびフェイルオーバーは、Web ゲートウェイ管理ページの **[アプリケーションアクセス]** セクションで構成されます。“[アプリケーション・アクセスの構成](#)”を参照してください。

[システム管理] > [構成] > [ウェブゲートウェイ管理] に移動して、**[アプリケーションアクセス]** を選択します。InterSystems IRIS サーバのリストを、アプリケーション (パス) に対して定義できます。**[代替サーバの使用目的]** パラメータの下にリストされているオプションを使用して、そのサーバの使用目的を選択します。以下のオプションを使用できます。

- ・ フェイルオーバー
- ・ 負荷分散とフェイルオーバー

既定の対策は、リストで定義されている最初の InterSystems IRIS サーバを使用することです。この既定のサーバに続いて代替 InterSystems IRIS サーバのリストがあり、それぞれ **Server #** と指定されます (# はサーバ番号です)。

構成画面には、最初は 3 つの空のサーバ・スロットのみが表示されていますが、追加のスロットが表示され、任意の数の代替サーバを定義できます。それぞれのサーバは、**[有効]** または **[無効]** としてマークできます。既定の設定は **[有効]** です。

負荷分散は、ラウンドロビン方式で実装されます。新しい各ユーザ・セッションは、次に使用可能な代替サーバに接続されます。ユーザ・セッションがサーバで確立されると、Web ゲートウェイはサーバが使用不可能にならない限り、そのサーバでセッションを維持しようとします。このサーバが使用不可能になった場合、セッションはリスト内で次に使用可能なサーバにフェイルオーバーされます。ステート認識セッション (保持モード = 1) の場合、どのような状況でもフェイルオーバーは行われないので、ホスト・サーバが使用不可能になるとセッションは閉じます。

CSP サーバが、**[サーバ応答タイムアウト]** で指定された期間に応答しない場合、Web ゲートウェイはそのサーバをオフラインとマークし、負荷分散には使用しません。ただし、Web ゲートウェイの**レジストリ機能**が有効な場合 (これらは既定で有効です)、Web ゲートウェイは定期的にオフラインの CSP サーバへの再接続を試みます。CSP サーバへの接続に成功すると、Web ゲートウェイはこれをオンラインにマークし、自動的に再度負荷分散に使用します。

17.3 ミラー構成

ミラー InterSystems IRIS 構成を使用すると、データベースは参加するミラー・メンバ間で複製 (またはミラーリング) されます。InterSystems IRIS ミラー・セット構成は、インストールに参加するミラー・メンバのセットを表します。InterSystems IRIS ミラーリングの詳しい説明は、“[高可用性ガイド](#)”の“[ミラーリング](#)”を参照してください。

プライマリ・メンバに対してネットワークのリダイレクトを提供するためにミラー仮想 IP (または同等のテクノロジー) が使用される場合は、そのアドレスに接続するように Web ゲートウェイを構成します。それ以上のアクションは必要ありません。仮想 IP アドレスは常にミラー・プライマリにマップされます。

ミラー仮想 IP を使用できない (または特定の災害シナリオでは機能しない) 構成の場合、Web ゲートウェイをミラー認識になるように構成できます。Web ゲートウェイがミラー認識の場合、どのメンバがプライマリかを決定する責任を負います。Web ゲートウェイをミラー認識にするには、Web ゲートウェイの **[サーバ接続]** セクションで、**[構成はミラーを認識する]** を選択し、ミラー・メンバのいずれかのアドレスを指定します。

注釈 Web ゲートウェイ構成をミラー認識にするのが適切でない状況があります。例えば、管理ポータルをサポートする Web ゲートウェイ構成は、ミラー認識になるように構成してはなりません。このポータルはミラーの状態にかかわらず、常に特定の InterSystems IRIS サーバに接続する必要があるからです。

ミラー認識の Web ゲートウェイ構成が、ミラー・メンバでない InterSystems IRIS サーバに接続する場合、接続は失敗し、影響を受けるクライアントは **Server Availability** エラーを受け取ります。

Web ゲートウェイは、最初に接続するメンバから、フェイルオーバー・メンバおよび災害復旧 (DR) メンバのリストを入手します。Web ゲートウェイはこのリストをローカル構成ファイル (CSPRT.ini) に維持します。その後、Web ゲートウェイがその構成で定義されたメンバに接続できなくなる場合、以前にローカルに記録されたリストを使用して、代替メンバを識別し、それに接続できるようにします。

Web ゲートウェイは、プライマリを見つけるまで、メンバ・リストを循環します。プライマリが見つからない場合、Web ゲートウェイは、ゲートウェイ構成で定義されたサーバに既定で設定されます。

- ・ Web ゲートウェイは、プライマリとして定義されたメンバが見つかるまで、リストを繰り返し循環します。
- ・ 密接なループ構造がパフォーマンスに与える悪影響を回避するため、Web ゲートウェイは各サイクルの終了後、試行回数に相当する秒数の間、停止します。
- ・ 特定の HTTP 要求では、Web ゲートウェイは、[サーバ応答タイムアウト] パラメータで定義された時間を超えてプライマリを見つけることはありません。
- ・ プライマリを検索する場合、Web ゲートウェイが最初に接続するのは、常にフェイルオーバー・メンバです。フェイルオーバー・メンバの中にプライマリが見つからない場合のみ、非同期メンバを試行します。非同期メンバを手動でプライマリと指定する場合のみ、非同期メンバはプライマリになります。

最初の接続時に、ミラー・メンバが Web ゲートウェイ・システム・ステータス・フォームに表示されます。ミラー・メンバは、(Web ゲートウェイの [サーバ接続] セクションで定義されたとおりに) 現行の構成名で表示されます。その際に、ツールのヒントとしてミラー・セット名、ミラー、ミラー・メンバ名が表示されます。

列 [ミラー名] および [ミラー・ステータス] が [InterSystems IRIS サーバ] テーブルに表示されます。ミラー・セットおよびミラー・メンバの名前が [ミラー名] 列に表示されます。現在のメンバ・ステータスが [ミラー・ステータス] 列に表示されます。[メンバタイプ] (Failover または Async) が表示され、プライマリ・メンバには Primary というラベルが付けられます。

18

プロセス親和性とステート認識モード (保持モード 1)

Web のアーキテクチャはステートレスです。Web アーキテクチャから最高のパフォーマンス、保守性、およびスケーラビリティを引き出すには、[Web アプリケーション](#)はステートレス・パラダイムを採用する必要があります。

既定では、Web アプリケーションはホストする InterSystems IRIS® サーバに対してステートレスな環境で動作します。[Web ゲートウェイ](#)は InterSystems IRIS への接続プールを維持して、それらに負荷を分散し、構成された制限以内で接続プールのサイズを増加または減少させます。各接続は (\$Job 変数によって指定される) 1 つの InterSystems IRIS プロセスに関連付けられます。

ステートレス・モードで動作する通常の Web アプリケーションの場合は、クライアント・セッションに対する各要求を処理するために使用されるバックエンド InterSystems IRIS プロセスの選択をランダムにすることを検討してください。Web ゲートウェイは、そのときに利用可能な任意の接続/プロセスを選択します。

ただし、効率性を高めるために、Web ゲートウェイは一種の InterSystems IRIS プロセス親和性を実装します。つまり、可能であれば、セッションに対する前回の要求を処理するために使用されたものと同じ InterSystems IRIS プロセスにセッションへの要求を転送しようと試みます。

Web ゲートウェイはセッション ID に基づいたプロセス親和性に加え、ネームスペースに基づいたプロセス親和性の実装も試みます。Web ゲートウェイは各接続がポイントするネームスペースを追跡し、可能な場合は、要求の処理に必要なネームスペースを既にポイントしている接続へ要求を配信します。これにより、各 Web 要求の受信時に異なるネームスペース間でリソースを移動する際に発生するオーバーヘッドを回避できます。

優先順位という点では、接続を選択する際は、セッション親和性が常に他のすべての考慮事項より優先されます。受け取った要求をそのクライアント・セッションの処理に前回使用された接続に割り当てることができない場合は、代わりにネームスペース親和性を使用して最終的な選択が行われます。

CSP には、セッションに対するすべての要求を予約済みの (またはプライベート) InterSystems IRIS 接続/プロセスに Web ゲートウェイが転送するモードが含まれています。この処理モードは、Web セッションとそれらに対応する InterSystems IRIS プロセスの関係に関してステート認識の環境を提供します。

ステート認識モードは CSP 保持モード 1 として実装

ステート認識モード ([保持モード 1](#)) による処理は、固定クライアント・サーバ環境 (ターミナル・アプリケーション) から Web への従来のアプリケーション・コードの移行を比較的簡単にすることを当初の目的として提供されました。また、複数の HTTP 要求に広がるトランザクションのサポートも考慮されました。ただし、ステート認識アプリケーションを作成する際は、以下の段落で概説する制限事項に注意することが必要です。

ステート認識アプリケーションは、ステートレス・アプリケーションほどスケーラビリティが高くないため、新しいアプリケーション (および既存のアプリケーションの変更) は実質的にできる限りステートレスに設計することが推奨されます。ステート認識モードを使用する場合は、大部分はステートレスなアプリケーションで、控えめに適用することをお勧めします。

アプリケーション全体をステート認識モードで動作するように作成することはお勧めできません。ステート認識アプリケーションでは、全セッションそれぞれに InterSystems IRIS プロセスを予約する必要があるためにスケーラビリティの問題が発生するほか、転送要求に対する非常に固有の要件により、最新の負荷分散およびフェイルオーバー・ソリューションをフルに利用することができません。また、ステート認識アプリケーションは、ステートレス・アプリケーションほど優れた耐障害性を備えていません。例えば、Web サーバ・ワーカ・プロセスのリサイクルはステートレス・アプリケーションの下で透過的に発生しますが、その結果、関連付けられたすべてのステート認識セッションが閉じてしまいます。もちろん、Web ゲートウェイの NSD コンポーネントを使用して、Web ゲートウェイ・プロセス・プールの管理をホスト Web サーバから分離することで、この後者の制限事項は回避できます。

適切なステート認識アプリケーション (または大部分がステートレス・アプリケーション内のステート認識セクション) を作成するには、ある程度のコントロールが必要です。

セッションに対するすべての要求は同じ InterSystems IRIS プロセスによって処理する必要があるため、クライアントが複数の要求を同時に送信する場合にプライベート InterSystems IRIS プロセスへのアクセスをシリアル化するためのキューを維持する必要があります。元の HTTP v1.1 標準は、クライアントは各サーバに対して同時に 3 つ以上の接続を開いてはいけなくと定めています (RFC2616)。ただし、この制限は構成可能で、実際には、最新世代の Web ブラウザは、既定で、各サーバに対して最大 8 つの接続をサポートしています。もちろん、各サーバへの最大接続数が増加すると、ステート認識 Web アプリケーションは深く影響されます。つまり、アプリケーションは、最大 8 つの要求が同時に送信され、1 つのプライベート InterSystems IRIS プロセスへのアクセスを制御するキューに保持されることを期待するようになります。

ステート認識モードで考えられるもう 1 つの落とし穴は、Web ゲートウェイと InterSystems IRIS の間で動作するサーバ応答タイムアウトの影響です。応答タイムアウトによる所定の制限時間内に Web ゲートウェイが応答を受信しなかった場合は、接続を閉じる以外の措置をとることはできず、結果的にステート認識セッションは失われてしまいます。

最後に、クライアントが中断すると、ステート認識モードで動作しているアプリケーションで問題が発生する可能性があります。InterSystems IRIS が応答を生成する時点以降にクライアントが要求を中断すると、Web ゲートウェイは接続を保持するために (不要になった) 応答ペイロードを吸収しようとして、この場合も、これをタイムリーに行えない場合は、InterSystems IRIS プロセスを中断する以外の措置をとることはできなくなり、接続が閉じられてセッションが失われることとなります。中断された要求のペイロードを Web ゲートウェイが吸収しようとしている間に、同じセッションに対するさらなる要求が到着してキューに置かれる可能性があることに注意してください。

ステート認識アプリケーションを作成する際に従うべき設計目標を以下にまとめます。

- ・ 多数の同時要求を生成するクライアント構造 (例 : HTML フレームセット・ドキュメント) は、できるだけ避けます (または控えめに使用します)。
- ・ 応答が迅速に生成されるようにします。これにより、タイムアウトやクライアント中断イベントに関連する問題の範囲が縮小します。また、セッション・キューの圧力が軽減されます。InterSystems IRIS のタスクを完了するのに長い時間がかかると予測される場合は、プライマリ・プライベート・プロセスが Web ゲートウェイ (とクライアント) にすばやく応答を返すことができるように、別のプロセスでそのタスクを実行することを検討してください。

18.1 ステート認識モードの起動

以下のように保持モードを設定することで、セッションをステート認識にします。

```
Set %session.Preserve = 1
```

フォームの OnPreHTTP メソッドでセッションをステート認識としてマークすることを推奨します。

```
<script language=objectscript method=OnPreHTTP arguments="" returntype=%Boolean>
Set %session.Preserve = 1
Quit 1
</script>
```


ここで指示を出すことで、**CSP エンジン**は HTTP 応答ヘッダを作成して Web ゲートウェイに送信する前に、セッションの cookie (またはトークン) をステート認識としてマークできます。

OnPreHTTP メソッドが起動した後でセッションをステート認識としてマークすることもできますが、この場合、セッションの cookie/トークンは既に作成されています。CSP エンジンは、`preserve=1` という指示を応答フッタで Web ゲートウェイに渡し (応答ペイロードの後で送信)、Web ゲートウェイは接続を `private` とマークして、そのセッション ID に対する指示をキャッシュします。これによって Web ゲートウェイは、以降の要求が到着した際、`unmodified` セッション・トークンをステート認識として識別することが可能になります。

OnPreHTTP メソッドでセッションがステート認識としてマークされた場合、その情報は事実上クライアントに常駐するセッションの cookie/トークンで運ばれるため、Web ゲートウェイはそのセッションの移行をキャッシュする必要はありません。

18.2 ステート認識モードの維持とエラーへの応答

セッションがステート認識としてマークされ、Web ゲートウェイがステートの移行を認識して接続をプライベートとマークすると、以下のイベントのいずれかが発生するまでセッションは透過的にステート認識モードで動作します。

- ・ アプリケーションがステートレス処理モードに戻った場合。
- ・ アプリケーションがプログラムによってセッションを終了するか、セッションがタイムアウトした場合。
- ・ エラー状態の結果としてプライベート接続が不完全に閉じた場合。

(おそらくエラー状態の結果として) ステート認識アプリケーションをホストするプライベート接続が不完全に閉じた場合、Web ゲートウェイはブールで利用可能なステートレス接続に要求を転送し、InterSystems IRIS エラー番号 5974 が返されます。

```
CSP error occurred
Error: The persistent session is no longer available because the server process does not exist
ErrorNo: 5974
CSP Page: /csp/samples/loop.csp
Namespace: %SYS
Class: <Unknown>
```

この時点では、要求はステートレス・モードで処理され、このエラーにはアプリケーションが例えばユーザにアプリケーションのログイン・フォームを再度表示するなどして応答します。

ステート認識モードで動作している際は、全ページの `%session.NewSession` の値をチェックする必要があります。または、アプリケーションへのアクセスをユーザが最初に許可されたときに、アプリケーションが `%session.Data` に格納されているユーザ固有の認証データの有効性をチェックする必要があります。これらのチェックはセキュリティのためにも、ユーザ・セッションが依然としてステート認識処理モードにしっかり固定されていることを確認するためにも重要です。これらの状況では、エラー状態は自動的に発生しません。なぜなら、セッションは既に (適切な手順を踏んで正常に) ステート認識モードから移行している可能性があるからです。例えば、受信セッションのトークンは依然としてステート認識とマークされているもののアプリケーションは既にステートレス・モードに移行している状況を考えてみましょう。この状況は、移行が行われる前に処理されたフォームに (CSPCHD として) セッションのトークンが埋め込まれているために起こります。

