



Web ゲートウェイ・ガイド

Version 2023.1
2024-01-02

Web ゲートウェイ・ガイド

InterSystems IRIS Data Platform Version 2023.1 2024-01-02

Copyright © 2024 InterSystems Corporation

All rights reserved.

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

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

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

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

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

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

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

InterSystems Worldwide Response Center (WRC)

Tel: +1-617-621-0700

Tel: +44 (0) 844 854 2917

Email: support@InterSystems.com

目次

1 Web ゲートウェイの概要	1
1.1 Web サーバ、Web ゲートウェイ、および CSP サーバの目的	1
1.2 Web サーバ、Web ゲートウェイ、および CSP サーバの連携方法	2
2 Web ゲートウェイでサポートされる構成	3
2.1 サポートされている構成	3
2.2 サポート対象の Web サーバ	3
3 Web ゲートウェイのインストール	5
3.1 既定の Web ゲートウェイのインストールと構成	5
3.2 Web ゲートウェイ管理モジュール	5
3.3 Web ゲートウェイ・コンポーネントと物理的なインストール・パス	6
3.4 Web ゲートウェイのスタンドアロン・コンポーネントとしての導入	8
4 プライベート Web サーバの使用または置換	9
4.1 プライベート Web サーバの目的	9
4.2 プライベート Web サーバの制限	9
4.2.1 Windows での制限	10
4.2.2 プライベート Web サーバの構築	10
4.2.3 UNIX® での制限	10
4.2.4 メモ	11
4.3 別の Web サーバを使用した管理ポータルの実行	11
4.4 プライベート Web サーバの管理	11
4.4.1 構成の変更	11
4.4.2 Windows でのプライベート Web サーバの管理	12
4.4.3 Unix® でのプライベート Web サーバの管理	12
5 Web ゲートウェイと連携させるための IIS の構成 (Windows)	13
5.1 一般的な手順 (すべてのバージョン)	13
5.1.1 Web ゲートウェイ・コンポーネントに対する許可の設定	13
5.1.2 Web アプリケーション・パスの構成	14
5.1.3 /bin を含む URLの有効化	15
5.1.4 IIS の再起動	15
5.1.5 トラブルシューティング	15
5.2 追加手順 : Microsoft IIS 7 以降	16
5.2.1 IIS 7 のインストール場所	16
5.2.2 推奨オプション : ネイティブ・モジュールの使用 (CSPms*.dll)	17
5.2.3 SOAP フォルトの詳細を返すための IIS の構成	19
6 Web ゲートウェイと連携させるための Apache の構成 (Windows)	21
6.1 想定	21
6.2 Apache サーバでのインストール場所 (すべての構成)	22
6.3 推奨オプション : Apache API モジュール (CSPa24.dll)	22
7 追加ファイル・タイプを渡すための Apache の構成 (すべてのプラットフォーム)	25
7.1 Location による Apache の構成	25
7.2 ファイル拡張子による Apache の構成 (CSPFileTypes 指示文)	26
7.3 MIME タイプによる Apache の構成	26
8 Web ゲートウェイと連携させるための Nginx の構築と構成 (Windows)	27
8.1 想定	27

8.2 インストール	27
8.3 CSP のための Nginx Web サーバの構築	28
8.3.1 CSP のための Nginx の構築手順	29
8.4 Nginx での NSD の使用	30
8.4.1 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	31
8.4.2 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	32
8.4.3 Nginx と NSD の起動および停止	32
8.5 非推奨：ユニバーサル・モジュールを使用する Nginx の構築	32
9 Web ゲートウェイと連携させるための Apache の構成 (UNIX®/Linux/macOS)	35
9.1 想定	36
9.2 UNIX®, Linux、macOS での Apache のインストール場所 (推奨オプション)	36
9.2.1 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)	37
9.3 推奨オプション：NSD を使用しない Apache API モジュール (CSPa24.so)	37
10 Nginx の構築と構成 (UNIX®/Linux/macOS)	39
10.1 想定	39
10.2 インストール	39
10.3 CSP のための Nginx Web サーバの構築	40
10.3.1 CSP のための Nginx の構築手順	41
10.4 Nginx での NSD の使用	41
10.4.1 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	43
10.4.2 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	43
10.4.3 Nginx と NSD の起動および停止	43
10.5 非推奨：ユニバーサル・モジュールを使用する Nginx の構築	43
11 Web ゲートウェイの構成の基本	45
11.1 Web ゲートウェイの構成ファイルとログ・ファイル	45
11.2 自動的に InterSystems IRIS にルーティングされるファイル・タイプ	45
11.3 InterSystems IRIS からの静的ファイルの提供	46
11.3.1 文字エンコードの指定	46
11.3.2 [静的ファイルの提供] オプションの有効化	46
11.3.3 InterSystems IRIS が静的ファイル进行处理できるようにするための Web サーバの構成	47
11.3.4 Web サーバからの静的ファイルの処理	47
11.4 高可用性ソリューションのハードウェア・ロード・バランサでのスティッキー・セッションの有効化	47
11.5 Web ゲートウェイ構成を再起動するスクリプトの有効化	47
11.6 マルチプロセス/マルチスレッドのハイブリッド型 Web サーバ・アーキテクチャ	48
12 Web ゲートウェイ管理ページの概要	49
12.1 Web ゲートウェイ管理ページへのアクセス	49
12.2 追加クライアント・アドレスからのアクセスの有効化	49
12.3 使用可能なオプション	50
12.4 ローカリゼーション	51
13 Web ゲートウェイの既定パラメータの構成	53
13.1 Web ゲートウェイ	53

13.2 セキュリティ	54
13.3 InterSystems IRIS への接続	56
13.4 ASP リダイレクト	59
13.5 [内部HTTPサーバ]	59
13.6 カスタム・エラー・ページ	60
13.7 イベント・ログ・パラメータ	60
14 サーバ・アクセスの構成	69
14.1 サーバ構成の追加	69
14.1.1 サーバ・アクセス・パラメータ	69
14.1.2 状態なしパラメータ	70
14.1.3 接続セキュリティのパラメータ	70
14.1.4 SSL/TLS パラメータ	71
14.1.5 [オプションパラメータ]	74
14.1.6 [エラーページ]	74
14.2 サーバ構成のコピー	74
14.3 構成したサーバへのアクセスの無効化	74
14.4 サーバ構成の削除	75
15 アプリケーション・アクセスの構成	77
15.1 アプリケーション・パスの追加	77
15.1.1 アプリケーション・パスの構成パラメータ	78
15.1.2 サーバ・パラメータ	79
15.2 アプリケーション・パス構成のコピー	80
15.3 アプリケーション・パスによるアクセスの無効化	80
15.4 アプリケーション・パス構成の削除	81
16 Web ゲートウェイの管理と監視	83
16.1 システム・ステータスの確認	83
16.1.1 InterSystems IRIS への接続	83
16.1.2 InterSystems IRIS サーバ・テーブル	84
16.1.3 アプリケーション・パス・テーブル	84
16.1.4 Web ゲートウェイ・キャッシュ・テーブル	85
16.1.5 手動による接続の切断	85
16.1.6 キャッシュのクリア	85
16.2 サーバ接続のテスト	86
16.3 イベント・ログの表示	86
16.4 HTTP トレース機能の使用法	87
17 InterSystems IRIS への Web ゲートウェイ接続の保護	89
17.1 Web ゲートウェイの接続セキュリティの構成	89
17.2 最小の接続セキュリティ (非推奨)	90
17.3 単純なユーザ名/パスワード認証	90
17.3.1 外部から導入されるパスワード	91
17.3.2 別のコンピュータで暗号化されたパスワード	91
17.4 Kerberos ベース認証とデータ保護	92
17.4.1 Kerberos ライブラリ	92
17.4.2 Windows	93
17.4.3 Kerberos のための UNIX® Web ゲートウェイ構成	94
17.5 SSL/TLS ベース認証とデータの保護	94
18 Web ゲートウェイによって渡される CGI 環境変数	95
19 Web ゲートウェイにより返される HTTP 応答ヘッダ	97

20 CSP フォームの要求に対する応答の圧縮 (GZIP/ZLIB)	99
20.1 GZIP/ZLIB ライブラリ	100
20.2 GZIP/ZLIB ライブラリの使用法	100
20.3 ページごとの圧縮の指定	101
20.4 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	101
20.5 監視	102
21 Web アプリケーション向けの HTTP 認証の実装	103
21.1 Apache での標準 HTTP 認証 (mod_auth)	103
21.2 要求の処理と同時に CSP で行われる認証	104
21.3 要求の処理前に CSP で行われる認証	105
22 ミラー構成、フェイルオーバー、および負荷分散	107
22.1 複数の Web サーバ間の負荷分散とフェイルオーバー	107
22.2 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	107
22.3 ミラー構成	108
23 プロセス親和性とステート認識モード (保持モード 1)	111
23.1 ステート認識モードの起動	112
23.2 ステート認識モードの維持とエラーへの応答	113
23.3 ステート認識モードの終了	114
24 InterSystems IRIS の Web ゲートウェイ・レジストリ	115
24.1 構成を再ロードするよう Web ゲートウェイを強制する	116
24.1.1 InterSystems IRIS Web ゲートウェイ・レジストリの使用	116
24.1.2 InterSystems IRIS の外部スクリプトの使用	117
付録A: NSD の使用法 (Windows)	119
A.1 NSD を使用する場合	119
A.2 NSD モジュールのインストール場所	119
A.3 NSD の操作	120
A.3.1 代替 TCP ポートでの NSD の開始	120
付録B: IIS 7 以降の代替オプション (Windows)	123
B.1 ISAPI サービスおよび CGI サービスのインストール	123
B.2 代替オプション 1 : ISAPI モジュールの使用 (CSPms*.dll)	124
B.2.1 ISAPI 拡張機能の有効化	124
B.2.2 InterSystems IRIS ファイル拡張子のマッピング	125
B.2.3 Web ゲートウェイの運用および管理	126
B.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)	126
B.3.1 ランタイム・ネイティブ・モジュールの登録	127
B.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	127
B.3.3 InterSystems IRIS ファイル拡張子のマッピング	128
B.3.4 Web ゲートウェイの運用および管理	130
B.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)	130
B.4.1 ランタイム ISAPI 拡張機能の有効化	130
B.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	131
B.4.3 InterSystems IRIS ファイル拡張子のマッピング	131
B.4.4 Web ゲートウェイの運用および管理	133
B.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)	133
B.5.1 CGI のモジュールの有効化	133
B.5.2 InterSystems IRIS ファイル拡張子のマッピング	134
B.5.3 Web ゲートウェイの運用および管理	135
付録C: Apache の代替オプション (Windows)	137

C.1 インストール場所 (すべての特殊オプション)	137
C.2 代替オプション 1 : NSD を使用した Apache および CGI モジュール (nph-CSPcgi.exe) ..	138
C.2.1 追加ファイル・タイプのマッピング	139
C.2.2 Apache NSD を使用した Web ゲートウェイの運用および管理	139
C.3 代替オプション 2 : NSD を使用した Apache API モジュール (mod_csp24.dll)	140
C.3.1 追加ファイル・タイプのマッピング	140
C.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	141
C.4 Microsoft Windows 向けにロック・ダウンした Apache 環境	141
C.4.1 推奨オプションの構成 : Apache API モジュール (CSPa24.dll)	142
C.4.2 代替オプション 2 の構成 : NSD を使用した CGI モジュール (nph-CSPcgi.exe) ..	142
C.4.3 代替オプション 3 の構成 : NSD を使用した Apache API モジュール (mod_csp24.dll)	142
.....	142
C.4.4 代替オプション 4 の構成 : ISAPI モジュール (CSPms.dll)	142
付録D: NSD の使用法 (UNIX®/Linux/macOS)	143
D.1 NSD を使用する場合	143
D.2 NSD モジュールのインストール場所	143
D.3 NSD の操作	143
D.3.1 代替 TCP ポートでの NSD の開始	145
付録E: Apache の代替オプション (UNIX®/Linux/macOS)	149
E.1 インストール場所 (すべての特殊オプション)	149
E.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	150
1)	150
E.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)	151
E.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	151
モジュールのビルド	151
E.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	151
E.2.3 実行時の構成	152
E.2.4 追加ファイル・タイプのマッピング	153
E.3 代替オプション 2 : NSD を使用した CGI モジュール (nph-CSPcgi)	153
E.3.1 追加ファイル・タイプのマッピング	155
E.3.2 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	155
E.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)	155
E.4.1 CSP モジュール・ソース・コードを含めた Apache のビルド	155
E.4.2 生成した Apache バイナリの確認	155
E.4.3 実行時の構成	156
E.4.4 追加ファイル・タイプのマッピング	156
E.4.5 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	156
付録F: ロック・ダウン Apache (UNIX®/Linux/macOS)	159
F.1 推奨オプション : Apache API モジュール (CSPa24.so)	160
F.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)	160
F.3 代替オプション 2 : NSD を使用した CGI モジュール (nph-CSPcgi)	161
F.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)	161
付録G: Apache に関する考慮事項 (UNIX®/Linux/macOS)	163
G.1 Apache プロセス管理および処理能力の計画	163
G.1.1 セキュリティ	164
G.1.2 Apache MPM と Web ゲートウェイ DSO	164
G.2 ステート認識セッション (保持モード 1)	164
付録H: IIS の技術メモ	167

H.1 IIS アプリケーション・プールおよび Web ガーデン	167
H.1.1 アプリケーション・プール	167
H.1.2 Web ガーデン	167
H.1.3 アプリケーション・プール、Web ガーデン、および CSP	168
H.1.4 ワーカ・プロセスのアイドル・タイムアウト	168
H.1.5 ワーカ・プロセスのリサイクル	169
H.2 ビットネス – Windows の 64 ビット・サーバ上での 32 ビット・アプリケーション	169
付録I: リモート Web サーバでの Web アプリケーションの使用	171
I.1 Web サーバと Web ゲートウェイの構成	171
I.1.1 Web サーバ・マシンへの Web ゲートウェイのインストール	171
I.1.2 Web ゲートウェイの構成	171
I.1.3 Web サーバから静的ファイルを提供する場合	172
I.1.4 Web サーバ・パスの構成	173
I.2 複数の InterSystems IRIS サーバ上での CSP へのアクセス	174
I.2.1 アプリケーション・パス用の InterSystems IRIS サーバの構成	174
I.2.2 URL での InterSystems IRIS サーバ名の変更	174
I.3 Apache 仮想ホストの構成	175
I.3.1 仮想ホストの概要	176
付録J: WebSocket の使用 (RFC 6455)	179
J.1 WebSocket プロトコル	179
J.2 WebSocket のクライアント・コード (JavaScript)	180
J.2.1 WebSocket の作成	180
J.2.2 WebSocket のクライアント・イベント	180
J.2.3 WebSocket のクライアント・メソッド	181
J.3 WebSocket のサーバ・コード	181
J.3.1 WebSocket のサーバ・イベント	181
J.3.2 WebSocket のサーバ・メソッド	181
J.3.3 WebSocket のサーバ・プロパティ	182
J.4 WebSocket サーバの例	183
J.5 WebSockets サーバの非同期動作	183

1

Web ゲートウェイの概要

InterSystems IRIS® [Web アプリケーション](#)は、そのアプリケーションの HTTP 要求および応答をまとめて処理する 3 つのソフトウェア・コンポーネントに依存します。

- Web サーバは、HTTP 要求を受信し、それらの処理方法を決定するソフトウェア・ユーティリティです。インターシステムズは、オペレーティング・システムによって異なる、[数種類の Web サーバ](#)をサポートしています。
- InterSystems Web ゲートウェイは、Web サーバからの HTTP 要求を受信し、それらを適切な InterSystems IRIS サーバに送信する、ソフトウェア・ユーティリティです。

プラットフォームに応じて、Web ゲートウェイは共有ライブラリ、.dll ファイル、または CGI スクリプトとなります。Web ゲートウェイは、[アーキテクチャニーズ](#)に応じて、InterSystems IRIS の一部として、またはスタンドアロン・ユーティリティとして[インストール](#)できます。

Web ゲートウェイには[管理ページ](#)があり、これを介して[グローバル構成の実行](#)、[サーバの構成](#)、[Web アプリケーションの構成](#)、および Web ゲートウェイの[監視](#)を行うことができます。

- CSP サーバは Web ゲートウェイからの HTTP 要求を受信し、InterSystems IRIS 内のコードを呼び出すことにより、これらを処理します。CSP サーバは、実際には InterSystems IRIS サーバ内で実行されるプロセスです。これは、Web ゲートウェイから受信した要求の処理を専門とするプロセスです。

1.1 Web サーバ、Web ゲートウェイ、および CSP サーバの目的

この Web サーバは以下を実行します。

- 通常はブラウザから、着信 HTTP 要求を受け入れる。
- 許可を確認する。
- 静的コンテンツを処理する（次の項目を参照してください）。
- 特定の URL の形式の要求を Web ゲートウェイに送信する。これは Web サーバの構成によって決定されます。どのような場合でも、以下の拡張子で終了する URL は、Web ゲートウェイによって処理される必要があります。

`.csp .cls .zen .cxw`

最初の 3 つの拡張子は、InterSystems IRIS で実行できるコードの種類を示します。最後の拡張子 (`.cxw`) は、Web ゲートウェイ管理ページで使用される、予約された拡張子です。Web ゲートウェイのコンテキスト内で、これらの拡張子はインターシステムズのファイル・タイプを表します。

追加のファイル・タイプを Web ゲートウェイに送信するよう、Web サーバを構成することもできます。

この Web ゲートウェイは以下を実行します。

- ・ どの InterSystems IRIS サーバに要求を送信するかを判断する。
- ・ 正しい InterSystems IRIS サーバ (具体的には、その InterSystems IRIS サーバ内の CSP サーバ) に要求を送信する。(`cxw` で終了する URL の場合、Web ゲートウェイは代わりに別の Web ゲートウェイ管理モジュールを呼び出します。)
- ・ InterSystems IRIS サーバへの接続を保持する (絶えず新しい接続を開くのを避けるため)。

InterSystems IRIS 内で、CSP サーバは以下を実行します。

- ・ アプリケーションの HTTP 要求を受信する。
- ・ [Web アプリケーション](#) の構成設定を確認する。これはインターシステムズの管理ポータルで指定され、InterSystems IRIS の [CPF ファイル](#) に保存されます。
- ・ 要求されたコードでコールバックを実行する。これにより HTML は Web ゲートウェイに送り返され、そこから Web サーバに送信されてブラウザに戻されます。

1.2 Web サーバ、Web ゲートウェイ、および CSP サーバの連携方法

クライアントが [Web アプリケーション](#) からの情報を要求する場合、情報の流れは次のようになります。

1. HTTP クライアント (通常は Web ブラウザ) は、標準の HTTP (または HTTPS) プロトコルを使用して Web サーバにページを要求します。
2. Web サーバはこれを CSP 要求として認識し、高速なサーバ API を使用して Web ゲートウェイに転送します。
3. Web ゲートウェイは、通信する InterSystems IRIS サーバを決定し、そのターゲット・システム上の CSP サーバに要求を転送します。
4. InterSystems IRIS で実行されている CSP サーバは要求を処理して Web ゲートウェイにページを返し、Web ゲートウェイはそれを Web サーバに渡します。
5. Web サーバはページをブラウザに送信し、ブラウザでページが表示されます。

2

Web ゲートウェイでサポートされる構成

ここでは、[InterSystems Web ゲートウェイ](#)でサポートされている構成の概要を示します。InterSystems Web ゲートウェイは、Web サーバと InterSystems IRIS® 間の通信（特に、[Web アプリケーション](#)や管理ポータルとの通信）を処理します。

注釈 InterSystems IRIS は、管理ポータルの実行のみを目的として、[プライベート Web サーバ](#) (Apache の最小ビルド) を提供します。プロダクション・システムでは、代わりに、[サポートされている Web サーバ](#)のいずれかをインストールして構成する必要があります。この方がより安全で堅牢です。

2.1 サポートされている構成

開発時には、Web サーバ、Web ゲートウェイ、および InterSystems IRIS インスタンスをすべて 1 つのマシン上に配置できます。大規模な導入では、複数の Web サーバと複数の InterSystems IRIS インスタンスがすべて異なるマシンに配置される場合があります。各 Web サーバには、専用の Web ゲートウェイが必要です。単一の Web ゲートウェイが複数の InterSystems IRIS インスタンスと通信できます。

