



# CSP ゲートウェイのインストールと構成

Version 5.1  
2006-03-14

CSP ゲートウェイのインストールと構成

Caché Version 5.1 2006-03-14

Copyright © 2006 InterSystems Corporation.

All rights reserved.

このドキュメントは、Sun Microsystems、RenderX Inc.、アドビ システムズ および ワールドワイド・ウェブ・コンソーシアム (www.w3c.org) のツールと情報を使用して、Adobe Portable Document Format (PDF) で作成およびフォーマットされました。主要ドキュメント開発ツールは、InterSystemsが構築したCaché と Javaを使用した特別目的のXML処理アプリケーションです。



Caché 製品とロゴは InterSystems Corporation の登録商標です。



Ensemble 製品とロゴは InterSystems Corporation の登録商標です。



InterSystems という名前とロゴは InterSystems Corporation の登録商標です

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

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

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

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

Caché および InterSystems Caché、Caché SQL、Caché ObjectScript および Caché Object は、インターシステムズ社の商標です。

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。

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 ワールドワイド カスタマサポート

Tel: +1 617 621-0700

Fax: +1 617 374-9391

Email: support@InterSystems.com

# 目次

1 はじめに .....	1
2 Microsoft Windows の Web サーバ .....	3
2.1 Microsoft Peer Web Server (PWS) と Internet Information Services (IIS) .....	3
2.1.1 インストール .....	4
2.1.2 オプション 1:ISAPI モジュール (CSPms*.dll) を使用する .....	5
2.1.3 オプション 2:NSD で ISAPI モジュール (CSPcms.dll) を使用する .....	10
2.1.4 オプション 3:NSD で CGI モジュール (nph-CSPcgi*.exe) を使用する .....	12
2.2 Netscape、iPlanet、および Sun の Web サーバ .....	14
2.2.1 インストール .....	15
2.2.2 オプション 1:NSAPI モジュール (CSPn3*.dll) を使用する .....	16
2.2.3 オプション 2:NSD で ISAPI モジュール (CSPcn3.dll) を使用する .....	18
2.3 Apache サーバ .....	19
2.3.1 インストール (すべての接続オプション) .....	20
2.3.2 オプション 1:Apache API モジュール (CSPa*.dll) を使用する .....	21
2.3.3 オプション 2:NSD で CGI モジュール (nph-CSPcgi*.exe) を使用する .....	23
2.3.4 オプション 3:NSD で Apache API モジュール (mod_csp*.dll) を使用する .....	26
2.3.5 オプション 4:ISAPI モジュール (CSPms*.dll) を使用する .....	27
2.4 NSD (Network Service Daemon) の操作 .....	29
3 UNIX、LINUX、および Mac OS X の Web サーバ .....	31
3.1 Netscape、iPlanet、および Sun の Web サーバ .....	31
3.1.1 インストール .....	32
3.1.2 オプション 1:NSAPI モジュール (CSPn3*.so) を使用する .....	34
3.1.3 オプション 2:NSD で NSAPI モジュール (CSPcn3.so) を使用する .....	35
3.2 Apache サーバ .....	37
3.2.1 インストール (すべての接続オプション) .....	37
3.2.2 オプション 1:NSD で動的 Apache API モジュール (mod_csp*.so) を使用する .....	39
3.2.3 オプション 2:NSD で CGI モジュール (nph-CSPcgi*) を使用する .....	42
3.2.4 オプション 3:NSD で組み込み Apache API モジュール (mod_csp.c) を使用する .....	42
4 .....	5
3.3 NSD (Network Service Daemon) を操作する .....	50
4 Web サーバ構成に関する一般的な問題 .....	53
4.1 付属の CGI モジュールについて .....	53
4.2 ゲートウェイのシステム管理スイートへアクセスする際のセキュリティ .....	54
4.3 Web サーバの既定ページ (“ホームページ”) としての CSP ページの設定 .....	55
4.3.1 Peer Web Server .....	56
4.3.2 Internet Information Services .....	56

4.3.3 Netscape、iPlanet、および Sun の Web サーバ .....	57
4.4 Apache サーバ .....	58

# 1

## はじめに

このドキュメントでは、Cache がサポートしているすべての Web サーバに対応できるように CSP ゲートウェイを構成する手順を説明します。CSP ゲートウェイは、ホスト Web サーバと Cache の間の通信層として機能します。

ほとんどの場合、Cache 配布キットに用意された自動インストール手順を実行すればシステムを起動できるので、このドキュメントで説明されている手順を手動で実行する必要はありません。ただし、Web サーバ・アーキテクチャが非標準の場合や、現在の環境でのさらなるパフォーマンス向上を望む上級ユーザを対象に、ここでは、インストールと構成に関する重要事項を説明します。

オペレーティング・システム	Web サーバ
Microsoft Windows	Microsoft – Internet Information Services (IIS) (Peer Web Server を含む)
	Sun – Netscape、iPlanet、Sun ONE、Sun Java System
	Apache
UNIX	Sun – Netscape、iPlanet、Sun ONE、Sun Java System
	Apache

CSP には、Microsoft、Netscape/iPlanet/Sun、および Apache Web サーバを対象とする高性能な接続ソリューションが用意されています。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 (Common Gateway Interface) 経由で Cache に接続できます。さらに、このドキュメントに記載されていない Web サーバもサポート対象となる可能性があります。

Microsoft と Netscape/iPlanet/Sun の Web サーバはどちらもマルチスレッド API をサポートしており、Web サーバのコア機能として、拡張機能を動的に取り込むことができます。CSP 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、Cache システムへの高性能な Web 接続を実

## はじめに

---

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 CSP ゲートウェイを活用できます。

UNIX バージョンの Apache は、Microsoft や Netscape/iPlanet/Sun の Web サーバとはアーキテクチャが異なり、マルチスレッド・サーバとしては動作しません。したがって、マルチスレッド方式の API は使用できません。これらのバージョンの Apache (および同様に設計されている Web サーバ) をサポートするため、マルチスレッド CSP ゲートウェイはスタンドアロンの実行可能コードとして実装され、Web サーバとは別にデーモン・プロセスとして動作します。こうした CSP ゲートウェイのスタンドアロン実装を NDS (Network Service Daemon) といいます。Web サーバは、ホスト Web サーバの独自仕様の API に対して動作する小さな専用モジュールを介して、または標準の CGI (Common Gateway Interface) を介して NSD と通信します。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 も発行します。

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。そのために、“CSP モジュール” のソース・コードが用意されています。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、バージョン 1.3 以降の Apache 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) または UNIX の “動的共有オブジェクト” (DSO) として構築できます。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。したがって、直接組み込む方法よりも、この方法を使用することをお勧めします。サポートされているほとんどのオペレーティング・システムでは、CSP モジュールが事前に組み込まれています。

前述の接続オプションに加え、高性能な CGI ベースのソリューションもあります。これは、Apache で事前組み込みのモジュールを使用できず、しかも、ソースからモジュールを組み込んだり、Apache コアへモジュールを直接組み込んだりしない場合に対応するためのものです。

CSP モジュールと CGI 実行可能コードはいずれも、排他的に NSD とやり取りできるように設計された小さな機能ブロックです。この場合、NSD は CSP ゲートウェイのコア機能を提供し、Caché との永続的な接続を実現します。CSP モジュールは、CGI ベースのモジュールよりもパフォーマンスに優れています。CSP モジュールはホスト Apache プロセスへ直接接続されます。一方、CGI モジュールは、それ自体のプロセスで開始する必要があります。小規模なシステムの場合、CSP モジュールと CGI 実行可能コードの間におけるパフォーマンス上の違いはそれほど顕著ではありません。ただし、システムの負荷が増大するにつれて、このパフォーマンスの違いも顕著となります。さらに、CGI ベースの接続モデルにおいては、余分な数のプロセスを開始および管理することが必要となり、それによるオーバーヘッドも明白となります。

# 2

## Microsoft Windows の Web サーバ

このセクションでは、Microsoft、Sun、および Apache の Web サーバについて説明します。オリジナルの Netscape Web サーバ (Enterprise および FastTrack) は、ごく最近まで “iPlanet” という名前で販売されていましたが、現在では、“Sun ONE” サーバ・スイートとして Sun によって開発および販売されています。Sun は Sun ONE Web サーバを再び改名し、このドキュメントの執筆時点では “Sun Java System Web サーバ” という名前に変更されています。

すべての Microsoft の Web サーバは、高性能 API によって拡張することができます。ISAPI (Internet Server Application Programming Interface) を使用すれば、Windows DLL として実装された “ISAPI 拡張” を通じて Web サーバを拡張できます。

Netscape/iPlanet/Sun の Web サーバも、高性能 API による機能拡張が可能です。NSAPI (NetScape Application Programming Interface) を使用することにより、Windows DLL として実装されたモジュールを介して Web サーバを拡張できます。

Apache にはいくつかの接続オプションがあります。インストールと構成が最も簡単なのは CGI ベースのソリューションです。Apache グループは、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もサポートしており、Microsoft の Web サーバ用に開発された ISAPI 拡張を利用するための手段も提供しています。

### 2.1 Microsoft Peer Web Server (PWS) と Internet Information Services (IIS)

PWS は、本質的には IIS の軽量バージョンです。通常、PWS は Microsoft のクライアント指向オペレーティング・システム (Windows 95、98、NT、WorkStation、Me) に搭載され、IIS はサーバ指向オペレーティング・システム (Windows NT Server、2000、2003) に搭載されています。Windows XP は、実質的にはクライアント指向オペレーティング・システムですが、IIS サーバを搭載しています。

このドキュメントは、以下のファイル・システムにインストールされている、さまざまな CSP およびゲートウェイ Web サーバのコンポーネントに基づいて作成されています。

C:\cachesys\csp\

システムのレイアウトが異なる場合は、必要に応じて、この後のセクションで説明する構成指示文を修正してください。

## 2.1.1 インストール

以下の CSP 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てください。

### 1. NSD モジュール (必要な場合)

- ・ **CSPnsd.exe**
- ・ **CSPnsdSv.exe**

これらのモジュールの既定の場所は **C:\cachesys\csp** です。

NSD は上記のホーム・ディレクトリ内から実行されます (上記参照)。NSD ベースの接続オプションの場合、構成ファイル (**CSP.INI**) とイベント・ログ (**CSP.LOG**) はこのディレクトリに書き込まれます。

### 2. ISAPI および CGI のモジュール

- ・ **CSPms.dll** (実行時モジュール)
- ・ **CSPmsSys.dll** (システム管理モジュール)
- ・ **CSPcms.dll** (提供される場合は、NSD の ISAPI クライアント)
- ・ **CSPcgi.exe** (実行時モジュール)
- ・ **nph-CSPcgi.exe** (CSPcgi のコピー)
- ・ **CSPcgiSys.exe** (システム管理モジュール)
- ・ **nph-CSPcgiSys.exe** (CSPcgiSys のコピー)

上記すべてのモジュールが各接続オプションで必要になるわけではありません。各オプションについてのセクションを参照し、実際に必要なモジュールを確認してください。

これらのモジュールの既定の場所は **C:\cachesys\csp\bin** です。

NSD ベース以外の接続オプションの場合、構成ファイル (**CSP.INI**) とイベント・ログ (**CSP.LOG**) はこのディレクトリに書き込まれます。

### 3. ハイパーイベントのコンポーネント

- ・ **CSPBroker.js**

- ・ CSPBroker.class
- ・ CSPBroker.jar
- ・ CSPxmlhttp.js

これらのファイルの既定の場所は `C:\%cachesys%\csp%broker` です。

#### 4. 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 `C:\%cachesys%\csp\samples` です。

## 2.1.2 オプション 1: ISAPI モジュール (CSPms\*.dll) を使用する

CSP 要求 (ファイル・タイプ .csp および .cls) を認識し、それらを処理担当の CSP ゲートウェイへ渡すように Web サーバを構成する必要があります。

### 2.1.2.1 Peer Web Server

1. HKEY\_LOCAL\_MACHINE (HKLM) キー・セット内の以下の 2 つのキー値を Windows レジストリに追加します。これらのキーは、CSP ファイルの取得要求を CSP ゲートウェイに渡して処理するよう、PWS に指示します。