最後に、(例えば、タイムアウトの後で) セッションが終了する際、**CSP エンジン**はそのセッションに関連付けられたすべての処理データを削除することに注意してください。その後は、そのセッションに対する受信要求は、すべて新しいセッションに対する要求のように処理されます。

InterSystems IRIS が Web アプリケーションに提供する埋め込みのセキュリティ・メカニズムは、前述の不測の事態から保護します。(InterSystems IRIS プロセスに関して) ステート認識アプリケーション内の継続性が失われた場合は必ず、ユーザには自動的にログイン・フォームが表示されます。

18.3 ステート認識モードの終了

以下のように保持モードを設定することで、アプリケーションをステートレス処理モードに戻すことができます。

```
Set %session.Preserve = 0
```

このコードは、フォームの OnPreHTTP メソッドで実行することを推奨します。

```
<script language=objectscript method=OnPreHTTP arguments="" returnType=%Boolean>  
    Set %session.Preserve = 0  
    Quit 1  
</script>
```

ここで指示を出すことで、[CSP エンジン](#)は HTTP 応答ヘッダを作成して Web ゲートウェイに送信する前に、セッションの cookie (またはトークン) をステートレスとしてマークできます。

セッションは以下のように即座に終了できます。

```
Set %session.EndSession = 1
```

このプロパティを設定すると、セッションは現在の要求を処理した後で直ちに終了します。

以下のようにセッションのタイムアウトを設定できます。

```
Set %session.AppTimeout = 900
```

アクティビティがない時間が所定の秒数に達すると、セッションはタイムアウトして終了します。既定値は 900 秒 (15 分) です。

19

InterSystems IRIS の Web ゲートウェイ・レジストリ

InterSystems Web ゲートウェイ・レジストリは、それぞれの接続された [Web ゲートウェイ](#) のインストール環境を InterSystems IRIS® に登録し、InterSystems IRIS コードがこれらのインストール環境と対話する（キャッシュのクリアなどのため）ことを可能にするインフラストラクチャを提供します。そのようなプログラムで制御された通信には、Web ゲートウェイの実行時構成の読み取りと変更、およびシステム・ステータスとログ情報の収集が含まれます。関連するクラスは以下のとおりです。

```
%CSP.Mgr.GatewayRegistry (The Gateway Registry)
%CSP.Mgr.GatewayMgr (A Connected Gateway)
```

以下のコードは、すべての接続された（つまり、アクティブな）Web ゲートウェイのインストール環境をリストし、Web サーバ IP アドレス、ポート、および Web ゲートウェイ・ビルド番号をコンソール・ウィンドウに書き込みます。

```
Set registry = $system.CSP.GetGatewayRegistry()
Set gateways = registry.GetGatewayMgrs()
For no=1:1:gateways.Count() {
    Set gateway = gateways.GetAt(no)
    Write !,no, " : "
    Write gateway.IPAddress,":",gateway.Port," ",gateway.Version
}
```

InterSystems IRIS が最初に起動されたときは、このリストは空です。管理者およびユーザのアクティビティが増加すると、インスタンスのアプリケーションを提供する Web ゲートウェイ用に少なくとも 1 つのエントリが表示されます。

上記のクラスについては、他のドキュメントで説明されています。一般的なタスクを説明するコード例を以下に示します。

既定のパラメータのリスト

```
Kill defaults
Do gateway.GetDefaultParams(.defaults)
ZWrite defaults
```

既定のパラメータの更新

```
Kill newpars
Set newpars("Server_Response_Timeout")=30
Do gateway.SetDefaultParams(.newpars)
```

サーバのリスト

```
Set status = gateway.GetServers(.servers)
For no=1:1:$ListLength(servers) {
    Set server = $List(servers,no)
    Write !,no, " : ",server
}
```

サーバ・パラメータのリスト

```
Kill serverpars
Do gateway.GetServerParams("LOCAL",.serverpars)
ZWrite serverpars
```

サーバ・パラメータの更新

```
Kill newpars
Set newpars("Maximum_Server_Connections")=250
Do gateway.SetServerParams("LOCAL",.newpars)
```

アプリケーション・パスのリスト

```
Set status = gateway.GetApplicationPaths(.paths)
For no=1:1:$ListLength(paths) {
    Set path = $List(paths,no)
    Write !,no, " : ",path
}
```

アプリケーション・パラメータのリスト

```
Kill pathpars
Do gateway.GetApplicationParams("/csp",.pathpars)
ZWrite pathpars
```

アプリケーション・パラメータの更新

```
Kill newpars
Set newpars("GZIP_Compression")="Enabled"
```

ゲートウェイ・キャッシュのクリア

```
Do gateway.ClearCache("**")
```

19.1 構成を再ロードするよう Web ゲートウェイを強制する

Web ゲートウェイの構成が外部エージェント（つまり、Web ゲートウェイ独自のシステム管理スイート以外のエージェント）によって変更される場合があります。

完全な再起動を必要とせずに、Web ゲートウェイに構成を再ロードするようインタラクティブに指示するには、以下の 2 つの方法があります。

19.1.1 InterSystems IRIS Web ゲートウェイ・レジストリの使用

以下のレジストリ・メソッドが提供されています。

```
Set status = %CSP.Mgr.GatewayMgr.ActivateCSPIni()
```

呼び出しが成功すると、Web ゲートウェイはその構成ファイルを読み取り、行われたすべての変更を有効化します。

19.1.2 InterSystems IRIS の外部スクリプトの使用

スクリプトでは、変更された Web ゲートウェイの構成ファイルの SYSTEM セクションに次の行（大文字と小文字が区別される）を追加する必要があります。

```
[SYSTEM]
RELOAD=1
```

Web ゲートウェイ・ケアテイカー・デーモンは、ほぼ 1 分おきに RELOAD フラグを確認し、正しく設定されていれば、その構成を再ロードおよび再起動して、ファイルからフラグを削除します。再ロード操作が成功すると、以下のメッセージがイベント・ログに書き込まれます。

```
Gateway Management  
Gateway Configuration Reloaded and Reactivated
```


Web ゲートウェイ構成ファイル (CSP.ini) パラメータ・リファレンス

InterSystems® Web ゲートウェイは、Web ゲートウェイ・バイナリと同じディレクトリにある CSP.ini ファイルに構成情報を保持します。

ほとんどの場合、CSP.ini ファイルを直接編集するのではなく、Web ゲートウェイ管理ページまたは Web ゲートウェイ・レジストリ・メソッドを使用して Web ゲートウェイ構成を変更することをお勧めします。ただし、ユース・ケースによっては、これが現実的ではないこともあります。例えば、事前構成された webgateway コンテナを拡張性の高い Kubernetes クラスタの一部として自動的に初期化するには、導入時に CSP.ini マージ機能を使用して、各 Web ゲートウェイの CSP.ini ファイルを生成する必要があります。このユース・ケースの詳細は、“InterSystems Web ゲートウェイ・コンテナの使用法”を参照してください。

CSP.ini パラメータに変更を加えたら、Web ゲートウェイ・レジストリの %CSP.Mgr.GatewayMgr.ActivateCSPIni() メソッドを使用するか、CSP.ini ファイルの [SYSTEM] セクションに RELOADを追加して、Web ゲートウェイ構成を手順どおり再ロードできます。

CSP.ini ファイルのパラメータは、複数のタイプの機能セクションに編成されています。このリファレンスでは、CSP.ini ファイルの内容をセクションごとに説明します。

[SYSTEM]

Web ゲートウェイの**既定のパラメータ**について説明します。CSP.ini ファイルには 1 つの [SYSTEM] セクションのみが含まれます。

使用可能なパラメータ

RELOAD

Web ゲートウェイ・ケアテイカー・デーモンに、Web ゲートウェイ構成を再ロードおよび再アクティブ化するよう指示するフラグ。その後、ケアテイカー・デーモンはフラグを削除します。手順どおりに**このフラグを使用して再ロードをトリガ**するには、RELOAD=1 を指定します。

DPAPI

(Windows マシンにインストールされた Web ゲートウェイの場合のみ。)CSP.ini ファイル内のパスワード・フィールドを、Microsoft のデータ保護 API (DPAPI) を使用して自動的に暗号化するかどうかを決定するフラグ。パスワードの暗号化は既定で有効になっており、有効のままにすることをお勧めします。**代替ソリューションがない**場合は、DPAPI=Disable を指定して DPAPI パスワード暗号化を無効にすることができます。

Instance_Host_Name

既定とは異なる場合の、**この Web ゲートウェイのネットワーク・ホスト名** (server_name:port)。このパラメータは、システム変数 CSPIHN として、要求データと共に InterSystems IRIS へ送信されます。

MAX_CONNECTIONS

この Web ゲートウェイが InterSystems IRIS アプリケーション・サーバ・プロセスに確立できる**最大接続数**。既定値は 1024 です。

MAX_CACHE_SIZE

アプリケーションの応答データをキャッシュするために割り当てられる**共有メモリの最大量**。キロバイト (128K) 単位またはメガバイト (16M) 単位で定義されます。既定値は 256K です。

Web_Server_ID_Cookie

ロード・バランサがパッシブなクッキーの親和性を実装するために、**Web サーバ ID クッキー** (CSPWSERVERID) を Enabled (有効) にするか Disabled (無効) にするか。既定値は Enabled です。

SM_Forms

Web ゲートウェイ管理ページへのアクセスを Enabled (有効) にするか Disabled (無効) にするか。既定値は Enabled です。

Username

指定されている場合、Web ゲートウェイ管理ページへのアクセスに必要なユーザ名。

Password

指定されている場合、Web ゲートウェイ管理ページへのアクセスに必要なパスワード。

SM_Timeout

Web ゲートウェイ管理ページに認証が構成されている場合、[認証されたセッションの期限が切れるまでの許容アイドル時間](#) (秒単位)。既定値は 28800 です。

Custom_SM_Login_Form

Web ゲートウェイ管理ページへのアクセスを制御する、[カスタム・ログイン・フォーム](#)への完全なファイル・パスまたは相対 URL パス。

System_Manager

[Web ゲートウェイ管理ページへのアクセスを許可されたマシン](#)に対応する IP アドレスのリスト。ワイルドカード文字 (*) を使用してアドレスの範囲を指定できます (例 : 190.8.7.* または *.*.*.*)。既定では、アクセス権はホスト・マシンにのみ付与されます。このパラメータを構成するための詳細なガイダンスは、["追加クライアント・アドレスからのアクセスの有効化"](#) を参照してください。既定では、ローカル・マシンのみ許可されます。

Accept_X_Forwarded_For

指定した場合、System_Manager パラメータで指定された元の IP アドレスからプロキシ・サーバによって転送された要求に応じて、Web ゲートウェイ管理ページへのアクセスを許可するかどうか。使用できる値は Enabled (許可する) と Disabled (拒否する) です。

System_Manager_UNPW_Override

指定した場合、Web ゲートウェイ管理ページで、System_Manager パラメータによって指定されたマシンのユーザ名とパスワードが必要かどうかを決定します。使用できる値は 1 (オーバーライドが有効) または 0 (オーバーライドが無効) です。

Server_Response_Timeout

[ターゲットの InterSystems IRIS アプリケーション・サーバが Web サーバからの要求に応答するために許容される最大時間](#) (秒単位)。許容される最小値は 5 です。既定値は 60 です。

Queued_Request_Timeout

[要求が、該当する InterSystems IRIS サーバをキュー内で待機できる最大時間](#) (秒単位)。許容される最小値は 5 です。既定値は

No_Activity_Timeout

ステートレスな接続の場合、[Web ゲートウェイが InterSystems IRIS アプリケーション・サーバへのアイドル状態の接続を閉じるまでの最大時間](#) (秒単位)。既定値は 86400 です。

Timeout_All_Connections

指定した場合、No_Activity_Timeout パラメータで指定された時間、[アイドル状態のままの接続を終了するかどうか](#)を決定します (最小接続プール内の接続を含む)。

Env_Parameters

[イベント・ログ・レベル](#)。

Event_Log_File

指定した場合、[Web ゲートウェイのイベント・ログ](#)を格納する代替ファイルシステムの場所とファイル名を決定します (指定しなかった場合、Web ゲートウェイはログを Web ゲートウェイ・バイナリ・ファイルと同じディレクトリに CSP.log として保存します)。

Event_Log_Rotation_Size

指定した場合、[イベント・ログのローテーション](#)が行われるまでにイベント・ログ・ファイルが到達できる最大サイズを決定します。この数値は、バイト (128000、接尾語なし)、キロバイト (256K)、またはメガバイト (8M) で定義できます。許容される最小値は 100K です。このパラメータは既定では指定されていません。つまり、イベント・ログ・ファイルを手動でクリアする必要があります。

Maximum_Logged_Request_Size

指定された HTTP 要求について、[ログに記録するデータの最大容量](#)。許容される最小値は 40K です。既定値は 256K です。

Retain_All_Log_Files

指定した場合、最新のものだけでなく、[イベント・ログのローテーション](#)で生成された[古いログ・ファイルをすべて保存するかどうか](#)を決定します。使用できる値は Enabled (すべて保持) または Disabled (1 つのみ保持) です。

SA_Exclude_File_Types

指定した場合、ステート認識アプリケーション (およびステートレス・アプリケーション) について、指定したタイプの[静的ファイルの非同期処理を保持](#)します。このパラメータは、ファイル・タイプ拡張子の空白で区切られたリストを受け入れます。

Document_Root

指定した場合、[Web サーバの Web ドキュメント・ルート・ディレクトリ](#)の完全な物理パスを提供します。このパラメータは、Microsoft ASP エンジンを使用してコンテンツをレンダリングする Web アプリケーションにのみ必要です。

ASP_Directory

指定した場合、Web ゲートウェイが Microsoft ASP のコンテンツを一時的に格納できるディレクトリの完全な物理パスを提供します。このパラメータは、Microsoft ASP エンジンを使用してコンテンツをレンダリングする Web アプリケーションにのみ必要です。

WS_Service_Status

Network Service Daemon (NSD) を使用する Web ゲートウェイ実装の場合、NSD の内部 HTTP サーバが、未処理の HTTP 要求に応答できるようにするかどうかを決定します。使用できる値は Enabled (有効) または Disabled (無効) です。既定値は Enabled です。

セキュリティ上の理由により、この機能は無効にすることをお勧めします。

NSD_Document_Root

NSD がスタンドアロン・サーバとして機能する特殊な状況で、[NSD の Web ドキュメント・ルート・ディレクトリ](#)への完全な物理パスを提供します。

Server_Error

内部エラーの発生時に、Web ゲートウェイで表示されるカスタム・ページのパスとファイル名。

Server_Busy

利用可能な接続がすべて使用中の場合に、Web ゲートウェイで表示されるカスタム・ページのパスとファイル名。

Server_Unavailable

InterSystems IRIS アプリケーション・サーバ (またはアプリケーション) が構成内で故意に無効にされている場合に、Web ゲートウェイで表示されるカスタム・ページのパスとファイル名。