InterSystems IRIS の[ゲートウェイ・レジストリ](#)では、接続を監視および管理できます。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。レジストリには、構成を読み書きし、システム状態および [Web ゲートウェイ・イベント・ログ](#)を監視するために、InterSystems IRIS コードが接続先の Web ゲートウェイ・インストールと通信することを可能にするインフラストラクチャが含まれています。

2.2 サポート対象の Web サーバ

プロダクション環境では、InterSystems Web ゲートウェイと共に、次の Web サーバを使用できます。

- ・ Microsoft Windows :
 - Microsoft IIS。接続オプションには、[既定の IIS](#) オプションと [IIS 7 以降のいくつかの代替オプション](#)が含まれます。
 - Apache。接続オプションには、[既定の Apache](#) オプションと [Windows Apache のいくつかの代替オプション](#)が含まれます。
 - [Nginx](#)
- ・ Linux、UNIX®, および macOS:

- Apache. 既定の Apache オプション、NSD、ロック・ダウン Apache Web サーバ、その他いくつかの特殊 Apache 構成といった、いくつかの接続オプションが利用可能です。
- Nginx

注釈 プライベート Web サーバはプロダクション・システムで使用することを意図していないため、このリストには含まれません。

サポート対象の Web サーバについての詳細は、“インターシステムズのサポート対象プラットフォーム”の“サポート対象の Web サーバ”を参照してください。

Web ゲートウェイには、Microsoft、Apache、および Nginx の各 Web サーバ向けの高性能な接続ソリューションが用意されています。これらのソリューションのほかに、サポートされているすべてのオペレーティング・システムでは、CGI 経由で InterSystems IRIS に接続できます。

Microsoft の Web サーバはマルチスレッド API をサポートしており、Web サーバのコア機能として、拡張機能を動的結合ライブラリの形式で取り込むことができます。Web ゲートウェイの最新バージョンは、これらの API を最大限に利用し、InterSystems IRIS システムへの高性能な Web 接続を実現します。Windows バージョンの Apache も排他的にマルチスレッド・モードで動作するので、動的結合ライブラリとして実装された Web ゲートウェイを活用できます。

UNIX® バージョンの Apache は、排他的にマルチスレッドではないという点で、Microsoft Windows ベースの Web サーバとはアーキテクチャが異なります。Apache バージョン 2.4 は、スレッドおよびマルチ・プロセスから構成されたハイブリッド型モデルを使用して実装されます。このモデルでは、各 UNIX® プロセスは、それ自体で事実上マルチスレッド・サーバです。

Apache Web サーバは、CGI モジュールとして実装されている拡張機能をサポートするだけでなく、独自仕様の API も公開します。Apache には、ユーザ定義のモジュール（コンパイル済みの C プログラム）としてその他の機能を追加できます。実際、Apache のコア機能の大部分はモジュール・セットとして実装されています。Apache にモジュールを追加するには 2 つの方法があります。1 つは、モジュールのソースを、Apache コアへ直接コンパイルする方法です。この方法では確実に最高のパフォーマンスが実現できると期待できますが、Web サーバの再構成と再構築が必要になります。モジュール・ソースを Apache コアに直接組み込む方法の代替手段として、Apache バージョン 1.3 以降は、ダイナミック・リンク・ライブラリとして実装された拡張機能をサポートしています。この機能によって、Apache のコアへモジュールを物理的に組み込まなくても、Apache モジュールの優れた性能を利用することができます。CSP モジュールは、Windows のダイナミック・リンク・ライブラリ (DLL)、および UNIX® の動的共有オブジェクト (DSO) として配布されます。UNIX® 共有オブジェクトは、概念的に Windows ダイナミック・リンク・ライブラリ (DLL) と類似しており、実行時にリンクされます。最近のオペレーティング・システムでは、非常に小さいオーバーヘッドで実行時にライブラリへリンクすることができます。

他の Web サーバとは異なり、Nginx は非同期のイベント駆動型アーキテクチャに基づいています。イベント駆動型アーキテクチャにより、個々の操作の開始と完了をマークするために通知またはシグナルが使用されます。この設計により、Web 要求の処理中に、リソースを一時的に解放して、他の操作によって使用できるようにすることができます。リソースは動的に割り当ておよび解放され、それらが実際に必要とされる間のみ Web 要求の処理に関連付けられます。これにより、メモリおよび CPU を高度に最適化して使用できるようになります。このアーキテクチャの非同期の性質により、複数のスレッドを相互にブロックすることなく同時に実行でき、リソースがブロック操作を待機するスレッドに関連付けられなくなるため、リソースの共有が向上します。Nginx は API に付属し、CSP などの拡張機能をコア機能に追加することができます。ただし、その他の Web サーバとは異なり、拡張モジュールはコンパイル時に Web サーバ・コアに組み込まれている必要があります。Nginx では、動的にロードされる拡張モジュールはサポートされません。

Web ゲートウェイの機能がスタンドアロンの実行可能コードとして実装され、それ自体のプロセスで動作し、Web サーバに直接接続しない代替アーキテクチャも用意されています。このバージョンの Web ゲートウェイを NSD といいます。この場合、NSD は Web ゲートウェイのコア機能を提供し、InterSystems IRIS との永続的な接続を維持します。Web サーバは、2 つのタイプがある小さなモジュール（ホスト Web サーバの独自の API に対して動作するモジュールと CGI 実行可能コードとして実装されたモジュール）を介して NSD と通信します。したがって、NSD ベースのアーキテクチャは、CGI 標準によって Web サーバを拡張する要件がある場合または Web ゲートウェイの機能をホスト Web サーバの機能から切り離すことが望ましい場合に使用されます。

3

Web ゲートウェイのインストール

Web ゲートウェイは、構成ニーズに応じて、InterSystems IRIS インストールの一部としてインストールすることも、スタンドアロンでインストールすることもできます。

3.1 既定の Web ゲートウェイのインストールと構成

InterSystems IRIS をインストールし、[セットアップ・タイプ](#)として [開発]、[サーバ]、[カスタム] ([Web サーバ・ゲートウェイ] コンポーネントを選択)、または [Web サーバ] (Windows プラットフォームのみ) を選択すると、インストーラは既存のプロダクション Web サーバ (既定タイプの場合、Windows では IIS、Linux/UNIX®/macOS では Apache) を構成し、Web ゲートウェイをインストールします。この構成がご自身のユース・ケースに適している場合は、Web サーバをインストールしてから InterSystems IRIS をインストールすると、InterSystems IRIS インスタンスの作業 Web ゲートウェイ構成が提供されます (調整は不要です)。

ただし、別の組み合わせの Web サーバとプラットフォームを使用している場合、Web サーバ・アーキテクチャが非標準の場合、または上級ユーザが現在の環境をカスタマイズする場合は、このドキュメントで InterSystems IRIS に接続するように Web サーバと Web ゲートウェイを構成する手順、および Web ゲートウェイが提供するサービスを使用する手順を確認してください。

インターシステムズのファイル・タイプ (.csp、.cls、.zen、および .cxw) に加え、管理ポータルをホストする Web サーバは、追加ファイル・タイプ jpg、gif、png、svg、css、および js の要求についても、Web ゲートウェイを介してルーティングする必要があります。“[追加ファイル・タイプを渡すための Apache の構成](#)” (Apache の場合)、または “[UNginx での NSD の使用](#)” (Nginx の場合) を参照してください。

3.2 Web ゲートウェイ管理モジュール

ホスト Web サーバの API に直接機能する Web ゲートウェイ・アーキテクチャは、通常 2 つのモジュールから構成されます (これらのファイル名はオペレーティング・システムによって異なります)。

- ・ 管理モジュール (CSPmsSys.dll など)。Web ゲートウェイ[管理ページ](#)を提供します。
- ・ 実行時モジュール (CSPms.dll など)。CSP ファイルに対する要求の処理、および管理要求のロードと管理モジュールへのルーティングを担当します。

これらのファイルは同じディレクトリに置く必要があります。

3.3 Web ゲートウェイ・コンポーネントと物理的なインストール・パス

このドキュメントの後半のセクションでは、サポートされているすべての Web サーバを使用して Web ゲートウェイ・コンポーネントを構成する方法を説明します。また、インターシステムズのインストーラは、同じホスト上に存在する可能性のある**プライベート Web サーバ**とサードパーティの Web サーバ用に別個の Web ゲートウェイ・インストールを作成して維持します。ここで、サードパーティの Web サーバは、インターシステムズがインストールしたソフトウェアの一部ではない Web サーバを指します。

Web ゲートウェイ・コンポーネントの正確なインストール場所は、以下の場合には特に重要ではありません。

- ・ 物理的なインストール・パスが、ホスト Web サーバの適切な構成で指定されたものと一致する場合。
- ・ 個々のコンポーネントに対する必要なアクセスに関連するセキュリティ設定が、適切に調整されている場合。Web サーバは通常、それらがアクセスできるファイル（および実行できる実行可能ファイル）を慎重に制御できるようにロック・ダウンされているため、これは Web サーバが直接アクセスする Web ゲートウェイ・コンポーネントにとって特に重要です。Web サーバのコア実行可能ファイルにバインドされている Web ゲートウェイ・バイナリがアクセスする Web ゲートウェイの構成（およびログ）ファイルでは、セキュリティの考慮も重要であることに注意してください。
- ・ ホスト Web サーバのセキュリティ・ポリシーが配慮されている場合。一部の Web サーバ（特に Secure Linux (SELinux) に同梱の Web サーバ）は、自らのファイル・システム外に置かれたファイルにはアクセスできないように構成されています。この制限により、Web サーバに対応する Web ゲートウェイ・コンポーネントをインストール可能な場所が左右されます。

次の 4 種類の Web ゲートウェイ・コンポーネントを考慮する必要があります。

1. Web サーバがロードするバイナリ (API ベースの拡張)。

これには、Windows DLL、および UNIX® 共有オブジェクトが含まれます。

```
CSPms[Sys].dll
CSPn3[Sys].(dll|so|exe)
CSPa*[Sys].(dll|so)
CSPx[Sys].(dll|so)
mod_csp*. (dll|so)
```

これらがインストールされる物理的な場所は、ホスト Web サーバ構成の対応する構成の指示文と一致する必要があります。これには、ロードする必要があるサードパーティ・モジュールを示す指示文が含まれます。Web サーバは、これらのモジュールを読み取ってロードするためのアクセス許可を必要とします。**CSP*** という名前のモジュールには、Web ゲートウェイ**構成ファイルおよびログ・ファイル**に対する読み取りおよび書き込み許可が必要です。これらは通常、Web ゲートウェイ・バイナリと同じ場所に作成されます。

これらのモジュールへのアクセス制御を考慮する際、依存する構成ファイルおよびログ・ファイルと共にモジュールにアクセスできる必要があるのは、Web サーバのワーカ・プロセスであることに注意してください。例えば、Apache の場合は、サーバは通常、スーパーユーザの権限で開始されますが、実際に Web 要求を処理するワーカ・プロセスは、これよりはるかに下位の権限 (Apache 構成ファイル内で User および Group 指示文で指定) で実行されます。ワーカ・プロセスに指定されたユーザとグループには、Web ゲートウェイ・モジュールをロードするためのアクセス許可と (適切な場合は) **構成ファイルおよびログ・ファイル**の読み取りおよび書き込み許可を付与する必要があります。

2. Web サーバが呼び出す実行可能ファイル (Common Gateway Interface (CGI) モジュール)。すべての構成でこれらの実行可能ファイルが必要なわけではありません。

```
[nph-]CSPcgi[Sys][.exe]
```


これらがインストールされる物理的な場所は、ホスト Web サーバ構成の対応する構成の指示文と一致することが必要です。これには、これらの CGI モジュールが処理する必要がある Web 要求を示す指示文が含まれます。

ホスト Web サーバのワーカ・プロセスは、これらのモジュールに対する実行許可を必要とします。それ以上の依存関係はありません。

3. Web サーバが返す静的ファイル。

注釈 現在の Web ゲートウェイ構成では、CSP はしばしば、Web サーバから静的ファイルを返すのではなく、InterSystems IRIS から直接静的ファイルを提供するように構成されます。このページは、このような構成には適用されません。

JavaScript モジュール (CSPBroker.js、CSPxmlhttp.js など)

Java アプレット (CSPBroker.class、CSPBroker.jar など)

イメージ (created-with-csp.gif など)

ホスト Web サーバのワーカ・プロセスは、これらのファイルに対する読み取り許可を必要とします。

4. CSP Network Service Daemon (NSD)。

注釈 すべての構成がこの機能を必要とするわけではありません。

CSPnsd[Sv][.exe]

NSD は任意の場所にインストールすることができ、これと Web サーバとの通信は TCP、通常はポート 7038 によって行われるため、Web サーバがその物理的場所を認識する必要はありません。

NSD は、Web ゲートウェイ構成ファイルおよびログ・ファイルに対する読み書きの許可を必要とします。通常、それらは同じ場所に作成されます。

注釈 セキュリティの理由から、このモジュールは Web サーバからアクセスできる場所にはインストールしないでください。このモジュールは、手順 1、2、または 3 にリストされているモジュールと場所を共有しないでください。このドキュメントで説明されている多くの Web サーバ構成では、Web サーバがアクセスできるアクセス可能ファイルのリストからこのモジュールを明示的に除外しています。そうではあっても、ファイル・システムの他の場所に NSD を物理的にインストールする方が、はるかに安全です。

5. 永続ストレージに保持される Web ゲートウェイ・キャッシュ。

可能であれば、Web ゲートウェイは、大きいファイル (大きい JavaScript ファイルなど) のコンテンツを、付随する HTTP 応答ヘッダとともに永続ストレージに配置します。重要な制御情報 (有効期限など) およびキャッシュされたすべてのファイルのインデックスは、共有メモリ・セクタにあります。このアーキテクチャでは、キャッシュ・エントリそれぞれが (メモリ使用量に関して) 小さなキャッシュ・スペースを使用します。ほとんどの場合、エントリあたり 1 個のキャッシュ・ブロックを超えることはありません。

キャッシュされたコンテンツは、Web ゲートウェイの temp ディレクトリにあるタイプ .dat のファイルに保存されます。このディレクトリは、インストール・スクリプトによって、Web ゲートウェイのインストール・ディレクトリの直下に配置されます。例えば、通常の IIS インストールでは、C:\Inetpub\CSPGateway\temp となります。この場所は、ホスト Web サーバのワーカ・プロセスに対する完全な読み取り/書き込み/削除許可が必要です。

3.4 Web ゲートウェイのスタンドアロン・コンポーネントとしての導入

例えば、1 つ以上のリモート InterSystems IRIS インスタンスで使用するために、Web サーバをホストするシステム上に Web ゲートウェイを配置するような場合、次の方法で 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導入することができます。

- ・ “インストール・ガイド” の “InterSystems IRIS のインストール準備” の章の “**セットアップ・タイプ**” に説明されているように、InterSystems IRIS インストーラを使用して Web ゲートウェイを別個にインストールできます。それには、Microsoft Windows システムで [Web サーバ] セットアップを選択するか、任意のプラットフォームで [カスタム] セットアップを選択して、[Web サーバ・ゲートウェイ] コンポーネントを含めます。
- ・ スタンドアロン・インストーラを使用して、Web ゲートウェイを単独でインストールすることもできます。インストーラを取得するには、WRC のダウンロード・ページ (<https://wrc.intersystems.com/wrc/coDistGen.csp>) を使用してください。Web ゲートウェイ キットのみを表示するには、[名前] 列に `web gateway` と入力します。

InterSystems IRIS インストーラと同様に、Web ゲートウェイ・スタンドアロン・インストーラは実行可能ファイルとして提供され、UNIX® スタンドアロン・インストーラはコマンド行スクリプトとして提供されます。

Windows スタンドアロン・インストーラの既定値では IIS Web サーバが使用されていることを前提としており、UNIX® スタンドアロン・インストーラの既定値では Apache Web サーバが使用されていることを前提としています。Web サーバ・アーキテクチャが非標準の場合、または環境をカスタマイズしたい場合は、このドキュメントを注意深く読んで構成オプションについて理解してください。

- ・ インターシステムズが提供する **webgateway** コンテナ・イメージには、Web ゲートウェイと Apache Web サーバまたは Nginx Web サーバの両方が含まれており、それによって InterSystems IRIS ベースのアプリケーションにコンテナ化された Web サーバを提供しています。**webgateway** イメージからの Web ゲートウェイ・コンテナの導入の詳細は、“コンテナ内でのインターシステムズ製品の実行” の “[InterSystems Web ゲートウェイ・コンテナの使用法](#)”、“InterSystems Cloud Manager ガイド” の “[ロール WS : Web サーバ](#)”、および “InterSystems Kubernetes Operator の使用” の “[webgateway: Web サーバ・ノードの定義](#)” を参照してください。

Web サーバごとに別個の Web ゲートウェイのインストールが必要であることに注意してください。リモート Web サーバの構成の詳細は、“[リモート Web サーバでの Web アプリケーションの使用](#)” を参照してください。

4

プライベート Web サーバの使用または置換

InterSystems IRIS® は、管理ポータルの実行のみを目的として、プライベート Web サーバ (Apache の最小ビルド) を提供します。

重要 HTTP ベースのアプリケーション (REST、CSP、Zen、および SOAP over HTTP/HTTPS など) を導入する際には、プライベート Web サーバを使用しないでください。代わりに、[サポートされる Web サーバ](#)のいずれかをインストールおよび構成してください。

4.1 プライベート Web サーバの目的

プライベート Web サーバ (PWS) の目的は、InterSystems IRIS に接続して管理ニーズを満たすことのみです。このプライベート・バージョンの Apache をインストールすると、以下のことが可能になります。

- ・ 管理ポータルをすぐに使用できます。
- ・ 開発環境用にすぐに使用可能なテスト機能が用意されています。

プライベート Web サーバは自己完結型で、よく知られている通常の HTTP サーバ・ポート 80 以外の TCP ポートで待ち受けるように構成されています。同じホストで動作する他の Web サーバ環境に影響を及ぼすことはありません。

管理ポータルのエントリ・ポイントは、通常、CSP パスとファイル `/csp/sys/UtilHome.csp` を使用します。例えば、`http://127.0.0.1:52773/csp/sys/UtilHome.csp` のようになります。

4.2 プライベート Web サーバの制限

プライベート Web サーバについて：

- ・ プロダクション環境での使用には適していません。
- ・ 最小のセキュリティで実行され、セキュリティで保護された環境内でのみ使用するように構築および構成されています。
- ・ 管理ポータルへのパブリック・アクセスがなく、すべてのユーザが信頼され、セキュリティで保護された接続を介してサーバに接続している環境でのみ使用する必要があります。

また、通常、使用される構成は大量の HTTP 要求が想定されるアプリケーションには適しません。インターシステムズでは、管理ポータルで使用する場合にのみ、プライベート Web サーバをテストします。

このセクションでは、プライベート Web サーバの構成と通常のプロダクション・グレードの Apache インストールの構成の違いについて説明します。

4.2.1 Windows での制限

Windows ベースの Apache インストールでは、オペレーティング・システムの最適化方法により適した特殊なマルチスレッド形式の Apache マルチプロセッシング・モジュール (MPM) を使用します。したがって、Windows でのプライベート Web サーバの動作は、同時に発生する負荷を処理する機能に関する限り、プロダクション・グレードでの Apache の構築の動作と似ています。

高可用性およびプロダクション・グレードのセキュリティが必要になる場合、Web 情報の他のソースと統合する必要がある場合、または Web サーバに対する高度な制御を必要とする場合は、Apache の別のプロダクション・グレードの構築をお勧めします。この場合、独自のサーバで動作することが理想です。一方、予期される HTTP トラフィックが少量で、高可用性やセキュリティに対する要求がそれほどない場合、このような状況でプライベート Web サーバを導入することは適切です。

4.2.2 プライベート Web サーバの構築

(既定の) 完全な Apache サーバは、通常、次のコマンドのシーケンスを使用して作成されます。

```
./configure --prefix=<install-dir>
make
make install
```

一般的に、Apache の最小ビルドは以下のように作成されます。

```
./configure --prefix=/usr/iris/httpd --with-port=57773
--with-pcre=$srcdir/pcre
--enable-mods-static="log_config mime alias unixd authz_core"
--disable-ssl
--enable-so --without-gdbm --without-ndbm
--without-berkeley-db --with-included-apr --with-expat=builtin
--with-mpm=prefork --disable-shared
make
make install
```