REGEDIT4

```
[HKLM\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\Script Map]
".csp"="C:\\cachesys\\csp\\bin\\CSPms.dll"
```

REGEDIT4

```
[HKLM\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\Script Map]
".cls"="C:\\cachesys\\csp\\bin\\CSPms.dll"
```

2. [インターネット サービス マネージャ] ウィンドウを開きます。
3. [WWW サービス] をダブル・クリックして、[WWW サービス プロパティ] ウィンドウを表示します。
4. [ディレクトリ] タブをクリックします。
5. [追加] ボタンをクリックし、以下の仮想ディレクトリを入力します。
  - ・ [ディレクトリ]: `c:\%cachesys%\csp`
  - ・ [仮想ディレクトリ]: チェックを付ける
  - ・ [エイリアス]: `/csp`
  - ・ [アクセス]: [読み取り] と [実行] にチェックを付ける
6. [保存] および [適用] をクリックしてすべての変更を保存し、コンピュータを再起動します。

## 2.1.2.2 Internet Information Services

1. [インターネット サービス マネージャ] ウィンドウを開きます。
2. 左側のウィンドウで、[既定の Web サイト] を選択します。
3. [既定の Web サイト] を右クリックします。メニューから [プロパティ] を選択し、[既定の Web サイトのプロパティ] ウィンドウを開きます。
4. [ホーム ディレクトリ] タブをクリックします。
5. [構成] ボタンをクリックして、[アプリケーション構成] ウィンドウを開きます。
6. [アプリケーションのマッピング] タブをクリックします。
7. [追加] ボタンをクリックして [アプリケーションの拡張子マッピングの追加/編集] ウィンドウを開き、以下のレコードを追加します。
  - ・ [実行可能ファイル]: c:\%cachesys%\csp\bin\CSPms.dll
  - ・ [拡張子]: csp
  - ・ [すべての動詞]: チェックを付ける
  - ・ [スクリプト エンジン]: チェックを付ける
  - ・ [ファイルの存在を確認する]: チェックを付けない
8. 以上の手順を繰り返し、以下のレコードを追加します。
  - ・ [実行可能ファイル]: c:\%cachesys%\csp\bin\CSPms.dll
  - ・ [拡張子]: cls
  - ・ [すべての動詞]: チェックを付ける
  - ・ [スクリプト エンジン]: チェックを付ける
  - ・ [ファイルの存在を確認する]: チェックを付けない
9. [インターネット インフォメーション サービス] ウィンドウに戻り、[既定の Web サイト] を選択します。
10. [既定の Web サイト] を右クリックし、表示されたメニューから [新規作成]、[仮想ディレクトリ] の順に選択すると、[仮想ディレクトリの作成] ウィンドウが開きます。以下のレコードを追加します。
  - ・ [エイリアス]: csp
  - ・ [ディレクトリ]: c:\%cachesys%\csp
  - ・ 実行の許可: チェックを付ける
11. [保存] および [適用] をクリックしてすべての変更を保存し、コンピュータを再起動します。

IIS バージョン 6 (Windows 2003 に搭載) に関する詳細は、次のセクションを参照してください。

### 2.1.2.3 Internet Information Services バージョン 6

このバージョンの IIS は、Windows Server 2003 (以前の Windows .Net Server 2003) に搭載されている Web サーバです。このサーバを使用できるように CSP を構成するには、CSP ゲートウェイ ISAPI DLL (CSPms.dll と CSPmsSys.dll) を “許可された” Web サービス拡張として登録する必要があります。

1. **[インターネット サービス マネージャ]** ウィンドウを開きます。
2. 左側のウィンドウで、**[Web サービス拡張]** に移動します。右側のパネルに、現在構成されている拡張 (またはアプリケーション) が一覧表示されます。
3. **[Web サービス拡張]** を右クリックします。メニューから **[新しい Web サービス拡張を追加]** を選択すると、**[新しい Web サービス拡張]** ウィンドウが表示されます。
4. **[拡張名]** フィールドに「CSP Gateway」と入力します。
5. **[追加]** ボタンをクリックして、**[ファイルの追加]** ダイアログ・ボックスを開きます。CSPms.dll を追加します (この DLL の完全な物理パスを指定してください)。CSPmsSys.dll についても、同様の手順を繰り返します。
6. **[拡張の状態を許可済みに設定する]** チェック・ボックスにチェックを付けます。
7. **[OK]** ボタンをクリックして、Web サービス拡張を保存します。

すべての ISAPI 拡張 ([すべての不明な ISAPI 拡張]) へのユーザ・アクセスを許可するオプションもあります。このオプションを有効にすると、CSP ゲートウェイの ISAPI モジュールへ自動的にアクセスできるようになります。ただし、セキュリティ上の理由から、上記の手順に従うことをお勧めします。その他のアクセスは CSP ゲートウェイ・モジュールにのみ付与してください。

ゲートウェイのモジュールは ISAPI 拡張なので、これを ISAPI フィルタとして登録するのは間違いです。CSP がまったく使用できなくなってしまうので、間違わないように注意してください。

登録した Web サービス拡張について、さらに以下の操作を実行できます。

#### CSP へのアクセスを禁止するには

CSP ゲートウェイ、および CSP ベースのすべてのリソースへのアクセスを一時的に無効にするには、以下の手順を実行します。

1. **[インターネット サービス マネージャ]** ウィンドウを開きます。
2. 左側のウィンドウで、**[Web サービス拡張]** に移動します。右側のパネルに、現在構成されている拡張 (またはアプリケーション) が一覧表示されます。
3. 右側のウィンドウで、**[CSP Gateway]** を右クリックして **[禁止]** を選択します。CSP ゲートウェイへのアクセスが禁止されました。

CSP ゲートウェイへのアクセスを再び有効にする場合は、上記の最後の手順で [許可] を選択します。

### CSP 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禁止するには

CSP ゲートウェイのシステム管理スイートへのアクセスを禁止するには、以下の手順に従います。この手順を実行すると、承認されていないユーザは、実行中のシステムのシステム管理スイートへアクセスできなくなります。この方法を使用すれば、その後、システム管理者がゲートウェイの構成を変更する際、アクセス許可を簡単に元に戻すことができます。

1. [インターネット サービス マネージャ] ウィンドウを開きます。
2. 左側のウィンドウで、[Web サービス拡張] に移動します。右側のパネルに、現在構成されている拡張 (またはアプリケーション) が一覧表示されます。
3. 右側のウィンドウで [CSP Gateway] をダブルクリックし、[Web サービス拡張のプロパティ] ウィンドウを開きます。
4. [必要なファイル] タブをクリックします。
5. [CSPmsSys.dll] をクリックして、このファイルを強調表示 (または選択) します。
6. 右側の [禁止] ボタンをクリックします。次に、ダイアログ・ボックスの下部にある [適用] ボタンと [OK] ボタンを順にクリックします。

CSP ゲートウェイ・システム管理モジュールへのアクセスを再び有効にする場合は、上記の最後の手順で [許可] ボタンをクリックします。

ゲートウェイの構成を大きく変更する間に、エンドユーザが CSP リソースへのアクセスを取得することを防止する場合にも、上記の手順が使用できます。その場合、システム管理モジュール (CSPmsSys.dll) ではなく、ゲートウェイの実行時モジュール (CSPms.dll) が “禁止” としてマークされます。

## 2.1.2.4 セキュリティの設定

多くの Windows のインストールでは (特に Windows 2000 以降のバージョン)、Web サーバ (PWS または IIS) に割り当てられた既定の権限は、CSP ゲートウェイが構成ファイル (CSP.ini) やログファイル (CSP.log) との読み込みや書き込みを行うには十分ではありません。

したがって、Web サーバの読み込みと書き込みの権限をゲートウェイのファイルに割り当てるか、または Web サーバの “Administrator” 権限を与える必要があります。

ファイル・アクセス権限は、Windows エクスプローラで変更できます。または、以下の 2 つのコマンドを使用することもできます。

```
cacls c:\cachesys\csp\bin\CSP.ini /E /G IUSR_xxx:F  
cacls c:\cachesys\csp\bin\CSP.log /E /G IUSR_xxx:F
```

“IUSR\_XXX” は、Web サーバのユーザ権限です。通常、“xxx” コンポーネントはコンピュータの名前です。インターネット サービス マネージャで特定の名前を検索するには、以下の手順で [認証方法] ダイアログ・ボックスを開きます。

1. [インターネット サービス マネージャ] ウィンドウを開きます。
2. 左側のウィンドウで、[既定の Web サイト] を選択します。
3. [既定の Web サイト] を右クリックします。  
メニューから [プロパティ] を選択して、[既定の Web サイトのプロパティ] ウィンドウを表示します。
4. [ディレクトリ セキュリティ] タブをクリックします。
5. [匿名アクセスおよび認証コントロール] パネルの [編集] ボタンをクリックします。[認証方法のユーザ名] ダイアログ・ボックスが表示されます。

“cacls” コマンドを実行する前に、その適用先となるファイルを作成する必要があります (ファイルが存在しない場合)。(Windows の [コマンド プロンプト] ウィンドウまたは “DOS” ボックスで) “copy con” コマンドを使用して、空のファイルを作成します。

```
Copy con c:\cachesys\csp\bin\CSP.ini
^Z
```

```
Copy con c:\cachesys\csp\bin\CSP.log
^Z
```

各コマンド行は “キャリッジ・リターン” で終了します。“^Z” は “Control-Z” を表します。

例 :

以下のコマンドを使用して、BOSTON という名前のコンピュータに対し、CSP ゲートウェイの構成とログ・ファイル・アクセス権を指定します。

```
cacls c:\cachesys\csp\bin\CSP.ini /E /G IUSR_BOSTON:F
cacls c:\cachesys\csp\bin\CSP.log /E /G IUSR_BOSTON:F
```

### 2.1.2.5 ゲートウェイの運用と管理

CSP 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、ブラウザで以下を指定します。

```
http://<ip_address>/csp/bin/CSPmsSys.dll.
```

承認され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、“セキュリティ” を参照してください。

.csp や .cls 拡張子を含むファイルが要求されると、CSP エンジンが自動的に呼び出されます。次に例を示します。

```
http://<ip_address>/csp/samples/menu.csp
```

## 2.1.3 オプション 2: NSD で ISAPI モジュール (CSPcms.dll) を使用する

ほとんどの環境では、包括的な ISAPI ベースのソリューション (オプション 1) 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 サーバとは別にゲートウェイを管理する必要がある場合は、ISAPI 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ゲートウェイ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、それぞれのゲートウェイ・インスタンスにバインドします。

この方法は、オプション 3 で説明する CGI/NSD 複合型よりもパフォーマンスに優れています。要求を処理するたびに新しいプロセスを開始する場合は待ち時間が長くなりますが、この実装ではその心配がありません。

### 2.1.3.1 Peer Web Server

オプション 1 の指示に従います。ただしレジストリ設定では、CSPms.dll ではなく、CSPcms.dll に CSP ファイルを関連付けます。

REGEDIT4

```
[HKLM\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\Script Map]
".csp"="C:\cachesys\csp\bin\CSPcms.dll"
```

REGEDIT4

```
[HKLM\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\Script Map]
".cls"="C:\cachesys\csp\bin\CSPcms.dll"
```

### 2.1.3.2 Internet Information Services

オプション 1 の指示に従います。ただし CSP ファイルは、CSPms.dll ではなく、CSPcms.dll に関連付けてください (手順 7 および 8)。

- ・ [実行可能ファイル]: c:\cachesys\csp\bin\CSPcms.dll
- ・ [拡張子]: csp
- ・ [すべての動詞]: チェックを付ける
- ・ [スクリプト エンジン]: チェックを付ける
- ・ [ファイルの存在を確認する]: チェックを付けない
- ・ [実行可能ファイル]: c:\cachesys\csp\bin\CSPcms.dll
- ・ [拡張子]: cls
- ・ [すべての動詞]: チェックを付ける
- ・ [スクリプト エンジン]: チェックを付ける