Server_Timeout

要求がタイムアウトしたときに、Web ゲートウェイで表示されるカスタム・ページのパスとファイル名。

Connection_Closed

ユーザがステート認識セッションからログアウトしたときに、Web ゲートウェイで表示されるカスタム・ページのパスとファイル名。

Default_Server

アプリケーション・パスのアプリケーション・アクセス・プロファイルで宛先サーバが指定されていない場合に、Web ゲートウェイが要求を転送するサーバを指定します。既定では、このパラメータは自動生成された LOCAL サーバ・アクセス・プロファイルを指定します。

[<server>]

<server>という名前の[サーバ・アクセス・プロファイル](#)について説明します。CSP.ini ファイルには、Web ゲートウェイ構成内で定義されるサーバ・アクセス・プロファイルごとに 1 つの [<server>] セクションが含まれます。

詳細

Web ゲートウェイが接続する各 InterSystems IRIS アプリケーション・サーバは、[サーバ・アクセス・プロファイル](#)に対応しています。CSP.ini ファイルは、各サーバ・アクセス・プロファイルを CSP.ini ファイルのセクションとして保持します。サーバ・アクセス・プロファイルの各セクションはヘッダ行 [<server>] で始まります。<server> はサーバ・アクセス・プロファイルの名前です。例えば、irisserver1 という名前のサーバ・アクセス・プロファイルに対応するセクションは、行 [irisserver1] で始まります。

このページで説明するパラメータのほかに、以下のパラメータの値も指定できます。指定した値は、[SYSTEM] セクションで指定されている[既定値](#)をオーバーライドします。

使用可能なパラメータ

Ip_Address

InterSystems IRIS アプリケーション・サーバの DNS ホスト名または IP アドレス。

TCP_Port

この InterSystems IRIS インスタンスのスーパーサーバが着信 Web ゲートウェイ接続を待ち受ける TCP ポート番号。

Mirror_Aware

指定した場合、InterSystems IRIS アプリケーション・サーバをミラーリングされたデータベースにアクセスするミラー・プライマリとして識別します。有効にするには、値 1 を指定します。

Minimum_Server_Connections

クライアント間で接続を共有し始める前に、Web ゲートウェイがこの InterSystems IRIS アプリケーション・サーバに対して行う、[プロセス親和性のある接続の最小数](#)。既定値は 3 です。

Maximum_Server_Connections

Web ゲートウェイがこの InterSystems IRIS アプリケーション・サーバに対して行うことができる、[接続の最大数](#)。既定では、この値は指定されておらず、Web ゲートウェイの MAX_CONNECTIONS の値を継承します。

Maximum_Session_Connections

個々のセッションが同時に使用可能な、この InterSystems IRIS アプリケーション・サーバへの最大接続数。既定値は 3 です。

Connection_Security_Level

Web ゲートウェイとこの InterSystems IRIS アプリケーション・サーバ間の[接続を保護](#)する方法を示す数値。使用できる値は次のとおりです。

- ・ 0 – パスワード
- ・ 1 – Kerberos
- ・ 2 – Kerberos パケット整合性

- ・ 3 – Kerberos 暗号化
- ・ 4 – [SSL/TLS](#)

Username

Web ゲートウェイが InterSystems IRIS サーバへの接続を認証するために使用する必要のあるユーザ名。

Password

Web ゲートウェイが InterSystems IRIS アプリケーション・サーバへの接続を認証するために使用する必要のあるパスワード。

また、UNIX®/Linux/macOS システムでは、このパラメータでオペレーティング・システム・コマンドを中括弧 ({ }) で囲んで指定することもできます (例: Password={sh /tmp/PWretrieve.sh})。Web ゲートウェイは、Web ゲートウェイ管理ページでコマンドが[サーバ・アクセス・プロファイル](#)の一部として保存されるとき、または **CSP.ini** ファイルの [SYSTEM] セクションで **RELOAD=1 フラグ**が検出されたときにコマンドを実行します。コマンドの出力は、アプリケーション・サーバのパスワードとしてメモリに格納されます。

このパラメータの値は、ハッシュ値として **CSP.ini** ファイルに格納されます。

Product

アプリケーション・サーバが関連付けられているインターシステムズ製品を示す数値 (InterSystems IRIS)。使用できる値は次のとおりです。

- ・ 0 – InterSystems Caché®
- ・ 1 – InterSystems Ensemble®
- ・ 2 – InterSystems IRIS、InterSystems IRIS for Health、または HealthShare® 製品

Service_Principal_Name

Web ゲートウェイ接続の [Kerberos ベース認証](#)の実装内で、この InterSystems IRIS サーバを識別するサービス・プリンシパル名。

Keytable

[Kerberos ベース認証](#)を使用している場合、keytab ファイルの場所。

SSLCC_Protocol_Min

Web ゲートウェイと InterSystems IRIS アプリケーション・サーバが接続を保護するために使用できる、最小 SSL/TLS プロトコル・バージョンを示す数値。使用できる値は次のとおりです。

- ・ 4 – TLSv1.0
- ・ 8 – TLSv1.1
- ・ 16 – TLSv1.2
- ・ 32 – TLSv1.3 (サポートされている場合)

TLSv1.3 がサポートされている場合、既定値は 16 です。それ以外の場合、既定値は 8 です。

SSLCC_Protocol_Max

Web ゲートウェイと InterSystems IRIS アプリケーション・サーバが接続を保護するために使用できる、最大 SSL/TLS プロトコル・バージョンを示す数値。使用できる値は次のとおりです。

- ・ 4 – TLSv1.0
- ・ 8 – TLSv1.1
- ・ 16 – TLSv1.2
- ・ 32 – TLSv1.3 (サポートされている場合)

TLSv1.3 がサポートされている場合、既定値は 32 です。それ以外の場合、既定値は 16 です。

SSLCC_Verify_Peer

指定した場合、InterSystems IRIS アプリケーション・サーバに対して相手証明書認証が必要になります。有効にするには、値 1 を指定します。

SSLCC_Cipher_Suites

接続が TLSv1.2 以下で保護される場合に受け入れられる暗号スイートを指定します。既定値は ALL:!aNULL:!eNULL:!EXP:!SSLv2 です。

SSLCC_Cipher_Suites_1_3

接続が TLSv1.3 で保護される場合に受け入れられる暗号スイートを指定します。既定値は、TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256 です。

SSLCC_Certificate_File

Web ゲートウェイの SSL/TLS 証明書ファイルへのフル・パスです。証明書ファイルでサポートされているファイル形式は、InterSystems IRIS [TLS 構成](#)でサポートされている形式と同じです。

SSLCC_Certificate_Key_File

Web ゲートウェイの SSL/TLS 証明書に関連付けられた秘密鍵へのフル・パスです。証明書ファイルでサポートされているファイル形式は、InterSystems IRIS [TLS 構成](#)でサポートされている形式と同じです。

SSLCC_Key_Type

キーが対応する暗号化アルゴリズムを示す数値。使用できる値は次のとおりです。

- ・ 1 – DSA
- ・ 2 – RSA

SSLCC_Private_Key_Password

指定した場合、Web ゲートウェイの秘密鍵ファイルにアクセスするために必要なパスワード。

また、UNIX®/Linux/macOS システムでは、このパラメータでオペレーティング・システム・コマンドを中括弧で囲んで指定することもできます (例: SSLCC_Private_Key_Password={sh /tmp/tlsPWretrieve.sh})。Web ゲートウェイは、コマンドが Web ゲートウェイ管理ページで[サーバ・アクセス・プロファイル](#)の一部として保存されるとき、または CSP.ini ファイルの [SYSTEM] セクションに [RELOAD=1 フラグ](#)がある場合にコマンドを実行します。コマンドの出力は、秘密鍵パスワードとしてメモリに格納されます。

このパスワードは、ハッシュ値として CSP.ini ファイルに格納されます。

SSLCC_CA_Certificate_File

Web ゲートウェイの証明書用の認証機関 (CA) の証明書へのフル・パスです。証明書ファイルでサポートされているファイル形式は、InterSystems IRIS [TLS 構成](#)でサポートされている形式と同じです。

[SYSTEM_INDEX]

Web ゲートウェイ構成で定義されている各サーバ・アクセス・プロファイルのステータス (Enabled または Disabled) について説明します。CSP.ini ファイルには 1 つの [SYSTEM_INDEX] セクションのみが含まれます。

詳細

CSP.ini ファイルの [SYSTEM_INDEX] セクションでは、Web ゲートウェイが要求をアクティブにルーティングする Inter-Systems IRIS アプリケーション・サーバを指定します。このセクションの各行は、サーバ・アクセス・プロファイルの名前によってアプリケーション・サーバを識別し、そのサーバへのルーティングが Enabled か Disabled かを指定します。

例えば、irisserver1 および irisserver2 という名前の 2 つのアクティブなサーバ・アクセス・プロファイルで構成されている Web ゲートウェイの CSP.ini ファイルには、以下の行が含まれます。

```
[SYSTEM_INDEX]
irisserver1=Enabled
irisserver2=Enabled
```


[APP_PATH:<appPath>]

アプリケーション・パス <appPath> の [アプリケーション・アクセス・プロファイル](#) について説明します。CSP.ini ファイルには、Web ゲートウェイ構成内で定義されるアプリケーション・アクセス・プロファイルごとに 1 つの [APP_PATH:<appPath>] セクションが含まれます。

詳細

[アプリケーション・アクセス・プロファイル](#) は、Web ゲートウェイがアプリケーション・パスの要求をルーティングする方法を定義します。CSP.ini ファイルは、各アプリケーション・アクセス・プロファイルを CSP.ini ファイルのセクションとして保持します。各アプリケーション・アクセス・プロファイル・セクションは、ヘッダ行 [APP_PATH:<appPath>] で始まります。<appPath> はアプリケーション・パスです。例えば、irisserver1 という名前のサーバ・アクセス・プロファイルに対応するセクションは、行 [irisserver1] で始まります。

使用可能なパラメータ

Default_Server

代替サーバの前に、Web ゲートウェイがこのアプリケーション・パスの要求をルーティングする必要のある既定のサーバ・アクセス・プロファイルの名前。このサーバを Alternative_Server (通常は、Alternative_Server_0) として定義することも必要です。

Alternative_Servers

代替 InterSystems IRIS アプリケーション・サーバが定義されている場合、Web ゲートウェイが代替サーバに要求をルーティングする方法を示す数値。使用できる値は次のとおりです。

- ・ FailOver – あるアプリケーション・サーバから次に使用可能な代替サーバに、指定した順序で [フェイルオーバー](#) できるようにします
- ・ LoadBalancing – フェイルオーバーと共に、アプリケーション・サーバ間でラウンドロビン方式で [負荷分散](#) できるようにします
- ・ Disabled – フェイルオーバー機能と負荷分散機能を無効にします

Alternative_Server_<n> パラメータを使用して、アプリケーション・パスに使用するすべてのアプリケーション・サーバを定義する必要があります。

Alternative_Server_0、Alternative_Server_1、... Alternative_Server_<n>

Web ゲートウェイのルーティング動作内でアプリケーション・アクセス・プロファイルに対するサーバ・アクセス・プロファイルを指定し、それらの InterSystems IRIS アプリケーション・サーバがアプリケーション・アクセス・プロファイル内で有効か無効かを指定する文字列。 [フェイルオーバー](#) が有効になっている場合、Web ゲートウェイはアプリケーションの要求を、まず Alternative_Server_0 パラメータ、次に Alternative_Server_1 パラメータ、その次に Alternative_Server_2 パラメータなどによって指定されたアプリケーション・サーバにルーティングしようとしています。Web ゲートウェイは任意の数 (<n>) の代替サーバに対して、つまり、Alternative_Server_<n> パラメータまで、このフェイルオーバー動作を続けることができます。

各 Alternative_Server_<n> パラメータは、アプリケーション・アクセス・プロファイル内でアプリケーション・サーバのステータスを示すビットで始まり (有効の場合は 1、無効の場合は 0)、目的の InterSystems IRIS アプリケーション・サーバのサーバ・アクセス・プロファイルの名前で終わります。Web ゲートウェイがこのパラメータの値を更新する際、見やすくするために、任意の数の ~ 文字を使用して、ステータスとサーバ・アクセス・プロファイル名を区切ります。

例えば、アプリケーションのフェイルオーバー順序で 3 番目のアクティブな InterSystems IRIS アプリケーション・サーバを、`irisserv1` というサーバ・アクセス・プロファイルに対応するサーバに指定する場合は、以下の行をアプリケーション・アクセス・プロファイル・ブロックに追加します。

```
Alternative_Server_2=1~~~~~irisserv1
```

KeepAlive

Web ゲートウェイがこのアプリケーションに HTTP 接続動作を実装する方法を指定します。使用できる値は次のとおりです。

- ・ No Action – Web ゲートウェイでは、各要求の HTTP 応答ヘッダで接続の Keep-Alive ステータスを決定できます。これが既定値です。
- ・ Enabled
- ・ Disabled

Non_Parsed_Headers

Web ゲートウェイが HTTP 応答ヘッダをクライアントに直接ストリーム転送するか (Enabled)、解析とフィルタリングのためにホスト Web サーバに送信するか (Disabled) を指定します。既定値は Enabled です。

Response_Size_Notification

このアプリケーションの HTTP 応答に含まれるデータの量をクライアントに通知するときに Web ゲートウェイが使用する方法を指定します。使用できる値は次のとおりです。

- ・ Chunked Transfer Encoding and Content Length – 既定値
- ・ Chunked Transfer Encoding
- ・ Content-Length

Response_Size_Notification_Always

Web ゲートウェイがこのアプリケーションに関連付けられているすべての HTTP 応答の応答サイズ通知を送信するかどうかを指定します。使用できる値は Enabled (有効) または Disabled (無効) です。既定値は Disabled です。

GZIP_Compression

Web ゲートウェイがこのアプリケーションに返されるすべてのページを圧縮するかどうかを指定します。使用できる値は Enabled (有効) または Disabled (無効) です。既定値は Disabled です。

GZIP_Minimum_File_Size

Web ゲートウェイが GZIP 圧縮を呼び出す最小応答サイズ (バイト単位) を指定します。既定値は 500 です。

GZIP_Exclude_File_Types

Web ゲートウェイがこのアプリケーションの GZIP 圧縮から除外するファイル・タイプ。ファイル・タイプ拡張子の空白で区切られたリストとして指定します。既定値には、ネイティブに圧縮されたファイルが含まれます (jpeg gif ico png gz zip mp3 mp4 tiff)。

Extra_CGI_Env_Variables

このアプリケーションの各要求で、Web ゲートウェイが InterSystems IRIS アプリケーション・サーバに提供する必要のある追加の CGI 環境変数。Web ゲートウェイが提供する既定の変数セットの詳細は、“[Web ゲートウェイによって渡される CGI 環境変数](#)” を参照してください。

Proc_Class

Web ゲートウェイがこのアプリケーションの要求の処理に使用する必要のあるカスタム要求ハンドラの名前。

[APP_PATH_INDEX]

Web ゲートウェイ構成で定義されている各[アプリケーション・アクセス・プロファイル](#)のステータス (Enabled または Disabled) について説明します。**CSP.ini** ファイルには 1 つの [APP_PATH_INDEX] セクションのみが含まれます。

詳細

CSP.ini ファイルの [APP_PATH_INDEX] セクションでは、Web ゲートウェイが要求をアクティブにルーティングするアプリケーション・パスを指定します。このセクションの各行は、[アプリケーション・アクセス・プロファイル](#)を構成したアプリケーション・パスを識別し、そのアプリケーション・パスのルーティングが Enabled か Disabled かを指定します。