ここでは、通常はプロダクション・グレードのインストールに必要なサービスの多くが除外されている点に注意してください。

このサーバを他の Web アプリケーションのホストに使用することは可能ですが、その目的では完全な独立した Web サーバのインストール環境を使用することを強くお勧めします。ホストの InterSystems IRIS インストールをアップグレードすると、管理ポータル of Apache インストールの構成に対する変更が上書きされる点に注意してください。

管理ポータルの Apache インストール環境は、次の Web ゲートウェイ・モジュールを使用して InterSystems IRIS と通信します。

4.2.3 UNIX® での制限

既定では、プライベート Web サーバは Apache グループの Prefork マルチプロセッシング・モジュール (MPM) を使用します。これは、非スレッド化サーバ・モデルです。同時に処理できる要求の数は、プール内の Apache ワーカー・プロセスの数に直接関係します。

プライベート Web サーバは、プールに対して作成を許可するワーカー・プロセスの数を 2 つまでとすることで、占有するフットプリントが最小限になるように構成されます。プライベート Web サーバに対する Apache 構成 (`httpd.conf`) の設定は以下ようになります。

```
MinSpareServers 1
MaxSpareServers 2
```

一方、プロダクション・グレードの構築に対する既定の Apache 構成は、通常以下のとおりです。

```
StartServers      5
MinSpareServers   2
MaxSpareServers   20
ServerLimit       256
MaxRequestWorkers 256
```

この構成によって、Apache は起動時に 5 つのワーカ・プロセスを作成でき、同時に発生する負荷の増加に伴って最大 256 個まで増やすことが可能です。構成においてこのような違いがあるために、プライベート Web サーバのパフォーマンスは、プロダクション・グレードでの Apache の構築のパフォーマンスよりも明らかに劣るようになります。このパフォーマンス不足は、同時に発生する負荷が増加するにつれて、より目立つようになります。ただし、プライベート Web サーバの構成を変更して、完全な Apache インストールの構成（上記）に合わせることができます。これらのパラメータを変更した後、Apache を完全に再起動する必要があります。

4.2.4 メモ

管理ポータルの Apache インストール環境は、次の Web ゲートウェイ・モジュールを使用して InterSystems IRIS と通信します。

- ・ Windows : CSPa24.dll および CSPa24Sys.dll
- ・ UNIX® : CSPa24.so および CSPa24Sys.so

4.3 別の Web サーバを使用した管理ポータルの実行

別の Web サーバを使用して管理ポータルをサポートすることもできます。

- ・ プライベート Web サーバ以外の Web サーバを使用して InterSystems IRIS のインスタンスを管理する場合は、ドキュメントへのリンクが引き続き機能するように Web サーバを構成する必要があります。そのためには、`/csp/docbook/` からドキュメントの正しい URL へのリダイレクトを含むように Web サーバを構成します。この情報は、Apache が `/csp/docbook/` をリダイレクトするために使用するファイル `install_dir/httpd/conf/httpd-doc.conf` にあります。リダイレクトの作成については、ご使用の Web サーバのドキュメントを参照してください。
- ・ プライベート Web サーバを無効にするには、`WebServer` CPF パラメータを 0 に設定します。

管理ポータルをスタンドアロン Apache で実行する場合は、追加のファイル・タイプを手動で構成する必要があります。[追加ファイル・タイプを渡すための Apache の構成](#) を参照してください。

4.4 プライベート Web サーバの管理

通常の動作環境では、InterSystems IRIS を起動すると特定の InterSystems IRIS インスタンスのプライベート Web サーバも起動し、InterSystems IRIS を停止すると Web サーバも終了します。プライベート Web サーバの構成を変更する場合など、場合によっては、対応する InterSystems IRIS サーバを中断せずにプライベート Web サーバを再起動する必要がありますが生じることがあります。

4.4.1 構成の変更

プライベート Web サーバの構成ファイルは、`install_dir/httpd/conf/httpd.conf` にあります。ただし、このファイルに加えた編集は、ソフトウェア・アップグレードでは保持されません。アップグレード時に保持する構成変更のために、構成ファ

イル `install_dir/httpd/conf/httpd-local.conf` が提供されています。このファイルがまだ存在しない場合は、新規インストール時またはアップグレード時に作成されます。存在する場合、アップグレードで内容に変更は加えられません。

4.4.2 Windows でのプライベート Web サーバの管理

プライベート Web サーバの起動：

```
<install-dir>\httpd\bin\httpd -k start -n <instname>httpd
-c "Listen port"
```

プライベート Web サーバの停止：

```
<install-dir>\httpd\bin\httpd -k stop -n <instname>httpd
```

例えば、以下のとおりであるとします。InterSystems IRIS のインストール場所：**C:\iris**

InterSystems IRIS のインスタンス名：**IRIS**

Apache の TCP ポート：**52773**

起動：

```
C:\iris\httpd\bin\httpd -k start -n IRIShttpd -c "Listen 52773"
```

停止：

```
C:\iris\httpd\bin\httpd -k stop -n IRIShttpd
```

4.4.3 Unix® でのプライベート Web サーバの管理

プライベート Web サーバの起動：

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

プライベート Web サーバの停止：

```
kill `cat <install-dir>/httpd/logs/httpd.pid`
```

例えば、InterSystems IRIS が **/usr/iris** にインストールされており、Apache の TCP ポートが **8972** であるとします。この場合、プライベート Web サーバは以下のようにして起動できます。

```
/usr/iris/httpd/bin/httpd -d /usr/iris/httpd -c "Listen 8972"
```

また、以下のようにして停止できます。

```
kill `cat /usr/iris/httpd/logs/httpd.pid`
```

注釈 AIX の場合にこの方法で **httpd** を手動で実行するには、**LD_LIBRARY_PATH** に **install-dir/bin** ディレクトリを含める必要があります。

5

Web ゲートウェイと連携させるための IIS の構成 (Windows)

ここでは、InterSystems Web ゲートウェイで使用するための、既存の Microsoft IIS Web サーバの構成方法について説明します。(Windows では、その他の選択肢として、[Apache](#) と [Nginx](#) がありますが、この場合のオプションは[その他のオペレーティング・システム](#)とは異なります。)

Web ゲートウェイのスタンドアロン・インストーラを使用して Web ゲートウェイをインストールする場合は、既存の Microsoft IIS Web サーバを自動的に構成するオプションを選択できます。このオプションを選択しない場合は、Web サーバを手動で構成できます。その方法をこのページで説明します。

5.1 一般的な手順 (すべてのバージョン)

IIS を Web サーバとして使用している場合は、以下の手順に従って、Web サーバを構成します。

1. [Web ゲートウェイ・コンポーネントに対する許可を設定](#)します。
2. [Web アプリケーション・パスを構成](#)します。
3. `/bin` を含む URLを有効化します。

IIS 7 以降を使用している場合は、[Microsoft IIS 7](#) のセクションを参照してください。一般的ではない構成がある場合は、“[IIS 7 以降の代替オプション](#)” を参照してください。

5.1.1 Web ゲートウェイ・コンポーネントに対する許可の設定

選択した Web ゲートウェイ構成オプションに関係なく、標準の IIS ドキュメント・ルート (例 : `C:\inetpub\wwwroot`) 以外の場所に保持されている Web リソースに適切な許可を割り当てる必要があります。

既定では、IIS 7 は Web アプリケーションのユーザに対して、あらかじめ構成されたドキュメント・ルートの範囲外にあるものへのアクセスを一切許可しません。ただし、以下のユーザまたはグループに対して、このような外部リソースへの読み取りと実行および書き込み許可を割り当てている場合は例外です。

`[machine_name]\IIS_IUSRS`

および、

`[machine_name]\Users`

IIS_IUSRS は IIS ワーカー・プロセスが動作するユーザ (グループ) を表していることに注意してください。これは、バージョン 7 より前の IIS バージョンで使用されていた IUSR_[machine_name] ユーザ・グループに替わるものです。IIS を通じてコントロールされるアプリケーション (Web ゲートウェイなど) は、IIS_IUSRS に割り当てられた特権のレベルで動作します。

CSP では、Web サーバのルートの外側にあるリソースには、通常、以下のものがあります。

Web ゲートウェイのバイナリ・コンポーネント：

C:\inetpub\CSPGateway

静的ファイル・コンポーネント：

install-dir\CSP¥

Windows エクスプローラ経由でこれらのフォルダに手動で特権を割り当てるには、以下の手順に従って操作します。

1. フォルダ名を右クリックし、[プロパティ] を選択します。
2. [セキュリティ] タブを選択します。
3. [編集] を選択します。
4. [追加] を選択します。
5. [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
6. [名前の確認]、[OK] の順に選択します。
7. [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。
8. [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。
9. [適用]、[OK] の順に選択します。
10. [[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。

IIS ユーザ・グループには、Web ゲートウェイの構成ファイルとログ・ファイルに対する完全な読み取りおよび書き込み許可が必要です。例えば、Windows のコマンド・プロンプトで、以下を入力します。

```
cacls CSP.ini /E /G IIS_IUSRS:F
```

```
cacls CSP.log /E /G IIS_IUSRS:F
```

もちろん、この操作は Windows エクスプローラ経由でもできます。

5.1.2 Web アプリケーション・パスの構成

このセクションでは、IIS で Web アプリケーション・パス (csp など) を構成するための手順について説明します。これらの手順は、IIS のすべての Web ゲートウェイ構成オプションに共通です。

IIS の構成は、[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。ドキュメント・ルートの下で構成したサブディレクトリは、[Virtual] または [Application] に分類されます。[Virtual] サブディレクトリ (またはエイリアス) は物理的な同等物 (Windows のディレクトリ) にマップされます。同じことは Applications に分類されたサブディレクトリにもいえますが、ただし、物理的な同等物が定義できるほか、アプリケーションを特定のアプリケーション・プールに関連付けられるという点が異なります。既定値は DefaultAppPool です。

Web アプリケーションには Web ゲートウェイを通じてサービスが提供されるので、ホスト・サブディレクトリ (/csp など) は [Applications] として構成する必要があります。

既定の CSP 構成では、/csp アプリケーション・パスは物理位置 install-dir¥CSP にマップされます。静的ファイルはすべて、このルートの下 (¥csp¥broker...) に置かれます。

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、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
```

3. 右側のパネルで、**[アプリケーションの表示]** を選択します。
4. もう一度、右側のパネルで、**[アプリケーション追加]** を選択します。
5. **[アプリケーション追加]** ダイアログに以下のように入力します。

エイリアス : **csp**

物理パス : **install-dir¥CSP¥**

6. **[OK]** を選択します。

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。以下はその例です。

アプリケーション **/csp/bin** を **C:¥Inetpub¥CSPGateway** にマップします。

5.1.3 /bin を含む URL の有効化

InterSystems IRIS インストーラを使用して Web ゲートウェイをインストールしている場合、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には、この手順を実行する必要があります(詳細およびこの手順に代わる方法については、次の外部 Web サイト <https://weblogs.asp.net/owscott/iis7-blocks-viewing-access-to-files-in-bin-and-other-asp-net-folders> を参照)。**/bin** を含む URL を有効にするには、**applicationHost.config** ファイルに以下の位置タグを追加します。

```
<location path="sitename.com/subfolder/bin/debug">
  <system.webServer>
    <security>
      <requestFiltering>
        <hiddenSegments>
          <remove segment="bin" />
        </hiddenSegments>
      </requestFiltering>
    </security>
  </system.webServer>
</location>
```

5.1.4 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が、その結果、アクティブ Web ゲートウェイの再初期化が行われることはありません(つまり、Web ゲートウェイの DLL は再ロードされません)。

Web ゲートウェイ・モジュールを再ロードするために IIS を強制的に再起動する場合、メインの Windows **[サービス]** コントロール・パネルを使用して **World Wide Web Publishing** サービスを再起動する必要があります。

5.1.5 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。

5.2 追加手順 : Microsoft IIS 7 以降

Microsoft ISAPI 拡張 (CSPms.dll、CSPmsSys.dll、および CSPcms.dll) は、IIS 7 以降のネイティブ・モジュール・インタフェースに直接作用できるように構成されています。

InterSystems IRIS で提供される Web ゲートウェイ・モジュールは、IIS 7 のネイティブ・モジュールと共に動作できます。また、ISAPI 拡張機能と組み合わせて使用することもできます。NSD を使用しているユーザ用に追加の構成オプションがあります。このセクションでは、ネイティブ・モジュールで動作するように IIS 7 Web サーバを構成する方法を説明します。その他の構成については、["IIS 7 以降の代替オプション"](#)を参照してください。

5.2.1 IIS 7 のインストール場所

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. ネイティブ・モジュールの既定の場所
 - ・ CSPms.dll (実行時モジュール)

- ・ CSPmsSys.dll (システム管理モジュール)

これらのモジュールの既定の場所は以下のとおりです。

C:\inetpub\CSPGateway

NSD ベース以外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。

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以下のとおりです。

install-dir\csp\broker

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以下のとおりです。

install-dir\csp\sys

5.2.2 推奨オプション : ネイティブ・モジュールの使用 (CSPms*.dll)

これは最も多く使用されている推奨オプションです。これは、IIS 7 で導入されたネイティブ・モジュール・インタフェースを使用します。このオプションでは最適なパフォーマンスが得られます。

その他の構成オプションについては、“[IIS 7 以降の代替オプション \(Windows\)](#)” および “[NSD の使用法 \(Windows\)](#)” を参照してください。

[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、ネイティブ・モジュールの登録および Web サーバの構成を行います。インストールで必要となる可能性のある追加ファイル (例えば、分析に必要な特別な CSP リソース) をすべて含めます。

5.2.2.1 ネイティブ・モジュールの登録

DLL : CSPms.dll および CSPmsSys.dll

これら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]** (または **[ネイティブ モジュールの構成]**) を選択します。正確な表現は使用中の IIS のビルドによって異なります。
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

パス : C:\inetpub\CSPGateway\CSPms.dll

[OK] を選択します。

- 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。[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
```

- 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。
- [CSPms] を選択して、[OK] を選択します。

5.2.2.2 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 ゲートウェイのネイティブ・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	CSPms	C:\Inetpub\¥CSPGateway¥CSPms.dll

これらのマッピングは REST をサポートしていません。REST を使用する場合は、“**追加ファイル・タイプのマッピング**”を参照してください。

- [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、[Web サイト]、[既定の Web サイト] セクションの順に展開します。[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。

- 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。
- 右側のパネルで、[モジュール マップの追加] を選択します。
- [モジュール マップの追加] ダイアログに以下の詳細を入力します。

要求パス : *.csp

モジュール : (ドロップダウン・リストから [CSPms] を選択します)

名前 : WebGateway_csp

- [要求の制限] を選択し、[要求がマップされている場合のみハンドラを呼び出す] の横にあるボックスにチェックが付けられていないことを確認します。このアクションにより、[ハンドラ マッピング] テーブルに示すように、[パスの種類] の値が [未指定] 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. 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 : (リストから [CSPms] を選択します)

名前 : WebGateway_cls

要求パス : *.zen

モジュール : (リストから [CSPms] を選択します)

名前 : WebGateway_zen

要求パス : *.cxw

モジュール : (リストから [CSPms] を選択します)

名前 : WebGatewayManagement

5.2.2.3 追加ファイル・タイプのマッピング

追加ファイル・タイプを Web ゲートウェイに送信するように Web サーバを構成するには、追加のファイル拡張子に対し、既定のエントリと同様の構成エントリを追加します。以下に例を示します。

拡張子	ネイティブ・モジュール	バイナリ
*.xxx	CSPms	C:\Inetpub\CSPGateway\CSPms.dll

iWeb ゲートウェイを経由して他の静的ファイル进行处理する必要がある場合、必要なファイル・タイプのマッピングを追加して、それらのファイルをロードできるようにします。特に、この Web サーバを経由して管理ポータルにアクセスする必要がある場合は、ファイル・タイプ **jpg**、**gif**、**png**、**css**、および **js** のマッピングを追加します。追加しない場合、管理ポータルのイメージ、Javascript、およびそのスタイル・シートは欠落します。

指定されたパスに対するすべての要求を送信するように Web サーバを構成するには、このパスについて、以下のワイルドカードをセットアップします。

拡張子	ネイティブ・モジュール	バイナリ
*	CSPms	C:\Inetpub\CSPGateway\CSPms.dll

重要 REST を使用する場合は、このマッピングを使用する必要があります。

5.2.2.4 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、ブラウザで以下の場所を指定します。

http://<ip_address>/csp/bin/Systems/Module.cxw

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、“[Web ゲートウェイとセキュリティ](#)”のセキュリティに関するメモを参照してください。

5.2.3 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わりに方法は、**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. 構成の変更後に、IIS を再起動します。

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/en-us/iis/configuration/system.webServer/httpErrors/>

6

Web ゲートウェイと連携させるための Apache の構成 (Windows)

ここでは、Windows 上の InterSystems Web ゲートウェイで使用するための、Apache Web サーバの構成方法について説明します。(Windows では、その他の選択肢として、IIS と Nginx がありますが、この場合のオプションは**その他のオペレーティング・システム**とは異なります。)

InterSystems IRIS® インストーラを使用して Web ゲートウェイをインストールした場合、または CSP で動作するように Apache サーバを構成する場合は、このページの手順に従ってください。

Apache Web サーバを使用している場合は、以下の指示に従って、ネイティブ・モジュールを使用して Apache を構成します。ネイティブ・モジュールは、ダイナミック・リンク・モジュール (DLL) として実装された拡張機能です。これにより、ISAPI (Internet Server Application Programming Interface) の拡張機能を利用できます。ISAPI は、Microsoft Web サーバ用に開発された高パフォーマンスの API です。

Apache 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。

Apache の完全なソース・コードが、サーバの構築に関する明確な解説と共に Apache からダウンロード可能です。Apache を Windows で構築するには、Microsoft C コンパイラ (Visual C++) のバージョン 5.0 以降が必要です。サーバを自身で構築する代わりに、Windows 用のビルド済みキットをダウンロードできます。このビルド済みキットは通常、最新の Apache ソース・コードよりも数回前のビルドとなります。

最初に、“**Apache サーバでのインストール場所 (すべての構成)**” の指示に従い、次に “**推奨オプション : Apache API モジュール (CSPa24.dll)**” の指示に従います (一般的ではない構成をインストールしている場合は、“**Windows Apache の代替オプション**” を参照してください)。

6.1 想定

ここでは、以下のように想定しています。

- Web ゲートウェイ・コンポーネントは **C:\Program Files\Apache Group\Apache\WebGateway** にインストールされている
- Web サーバは **C:\Program Files\Apache Group\Apache** 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。

6.2 Apache サーバでのインストール場所 (すべての構成)

Apache サーバのユーザはすべて、このセクションの指示に従う必要があります。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. CGI およびその他のダイナミック・リンク・モジュール :

すべての Apache バージョンの共通ファイルには、以下のものがあります。

- ・ **CSPcgi.exe** (実行時モジュール)
- ・ **nph-CSPcgi.exe** (CSPcgi.exe のコピー)
- ・ **CSPcgiSys.exe** (システム管理モジュール)
- ・ **nph-CSPcgiSys.exe** (CSPcgiSys.exe のコピー)

Apache バージョン 2.4.x の個別のバイナリは、以下のとおりです。

- ・ **mod_csp24.dll** (DLL として機能する Apache 組み込み **module** が用意されている場合はそれを使用)
- ・ **CSPa24.dll** (実行時モジュールが用意されている場合はそれを使用)
- ・ **CSPa24Sys.dll** (ゲートウェイのシステム管理モジュールが用意されている場合はそれを使用)

これらのバイナリの既定の場所は **C:\Program Files\Apache Group\Apache\WebGateway\bin** です。

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。

NSD ベース以外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。

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するための特別なモジュールです。実行時モジュール (**Sys** が追加されていないモジュール) はシステム管理のフォームにアクセスできません。

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。

3. CSP サンプルで使用するその他の静的リソース

CSP サンプルでは、(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\samples** です。

4. 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。

6.3 推奨オプション : Apache API モジュール (CSPa24.dll)

これは管理ポータルを提供するプライベート Web サーバで使用するオプションです。

この比較的新しい接続オプションは、最適なパフォーマンスをもたらすだけでなく、構成が最も簡単です。Windows で実行する Apache は完全にマルチスレッド化されており、そのモジュールは Apache の起動後メモリに永続します。これら 2 つの重要な特性により、Web ゲートウェイの機能を一連のスタンドアロン・モジュールとして実装することができます。