- ・ [ファイルの存在を確認する]:チェックを付けない

IIS バージョン 6 (Windows 2003 に搭載) に関する詳細は、次のセクションを参照してください。

### 2.1.3.3 Internet Information Services バージョン 6

オプション 1 の指示に従います。ただし、CSP ゲートウェイについては、**CSPms.dll** および **CSPmsSys.dll** ではなく、以下の実行可能ファイルを許可されたものとして登録します。

- ・ **CSPcms.dll**
- ・ **nph-CSPcgi.exe**
- ・ **nph-CSPcgiSys.exe**

#### CSP へのアクセスを禁止するには

以下の実行可能ファイルを “禁止” に設定します。

- ・ **CSPcms.dll**
- ・ **nph-CSPcgi.exe**
- ・ **nph-CSPcgiSys.exe**

#### CSP 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禁止するには

以下の実行可能ファイルを “禁止” に設定します。

- ・ **nph-CSPcgi.exe**
- ・ **nph-CSPcgiSys.exe**

#### CSP ゲートウェイの実行時モジュールへのアクセスを禁止するには

実行可能ファイル **CSPcms.dll** を “禁止” に設定します。

### 2.1.3.4 ゲートウェイの運用と管理

この接続オプションは、CSP ゲートウェイの Network Service Daemon (NSD) を使用します。

- ・ このサービスに関するセクションの説明に従って、CSP NSD を起動します。
- ・ 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。

CSP ページは高性能なモジュール (**mod\_csp.so**) によって管理されますが、ゲートウェイの管理スイートへアクセスするには専用の CGI モジュール (**nph-CSPcgiSys**) を使用します。

CSP 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、ブラウザで以下を指定します。

`http://<ip_address>/csp-bin/nph-CSPcgiSys`

承認され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、“セキュリティ”を参照してください。

.csp や .cls 拡張子を含むファイルが要求されると、CSP エンジンが自動的に呼び出されます。次に例を示します。

```
http://<ip_address>/csp/samples/menu.csp
```

## 2.1.4 オプション 3: NSD で CGI モジュール (nph-CSPcgi\*.exe) を使用する

ほとんどの環境では、包括的な ISAPI ベースのソリューション (オプション 1) 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。ただし、運用上の理由から、ホスト Web サーバとは別にゲートウェイを管理する必要がある場合は、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ゲートウェイ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。

### 2.1.4.1 Peer Web Server

オプション 1 の指示に従います。ただしレジストリ設定では、CSPms.dll ではなく、nph-CSPcgi.exe に CSP ファイルを関連付けてください。

```
REGEDIT4
```

```
[HKLM\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\Script Map]
".csp"="C:\\cachesys\\csp\\bin\\nph-CSPcgi.exe"
```

```
REGEDIT4
```

```
[HKLM\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\Script Map]
".cls"="C:\\cachesys\\csp\\bin\\nph-CSPcgi.exe"
```

### 2.1.4.2 Internet Information Services

オプション 1 の指示に従います。ただし CSP ファイルは、CSPms.dll ではなく、nph-CSPcgi.exe に関連付けてください (手順 7 および 8)。

- ・ [実行可能ファイル]: c:¥cachesys¥csp¥bin¥nph-CSPcgi.exe
- ・ [拡張子]: csp
- ・ [すべての動詞]: チェックを付ける
- ・ [スクリプト エンジン]: チェックを付ける
- ・ [ファイルの存在を確認する]: チェックを付けない

- ・ [実行可能ファイル]: c:\%cachesys%\csp\bin\nph-CSPcgi.exe
- ・ [拡張子]: cls
- ・ [すべての動詞]: チェックを付ける
- ・ [スクリプト エンジン]: チェックを付ける
- ・ [ファイルの存在を確認する]: チェックを付けない

IIS バージョン 6 (Windows 2003 に搭載) に関する詳細は、次のセクションを参照してください。

### 2.1.4.3 Internet Information Services バージョン 6

オプション 1 の指示に従います。ただし、CSP ゲートウェイについては、CSPms.dll および CSPmsSys.dll ではなく、以下の実行可能ファイルを許可されたものとして登録します。

- ・ nph-CSPcgi.exe
- ・ nph-CSPcgiSys.exe

#### CSP へのアクセスを禁止するには

以下の実行可能ファイルを “禁止” に設定します。

- ・ nph-CSPcgi.exe
- ・ nph-CSPcgiSys.exe

#### CSP 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禁止するには

実行可能ファイル nph-CSPcgiSys.exe を “禁止” に設定します。

#### CSP ゲートウェイの実行時モジュールへのアクセスを禁止するには

実行可能ファイル nph-CSPcgi.exe を “禁止” に設定します。

### 2.1.4.4 ゲートウェイの運用と管理

この接続オプションは、CSP ゲートウェイの Network Service Daemon (NSD) を使用します。

- ・ このサービスに関するセクションの説明に従って、CSP NSD を起動します。
- ・ 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。

CSP ページは高性能なモジュール (mod\_csp.so) によって管理されますが、ゲートウェイの管理スイートへアクセスするには専用の CGI モジュール (nph-CSPcgiSys) を使用します。

CSP 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、ブラウザで以下を指定します。

```
http://<ip_address>/csp-bin/nph-CSPcgiSys
```

承認され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、“セキュリティ”を参照してください。

.csp や .cls 拡張子を含むファイルが要求されると、CSP エンジンが自動的に呼び出されます。次に例を示します。

```
http://<ip_address>/csp/samples/menu.csp
```

## 2.2 Netscape、iPlanet、および Sun の Web サーバ

このセクションでは、オリジナルの Netscape Web サーバ (Enterprise および FastTrack) を介して CSP を実行するための構成手順と操作手順を説明します。これらのサーバは、ごく最近まで iPlanet という名前で販売されていましたが、現在では Sun ONE サーバ・スイートとして Sun により開発および販売されています。Sun は Sun ONE Web サーバをさらに改名し、このドキュメントの執筆時点では Sun Java System Web サーバという名前に変更されています。

従来より、Netscape ベースのサーバは、FastTrack と Enterprise という 2 つの形式が作成されています。FastTrack サーバは、Enterprise サーバから一部の機能を省いた軽量バージョンであり、通常は無料で提供されます。インストールと構成の手順は、(CSP に関しては) どちらのサーバも同じです。

このドキュメントは、以下のファイル・システムにインストールされている、さまざまな CSP およびゲートウェイ Web サーバのコンポーネントに基づいて作成されています。

```
C:\cachesys\csp\
```

Web サーバは以下のディレクトリにインストールされていることが前提となっています。

```
C:\Netscape\SuiteSpot\
```

説明をわかりやすくするため、これ以降は “Netscape” という製品名を使用します。ただし、これまで、この製品にはさまざまな名前が付けられていたことを覚えておいてください。例えば、iPlanet サーバは本質的には Netscape Enterprise と同じ製品ですが、以下の場所にインストールされます。

```
C:\iPlanet\SuiteSpot\
```

Sun の最近のサーバ製品は以下の場所にインストールされます。

```
C:\Sun\WebServer\
```

Netscape サーバの個々のインスタンスは、以下のフォームのディレクトリにインストールされます。

```
C:\Netscape\SuiteSpot\https-<server_name>\
```

または

```
C:\Netscape\SuiteSpot\httpd-<server_name>\
```

server\_name は、ホスト・コンピュータに割り当てられた論理名です。

システムのレイアウトが異なる場合は、必要に応じて、この後のセクションで説明する構成指示文を修正してください。

通常、これらのサーバのドキュメント・ルート・ディレクトリは以下のとおりです。

C:\Netscape\SuiteSpot\docs\

## 2.2.1 インストール

以下の CSP 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てください。

### 1. NSD モジュール (必要な場合)

- ・ **CSPnsd.exe**
- ・ **CSPnsdSv.exe**

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:\cachesys\csp

NSD は上記のホーム・ディレクトリ内から実行されます (上記参照)。NSD ベースの接続オプションの場合、構成ファイル (**CSP.INI**) とイベント・ログ (**CSP.LOG**) はこのディレクトリに書き込まれます。

### 2. NSAPI と CGI のモジュール

- ・ **CSPn3.dll** (実行時モジュール)
- ・ **CSPn3Sys.dll** (システム管理モジュール)
- ・ **CSPcn3.dll** (提供される場合は、NSD の NSAPI クライアント)
- ・ **CSPcgi.exe** (実行時モジュール)
- ・ **nph-CSPcgi.exe** (CSPcgi のコピー)
- ・ **CSPcgiSys.exe** (システム管理モジュール)
- ・ **nph-CSPcgiSys.exe** (CSPcgiSys のコピー)

上記すべてのモジュールが各接続オプションで必要になるわけではありません。各オプションについてのセクションを参照し、実際に必要なモジュールを確認してください。

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:\cachesys\csp\bin

NSD ベース以外の接続オプションの場合、構成ファイル (**CSP.INI**) とイベント・ログ (**CSP.LOG**) はこのディレクトリに書き込まれます。

### 3. ハイパーイベントのコンポーネント

- ・ CSPBroker.js
- ・ CSPBBroker.class
- ・ CSPBroker.jar
- ・ 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。

```
C:\cachesys\csp\broker
```

### 4. 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。

```
C:\cachesys\csp\samples
```

## 2.2.2 オプション 1: NSAPI モジュール (CSPn3\*.dll) を使用する

CSP 要求 (ファイル・タイプ .csp および .cls) を認識し、それら进行处理担当の CSP ゲートウェイへ渡すように Web サーバを構成する必要があります。

Web サーバ構成ファイル (**magnus.conf** と **obj.conf**) は、以下のディレクトリに配置されます。

```
C:\Netscape\SuiteSpot\https-<server_name>\config\
```

NSAPI モジュール、および CSP ファイルを認識して処理する手順をロードするための指示文を、Web サーバ構成に追加する必要があります。

### 2.2.2.1 指示文

#### NSAPI モジュールをロードするための指示文

“Init” 指示文は、NSAPI モジュールをロードするよう Web サーバに指示します。使用する Web サーバのバージョンに応じて、コアの **magnus.conf** ファイルまたはオブジェクトの **obj.conf** 構成ファイルにこれらの指示文を追加します。Netscape Web サーバおよび以前のバージョンの iPlanet サーバは、“Init” 指示文を **obj.conf** に格納します。一方、最近の iPlanet サーバおよび Sun サーバは **magnus.conf** ファイルに格納します。2 つのうち、どちらの構成ファイルを使用するかを判断するため、両方を試してみて、Web サーバのコア “Init” 指示文がどこに格納されるかを確認してください。これらのコア構成指示文は、以下のような形式で常に使用されます。

```
Init fn=load-types mime-types=mime.types
```

コア “Init” 指示文のブロックが見つければ、そのブロックの前に以下のセクションを追加します。

```

Init fn=load-modules shlib=C:/cachesys/csp/bin/CSPn3.dll \
    funcs=csp_term,csp_init,csp_req
Init fn=csp_init shlib="C:/cachesys/csp/bin/CSPn3.dll"
Init fn=load-modules shlib= C:/cachesys/csp/bin/CSPn3Sys.dll \
    funcs=csp_term_sys,csp_init_sys,csp_req_sys
Init fn=csp_init_sys shlib="C:/cachesys/csp/bin/CSPn3Sys.dll"

```

“Init” 指示文は 1 行で構成されることに注意してください。ただし表示スペースに制限があるため、上に示す行は、機能宣言 (funcs) の前で折り返されています。

### 静的コンポーネントを見つけるための指示文

CSP には、Web サーバによって管理されるいくつかの“静的”ファイルが含まれます。例えば、Java/JavaScript ファイルは、CSP サンプルで必要となるハイパーイベントとイメージを実装するために使用されます。これらのファイルについては、“インストール”のセクション 3 および 4 をそれぞれ参照してください。

Web サーバは、仮想 CSP ドキュメント・ルート・ディレクトリを基準として、これらのファイルの場所を把握する必要があります。

obj.conf の “default” 指示文セクションを探します。

```
<Object name="default">
```

“default” セクション、つまり上記の行の後に、以下の 2 行を追加します。

```

NameTrans fn="pfx2dir" from="/csp/samples" dir="c:/cachesys/csp/samples"
NameTrans fn="pfx2dir" from="/csp/broker" dir="c:/cachesys/csp/broker"