例えば、アプリケーション・パス / (root)、/csp、/irisservice1、/irisservice2、および /apps/criticalapp を提供するように構成された Web ゲートウェイの **CSP.ini** ファイルには、以下の行が含まれます。

```
[APP_PATH_INDEX]
/=Enabled
/csp=Enabled
/irisservice1
/irisservice2
/apps/criticalapp
```


A

NSD の使用法 (Windows)

ここでは、Microsoft Windows 上の InterSystems [Web ゲートウェイ](#)で Network Service Daemon (NSD) を使用する方法について説明します。これは典型的なインストールではありませんが、これが適切である場合があります。

A.1 NSD を使用する場合

Web サーバから独立して Web ゲートウェイを管理できるように、NSD を使用して Web サーバと Web ゲートウェイを分離する状況として 3 種類が考えられます。

- ・ Web サーバがその負荷を複数のサーバ・プロセスに分散する場合に、Web ゲートウェイのインスタンスを各 Web サーバ・プロセスにアタッチする。
- ・ CSP の占める割合が少ない、きわめて大規模な Web サーバ (例えば、[Web アプリケーション](#)のほかに、php、静的コンテンツ、.NET、.ASP アプリケーションなどを扱う Web サーバ) を運用する。
- ・ [Nginx](#) Web サーバを使用している。

A.2 NSD モジュールのインストール場所

Microsoft Windows で NSD モジュールを使用する場合、以下の 2 つのユーティリティをインストールします。

- ・ CSPnsd.exe
- ・ CSPnsdSv.exe

IIS のインストール環境では、これらは以下の場所にインストールされます。

C:\¥Inetpub¥CSPGateway¥nsd

Apache のインストール環境では、これらは以下の場所にインストールされます。

C:\¥Program Files¥Apache Group¥Apache¥WebGateway¥nsd

NSD をホーム・ディレクトリ **C:\¥Inetpub¥CSPGateway¥nsd** から実行します。NSD ベースの接続オプションの場合、[構成ファイル](#)と[ログ・ファイル](#)はこのディレクトリに書き込まれます。

A.3 NSD の操作

以下の手順で、NSD を開始します。

1. 以下のように、NSD のホーム・ディレクトリに変更します。

```
C:\Inetpub\CSPGateway>nsd
```

2. 以下を使用して、NSD を開始します。

```
CSPnsd
```

NSD が Windows サービス (CSPnsdSv.exe) として開始します。サービスとして登録すると、Windows サービス・マネージャから NSD 全体を管理できるようになります。

3. 以下のコマンドを発行して、NSD を終了します。

```
CSPnsd -stop
```

または、以下のように入力します。

```
CSPnsd
```

NSD の Windows サービスの状態が表示され、以下のいずれかの操作を実行できるようになります。

- ・ 実行中の NSD サービスを停止します。
- ・ 停止中の NSD サービスを再開します。
- ・ サービス・データベースから NSD サービスを削除します。

Windows サービス・マネージャを使用して、NSD を管理することもできます。NSD はサービス・マネージャで、以下の説明によって識別されます。

```
Cache Server Pages - Network Service Daemon
```

すべてのエラーは [Web ゲートウェイのイベント・ログ](#) で報告されます。

その他の起動オプション

1. ヘルプ情報を表示します。

```
CSPnsd -h
```

2. Windows サービスとは対照的に、コマンド・ウィンドウから対話形式で NSD を実行します。NSD のインスタンスを複数実行している場合は、この操作モードを使用する必要があります。

```
CSPnsd -v
```

A.3.1 代替 TCP ポートでの NSD の開始

既定では、NSD は TCP ポート 7038 で着信する要求を待ち受けます。以下のようにサービスを開始することでこれをオーバーライドできます。

```
CSPnsd -v [port_no]
```

または以下のようにします。

```
CSPnsd -v -p[port_no]
```


port_no は、指定する TCP ポート番号です。

起動時に、NSD は **CSPnsd.ini** ファイルを作成します。このファイルには、通常、以下の行が含まれます。

```
[SYSTEM]
Ip_Address=127.0.0.1
TCP_Port=7038
```

ここでは、クライアントは、Web サーバに格納された Web ゲートウェイ・モジュールまたは Web サーバに動的にリンクされる Web ゲートウェイ・モジュール、およびサーバから呼び出された CSP CGI モジュール（またはこのどちらかのモジュール）です。したがって、このファイルを削除、または移動することはできません。また、Web サーバ・プロセスがこのファイルを読み取り可能であることも重要です。Web サーバを操作する Windows ユーザを考慮し、適切な特権を設定してください。NSD クライアントは、Windows の PATH 変数で指定された場所 (**C:\Windows** など) でこのファイルを検索しようとします。したがって、Web サーバを起動する前に、**CSPnsd.ini** ファイルをこの場所に移動しておく必要があります。

NSD の複数のインスタンスが実行されている状況で、**CSPnsd.ini** ファイルに NSD ポート番号を格納することは不適切です。Apache サーバの場合、NSD の TCP ポート番号をクライアントに伝えるためのより適切なメカニズムがあります。具体的には、Apache 構成で以下の環境変数を設定して、ターゲットの NSD インストールのアドレスとポートを指定します。これらの環境変数に指定した値は、**CSPnsd.ini** ファイルの値よりも優先されます。

- ・ **CSP_NSD_NAME** – NSD の IP アドレスです。NSD がリモート・コンピュータで動作している場合のみ、このパラメータを使用します。
- ・ **CSP_NSD_PORT** – NSD の TCP ポート番号です。

A.3.1.1 例 1 : 2 つの Apache 仮想ホスト

2 つの Apache 仮想ホスト (123.123.1.1 および 123.123.1.2) の負荷を、2 つの独立した NSD インストール (TCP ポート 7038 および 7039 で待ち受けます) 間で分散します。

以下の指示文を、Apache 構成 (**httpd.conf**) に追加します。

```
<VirtualHost 123.123.1.1>
    ServerName 123.123.1.1
    SetEnv CSP_NSD_PORT 7038
</VirtualHost>
<VirtualHost 123.123.1.2>
    ServerName 123.123.1.2
    SetEnv CSP_NSD_PORT 7039
</VirtualHost>
```

A.3.1.2 例 2 : 2 つの Web アプリケーション

2 つの **Web アプリケーション** (/csp1 および /csp2) の負荷を、2 つの独立した NSD インストール (TCP ポート 7038 および 7039 で待ち受けます) 間で分散します。

1. 以下の指示文を、Apache 構成 (**httpd.conf**) に追加します。

```
<Location /csp1>
    SetEnv CSP_NSD_PORT 7038
</Location>
<Location /csp2>
    SetEnv CSP_NSD_PORT 7039
</Location>
```

2. Apache 構成の変更後は、Apache を再起動します。

NSD の複数のインスタンスを実行している場合は、各インスタンスを別々のディレクトリにインストールし、それぞれのインスタンスに専用の**構成ファイル**と**ログ・ファイル**を管理することをお勧めします。NSD の内部 HTTP サーバを使用すると、各インスタンスの Web ゲートウェイ管理ページに簡単にアクセスできます。以下に例を示します。

http://localhost:7038/csp/bin/Systems/Module.cwx

`http://localhost:7039/csp/bin/Systems/Module.cxw`

B

NSD の使用法 (UNIX®/Linux/macOS)

ここでは、UNIX®、Linux、または macOS 上の [Web ゲートウェイ](#) で Network Service Daemon (NSD) を使用方法について説明します。これは典型的なインストールではありませんが、これが適切である場合があります。

B.1 NSD を使用する場合

Web サーバから独立して Web ゲートウェイを管理できるように、NSD を使用して Web サーバと Web ゲートウェイを分離する状況として 3 種類が考えられます。

- ・ Web サーバがその負荷を複数のサーバ・プロセスに分散する場合に、Web ゲートウェイのインスタンスを各 Web サーバ・プロセスにアタッチする。
- ・ CSP の占める割合が少ない、きわめて大規模な Web サーバ (例えば、[Web アプリケーション](#)のほかに、php、静的コンテンツ、.NET、.ASP アプリケーションなどを扱う Web サーバ) を運用する。
- ・ [Nginx](#) Web サーバを使用している。

B.2 NSD モジュールのインストール場所

必要に応じて、NSD モジュールは **CSPnsd** となります。

このモジュールの既定の場所は以下のとおりです。

```
/opt/webgateway/bin
```

NSD は、そのホーム・ディレクトリ (上記参照) から実行されます。NSD ベースの接続オプションの場合、[構成ファイル](#)と[ログ・ファイル](#)はこのディレクトリに書き込まれます。

B.3 NSD の操作

NSD を実行するには、以下の手順に従います。

1. 以下のディレクトリに変更します。

```
/opt/webgateway/bin
```

2. 以下のコマンドを入力して、NSD を開始します。

```
./CSPnsd
```

バックグラウンドへ移動する前に、NSD は実行している構成を示すバナーを表示します。このバナーには、このサービスに割り当てられている TCP ポート番号 (既定ではポート 7038) が表示されます。

このコマンドで `-s` 修飾子を使用すると、起動時のすべてのメッセージが非表示になります。例えば、システムが起動したときに呼び出されたスクリプトから、NSD を開始するには以下を使用します。

```
/opt/webgateway/bin/CSPnsd -s
```

その他の一般的なスタートアップ・オプション：

- ヘルプ情報を表示します。

```
./CSPnsd -h
```

- NSD の動作を一時停止します。このコマンドは、停止信号 (SIGSTOP) を NSD のプロセスに送信します。

```
./CSPnsd -pause
```

- (一時停止の後)、NSD の動作を続行します。このコマンドは続行信号 (SIGCONT) を NSD のプロセスに送信します。

```
./CSPnsd -cont
```

- 他のユーザに NSD を実行する許可を与えます。NSD (CSPnsd) コンポーネントの管理者は、`CSPnsd -m=s` を使用して NSD を開始/停止する許可をグループまたはその他のユーザに与えることができます。ここで、`s` は起動オプションです。

`s` には以下のいずれかを指定できます。

- 現在のユーザには `u` (既定)
- 現在のグループには `g`
- その他には `o`
- すべてのユーザには `a` (`m=ugo`)

例：`CPSnsd -m=ug` は、NSD を実行する許可をグループ (管理者グループ) に与えます。このコマンドは、`-rw-rw---` の `CPSnsd.pid` 許可を与えます。

CSPnsd を停止するコマンドが発行されると、CSPnsd 親プロセスに以前と同様にシャットダウンするようシグナルを送信しようとします。サービスが別のユーザによって開始されたためにこれが不可能な場合は、CSPnsd.ini ファイルにフラグが書き込まれ、サービスはフラグを認識したときに自体を正常に終了します。このプロセスの完了には、最大 20 秒かかります。

NSD を終了するには、以下を入力します。

```
./CSPnsd -stop
```

または、以下を実行します。

```
kill -TERM `cat /opt/webgateway/bin/CSPnsd.pid`
```

これらのコマンドは、NSD を正しい順序でクローズします。つまり、InterSystems IRIS との間でオープンしているすべての接続を正しく終了し、終了前にすべてのシステム・リソースを解放します。kill -9 コマンドを使用して NSD を終了しないでください。

すべてのエラーは [Web ゲートウェイのイベント・ログ](#) で報告されます。

B.3.1 代替 TCP ポートでの NSD の開始

既定では、NSD は TCP ポート 7038 で着信する要求を待ち受けます。以下のようにサービスを開始することでこれをオーバーライドできます。port_no は、指定する TCP ポート番号です。

```
./CSPnsd [port_no]
```

または以下のようにします。

```
./CSPnsd -p=[port_no]
```

開始時に、NSD は以下のファイルを作成します。

```
/opt/webgateway/bin/CSPnsd.ini
```

このファイルには、通常、以下の行が記述されます。

```
[SYSTEM]
Ip_Address=127.0.0.1
TCP_Port=7038
```

ここでは、クライアントは、Web サーバに格納された Web ゲートウェイ・モジュールまたは Web サーバに動的にリンクされる Web ゲートウェイ・モジュール、およびサーバから呼び出された Web ゲートウェイ CGI モジュール（またはこのどちらかのモジュール）です。したがって、このファイルを削除、または移動することはできません。また、Web サーバ・プロセスがこのファイルを読み取り可能であることも重要です。Web サーバを操作する UNIX® ユーザ名を考慮して、適切な特権を設定してください。NSD クライアントは、以下の場所でこのファイルを検索しようとします。

```
/opt/webgateway/bin
```

```
/etc
```

NSD が別のディレクトリで動作している場合、**CSPnsd.ini** ファイルを上記のいずれかの場所に移動する必要があります。

NSD の複数のインスタンスが実行されている状況で、**CSPnsd.ini** ファイルに NSD ポート番号を格納することは不適切です。Apache サーバの場合、NSD の TCP ポート番号をクライアントに伝えるためのより適切なメカニズムがあります。具体的には、Apache 構成で以下の環境変数を設定して、ターゲットの NSD インストールのアドレスとポートを指定します。

- ・ **CSP_NSD_NAME** — NSD の IP アドレスです。NSD がリモート・コンピュータで動作している場合のみ、このパラメータを使用します。
- ・ **CSP_NSD_PORT** — NSD の TCP ポート番号です。

これらの環境変数に指定した値は、**CSPnsd.ini** ファイルの値よりも優先されます。

B.3.1.1 例 1 : 2 つの Apache 仮想ホスト

2 つの Apache 仮想ホスト (123.123.1.1 および 123.123.1.2) の負荷を、2 つの独立した NSD インストール (TCP ポート 7038 および 7039 で待ち受ける) 間で分散するには、Apache 構成 (**httpd.conf**) に以下の指示文を追加します。

```
<VirtualHost 123.123.1.1>
    ServerName 123.123.1.1
    SetEnv CSP_NSD_PORT 7038
</VirtualHost>
<VirtualHost 123.123.1.2>
    ServerName 123.123.1.2
    SetEnv CSP_NSD_PORT 7039
</VirtualHost>
```

B.3.1.2 例 2 : 2 つの Web アプリケーション

2 つの [Web アプリケーション](#) (`/csp1` および `/csp2`) の負荷を、2 つの独立した NSD インストール (TCP ポート 7038 および 7039 で待ち受ける) 間で分散するには、Apache 構成 (`httpd.conf`) に以下の指示文を追加します。

```
<Location /csp1>
    SetEnv CSP_NSD_PORT 7038
</Location>
<Location /csp2>
    SetEnv CSP_NSD_PORT 7039
</Location>
```

Apache 構成の変更後は、Apache を再起動します。

NSD の複数のインスタンスを実行している場合は、各インスタンスを別々のディレクトリにインストールし、それぞれのインスタンスに専用の[構成ファイル](#)と[ログ・ファイル](#)を管理することをお勧めします。NSD の内部 HTTP サーバを使用すると、各インスタンスの Web ゲートウェイ管理ページに簡単にアクセスできます。以下に例を示します。

```
http://localhost:7038/csp/bin/Systems/Module.cwx
```

```
http://localhost:7039/csp/bin/Systems/Module.cwx
```

B.3.1.3 複数の NSD プロセスへの負荷分散

既定では、NSD は 2 プロセス処理モード (親ワーカ 1 つと子ワーカ 1 つ) で動作します。

ただし、1 つの UNIX® プロセスが起動できるスレッドの数には上限があります。同時に発生している Web アプリケーションの負荷により、要求が使用可能なスレッドのキューを作ってしまった場合は、NSD で使用されるプロセスの数を増やすことを検討してください。

```
./CSPnsd -c=[no_processes]
```