一般的ではない構成をインストールしている場合は、“[Apache の代替オプション \(Windows\)](#)” を参照してください。

CSPap*.dll (実行時) モジュールと **CSPapSys*.dll** (Web ゲートウェイのシステム管理) モジュールは、対応する Microsoft ISAPI DLL 同様に機能するように設計されたダイナミック・リンク・モジュールです。Apache 2.4.x では、これらのモジュールの名前は **CSPa24.dll** および **CSPa24Sys.dll** のようになります。

[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・モジュール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 **C:\Program Files\Apache Group\Apache\conf** にあります。

1. Apache 2.4.x : 以下のセクションを **httpd.conf** の最後に追加します。

```
LoadModule csp_module_sa c:/iris/csp/bin/CSPa24.dll
CSPFileTypes csp cls zen cxw
Alias /csp/ c:/iris/csp/
<Directory "c:/iris/csp">
    AllowOverride None
    Options MultiViews FollowSymLinks ExecCGI
    Require all granted
    <FilesMatch "\.(log|ini|pid|exe)$">
        Require all denied
    </FilesMatch>
</Directory>
```

2. **httpd.conf** の変更後に、Apache を再起動します。
3. 必要に応じて、“[追加ファイル・タイプを渡すための Apache の構成](#)” を参照してください。

これで、[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。

7

追加ファイル・タイプを渡すための Apache の構成 (すべてのプラットフォーム)

すべてのプラットフォームで、Apache API モジュールは常に次の [インターシステムズのファイル・タイプ](#) の要求を認識し、それらを Web ゲートウェイに送信します。

`.csp .cls .zen .cxw`

適切な InterSystems IRIS インスタンスの [CSP サーバ](#) にルーティングするため、[Web ゲートウェイに送信](#) する別のファイルの要求がある場合もあります。例えば、この Web サーバを経由して管理ポータルにアクセスする必要がある場合は、静的ファイル・タイプ `.jpg`、`.gif`、`.png`、`.svg`、`.css`、および `.js` のマッピングも追加する必要があります。

次のいずれかの方法で追加ファイルの要求を Web ゲートウェイに渡すよう、Apache を構成できます。

- ・ [CSP の Location 指示文の使用](#)
- ・ [ファイル拡張子の使用](#)
- ・ [MIME タイプの使用](#)

注釈 このページで説明する CSP 指示文は、`<VirtualHost>` 構成の一部として呼び出すことはできません。

7.1 Location による Apache の構成

`<Location>` ブロック内で CSP 指示文を使用することで、特定のディレクトリにあるリソースのすべての要求を Web ゲートウェイに渡します。以下により、Apache は `/csp` ディレクトリにあるリソースのすべての要求を Web ゲートウェイに転送します。

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

例えば、次のすべてのリソースに対する要求が Web ゲートウェイに送信され、適切な InterSystems IRIS インスタンスにルーティングされます。

```
/csp/
/csp/samples/menu.csp
/csp/sys/
```

7.2 ファイル拡張子による Apache の構成 (CSPFileTypes 指示文)

CSPFileTypes 指示文は、特定のタイプのファイルの場合に、ファイルの要求を Web ゲートウェイにルーティングするよう Web サーバを構成します。この指示文は、拡張子を持つファイル (/csp/menu.csp など) に対する要求にのみ機能します。ファイル拡張子を持たないリソース (/csp/menu など) には機能しません。

次の指示文により、Apache はファイル・タイプ **xxx** および **yyy** のファイルの要求を Web ゲートウェイに渡します。

```
CSPFileTypes xxx yyy
```

この指示文を、サーバ全体に対してグローバルに発行することも、<Location> または <Directory> ブロックに限定することもできます。例えば、以下により Apache は、要求が /csp ディレクトリの下 (/csp/samples などのサブディレクトリを含む) に送信される場合のみ、タイプ **xxx** と **yyy** のファイルの要求を Web ゲートウェイに渡します。

```
<Location /csp/>
    CSPFileTypes xxx yyy
</Location>
```

次のように CSPFileTypes 指示文でワイルドカード文字 (*) を使用すると、Apache は /csp ディレクトリ (/csp/samples などを含む) にあるすべてのタイプのファイルの要求を Web ゲートウェイに渡します。

```
<Location /csp/>
    CSPFileTypes *
</Location>
```

7.3 MIME タイプによる Apache の構成

CSP エンジン¹は、前述のファイル拡張子に加えて、以下の MIME タイプのファイルを認識することもできます。

```
application/x-csp
```

および

```
text/csp
```

例えば、InterSystems IRIS によって (特に CSP エンジン¹を使用して) 処理されるファイルのリストにファイル拡張子 **xxx** を追加するには 以下のように指定します。

```
LoadModule csp_module_sa /iris/csp/bin/CSPa24.dll
AddType application/x-csp csp cls zen xxx
```

MIME タイプを使用してファイルの種類を CSP と関連付けた場合の問題の 1 つに、リソースへのパス (ホスト・ディレクトリ) が物理的に存在することを Apache が確認し、存在しない場合には「`404 Not Found`」というエラーを返すことがあります。しかし、要求されたファイルが物理的に存在することを保証するためのチェックは行われません。これは、CSP エンジン¹によりサービスを受けるリソースには適切です。なぜならば、このようなリソースは Web サーバからは仮想的である可能性があるからです。したがって、「MIME タイプによる指定」というアプローチは、アプリケーションのパス構造を Web サーバ上に複製できる場合のみ適しています。

8

Web ゲートウェイと連携させるための Nginx の構築と構成 (Windows)

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。(Windows では、その他の選択肢として、[Apache](#) と [IIS](#) がありますが、この場合のオプションは[その他のオペレーティング・システム](#)とは異なります。)

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。

8.1 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・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。

8.2 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。

8.3 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/>

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_FORTIFY_SOURCE=1 \
--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`

8.3.1 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 --builddir=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_FORTIFY_SOURCE=1 --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。

8.4 Nginx での NSD の使用

[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。CSP on と同じ効果があります。

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。

8.4.1 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;
}
```


8.4.2 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**) 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;
    CSPNSD_pass localhost:7038;
}
```

8.4.3 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。

8.5 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

9

Web ゲートウェイと連携させるための Apache の構成 (UNIX®/Linux/macOS)

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、Apache Web サーバの構成方法について説明します。(これらのオペレーティング・システムでは、もう 1 つの選択肢として [Nginx](#) がありますが、この場合のオプションは[その他のオペレーティング・システム](#)とは異なります。)

Apache にはいくつかの接続オプションがあります。Apache 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。Apache グループは、ダイナミック・リンク・モジュール (DSO) として実装された拡張機能をサポートしています。Apache モジュールとして記述された拡張機能は、Apache コアに直接構築できます。これが推奨オプションです。その他のオプションには、[NSD](#)、[ロック・ダウン Apache Web サーバ](#)、および[その他いくつかの特殊 Apache 構成](#)があります。

一部の UNIX® システムではビルド済みキットが利用できますが、一般にこれは最新バージョンよりもいくつか前のバージョンをビルドしたものです。Apache の完全なソース・コードが、Apache サーバの構築に関する明確な解説と共にダウンロード可能です。このような目的で GNU C コンパイラ (gcc) を無料で入手できますが、Apache 構築プロセスは固有の C コンパイラを使用します。

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。ほとんどの Linux ディストリビューションには Apache が含まれています。IBM は、独自の UNIX® 実装である AIX に Apache を組み込んで提供しています。

注釈 場合によっては、macOS および AIX に事前インストールされているバージョンの Apache は、InterSystems IRIS での使用には適していないことがあります。代替の導入オプションについては、“[NSD の使用法 \(UNIX®/Linux/macOS\)](#)” を参照してください。

このセクションでは、Web ゲートウェイをインストールするための推奨オプションについて説明します。

1. すべての構成については、“[UNIX®, Linux, macOS での Apache のインストール場所 \(推奨オプション\)](#)” の指示に従います。
2. 次に、“[推奨オプション : NSD を使用しない Apache API モジュール \(CSPa24.so\)](#)” の説明に従います。

あまり一般的でないシナリオの場合は、“[ロック・ダウン Apache Web サーバ](#)” およびその他の[特殊 Apache 構成](#)を参照してください。

9.1 想定

ここでは、以下のように想定しています。

- Web ゲートウェイ・コンポーネントは `/opt/webgateway/bin/` にインストールされている。
- Apache は `/usr/apache/` 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。

9.2 UNIX®, Linux、macOS での Apache のインストール場所 (推奨オプション)

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。インストール・ディレクトリは、`/iris` です。

1. Apache バージョン 2.4.x では、ダイナミック・リンク・モジュール **CSPa24.so**。

InterSystems IRIS のアップグレード時に既存の Web ゲートウェイ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。

```
/opt/webgateway/bin
```

元の場所 (`/iris/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。

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以下のとおりです。

```
/iris/csp/broker
```

3. CSP サンプルで使用するその他の静的リソース

CSP サンプルでは、(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以下のとおりです。

```
/iris/csp/samples
```

4. 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以下のとおりです。

```
/iris/csp/sys
```

9.2.1 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代替オプション 1 の手順を実行する前に、共有オブジェクトを管理するためのモジュール (`mod_so`) が、使用している Apache ビルドに組み込まれていることを確認してください。Red Hat Linux でこのチェックを実行するには、以下のコマンドを発行します。

```
httpd -l
```

Ubuntu または SUSE でこのチェックを実行するには、以下のコマンドを発行します。

```
apache2 -l
```

これらのコマンドにより、Apache 内で現在使用可能なモジュールのリストが表示されます。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。

9.3 推奨オプション：NSD を使用しない Apache API モジュール (CSPa24.so)

このオプションは、管理ポータルで使用される [プライベート Web サーバ](#) の構成で使用されます。

この接続オプションは、最適なパフォーマンスをもたらすだけでなく、構成が最も簡単です。

このオプションを使用する前に、Apache v2.4 は部分的にマルチスレッド対応であり、マルチプロセスとマルチスレッド・サーバの組み合わせとして実装されていることを念頭に置く必要があります。つまり、実際の稼働では、Apache の子プロセスごとに Web ゲートウェイのインスタンスが 1 つあることを意味します。これ自体は問題ではありませんが、このアーキテクチャでは、Web ゲートウェイの各インスタンスが InterSystems IRIS 接続プールを各自で管理するため、Web ゲートウェイで使用する InterSystems IRIS (および InterSystems IRIS プロセス) への接続数の制御が難しくなります。

これら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。

CSPa24.so (実行時) モジュールおよび **CSPa24Sys.so** (Web ゲートウェイ・システム管理) モジュールは、ダイナミック・リンク・モジュール (DSO) です。

[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・モジュールに渡して処理するように Web サーバを構成します。Apache 2.4.x : **CSPa24.so** モジュールと **CSPa24Sys.so** モジュールを使用します。

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`

Ubuntu または SUSE 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/apache2/conf`

1. Apache 2.4.x : 以下を **httpd.conf** の最後に追加します。

```
LoadModule csp_module_sa /opt/webgateway/bin/CSPa24.so
CSPModulePath /opt/webgateway/bin/
CSPFileTypes csp cls zen cxw
Alias /csp/ /opt/webgateway/bin/
<Directory "/opt/webgateway/bin">
    AllowOverride None
    Options MultiViews FollowSymLinks ExecCGI
    Require all granted
    <FilesMatch "\.(log|ini|pid|exe)$">
        Require all denied
    </FilesMatch>
</Directory>
```

2. **httpd.conf** の変更後に、Apache を再起動します。
3. “[追加ファイル・タイプを渡すための Apache の構成](#)” を参照してください。

これで、[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。

10

Nginx の構築と構成 (UNIX®/Linux/macOS)

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。(これらのオペレーティング・システムでは、もう 1 つの選択肢として [Apache](#) がありますが、この場合のオプションは [Windows](#) とは異なります。)

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。

10.1 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。

10.2 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。

10.3 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/>

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`

10.3.1 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。例：

```
/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/
```

10.4 Nginx での NSD の使用

インターシステムズのファイル・タイプの要求（および、インターシステムズ・アプリケーションが処理する必要のあるその他の要求）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。CSP on と同じ効果があります。

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。

10.4.1 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;
}
```

10.4.2 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**) 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;
    CSPNSD_pass localhost:7038;
}
```

10.4.3 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。

10.5 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`

11

Web ゲートウェイの構成の基本

Web ゲートウェイと連携するように Web サーバを構成したら、必要に応じて Web ゲートウェイを構成します。管理ページを使用して、既定のパラメータ、個々のサーバ、および Web アプリケーションを構成できます。

ここでは、いくつかの一般的な構成トピックに関する追加情報を提供します。

11.1 Web ゲートウェイの構成ファイルとログ・ファイル

Web ゲートウェイでは、それぞれ `CSP.ini`、`CSP.log` という名前の構成ファイルとログ・ファイルを作成し、使用します。便宜のため、管理ページに、これらのファイルの場所が示されています。

Tip 通常は管理ページを介して構成を行うため、構成ファイルを直接編集する必要はほとんどありません。ドキュメントでは、必要に応じてその構成内の指示文について説明します。直接構成ファイルを編集する必要がある場合は、`%CSP.Mgr.GatewayMgr` のクラス・リファレンスを参照してください。メソッド `SetApplicationParams`、`SetDefaultParms`、および `SetServerParams` には、パラメータの簡単な説明が示されています。

11.2 自動的に InterSystems IRIS にルーティングされるファイル・タイプ

インターシステムズのファイル・タイプは、InterSystems IRIS で (特に、CSP エンジンによって) 処理されます。その他のファイル (静的ファイル) はすべて Web サーバまたは CSP エンジンによって処理されます。CSP エンジンは Web アプリケーションのパスに配置されているあらゆる種類のファイル进行处理できます (静的ファイルを含む)。静的ファイル进行处理するための CSP エンジンの設定により、アプリケーションの静的ファイルが存在する場所を表すために Web サーバ構成にエイリアスを作成する必要がなくなるので、Web アプリケーションの Web サーバ構成はより単純化されます。静的ファイルを提供するように CSP エンジンを設定することで、単一 (共通) の Web サーバで 2 つの異なるバージョンの InterSystems IRIS を提供する場合に、それぞれで異なるバージョンの特定の静的ファイル (例えば、ハイパーイベントのブローカ・コンポーネント) を必要とするといった競合上の問題を解決できます。

特定の Web アプリケーションの静的ファイルを CSP エンジンで処理するには、静的ファイルを (Web サーバ独自のドキュメント・ファイル・システムではなく) Web アプリケーションのファイル・システム内の、アプリケーションを構成する CSP ファイルに関する適正位置に配置します。

注釈 Zen をベースとしたアプリケーションを実行するには、[ファイルの提供] オプションを有効にし、Web サーバを適切に構成することが必要です。

11.3 InterSystems IRIS からの静的ファイルの提供

InterSystems IRIS が静的ファイルを提供できるように Web サーバおよび Web ゲートウェイのインストールを構成できます。InterSystems IRIS がすべてのコンポーネントをアプリケーションに提供するように、管理ポータルが構成されています。ただし、Web サーバが静的ファイル进行处理する役割を保持するように Web サーバを構成することも可能です。

InterSystems IRIS データベース・サーバはすべての CSP を処理します。また、[Web ゲートウェイ](#)経由で、Web アプリケーションにあらゆる種類の静的ファイル进行处理することもできます。標準の [Web アプリケーション](#)では、通常、Web サーバ (データベース・サーバではない) が静的コンテンツ进行处理します。

11.3.1 文字エンコードの指定

CSP エンジン、ストリーム・サーバを介して、JavaScript ファイルの文字エンコードの決定に関して、主要な Web サーバと同じ方法で静的ファイル进行处理します。最近の慣例では、すべての JavaScript ファイルが `application/javascript` の Content-Type としてマークされるため、ページで使用されるすべての JavaScript がこのとおりであることを確認してください。

JavaScript ファイルがこのようにマークされている場合、

- ・ ファイルに BOM (バイト・オーダー・マーク) が含まれていれば、これをブラウザが自動的に検出し、適切な文字セットを利用してこれを読み取ります。
- ・ ファイルに BOM が含まれていなければ、ブラウザはこのファイルを UTF-8 と見なします。

この動作をオーバーライドして、JavaScript ファイルに文字セットを指定する必要がある場合は、グローバル `^%SYS("CSP","MimeFileClassify","JS")` をリスト値 `$listbuild(contenttype, binary, charset)` に設定します。以下に例を示します。

```
SET ^%SYS("CSP","MimeFileClassify","JS") = $listbuild("text/javascript", 0, "ISO-8859-1")
```

これにより、古い Content-Type が設定され、ISO-8859-1 文字セットが使用されます。また、Cache 変換テーブルが空の文字列以外に定義されている場合、またはグローバル・ノード `^%SYS("CSP","DefaultFileCharset")` が NULL 値に設定されている場合、Cache はすべての JavaScript およびその他のテキスト・ファイルにこの文字セットを使用します。既定では、これらのグローバル・ノードはどちらも設定されていません。

11.3.2 [静的ファイルの提供] オプションの有効化

[Web アプリケーション](#)の [静的ファイルの提供] オプションには、以下の値を指定できます。

- ・ 常に – [静的ファイルの提供] はオンです。
- ・ いいえ – [静的ファイルの提供] はオフです。
- ・ 常時かつキャッシュ – Web ゲートウェイは Web サーバ上で静的ファイルをキャッシュできます。この設定を使用すると、システムは InterSystems IRIS サーバに戻ることなく、キャッシュされた静的ページを提供できるため、効率が向上します。
- ・ インターシステムズのセキュリティを使用 – InterSystems IRIS インスタンスが Web アプリケーションの一部として機能する動的な `.csp` または `.cls` ページにアクセスするための適切な承認をユーザが持っている場合、そのインスタンスの CSP エンジン、その Web アプリケーションの場所にある静的ファイルに対する要求进行处理します。ユーザに適切な承認がない場合、CSP サーバは HTTP 404 応答を返します。

11.3.3 InterSystems IRIS が静的ファイルを処理できるようにするための Web サーバの構成

既定では、インターシステムズが提供するインストール・スクリプトは、外部 Web サーバを検索または構成しません。カスタム・インストールを実行する場合は、以前にインストールされた IIS または Apache Web サーバを構成して [CSP エンジン](#) のサポートを有効にするオプションを選択できます。インストール・スクリプトは、`/csp` 仮想ディレクトリを作成して、Web ゲートウェイによって処理される [インターシステムズのファイル・タイプ](#) のマッピングを作成します。CSP エンジンがその Web サーバを使用して正常に機能するには、これで十分です。**[静的ファイルの提供]** 機能のサポートは有効になりません。Web サーバを手動で構成して、Web ゲートウェイで処理される特定のファイル拡張子をマップする必要があります。このように設計されている理由は、このメカニズムによってデータベース・サーバを開いて公開するのはセキュリティ上のリスクであり、バックグラウンドで実行されてはならないからです。

Web サーバの手順については、“[追加ファイル・タイプのマッピング](#)” を参照してください

11.3.4 Web サーバからの静的ファイルの処理

Web サーバから静的ページを提供する従来の構成を使用できます。この場合、**[静的ファイル]** オプションの設定は関係ありません。これにより、共通の Web サーバで 2 つの異なるバージョンの InterSystems IRIS を提供する場合に、それぞれが異なるバージョンの特定の静的ファイル（例えば、ハイパーイベントのブローカ・コンポーネント）を必要とするといった競合上の問題を解決できます。

Web サーバ自体で静的ファイルを提供するように構成する場合は、その静的コンテンツがすべての Web サーバに存在するようにしてください。

11.4 高可用性ソリューションのハードウェア・ロード・バランサでのスティッキー・セッションの有効化

CSP 上で実行している高可用性ソリューションについては、ハードウェア・ロード・バランサを使用して負荷分散とフェイルオーバーを行うことをお勧めします。この場合のロード・バランサでは、スティッキー・セッションのサポートを有効にする必要があります。これにより、Web ゲートウェイの指定したインスタンスと指定のアプリケーション・サーバとの間でセッションを確立しておけば、そのユーザがそれ以降に発行するすべての要求は、必ず同じインスタンスとサーバの組み合わせで処理されるようになります。この構成では、セッション ID とサーバ側のセッション・コンテキストが常に同期していることが保証されます。この構成を使用しない場合は、セッションがあるサーバ上で生成されたにもかかわらず、そのユーザからの次の要求はこのセッションが存在しない別のシステムで処理される可能性があり、その結果として実行時エラーが発生します（特に、要求を復号化するためにセッション・キーを必要とするハイパーイベントを使用している場合）。スティッキー・セッションのサポートを有効にする方法については、使用しているロード・バランサのドキュメントを参照してください。