```

### CSP 要求を認識および処理するための指示文

obj.conf の最後に以下のセクションを追加します。

```

<Object ppath="*/*.csp">
Service method=(GET|HEAD|POST) fn=csp_req
</Object>
<Object ppath="*/*.cls">
Service method=(GET|HEAD|POST) fn=csp_req
</Object>
<Object ppath="*/CSPn3Sys.dll">
Service method=(GET|HEAD|POST) fn=csp_req_sys
</Object>
<Object ppath="*/CSPn3.dll">
Service method=(GET|HEAD|POST) fn=csp_req
</Object>

```

## 2.2.2.2 ゲートウェイの運用と管理

Web サーバの構成ファイル (magnus.conf および obj.conf) を変更したら、その Web サーバを再起動する必要があります。

CSP 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、ブラウザで以下を指定します。

```
http://<ip_address>/csp/bin/CSPn3Sys.dll
```

承認され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、“セキュリティ”を参照してください。

.csp や .cls 拡張子を含むファイルが要求されると、CSP エンジンが自動的に呼び出されます。次に例を示します。

```
http://<ip_address>/csp/samples/menu.csp
```

## 2.2.3 オプション 2: NSD で ISAPI モジュール (CSPcn3.dll) を使用する

ほとんどの環境では、包括的な NSAPI ベースのソリューション (オプション 1) 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 サーバとは別にゲートウェイを管理する必要がある場合には、NSAPI 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ゲートウェイ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、それぞれのゲートウェイ・インスタンスにバインドします。

構成手順は、オプション 1 で説明した手順と非常によく似ています。重要な違いは以下のとおりです。

オプション 1 の説明に従って、Web サーバのコア “Init” 指示文のブロックを探します。これらは **magnus.conf** または **obj.conf** にあります。コア “Init” ブロックの前に、以下のセクションを追加します。

```
Init fn=load-modules shlib=C:/cachesys/csp/bin/CSPcn3.dll \
    funcs=cspc_term,cspc_init,cspc_req
Init fn=cspc_init shlib="C:/cachesys/csp/bin/CSPcn3.dll"
```

“Init” 指示文は 1 行で構成されることに注意してください。ただし表示スペースに制限があるため、上に示す行は、機能宣言 (funcs) の前で折り返されています。

### 2.2.3.1 指示文

#### 静的コンポーネントを見つけるための指示文

**obj.conf** の “default” 指示文セクションを探します。

```
<Object name="default">
```

“default” セクションで、上記の行の後に、以下の行を追加します。

```
NameTrans fn="pfx2dir" from="/csp/samples" dir="c:/cachesys/csp/samples"
NameTrans fn="pfx2dir" from="/csp/broker" dir="c:/cachesys/csp/broker"

NameTrans fn="pfx2dir" from="/csp-bin" \
    dir="c:/cachesys/csp/bin" name="cgi"
```

#### CSP 要求を認識および処理するための指示文

**obj.conf** の最後に以下のセクションを追加します。

```
<Object ppath="*/*.csp">Service method=(GET|HEAD|POST) fn=csp_req
</Object>
<Object ppath="*/*.cls">
Service method=(GET|HEAD|POST) fn=csp_req
</Object>
```

### 2.2.3.2 ゲートウェイの運用と管理

この接続オプションは、CSP ゲートウェイの Network Service Daemon (NSD) を使用します。

- ・ このサービスに関するセクションの説明に従って、CSP NSD を起動します。
- ・ 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。

CSP ページは高性能なモジュール (`mod_csp.so`) によって管理されますが、ゲートウェイの管理スイートへアクセスするには専用の CGI モジュール (`nph-CSPcgiSys`) を使用します。

CSP 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、ブラウザで以下を指定します。

```
http://<ip_address>/csp-bin/nph-CSPcgiSys
```

承認され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、“**セキュリティ**”を参照してください。

.csp や .cls 拡張子を含むファイルが要求されると、CSP エンジンが自動的に呼び出されます。次に例を示します。

```
http://<ip_address>/csp/samples/menu.csp
```

## 2.3 Apache サーバ

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。

Windows ではビルド済みキットが利用できますが、一般にこれは最新の Apache 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。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ともに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。

このドキュメントは、以下のファイル・システムにインストールされている、さまざまな CSP およびゲートウェイ Web サーバのコンポーネントに基づいて作成されています。

```
C:\cachesys\csp\
```

Web サーバは以下のディレクトリにインストールされていることが前提となっています。

```
C:\Program Files\Apache Group\Apache\
```

システムのレイアウトが異なる場合は、必要に応じて、この後のセクションで説明する構成指示文を修正してください。

以下のセクションから、接続オプションを 1 つ選択します。最初は、CGI ベースの方法が最も簡単です。一方、ダイナミック・リンク・モジュールに基づくソリューションはパフォーマンスに優れています。ISAPI ベースのソリューションを使用する場合、Apache サーバを ISAPI モジュールの変更バージョン (mod\_isapi.c) で再構築する必要があります。

## 2.3.1 インストール (すべての接続オプション)

以下の CSP 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てください。

### 1. NSD モジュール (必要な場合)

- ・ **CSPnsd.exe**
- ・ **CSPnsdSv.exe**

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:\cachesys\csp
```

NSD は上記のホーム・ディレクトリ内から実行されます (上記参照)。NSD ベースの接続オプションの場合、構成ファイル (**CSP.INI**) とイベント・ログ (**CSP.LOG**) はこのディレクトリに書き込まれます。

### 2. CGI およびその他のダイナミック・リンク・モジュール

注釈: Apache サーバ API は、バージョン 1.3.x から 2.x へ移行する際に大幅に変更されました。現在では、以下のように、サーバごとに別々のバイナリが存在します。

共通のファイル

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

Apache バージョン 1.3.x

- ・ **mod\_csp.dll** (提供される場合は、DLL として機能する Apache “組み込み” モジュール)
- ・ **CSPap.dll** (提供される場合は、実行モジュール)
- ・ **CSPapSys.dll** (提供される場合は、ゲートウェイのシステム管理モジュール)

Apache バージョン 2.x

- ・ **mod\_csp2.dll** (提供される場合は、DLL として機能する Apache “組み込み” モジュール)
- ・ **CSPa2.dll** (提供される場合は、実行時モジュール)
- ・ **CSPa2Sys.dll** (提供される場合は、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:\cachesys\csp\bin
```

NSD ベース以外の接続オプションの場合、構成ファイル (**CSP.INI**) とイベント・ログ (**CSP.LOG**) 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、CSP 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。

### 3. ハイパーイベントのコンポーネント

- ・ **CSPBroker.js**
- ・ **CSPBBroker.class**
- ・ **CSPBroker.jar**
- ・ **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。

```
C:\cachesys\csp\broker
```

### 4. 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。

```
C:\cachesys\csp\samples
```

## 2.3.2 オプション 1: Apache API モジュール (CSPa\*.dll) を使用する

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。このオプションを目的として NSD を起動する必要はありません。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、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。

**CSPap.dll** (実行時) モジュールと **CSPapSys.dll** (ゲートウェイのシステム管理) モジュールは、対応する ISAPI DLL と同様に機能するよう設計されたダイナミック・リンク・モジュールです。Apache バージョン 2.x では、**CSPa2.dll** (実行時) モジュールと **CSPa2Sys.dll** モジュールを使用します。

CSP 要求 (ファイル・タイプ .csp および .cls) を認識し、それらを処理担当の CSP ゲートウェイ・モジュールへ渡すように Web サーバを構成する必要があります。

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。

```
LoadModule cspsys_module_sa c:/cachesys/csp/bin/CSPapSys.dll
LoadModule csp_module_sa c:/cachesys/csp/bin/CSPap.dll
<Location "/csp/bin/Systems/">
SetHandler cspsys-handler-sa
</Location>
<Location "/csp/bin/RunTime/">
SetHandler csp-handler-sa
</Location>
AddHandler csp-handler-sa csp cls
Alias /csp/ c:/cachesys/csp/
<Directory "c:/cachesys/csp">
    AllowOverride None
    Options MultiViews FollowSymLinks ExecCGI
    Order allow,deny
    Allow from all
    <FilesMatch "\.(log|ini|pid|exe)$">
        Deny from all
    </FilesMatch>
</Directory>
```

Apache バージョン 2.x を使用する場合は、上記の構成ブロックで、(`CSPap.dll` ではなく) `CSPa2.dll`、および (`CSPapSys.dll` ではなく) `CSPa2Sys.dll` を指定してください。

### 2.3.2.1 ゲートウェイの運用と管理

CSP 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、ブラウザで以下を指定します。

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

“cxw” ファイル拡張子を使用することに注意してください。この拡張子は、Apache が、Apache グループの ISAPI インタフェースを使用してこれらの DLL をロードおよび実行しないようにします。また、Apache の場合、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。

承認され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、“[セキュリティ](#)”を参照してください。

.csp や .cls 拡張子を含むファイルが要求されると、CSP エンジンが自動的に呼び出されます。次に例を示します。

```
http://<ip_address>/csp/samples/menu.csp
```

## 2.3.3 オプション 2: NSD で CGI モジュール (nph-CSPcgi\*.exe) を使用する

CSP 要求 (ファイル・タイプ .csp および .cls) を認識し、それらを CSP ゲートウェイに渡して処理するように Web サーバを構成する必要があります。

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])$" >
    AllowOverride None
    Options FollowSymLinks ExecCGI
    Order allow,deny
    Allow from all
</LocationMatch>
ScriptAliasMatch /*\.([Cc][Ss][Pp]|[Cc][Ll][Ss])$ \
    "c:/cachesys/csp/bin/nph-CSPcgi.exe"
Alias /csp/ c:/cachesys/csp/
<Directory "c:/cachesys/csp">
    AllowOverride None
    Options MultiViews FollowSymLinks ExecCGI
    Order allow,deny
    Allow from all
    <FilesMatch "\.(log|ini|pid|exe)$">
        Deny from all
    </FilesMatch>
</Directory>
ScriptAlias /csp-bin/ "c:/cachesys/csp/bin/"
<Directory "c:/cachesys/csp/bin/">
    AllowOverride None
    Options None
    Order allow,deny
    Allow from all
    <FilesMatch "\.(exe)$">
        Allow from all
    </FilesMatch>
</Directory>
```

上記の構成ブロックは、Apache 環境で利用できる Regular Expressions (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 2000 システムで) は、結果として CSP ファイルは提供されません ("ファイルが見つかりません" 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 CSP 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 CSP アプリケーションが "/csp" の下に含まれているとします。CSP CGI モジュールを "/csp" 下のファイルに関連付けるには、以下の構成ブロックを

```
<LocationMatch "/*\.([Cc][Ss][Pp]|[Cc][Ll][Ss])$" >
    AllowOverride None
    Options FollowSymLinks ExecCGI
    Order allow,deny
    Allow from all
</LocationMatch>
ScriptAliasMatch /*\.([Cc][Ss][Pp]|[Cc][Ll][Ss])$
    "c:/cachesys/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
    Order allow,deny
    Allow from all
</Location>
ScriptAlias /csp "c:/cachesys/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://<ip_address>/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://<ip_address>/myapps/*.csp
```

”regex”の問題を回避するには、CSP MIME タイプとの連結で”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のフィルタ技術なので、Cache 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/ c:/cachesys/csp/
<Directory "c:/cachesys/csp">
    AllowOverride None
    Options MultiViews FollowSymLinks ExecCGI
    Order allow,deny
    Allow from all
<Files CSPnsd.exe>
    Deny from all
</Files>
<Files CSP.ini>
    Deny from all
</Files>
<Files CSP.log>
    Deny from all
</Files>
<Files CSPnsd.ini>
    Deny from all
</Files>
<Files CSPnsd.pid>
    Deny from all
</Files>
<FilesMatch "\.(log|ini|pid|exe)$">
    Deny from all
</FilesMatch>
</Directory>
ScriptAlias /csp-bin/ "c:/cachesys/csp/bin/"
<Directory "c:/cachesys/csp/bin/">
    AllowOverride None
    Options None
    Order allow,deny
    Allow from all
</Directory>
Action text/csp "/csp-bin/nph-CSPcgi.exe"
```