ここで、`no_processes` は開始される子 (またはワーカ) プロセスの数です。

子プロセスの数を 1 に設定することには長所があるということにも注目してください。

```
./CSPnsd -c=1
```

このような状況では、NSD により、親プロセス 1 つと子ワーカ・プロセス 1 つの合計 2 プロセスが開始されます。‘`-c`’ 指示文を使用しているときに、親プロセスが存在すると、ワーカ・プロセスの 1 つで障害が発生している場合に親プロセスがその処理を引き継ぐことができるため、NSD の復元力が向上します。単一のマルチスレッド・アーキテクチャでは、NSD は必ずしも重大な内部エラー状態から回復できるわけではありません。

ワーカ・プロセスの数が 2 つ以上の場合は、ステート認識接続 ([保持モード 1](#)) を使用しないでください。

B.3.1.4 NSD に対する管理者権限の付与

NSD (`CSPnsd`) コンポーネントの管理者は、このサービスの開始/停止が許可されたユーザ (またはグループ) を一部制御することができます。

既定のシナリオでは、`CSPnsd` マスタ・プロセス ID (PID) ファイル (`CSPnsd`) は、サービスを開始したユーザのみがその後サービスを終了できるように作成されます。

例えば、管理者は、現在の UNIX® グループに属するすべてのユーザがサービスを管理できるように選択することができます。これは管理ユーザが属するグループです。

```
NSD start-up option: [-m=s]
Define the user(s) permitted to manage this service
where 's' is:
    'u' for the current user (the default),
    'g' for the current group,
    'o' for others,
    'a' for everyone (m=ugo),
```

例:

```
./CSPnsd -m=ug
```

これにより、現在のユーザ・グループ内の現在のユーザおよびその他すべてのユーザが NSD を管理できます。

NSD を停止するコマンドが発行されると、まず **CSPnsd** 親プロセスに以前と同様にシャットダウンするようシグナルを送信しようします。サービスが別のユーザによって開始されたためにこれが不可能な場合は、**CSPnsd.ini** ファイルにフラグが書き込まれ、サービスはこのフラグを認識したときに自体を正常に終了します。このプロセスの完了には、最大 20 秒かかります。

C

Apache の代替オプション (UNIX®/Linux/macOS)

ここでは、UNIX®、Linux、および macOS の InterSystems Web ゲートウェイで利用できる追加 Apache 構成について説明します (ロック・ダウン Apache 環境については別途説明します)。これらのすべての構成を開始するには、最初のセクションをご覧ください。次に、それぞれの構成に該当するセクションの指示に従います。

C.1 インストール場所 (すべての特殊オプション)

このセクションでは、Web ゲートウェイ・ファイルと CSP 静的ファイルのディレクトリの場所について説明します。

1. NSD モジュールは以下のとおりです。

CSPnsd

このモジュールの既定の場所は以下のとおりです。

/opt/webgateway/bin

NSD は、NSD のホーム・ディレクトリ /opt/webgateway/bin から実行する必要があります。CSP.ini と CSP.log は、このディレクトリに書き込まれます。

InterSystems IRIS® のアップグレード時に既存のゲートウェイに影響を与えないように、インストール時にこれらのモジュールは共通の場所 /opt/webgateway/bin に配置されます。この場所は、特定の InterSystems IRIS インスタンスに関するものではありません。

2. CGI およびその他のダイナミック・リンク・モジュール :

- ・ CSPcgi (実行時モジュール)
- ・ nph-CSPcgi (CSPcgi のコピー)
- ・ CSPcgiSys (システム管理モジュール)
- ・ nph-CSPcgiSys (CSPcgiSys のコピー)
- ・ mod_csp24.so (Apache バージョン 2.4.x – 提供される場合は、DSO として機能する Apache モジュール)

InterSystems IRIS のアップグレード時に既存のゲートウェイのインストールに影響を与えないように、インストールの際にこれらのモジュールは以下の共通の場所に配置されます。この場所は、特定の InterSystems IRIS インスタンスに関するものではありません。

```
/usr/cspgateway/bin
```

元の場所 (install-dir/csp/bin) は、特定の InterSystems IRIS インスタンスの管理ポータルを提供に必要な Web ゲートウェイ・コンポーネントの保持に使用されます。

Sys が追加されたモジュールは、Web ゲートウェイ管理ページにアクセスします。実行時モジュール (**Sys** が追加されていないモジュール) は Web ゲートウェイ管理ページにアクセスできません。

3. ハイパーイベントのコンポーネントの既定の場所は以下のとおりです。

- ・ CSPBroker.js
- ・ CSPxmlhttp.js

CSP サンプルおよび管理ポータルに必要なその他の静的リソース (イメージ・ファイルなど) は以下のとおりです。

```
install-dir¥csp¥broker
```

C.1.1 Apache API モジュールを使用する場合の要件 (推奨オプションおよび代替オプション 1)

[推奨オプション](#)または特殊オプション 1 ([代替オプション 1](#) : NSD を使用した Apache API モジュール (mod_csp24.so)) のいずれかの指示に従う前に、それぞれの Apache ビルドに共有オブジェクト管理のためのモジュール (mod_so) が組み込まれていることを確認します。この確認を行うには、以下のコマンドを実行します。その結果、Apache で現在使用可能なモジュールが一覧表示されます。

```
httpd -l
```

表示されるモジュール・リストには、共有オブジェクト・モジュール (mod_so) が含まれているはずです。一般的なモジュール・リストは以下のとおりです (mod_so が含まれています)。

```
Compiled in modules:
  core.c
  mod_access.c
  mod_auth.c
  mod_include.c
  mod_log_config.c
  mod_env.c
  mod_setenvif.c
  prefork.c
  http_core.c
  mod_mime.c
  mod_status.c
  mod_autoindex.c
  mod_asis.c
  mod_cgi.c
  mod_negotiation.c
  mod_dir.c
  mod_imap.c
  mod_actions.c
  mod_userdir.c
  mod_alias.c
  mod_so.c
```

Apache インストールのリストに **mod_so** が含まれていない場合は、Apache ドキュメントを参照し、Apache を再構築してこのモジュールを追加するための手順に従ってください。

C.2 代替オプション 1 : NSD を使用した Apache API モジュール (mod_csp24.so)

使用するディストリビューションに、事前に構築された共有オブジェクトとして CSP モジュール (mod_csp24.so) が付属している場合は、“**実行時の構成**” から開始してください。付属のソース・ファイル `mod_csp.c` から共有オブジェクトを構築する場合は、以下の“**方法 1**”または“**方法 2**”を選択してください。方法 1 をお勧めします。

共有オブジェクトの生成については、ご使用の Apache のマニュアルとあわせて、以下の説明もお読みください。以下の説明は、Apache のルート・ディレクトリが `/usr/apache` であることを前提としています。実際は、多くの場合、このディレクトリ名に Apache のバージョン番号が追加されます。

C.2.1 方法 1 : apxs (APache eXtenSion) ツールを使用した、共有オブジェクトとしての CSP モジュールのビルド

以下のコマンドを実行すると、Apache 拡張ツール `apxs` を使用して、Apache の `/modules` ディレクトリに共有ライブラリ `mod_csp24.so` がビルドおよびインストールされます。モジュールをロードする指示文が、Apache 構成ファイル `/conf/httpd.conf` に追加されます。

```
apxs -c -o mod_csp24.so mod_csp.c
```

生成された共有オブジェクト (mod_csp24.so) をディレクトリ `/opt/webgateway/bin` にコピーします。

C.2.2 方法 2 : 共有オブジェクトとしての CSP モジュールの手動ビルド

CSP モジュールを共有オブジェクトとして手動でビルドするには、以下の手順を実行します。

1. モジュール・ソース・ファイル `mod_csp.c` をディレクトリ `/usr/apache/src/modules/extra` にインストールします。
2. `/usr/apache/src` ディレクトリに戻り、`Configuration` ファイルを編集します。このファイルの末尾に近いところから、以下の行を見つけてください。

```
# AddModule modules/example/mod_example.o
```

この行の後に、以下の行を追加してください。

```
ShareModule modules/extra/mod_csp24.so
```

3. 以下のコマンドを実行して、構築プロセスを構成します。

```
./Configure
```

4. 以下のコマンドを実行して、共有オブジェクトを構築します。

```
make
```

共有オブジェクト `mod_csp24.so` を `/usr/apache/src/modules/extra` に作成します。

注釈 `apxs` ツールの詳細は、Apache ドキュメント (<https://httpd.apache.org/docs/2.4/programs/apxs.html>) を参照してください。

C.2.3 実行時の構成

Apache 構成ファイル **httpd.conf** を編集します。標準の Apache ディストリビューションでは、このファイルは以下の場所に配置されています。

```
/usr/apache/conf
```

Red Hat Linux の場合、**httpd.conf** の実行時バージョンは以下の場所に配置されています。

```
/etc/httpd/conf
```

拡張子 **.csp**、**.cls**、または **.zen** を含む要求されたファイルに対して **CSP エンジン** を呼び出す場合は、以下のセクションを **httpd.conf** の最後に追加します。

```
LoadModule csp_module /opt/webgateway/bin/mod_csp24.so
CSPFileTypes csp cls zen cxw
Alias /csp/ /opt/webgateway/csp/
<Directory "/opt/webgateway/csp">
    AllowOverride None
    Options MultiViews FollowSymLinks ExecCGI
    Require all granted
    <FilesMatch "\.(log|ini|pid|exe)$">
        Require all denied
    </FilesMatch>
    <Files CSPnsd>
        Require all denied
    </Files>
</Directory>
ScriptAlias /csp-bin/ "/opt/webgateway/bin/"
ScriptAliasMatch /csp/bin/Systems/Module.cwx "/opt/webgateway/bin/nph-CSPcgiSys"
ScriptAliasMatch /csp/bin/RunTime/Module.cwx "/opt/webgateway/bin/nph-CSPcgi"
<Directory "/opt/webgateway/bin/">
    AllowOverride None
    Options None
    Require all granted
</Directory>
```

httpd.conf の変更後に、Apache を再起動します。

C.2.3.1 接続プーリングの制御

接続プールのサイズは、**http.conf** で指定されている以下の Apache 構成パラメータで制御できます。

```
CSPMaxPooledNSDConnections <no>
```

このパラメータが存在しない場合は、既定値の 32 が内部的に使用されます。これは事実上、以下と同じです。

```
CSPMaxPooledNSDConnections 32
```

接続プーリングをオフにするには、以下のようにしてこのパラメータをゼロに設定します。

```
CSPMaxPooledNSDConnections 0
```

何らかの理由により、従来の操作モード (非対称モード) の使用 (これにより、Web ゲートウェイ側の接続が閉じられ、応答の送信終了が通知される) が必要になった場合は、このパラメータをマイナス 1 に設定します。

```
CSPMaxPooledNSDConnections -1
```

C.2.3.2 Apache API および NSD を使用した Web ゲートウェイの運用および管理

この接続オプションは、Web ゲートウェイの Network Service Daemon (NSD) に依存します。

1. “**NSD の操作**” の説明に従って、CSP NSD を起動します。
2. Apache の構成 (**httpd.conf**) を変更した後、その Apache を再起動する必要があります。

Apache と NSD を起動する順序は重要ではありません。

3. Web ゲートウェイ管理ページにアクセスするには、ブラウザに以下の URL を入力します。

```
http://<hostname>:<port_no>/csp-bin/nph-CSPcgiSys
```

であることを通知するエラー・メッセージが表示された場合は、“[追加クライアント・アドレスからのアクセスの有効化](#)”を参照してください。

C.3 代替オプション 2 : NSD を使用した CGI モジュール (nph-CSPcgi)

インターシステムズのファイル・タイプを認識し、それらを Web ゲートウェイに渡して処理するように Web サーバを構成します。

Web サーバ構成ファイル (`httpd.conf`) は、以下のディレクトリにあります。

```
/usr/apache/conf
```

Red Hat Linux の場合、`httpd.conf` の実行時バージョンは以下の場所に配置されています。

```
/etc/httpd/conf
```

以下のセクションを、`httpd.conf` の最後に追加します。

```
<LocationMatch "/*\.[Cc][Ss][Pp]|[Cc][Ll][Ss][Zz][En][Nn])$">
    AllowOverride None
    Options FollowSymLinks ExecCGI
    Require all granted
</LocationMatch>
ScriptAliasMatch /csp/bin/Systems/Module.cwx "/opt/webgateway/bin/nph-CSPcgiSys"
ScriptAliasMatch /csp/bin/RunTime/Module.cwx "/opt/webgateway/bin/nph-CSPcgi"
ScriptAliasMatch /*\.[Cc][Ss][Pp]|[Cc][Ll][Ss])$ "/opt/webgateway/bin/nph-CSPcgi"
Alias /csp/ instance-installation-directory
<Directory "instance-installation-directory">
    AllowOverride None
    Options MultiViews FollowSymLinks ExecCGI
    Require all granted
    <FilesMatch "\.(log|ini|pid|exe)$">
        Require all denied
    </FilesMatch>
    <Files CSPnsd>
        Require all denied
    </Files>
</Directory>
ScriptAlias /csp-bin/ "/opt/webgateway/bin/"
<Directory "/opt/webgateway/bin/">
    AllowOverride None
    Options None
    Require all granted
</Directory>
```

上記の構成ブロックは、Apache 環境で利用できる正規表現 (regex) プロセッサに依存しています。このケースに相当しない場合、結果として CSP ファイルは提供されません (“ ” というエラーが返されます)。この状況を修復するには、CSP ファイル拡張子で連結を行う代わりに、使用している [Web アプリケーション](#) の仮想のルート位置と、

CGI モジュールを連結させます。例えば、使用している Web アプリケーションが **/csp** に含まれているとします。CSP CGI モジュールを **/csp** にあるファイルに関連付けるには、以下の手順に従います。

```
<LocationMatch "/*\.([Cc][Ss][Pp]|[Cc][Ll][Ss]|[Zz][En][Nn])$" >
    AllowOverride None
    Options FollowSymLinks ExecCGI
    Require all granted
</LocationMatch>
ScriptAliasMatch /*\.([Cc][Ss][Pp]|[Cc][Ll][Ss])$ "/opt/webgateway/bin/nph-CSPcgi"
```

以下のブロックと置換します。

```
<Location "/csp">
    AllowOverride None
    Options FollowSymLinks ExecCGI
    Require all granted
</Location>
ScriptAlias /csp "/opt/webgateway/bin/nph-CSPcgi"
```

これらの指示文は、次のフォームの URL に対して作用します。[<baseURL>](#) はインスタンスのベース URL です。

```
http://<baseURL>/csp/*.csp
```

この構成ブロックを、他のルート位置で繰り返します。例えば、フォームの URL の **/myapps** に対して同様の処理を繰り返します。

```
http://<baseURL>/myapps/*.csp
```

regex の問題を回避するには、CSP MIME タイプと組み合わせて **Action** 指示文を使用する方法もあります。ただし、**Action** は本質的にコンテンツに対するフィルタ技術なので、InterSystems IRIS サーバが別のコンピュータにインストールされている場合でも、CSP ファイルが物理的に Web サーバ・ホストに存在している必要があります。この方法を使用する場合は、最初に新規の MIME タイプを Apache の **mime.types** ファイルの最後に追加し、それを CSP コンテンツを表すファイル・タイプと連結させます。**mime.types** ファイルは、**httpd.conf** ファイルと同じディレクトリにあります。

```
text/csp                csp cls
```