注釈 スティッキー・セッションを使用しなくてもシステムが動作するように構成することは可能ですが、そのためには、Web セッション・グローバルを企業内のすべてのシステムにわたってマップする必要があります。その結果、重大なロック競合が発生する可能性があるため、このような構成はお勧めできません。

11.5 Web ゲートウェイ構成を再起動するスクリプトの有効化

InterSystems IRIS の外部にあるスクリプトを有効にして、Web ゲートウェイの構成を再起動することができます。

スクリプトでは、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 [Web ゲートウェイ・イベント・ログ](#) に書き込まれます。

```
Gateway Management
Gateway Configuration Reloaded and Reactivated
```

11.6 マルチプロセス/マルチスレッドのハイブリッド型 Web サーバ・アーキテクチャ

Web ゲートウェイには、マルチプロセス/マルチスレッドのハイブリッド型 Web サーバ・アーキテクチャの拡張サポートが含まれています。UNIX® の Apache バージョン 2.4 は、このアーキテクチャに従って実装された Web サーバの例です。

コア Web ゲートウェイ・リソースは、共有メモリ・セクタに保持されるようになりました。すべての Web サーバのワーカ・プロセスは、共通の実行中構成、接続テーブルおよびフォーム・キャッシュを共有します。Web ゲートウェイ・システム・ステータス・フォームには、単一のワーカ・プロセスのステータスのみではなく、Web サーバのインストール全体のステータスが表示されます。ステータス・フォームの接続テーブルには、InterSystems IRIS への各接続に対応する Web サーバ・プロセス ID を含む追加の列が含まれます。

最後に、ステート認識セッションはマルチプロセス・アーキテクチャでサポートされます。(InterSystems IRIS への) 接続プールはいくつかの Web サーバ・プロセス間で配布されますが、Web ゲートウェイは IPC (InterProcess Communications) プロトコルを使用して、ステート認識セッションの要求を Web サーバ環境の正しいホスティング・プロセスにルーティングします。

12

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。

12.1 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。例えば、以下のようになります。

`http://localhost:<port_no>/csp/bin/Systems/Module.cxw`

必ず **cxw** ファイル拡張子を含めてください。(また、Apache を使用している場合、Apache では URL のパス名とファイル名で大文字と小文字が区別されることに注意してください。)

InterSystems IRIS Web サービス・サーバの既定のポート番号の詳細は、“構成パラメータ・ファイル・リファレンス” の “**WebServerPort**” エントリを参照してください。

これらのページへのアクセス時には、ユーザ名とパスワードの入力を求められます。**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、インターシステムズ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。

12.2 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のため、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。

12.3 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。

メニュー項目	アクション
[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。

12.4 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、以下のようにします。

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、以下のようにします。

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

13

Web ゲートウェイの既定パラメータの構成

ここでは、Web ゲートウェイ管理ページから、InterSystems IRIS® Web ゲートウェイの既定のパラメータを構成する方法について説明します。サーバおよびアプリケーションの構成方法については、別の項目で説明します。

Web ゲートウェイ管理ページの [デフォルトパラメータ] オプションでは、Web ゲートウェイのすべてのグローバルな (システム全体の) 構成パラメータを管理します。このオプションを使用するには、システム管理者である必要があります。

特定の InterSystems IRIS サーバへのアクセスを構成する場合、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・フォームはグローバル構成から自動的に継承されます。例えば、特定のサーバに [サーバ応答タイムアウト] のパラメータを設定していない場合でも、そのサーバはグローバルな [サーバ応答タイムアウト] の設定を継承します。

13.1 Web ゲートウェイ

このセクションでは、Web ゲートウェイ全体に対してグローバルに関係するパラメータについて説明します。

[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 (CSPWSERVERID) 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。このルールの例外は、アプリケーションがセッション・クッキーを使用 しよう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まれます (以前と同じ)。

13.2 セキュリティ

ここでユーザ名とパスワードを定義しておくと、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. **構成ファイル**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
```

以下の CSP 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。

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.cmx">  
<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>
```

13.3 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

い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なります。

```
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、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)
```

13.4 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。

13.5 [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](#) を処理する場合は、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/
```

13.6 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。

13.7 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。

ログ・オプション	機能
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。
V2	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。 <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	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。 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。
V4	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。 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。
V5	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。 <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; 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。

ログ・オプション	機能
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>
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 c 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 C 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 c 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 c3 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・ログ設定にのみ適用されます。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。w 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。W 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 = 0</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>

14

サーバ・アクセスの構成

このページでは、InterSystems IRIS® [Web ゲートウェイ](#)の接続先サーバを構成する方法について説明します。このような構成タスクでは、Web ゲートウェイの[管理ページ](#)を使用します。[既定の設定](#)および[アプリケーション](#)の構成方法については、別の項目で説明します。

Web ゲートウェイからアクセスする各 InterSystems IRIS システムをここ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。

14.1 サーバ構成の追加

InterSystems IRIS サーバへのアクセスを構成するには、以下の手順で行います。

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. このサーバ構成のシステム・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。

14.1.1 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構成を識別するための論理名。
[サービス状態]	この構成を有効および無効にすることができます（既定は [有効] ）。
[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。

サーバ構成のパラメータ	機能
[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、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・メンバーのいずれかの 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・メンバーから、ミラーにあるフェイルオーバー・メンバーおよび災害復旧 (DR) 非同期メンバーのリストを取得し、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・メンバーの中になければ、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、災害復旧の状態で 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>

14.1.2 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]	<p>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。</p>
[サーバ接続最大数]	<p>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。</p>
[セッションあたりの最大接続数]	<p>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。</p>

14.1.3 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 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、以下のセクションで説明します。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。
[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。

14.1.4 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。

14.1.5 [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)。

14.1.6 [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。

14.2 サーバ構成のコピー

新しいサーバを簡単に構成するには、既存のサーバの構成エントリをコピーします。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)。

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。

既存のサーバ構成をコピーするには、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。

14.3 構成した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。

14.4 サーバ構成の削除

構成したサーバを削除するには、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。

15

アプリケーション・アクセスの構成

このページでは、InterSystems IRIS® Web ゲートウェイの接続先アプリケーションを構成する方法について説明します。このような構成タスクでは、Web ゲートウェイの管理ページを使用します。既定の設定およびサーバの構成方法については、別の項目で説明します。

Web アプリケーションごとに、CSP ファイルへのパスを構成する必要があります。各パスの構成によって、アプリケーションの実行を扱う InterSystems IRIS サーバが識別されます。フェイルオーバーや負荷分散を指定するオプションの指示文を、アプリケーション・パスの構成で指定します。既定のアプリケーション・パス、**root (/)** は、Web ゲートウェイを初めて起動するときに自動的に構成されます。継承がアプリケーション・パスに適用されます。例えば、CSP 要求が **/Accounts/Invoices** 内のファイルを要求したとき、**/Accounts/Invoices** の構成が存在しない場合、Web ゲートウェイは **/Accounts** に定義された構成を使用します。これが定義されていない場合は、既定のパスである **/** に対する構成が使用されます。

15.1 アプリケーション・パスの追加

アプリケーションへのパスを構成するには、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。

注釈 InterSystems IRIS インストールにより、新しい **/csp** 構成が作成されます。**/csp** をアプリケーションとして構成した場合、InterSystems IRIS の新しいビルドをインストールすると、その構成は上書きされます。アプリケーション構成を維持するには、**/csp** 以外のパスを入力します。

/csp/myapplication のように **/csp** にあるディレクトリであればどれでもかまいませんが、パスの指定にドット (ピリオド) は使用できません。これにより Web ゲートウェイであいまいさが発生するためです。**/csp/samples/menu.csp/csp/aaa/bbb/ccc.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/aaa/bbb/ccc.cls** への要求とも、**/csp/samples/menu.csp** への REST 要求 (この場合、PATH_INFO は **/csp/aaa/bbb/ccc.cls**) とも解釈できます。Web サーバ環境で動作している Web ゲートウェイには、このようなあいまいさを解決する手段がありません。

CSP は、大文字と小文字を区別します。CSP を構成している場合、整合性を確保しながらパス名を指定してください。

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. 入力を終えたら、[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。

15.1.1 アプリケーション・パスの構成パラメータ

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（特に CSP エンジン ）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、実行が拒否される（ CSP エンジン にアクセスする）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]	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。 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: jpeg gif ico png gz zip mp3 mp4 tiff 追加の種類または拡張子をスペースで区切ります。

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>

15.1.2 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>

15.2 アプリケーション・パス構成のコピー

新しいアプリケーション・パスを簡単に構成するには、既存のパスの構成エントリ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・パスの一時的な構成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・パス構成をコピーするには、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。

15.3 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。

15.4 アプリケーション・パス構成の削除

構成したアプリケーション・パスを削除するには、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。

16

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。

16.1 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。

16.1.1 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。

16.1.2 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。

16.1.3 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。

16.1.4 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。

16.1.5 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。

16.1.6 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。

16.2 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。

16.3 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。

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:svr=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ーに指示します(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。

16.4 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。

17

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](#)

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。[CSP エンジン](#) (%cspServer) に対し、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。

17.1 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]

17.2 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では、[接続セキュリティレベル] が [パスワード] に設定され、[ユーザ名] フィールドおよび [パスワード] フィールドは空欄のままです。

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。

17.3 単純なユーザ名/パスワード認証

ユーザ名/パスワードでは、[接続セキュリティレベル] は [] に設定され、[ユーザ名] および [パスワード] の値が指定されます。

これは、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。

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。

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。インストール・プロセスでは、この目的のために CSPSystem ユーザが作成されます。このユーザ (CSPSystem またはその他のユーザ) には、有効期限を設定する必要はありません。つまり、Expiration Date プロパティの値は 0 になります。

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

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。

17.3.1 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: ゲートウェイ管理フォームへのパスワード
- ・ 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:\cachesys\csp\bin\CSPx.dll 0 MyGatewayManagementPassword
CSPpwd C:\cachesys\csp\bin\CSPx.dll 2 MyIRISServerPassword
```

例 (UNIX®):

```
CSPpwd /opt/cspgateway/bin/CSPx.so 0 MyGatewayManagementPassword
CSPpwd /opt/cspgateway/bin/CSPx.so 2 MyIRISServerPassword
```

17.3.2 別のコンピュータで暗号化されたパスワード

Web ゲートウェイをホストしている Web サーバは、暗号化の基礎となる使用可能なユーザ・プロファイルがない保護された環境内で稼動するため、ユーザ・ストアではなくマシン・ストアを使用する必要があります。したがって、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。これによって、**構成ファイル**が共有ドライブに配置され、複数の関係するコンピュータ間で共有されるクラスタ環境の状況が作り出されます。実際にパスワード暗号化を行うコンピュータのみが、これを解読できます。暗号化されたパスワードを含む**構成ファイル**を別のコンピュータに移動することはできません。パスワードは、新しいマシンで再入力および再暗号化される必要があります。

この問題に対して考えられるアプローチを以下に示します。

- ・ 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
```

17.4 Kerberos ベース認証とデータ保護

Kerberos ベース認証とデータ保護では、[\[接続セキュリティレベル\]](#) 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。

17.4.1 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。

17.4.1.1 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。

17.4.2 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。

17.4.2.1 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 サーバに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。

17.4.3 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。

17.4.3.1 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 サーバに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。

17.5 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 構成名]** 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。

18

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。

19

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を呼び出すために 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。

20

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。

20.1 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。

20.2 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。

20.3 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。

20.4 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]	<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。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。</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す。</p> <p>GZIP Exclude File Types: <code>jpeg gif ico png gz zip mp3 mp4 tiff</code></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>

20.5 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)
```

21

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。

21.1 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

21.2 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。

21.3 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。

22

ミラー構成、フェイルオーバー、および負荷分散

22.1 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。セッションが継続している間は、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。

22.2 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。

22.3 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。

23

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。

23.1 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。

23.2 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。

23.3 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。

24

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、少なくとも 2 つのエントリが表示されます。管理ポータルを処理するプライベート Web サーバ用に 1 つ、およびアプリケーションをサポートする外部 Web サーバ用に最低 1 つです。

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("")
```

24.1 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。

24.1.1 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。

24.1.2 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
```


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**構成およびログ・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.cxw

`http://localhost:7039/csp/bin/Systems/Module.cxw`

B

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。

- ・ “Microsoft IIS の全バージョン” の手順に従います。
- ・ “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。
- ・ 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)

B.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。

B.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。

B.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。

B.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

B.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://<ip_address>/csp/bin/Systems/Module.cxw

http://<ip_address>/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、“[Web ゲートウェイとセキュリティ](#)”を参照してください。

B.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。

B.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。

B.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。
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。

B.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。
- 右側のパネルで、[モジュール マップの追加] を選択します。
- [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

- [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 : *.cxw

実行可能ファイル : C:\inetpub\CSPGateway\nph-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
```

B.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://<ip_address>/csp/bin/Systems/Module.cwx`

`http://<ip_address>/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、“[Web ゲートウェイとセキュリティ](#)” を参照してください。

B.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。

B.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。

B.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。

B.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を処理するように 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	バイナリ
*.csp	<code>C:¥Inetpub¥CSPGateway¥CSPcms.dll</code>
*.cls	<code>C:¥Inetpub¥CSPGateway¥CSPcms.dll</code>
*.zen	<code>C:¥Inetpub¥CSPGateway¥CSPcms.dll</code>
*.cxw	<code>C:¥Inetpub¥CSPGateway¥nph-CSPcgiSys.exe</code>

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

B.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://<ip_address>/csp/bin/Systems/Module.cwx`

`http://<ip_address>/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、“[Web ゲートウェイとセキュリティ](#)”を参照してください。

B.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。

B.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。
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-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。

6. `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

B.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 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

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\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

B.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://<ip_address>/csp/bin/Systems/Module.cxw

http://<ip_address>/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、“[Web ゲートウェイとセキュリティ](#)”を参照してください。

C

Apache の代替オプション (Windows)

このページには、InterSystems [Web ゲートウェイ](#)で使用するため、Microsoft Windows 上の Apache Web サーバでのインストールに関する情報が記載されています。それぞれのインストール環境に該当するセクションを参照してください。

このページでは、Microsoft Windows 用の追加の [Apache](#) 構成について説明します。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、それぞれのインストール環境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)

以下のモジュールがインストールされます。

- ・ **CSPcgi.exe** (実行時モジュール)
- ・ **nph-CSPcgi.exe** (CSPcgi.exe のコピー)
- ・ **CSPcgiSys.exe** (システム管理モジュール)
- ・ **nph-CSPcgiSys.exe** (CSPcgiSys.exe のコピー)

注釈 Apache バージョン 2.4.x の個別のバイナリは、次のとおりです。

- ・ **mod_csp24.dll** (DLL として機能する Apache 組み込み **module** が用意されている場合はそれを使用)
- ・ **CSPa24.dll** (実行時モジュールが用意されている場合はそれを使用)
- ・ **CSPa24Sys.dll** (提供される場合は、Web ゲートウェイのシステム管理モジュール)

これらのバイナリの既定の場所は以下のとおりです。

C:\Program Files\Apache Group\Apache\WebGateway\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。

NSD ベース以外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。

Sys が追加されたモジュールは、Web ゲートウェイ・システム管理スイートにアクセスするための特別なモジュールです。実行時モジュール (**Sys** が追加されていないモジュール) はシステム管理のフォームにアクセスできません。

C.2 代替オプション 1 : NSD を使用した Apache および CGI モジュール (nph-CSPcgi.exe)

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。

C:\Program Files\Apache Group\Apache\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])$" >
    AllowOverride None
    Options FollowSymLinks ExecCGI
    Require all granted
</LocationMatch>
ScriptAliasMatch /*\.[Cc][Ss][Pp]|[Cc][Ll][Ss])$ "c:/iris/csp/bin/nph-CSPcgi.exe"
Alias /csp/ c:/iris/csp/
<Directory "c:/iris/csp">
    AllowOverride None
    Options MultiViews FollowSymLinks ExecCGI
    Require all granted
    <FilesMatch "\.(log|ini|pid|exe)$">
        Require all denied
    </FilesMatch>
</Directory>
ScriptAlias /csp-bin/ "c:/iris/csp/bin/"
ScriptAliasMatch /csp-bin/Systems/Module.cxx "c:/iris/csp/bin/nph-CSPcgiSys.exe"
ScriptAliasMatch /csp-bin/RunTime/Module.cxx "c:/iris/csp/bin/nph-CSPcgi.exe"
<Directory "c:/iris/csp/bin/">
    AllowOverride None
    Options None
    Require all granted
    <FilesMatch "\.(exe)$">
        Allow from all
    </FilesMatch>
</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 (特に Windows システム) は、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)。この状況を解決するには、CSP ファイル拡張子に関連付ける代わりに、使用している [Web アプリケーション](#) の仮想ルート位置を CGI モジュールに関連付けます。例えば、使用している Web アプリケーションが /csp にあるとします。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])$" >
    AllowOverride None
    Options FollowSymLinks ExecCGI
    Require all granted
</LocationMatch>
ScriptAliasMatch /*\.[Cc][Ss][Pp]|[Cc][Ll][Ss])$
    "c:/iris/csp/bin/nph-CSPcgi.exe"
```

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 "c:/iris/csp/bin/nph-CSPcgi.exe"
```

これらの指示文は、フォームの URL に対して作用します。

http://localhost:<port_no>/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://localhost:<port_no>/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には、以下の手順に従います。

1. 新規の MIME タイプを Apache の `mime.types` ファイルの最後に追加し、それをクラスを表す [インターシステムズのファイル・タイプ](#) (`.csp`、`.cls`、および `.zen`) に関連付けます。`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
```

2. `httpd.conf` の CGI 構成ブロックの最後に、`Action` 指示文を次のように追加します。

```
Alias /csp/ c:/iris/csp/
<Directory "c:/iris/csp">
    AllowOverride None
    Options MultiViews FollowSymLinks ExecCGI
    Require all granted
<Files CSPnsd.exe>
    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>
<FilesMatch "\.(log|ini|pid|exe)$">
    Require all denied
</FilesMatch>
</Directory>
ScriptAlias /csp-bin/ "c:/iris/csp/bin/"
<Directory "c:/iris/csp/bin/">
    AllowOverride None
    Options None
    Require all granted
</Directory>
Action text/csp "/csp-bin/nph-CSPcgi.exe"
```

最後に、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 サーバで機能します。

3. `httpd.conf` の変更後に、Apache を再起動します。

C.2.1 追加ファイル・タイプのマッピング

Apache API モジュールは、[インターシステムズのファイル・タイプ](#)を常に認識します。その他のファイル拡張子については、“[追加ファイル・タイプを渡すための Apache の構成](#)”を参照してください。

C.2.2 Apache 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://localhost:<port_no>/csp/bin/Systems/Module.cwx
http://localhost:<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 Apache API モジュール (mod_csp24.dll)

注釈 この接続オプションは、オプション 1 で説明したスタンドアロンの API モジュールほど頻繁に使用されませんが、[NSD を使用する必要がある](#)場合に使用できます。DLL (Apache 2.4 の場合 – **mod_csp24.dll**) としてビルドされた CSP モジュールは、CGI ベースのソリューション (オプション 2) よりも高い性能を発揮します。

1. 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。

C:\Program Files\Apache Group\Apache\conf

拡張子が **.csp**、**.cls**、および **.zen** のファイルの CSP を呼び出すには、**httpd.conf** の最後に以下のセクションを追加します。Apache v2.4.x の場合、**mod_csp24.dll** を指定します。

```
LoadModule csp_module c:/iris/csp/bin/mod_csp24.dll
CSPFileTypes csp cls zen cwx
Alias /csp/ /iris/csp/
<Directory "c:/iris/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/ "c:/iris/csp/bin/"
ScriptAliasMatch /csp/bin/Systems/Module.cwx
    "c:/iris/csp/bin/nph-CSPcgiSys.exe"
ScriptAliasMatch /csp/bin/RunTime/Module.cwx
    "c:/iris/csp/bin/nph-CSPcgi.exe"
<Directory "c:/iris/csp/bin/">
    AllowOverride None
    Options None
    Require all granted
</Directory>
```