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。

### 2.3.3.1 ゲートウェイの運用と管理

この接続オプションは、CSP ゲートウェイの Network Service Daemon (NSD) を使用します。

- ・ このサービスに関するセクションの説明に従って、CSP NSD を起動します。
- ・ 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。

CSP 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、ブラウザで以下を指定します。

```
http://<ip_address>/csp-bin/nph-CSPcgiSys
```

承認され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、“[セキュリティ](#)”を参照してください。

.csp や .cls 拡張子を含むファイルが要求されると、CSP エンジンが自動的に呼び出されます。次に例を示します。

```
http://<ip_address>/csp/samples/menu.csp
```

## 2.3.4 オプション 3: NSD で Apache API モジュール (mod\_csp\*.dll) を使用する

注釈: ほとんどの場合、この接続オプションより、オプション 1 で説明したスタンドアロン API モジュールの方が優先されます。

CSP モジュールが DLL (**mod\_csp.dll**) として提供されている場合、これによって前述の CGI ベースのソリューションよりも高性能なソリューションが提供されます。Apache バージョン 2.x では **mod\_csp2.dll** モジュールを使用してください。

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 .csp と .cls 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 c:/cachesys/csp/bin/mod_csp.dll
<LocationMatch "/*\.( [Cc][Ss][Pp]|[Cc][Ll][Ss])$" >
    SetHandler csp-handler
</LocationMatch>
Alias /csp/ /cachesys/csp/
<Directory "c:/cachesys/csp/" >
    AllowOverride None
    Options MultiViews FollowSymLinks ExecCGI
    Order allow,deny
    Allow from all
    <FilesMatch "\.(log|ini|pid|exe)$" >
        Deny from all
    </FilesMatch>
    <Files CSPnsd >
        Deny from all
    </Files>
</Directory>
ScriptAlias /csp-bin/ "c:/cachesys/csp/bin/"
<Directory "c:/cachesys/csp/bin/" >
    AllowOverride None
    Options None
    Order allow,deny
    Allow from all
</Directory>
```

もちろん、Apache バージョン 2.x を使用する場合は、上記の構成ブロックで (**mod\_csp.dll** ではなく) **mod\_csp2.dll** を指定してください。

上記の構成ブロックは、Apache 環境で利用できる Regular Expressions (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 2000 システムで) は、結果として CSP ファイルは提供されません ("ファイルが見つかりません" というエラーが返されます)。この状況を修復するには、以下の構成ブロックを

```
<LocationMatch "/*\.( [Cc][Ss][Pp]|[Cc][Ll][Ss])$" >
    SetHandler csp-handler
</LocationMatch>
```

以下のブロックと置換します。

```
AddHandler csp-handler csp cls
```

### 2.3.4.1 ゲートウェイの運用と管理

この接続オプションは、CSP ゲートウェイの Network Service Daemon (NSD) を使用します。

- ・ このサービスに関するセクションの説明に従って、CSP NSD を起動します。
- ・ 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。

CSP ページは高性能なモジュール (`mod_csp.dll`) によって管理されますが、ゲートウェイの管理スイートへアクセスするには専用の CGI モジュール (`nph-CSPcgiSys.exe`) を使用します。

CSP 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、ブラウザで以下を指定します。

```
http://<ip_address>/csp-bin/nph-CSPcgiSys.exe
```

承認され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、“[セキュリティ](#)”を参照してください。

`.csp` や `.cls` 拡張子を含むファイルが要求されると、CSP エンジンが自動的に呼び出されます。次に例を示します。

```
http://<ip_address>/csp/samples/menu.csp
```

## 2.3.5 オプション 4: ISAPI モジュール (CSPms\*.dll) を使用する

この接続オプションより、オプション 1 で説明したスタンドアロン API モジュールの方が優先されます。したがって、このオプションは使用しないでください。ここでは、すでにこのオプションが採用されている旧来のシステムを対象に、参考として記載しています。

Apache グループは、Microsoft の ISAPI インタフェースをエミュレートするモジュールを提供しています。このモジュールを使用するように Apache が構成されている場合は、ISAPI 拡張の実行が可能です。ただし、Apache グループの ISAPI インタフェースと Microsoft 社の独自のインタフェースには大きな違いがあります。Apache ISAPI モジュールの最大の難点は、要求を処理するたびに ISAPI 拡張 (DLL) をアンロードすることです。この動作は CSP では許容されません。CSP ゲートウェイが Caché への接続の永続プールを管理するためには、ISAPI DLL がメモリにそのまま格納されている必要があるからです。

CSP に付属の変更済み ISAPI モジュールでは、CSP ゲートウェイの ISAPI 拡張は各要求の間もメモリにロードされたままとなります。この変更によって影響を受けるのはゲートウェイの ISAPI DLL だけです。その他すべての ISAPI DLL は、元の Apache グループの機能に従います。

### 2.3.5.1 Apache 実行可能ファイルの再構築

#### 1. Apache ISAPI モジュール (mod\_isapi.c) のアップグレード

CSP から提供されるバージョンで、Apache グループの ISAPI モジュールを上書きします。

```
C:\Program Files\Apache Group\Apache\src\os\win32\mod_isapi.c
```

#### 2. Apache 実行可能ファイルの再構築

このステップを実行するには、Microsoft C コンパイラ (Microsoft Visual C++) のバージョン 5.0 以降が必要です。

以下のディレクトリに変更します。

```
C:\Program Files\Apache Group\Apache\src\os\win32\mod_isapi.c
```

以下の Apache を構築します。

```
nmake /f Makefile.nt installr INSTDIR=d:\progra~1\apache~1\apache
```

構築プロセスでは多数の警告メッセージが表示されますが、いずれも無視してかまいません。

#### 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。

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

拡張子 .csp と .cls 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。

```
AddHandler isapi-isa dll
AddHandler isapi-isa csp
AddHandler isapi-isa cls
Alias /csp/ /cachesys/csp/
<Directory "c:/cachesys/csp">
    AllowOverride None
    Options MultiViews FollowSymLinks ExecCGI
    Order allow,deny
    Allow from all
    <FilesMatch "\.(log|ini|pid|exe)$">
        Deny from all
    </FilesMatch>
</Directory>
```

### 2.3.5.2 ゲートウェイの運用と管理

CSP 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、ブラウザで以下を指定します。

```
http://<ip_address>/csp/bin/CSPmsSys.dll
```

承認され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、“セキュリティ”を参照してください。

.csp や .cls 拡張子を含むファイルが要求されると、CSP エンジンが自動的に呼び出されます。次に例を示します。

```
http://<ip_address>/csp/samples/menu.csp
```

## 2.4 NSD (Network Service Daemon) の操作

以下の手順で、NSD を開始します。

1. NSD のホーム・ディレクトリに変更します。

```
C:\cachesys\csp
```

2. NSD を開始します。

```
CSPnsd
```

Windows 95 では、コマンド・ウィンドウから対話形式でこのサービスを実行します。Windows の他のバージョンでは、NSD は Windows サービス (CSPnsd.Sv.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。

- ・ Windows 95:

NSD のコマンド・ウィンドウで Control-C を入力

- ・ 他の Windows システム:

```
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・マネージャにおいて以下の記述によって識別されます。

エラーはすべて CSP イベント・ログ ("CSP.log") に記録されます。このファイルは "C:\%cachesys%\csp" ディレクトリに作成され、このディレクトリで管理されます。CSP 構成ファイル "CSP.ini" もこのディレクトリに配置されます。

NSD は以下のファイルも作成します。

C:\cachesys\csp\CSPnsd.ini

これは、"クライアント" に対し、NSD と通信する方法を伝える非常に重要なファイルです。この場合の "クライアント" は、Apache にリンクされている CSP モジュールと、Apache によって呼び出される CSP CGI 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、Apache プロセスがこのファイルを読み込むことができるということも重要です。Apache 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。

# 3

## UNIX、LINUX、および Mac OS X の Web サーバ

このセクションでは、Sun と Apache の Web サーバについて説明します。オリジナルの Netscape Web サーバ (Enterprise および FastTrack) は、ごく最近まで “iPlanet” という名前で販売されていましたが、現在では Sun によって開発および販売されています。

Netscape/iPlanet/Sun の Web サーバも、高性能 API による機能拡張が可能です。NSAPI (Netscape Application Programming Interface) により、UNIX 共有オブジェクト (共有ライブラリ) として実装されたモジュールを使用して Web サーバを拡張できます。

Apache にはいくつかの接続オプションがあります。インストールと構成が最も簡単なのは CGI ベースのソリューションです。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。ただし、後者のオプションでは Apache を再構築する必要があります。

### 3.1 Netscape、iPlanet、および Sun の Web サーバ

このセクションでは、オリジナルの Netscape Web サーバ (Enterprise および FastTrack) を介して CSP を実行するための構成手順と操作手順を説明します。これらのサーバは、ごく最近まで iPlanet という名前で販売されていましたが、現在では Sun ONE サーバ・スイートとして Sun によって開発および販売されています。Sun は Sun ONE Web サーバをさらに改名し、このドキュメントの執筆時点では Sun Java System Web サーバという名前に変更されています。

従来より、Netscape ベースのサーバは、FastTrack と Enterprise という 2 つの形式が作成されています。FastTrack サーバは、Enterprise サーバから一部の機能を省いた軽量バージョンであり、通

常は無料で提供されます。インストールと構成の手順は、(CSP に関しては) どちらのサーバも同じです。

このドキュメントは、以下のファイル・システムにインストールされている、さまざまな CSP Web サーバのコンポーネントに基づいて作成されています。

```
/usr/cachesys/csp/
```

Web サーバは以下のディレクトリにインストールされていることが前提となっています。

```
/usr/netscape/suitespot/ (or /opt/netscape/suitespot/)
```

説明をわかりやすくするため、これ以降は "Netscape" という製品名を使用します。ただし、これまで、この製品にはさまざまな名前が付けられていたことを覚えておいてください。例えば、iPlanet サーバは本質的には Netscape Enterprise と同じ製品ですが、一般に以下の場所にインストールされます。

```
/usr/iplanet/ (or /opt/iplanet/)
```

Sun の最近のサーバ製品は以下の場所にインストールされます。

```
/usr/SUNWwbsvr/ (or /opt/SUNWwbsvr/)
```

Netscape サーバの個々のインスタンスは、以下のフォームのディレクトリにインストールされます。

```
/usr/netscape/suitespot/https-<server_name>/
```

または

```
/usr/netscape/suitespot/httpd-<server_name>/
```

"server\_name" は、ホスト・コンピュータに割り当てられた論理名です。

システムのレイアウトが異なる場合は、必要に応じて、この後のセクションで説明する構成指示文を修正してください。

通常、これらのサーバのドキュメント・ルート・ディレクトリは以下のとおりです。

```
/usr/netscape/suitespot/docs/
```

### 3.1.1 インストール

CSP ゲートウェイのコンポーネントおよび CSP 静的ファイルは以下のようにインストールしてください。

#### 1. NSD モジュール (必要な場合)

```
CSPnsd
```

このモジュールの既定の場所は以下のとおりです。

```
/usr/cachesys/csp
```

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 (**CSP.INI**) とイベント・ログ (**CSP.LOG**) はこのディレクトリに書き込まれます。

## 2. NSAPI と CGI のモジュール

- ・ **CSPn3.so** (実行時モジュール)
- ・ **CSPn3Sys.so** (システム管理モジュール)
- ・ **CSPcn3.so** (提供される場合は、NSD の NSAPI クライアント)
- ・ **CSPcgi.exe** (実行時モジュール)
- ・ **nph-CSPcgi.exe** (CSPcgi のコピー)
- ・ **CSPcgiSys.exe** (システム管理モジュール)
- ・ **nph-CSPcgiSys.exe** (CSPcgiSys のコピー)