次に、**httpd.conf** の CGI 構成ブロックの最後に、**Action** 指示文を追加します。

```
Alias /csp/ /opt/webgateway/csp/
<Directory "/opt/webgateway/csp">
    AllowOverride None
    Options MultiViews FollowSymLinks ExecCGI
    Require all granted
<Files CSPnsd>
    Require all denied
</Files>
<Files CSP.ini>
    Require all denied
</Files>
<Files CSP.log>
    Require all denied
</Files>
<Files CSPnsd.ini>
    Require all denied
</Files>
<Files CSPnsd.pid>
    Require all denied
</Files>
</Directory>
ScriptAlias /csp-bin/ "/opt/webgateway/bin/"
<Directory "/opt/webgateway/bin/">
    AllowOverride None
    Options None
    Require all granted
</Directory>
Action text/csp "/csp-bin/nph-CSPcgi"
```

httpd.conf の変更後に、Apache を再起動します。

CGI はオープン・スタンダードなので、CSP の CGI モジュールはすべての Web サーバで作動します。

C.3.1 CGI および NSD を使用した Web ゲートウェイの運用および管理

この接続オプションは、Web ゲートウェイの Network Service Daemon (NSD) に依存します。

1. “NSD の操作” の説明に従って、CSP NSD を起動します。
2. Apache の構成 (httpd.conf) を変更した後、その Apache を再起動する必要があります。
Apache と NSD を起動する順序は重要ではありません。
3. Web ゲートウェイ管理ページにアクセスするには、ブラウザに以下のいずれかの URL を入力します。

```
http://<hostname>:<port_no>/csp/bin/Systems/Module.cwx
http://<hostname>:<port_no>/csp-bin/nph-CSPcgiSys
```

であることを通知するエラー・メッセージが表示された場合は、“[追加クライアント・アドレスからのアクセスの有効化](#)” を参照してください。

C.4 代替オプション 3 : NSD を使用した組み込み Apache API モジュール (mod_csp.c)

さらに複雑なこのオプションの設定を開始する前に留意する点があります。それは、現在のほとんどの UNIX® システムでは、共有オブジェクトとして実行時にモジュールをリンクする方法 (オプション 1) に対して、スタティック・リンクを使用する方法が持つパフォーマンス上の優位性はほとんどないという点です。

ご使用の Apache のマニュアルとあわせて、これらの説明をよく読んでください。

C.4.1 CSP モジュール・ソース・コードを含めた Apache のビルド

この手順の詳細は、以下の Apache ドキュメントを参照してください。

<http://httpd.apache.org/>

C.4.2 生成した Apache バイナリの確認

以下のコマンドを実行して、CSP モジュールが Apache コアに正常に追加されたことを確認します (このコマンドを実行すると、現在 Apache に組み込まれているすべてのモジュールが表示されます)。

```
./httpd -l
```

以下に例を示します。

```
Compiled in modules:
  core.c
  mod_access.c
  mod_auth.c
  mod_include.c
  mod_log_config.c
  mod_env.c
  mod_setenvif.c
  prefork.c
  http_core.c
  mod_mime.c
  mod_status.c
  mod_autoindex.c
  mod_asis.c
  mod_cgi.c
  mod_negotiation.c
```

```

mod_dir.c
mod_imap.c
mod_actions.c
mod_userdir.c
mod_alias.c
mod_csp.c

```

C.4.3 実行時の構成

Apache 構成ファイル **httpd.conf** を編集します。標準の Apache では、このファイルは以下の場所に配置されています。

```
/usr/apache/conf
```

Red Hat Linux の場合、**httpd.conf** の実行時バージョンは以下の場所に配置されています。

```
/etc/httpd/conf
```

拡張子 **.csp**、**.cls**、または **.zen** を含む要求されたファイルに対して **CSP エンジン** を呼び出す場合は、以下のセクションを **httpd.conf** の最後に追加します。

```

CSPFileTypes csp cls zen cxw
ScriptAliasMatch /csp/bin/Systems/Module.cwx "/opt/webgateway/bin/nph-CSPcgiSys"
ScriptAliasMatch /csp/bin/RunTime/Module.cwx "/opt/webgateway/bin/nph-CSPcgi"
Alias /csp/ /opt/webgateway/csp/
<Directory "/opt/webgateway/csp">
    AllowOverride None
    Options MultiViews FollowSymLinks ExecCGI
    Require all granted
    <FilesMatch "\.(log|ini|pid|exe)$">
        Require all denied
    </FilesMatch>
    <Files CSPnsd>
        Require all denied
    </Files>
</Directory>
ScriptAlias /csp-bin/ "/opt/webgateway/bin/"
<Directory "/opt/webgateway/bin/">
    AllowOverride None
    Options None
    Require all granted
</Directory>

```

Apache に対するすべての要求は、事前に定義されたシーケンス内で実行されるモジュールのセットにより提供されます。CSP モジュールは最初に実行されるモジュールの 1 つです。ただし、その定義が構成ファイルの最後の部分に追加されていることが条件となります。

httpd.conf の変更後に、Apache を再起動します。

C.4.4 Apache API および NSD を使用した Web ゲートウェイの運用および管理

この接続オプションは、Web ゲートウェイの Network Service Daemon (NSD) に依存します。

1. “**NSD の操作**” の説明に従って、CSP NSD を起動します。
2. Apache の構成 (**httpd.conf**) を変更した後、その Apache を再起動する必要があります。
Apache と NSD を起動する順序は重要ではありません。
3. Web ゲートウェイ管理ページにアクセスするには、ブラウザで以下の場所のいずれかを指定します。

```

http://<hostname>:<port_no>/csp/bin/Systems/Module.cwx
http://<hostname>:<port_no>/csp-bin/nph-CSPcgiSys

```

であることを通知するエラー・メッセージが表示された場合は、“**追加クライアント・アドレスからのアクセスの有効化**” を参照してください。

D

ロック・ダウン Apache インストールへの Web ゲートウェイの追加 (UNIX®/Linux/macOS)

場合によっては (特に RHEL システム上の Security-Enhanced Linux (SELinux) で顕著ですが)、[Apache](#) Web サーバがロック・ダウンされ、追加構成なしでは Apache ファイル・システムの外部のファイルにアクセスできないことがあります。この追加構成がない場合、[Web ゲートウェイ](#)接続を試行すると HTTP 403 Forbidden エラーが表示されます。

インストーラが InterSystems IRIS インスタンスまたはスタンドアロン Web ゲートウェイに [Web サーバ接続を自動構成](#)した場合、インストーラは SELinux システムで Web ゲートウェイ接続を有効にするための追加構成手順を既に実行しています。このページでは、この追加構成を手動で実行するための手順を示します。

2 つのソリューションを利用できます。

- ・ Web ゲートウェイのホーム・ディレクトリのセキュリティ・コンテキストを変更して、Apache がこの場所に保存されているファイルにアクセスできるようにします。多くの場合、これが最も簡単なソリューションです。
- ・ Web ゲートウェイのホーム・ディレクトリを Apache ルート・ファイル・システムの下場所 (SELinux の設定では、Apache がアクセスできるように事前構成されています) に移動します。

D.1 Web ゲートウェイ・ファイルのセキュリティ・コンテキストの変更

まず、Web ゲートウェイのホーム・ディレクトリに対する SELinux のセキュリティ・コンテキストを変更します。それには、以下の手順を実行します。

ここでは、例として Web ゲートウェイのホーム・ディレクトリ `/opt/webgateway/bin`、ポート 1972 で待ち受ける InterSystems IRIS スーパーサーバ、および `/usr/iris/` にインストールされている InterSystems IRIS を使用します。

`chcon` コマンドでファイル・コンテキストを設定します。変更は直ちに有効になります。

```
sudo chcon -R -t httpd_sys_content_t /usr/iris/csp
sudo chcon -R -t httpd_sys_rw_content_t /opt/webgateway/conf/CSP.ini
sudo chcon -R -t httpd_sys_rw_content_t /opt/webgateway/conf/CSPRT.ini
sudo chcon -R -t httpd_sys_rw_content_t /opt/webgateway/logs/CSP.log
sudo chcon -t httpd_modules_t /opt/webgateway/bin/CSPa2.so
sudo chcon -t httpd_modules_t /opt/webgateway/bin/CSPa2Sys.so
sudo chcon -t httpd_modules_t /opt/webgateway/bin/CSPa22.so
sudo chcon -t httpd_modules_t /opt/webgateway/bin/CSPa22Sys.so
sudo chcon -t httpd_modules_t /opt/webgateway/bin/CSPa24.so
sudo chcon -t httpd_modules_t /opt/webgateway/bin/CSPa24Sys.so
```

ただし、**chcon** コマンドで行った変更は、次回ラベルを再適用すると失われます。このため、**chcon** のほかに **semanage** の **fcontext** 機能を使用する必要があります。以下のコマンドを使用すると、SELinux が強制されているときに、Web ゲートウェイを実行できます。

```
semanage fcontext -a -t lib_t "/opt/webgateway/bin/(.*\.so)?" 2> /dev/null
semanage fcontext -a -t httpd_sys_rw_content_t "/opt/webgateway/bin/temp/(.*)?" 2> /dev/null
semanage fcontext -a -t httpd_sys_rw_content_t "/opt/webgateway/bin/CSP/(.*\.ini)?" 2> /dev/null
semanage fcontext -a -t httpd_sys_rw_content_t "/opt/webgateway/bin/CSP.log" 2> /dev/null
restorecon -vr /opt/webgateway/bin > /dev/null
```

次に、以下のコマンドを使用します。スーパーサーバ・ポートのコンテキストを正しく設定することが非常に重要です。そうしないと、Web ゲートウェイはポートにアクセスできず、“サーバが利用可能ではありません” というエラーが返されます。

```
sudo /usr/sbin/semanage fcontext -a -t httpd_sys_content_t
"/usr/iris/csp(/.)?"
sudo /usr/sbin/semanage fcontext -a -t httpd_sys_rw_content_t
"/opt/webgateway/conf/CSP.ini"
sudo /usr/sbin/semanage fcontext -a -t httpd_sys_rw_content_t
"/opt/webgateway/conf/CSPRT.ini"
sudo /usr/sbin/semanage fcontext -a -t httpd_sys_rw_content_t
"/opt/webgateway/logs/CSP.log"
sudo /usr/sbin/semanage fcontext -a -t httpd_modules_t
"/opt/webgateway/bin/CSPa2.so"
sudo /usr/sbin/semanage fcontext -a -t httpd_modules_t
"/opt/webgateway/bin/CSPa2Sys.so"
sudo /usr/sbin/semanage fcontext -a -t httpd_modules_t
"/opt/webgateway/bin/CSPa22.so"
sudo /usr/sbin/semanage fcontext -a -t httpd_modules_t
"/opt/webgateway/bin/CSPa22Sys.so"
sudo /usr/sbin/semanage fcontext -a -t httpd_modules_t
"/opt/webgateway/bin/CSPa24.so"
sudo /usr/sbin/semanage fcontext -a -t httpd_modules_t
"/opt/webgateway/bin/CSPa24Sys.so"
sudo /usr/sbin/semanage port -a -t http_port_t -p tcp 51773
```

これらは、Web ゲートウェイ (Apache サーバをホストするコンテキストで動作している) にホーム・ディレクトリ内のファイルへのアクセス権を付与する基本的な手順です。

D.2 Web ゲートウェイ・ディレクトリの移動

Apache で提供されている事前構成されたディレクトリで機能するように Web ゲートウェイを構成する方法もあります。これは、上記の方法を利用できない場合に使用すべき方法です。以下のコマンドは、Apache が **/usr/apache** にインストールされていることを前提としています。

- CGI モジュールを **/usr/apache/cgi-bin/** にコピーする必要があります

```
cp /usr/iris/csp/bin/*cgi* /usr/apache/cgi-bin/
```

- API モジュールを **/usr/apache/modules/** にコピーする必要があります

```
cp /usr/iris/csp/bin/*.so /usr/apache/modules/
```

- 静的ファイルを **/usr/apache/htdocs/** の下層の場所にコピーする必要があります

```
cp /usr/iris/csp/samples/* /usr/apache/htdocs/csp/samples/
cp /usr/iris/csp/broker/* /usr/apache/htdocs/csp/broker/
cp /usr/iris/csp/sys/* /usr/apache/htdocs/csp/sys/
```

さらに、上記の場所の下にあるサブディレクトリをコピーします。

Web ゲートウェイのインストールを移動した後、Apache 構成で指定したパスを適切に変更します。

以下のセクションでは、Web ゲートウェイを構成するための Apache 構成指示文の例を導入オプションごとに示します。これらの例では、Web ゲートウェイをグローバルに呼び出すように Apache サーバを構成します（ただし、InterSystems ファイル・タイプに対してのみ）。構成指示文は、Web ゲートウェイを呼び出して、組織のニーズに応じて要求を処理する必要があります。

D.2.1 推奨オプション : Apache API モジュール (CSPa24.so)

```
LoadModule cspsys_module_sa /usr/apache/modules/CSPa24.so
CSPSYSModulePath /usr/apache/modules/
CSPFileTypes csp cls zen cxw
```

D.2.2 代替オプション 1 : NSD を使用した Apache API モジュール (mod_csp.so)

```
LoadModule csp_module /usr/apache/modules/mod_csp.so
CSPFileTypes csp cls zen cxw
ScriptAliasMatch /csp/bin/Systems/Module.cwx "/usr/apache/cgi-bin/nph-CSPcgiSys"
ScriptAliasMatch /csp/bin/RunTime/Module.cwx "/usr/apache/cgi-bin/nph-CSPcgi"
```

D.2.3 代替オプション 2 : NSD を使用した CGI モジュール (nph-CSPcgi)

```
<LocationMatch "/*\.[Cc][Ss][Pp]|[Cc][Ll][Ss]|[Zz][En][Nn])$" >
AllowOverride None
Options FollowSymLinks ExecCGI
Require all granted
</LocationMatch>
ScriptAliasMatch /csp/bin/Systems/Module.cwx "/usr/apache/cgi-bin/nph-CSPcgiSys"
ScriptAliasMatch /csp/bin/RunTime/Module.cwx "/usr/apache/cgi-bin/nph-CSPcgi"
ScriptAliasMatch /*\.[Cc][Ss][Pp]|[Cc][Ll][Ss])$ "/usr/apache/cgi-bin/nph-CSPcgi"
```

D.2.4 代替オプション 3 : NSD を使用した組み込み Apache API モジュール (mod_csp.c)

```
CSPFileTypes csp cls zen cxw
ScriptAliasMatch /csp/bin/Systems/Module.cwx "/usr/apache/cgi-bin/nph-CSPcgiSys"
ScriptAliasMatch /csp/bin/RunTime/Module.cwx "/usr/apache/cgi-bin/nph-CSPcgi"
```


E

IIS 7 以降の代替オプション (Windows)

このページには、InterSystems [Web ゲートウェイ](#)で使用する [Microsoft IIS](#) を構成するための特殊オプションの構成手順が記載されています。これらのオプションについては、以下のとおりです。

1. 推奨手順の説明に従って、以下の手順を実行します。
 - a. [Web ゲートウェイ・コンポーネントに対する許可の設定](#)
 - b. アプリケーションの [Web アプリケーション・パスの構成](#)
 - c. [/bin](#)
2. “ISAPI サービスおよび CGI サービスのインストール” の説明のとおり、ISAPI サービスおよび CGI サービスをインストールします。
3. 次のセクションのいずれかの手順を使用します。
 - ・ 代替オプション 1 : ISAPI モジュールの使用 (CSPms*.dll)
 - ・ 代替オプション 2 : NSD を使用したネイティブ・モジュールの使用 (CSPcms.dll)
 - ・ 代替オプション 3 : NSD を使用した ISAPI モジュールの使用 (CSPcms.dll)
 - ・ 代替オプション 4 : NSD を使用した CGI モジュールの使用 (nph-CSPcgi*.exe)