2. **httpd.conf** の変更後に、Apache を再起動します。

C.3.1 追加ファイル・タイプのマッピング

[CSP エンジン](#)により処理されるファイルの種類を追加構成するには、CGI モジュールにより処理される通常のファイル拡張子 (**.csp**、**.cls**、**.zen**) のリストに新しいファイル拡張子を追加します。例えば、これらを以下の行に追加します。

```
ScriptAliasMatch /*\.(?!(Cc)(Ss)(Pp)|(?!(Cc)(Ll)(Ss)|(?!(Zz)(En)(Nn)))$ "C:/iris/csp/bin/nph-CSPcgi.exe"
```

Web ゲートウェイを経由して他の静的ファイル进行处理する必要がある場合、またはこの Web サーバを経由して管理ポータルにアクセスする必要がある場合は、ファイルの .jpg、.gif、.png、.svg、.css、.js の各種類に対してマッピングを追加します。

以下の指示文は、すべてのファイルに対する要求を、指定された（特に [CSP エンジン](#) によって）パスの InterSystems IRIS にマップするために使用できます。

```
ScriptAliasMatch ^/csp/*/. * "C:/iris/csp/bin/nph-CSPcgi.exe"
```

したがって、/csp パス内のすべてのファイルに対する要求をマップするための基本的な構成ブロックは以下のようになります。

```
ScriptAliasMatch ^/csp/*/. * "C:/iris/csp/bin/nph-CSPcgi.exe"
<Directory "/iris/csp/bin/">
    AllowOverride None
    Options None
    Require all granted
</Directory>
```

C.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、ブラウザで以下の場所のいずれかを指定します。

```
http://localhost:<port_no>/csp/bin/Systems/Module.cwx
http://localhost:<port_no>/csp-bin/nph-CSPcgiSys.exe
```

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 Microsoft Windows 向けにロック・ダウンした Apache 環境

Apache がロック・ダウンされているために、Apache ファイル・システムの外部のファイルにアクセスするようにサーバを簡単に構成できない場合があります。

このようなロック・ダウンの構成では、上記のセクションで説明した Web ゲートウェイ構成で CSP リソースにアクセスしようとすると、HTTP 403 Forbidden のエラー・コードが返されます。これらの安全な構成を使用するには、以下の手順に従います。install-dir¥csp¥

上記にあるファイル・システムを、以下の Apache ルートの場所にコピーします。

C:¥Program Files¥Apache Group¥Apache¥

Apache 構成で指定したパスを適切に変更します。

Apache で用意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。

1. CGI モジュールを次のように C:¥Program Files¥Apache Group¥Apache¥cgi-bin¥ にコピーします。

```
copy c:\iris\csp\bin\*cgi*.exe C:\Program Files\Apache Group\Apache\cgi-bin\
```

- API モジュールを C:\Program Files\Apache Group\Apache\modules にコピーします。

```
copy c:\iris\csp\bin\*.dll C:\Program Files\Apache Group\Apache\modules\
```

- 静的ファイル (およびそのサブディレクトリ) を C:\Program Files\Apache Group\Apache\htdocs\csp\samples にコピーします。

```
copy c:\iris\csp\samples\*.*
C:\Program Files\Apache Group\Apache\htdocs\csp\samples\
copy c:\iris\csp\broker\*.*
C:\Program Files\Apache Group\Apache\htdocs\csp\broker\
copy c:\iris\csp\sys\*.*
C:\Program Files\Apache Group\Apache\htdocs\csp\sys\
```

- NSD コンポーネント (必要な場合) を C:\Program Files\Apache Group\Apache\nsd. にインストールします。

Apache で事前構成されたディレクトリを使用すると、httpd.conf での Web ゲートウェイの構成が簡単になります。修正した構成ブロックを以下に示します。

C.4.1 推奨オプションの構成 : Apache API モジュール (CSPa24.dll)

```
LoadModule csp_module_sa
C:/Program Files/Apache Group/Apache/modules/CSPa24.dll
CSPFileTypes csp cls zen cxw
```

C.4.2 代替オプション 2 の構成 : NSD を使用した CGI モジュール (nph-CSPcgi.exe)

```
<LocationMatch "/*\.([Cc][Ss][Pp]|[Cc][Ll][Ss]|[Zz][En][Nn])$" >
AllowOverride None
Options FollowSymLinks ExecCGI
Require all granted
</LocationMatch>
ScriptAliasMatch /csp/bin/Systems/Module.cwx "C:/Program Files/Apache
Group/Apache/cgi-bin/nph-CSPcgiSys.exe"
ScriptAliasMatch /csp/bin/RunTime/Module.cwx "C:/Program Files/Apache Group/Apache/cgi-bin/nph-CSPcgi.exe"
ScriptAliasMatch /*\.([Cc][Ss][Pp]|[Cc][Ll][Ss])$ "C:/Program Files/Apache
Group/Apache/cgi-bin/nph-CSPcgi.exe"
```

C.4.3 代替オプション 3 の構成 : NSD を使用した Apache API モジュール (mod_csp24.dll)

```
LoadModule csp_module
C:/Program Files/Apache Group/Apache/modules/mod_csp24.dll
CSPFileTypes csp cls zen cxw
ScriptAliasMatch /csp/bin/Systems/Module.cwx "C:/Program Files/Apache
Group/Apache/cgi-bin/nph-CSPcgiSys.exe"
ScriptAliasMatch /csp/bin/RunTime/Module.cwx "C:/Program Files/Apache Group/Apache/cgi-bin/nph-CSPcgi.exe"
```

C.4.4 代替オプション 4 の構成 : ISAPI モジュール (CSPms.dll)

従来のみ

```
AddHandler isapi-isa dll
AddHandler isapi-isa csp
AddHandler isapi-isa cls
AddHandler isapi-isa zen
AddHandler isapi-isa cxw
Alias /csp/bin/Systems/Module.cwx /csp/bin/CSPmsSys.dll
```

D

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。

D.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。

D.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。

D.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。

D.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。

D.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>
```


D.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 [構成およびログ・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
```

D.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。

D.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。

E

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。

E.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。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。

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
```

E.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 (“[推奨オプション：NSD を使用しない Apache API モジュール \(CSPa24.so\)](#)”) 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。

E.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。

E.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。

E.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。

E.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。

E.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
```

E.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://localhost:<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。

E.2.4 追加ファイル・タイプのマッピング

Apache API モジュールは、[インターシステムズのファイル・タイプ](#)を常に認識します。その他のファイル拡張子については、“[追加ファイル・タイプを渡すための Apache の構成](#)”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 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、フォームの URL に対して作用します。

```
http://localhost:<port_no>/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://localhost:<port_no>/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。

E.3.1 追加ファイル・タイプのマッピング

Apache API モジュールは、[インターシステムズのファイル・タイプ](#)を常に認識します。その他のファイル拡張子については、["追加ファイル・タイプを渡すための Apache の構成"](#)を参照してください。

E.3.2 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://localhost:<port_no>/csp/bin/Systems/Module.cxxw
http://localhost:<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。

E.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。

E.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/>

E.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

```

E.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。

E.4.4 追加ファイル・タイプのマッピング

Apache API モジュールは、[インターシステムズのファイル・タイプ](#) を常に認識します。その他のファイル拡張子については、[“追加ファイル・タイプを渡すための Apache の構成”](#) を参照してください。

E.4.5 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://localhost:<port_no>/csp/bin/Systems/Module.cwx  
http://localhost:<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。

F

ロック・ダウン Apache (UNIX®/Linux/macOS)

Apache がロック・ダウンされているために、Apache ファイル・システムの外部のファイルにアクセスするようにサーバを簡単に構成できない場合があります。例えば、Security Enhanced Linux (SELinux) の場合です。

このようなロック・ダウンの構成では、別途説明した [Web ゲートウェイ](#) 構成で CSP リソースにアクセスしようとすると、HTTP 403 Forbidden のエラー・コードが返されます。この問題に対処するための戦略は 2 つあります。

- ・ 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。
- ・ 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。

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。これは通常は簡単な作業であり、以下の手順が含まれます。

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。

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。

F.1 推奨オプション : Apache API モジュール (CSPa24.so)

```
LoadModule cspsys_module /usr/apache/modules/CSPa24.so
CSPSYSModulePath /usr/apache/modules/
CSPFileTypes csp cls zen cxw
```

F.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"
```

F.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"
```

F.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 cwx
ScriptAliasMatch /csp/bin/Systems/Module.cwx "/usr/apache/cgi-bin/nph-CSPcgiSys"
ScriptAliasMatch /csp/bin/RunTime/Module.cwx "/usr/apache/cgi-bin/nph-CSPcgi"
```


G

Apache に関する考慮事項 (UNIX®/Linux/macOS)

このページには、UNIX®、Linux、および macOS の推奨オプション (推奨オプション : NSAPI モジュール (CSPn3.so)) と特殊オプション 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)) に関する情報を記載しています。

G.1 Apache プロセス管理および処理能力の計画

Apache は、UNIX® オペレーティング・システム向けに 3 つのプロセス管理モジュールを提供しています。このアーキテクチャでは、InterSystems Web ゲートウェイ・モジュールは Apache ワーカー・プロセスに直接バインドされます。したがって、プロセス・プールを管理するための Apache の構成方法が、Web ゲートウェイに直接の影響を与えます。

Apache は各プロセス管理モデルをマルチプロセッシング・モジュール (MPM) として実装します。

Prefork MPM は、従来のマルチプロセス (UNIX®) サーバ・アーキテクチャです。これはスレッドを使用しないため、サードパーティの API モジュール (DSO) がスレッドセーフである必要はありません。参照 :

<http://httpd.apache.org/docs/current/mod/prefork.html>

Worker MPM は、より新しいマルチスレッド/マルチプロセスのハイブリッド型サーバ・アーキテクチャです。これはスレッドを使用するため、使用されるすべてのサードパーティ製 API モジュール (DSO) はスレッドセーフであることが必要です。参照 : <http://httpd.apache.org/docs/current/mod/worker.html>

Event MPM は、一部の処理作業をリスナ・スレッドに渡してワーカー・スレッドを解放し、新しい要求を処理できるようにすることで、より多くの要求を同時に処理できるように設計されています。参照 : <http://httpd.apache.org/docs/current/mod/event.html>

どちらのサーバ・モデルが既存のインストールに使用されているかを判別するには、Apache 実行可能プログラムを次のように修飾して直接呼び出します。

```
httpd -V
```

さらに、以下の 2 つの関連する詳細リストが用意されています。

- ・ `httpd -l` : サーバに組み込まれたすべてのモジュールをリストします。
- ・ `httpd -L` : すべてのモジュールと関連する構成指示文をリストします。

Web ゲートウェイ DSO はスレッドセーフで、どのサーバ・モデルにも展開できます。Apache の調整に関する役立つガイドが、<http://httpd.apache.org/docs/current/misc/perf-tuning.html> にあります。

G.1.1 セキュリティ

3 つのサーバ・アーキテクチャすべてで、親プロセスは、TCP ポート 80 にバインドするために、通常、スーパーユーザ特権が割り当てられたアカウント (UNIX® の root) から開始します。Apache が起動する子プロセスは、特権の少ないユーザとして実行されます。(Apache 構成内の) User および Group 指示文は、Apache 子プロセスの特権を設定するために使用されます。子プロセスは、それらが処理するすべてのコンテンツを読み取る能力 (および Web ゲートウェイの構成およびイベント・ログ・ファイルへの読み取り/書き込みアクセス権) が必要ですが、これ以外に付与する特権は最小限にする必要があります。詳細は、Apache ドキュメントを参照してください。

G.1.2 Apache MPM と Web ゲートウェイ DSO

Web ゲートウェイ・[ダイナミック・リンク・モジュール \(DSO\)](#) はスレッドセーフで、どのサーバ・モデルにも展開できます。

[StartServers](#) 指示文は、すべての[マルチプロセッシング・モジュール \(MPM\)](#) について、開始する子 (ワーカ) プロセスの数を指定します。この指示文はまた、存在できる Web ゲートウェイ DSO のインスタンスの数を示します (例 : Apache の子プロセスごとに 1 つ)。

すべての MPM が、複数の子 (ワーカ) プロセスにわたる負荷分散に関係します。

各ゲートウェイ・インスタンスはすべての Apache 子プロセスそれぞれによって個々にロードされますが、実行中の構成、接続テーブル、およびフォーム・キャッシュは共有メモリ・セクタに保持されます。Web ゲートウェイ・システム・ステータス・フォームのコンテンツは、更新しても同じになります (もちろん、アクティビティの更新結果による変更は除きます)。表示される接続テーブル (および接続数) は Apache インスタンス全体に共通で、この理由から、各 InterSystems IRIS 接続が関連付けられる Web サーバ・プロセス ID を示す追加列が含まれます。

G.1.2.1 サーバ接続最大数

Web ゲートウェイの負荷は複数の Web サーバ・プロセスに分散されますが、サーバ接続最大数パラメータは、Web ゲートウェイが特定の InterSystems IRIS サーバに対して確立できる接続数についての全体的な制限を 1 つだけ設定します。つまり、ホスト Web サーバによって開始されるワーカ・プロセスの数は、Web ゲートウェイが作成できる最大接続数に影響を与えません。この最大数は、接続を行うプロセスのタイプや使用中の MPM の影響も受けません。(このモデルは、サーバ接続最大数パラメータがプロセスごとに影響を受け、複数の要因の影響を受ける一般的なスロットルとして機能していた、以前のバージョンからの変更を表しています。)

Apache の負荷のほとんどが[インターシステムズのファイル・タイプ](#)から構成されるインストールでは、Web ゲートウェイのサーバ接続最大数指示文に値を指定しないで、対応する Apache 構成パラメータを使用して実行できる並行作業量 (したがって、InterSystems IRIS への接続数) を制御することをお勧めします。ただし、インターシステムズのファイル・タイプが Apache インストール全体の負荷の一部でしかないインストールでは、Web ゲートウェイのサーバ接続最大数指示文に独立した値を設定するのは有効です。

G.2 ステート認識セッション (保持モード 1)

複数のワーカ・プロセスに負荷を分散する Web サーバでのステート認識セッションのサポートは、個々のワーカ・プロセス間の要求の転送を管理する IPC (プロセス間通信) プロトコルに依存します。この Web サーバ・アーキテクチャで動作する Web ゲートウェイは、どのワーカ・プロセスが特定の要求を処理するかを制御することはできません。

Web ゲートウェイはその IPC プロトコルに UNIX® ドメイン・ソケットを使用します。ステート認識セッションをサポートする方法について、以下に説明します。

例として、P1、P2、および P3 の 3 つのワーカ・プロセスに負荷を分散する Web サーバ・インストールについて考えてみましょう。各ワーカ・プロセスは、使用されている Web サーバの MPM と構成に従って、任意の数のスレッド (T1、T2 … Tn) を開始できます。

アプリケーションがそのセッションをステート認識 (保持モード 1) とマークする要求を行い、Web ゲートウェイがプロセス P2 でその指示を認識すると仮定します。今ではプライベートの InterSystems IRIS プロセスへの接続 (およびセキュリティ・コンテキスト) は Web サーバのワーカ・プロセス P2 によってホストされます。そのユーザ/セッションに対する以降の要求は、すべてワーカ・プロセス P2 で処理する必要があります。ただし、Web ゲートウェイは Web サーバが以降の要求をどのワーカ・プロセスに転送するかを制御できないため、Web ゲートウェイは P2 と (可能性として) そのセット内の他の任意のワーカ・プロセスとの間で IPC チャンネルを確立する必要があります。

Web ゲートウェイは P2 でステート認識として接続をマークすると、別個の分離されたスレッドで待ち受けサービスを開始します。[ログ・レベル v2](#) の場合は、以下のようなメッセージがイベント・ログに書き込まれます。

```
IPC Server
Process ID: 28457 Listening on Domain Socket: /tmp/csp28457.str
```

次に、例えば、同じセッションに対するさらなる要求がワーカ・プロセス P3 によって処理されると仮定します。Web ゲートウェイはその要求を、以前に確立された IPC チャンネルを介してプロセス P2 に転送し、応答を待機します。[ログ・レベル v2](#) の場合は、以下のようなメッセージが記録されます。

```
Route request over IPC to another web server process
PrivateSession=2; pid_self=28456; ipc_to_pid=28457;
```

もちろん、そのセッションに対する要求が Web サーバによって P2 に直接転送される場合は、P2 がセッションのプライベート接続をホストしているため、Web ゲートウェイ環境でさらなる転送は必要ありません。

以前作成された IPC チャンネルに Web ゲートウェイが接続して要求を転送できない場合は、エラーが発生したコンテキストに応じて以下のメッセージのいずれかが記録されます。