上記すべてのモジュールが各接続オプションで必要になるわけではありません。各オプションについてのセクションを参照し、実際に必要なモジュールを確認してください。

これらのモジュールの既定の場所は以下のとおりです。

```
/usr/cachesys/csp/bin
```

NSD ベース以外の接続オプションの場合、構成ファイル (**CSP.INI**) とイベント・ログ (**CSP.LOG**) はこのディレクトリに書き込まれます。

## 3. ハイパーイベントのコンポーネント

- ・ **CSPBroker.js**
- ・ **CSPBBroker.class**
- ・ **CSPBroker.jar**
- ・ **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。

```
/usr/cachesys/csp/broker
```

## 4. 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。

```
/usr/cachesys/csp/samples
```

## 3.1.2 オプション 1: NSAPI モジュール (CSPn3\*.so) を使用する

CSP 要求 (ファイル・タイプ .csp および .cls) を認識し、それらを CSP ゲートウェイに渡して処理するように Web サーバを構成する必要があります。

Web サーバ構成ファイル (`magnus.conf` と `obj.conf`) は、以下のディレクトリに配置されます。

```
/usr/netscape/suitespot/https-<server_name>/config/
```

NSAPI モジュール、および CSP ファイルを認識して処理するための手順をロードする指示文を Web サーバ構成に追加する必要があります。

### 3.1.2.1 指示文

#### NSAPI モジュールをロードするための指示文

”Init” 指示文は、NSAPI モジュールをロードするよう Web サーバに指示します。使用する Web サーバのバージョンに応じて、コアの `magnus.conf` ファイルまたはオブジェクトの `obj.conf` 構成ファイルにこれらの指示文を追加します。Netscape Web サーバおよび以前のバージョンの iPlanet サーバは、”Init” 指示文を `obj.conf` に格納します。一方、最近の iPlanet サーバおよび Sun サーバは `magnus.conf` ファイルに格納します。2 つのうち、どちらの構成ファイルを使用するかを判断するため、両方を試してみて、Web サーバのコア ”Init” 指示文がどこに格納されるかを確認してください。これらのコア構成指示文は、以下のような形式で常に使用されます。

```
Init fn=load-types mime-types=mime.types
```

コア ”Init” 指示文のブロックが見つければ、そのブロックの前に以下のセクションを追加します。

```
Init fn=load-modules shlib=/usr/cachesys/csp/bin/CSPn3.so \
  funcs=csp_term,csp_init,csp_req
Init fn=csp_init shlib="/usr/cachesys/csp/bin/CSPn3.so"
Init fn=load-modules shlib=/usr/cachesys/csp/bin/CSPn3Sys.so \
  funcs=csp_term_sys,csp_init_sys,csp_req_sys
Init fn=csp_init_sys "shlib=/usr/cachesys/csp/bin/CSPn3Sys.so"
```

”Init” 指示文は 1 行で構成されることに注意してください。ただし表示スペースに制限があるため、上に示す行は、機能宣言 (funcs) の前で折り返されています。

#### 静的コンポーネントを見つけるための指示文

CSP には、Web サーバによって管理されるいくつかの ”静的” ファイルが含まれます。例えば、Java/JavaScript ファイルは、CSP サンプルで必要となるハイパーイベントとイメージを実装するために使用されます。これらのファイルについては、”インストール” のセクション 2 および 3 をそれぞれ参照してください。

Web サーバは、仮想 CSP ドキュメント・ルート・ディレクトリを基準として、これらのファイルの場所を把握する必要があります。

`obj.conf` の ”default” 指示文のセクションを探します。

```
<Object name="default">
```

“default” セクション、つまり上記の行の後に、以下の 2 行を追加します。

```
NameTrans fn="pfx2dir" from="/csp/samples" \
           dir="/usr/cachesys/csp/samples"
NameTrans fn="pfx2dir" from="/csp/broker" \
           dir="/usr/cachesys/csp/broker"
```

## CSP 要求を認識および処理するための指示文

obj.conf の最後に以下のセクションを追加します。

```
<Object ppath="*/*.csp">
Service method=(GET|HEAD|POST) fn=csp_req
</Object>
<Object ppath="*/*.cls">
Service method=(GET|HEAD|POST) fn=csp_req
</Object>
<Object ppath="*/CSPn3Sys.so">
Service method=(GET|HEAD|POST) fn=csp_req_sys
</Object>
<Object ppath="*/CSPn3.so">
Service method=(GET|HEAD|POST) fn=csp_req
</Object>
```

### 3.1.2.2 ゲートウェイの運用と管理

Web サーバの構成ファイル (magnus.conf および obj.conf) を変更したら、その Web サーバを再起動する必要があります。

CSP 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、ブラウザで以下を指定します。

```
http://<ip_address>/csp/bin/CSPn3Sys.so
```

承認され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、“セキュリティ”を参照してください。

CSP エンジンは、.csp や .cls 拡張子を含む要求されたファイルに対して、自動的に呼び出されます。次に例を示します。

```
http://<ip_address>/csp/samples/menu.csp
```

### 3.1.3 オプション 2: NSD で NSAPI モジュール (CSPcn3.so) を使用する

ほとんどの環境では、包括的な NSAPI ベースのソリューション (オプション 1) 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 サーバとは別にゲートウェイを管理する必要がある場合には、NSAPI 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ゲートウェイを共有するとします。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、それぞれのゲートウェイ・インスタンスにバインドします。

構成手順は、オプション 1 で説明した手順と非常によく似ています。重要な違いは以下のとおりです。

オプション 1 の説明に従って、Web サーバのコア "Init" 指示文のブロックを探します。これらは `magnus.conf` または `obj.conf` にあります。コア "Init" ブロックの前に、以下のセクションを追加します。

```
Init fn=load-modules shlib=/usr/cachesys/csp/bin/CSPcn3.so \
      funcs=cspc_term,cspc_init,cspc_req
Init fn=cspc_init shlib="/usr/cachesys/csp/bin/CSPcn3.so"
```

"Init" 指示文は 1 行で構成されることに注意してください。ただし表示スペースに制限があるため、上に示す行は、機能宣言 (funcs) の前で折り返されています。

### 3.1.3.1 指示文

#### 静的コンポーネントを見つけるための指示文

`obj.conf` の "default" 指示文セクションを探します。

```
<Object name="default">
```

"default" セクションで、上記の行の後に、以下の行を追加します。

```
NameTrans fn="pfx2dir" from="/csp/samples" \
          dir="/usr/cachesys/csp/samples"
NameTrans fn="pfx2dir" from="/csp/broker" \
          dir="/usr/cachesys/csp/broker"

NameTrans fn="pfx2dir" from="/csp-bin" \
          dir="/usr/cachesys/csp/bin" name="cgi"
```

#### CSP 要求を認識および処理するための指示文

`obj.conf` の最後に以下のセクションを追加します。

```
<Object ppath="*/*.csp">
Service method=(GET|HEAD|POST) fn=csp_req
</Object>
<Object ppath="*/*.cls">
Service method=(GET|HEAD|POST) fn=csp_req
</Object>
```

### 3.1.3.2 ゲートウェイの運用および管理

この接続オプションは、CSP ゲートウェイの Network Service Daemon (NSD) を使用します。

- ・ このサービスに関するセクションの説明に従って、CSP NSD を起動します。
- ・ 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。

CSP ページは高性能なモジュール (`mod_csp.so`) によって管理されますが、ゲートウェイの管理スイートへアクセスするには専用の CGI モジュール (`nph-CSPcgiSys`) を使用します。

CSP 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、ブラウザで以下を指定します。

```
http://<ip_address>/csp-bin/nph-CSPcgiSys
```

承認され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、“セキュリティ”を参照してください。

CSP エンジンは、.csp や .cls 拡張子を含む要求されたファイルに対して、自動的に呼び出されます。次に例を示します。

```
http://<ip_address>/csp/samples/menu.csp
```

## 3.2 Apache サーバ

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。

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とも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 製品 (AIX) を組み込んだ Apache を提供しています。

このドキュメントは、以下のファイル・システムにインストールされている、さまざまな CSP Web サーバのコンポーネントに基づいて作成されています。

```
/usr/cachesys/csp/
```

Web サーバは以下のディレクトリにインストールされていることが前提となっています。

```
/usr/apache/
```

システムのレイアウトが異なる場合は、必要に応じて、この後のセクションで説明する構成指示文を修正してください。

以下のセクションから、接続オプションを 1 つ選択します。最初は、CGI ベースの方法が最も簡単です。一方、動的にリンクされたモジュールや組み込みモジュールを利用したソリューションは、より高い性能を発揮します。

### 3.2.1 インストール (すべての接続オプション)

CSP ゲートウェイのコンポーネントおよび CSP 静的ファイルは以下のようにインストールしてください。

1. NSD モジュール

## CSPnsd

このモジュールの既定の場所は以下のとおりです。

```
/usr/cachesys/csp
```

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。構成ファイル (**CSP.INI**) とイベント・ログ (**CSP.LOG**) は、このディレクトリに記述されます。

## 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\_csp.so** (Apache バージョン 1.3.x – 提供される場合は、DSO として機能する Apache モジュール)
- ・ **mod\_csp2.so** (Apache バージョン 2.x – 提供される場合は、DSO として機能する Apache 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。

```
/usr/cachesys/csp/bin
```

“Sys” が追加されたモジュールは、CSP 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。

## 3. ハイパーイベントのコンポーネント

- ・ **CSPBroker.js**
- ・ **CSPBBroker.class**
- ・ **CSPBroker.jar**
- ・ **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。

```
/usr/cachesys/csp/broker
```

## 4. 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。

```
/usr/cachesys/csp/samples
```

## 3.2.2 オプション 1: NSD で動的 Apache API モジュール (mod\_csp\*.so) を使用する

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 (mod\_csp.so または mod\_csp2.so) として CSP モジュールが付属している場合は、“構成”に関するセクションに進んでください。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。

Apache バージョン 2.x では mod\_csp2.so モジュールを使用します。

この構成処理を開始する前に、ご使用の Apache ビルドに、共有オブジェクトを管理するためのモジュール (mod\_so) が組み込まれていることを確認してください。これを確認するには、以下のコマンドを実行します (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、共有オブジェクト・モジュール (mod\_so) が含まれている必要があります。

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
```

インストールされている Apache 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。

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 “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。

付属のモジュール・ソース・ファイル (mod\_csp.c) は、Apache v1.3.x および Apache v2.x と完全に互換性があります。

### 3.2.2.1 方法 1: "apxs" (APache eXtenSion) ツールを使用し、CSP モジュールを共有オブジェクトとして構築する

```
apxs -c mod_csp.c
```

すべて正常に処理されると、共有ライブラリ "mod\_csp.so" が構築されます。

このドキュメントの指示と一致するよう、必要であれば、このモジュールの名前を `mod_csp2.so` (Apache バージョン 2.x.) に変更してください。または、以下のように、`mod_csp2.so` を直接構築することもできます。

```
apxs -c -o mod_csp2.so mod_csp.c
```

共有オブジェクトによって生成されたバイナリを以下のディレクトリにインストールします。

```
/usr/cachesys/csp/bin
```

### 3.2.2.2 方法 2: CSP モジュールを共有オブジェクトとして手動で構築する

モジュール・ソース・ファイル "mod\_csp.c" を以下のディレクトリにインストールします。

```
/usr/apache/src/modules/extra
```

"src" ディレクトリに戻ります。

```
/usr/apache/src
```

"Configuration" ファイルを編集します。

このファイルの最終部分に、以下のような行があります。

```
# 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_csp.so
```

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
```

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
```

これにより、共有オブジェクト "mod\_csp.so" が以下のディレクトリに生成されます。

```
/usr/apache/src/modules/extra
```

このドキュメントの指示と一致するよう、必要であれば、このモジュールの名前を `mod_csp2.so` (Apache バージョン 2.x.) に変更してください。

`mod_csp.so` (または `mod_csp2.so`) を以下にコピーします。