E.1 ISAPI サービスおよび CGI サービスのインストール

既定の設定では、IIS 7 は **ISAPI 拡張**、**ISAPI フィルタ**、または **CGI モジュール**を実行しません。IIS 7 のすべての特殊オプションでは、これらのサービスをインストールする必要があります。

ISAPI 拡張サービスがインストールされていれば、Web ゲートウェイのすべてのバージョンが IIS 7 と組み合わせて動作することに注意してください。

これらの従来のサービスは、Windows のコントロール・パネルを使用してインストールします。

1. Windows のコントロール・パネルを開きます。
2. [プログラムと機能] を選択し、[Windows の機能の有効化または無効化] を選択します。
3. [インターネット インフォメーション サービス] に移動し、[World Wide Web サービス] および [アプリケーション開発機能] を展開します。

[ISAPI 拡張機能] を選択します。また、ISAPI フィルタと CGI が必要な場合は、これらのサービスを選択します。[OK] をクリックします。

4. Windows の [コントロール パネル] で [管理ツール]、[インターネット インフォメーション サービス (IIS) マネージャ] の順に開きます。
5. 左側のパネルで、[MACHINE_NAME] ([machine_name]¥[user_name]) をハイライト表示します。
6. 中央のパネルで、[モジュール] アイコンをダブルクリックします
7. 右側のパネルで、[ネイティブ モジュールの追加] を選択します。
8. 左側のパネルで、最上位レベルを展開し、[Web サイト]、[既定の Web サイト] の順に展開します。

```
[MACHINE_NAME] ([machine_name]\[user_name])
    Web Sites
        Default Web Site
```

9. 中央のパネルで、[ハンドラ マッピング] をダブルクリックします。
10. 中央のパネルで、[ISAPI-dll] ハンドラをハイライト表示します。
11. 右側のパネルで、[ハンドラのアクセス許可の編集] を選択します。
12. [実行] を選択して、[OK] を選択します。これにより、ISAPI DLL の名前を直接呼び出して、ISAPI 拡張機能を起動できるようになります。

E.2 代替オプション 1 : ISAPI モジュールの使用 (CSPms*.dll)

使用している Web ゲートウェイ DLL でネイティブ・モジュール・インタフェースをサポートできない場合は、このオプションを使用します (推奨オプション)。これは IIS の旧バージョンで提供されていた既定の (また、最もパフォーマンスのよい) ソリューションです。

既定の設定では、IIS 7 は ISAPI 拡張、ISAPI フィルタ、または CGI モジュールを実行しません。このオプションには、ISAPI 拡張サービスが必要です。

“ISAPI サービスおよび CGI サービスのインストール (必要な場合)” の手順に従って、ISAPI 拡張サービスをインストールおよび構成します。

インターネットシステムのファイル・タイプを認識し、それらを Web ゲートウェイに渡して処理するように Web サーバを構成します。

E.2.1 ISAPI 拡張機能の有効化

DLL : CSPms.dll および CSPmsSys.dll

これらの拡張機能を使用できるようにするには、まず“許可済み”アプリケーションとして IIS に登録する必要があります。これは、[インターネット インフォメーション サービス (IIS) マネージャ] コントロール パネルで行います。

1. [インターネット インフォメーション サービス (IIS) マネージャ] ウィンドウを開きます。
2. 左側のパネルで、[MACHINE_NAME] ([machine_name]¥[user_name]) をハイライト表示します。
3. 中央のパネルで、[ISAPI および CGI の制限] をダブルクリックします。
4. 右側のパネルで、[追加] を選択します。
5. [ISAPI または CGI の制限の追加] ダイアログに、以下の詳細を入力します。

ISAPI または CGI パス : C:¥Inetpub¥CSPGateway¥CSPms.dll

説明 : WebGatewayRunTime

拡張パスの実行を許可する : 選択

[OK] を選択します。

E.2.2 InterSystems IRIS ファイル拡張子のマッピング

以下の構成方法のいずれか 1 つを選択します。

- ・ InterSystems IRIS からすべてのコンテンツ (静的なコンテンツを含む) を処理します。Web ゲートウェイに * をマップします。“[追加ファイル・タイプのマッピング](#)” を参照してください。
- ・ Web サーバから静的なコンテンツを処理します。[インターシステムズのファイル・タイプ](#)のみを Web ゲートウェイにマップします。

Web サーバから静的ファイル进行处理する場合、以下のように、[インターシステムズのファイル・タイプ](#)を Web ゲートウェイの ISAPI 拡張機能にマップします。

拡張子	バイナリ
*.csp	C:\Inetpub\CSPGateway\CSPms.dll
*.cls	C:\Inetpub\CSPGateway\CSPms.dll
*.zen	C:\Inetpub\CSPGateway\CSPms.dll
*.cxw	C:\Inetpub\CSPGateway\CSPms.dll

1. [インターネット インフォメーション サービス (IIS) マネージャ] ウィンドウを開きます。
2. 左側のパネルで、最上位レベルを展開して [Web サイト] セクション、[既定の Web サイト] セクションの順に表示します。[既定の Web サイト] セクションをハイライト表示します。

```
[MACHINE_NAME] ([machine_name]\[user_name])
  Web Sites
    Default Web Site
```

注釈 これによって、Web サイト全体に対して CSP が有効になります。特定の仮想サブディレクトリ (/csp/ など) に CSP の使用を制限するには、マッピングを作成する前に ([既定の Web サイト] の下の) 適切なサブディレクトリにコントロールをフォーカスします。CSP のコンテンツを処理するそれぞれの仮想サブディレクトリについて、この手順を繰り返してください。

3. 中央のパネルで、[ハンドラ マッピング] アイコンをダブルクリックします。
4. 右側のパネルで、[スクリプト マップの追加] を選択します。
5. [スクリプト マップの追加] ダイアログに以下のように入力します。

要求パス : *.csp

実行可能ファイル : C:\Inetpub\CSPGateway\CSPms.dll

名前 : WebGateway_csp

6. [要求の制限] を選択します。

[要求がマップされている場合のみハンドラを呼び出す] のチェックを外します。

[OK] を選択して [スクリプト マップの追加] ダイアログに戻ります。

[OK] をクリックします。

7. この時点で、以下のようなプロンプトが表示されることがあります。

“この ISAPI 拡張を有効にしますか? [はい] をクリックすると、[ISAPI および CGI の制限] の一覧の 許可済み の項目に、この拡張が追加されます。拡張が既に存在する場合は、許可されます。”

[はい] を選択します。

以下のようにして、後から許可済みアプリケーションのリストを探すことができます。

左側のパネルで、

[MACHINE_NAME] ([machine_name]¥[user_name]) をハイライト表示します。

中央のパネルで、[ISAPI および CGI の制限] をダブルクリックします。

許可済みアプリケーションのリストに Web ゲートウェイ ISAPI コンポーネントが含まれていない場合は、これを追加します。

[説明] フィールドには自由にテキストを追加できます。以下はその例です。

CSPmsSys.dll WebGatewayManagement

CSPms.dll WebGatewayRunTime

8. 前述の手順を繰り返します。[スクリプト マップの追加] ダイアログを使用して、以下の 2 つのマップを入力します。

要求パス : *.cls

実行可能ファイル : C:¥Inetpub¥CSPGateway¥CSPms.dll

名前 : WebGateway_cls

要求パス : *.zen

実行可能ファイル : C:¥Inetpub¥CSPGateway¥CSPms.dll

名前 : WebGateway_zen

要求パス : *.cxw

実行可能ファイル : C:¥Inetpub¥CSPGateway¥CSPms.dll

名前 : WebGatewayManagement

E.2.3 Web ゲートウェイの運用および管理

Web ゲートウェイのシステム管理スイートへアクセスするには、ブラウザで以下のいずれかの場所を指定します。

http://<hostname>:<port>/csp/bin/Systems/Module.cwx

http://<hostname>:<port>/csp/bin/CSPmsSys.dll

承認されていないユーザであることを通知するエラー・メッセージが表示された場合は、“[追加クライアント・アドレスからのアクセスの有効化](#)”と“[Web ゲートウェイとセキュリティ](#)”を参照してください。

E.3 代替オプション 2 : NSD を使用したネイティブ・モジュールの使用 (CSPcms.dll)

既定の設定では、IIS 7 は ISAPI 拡張、ISAPI フィルタ、または CGI モジュールを実行しません。このオプションを使用するには、Web ゲートウェイ管理モジュール (nph-CSPcgiSys.exe) を実行するために、CGI モジュール・サービスが必要です。

“ISAPI サービスおよび CGI サービスのインストール” の手順に従います (必要な場合)。

インターシステムズのファイル・タイプを認識し、それらを Web ゲートウェイに渡して処理するように Web サーバを構成します。

E.3.1 ランタイム・ネイティブ・モジュールの登録

DLL : CSPcms.dll

このモジュールを使用できるようにするには、まず IIS に登録する必要があります。これは、[インターネット インフォメーション サービス (IIS) マネージャ] コントロール パネルで行います。

1. [インターネット インフォメーション サービス (IIS) マネージャ] ウィンドウを開きます。
2. 左側のパネルで、
[MACHINE_NAME] ([machine_name]¥[user_name]) をハイライト表示します。
3. 中央のパネルで、[モジュール] アイコンをダブルクリックします
4. 右側のパネルで、[ネイティブ モジュールの追加] を選択します。
5. [登録] を選択し、[ネイティブ モジュールの登録] ダイアログに以下の詳細を入力します。
名前 : CSPcms
パス : C:¥Inetpub¥CSPGateway¥CSPcms.dll
[OK] を選択します。
6. 左側のパネルで、最上位レベルを展開して [Web サイト] セクション、[既定の Web サイト] セクションの順に表示します。[既定の Web サイト] セクションをハイライト表示します。
[MACHINE_NAME] ([machine_name]\[user_name])
Web Sites
Default Web Site
7. 右側のパネルで、[ネイティブ モジュールの追加] を選択します。
8. [ネイティブ モジュールの追加] ダイアログで、[CSPcms] を選択し、[OK] を選択します。

E.3.2 Web ゲートウェイ管理のための CGI モジュールの有効化

実行可能ファイル : nph-CSPcgiSys.exe

このモジュールを使用できるようにするには、まず許可済みアプリケーションとして IIS に登録する必要があります。これは、[インターネット インフォメーション サービス (IIS) マネージャ] コントロール パネルで行います。

1. [インターネット インフォメーション サービス (IIS) マネージャ] を開きます。
2. 左側のパネルで、

[MACHINE_NAME] ([machine_name]¥[user_name]) をハイライト表示します。

- 中央のパネルで、[ISAPI および CGI の制限] アイコンをダブルクリックします。
- 右側のパネルで、[追加] を選択します。
- [ISAPI または CGI の制限の追加] ダイアログに、以下のように入力します。

ISAPI または CGI パス : C:¥Inetpub¥CSPGateway¥nph-CSPcgiSys.exe

説明 : WebGatewayManagement

拡張パスの実行を許可する : 選択

[OK] を選択します。

E.3.3 InterSystems IRIS ファイル拡張子のマッピング

注釈 ここでのファイル拡張子のマッピング処理で、**ワイルドカードスクリプトマップの追加**ユーティリティを使用しないでください。使用した場合はエラーとなります。代わりに、**モジュールマップの追加**というユーティリティを使用します。

以下の構成方法のいずれか 1 つを選択します。

- InterSystems IRIS からすべてのコンテンツ (静的なコンテンツを含む) を処理します。Web ゲートウェイに * をマップします。InterSystems IRIS サーバがすべての静的ファイルを処理するように [Web アプリケーション](#) を構成する場合は、[“追加ファイル・タイプのマッピング”](#) を参照してください。
- Web サーバから静的なコンテンツを処理します。

[インターシステムズのファイル・タイプのみ](#)を Web ゲートウェイにマップします。

Web サーバから静的ファイルを処理する場合、以下のように、[インターシステムズのファイル・タイプ](#)を Web ゲートウェイ・モジュールにマップします。

拡張子	ネイティブ・モジュール	バイナリ
*.csp	CSPms	C:¥Inetpub¥CSPGateway¥CSPms.dll
*.cls	CSPms	C:¥Inetpub¥CSPGateway¥CSPms.dll
*.zen	CSPms	C:¥Inetpub¥CSPGateway¥CSPms.dll
*.cxw		C:¥Inetpub¥CSPGateway¥nph-CSPcgiSys.exe

- [インターネット インフォメーション サービス (IIS) マネージャ] ウィンドウを開きます。
- 左側のパネルで、最上位レベルを展開して [Web サイト] セクション、[既定の Web サイト] セクションの順に表示します。[既定の Web サイト] セクションをハイライト表示します。

```
[MACHINE_NAME] ([machine_name]\[user_name])
    Web Sites
        Default Web Site
```

注釈 これによって、Web サイト全体に対して CSP が有効になります。特定の仮想サブディレクトリ (/csp/ など) に CSP の使用を制限するには、マッピングを作成する前に ([既定の Web サイト] の下の) 適切なサブディレクトリにコントロールをフォーカスします。CSP のコンテンツを処理するそれぞれの仮想サブディレクトリについて、この手順を繰り返してください。

- 中央のパネルで、[ハンドラ マッピング] アイコンをダブルクリックします。

4. 右側のパネルで、[モジュール マップの追加] を選択します。
5. [モジュール マップの追加] ダイアログに以下のように入力します。
 要求パス : *.csp
 モジュール : CSPcms を選択します。
 名前 : WebGateway_csp
6. [要求の制限] を選択します。
 [要求がマップされている場合のみハンドラを呼び出す] のチェックを外します。
 [OK] を選択して [モジュール マップの追加] ダイアログに戻ります。
 [OK] をクリックします。
7. 以上の手順を繰り返し、以下のモジュール・マップを追加します。
 要求パス : *.cls
 モジュール : [CSPcms] を選択します。
 名前 : WebGateway_cls
 あるいは以下のコードを入力します。
 要求パス : *.zen
 モジュール : [CSPcms] を選択します。
 名前 : WebGateway_zen
8. 左側のパネルで、[既定の Web サイト] セクションをハイライト表示します。

```
[MACHINE_NAME] ([machine_name]\[user_name])
  Web Sites
    Default Web Site
```
9. 中央のパネルで、[ハンドラ マッピング] アイコンをダブルクリックします。
10. 右側のパネルで、[スクリプト マップの追加] を選択します。
11. [スクリプト マップの追加] ダイアログに以下のように入力します。
 要求パス : *.cwx
 実行可能ファイル : C:\Inetpub\CSPGateway\ncph-CSPcgiSys.exe
 名前 : WebGatewayManagement
12. [要求の制限] を選択します。
 [要求がマップされている場合のみハンドラを呼び出す] のチェックを外します。
 [OK] を選択して [スクリプト マップの追加] ダイアログに戻ります。
 [OK] をクリックします。
13. 以下のようなプロンプトが表示されることがあります。“この ISAPI 拡張を有効にしますか? [はい] をクリックすると、[ISAPI および CGI の制限] の一覧の 許可済み の項目に、この拡張が追加されます。拡張が既に存在する場合は、許可されます。”
 [はい] を選択します。
 以下のようにして、後から許可済みアプリケーションのリストを探すことができます。
 左側のパネルで、