```
IPC CLIENT: Error
Cannot connect
```

または

```
IPC CLIENT: Error
Cannot send request
```

この領域で問題が発生する最も一般的な理由は、Apache がワーカ・プロセスを閉じたこと (またはリサイクルしたこと) です (P2 の例の場合)。もちろん、(アクセス違反/SIGSEGV エラーなどで) プロセスはクラッシュし、この場合は、おそらくエラー・メッセージが Apache エラー・ログに報告されます。

Apache はまた、既定では、ワーカ・プロセスを定期的に関断します。

ステート認識セッションを使用する場合は、以下のようにインストールを構成して、ワーカ・プロセスをリサイクルしないように Apache を構成してください。

- ・ `MaxConnectionsPerChild` の値をゼロに設定します。
- ・ `MaxSpareThreads` の値を `MaxRequestWorkers` と同じ値に設定します。

(おそらくモジュールの故障が原因で) Apache が定期的にプロセスをリサイクルするのを阻止できない場合に、ステート認識セッションを使用する必要があるときは、NSD ベースのゲートウェイ構成を使用できます。NSD ベースのアーキテクチャは、Web ゲートウェイのプロセス管理を Web サーバから効果的に切り離すため、前述の問題を回避します。Web ゲートウェイの Network Service Daemon (NSD) を使用する際のオプションについては、“[Microsoft Windows での NSD の使用方法](#)” および “[UNIX®、Linux、および macOS での NSD の使用方法](#)” を参照してください。

H

IIS の技術メモ

IIS に関心のあるユーザ向けに、このページではアプリケーション・プール、Web ガーデン、およびビットネスについて説明します。

H.1 IIS アプリケーション・プールおよび Web ガーデン

H.1.1 アプリケーション・プール

アプリケーション・プールとは、1 つ以上のアプリケーションを、1 つ以上のワーカ・プロセスのセットに関連付ける構成です。アプリケーション・プールにある各アプリケーションは、ワーカ・プロセスの境界によって他のアプリケーションから分離されているので、あるアプリケーション・プールで実行中のアプリケーションに起因する問題によって、別のアプリケーション・プールにあるアプリケーションが影響を受けることはありません。

新しいアプリケーション・プールを作成して、Web サイトやアプリケーションをこれらのプールに割り当てることで、サーバの効率性と信頼性を高めることができます。あるアプリケーションを処理しているワーカ・プロセスに不具合が生じた場合でも、プールを介して動作する他のアプリケーションは常に利用できます。

アプリケーションは IIS 内でそのパスによって定義されます。例えば、`/csp` のようになります。

H.1.2 Web ガーデン

信頼性をさらに高めるために、アプリケーション・プールを複数のワーカ・プロセスでサポートするように構成できます。複数のワーカ・プロセスを使用するアプリケーション・プールを Web ガーデンといいます。Web ガーデン内のワーカ・プロセスは、特定のアプリケーション・プールに到着する要求を共有します。あるワーカ・プロセスに障害が発生しても、別のワーカ・プロセスが引き続き他の要求を処理できます。

Web ガーデンと Web ファームは異なることに注意してください。Web ガーデンは、1 つのアプリケーション・プールに複数のワーカ・プロセスを指定することで、1 台のサーバ上に構成されます。Web ファームは、複数のサーバを使用して 1 つの Web サイトをサポートするものです。

1 つのアプリケーション・プールに対して 1 つの Web ガーデンを作成すると、以下の状況でパフォーマンスを強化できます。

- ・ 要求処理の堅牢化：アプリケーション・プール内の 1 つのワーカ・プロセスが停止した場合（例えば、スクリプト・エンジンの応答が停止した場合）、他のワーカ・プロセスがそのアプリケーション・プールに対する要求を受け入れ、処理できます。

- ・ リソース競合の軽減：Web ガーデンが安定した状態になると、ラウンドロビン方式によって、新しい各 TCP/IP 接続が Web ガーデン内のワーカ・プロセスに割り当てられます。これにより、負荷が緩和され、ワーカ・プロセスにバインドしているリソースの競合が軽減されます。

H.1.3 アプリケーション・プール、Web ガーデン、および CSP

アプリケーション・プールと Web ガーデンの構成は、NSD ベースのゲートウェイ構成の動作に影響を与えません。これは NSD と通信する ISAPI モジュールは、永続情報やその他のリソース (InterSystems IRIS への接続など) をプールしないためです。永続リソースはすべて NSD モジュールで保持されます。NSD と通信する ISAPI モジュールは、IIS で ISAPI モジュールを管理する方法が変更されてもその影響を受けません。

NSD 以外をベースとするゲートウェイ構成 (CSPms.dll および CSPmsSys.dll) の方が、IIS で ISAPI 拡張を管理する方法の変更に大きく影響を受けます。これは、永続リソース (InterSystems IRIS への接続など) のプーリングが拡張自体で行われるためです。

使用するワーカ・プロセスを 1 つのみとするように構成したアプリケーション・プールは、単一の [Web アプリケーション・パス \(/csp など\)](#) のコンテキスト内における Web ゲートウェイの動作に対して目に見える影響を及ぼしません。ただし、複数のワーカ・プロセスを使用する構成の場合 (Web ガーデン)、Web ゲートウェイの負荷は、プールに属するすべてのワーカ・プロセス間で均等に分散されます。各ワーカ・プロセスは、Web ゲートウェイ・モジュールのそれぞれのインスタンスを管理します。この処理管理アーキテクチャにより、Web ゲートウェイの動作に関する問題は発生しませんが、以下の制限事項を念頭に置く必要があります。

- ・ Web ゲートウェイの構成に対する変更を有効にするには、IIS を再起動する必要があります。この場合、インターネット・サービス・マネージャのコントロール・パネルではなく、メインの Windows サービスのコントロール・パネルから World Wide Web Publishing サービス全体を再起動する必要があります。
- ・ Web ゲートウェイのシステム管理のフォーム (システム・ステータス) を使用して、Web アプリケーションで使用する接続を正確に監視することはできません。システム・ステータスには、現在のワーカ・プロセス (つまり、Web ゲートウェイの要求を処理するワーカ・プロセス) にアタッチされている、Web ゲートウェイのインスタンスのステータスが、指定されたどの時点でも反映されています。
- ・ 各 Web アプリケーション (アプリケーションの Web パスで定義) は、InterSystems IRIS に対する永続接続のそれぞれのプールを維持します。また、アプリケーション・プール内の各ワーカ・プロセスは、InterSystems IRIS への永続接続のプールを各自維持します。Web ゲートウェイが使用する、InterSystems IRIS への最大接続数および最小接続数を構成する際に、この動作を念頭に置く必要があります。これらの設定は、プール内のすべてのゲートウェイ・インスタンスに適用されます。
- ・ ステート認識セッション (**保持モード 1**) は、Web ガーデンの構成では使用できません。特定の要求の処理に使用する、Web ゲートウェイのインスタンスを制御できないためです。したがって、これらの構成では、ステート認識要求を専用の InterSystems IRIS プロセスにルーティングできません。

Web ゲートウェイは IIS とは別に管理されているため、NSD ベースのオプションはこれらの制限事項の影響を受けないことに注意してください。

最後に、NSD 以外のバージョンの Web ゲートウェイに対してワーカ・プロセス構成パラメータが及ぼす影響を考慮する必要があります。特に、アイドル・タイムアウトおよびプロセスのリサイクルの影響に留意する必要があります。

H.1.4 ワーカ・プロセスのアイドル・タイムアウト

多くの場合、未使用のワーカ・プロセスを終了して、システム・リソースの浪費を避ける必要があります。指定した時間が経過した後、ワーカ・プロセスを適切に閉じるように構成できます。この機能を使用すると、処理の負荷が大きい場合、特定のアプリケーションが頻繁にアイドル状態になる場合、または新しい処理領域がない場合に、リソースの管理を改善できます。

ワーカ・プロセスが終了すると、そのプロセスが管理するゲートウェイのインスタンスも閉じ、この Web ゲートウェイ・インスタンスが保持する InterSystems IRIS への接続のプールも終了します。もちろん、他のステートレスな接続は Web アプリ

ケーションのユーザに対して透過的な方法で常に置き換えられますが、ステート認識セッション (保持モード 1) は、ホスト接続が閉じると終了します。

H.1.5 ワーカ・プロセスのリサイクル

ワーカ・プロセスを定期的に再起動して障害のある Web アプリケーションをリサイクルできるように IIS を構成できます。この機能は、アプリケーション・プールを正常な状態に保ち、リークしたシステム・リソースを復元するのに役立ちます。

経過時間、処理された要求の数、予約した時間、およびメモリの使用状況に基づいて再起動するようにワーカ・プロセスを構成できます。

ワーカ・プロセスを閉じることによる Web ゲートウェイへの影響については、前のセクション (アイドル・タイムアウト) に説明があります。このセクションでも同じ考慮事項が適用されます。Web アプリケーションは、慎重に管理されるチャンネルを介してのみ Web ゲートウェイと相互運用できるので、Web アプリケーションをサポートするワーカ・プロセスのリサイクルはお勧めしません。

H.2 ビットネス – Windows の 64 ビット・サーバ上での 32 ビット・アプリケーション

注釈 このセクションは、ホスト Web サーバのアドレス空間にロードされるモジュール、つまり ISAPI 拡張およびネイティブ・モジュール (CSPMs[Sys].dll および CSPcms.dll) に適用されます。CGI モジュールは、IIS に関しては分離されたプロセスとして動作するため、影響を受けません。

[32 ビット アプリケーションの有効化] 設定はアプリケーション・プール・レベルに適用されるため、特定のアプリケーション・プールにビットネスを設定できます。単一のサーバ・インストールで、1 つのアプリケーション・プールでネイティブ 64 ビット・アプリケーションを、もう 1 つのアプリケーション・プールで 32 ビット・アプリケーションを実行するように構成することが可能です。

アプリケーション・プールのビットネス設定にアクセスするには、IIS コントロール・パネルで以下の手順に従います。

1. 左側のパネルで **[アプリケーション プール]** を選択します。
2. 適切なアプリケーション・プールを選択します。
3. 右側のパネルで **[詳細設定]** を選択します。
4. **[詳細設定]** ダイアログが表示されます。**[全般]** セクションに **[32 ビット アプリケーションの有効化]** 設定が表示されます。これは True または False に設定できます。

ちなみに、この構成設定は、Windows コマンド行で `appcmd` コマンドを使用して操作できます。以下はその例です。

```
appcmd set apppool /apppool.name:DefaultAppPool/enable32bitapponwin64:true
```

この例では、アプリケーション・プール `DefaultAppPool` を 32 ビット・モードで実行するように設定しています。

また、`appcmd` コマンドを使用して、ビットネスに基づいてアプリケーション・プールをリストすることもできます。例えば、64 ビット・モードで動作するすべてのアプリケーション・プールをリストするには、以下のコマンドを使用します。

```
appcmd list apppools /enable32bitapponwin64:false
```

最後に、アプリケーション・プールは異なるビットネス・モードで実行できるため、アプリケーション・プールがロードするネイティブ・モジュール (および ISAPI 拡張) 自体がそのプールの正しいビットネスでなければなりません。例えば、ホストのアプリケーション・プールが 64 ビットの場合は、64 ビットのゲートウェイ・モジュール (CSPMs[Sys].dll など) を使用す

する必要があります。ホストのアプリケーション・プールが 32 ビットの場合は、32 ビットのゲートウェイ・モジュールを使用する必要があります。

個々のモジュールのビットネス・チェックは、モジュールの **web.config** ファイルの **preCondition** を通して行われます。Web ゲートウェイの場合、このファイルは通常、次のようになります。

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <system.webServer>
    <handlers>
      <add name="WebGateway_All" path="*" verb="*" modules="CSPms" resourceType="Unspecified" %
        preCondition="bitness64" />
    </handlers>
    <security>
      <requestFiltering>
        <hiddenSegments>
          <remove segment="bin" />
        </hiddenSegments>
      </requestFiltering>
    </security>
  </system.webServer>
</configuration>
```

precondition 節のビットネス設定に注意してください。ここでは、ビットネスは **bitness64** に設定されています。これは、IIS が、64 ビット・アプリケーション・プールで動作する 64 ビット・ゲートウェイ・モジュールをチェックすることを意味します。

32 ビット・アプリケーション・プールが使用されている場合は、32 ビット・ゲートウェイ・モジュールを使用し、preCondition を **bitness32** に設定する必要があります。

インストールされたモジュール、precondition 節、またはホストのアプリケーション・プールが予期するものの間に不整合がある場合、IIS は以下のようなエラー状態を返します。

```
Error:
The module(s) assigned to this handler mapping has the following preconditions that are not
present in the handler mapping:
bitness64
Runtime errors may occur if a handler mapping does not have a set of preconditions that are
equally as restrictive as, or more restrictive, than the module(s) assigned to the mapping.
Please ensure that this handler mapping has the correct preconditions, and that the
preconditions are not in conflict.
```


I

リモート Web サーバでの Web アプリケーションの使用

I.1 Web サーバと Web ゲートウェイの構成

このセクションでは、Web サーバおよび [Web ゲートウェイ](#)を設定して、リモート InterSystems IRIS サーバにインストールされた [Web アプリケーション](#)にアクセスする方法について説明します。この手順では、Web サーバを Machine W、InterSystems IRIS® を実行しているコンピュータを Machine I と呼びます。設定は以下の手順から構成されます。

1. [Web サーバ・マシンへの Web ゲートウェイのインストール](#)
2. [Web ゲートウェイの構成](#)
3. [Web サーバから静的ファイルを提供する場合](#)
4. [Web サーバ・パスの構成](#)

I.1.1 Web サーバ・マシンへの Web ゲートウェイのインストール

IIS または Apache が実行中の Web サーバ・マシン (Machine W) に Web ゲートウェイをインストールします。詳細は、“[インストール・ガイド](#)”の“[Web ゲートウェイのみのインストール](#)”を参照してください。インストール中は、以下の指示に従ってください。

1. [セッアップ・タイプ] ダイアログ・ボックスで、[Web サーバ] を選択し、[次へ] を選択します。
2. インストール名、インストール・タイプおよびインストール先ディレクトリを確認し、正しければ [インストール] を選択します。

このインストール・プロセスによって、Web サーバ Machine W に **CSP** ディレクトリ構造が生成され、/CSP ファイルおよび /CSP/Bin ファイルへの仮想ディレクトリ参照が生成されます。

I.1.2 Web ゲートウェイの構成

次に、Web サーバ Machine W の Web ゲートウェイ構成を調整します。この構成情報は[構成ファイル](#)に保存されていますが、常に Web ゲートウェイ管理ページを使用して、構成を更新します。

1. ブラウザで以下を指定し、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。

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

(この URL をブックマークに登録しておくとう便利です)。このリンクは、外部 Web サーバ用です。InterSystems IRIS が提供する[プライベート Web サーバ](#)用ではありません。

上記のリンクは、Web サーバが実行されている同じシステム上でポート 80 を使用してアクセスする場合のみ正しく動作します。リモート・システムから管理ページにアクセスしようとする、既定ではアクセスを拒否されます。Web ゲートウェイ管理ページの **[デフォルトパラメータ]** ページを使用して、リモート・アクセスを許可するように Web サーバを構成できます。**[システム管理マシン]** フィールドをリモート・システムの IP アドレスに設定します。このフィールドは、Web ゲートウェイ管理ページにアクセス可能なマシンの IP アドレスをコンマまたはプラス記号で区切ったリストを受け入れます。

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。

2. 左側のメニューで **[サーバ接続]** を選択します。IRIS が Web サーバにインストールされている場合、ローカル・マシン (Machine W) の InterSystems IRIS インスタンスに接続するように **LOCAL** Web サーバが構成されます。
3. リモート・マシン上で動作している InterSystems IRIS インスタンスを表す新しいサーバ (Machine I) を作成します。
 - a. **[サーバ追加]** を選択します。
 - b. サーバの名前を入力します (例えば、Machine I)。
 - c. Machine I 上のリモート InterSystems IRIS サーバの **[IP アドレス]** および **[スーパーサーバの TCP ポート]** を入力します。
 - d. Web ゲートウェイ接続を確立するために Machine I が必要とする認証のレベルに合うように、**[接続セキュリティ]** 設定を変更します。詳細は、[“Web ゲートウェイとセキュリティ”](#) を参照してください。
 - e. **[設定を保存]** を選択します。

詳細は、[“サーバ構成の追加”](#) を参照してください。

4. 左側のメニューで **[アプリケーションアクセス]** を選択して、リモート InterSystems IRIS サーバ (Machine I) 上の Web アプリケーションへのパスを Machine I 用に以前作成したサーバ構成と関連付けます。既定のパスは / および **/csp** に事前定義されています。
5. リモート・マシン (Machine I) 上で動作している Web アプリケーションを表す新しいアプリケーション・パスを作成します。既存の構成 (**/csp** など) をコピーするか、または **[アプリケーション追加]** を選択し、新しいパス構成を手動で作成できます。アプリケーション用に作成するパスは、Machine I 上の InterSystems IRIS インスタンスにあるアプリケーション用に定義したものと一致する必要があります。

例えば、管理ポータルへの既定のパスは **/csp/sys** です。新しいアプリケーションを作成している場合、独自のパス名を選択します。例: **/myapp** や **/csp/myapp**。新しいパスを作成したら、Machine I 用に以前設定した InterSystems IRIS サーバ構成の値になるように、パスの **[サーバ 0]** パラメータを変更します。

6. 最後に、新しいパス構成を保存します。

詳細は、[“アプリケーション・パスの追加”](#) を参照してください。

I.1.3 Web サーバから静的ファイルを提供する場合

Web サーバから静的ファイルを提供することを計画している場合、Machine W 上にアプリケーション・パスを表すディレクトリを作成します。これらのディレクトリは、イメージ・ファイルなどの静的コンテンツを保持するためにのみ存在します。すべての CSP ファイルをここに配置する必要はありません。これらは Machine I 上にあります。

Machine W のディレクトリ **install-dir¥CSP** の下に、**¥Samples** ディレクトリおよび **¥User** ディレクトリを作成します。また、CSP ページで参照される静的コンポーネントを含むことができる、その他のパスを表すディレクトリを作成します。前のセクションの例では **¥myapp** のディレクトリを作成する必要があります。

I.1.4 Web サーバ・パスの構成

前の手順のアプリケーション・パスは、対応する場所にある CSP ページの要求に対応します。以下はその例です。

```
http://domain.com/myapp/login.csp
http://domain.com/csp/myapp/login.csp
http://domain.com/csp/sys/login.csp
```

継承は階層的に適用されます。以下の要求を考えてみます。

```
http://domain.com/csp/newapp/login.csp
```

/csp/newapp のアプリケーション・パス構成がある場合、これが使用されます。存在しない場合、代わりに /csp 用に定義される構成が使用されます。

Web ゲートウェイインストール手順では、ホスト Web サーバが構成され、/csp 仮想パスを認識します。通常、これらの同じ設定は、/csp の下にあるディレクトリ (/csp/myapp など) にも適用されます。

新しいパス (例えば、最初の例では /myapp) を作成する場合、Web サーバを構成して、この新しい仮想パスを認識する必要があります。これらの手順は、使用する Web サーバに応じて異なります。使用中の Web サーバに適用するセクションの手順に従います。

- ・ [IIS への仮想ディレクトリの追加](#)
- ・ [Apache 構成へのエイリアスの追加](#)

I.1.4.1 IIS への仮想ディレクトリの追加

Web ゲートウェイのインストール手順では、Web アプリケーション用に仮想ディレクトリ install-dir¥csp が構成されます。使用中のすべてのアプリケーションがこの仮想ディレクトリの下 (例えば install-dir¥csp¥myapp) にあり、仮想ホストを使用していない場合、仮想ディレクトリを追加する必要はありません。このセクションの手順は、具体的には IIS (インターネット インフォメーション サービス) バージョン 7 以降の Web サーバに適用されます。

前の例の /myapp と同様のアプリケーション・パス、および Web ゲートウェイのインストール中に自動的に作成される /csp 仮想ディレクトリと同様のプロパティを設定します。

1. [インターネット インフォメーション サービス] 管理ダイアログ・ボックスに移動します。これは、通常、Windows の [コントロール パネル] の [管理ツール] メニューからアクセスできます。
2. フォルダを展開し、[既定の Web サイト] を右クリックします。
3. [新規作成] を指定して [仮想ディレクトリ] を選択し、以下の値を指定して新しいディレクトリ・レコードを作成します。

Alias:	myapp
Directory:	C:\iris\csp\myapp

4. [保存] と [適用] を選択してすべての変更を適用するか、またはウィザードを使用している場合、[完了] を選択します。

IIS を再起動して変更内容を適用します。

I.1.4.2 Apache 構成へのエイリアスの追加

Apache Web サーバを使用してリモート InterSystems IRIS サーバを制御していて、アプリケーション・パスが既定の /csp から変更されている場合、ローカル CSP ディレクトリを指す Apache 構成ファイルに対応するエイリアスを手動で追加する必要があります。

例えば、Web サーバで定義されたアプリケーション・パス (/myapp/csp) から Machine I の InterSystems IRIS インスタンスの **iris** に Web アプリケーションをリモートで提供するには、以下のエイリアス行を Machine W の **httpd.conf** ファイルに追加します。

```
Alias /myapp/csp "C:/iris/CSP"
```

Apache Web サーバを再起動して変更内容を適用します。

I.2 複数の InterSystems IRIS サーバ上での CSP へのアクセス

1 つの Web サーバを構成して、複数の InterSystems IRIS サーバ上で 1 つまたは複数の Web アプリケーションにアクセスする必要がある場合、このセクションを参照してください。このセクションでは、アプリケーションの例として管理ポータルを使用します。以下の手順を独自の Web アプリケーションに当てはめてください。管理ポータル・アプリケーションは、通常、以下の形式の URL で呼び出されます。

```
http://domain.com/csp/sys/UtilHome.csp
```

詳細は、“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。

I.2.1 アプリケーション・パス用の InterSystems IRIS サーバの構成

Web アプリケーションの URL に InterSystems IRIS サーバ名を使用する場合、以下の手順に従います。InterSystems IRIS サーバ名を URL に表示しない場合、次のセクション [URL での InterSystems IRIS サーバ名の変更](#) の手順に従ってください。

1. Web サーバで、以下を使用して Web ゲートウェイ[管理ページ](#)のメイン・メニューにアクセスします。

```
http://localhost:<port no>/csp/bin/Systems/Module.cwx
```

2. [\[サーバ接続\]](#)を選択します。iris1 および iris2 のサーバ構成を追加します。詳細は、“[サーバ・アクセスの構成](#)”を参照してください。
3. [\[アプリケーション・アクセス\]](#)を選択します。iris1 の [\[デフォルトサーバ\]](#)を使用して、アプリケーション・パス /iris1/csp/sys/ を作成します。iris2 の [\[デフォルトサーバ\]](#)を使用して、アプリケーション・パス /iris2/csp/sys/ を作成します。詳細は、“[アプリケーション・アクセスの構成](#)”を参照してください。
4. Web サーバが IIS の場合、/iris1 と /iris2 の仮想ディレクトリを設定します (“[IIS への仮想ディレクトリの追加](#)”を参照)。

Apache Web サーバを使用している場合、“[Apache 構成へのエイリアスの追加](#)”を参照してください。

InterSystems IRIS サーバ iris1 および iris2 上で管理ポータルにアクセスするには、次のように、サーバ名を URL の一部として含めます。

```
http://domain.com/iris1/csp/sys/UtilHome.csp  
http://domain.com/iris2/csp/sys/UtilHome.csp
```