```
/usr/cachesys/csp/bin
```

### 3.2.2.3 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` と `.cls` 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 /usr/cachesys/csp/bin/mod_csp.so
<LocationMatch "/*\.[Cc][Ss][Pp]|[Cc][Ll][Ss])$" >
    SetHandler csp-handler
</LocationMatch>
Alias /csp/ /usr/cachesys/csp/
<Directory "/usr/cachesys/csp">
    AllowOverride None
    Options MultiViews FollowSymLinks ExecCGI
    Order allow,deny
    Allow from all
    <FilesMatch "\.(log|ini|pid|exe)$">
        Deny from all
    </FilesMatch>
    <Files CSPnsd>
        Deny from all
    </Files>
</Directory>
ScriptAlias /csp-bin/ "/usr/cachesys/csp/bin/"
<Directory "/usr/cachesys/csp/bin/">
    AllowOverride None
    Options None
    Order allow,deny
    Allow from all
</Directory>
```

もちろん、Apache バージョン 2.x を使用する場合は、上記の構成ブロックで (`mod_csp.so` ではなく) `mod_csp2.so` を指定してください。

上記の構成ブロックは、Apache 環境で利用できる Regular Expressions (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 ("ファイルが見つかりません" というエラーが返されます)。この状況を修復するには、以下の構成ブロックを

```
<LocationMatch "/*\.[Cc][Ss][Pp]|[Cc][Ll][Ss])$" >
    SetHandler csp-handler
</LocationMatch>
```

以下のブロックと置換します。

```
AddHandler csp-handler csp cls
```

## ゲートウェイの運用と管理

この接続オプションは、CSP ゲートウェイの Network Service Daemon (NSD) を使用します。

- ・ このサービスに関するセクションの説明に従って、CSP NSD を起動します。
- ・ 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。

CSP ページは高性能なモジュール (`mod_csp.so`) によって管理されますが、ゲートウェイの管理スイートへアクセスするには専用の CGI モジュール (`nph-CSPcgiSys`) を使用します。

CSP 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、ブラウザで以下を指定します。

```
http://<ip_address>/csp-bin/nph-CSPcgiSys
```

承認され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、“セキュリティ”を参照してください。

CSP エンジンには、`.csp` や `.cls` 拡張子を含む要求されたファイルに対して、自動的に呼び出されます。次に例を示します。

```
http://<ip_address>/csp/samples/menu.csp
```

## 3.2.3 オプション 2: NSD で CGI モジュール (`nph-CSPcgi*`) を使用する

CSP 要求 (ファイル・タイプ `.csp` および `.cls`) を認識し、それらを処理担当の CSP ゲートウェイへ渡すように Web サーバを構成する必要があります。

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])$" >
    AllowOverride None
    Options FollowSymLinks ExecCGI
    Order allow,deny
    Allow from all
</LocationMatch>
ScriptAliasMatch /*\.([Cc][Ss][Pp]|[Cc][Ll][Ss])$ \
    "/usr/cachesys/csp/bin/nph-CSPcgi"
Alias /csp/ /usr/cachesys/csp/
<Directory "/usr/cachesys/csp">
    AllowOverride None
    Options MultiViews FollowSymLinks ExecCGI
    Order allow,deny
    Allow from all
    <FilesMatch "\.(log|ini|pid|exe)$">
        Deny from all
    </FilesMatch>
    <Files CSPnsd>
        Deny from all
    </Files>
</Directory>
ScriptAlias /csp-bin/ "/usr/cachesys/csp/bin/"
<Directory "/usr/cachesys/csp/bin/">
    AllowOverride None
    Options None
    Order allow,deny
    Allow from all
</Directory>

```

上記の構成ブロックは、Apache 環境で利用できる Regular Expressions (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（“ファイルが見つかりません” 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 CSP 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 CSP アプリケーションが “/csp” の下に含まれているとします。CSP CGI モジュールと “/csp” 下のファイルを連結させるには、以下の構成ブロックを

```

<LocationMatch "/*\.([Cc][Ss][Pp]|[Cc][Ll][Ss])$" >
    AllowOverride None
    Options FollowSymLinks ExecCGI
    Order allow,deny
    Allow from all
</LocationMatch>
ScriptAliasMatch /*\.([Cc][Ss][Pp]|[Cc][Ll][Ss])$ \
    "/usr/cachesys/csp/bin/nph-CSPcgi"

```

以下のブロックと置換します。

```

<Location "/csp">
    AllowOverride None
    Options FollowSymLinks ExecCGI
    Order allow,deny
    Allow from all
</LocationMatch>
ScriptAlias /csp "/usr/cachesys/csp/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://<ip\_address>/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"` への処理を繰り返します。

```
http://<ip_address>/myapps/*.csp
```

“regex” の問題を回避するには、CSP MIME タイプとの連結で “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のフィルタ技術なので、Cache 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/ /usr/cachesys/csp/
<Directory "/usr/cachesys/csp">
    AllowOverride None
    Options MultiViews FollowSymLinks ExecCGI
    Order allow,deny
    Allow from all
<Files CSPnsd>
    Deny from all
</Files>
<Files CSP.ini>
    Deny from all
</Files>
<Files CSP.log>
    Deny from all
</Files>
<Files CSPnsd.ini>
    Deny from all
</Files>
<Files CSPnsd.pid>
    Deny from all
</Files>
</Directory>
ScriptAlias /csp-bin/ "/usr/cachesys/csp/bin/"
<Directory "/usr/cachesys/csp/bin/">
    AllowOverride None
    Options None
    Order allow,deny
    Allow from all
</Directory>
Action text/csp "/csp-bin/nph-CSPcgi"
```

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。

### 3.2.3.1 ゲートウェイの運用と管理

この接続オプションは、CSP ゲートウェイの Network Service Daemon (NSD) を使用します。

- ・ このサービスに関するセクションの説明に従って、CSP NSD を起動します。
- ・ 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。

CSP 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、ブラウザで以下を指定します。

```
http://<ip_address>/csp-bin/nph-CSPcgiSys
```

承認され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、“セキュリティ”を参照してください。

CSP エンジンは、.csp や .cls 拡張子を含む要求されたファイルに対して、自動的に呼び出されません。次に例を示します。

```
http://<ip_address>/csp/samples/menu.csp
```

## 3.2.4 オプション 3: NSD で組み込み Apache API モジュール (mod\_csp.c) を使用する

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。以下の説明は、Apache のルート・ディレクトリが “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。

### 3.2.4.1 CSP モジュール・ソース・コードを Apache ソース・ディレクトリに追加する

ユーザ定義のコードは、以下のディレクトリに存在します。

```
/usr/apache/src/modules
```

CSP モジュールに対する以下のディレクトリを作成します。

```
/usr/apache/src/modules/csp
```

“/usr/apache/src/modules” ディレクトリで、以下を入力します。

```
mkdir csp
```

CSP モジュールのソース (mod\_csp.c) をこのディレクトリにコピーします。以下のようになります。

```
/usr/apache/src/modules/csp/mod_csp.c
```

### 3.2.4.2 CSP モジュール用の “Makefile” を生成する

次に、“Makefile” を作成します。Makefile には、CSP モジュールのコンパイル方法とリンク方法に関する指示が含まれます。Apache には必ずサンプル・モジュールが用意されています。サンプルの Makefile をコピーし、CSP に合わせて内容を変更してください。

“csp” ディレクトリに移動します。

```
/usr/apache/src/modules/csp
```

以下のように入力して、サンプルの Makefile をコピーします。

```
cp ../example/Makefile* ./
```

使用している Apache のバージョンによって、以下のいずれかのファイルが "CSP" ディレクトリに配置されます。

- ・ **Makefile** – Apache 1.2.x シリーズ
- ・ **Makefile.tmpl** – Makefile テンプレート、Apache 1.3.x シリーズ

該当するファイルを編集し、"mod\_example" となっているすべての箇所を "mod\_csp" に置き換えます。

### 3.2.4.3 CSP モジュールを認識するための Apache 構築プロセスを構成する

1. "src" ディレクトリに戻ります。

```
/usr/apache/src
```

2. "Configuration" ファイルを編集します。

- ・ Apache 1.2.x:

このファイルの最終部分に、以下のような行があります。

```
# Module example_module modules/example/mod_example.o
```

この行の後に、以下の行を追加してください。

```
Module CSP_module modules/csp/mod_csp.o
```

- ・ Apache 1.3.x:

このファイルの最終部分に、以下のような行があります。

```
# 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。

```
AddModule modules/csp/mod_csp.o
```

### 3.2.4.4 "Configure" スクリプトを実行してメイン "Makefile" を構築する

ここで、構築プロセスを構成します。"src" ディレクトリ内で作業していることを確認してください。

```
/usr/apache/src
```

以下のコマンドを入力します。

```
./Configure
```

### 3.2.4.5 Apache 構築の前に - Red Hat Linux ユーザへの特記事項

Red Hat Linux は、事前に Apache が組み込まれた状態で配布されます。Linux インストール手順の一部として Apache をインストールした場合、Linux を起動すると、Apache も自動的に開始されます。このビルド済バージョン用の実行時ファイル・システムのレイアウトは、Apache ソース・コードの既定のレイアウトと多少異なります。これを補うために、Apache C ヘッダ・ファイル `"httpd.h"` を以下のように変更します。

Apache 1.2.x:

```
/usr/apache/src/httpd.h
```

Apache 1.3.x:

```
/usr/apache/src/include/httpd.h
```

Web サーバ・ルート (HTTPD\_ROOT) は、このファイルの中で最初に定義される項目です。Red Hat Linux では、これは以下のとおり定義されます。

```
#define HTTPD_ROOT "/etc/httpd"
```

### 3.2.4.6 Apache 実行可能ファイルの構築

次に、Apache 実行可能ファイルを構築します。`"src"` ディレクトリに移動します。

```
/usr/apache/src
```

以下のコマンドを入力します。

```
make
```

これにより、CSP モジュールの機能が組み込まれた新規の Web サーバ実行可能ファイル (httpd) が生成されます。

以下のコマンドを実行して、CSP モジュールが Apache コアへ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リストアップされます)。

```
./httpd -l
```

次に例を示します。

```
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
```

httpd 実行可能ファイルが "src" ディレクトリに作成されます。正常に動作するようにするため、このファイルを "apache" ディレクトリにコピーします (さらに、このディレクトリで実行します)。

Red Hat Linux の場合は、実行可能ファイル "httpd" を以下にコピーします。

```
/usr/sbin
```

既存のファイルのバックアップを取っておいてください。

### 3.2.4.7 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 と .cls 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。

```

<LocationMatch "/*\.[Cc][Ss][Pp]|[Cc][Ll][Ss])$" >
    SetHandler csp-handler
</LocationMatch>
Alias /csp/ /usr/cachesys/csp/
<Directory "/usr/cachesys/csp">
    AllowOverride None
    Options MultiViews FollowSymLinks ExecCGI
    Order allow,deny
    Allow from all
    <FilesMatch "\.(log|ini|pid|exe)$">
        Deny from all
    </FilesMatch>
    <Files CSPnsd>
        Deny from all
    </Files>
</Directory>
ScriptAlias /csp-bin/ "/usr/cachesys/csp/bin/"
<Directory "/usr/cachesys/csp/bin/">
    AllowOverride None
    Options None
    Order allow,deny
    Allow from all
</Directory>

```

上記の構成ブロックは、Apache 環境で利用できる Regular Expressions (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（“ファイルが見つかりません” というエラーが返されます）。この状況を修復するには、以下の構成ブロックを

```

<LocationMatch "/*\.[Cc][Ss][Pp]|[Cc][Ll][Ss])$" >
    SetHandler csp-handler
</LocationMatch>

```

以下のブロックと置換します。

```
AddHandler csp-handler
```

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行われるモジュールの 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。

### 3.2.4.8 ゲートウェイの運用と管理

この接続オプションは、CSP ゲートウェイの Network Service Daemon (NSD) を使用します。

- ・ このサービスに関するセクションの説明に従って、CSP NSD を起動します。
- ・ 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。

CSP ページは高性能なモジュール (`mod_csp.c`) によって管理されますが、ゲートウェイの管理スイートへアクセスするには専用の CGI モジュール (`nph-CSPcgiSys`) を使用します。

CSP 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、ブラウザで以下を指定します。

```
http://<ip_address>/csp-bin/nph-CSPcgiSys
```