[MACHINE_NAME] ([machine_name]¥[user_name]) をハイライト表示します。

中央のパネルで、[ISAPI および CGI の制限] アイコンをダブルクリックします。

許可済みアプリケーションのリストに Web ゲートウェイ管理 CGI モジュールが含まれていない場合は、これを追加します。

[説明] フィールドには自由にテキストを追加できます。以下はその例です。

nph-CSPcgiSys.exe WebGatewayManagement

E.3.4 Web ゲートウェイの運用および管理

この接続オプションは、Web ゲートウェイの Network Service Daemon (NSD) に依存します。

“[NSD の開始](#)” のセクションの説明に従って、CSP NSD を開始します。

Web ゲートウェイのシステム管理スイートへアクセスするには、ブラウザで以下のいずれかの場所を指定します。

`http://<hostname>:<port>/csp/bin/Systems/Module.cxx`

`http://<hostname>:<port>/csp-bin/nph-CSPcgiSys`

承認されていないユーザであることを通知するエラー・メッセージが表示された場合は、“[追加クライアント・アドレスからのアクセスの有効化](#)” と “[Web ゲートウェイとセキュリティ](#)” を参照してください。

E.4 代替オプション 3 : NSD を使用した ISAPI モジュールの使用 (CSPcms.dll)

使用している Web ゲートウェイ DLL でネイティブ・モジュール・インタフェースをサポートできない場合は、このオプションを使用します (代替オプション 2)。

既定の設定では、IIS 7 は ISAPI 拡張、ISAPI フィルタ、または CGI モジュールを実行しません。このオプションには、ISAPI 拡張サービスと CGI モジュール・サービスが必要です。

“[ISAPI サービスおよび CGI サービスのインストール](#)” の手順に従います。

[インターシステムズのファイル・タイプ](#)を認識し、それらを Web ゲートウェイに渡して処理するように Web サーバを構成します。

E.4.1 ランタイム ISAPI 拡張機能の有効化

DLL : CSPcms.dll

この拡張機能を使用できるようにするには、まず “許可済み” アプリケーションとして IIS に登録する必要があります。これは、[インターネット インフォメーション サービス (IIS) マネージャ] コントロール パネルで行います。

1. [インターネット インフォメーション サービス (IIS) マネージャ] ウィンドウを開きます。
2. 左側のパネルで、[MACHINE_NAME] ([machine_name]¥[user_name]) をハイライト表示します。
3. 中央のパネルで、[ISAPI および CGI の制限] アイコンをダブルクリックします。
4. 右側のパネルで、[追加] を選択します。
5. [ISAPI または CGI の制限の追加] ダイアログに、以下のように入力します。

ISAPI または CGI パス : C:\Inetpub\CSPGateway\CSPcms.dll

説明 : WebGatewayRunTime

拡張パスの実行を許可する : 選択

[OK] を選択します。

E.4.2 Web ゲートウェイ管理のための CGI モジュールの有効化

実行可能ファイル : nph-CSPcgiSys.exe

このモジュールを使用できるようにするには、まず許可済みアプリケーションとして IIS に登録する必要があります。これは、[インターネット インフォメーション サービス (IIS) マネージャ] コントロール パネルで行います。

1. [インターネット インフォメーション サービス (IIS) マネージャ] ウィンドウを開きます。
2. 左側のパネルで、[`MACHINE_NAME`] (`[machine_name]`¥`[user_name]`) をハイライト表示します。
3. 中央のパネルで、[ISAPI および CGI の制限] アイコンをダブルクリックします。
4. 右側のパネルで、[追加] を選択します。
5. [ISAPI または CGI の制限の追加] ダイアログに、以下のように入力します。

ISAPI または CGI パス : C:\Inetpub\CSPGateway\nph-CSPcgiSys.exe

説明 : WebGatewayManagement

拡張パスの実行を許可する : 選択

[OK] を選択します。

E.4.3 InterSystems IRIS ファイル拡張子のマッピング

以下の構成方法のいずれか 1 つを選択します。

- ・ InterSystems IRIS からすべてのコンテンツ (静的なコンテンツを含む) を処理します。Web ゲートウェイに * をマップします。InterSystems IRIS サーバがすべての静的ファイル进行处理するように InterSystems IRIS で [Web アプリケーション](#)を構成する場合は、“[追加ファイル・タイプのマッピング](#)” を参照してください。
- ・ Web サーバから静的なコンテンツ进行处理します。

[インターシステムズのファイル・タイプ](#)のみを Web ゲートウェイにマップします。

Web サーバから静的ファイル进行处理する場合、以下のように、[インターシステムズのファイル・タイプ](#)を Web ゲートウェイ・モジュールにマップします。

拡張子	バイナリ
*.csp	C:\Inetpub\CSPGateway\CSPcms.dll
*.cls	C:\Inetpub\CSPGateway\CSPcms.dll
*.zen	C:\Inetpub\CSPGateway\CSPcms.dll
*.cxw	C:\Inetpub\CSPGateway\nph-CSPcgiSys.exe

1. [インターネット インフォメーション サービス (IIS) マネージャ] ウィンドウを開きます。

2. 左側のパネルで、最上位レベルを展開し、**[Web サイト]** を展開します。**[既定の Web サイト]** をハイライト表示します。

```
[MACHINE_NAME] ([machine_name]\[user_name])
  Web Sites
    Default Web Site
```

注釈 これによって、Web サイト全体に対して CSP が有効になります。特定の仮想サブディレクトリ (/csp/ など) に CSP の使用を制限するには、マッピングを作成する前に (**[既定の Web サイト]** の下の) 適切なサブディレクトリにコントロールをフォーカスします。CSP のコンテンツを処理するそれぞれの仮想サブディレクトリについて、この手順を繰り返してください。

3. 中央のパネルで、**[ハンドラ マッピング]** をダブルクリックします。
4. 右側のパネルで、**[スクリプト マップの追加]** を選択します。
5. **[スクリプト マップの追加]** ダイアログに以下のように入力します。

要求パス : *.csp

実行可能ファイル : C:\inetpub\CSPGateway\CSPcms.dll

名前 : WebGateway_csp

6. **[要求の制限]** を選択します。

[要求がマップされている場合のみハンドラを呼び出す] のチェックを外します。

[OK] を選択して **[スクリプト マップの追加]** ダイアログに戻ります。

[OK] をクリックします。

7. この時点で、以下のようなプロンプトが表示されることがあります。

“この ISAPI 拡張を有効にしますか? [はい] をクリックすると、**[ISAPI および CGI の制限]** の一覧の 許可済み の項目に、この拡張が追加されます。拡張が既に存在する場合は、許可されます。”

[はい] を選択します。

以下のようにして、後から許可済みアプリケーションのリストを探すことができます。

左側のパネルで、

[MACHINE_NAME] ([machine_name]\[user_name]) をハイライト表示します。

中央のパネルで、**[ISAPI および CGI の制限]** アイコンをダブルクリックします。

許可済みアプリケーションのリストに Web ゲートウェイ ISAPI モジュールが含まれていない場合は、これを追加します。

[説明] フィールドには自由にテキストを追加できます。以下はその例です。

CSPcms.dll WebGatewayRunTime

nph-CSPcgiSys.exe WebGatewayManagement

8. 前述の手順を繰り返します。**[スクリプト マップの追加]** ダイアログを使用して、以下の 2 つのマップを入力します。

要求パス : *.cls

実行可能ファイル : C:\inetpub\CSPGateway\CSPcms.dll

名前 : WebGateway_cls

要求パス : *.zen

実行可能ファイル : C:\inetpub\CSPGateway\CSPcms.dll

名前 : WebGateway_zen

要求パス : *.cxw

実行可能ファイル : C:\Inetpub\CSPGateway\nph-CSPcgiSys.exe

名前 : WebGatewayManagement

E.4.4 Web ゲートウェイの運用および管理

この接続オプションは、Web ゲートウェイの Network Service Daemon (NSD) に依存します。

1. このサービスに関するセクションで説明されているとおり CSP NSD を開始します。

Web ゲートウェイのシステム管理スイートへアクセスするには、ブラウザで以下のいずれかの場所を指定します。

`http://<hostname>:<port>/csp/bin/Systems/Module.cxw`

`http://<hostname>:<port>/csp-bin/nph-CSPcgiSys`

承認されていないユーザであることを通知するエラー・メッセージが表示された場合は、“[追加クライアント・アドレスからのアクセスの有効化](#)”と“[Web ゲートウェイとセキュリティ](#)”を参照してください。

E.5 代替オプション 4 : NSD を使用した CGI モジュールの使用 (nph-CSPcgi*.exe)

ほとんどの環境では、包括的なネイティブ・モジュール・ベースのソリューション (推奨オプション) が最も適しており、また、パフォーマンスに優れた実装でもあります。運用上の理由から、ホスト Web サーバとは別に Web ゲートウェイを管理する必要がある場合には、CGI と NSD の組み合わせが役立ちます。例えば、Web サーバの複数のインスタンスが同じ Web ゲートウェイのインストール環境を共有する場合などがこれにあたります。オプション 1 では、コア Web サーバ・プロセスの各インスタンスが、それぞれの Web ゲートウェイ・インスタンスにバインドします。

また、社内の Web マスタ (または ISP) から、すべての Web サーバ拡張を CGI プロトコルを使用して実装するよう指示されている場合も、この方法を選択します。

既定の設定では、IIS 7 は ISAPI 拡張、ISAPI フィルタ、または CGI モジュールを実行しません。このオプションには、CGI モジュール・サービスが必要です。

“[ISAPI サービスおよび CGI サービスのインストール](#)”で説明した CGI サービスのインストール手順に従って操作します。

[インターシステムズのファイル・タイプ](#)を認識し、それらを Web ゲートウェイに渡して処理するように Web サーバを構成します。

E.5.1 CGI のモジュールの有効化

実行可能ファイル : nph-CSPcgi.exe および nph-CSPmsSys.exe

これらのモジュールを使用できるようにするには、まず“許可済み”アプリケーションとして IIS に登録する必要があります。これは、[\[インターネット インフォメーション サービス \(IIS\) マネージャ\]](#) コントロール パネルで行います。

1. [\[インターネット インフォメーション サービス \(IIS\) マネージャ\]](#) ウィンドウを開きます。
2. 左側のパネルで、

[MACHINE_NAME] ([machine_name]¥[user_name]) をハイライト表示します。

- 中央のパネルで、[ISAPI および CGI の制限] アイコンをダブルクリックします。
- 右側のパネルで、[追加] を選択します。
- [ISAPI または CGI の制限の追加] ダイアログに、以下のように入力します。

ISAPI または CGI パス : C:¥Inetpub¥CSPGateway¥nph-CSPcgi.exe

説明 : WebGatewayRunTime

拡張パスの実行を許可する : 選択

[OK] を選択します。

- nph-CSPcgiSys.exe に対して前述の手順を繰り返し、[制限条件] ダイアログに以下の詳細を入力します。

ISAPI または CGI パス : C:¥Inetpub¥CSPGateway¥nph-CSPcgiSys.exe

説明 : WebGatewayManagement

拡張パスの実行を許可する : 選択

E.5.2 InterSystems IRIS ファイル拡張子のマッピング

以下の構成方法のいずれか 1 つを選択します。

- InterSystems IRIS からすべてのコンテンツ (静的なコンテンツを含む) を処理します。Web ゲートウェイに * をマップします。InterSystems IRIS サーバがすべての静的ファイル进行处理するように InterSystems IRIS で [Web アプリケーション](#) を構成する場合は、"[追加ファイル・タイプのマッピング](#)" を参照してください。
- Web サーバから静的なコンテンツ进行处理します。

[インターシステムズのファイル・タイプのみ](#)を Web ゲートウェイにマップします。

Web サーバから静的ファイル进行处理する場合、以下のように、[インターシステムズのファイル・タイプ](#)を Web ゲートウェイの CGI モジュールにマップします。

拡張子	バイナリ
*.csp	C:¥Inetpub¥CSPGateway¥nph-CSPcgi.exe
*.cls	C:¥Inetpub¥CSPGateway¥nph-CSPcgi.exe
*.zen	C:¥Inetpub¥CSPGateway¥nph-CSPcgi.exe
*.cxw	C:¥Inetpub¥CSPGateway¥nph-CSPcgiSys.exe

- [インターネット インフォメーション サービス (IIS) マネージャ] ウィンドウを開きます。
- 左側のパネルで、最上位レベルを展開して [Web サイト] セクション、[既定の Web サイト] セクションの順に表示します。[既定の Web サイト] セクションをハイライト表示します。

```
[MACHINE_NAME] ([machine_name]\[user_name])
    Web Sites
        Default Web Site
```

注釈 これによって、Web サイト全体に対して CSP が有効になります。特定の仮想サブディレクトリ (/csp/ など) に CSP の使用を制限するには、マッピングを作成する前に ([既定の Web サイト] の下の) 適切なサブディレクトリにコントロールをフォーカスします。CSP のコンテンツ进行处理するそれぞれの仮想サブディレクトリについて、この手順を繰り返してください。

3. 中央のパネルで、[ハンドラ マッピング] アイコンをダブルクリックします。
4. 右側のパネルで、[スクリプト マップの追加] を選択します。
5. [スクリプト マップの追加] ダイアログに以下のように入力します。
 要求パス : *.csp
 実行可能ファイル : C:\¥Inetpub¥CSPGateway¥nph-CSPcgi.exe
 名前 : WebGateway_csp
6. [要求の制限] を選択します。
 [要求がマップされている場合のみハンドラを呼び出す] のチェックを外します。
 [OK] を選択して [スクリプト マップの追加] ダイアログに戻ります。
 [OK] をクリックします。
7. この時点で、以下のようなプロンプトが表示されることがあります。“この ISAPI 拡張を有効にしますか? [はい] をクリックすると、[ISAPI および CGI の制限] の一覧の 許可済み の項目に、この拡張が追加されます。拡張が既に存在する場合は、許可されます。”
 [はい] を選択します。
8. 以下のようにして、後から許可済みアプリケーションのリストを探することができます。
 左側のパネルで、
 [MACHINE_NAME] ([machine_name]¥[user_name]) をハイライト表示します。
 中央のパネルで、[ISAPI および CGI の制限] アイコンをダブルクリックします。
 許可済みアプリケーションのリストに Web ゲートウェイ CGI コンポーネントが含まれていない場合は、これを追加します。
 [説明] フィールドには自由にテキストを追加できます。以下はその例です。
 nph-CSPcgiSys.exe WebGatewayManagement
 nph-CSPcgi.exe WebGatewayRunTime
9. 前述の手順を繰り返します。[スクリプト マップの追加] ダイアログを使用して、以下の 2 つのマップを入力します。
 要求パス : *.cls
 実行可能ファイル : C:\¥Inetpub¥CSPGateway¥nph-CSPcgi.exe
 名前 : WebGateway_cls
 要求パス : *.zen
 実行可能ファイル : C:\¥Inetpub¥CSPGateway¥nph-CSPcgi.exe
 名前 : WebGateway_zen
 要求パス : *.cxw
 実行可能ファイル : C:\¥Inetpub¥CSPGateway¥nph-CSPcgiSys.exe
 名前 : WebGatewayManagement

E.5.3 Web ゲートウェイの運用および管理

この接続オプションは、Web ゲートウェイの Network Service Daemon (NSD) に依存します。

1. このサービスに関するセクションで説明されているとおり CSP NSD を開始します。

Web ゲートウェイのシステム管理スイートへアクセスするには、ブラウザで以下のいずれかの場所を指定します。

`http://<hostname>:<port>/csp/bin/Systems/Module.cxx`

`http://<hostname>:<port>/csp-bin/nph-CSPcgiSys`

承認されていないユーザであることを通知するエラー・メッセージが表示された場合は、“[追加クライアント・アドレスからのアクセスの有効化](#)”と“[Web ゲートウェイとセキュリティ](#)”を参照してください。