I.2.2 URL での InterSystems IRIS サーバ名の変更

InterSystems IRIS サーバ名を Web アプリケーションの URL に表示しない場合、このセクションの手順に従って、代わりの名前を作成します。

各サーバに対して %System.CSP.SetConfig メソッドの CSPConfigName パラメータを使用します。この例では、サーバ iris1 の代わりの名前として linda、サーバ iris2 の代わりの名前として perry を使用します。独自のサーバと代わりの名前を使用する必要があります。

iris1 サーバのターミナル・ウィンドウで、以下のプログラムを実行します。

```
d $System.CSP.SetConfig("CSPConfigName","linda")
```

iris2 サーバのターミナル・ウィンドウで、以下のプログラムを実行します。

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

次に、以下の手順を実行します。

1. Web サーバで、以下を使用して Web ゲートウェイ管理ページのメイン・メニューにアクセスします。
`http://localhost/csp/bin/Systems/Module.cwx`
2. [サーバ接続] を選択します。iris1 および iris2 のサーバ構成を追加します。詳細は、“サーバ・アクセスの構成” を参照してください。
3. [アプリケーション・アクセス] を選択します。iris1 の [デフォルトサーバ] を使用して、アプリケーション・パス /linda/csp/sys/ を作成します。iris2 の [デフォルトサーバ] を使用して、アプリケーション・パス /perry/csp/sys/ を作成します。詳細は、“アプリケーション・アクセスの構成” を参照してください。
4. Web サーバが IIS の場合、/iris1 と /iris2 の仮想ディレクトリを設定します (“IIS への仮想ディレクトリの追加” を参照)。
5. Apache Web サーバを使用している場合、“Apache 構成へのエイリアスの追加” を参照してください。

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、単一値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","linda,linda1,linda2,linda3")
```

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。

CSPConfigName が設定されている場合は、WebServerURLPrefix も設定して、スタジオで同じ URL 構造を使用できます。“WebServerURLPrefix” も参照してください。

1.3 Apache 仮想ホストの構成

複数のサーバでアプリケーションにアクセスする代替方法は、仮想ホストの配置を使用することです。仮想ホストは、Apache Web サーバ構成の共通機能で、このサーバ環境では簡単に設定できます。例えば、2 つの仮想ホストがあり、それぞれが別の TCP ポートで待ち受けているとします。

```
http://virtual_host1:81/csp/sys/UtilHome.csp
http://virtual_host2:82/csp/sys/UtilHome.csp
```

virtual_host1 と virtual_host2 の両方が同じ Web サーバおよび Web ゲートウェイによって提供されます。

以下に、この配置の Apache 構成 (httpd.conf) を示します。

```
<VirtualHost virtual_host1:81>
    ServerName virtual_host1
</VirtualHost>

<VirtualHost virtual_host2:82>
    ServerName virtual_host2
</VirtualHost>
```

これらの仮想ホストの使用を以下のように構成します。

1. ブラウザで以下を指定し、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。

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

2. [サーバ接続] を選択して、iris1 および iris2 のサーバ構成を作成します。
3. [アプリケーションアクセス] を選択して、アプリケーション・パス //virtual_host1/csp/sys/ および //virtual_host2/csp/sys/ を作成します。

仮想ホスト名の前に二重スラッシュ (/ /) を使用することに注意してください。

前の手順の iris1 のサーバ構成設定の名前になるように、パス //virtual_host1/csp/sys/ の [サーバ 0] を設定します。

前の手順の iris2 のサーバ構成設定の名前になるように、パス //virtual_host2/csp/sys/ の [サーバ 0] を設定します。

4. 2 つのリモート InterSystems IRIS サーバの構成を変更する必要はありません。ポータルアプリケーション・パスは、どちらの場合も /csp/sys/ のままです。

詳細は、“[仮想ホストの概要](#)” を参照してください。

1.3.1 仮想ホストの概要

仮想ホストは、共通の Web サーバを介して 1 つまたは複数のインスタンスにアプリケーションを透過的に提供する方法です。各サーバ・インストールは、別々の Web サーバのように動作します。

仮想ホスト設定を区別する要素は、以下のいずれかです。

1. Web サーバ IP アドレス – Web サーバをホストするサーバは、2 つの IP アドレスで公開されます。以下はその例です。

```
123.123.123.1 == www.serverA.com
123.123.123.2 == www.serverB.com
```

2. Web サーバ・ポート – この方法は、異なる構成をテストする際に便利ですが、非標準の TCP ポート (80 以外の TCP ポート) が使用される場合に、要求のポート番号を組み込みます。以下はその例です。

```
Web Server TCP Port 80 == www.serverA.com
Web Server TCP Port 81 == www.serverB.com
```

3. パス – 仮想ホストを実装する際の好ましい方法です。2 つの名前を登録し、これらが Web サーバの 1 つの物理 IP アドレスに変換されます。以下はその例です。

```
www.serverA.com == 123.123.123.1
www.serverB.com == 123.123.123.1
```

どの方法を選択しても、Web ゲートウェイ構成のそれぞれの InterSystems IRIS インストールに名前付きのスロットを設定します (InterSystems IRIS インスタンス名と同じである必要はありません)。(各サーバの) Web ゲートウェイ構成が指しているスーパーサーバ・ポートが重要です。

以下はその例です。

```
www.serverA.com
www.serverB.com
```

両方とも、1 つの Web サーバ・インストールによって提供されます。

3 つすべての組み合わせを含むサーバを実装できます。オプション 1 および 3 は、ブラウザの観点では同じです。各仮想ホストを構成して、独自のドキュメント・ルートなどを作成できます。

仮想ホストの概念を CSP に拡張するには、2 つの仮想ホストを介して同じ Web アプリケーションを異なる InterSystems IRIS インスタンスで実行することを仮定します。例えば、一方のサイトをテスト用、他方を実稼働用にします。

```
www.serverA.com/csp/login.csp ==> irisA
www.serverB.com/csp/login.csp ==> irisB
```

InterSystems IRIS サーバへの Web アプリケーションのアクセスは、Web ゲートウェイの [アプリケーションアクセス] 構成オプションを使用して制御されます。通常、以下の 2 つのエントリが定義されます。

```
/
/csp
```

InterSystems IRIS サーバの名前は、これらのアプリケーション・パス定義と関連付けられます。

```
/ (Default Server == irisA)
/csp (Default Server == irisA)
```

Web ゲートウェイでは、この構成を拡張して、アプリケーションへのアクセスに使用する仮想ホストの名前を組み込むことができます。

```
/ (Default Server == irisA)
/csp (Default Server == irisA)
//www.serverA.com/csp (Default Server == irisA)
//www.serverB.com/csp (Default Server == irisB)
```

上記のように **www.serverA.com/csp** と **www.serverB.com/csp** に対して、別の InterSystems IRIS サーバを構成できます。示したように、**//** でサーバ名を付けます。

継承の現在のルールが適用されます。例えば、**www.serverA.com/xxx/yyy.csp** を要求する場合、以下に示すように **serverA** の最終的な既定値を定義しない限り、**/** に定義された InterSystems IRIS サーバが最終的に使用されます。

```
/ (Default Server == irisL)
/csp (Default Server == irisL)
//www.serverA.com/ (Default Server == irisL)
//www.serverA.com/csp (Default Server == irisA)
//www.serverB.com/csp (Default Server == irisB)
```

注釈 Web ゲートウェイ構成で指定するサーバは、仮想である必要はありません。例えば、1 つの NSD インストールを構成して、いくつかの実際の Apache インストールをそれぞれに対して定義された InterSystems IRIS サーバの異なるセットとともにサポートできます。さらに、各 Apache サーバを構成して、多数の仮想ホストをサポートできます。

Web ゲートウェイは、CGI 環境変数 **SERVER_NAME** を使用してアプリケーションのホストを特定します。

J

WebSocket の使用 (RFC 6455)

WebSocket プロトコル (RFC 6455) は、クライアントとサーバの間で全二重のメッセージ指向の通信チャンネルを指定することによって、サーバがクライアントにメッセージを事前にプッシュできるようにするという、基本的な要件に対処します。このプロトコルは、クライアントとサーバの間に既に確立されている標準 TCP チャンネルを介して動作するように (したがって保護されるように) 設計されており、Web ブラウザと Web サーバの間で HTTP プロトコルをサポートするために使用されています。

WebSocket プロトコルおよびその API は、W3C によって標準化され、クライアント部分は HTML 5 に組み込まれています。

媒体 (プロキシやファイアウォールなど) は、WebSocket プロトコルを認識していること (およびサポートしていること) が期待されています。

J.1 WebSocket プロトコル

WebSocket の作成には、クライアントとサーバの間のメッセージの順序付けされた交換が含まれます。まず、WebSocket ハンドシェイクを実行する必要があります。ハンドシェイクは、HTTP メッセージ交換に基づいていて、これに似ています。そのため、既存の HTTP インフラストラクチャを使用して問題なく渡すことができます。

- ・ クライアントは、WebSocket 接続のハンドシェイク要求を送信します。
- ・ サーバは、ハンドシェイク応答を送信します (可能な場合)。

Web サーバは、ハンドシェイク要求メッセージの従来の HTTP ヘッダ構造を認識し、WebSocket プロトコルをサポートしていることを示す、同じように構成された応答メッセージをクライアントに送信します (応答メッセージを送信できることが前提)。両者が同意した場合、チャンネルは HTTP (<http://>) から WebSocket プロトコル (<ws://>) に切り替わります。

- ・ プロトコルが正常に切り替わると、チャンネルは、クライアントとサーバの間の全二重通信を許可します。
- ・ 個別のメッセージのデータ・フレーミングは最小です。

クライアントからの典型的な WebSocket ハンドシェイク・メッセージ

```
GET /csp/user/MyApp.MyWebSocketServer.cls HTTP/1.1
Host: localhost
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Key: x3JJHMbDL1EzLkh9GBhXDw==
Sec-WebSocket-Protocol: chat
Sec-WebSocket-Version: 13
Origin: http://localhost
```

サーバからの典型的な WebSocket ハンドシェイク・メッセージ

```
HTTP/1.1 101 Switching Protocols
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Accept: HSmrc0sMlYUkAGmm5OPpG2HaGWk=
Sec-WebSocket-Protocol: chat
```

プロトコルを HTTP から WebSocket にアップグレードすることをクライアントのハンドシェイク・メッセージが要求する方法に注意してください。クライアント (Sec-WebSocket-Key) とサーバ (Sec-WebSocket-Accept) の間の一意のキーの交換にも注意してください。

J.2 WebSocket のクライアント・コード (JavaScript)

ブラウザ環境では、クライアントは JavaScript を使用します。標準のテキスト本に使用法のモデルについて詳細に説明されています。ここでは、その基本について簡単に説明します。

J.2.1 WebSocket の作成

最初のパラメータは、WebSocket アプリケーションのサーバ側を特定する URL を表します。2 番目のパラメータは任意です。これがある場合、WebSocket 接続が成功することをサーバがサポートする必要がある、サブプロトコルを指定します。

```
var ws = new WebSocket(url, [protocol]);
```

例:

```
ws = new WebSocket(((window.location.protocol == "https:")
  ? "wss:" : "ws:")
  + "://" + window.location.host
  + "/csp/user/MyApp.MyWebSocketServer.cls");
```

基盤となるトランスポートが SSL/TLS を使用して保護されているかどうかに応じて、プロトコルが `ws` または `wss` に定義される方法に注意してください。

読み取り専用属性 `ws.readyState` は、接続の状態を定義します。以下の値のいずれかになります。

- ・ 0 : 接続はまだ確立されていません。
- ・ 1 : 接続が確立され、通信可能です。
- ・ 2 : 接続は閉じているハンドシェイクに従います。
- ・ 3 : 接続が閉じられているかまたは開くことができませんでした。

読み取り専用属性 `ws.bufferedAmount` は、`send()` メソッドを使用してキューに入っている UTF-8 テキストのバイト数を定義します。

J.2.2 WebSocket のクライアント・イベント

以下のイベントを使用できます。

- ・ `ws.onopen` : ソケット接続が確立されるとトリガされます。
- ・ `ws.onmessage` : クライアントがサーバからデータを受信するとトリガされます。

`event.data` で受信されるデータ。

- ・ `ws.onerror` : 通信でエラーが発生するとトリガされます。
- ・ `ws.onclose` : 接続が閉じられるとトリガされます。

J.2.3 WebSocket のクライアント・メソッド

以下のメソッドを使用できます。

- ・ `ws.send(data)` : クライアントにデータを送信します。
- ・ `ws.close()` : 接続を閉じます。

J.3 WebSocket のサーバ・コード

WebSocket サーバを実装するベース InterSystems IRIS® クラスは `%CSP.WebSocket` です。

クライアントが WebSocket 接続を要求すると、初期 HTTP 要求 (初期ハンドシェイク・メッセージ) は、アプリケーションの WebSocket サーバを初期化するように [CSP エンジン](#) に指示します。WebSocket サーバは、要求元の URL で名前を付けられたクラスです。例えば、WebSocket サーバが `MyApp.MyWebSocketServer` という名前で、**USER** ネームスペースで動作するように設計されている場合、WebSocket 接続を要求するために使用される URL は以下のようになります。

```
/csp/user/MyApp.MyWebSocketServer.cls
```

J.3.1 WebSocket のサーバ・イベント

WebSocket サーバを実装するには、`%CSP.WebSocket` のサブクラスを作成し、そのクラスに必要な応じてコールバックを定義します。これらのメソッドのいずれかを呼び出す前に、Web セッションがロック解除されることに注意してください。

OnPreServer()

このメソッドを実装して、WebSocket サーバが確立される前に実行する必要があるコードを呼び出します。
SharedConnection プロパティの変更はここで行う必要があります。

Server() (required)

このメソッドを実装して、WebSocket サーバを作成します。これは、WebSocket アプリケーションのサーバ側の実装です。Read() メソッドおよび Write() メソッドを使用して、クライアントとメッセージを交換できます。EndServer() メソッドを使用して、サーバ側から WebSocket を正常に閉じます。

OnPostServer()

このメソッドを実装して、WebSocket サーバが閉じられた後に実行する必要があるコードを呼び出します。

J.3.2 WebSocket のサーバ・メソッド

これらのコールバック内から以下のメソッドを呼び出すことができます。

Read()

```
Method Read(ByRef len As %Integer = 32656,
            ByRef sc As %Status,
            timeout As %Integer = 86400) As %String
```

このメソッドは、クライアントから `len` で指定された文字まで読み取ります。呼び出しに成功すると、ステータス (`sc`) は `$$$OK` として返されます。失敗すると、以下のエラー・コードのいずれかが返されます。

- ・ `$$$CSPWebSocketTimeout` : 読み取りメソッドはタイムアウトになりました。
- ・ `$$$CSPWebSocketClosed` : クライアントが WebSocket を終了しました。

Write()

```
Method Write(data As %String) As %Status
```

このメソッドは、データをクライアントに書き込みます。

EndServer()

```
Method EndServer() As %Status
```

このメソッドは、クライアントとの接続を閉じることによって WebSocket サーバを正常に終了します。

OpenServer()

```
Method OpenServer(WebSocketID As %String = "") As %Status
```

このメソッドは、既存の WebSocket サーバを開きます。このメソッドを使用してアクセスできるのは、非同期で動作している WebSocket (`SharedConnection=1`) のみです。

J.3.3 WebSocket のサーバ・プロパティ

これらのコールバック内から以下のプロパティを設定または取得することができます。

SharedConnection (default: 0)

このプロパティによって、クライアントと WebSocket サーバの間の通信を専用の [Web ゲートウェイ](#) 接続を介して行う必要があるか、または共有 Web ゲートウェイ接続のプールを介して非同期で行う必要があるかが決まります。このプロパティは、`OnPreServer()` メソッドで設定する必要があり、以下のように設定できます。

- ・ `SharedConnection=0` : WebSocket サーバは、専用の Web ゲートウェイ接続を介してクライアントと同期して通信します。このモードの処理では、ホスト接続は、アプリケーションの WebSocket サーバに対して事実上「プライベート」です。
- ・ `SharedConnection=1` : WebSocket サーバは、共有 Web ゲートウェイ接続のプールを介してクライアントと非同期で通信します。また、CSP セッション・タイムアウトの期間アクティビティがないと、ソケットがタイムアウトします。

WebSocketID

このプロパティは、WebSocket の一意の ID を表します。

SessionId

このプロパティは、WebSocket が作成されたホスト CSP セッション ID を表します。

BinaryData

このプロパティは、転送されたデータ・ストリームを UTF-8 エンコード・テキストとして解釈する機能をバイパスし、WebSocket フレーム・ヘッダに適切なバイナリ・データ・フィールドを設定するよう Web ゲートウェイに指示します。

これは、バイナリ・データのストリームをクライアントに書き込む前に 1 に設定する必要があります。次に例を示します。

```
Set ..BinaryData = 1
```

J.4 WebSocket サーバの例

以下の単純な WebSocket サーバ・クラスは、クライアントからの接続を受け入れ、受信したデータを単純にエコー・バックします。

タイムアウトは 10 秒に設定され、Read() メソッドがタイムアウトになるたびに、メッセージがクライアントに書き込まれます。これは、WebSocket を実証する重要な概念の 1 つ（サーバからクライアントとのメッセージ交換の開始）を示します。

最後に、クライアント（つまりユーザ）が文字列 exit を送信すると、WebSocket は正常に閉じます。

```
Method OnPreServer() As %Status
{
    Quit $$$OK
}

Method Server() As %Status
{
    Set timeout=10
    For {
        Set len=32656
        Set data=..Read(.len, .status, timeout)
        If $$$ISERR(status) {
            If $$$GETERRORCODE(status) = $$$CSPWebSocketClosed {
                Quit
            }
            If $$$GETERRORCODE(status) = $$$CSPWebSocketTimeout {
                Set status=..Write("Server timed-out at "._$Horolog)
            }
        }
        else {
            If data="exit" Quit
            Set status=..Write(data)
        }
    }
    Set status=..EndServer()
    Quit $$$OK
}

Method OnPostServer() As %Status
{
    Quit $$$OK
}
```

J.5 WebSockets サーバの非同期動作

前の節の例は、専用の InterSystems IRIS 接続を介して、WebSocket サーバがクライアントと同期して動作する例を示しています。このような接続が確立されると、Web ゲートウェイ・システム・ステータス・フォームのステータス列で WebSocket というラベルが付けられます。このモードでは、WebSocket は、ホスト Web セッションのセキュリティ・コンテキスト内で動作し、そのセッションに関連付けられたすべてのプロパティに簡単にアクセスできます。

非同期モードでの動作 (SharedConnection=1) では、ホスト接続は、WebSocket オブジェクトが作成されるとすぐに解放され、クライアントとのその後の対話は、共有接続のプールを介して行われます。クライアントからのメッセージは、Web ゲートウェイ接続の従来のプールを介して InterSystems IRIS に到達し、クライアントへのメッセージは、Web ゲートウェイと InterSystems IRIS の間に確立されたサーバ接続のプールを介して送信されます。

非同期モードでは、WebSocket サーバは、メイン Web セッションから分離されます。つまり、**SessionId** プロパティは、ホスト・セッション ID の値を保持しますが、セッション・オブジェクトのインスタンスは自動的に作成されません。

前に示した例は、単純に OnPreServer() メソッドの **SharedConnection** プロパティを設定することによって、非同期で実行できます。ただし、InterSystems IRIS プロセスを永続的に WebSocket と関連付ける必要はありません。Server() メソッドは、WebSocket を閉じずに終了できます (ホスト・プロセスは停止します)。WebSocketID が保持された場合、WebSocket は、次に別の InterSystems IRIS プロセスで開くことができ、クライアントとの通信が再開されます。

以下の例で、MYAPP.SAVE() と MYAPP.RETRIEVE() は WebSocket ID を保存および取得するために作成するカスタム・コードのプレースホルダです。

例：

```
Class MyApp.MyWebSocketServer Extends %CSP.WebSocket
{
    Method OnPreServer() As %Status
    {
        MYAPP.SAVE(..WebSocketID)
        Set ..SharedConnection = 1
        Quit $$$OK
    }

    Method Server() As %Status
    {
        Quit $$$OK
    }

    Method OnPostServer() As %Status
    {
        Quit $$$OK
    }
}
```

WebSocketID は OnPreServer() メソッドを次に使用できるように保持されることに注意してください。また、OnPreServer() メソッドおよび Server() メソッドの **SharedConnection** プロパティの設定は単に終了することにも注意してください。

WebSocketID の後続の取得：

```
Set WebSocketID = MYAPP.RETRIEVE()
```

クライアントとのリンクの再確立：

```
Set ws=##class(%CSP.WebSocket).%New()
Set %status = ws.OpenServer(WebSocketID)
```

クライアントからの読み取りとクライアントへの書き込み：

```
Set %status=ws.Write(message)
Set data=ws.Read(.len, .%status, timeout)
```

最後に、サーバ側からの WebSocket のクローズ：

```
Set %status=ws.EndServer()
```