承認され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、“セキュリティ”を参照してください。

CSP エンジンは、.csp や .cls 拡張子を含む要求されたファイルに対して、自動的に呼び出されま  
す。次に例を示します。

```
http://<ip_address>/csp/samples/menu.csp
```

## 3.3 NSD (Network Service Daemon) を操作する

以下のディレクトリに変更します。

```
/usr/cachesys/csp
```

以下のコマンドを実行して、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られている TCP ポート番号 (既定ではポート 7083) が表示さ  
れます。ポートを変更するには、以下のようにサービスを開始します。

```
./CSPnsd <port_no>
```

port\_no は、指定する TCP ポート番号です。

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、システムが起動されたときに呼び出されたスクリプトから、NSD を “静かに” 開始したい場合、以  
下を使用します。

```
/usr/cachesys/csp/CSPnsd -s [port_no]
```

以下のコマンドで、NSD をクローズします。

```
./CSPnsd -stop
```

または、以下を実行します。

```
kill -TERM `cat /usr/cachesys/csp/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、Caché との間のすべての接続を  
正しく終了し、終了前にすべてのシステム・リソースを開放します。“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、イベント・ログ (“CSP.log”) に記録されます。このファイルは “/usr/cachesys/csp”  
ディレクトリに作成され、保存されます。構成ファイル “CSP.ini” もこのディレクトリに配置されます。

NSD は以下のファイルも作成します。

---

```
/usr/cachesys/csp/CSPnsd.ini
```

これは、“クライアント” に対し、NSD と通信する方法を伝える非常に重要なファイルです。ここでは、“クライアント” は、Apache モジュールの内部に含まれた、または Apache に動的にリンクされる CSP モジュールと、Apache によって呼び出された CSP CGI 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、Apache プロセスがこのファイルを読み込むことができるということも重要です。Apache サーバを操作する UNIX ユーザを考慮して、適切な特権を設定してください。



# 4

## Web サーバ構成に関する一般的な問題

### 4.1 付属の CGI モジュールについて

一般に、ほとんどの CSP は、ホスト Web サーバがサポートしている独自仕様の高性能な API を利用します。ただし、一部の接続オプションでは、少なくとも CSP ゲートウェイのシステム管理を実行する場合は、サイズの小さい CGI 実行可能コードが必要になります。このセクションに記載されている注意事項は、オペレーティング・システムと Web サーバのあらゆる組み合わせに適用されます。

提供される CGI モジュールには、先頭に "nph-" が付いているものと、付いていないものがあります。実はこれらのモジュールは、"nph-" 接頭語の有無にかかわらず、まったく同じものです。"nph-" 接頭語は、Web サーバに対して HTTP ヘッダを解析をしないように (非解析ヘッダ) 指示するための CGI 規約です。Web サーバは単にヘッダをブラウザに転送します。URL のコンテキストでは、"nph-" 接頭語は常に小文字で表記する必要があります。

CSP と CSP ベースのアプリケーションは通常、完全な HTTP ヘッダを送信するという役割を果たします。CGI では、ヘッダの最初の行 (要求のステータスを示す行) が、CGI インタフェースのサーバ側での役割を果たしてきました。Web サーバの中には、これに関して厳密なものがあり、Apache もその中の 1 つです。Apache は、アプリケーションから送信された最初のヘッダ行が、要求ステータス行 (例えば HTTP/1.1 200 OK) ではなく、"コンテンツ・タイプ" 行であるものとして処理します。したがって、ヘッダを解析し、"コンテンツ・タイプ" 行が最初に検出されなかった場合はエラーを返します。"nph" バージョンの CGI モジュールを使用すると、この問題が軽減されます。すべての Web サーバがこの規約に厳密なわけではありません。例えば、Microsoft の Web サーバは、"nph" 接頭語のない CGI 実行可能コード (CSPcgi.exe など) から完全な HTTP ヘッダを受け取ることを拒否しません。

## 4.2 ゲートウェイのシステム管理スイートへアクセスする際のセキュリティ

既定では、ゲートウェイのホスト・コンピュータの管理下にあるクライアントのみがシステム管理スイートにアクセスできます。次に例を示します。

```
http://localhost/csp/bin/CSPmsSys.dll
```

または

```
http://127.0.0.1/csp/bin/CSPmsSys.dll
```

つまり、ブラウザ経由で管理フォームへアクセスする場合、そのブラウザは、Web サーバおよび CSP 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。

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、**CSP.ini** の **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。このパラメータは **SYSTEM** セクションにあります。次に例を示します。

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

このパラメータには、ゲートウェイのシステム管理フォームおよび構成フォームへのアクセスを許可するクライアント (IP アドレス) をコンマ (または "+") で区切って指定します。上記の指示文は、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。

ゲートウェイの新規インストールにおいて、対応可能なローカル・ブラウザがない場合は、この構成設定を手動で作成する必要があります。

**System\_Manager** パラメータは、ゲートウェイ管理スイートの既定パラメータ・セクションにある **Systems Manager Machines** 設定と同じです。

ビルド 662.698 以降のゲートウェイでは、**Systems Manager Machines** パラメータのエントリとしてワイルドカードおよび数値の範囲を指定できます。

次に例を示します。

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

これは、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。つまり、以下の記述をより簡単にしたものとなります。

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

ワイルドカードも使用できます。次に例を示します。

```
[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、オペレーティング・システムでこのような指示文を使用することはお勧めしません。

ゲートウェイの管理スイートを保護する手段としてこの方法を使用した場合には、欠点もあることに注意してください。

- ・ この方法では強力なセキュリティを実現できません。Web クライアントのチェックを目的として、クライアントの IP アドレスが CGI 環境変数 REMOTE\_ADDR から取得されます。クライアント IP アドレスはなりすましが可能です。
- ・ 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、すべてのクライアントの IP アドレスがプロキシの IP アドレスへ効果的に変換されます。この場合、Gateway Systems Manager としてプロキシの IP アドレスを指定する (Systems Manager は、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します) か、または (できれば)、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必要があります。

IP 方式は防御の第一線としては有効ですが、ゲートウェイ管理スイート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 には適していません。運用システムについては、ホスト Web サーバの構成によって、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。

## 4.3 Web サーバの既定ページ (“ホームページ”) としての CSP ページの設定

このセクションでは、CSP ページを Web サーバの既定ページ (ホームページ) にするために必要な構成変更について説明します。ここでは、CSP の [サンプル] メニュー (`/csp/samples/menu.csp`) を Web サーバの既定ページにする手順を説明します。Web サーバの既定ページには以下のようにアクセスします。

```
http://<web_server>/
```

CSP ページをこのように設定するときは、埋め込み URL は省略せずに指定する必要があります。埋め込みハイパーリンクに対して相対 URL を使用した場合、ブラウザは、それらの URL が CSP ルートではなく、ドキュメント・ルート・ディレクトリを基準にしているものと解釈します。例えば、[サンプル]メニューがホームページであるとする、[検査]オプションに対する URL は次のようになります。

```
http://<web_server>/csp/samples/inspector.csp
```

相対 URL を使用した場合、このリンクは、次のように間違って解釈されてしまいます。

```
http://<web_server>/inspector.csp
```

### 4.3.1 Peer Web Server

1. [インターネット サービス マネージャ] ウィンドウを開きます。
2. [WWW サービス] をダブル・クリックして、[WWW サービス プロパティ] ウィンドウを表示します。
3. [ディレクトリ] タブをクリックします。
4. [デフォルトドキュメント] に `/csp/samples/menu.csp` を指定します。
5. [デフォルトドキュメントを使用可能にする] チェックボックスにチェックが付いていることを確認します。
6. [適用] クリックして、すべての変更を保存します。
7. 新しい既定ページ (`/csp/samples/menu.csp`) は、Web サーバのドキュメント・ルート・ディレクトリを基準とする場所に存在している必要があります。これが必要になるのは、空のファイルを作成するときだけです。例えば、ドキュメントのルートが `c:\inetpub\wwwroot` の場合は、次のように操作します。

```
cd c:\inetpub\wwwroot
md csp
cd csp
md samples
cd samples
copy con menu.csp
^Z
```

### 4.3.2 Internet Information Services

1. [インターネット サービス マネージャ] ウィンドウを開きます。
2. 左側のウィンドウで、[既定の Web サイト] に移動します。
3. [既定の Web サイト] を右クリックします。メニューから [プロパティ] を選択して、[既定の Web サイトのプロパティ] ウィンドウを表示します。
4. [ディレクトリ] タブをクリックします。
5. [追加] ボタンをクリックして、[既定のドキュメント名の追加] ウィンドウを表示します。
6. [既定のドキュメント名] フィールドにドキュメント名 (`/csp/samples/menu.csp`) を入力します。
7. [OK] ボタンをクリックします。
8. [既定のドキュメントを有効にする] チェックボックスにチェックが付いていることを確認します。

9. [ディレクトリ] タブで、矢印キーを使用して、新しい既定のドキュメント (/csp/samples/menu.csp) をリストの一番上へ移動します。
10. [適用] および [OK] をクリックして、すべての変更を保存して有効にします。
11. 新しい既定ページ (/csp/samples/menu.csp) は、Web サーバのドキュメント・ルート・ディレクトリを基準とする場所に存在している必要があります。これが必要になるのは、空のファイルを作成するときだけです。例えば、ドキュメントのルートが c:\inetpub\wwwroot の場合は、次のように操作します。

```
cd c:\inetpub\wwwroot
md csp
cd csp
md samples
cd samples
copy con menu.csp
^Z
```

### 4.3.3 Netscape、iPlanet、および Sun の Web サーバ

このセクションで説明する手順は、ビルド 663.763 以降のゲートウェイのみを対象としています。

以下の指示文は、obj.conf の既定のセクションで、Netscape サーバのホームページを指定します。

```
NameTrans fn="home-page" path="/csp/samples/menu.csp"
```

ただし、この指示文では、CSP フォーム・メニュー .csp がサーバのホームページになりません。これは、サーバが CSP ゲートウェイへ制御を渡す前に、要求されたページを基準にして環境変数を更新しないためです。ゲートウェイは、/csp/samples/menu.csp ではなく、/ に対する要求を受け取ったと解釈します。Netscape ベースのサーバでは、NSAPI 拡張によってこのシナリオの認識に対応し、要求されたページとそのパスを識別する変数を更新することが要求されます。この動作に対処する方法は次のとおりです。

- obj.conf の既定セクションで CSP ホームページを定義します。

```
NameTrans fn="home-page" path="/csp/samples/menu.csp"
```

- 次のように、CSP ファイルとゲートウェイ・モジュール間のマッピングを指定するセクションに home-page-path 指示文を追加します。

```
<Object ppath="*/*. [Cc][Ss][Pp]">
Service method=(GET|HEAD|POST) fn=csp_req \
home-page-path="/usr/cachesys/csp/samples"
</Object>
<Object ppath="*/*. [Cc][Ll][Ss]">
Service method=(GET|HEAD|POST) fn=csp_req \
home-page-path="/usr/cachesys/csp/samples"
</Object>
<Object ppath="*/CSPn3Sys.so">
Service method=(GET|HEAD|POST) fn=csp_req_sys
</Object>
<Object ppath="*/CSPn3.so">
Service method=(GET|HEAD|POST) fn=csp_req
</Object>
```

Service 指示文内の `home-page-path` プロパティでホームページのパスを指定する必要はありません。ただし、このパスを指定すると、`/csp/samples/menu.csp` を直接要求した場合と同様の値が `PATH_TRANSLATED` 環境変数に渡されます。つまり、以下のようなホームページ (/) の `PATH_TRANSLATED` が返されます。

```
/cachsys/csp/samples/inspector.csp
```

以下は返されません。

```
/csp/samples/inspector.csp
```

## 4.4 Apache サーバ

Apache 構成ファイルで `DirectoryIndex` 指示文を探します。次に例を示します。

```
DirectoryIndex index.html index.html.var
```

リストの先頭に、Web サーバの新しい既定ページを追加します。

```
DirectoryIndex /csp/samples/menu.csp index.html index.html.var
```